

Methods in
Molecular Biology 1958

Springer Protocols



Alexander E. Kister *Editor*

Protein Supersecondary Structures

Methods and Protocols
Second Edition

EXTRAS ONLINE

 Humana Press

METHODS IN MOLECULAR BIOLOGY

Series Editor

John M. Walker

School of Life and Medical Sciences
University of Hertfordshire
Hatfield, Hertfordshire, AL10 9AB, UK

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

Protein Supersecondary Structures

Methods and Protocols

Second Edition

Edited by

Alexander E. Kister

Department of Mathematics, Rutgers University, Piscataway, NJ, USA



Editor

Alexander E. Kister
Department of Mathematics
Rutgers University
Piscataway, NJ, USA

ISSN 1064-3745

ISSN 1940-6029 (electronic)

Methods in Molecular Biology

ISBN 978-1-4939-9160-0

ISBN 978-1-4939-9161-7 (eBook)

<https://doi.org/10.1007/978-1-4939-9161-7>

Library of Congress Control Number: 2019934749

© Springer Science+Business Media, LLC, part of Springer Nature 2019

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors, and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Humana Press imprint is published by the registered company Springer Science+Business Media, LLC part of Springer Nature.

The registered company address is: 233 Spring Street, New York, NY 10013, U.S.A.

Annotation

After the publication of the first edition of *Protein Supersecondary Structures* in 2013, many new interesting works on this subject have appeared, and it became necessary to collect and consider these results. The concept of supersecondary structure (SSS) as conserved combinations of consecutive secondary structure elements in space was proved very useful to clarify the general principles of protein folding, the relationship between amino acid sequences and protein structures, and the other aspects of proteomics. The authors of the volume are eminent experts in the field of protein research and bioinformatics.

Preface

After the publication of the first edition of *Protein Supersecondary Structures* in 2013 [1], many new interesting works on this subject have appeared, and it became necessary to collect and consider these results.

In 1973, Michael Rossmann introduced the concept of supersecondary structure (SSS) as conserved combinations of consecutive secondary structure elements in space such as beta-hairpins, helix hairpins, beta-alpha-beta, the Greek key, and others. The first supersecondary structure—beta-alpha-beta—was found in proteins that bind **nucleotides**. Later, this structural motif was named as the Rossmann fold. Rossmann outlined the main considerations that led him to formulate a new level of protein structural classification in the first chapter of the previous edition (Rossmann M., “Supersecondary structure: A historical perspective”). A modern review of the structural analysis of proteins with the Rossmann fold is provided by Kihara and Shin in Chapter 1 of this volume.

The concepts of secondary and supersecondary structures may be used to clarify the general principles of protein folding. In 1973, Ptitsyn suggested that protein folding could be conceptualized as a hierarchical process, which involves, first, the formation of the secondary structures and then, via several consecutive stages of folding and refolding, the formation of a stable 3D structure [2]. Elaborating on this idea, we can suggest that after helices and parts of beta-sheets are constructed, SSSs are formed in consecutive stages. In this connection, an important question arises: is it always necessary to achieve an accurate, atomic level, description of the protein structure, or is SSS sufficient for most purposes? There is considerable evidence that protein function is accompanied by relatively minor structural fluctuations or conformational changes [3]. Perhaps, important conclusions about the functioning of the protein can be deduced solely from the knowledge of the secondary and supersecondary structure.

There are many approaches for determining secondary structure with a high degree of reliability. Review of the computer algorithms for secondary and supersecondary structure prediction is presented in Chapter 2. Kc with coauthors discusses the trend for developing the two main types of structure prediction: template-based modeling and template-free modeling.

A prerequisite to the successful classification of secondary structures is a reasonably accurate annotation of the secondary structure elements. This may include, for example, the type of element (a helix or a strand); its location in structure; its size; contacts with other secondary structure elements and with other molecules; a number of hydrophobic residues, which participate in hydrophobic core formation; and other parameters. In Chapter 3, Midlick and colleagues address the main principles of annotation and describe the software for the automatic annotation of secondary structure elements in proteins.

Chapter 4 outlines the new developments in secondary and supersecondary structure prediction. Many new methods have been developed over the last several years. This chapter is an extension of the review by Kurgan and his colleagues published in the first edition in 2013. The authors consider a useful method for comparing SSS of proteins with different amino acid sequences.

In Chapter 5, Hoque and coauthors propose an original approach to SSS prediction. Unlike the commonly used methods that entail predicting secondary structure elements first and then putting them together into SSS, the authors predict SSS directly from the amino acid sequence. The main idea behind their approach is to use several machine learning techniques for the analysis of a variety of sequence and structural characteristics, such as secondary structure probabilities and torsion angles. For the final prediction, a stacked generalization technique is used.

Dictionary of protein secondary structures developed by Kabsch and Sander in 1983 has proven to be very useful for the structural analysis of proteins [4]. Kabsch and Sander proposed criteria for determining secondary structures that can be used in pattern recognition process. In Chapter 6, Konagurthu and Lesk, with coauthors, develop the method for compact presentation of SSS, which describes relative orientation and interactions between secondary structure elements. This methodology was used to create a dictionary of supersecondary structures, which may become an important tool for investigating folding patterns.

Despite the many successful predictions of protein structures based on the sequence similarity, the sequence alignment approach has a significant drawback, which is a consequence of its main advantage. This drawback is not related to the known limitation of the method: it works only when similar sequences with known structure are available in PDB, but does not work if query sequence has low similarity with known sequences. Rather, the problem is more conceptual in nature: homology-based methods do not allow one to assess relative structural roles of residues in protein folding. Without this understanding, it is difficult, if not impossible, to develop rational principles for protein engineering. The most direct approach to understanding the structural role of residues involves the analysis of interactions between residues in protein structures. However, the exact accounting of all interaction parameters is an extremely difficult and possibly insoluble problem at present. An alternative way to approach this problem and not lose too much accuracy is to use coarse-graining methods, i.e., to perform a calculation on a simpler system. In Chapter 7, Liwo and his colleagues describe coarse-grained force fields and the potential of mean force that drive protein structure formation.

It is now widely accepted that protein folding process can be described by “funneled free-energy landscapes.” The analogy with mountain terrain is quite appropriate: without a map, it is very challenging to find thermodynamically favorable pathways through different lower energy mountain passes in a multidimensional phase space. A common approach to overcoming these difficulties is to create a set of different structures (“decoys”) and then calculate which of these structures lies at the bottom of the energy funnel. The concept of SSS may be relevant here as well. A sequential analysis of the formation of, first, beta-strands and alpha-helices and, second, SSS and substructures can be used to select a thermodynamically favorable folding pathway since the formation of these substructures will greatly enhance the stability of the overall polypeptide chain conformation. The final result of energy landscape calculation is the structure with the most favorable conformation, which is considered to correspond to protein’s native structure. However, proteins often undergo structural changes to perform biological functions, for example, to interact with other molecules. In order to predict such interactions, it is important to estimate possible structural fluctuations. From the point of view of an energy landscape, the solution should not be a narrow well but a deep and relatively wide valley that corresponds to thermodynamically stable structural states. In Chapter 8, Shehu and her colleagues develop a new approach to this problem.

In Chapter 9, Pires and his colleagues present two methods for predicting effects of mutations on protein stability. The ability to predict how a point mutation affects protein structures would significantly advance our understanding of protein function and the diseases of protein misfolding. The consequences of the mutation can vary greatly depending on the secondary structure elements and type of SSS in which mutation occurs. For example, it was shown recently that beta-strands are more subject to structural changes than alpha-helices as a result of mutations [5].

Many protein domains in common structural folds, such as immunoglobulins, can be represented as a set of symmetrically connected SSSs. In Chapter 10, Youkharibache derives important conclusions drawn from the analysis of this phenomenon. He shows that there exists a relationship between structural symmetry and gene duplication during protein evolution and also a correlation between a geometric arrangement of SSS in domains and function of proteins.

One of the most common SSSs is $\beta\alpha\beta$ which consists of two parallel beta-strands connected by an alpha-helix. The proteins in TIM barrel folds usually consist of eight such SSSs. Each SSS forms a “micro-barrel” with a hydrophobic core. It was shown that this monomeric substructure is itself a stable tertiary substructure. An important feature of TIM barrel protein families is the high degree of structural similarity and low degree of sequence similarity. In Chapter 11, Vadrevu and colleagues consider the sequence and structural features of different $\beta\alpha\beta$ SSSs.

In Chapter 12, Ventura and colleagues discuss a hotly debated topic in proteomics—protein misfolding and amyloid structural formation. The long-term interest in this problem is mainly due to the fact that amyloids are associated with many human diseases known as amyloidoses. Amyloid fibrils with very similar cross-beta-sheet structures can be formed from very different alpha- and beta-proteins. To explain this interesting phenomenon, it has been suggested that even though alpha- and beta-proteins do not have obvious sequence and structural similarities, they share certain common features that predispose conversions of soluble proteins to the insoluble amyloid-like structures. The authors report the discovery of the soft amyloid core, which may provide the valuable insight for understanding the mechanism of amyloid formation and stability.

In order to understand protein folding pathways and to develop protein design methods, it is important to determine at what stage of SSS formation the incomplete structure becomes stable. It can be assumed that this stable substructure—or “pre-structure”—is a structural nucleus of the protein motif and that after the formation of this substructure, other elements of the secondary structure are superadded, resulting in a fully formed protein domain. A very careful description of the single-molecule force spectroscopy method for the discovery of stable substructures and supersecondary structures is presented by Zaldock and Tych in Chapter 13.

Prediction of protein structure can be made using “building-substructure” approach, which is based on the important finding that parts of a protein structure could be modeled using fragments from other proteins with known structures. These fragments are short segments of the peptide backbone. The accuracy and complexity of a prediction model crucially depend on the size and structural characteristics of the fragments. Computational algorithms can be used to assemble the 3D structure of a protein from its constituent supersecondary structural motifs. In Chapter 14, Trevizani and Custodio describe a new method for finding fragments of optimal size and structural features for structure prediction.

Proteins are stable and, at the same time, relatively fragile molecules. Their SSS and functional properties may transform due to slight changes in their sequence or environment,

or contacts with another molecules. Even changing a single residue that is critical for the folding pattern residue can trigger a transformation in structure. A number of proteins which can undergo such transformations have been discovered. They are known as “transformer proteins.” The detection of a method for determining residues, which are mostly responsible for secondary structure transitions and rearrangements in a structure, would be of decisive importance for protein engineering. In Chapter 15, Gerstman and his colleagues describe computational methods that use molecular dynamics simulations to determine the key residues and conditions that underlie protein transformations.

Sequence-structural analysis of transformer proteins is one of the many lines of evidence for the inference that the relative contribution of residues to the stability of the structure differs widely. Residues in a sequence can then provisionally be classified as “key” residues that are conserved in protein evolution because they are mainly responsible for a given fold formation and “supportive residues” that play a secondary role in helping to maintain the stability of the structure. This classification allows one to explain two seemingly contradictory observations: why proteins with very high sequence identity may have very different structures and why proteins from different families may have the same tertiary fold despite very low sequence similarity. Both of these examples could be made compatible with the famous Chothia-Lesk’s rule—“the extent of the structural changes is directly related to the extent of the sequence changes” [6]—so long as we substitute “the extent of key residue changes” for “the extent of the sequence changes.” Proteins with differing sequences but same fold have similar structure because they share most of the key residues, while the majority of their supporting residues are different. Conversely, in proteins with dissimilar structures, but similar sequences, supportive residues are the mostly same, while the key residues are different. One corollary of this classification is that applying the sequence comparison algorithms to predicting protein structures can be problematic in some cases.

Many of the problems in protein structure prediction can be avoided if we represent three-dimensional structure as an arrangement of secondary structure elements in space. Conserved combinations of consecutive secondary structure elements in the whole domain can be considered as a “skeleton” of a domain. Proteins from different folds can have similar skeleton but little sequence homology. These proteins can be used to identify the residues that play decisive role for particular SSS formation and provide insight on an evolution of protein structures and protein sequence families.

Such a simplified description of the tertiary structure can be sufficient for many purposes. This description was used to collect proteins with the same SSS from different families and to identify their common sequence characteristics (Kister, Chapter 16). The essential goals were (1) to determine how the residues should be distributed in the polypeptide chain in order to form a certain secondary structure and (2) where the hydrophobic residues should be distributed across strands and helices in order to create a hydrophobic core of the given SSS. This analysis was performed for one of the most common immunoglobulin SSS.

If the analysis of a sequence-SSS relationship may be simplified by considering mainly the key residues, analysis of the relationship between protein structure and function needs to take into account all residues in the polypeptide chain. In Chapter 17, Izumi considers all torsion angles of a protein backbone to define the conformation of a protein. The biological activity of proteins—in particular, their binding to ligands—is strongly dependent on the conformation of a polypeptide chain and is regulated by conformational changes. To quantify the conformation of proteins, the authors develop a supersecondary structure code that describes combination of helices, strands, and loops in the protein structures.

This structural codification allowed them to compare the conformations of proteins and quantify the conformational changes for the analysis of protein interactions.

In Chapter 18, Rackovsky suggests constructing a polypeptide chain as a sequence of numerical parameters derived from amino acid physical properties. He considers a set of ten properties for each amino acid—its hydrophobicity values, propensity to form secondary structures, and others. To compare the sequences presented in this way, an alignment method was developed that allows efficient determination of the similarity of polypeptide chains for purposes of predicting protein structure.

Protein folding process can be analogized to the process of micelle formation. Like the amphiphilic molecules that form micelles in an aqueous solution, proteins also have polar and nonpolar parts that determine their shape and spatial structure. Taking into account the critical importance of the hydrophobic nucleus for protein structure formation, the problem of the relationship between sequences and structure was reformulated by Rotterman and her colleagues as a problem of the ratio of hydrophobic residues that make up the hydrophobic core of the structure to the total number of residues that make up the supersecondary structure of the protein domain. The relationship between the structure of the hydrophobic nucleus and the SSS is the subject of Chapter 19.

In Chapter 20, Hellman and Schneider apply tryptophan fluorescence measurements to the study of protein folding/unfolding and ligand binding to a protein. The experiments are described systematically, step-by-step. Authors also point out experimental pitfalls, incorrect or misleading interpretations of data, and ways to deal with experimental artifacts. This paper should prove very useful for the many researchers who wish to properly utilize the technique.

In Chapter 21, Moreira and coauthors discuss the mechanism of membrane protein dimerization. The process of dimerization in membranes is an essential phenomenon of cellular physiology: it involves membrane protein folding, conformational stability, and cell receptors that regulate information flow. The key problem in dimer formation is the detection of residues that make up a protein-protein interface. The authors present a step-by-step protocol for computer simulation to identify these residues.

The papers presented in this book illustrate the fruitfulness of SSS classification for a variety of problems in protein science. Yet, another application of the SSS concept involves the analysis of conservative connections of two or more SSSs in proteins, such as two beta-hairpins with the particular mutual orientation—"the interlock." This invariant substructure was found in practically all beta-sheet sandwich proteins [7]. One can regard the substructures such as the interlock as the next hierarchical level in protein structural classification—the "substructure level," intermediate between SSS and tertiary structure. The stable substructures could be considered as a structural nucleus of protein domains. Study of protein substructures could yield important insights into the evolution of proteins and dynamics of protein formation.

The cover page figure is presented by Dr. Philippe Youkharibache: 3D ribbon structure of T-cell surface Ig domain of CD8a, formed by two symmetry-related supersecondary structures (protodomains, see Chapter 10) (structure 2ATP.A, graphics Chimera).

Piscataway, NJ, USA

Alexander E. Kister

References

1. Protein supersecondary structures. In: Kister A (ed) Methods in molecular biology (Methods and protocols), vol. 932. Humana Press
2. Ptitsyn OB (1973) Stages in the mechanism of self-organization of protein molecules. Dokl Akad Nauk SSSR 210:1213–1215 (in Russian)
3. Micheletti C (2013) Comparing proteins by their internal dynamics: exploring structure-function relationships beyond static structural alignments. *Phys Life Rev* 10(1):1–26
4. Kabsch W, Sander C (1983) Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers* 22(12):2577–2637
5. Abrusán G, Marsh JA (2016) Alpha helices are more robust to mutations than beta strands. *PLoS Comput Biol* 12(12):e1005242
6. Chothia C, Lesk AM (1986) The relation between the divergence of sequence and structure in proteins. *EMBO J* 5(4):823–826
7. Kister AE, Finkelstein AV, Gelfand IM (2002) Common features in structures and sequences of sandwich-like proteins. *Proc Natl Acad Sci U S A* 99(22):14137–14141

Contents

<i>Annotation</i>	v
<i>Preface</i>	vii
<i>Contributors</i>	xv
1 55 Years of the Rossmann Fold	1
<i>Woong-Hee Shin and Daisuke Kihara</i>	
2 Advances in Protein Super-Secondary Structure Prediction and Application to Protein Structure Prediction	15
<i>Elijah MacCarthy, Derrick Perry, and Dukka B. KC</i>	
3 Automated Family-Wide Annotation of Secondary Structure Elements	47
<i>Adam Midlik, Ivana Hutařová Váreková, Jan Hutař, Taraka Ramji Moturu, Veronika Navrátilová, Jaroslav Koča, Karel Berka, and Radka Svobodová Váreková</i>	
4 Computational Prediction of Secondary and Supersecondary Structures from Protein Sequences.....	73
<i>Christopher J. Oldfield, Ke Chen, and Lukasz Kurgan</i>	
5 StackSSSPred: A Stacking-Based Prediction of Supersecondary Structure from Sequence.....	101
<i>Michael Flot, Avdesh Mishra, Aditi Sharma Kuchi, and Md Tamjidul Hoque</i>	
6 Information-Theoretic Inference of an Optimal Dictionary of Protein Supersecondary Structures	123
<i>Arun S. Konagurthu, Ramanan Subramanian, Lloyd Allison, David Abramson, Maria Garcia de la Banda, Peter J. Stuckey, and Arthur M. Lesk</i>	
7 Formation of Secondary and Supersecondary Structure of Proteins as a Result of Coupling Between Local and Backbone-Electrostatic Interactions: A View Through Cluster-Cumulant Scope	133
<i>Adam Liwo, Adam K. Sieradzan, and Cezary Czaplewski</i>	
8 Learning Organizations of Protein Energy Landscapes: An Application on Decoy Selection in Template-Free Protein Structure Prediction.....	147
<i>Nasrin Akhter, Liban Hassan, Zahra Rajabi, Daniel Barbará, and Amarda Shehu</i>	
9 Exploring Protein Supersecondary Structure Through Changes in Protein Folding, Stability, and Flexibility	173
<i>Douglas E. V. Pires, Carlos H. M. Rodrigues, Amanda T. S. Albanoz, Malancha Karmakar, Yoochan Myung, Joicymara Xavier, Eleni-Maria Michanetzi, Stephanie Portelli, and David B. Ascher</i>	
10 Protodomains: Symmetry-Related Supersecondary Structures in Proteins and Self-Complementarity	187
<i>Philippe Youkharibache</i>	

11	$\beta\alpha\beta$ Super-Secondary Motifs: Sequence, Structural Overview, and Pursuit of Potential Autonomously Folding $\beta\alpha\beta$ Sequences from $(\beta/\alpha)_8$ /TIM Barrels	221
	<i>Rajasekhar Varma Kadamuri, Shivkumar Sharma Irukuvajjula, and Ramakrishna Vadrevu</i>	
12	Formation of Cross-Beta Supersecondary Structure by Soft-Amyloid Cores: Strategies for Their Prediction and Characterization	237
	<i>M. Rosario Fernández, Irantzu Pallarès, Valentín Iglesias, Jaime Santos, and Salvador Ventura</i>	
13	Stable Substructures in Proteins and How to Find Them Using Single-Molecule Force Spectroscopy	263
	<i>Katarzyna Tych and Gabriel Žoldák</i>	
14	Supersecondary Structures and Fragment Libraries	283
	<i>Raphael Trevizani and Fábio Lima Custódio</i>	
15	Molecular Dynamics Simulations of Conformational Conversions in Transformer Proteins	297
	<i>Bernard S. Gerstman, Prem P. Chapagain, Jeevan GC, and Timothy Steckmann</i>	
16	Sequence Pattern for Supersecondary Structure of Sandwich-Like Proteins	313
	<i>Alexander E. Kister</i>	
17	Homology Searches Using Supersecondary Structure Code	329
	<i>Hiroshi Izumi</i>	
18	Beyond Supersecondary Structure: Physics-Based Sequence Alignment	341
	<i>S. Rackovsky</i>	
19	Secondary and Supersecondary Structure of Proteins in Light of the Structure of Hydrophobic Cores	347
	<i>Mateusz Banach, Leszek Konieczny, and Irena Roterman</i>	
20	Hands On: Using Tryptophan Fluorescence Spectroscopy to Study Protein Structure	379
	<i>Nadja Hellmann and Dirk Schneider</i>	
21	Structural Characterization of Membrane Protein Dimers	403
	<i>António J. Preto, Pedro Matos-Filipe, Panagiotis I. Koukos, Pedro Renault, Sérgio F. Sousa, and Irina S. Moreira</i>	
	<i>Index</i>	437

Contributors

- DAVID ABRAMSON • *Research Computing Centre, University of Queensland, St Lucia, QLD, Australia*
- NASRIN AKHTER • *Department of Computer Science, George Mason University, Fairfax, VA, USA*
- AMANDA T. S. ALBANAZ • *Instituto René Rachou, Fundação Oswaldo Cruz, Rio de Janeiro, Brazil*
- LLOYD ALLISON • *Faculty of Information Technology, Monash University, Clayton, VIC, Australia*
- DAVID B. ASCHER • *Instituto René Rachou, Fundação Oswaldo Cruz, Rio de Janeiro, Brazil; Department of Biochemistry and Molecular Biology, Bio21 Institute, University of Melbourne, Melbourne, VIC, Australia; Department of Biochemistry, University of Cambridge, Cambridge, UK*
- MATEUSZ BANACH • *Department of Bioinformatics and Telemedicine, Jagiellonian University, Medical College, Kraków, Poland*
- DANIEL BARBARA • *Department of Computer Science, George Mason University, Fairfax, VA, USA*
- KAREL BERKÁ • *Faculty of Science, Department of Physical Chemistry, Regional Centre of Advanced Technologies and Materials, Palacký University, Olomouc, Czech Republic*
- PREM P. CHAPAGAIN • *Department of Physics, Florida International University, Miami, FL, USA*
- KE CHEN • *School of Computer Science and Software Engineering, Tianjin Polytechnic University, Tianjin, People's Republic of China*
- FÁBIO LIMA CUSTÓDIO • *LNCC—Laboratório Nacional de Computação Científica, Petrópolis, Brazil*
- CEZARY CZAPLEWSKI • *Faculty of Chemistry, University of Gdańsk, Gdańsk, Poland*
- MARIA GARCIA DE LA BANDA • *Faculty of Information Technology, Monash University, Clayton, VIC, Australia*
- M. ROSARIO FERNÁNDEZ • *Institut de Biotecnología i de Biomedicina, Universitat Autònoma de Barcelona, Barcelona, Spain; Departament de Bioquímica i Biologia Molecular, Universitat Autònoma de Barcelona, Barcelona, Spain*
- MICHAEL FLOT • *Department of Computer Science, University of New Orleans, New Orleans, LA, USA*
- JEEVAN GC • *Department of Physics, Florida International University, Miami, FL, USA*
- BERNARD S. GERSTMAN • *Department of Physics, Florida International University, Miami, FL, USA*
- LIBAN HASSAN • *Department of Computer Science, George Mason University, Fairfax, VA, USA*
- NADJA HELLMANN • *Institute for Pharmacy and Biochemistry, Johannes Gutenberg-University Mainz, Mainz, Germany*
- MD TAMJIDUL HOQUE • *Department of Computer Science, University of New Orleans, New Orleans, LA, USA*

JAN HUTAŘ • CEITEC—Central European Institute of Technology, Masaryk University, Brno, Czech Republic; Faculty of Science, National Centre for Biomolecular Research, Masaryk University, Brno, Czech Republic

IVANA HUTAŘOVÁ VÁREKOVÁ • CEITEC—Central European Institute of Technology, Masaryk University, Brno, Czech Republic; Faculty of Science, National Centre for Biomolecular Research, Masaryk University, Brno, Czech Republic; Faculty of Informatics, Masaryk University, Brno, Czech Republic

VALENTÍN IGLESIAS • Institut de Biotecnologia i de Biomedicina, Universitat Autònoma de Barcelona, Barcelona, Spain; Departament de Bioquímica i Biologia Molecular, Universitat Autònoma de Barcelona, Barcelona, Spain

SHIVKUMAR SHARMA IRUKUVAJJULA • Department of Biological Sciences, Birla Institute of Technology and Science-Pilani, Hyderabad, Telangana, India

HIROSHI IZUMI • National Institute of Advanced Industrial Science and Technology (AIST), AIST Tsukuba West, Ibaraki, Japan

RAJASEKHAR VARMA KADAMURI • Department of Biological Sciences, Birla Institute of Technology and Science-Pilani, Hyderabad, Telangana, India

MALANCHA KARMAKAR • Department of Biochemistry and Molecular Biology, Bio21 Institute, University of Melbourne, Melbourne, VIC, Australia

DUKKA B. KC • Department of Computational Science and Engineering, North Carolina A&T State University, Greensboro, NC, USA

DAISUKE KIHARA • Department of Biological Science, Purdue University, West Lafayette, IN, USA; Department of Computer Science, Purdue University, West Lafayette, IN, USA

ALEXANDER E. KISTER • Department of Mathematics, Rutgers University, Piscataway, NJ, USA

JAROSLAV KOČA • CEITEC—Central European Institute of Technology, Masaryk University, Brno, Czech Republic; Faculty of Science, National Centre for Biomolecular Research, Masaryk University, Brno, Czech Republic

ARUN S. KONAGURTHU • Faculty of Information Technology, Monash University, Clayton, VIC, Australia

LESZEK KONIECZNY • Chair of Medical Biochemistry, Jagiellonian University, Medical College, Kraków, Poland

PANAGIOTIS I. KOUKOS • Faculty of Science-Chemistry, Bijvoet Center for Biomolecular Research, Utrecht University, Utrecht, The Netherlands

ADITI SHARMA KUCHI • Department of Computer Science, University of New Orleans, New Orleans, LA, USA

LUKASZ KURGAN • Department of Computer Science, College of Engineering, Virginia Commonwealth University, Richmond, VA, USA

ARTHUR M. LESK • Department of Biochemistry and Molecular Biology, Pennsylvania State University, University Park, PA, USA

ADAM LIWO • Faculty of Chemistry, University of Gdańsk, Gdańsk, Poland

ELIJAH MACCARTHY • Department of Computational Science and Engineering, North Carolina A&T State University, Greensboro, NC, USA

PEDRO MATOS-FILIPE • Centro de Neurociências e Biologia Celular, UC - Biotech, Cantanhede, Portugal

ELENI-MARIA MICHAETZI • Department of Biochemistry and Molecular Biology, Bio21 Institute, University of Melbourne, Melbourne, VIC, Australia

- ADAM MIDLIK • CEITEC—Central European Institute of Technology, Masaryk University, Brno, Czech Republic; Faculty of Science, National Centre for Biomolecular Research, Masaryk University, Brno, Czech Republic
- AVDESH MISHRA • Department of Computer Science, University of New Orleans, New Orleans, LA, USA
- IRINA S. MOREIRA • Centro de Neurociências e Biologia Celular, UC - Biotech, Cantanhede, Portugal; Faculty of Science-Chemistry, Bijvoet Center for Biomolecular Research, Utrecht University, Utrecht, The Netherlands
- TARAKA RAMJI MOTURU • CEITEC—Central European Institute of Technology, Masaryk University, Brno, Czech Republic
- YOOCHAN MYUNG • Department of Biochemistry and Molecular Biology, Bio21 Institute, University of Melbourne, Melbourne, VIC, Australia
- VERONIKA NAVRÁTILOVÁ • Faculty of Science, Department of Physical Chemistry, Regional Centre of Advanced Technologies and Materials, Palacký University, Olomouc, Czech Republic
- CHRISTOPHER J. OLDFIELD • Department of Computer Science, College of Engineering, Virginia Commonwealth University, Richmond, VA, USA
- IRANTZU PALLARÈS • Institut de Biotecnologia i de Biomedicina, Universitat Autònoma de Barcelona, Barcelona, Spain; Departament de Bioquímica i Biologia Molecular, Universitat Autònoma de Barcelona, Barcelona, Spain
- DERRICK PERRY • Department of Computational Science and Engineering, North Carolina A&T State University, Greensboro, NC, USA
- DOUGLAS E. V. PIRES • Instituto René Rachou, Fundação Oswaldo Cruz, Rio de Janeiro, Brazil; Department of Biochemistry and Molecular Biology, Bio21 Institute, University of Melbourne, Melbourne, VIC, Australia
- STEPHANIE PORTELLI • Department of Biochemistry and Molecular Biology, Bio21 Institute, University of Melbourne, Melbourne, VIC, Australia
- ANTÓNIO J. PRETO • Centro de Neurociências e Biologia Celular, UC - Biotech, Cantanhede, Portugal
- S. RACKOVSKY • Department of Chemistry and Chemical Biology, Cornell University, Ithaca, NY, USA; Department of Pharmacological Sciences, Icahn School of Medicine at Mount Sinai, New York, NY, USA
- ZAHRA RAJABI • Department of Computer Science, George Mason University, Fairfax, VA, USA
- PEDRO RENAULT • Centro de Neurociências e Biologia Celular, UC - Biotech, Cantanhede, Portugal
- CARLOS H. M. RODRIGUES • Department of Biochemistry and Molecular Biology, Bio21 Institute, University of Melbourne, Melbourne, VIC, Australia
- IRENA ROTERMAN • Department of Bioinformatics and Telemedicine, Jagiellonian University, Medical College, Kraków, Poland
- JAIME SANTOS • Institut de Biotecnologia i de Biomedicina, Universitat Autònoma de Barcelona, Barcelona, Spain; Departament de Bioquímica i Biologia Molecular, Universitat Autònoma de Barcelona, Barcelona, Spain
- DIRK SCHNEIDER • Institute for Pharmacy and Biochemistry, Johannes Gutenberg-University Mainz, Mainz, Germany
- AMARDA SHEHU • Department of Computer Science, George Mason University, Fairfax, VA, USA

WOONG-HEE SHIN • *Department of Biological Science, Purdue University, West Lafayette, IN, USA*

ADAM K. SIERADZAN • *Faculty of Chemistry, University of Gdańsk, Gdańsk, Poland*

SÉRGIO F. SOUSA • *UCIBIO@REQUIMTE, BioSIM, Departamento de Biomedicina, Faculdade de Medicina da Universidade do Porto, Porto, Portugal*

TIMOTHY STECKMANN • *Department of Physics, Florida International University, Miami, FL, USA*

PETER J. STUCKEY • *Faculty of Information Technology, Monash University, Clayton, VIC, Australia; Department of Computing and Information Systems, University of Melbourne, Parkville, VIC, Australia*

RAMANAN SUBRAMANIAN • *Faculty of Information Technology, Monash University, Clayton, VIC, Australia*

RADKA SVOBODOVÁ VÁŘEKOVÁ • *CEITEC—Central European Institute of Technology, Masaryk University, Brno, Czech Republic; Faculty of Science, National Centre for Biomolecular Research, Masaryk University, Brno, Czech Republic*

RAPHAEL TREVIZANI • *Fiocruz—Fundação Oswaldo Cruz, Eusébio, Brazil*

KATARZYNA TYCH • *Physics Department E22, Technical University of Munich, Garching, Germany*

RAMAKRISHNA VADREVU • *Department of Biological Sciences, Birla Institute of Technology and Science-Pilani, Hyderabad, Telangana, India*

SALVADOR VENTURA • *Institut de Biotecnologia i de Biomedicina, Universitat Autònoma de Barcelona, Barcelona, Spain; Departament de Bioquímica i Biologia Molecular, Universitat Autònoma de Barcelona, Barcelona, Spain*

JOICYMARA XAVIER • *Instituto René Rachou, Fundação Oswaldo Cruz, Rio de Janeiro, Brazil*

PHILIPPE YOUNKHARIBACHE • *National Cancer Institute, NIH, Bethesda, MD, USA*

GABRIEL ŽOLDÁK • *Center for Interdisciplinary Biosciences, Technology and Innovation Park, P.J. Šafárik University, Košice, Slovakia*



Chapter 1

55 Years of the Rossmann Fold

Woong-Hee Shin and Daisuke Kihara

Abstract

The Rossmann fold is one of the most commonly observed structural domains in proteins. The fold is composed of consecutive alternating β -strands and α -helices that form a layer of β -sheet with one (or two) layer(s) of α -helices. Here, we will discuss the Rossmann fold starting from its discovery 55 years ago, then overview entries of the fold in the major protein classification databases, SCOP and CATH, as well as the number of the occurrences of the fold in genomes. We also discuss the Rossmann fold as an interesting target of protein engineering as the site-directed mutagenesis of the fold can alter the ligand-binding specificity of the structure.

Key words Protein fold, Rossmann fold, Protein structure classification, Nucleotide-binding fold, CATH, SCOP, Protein engineering

1 Introduction

The Rossmann fold was originally called the nucleotide-binding fold by Michael Rossmann, who was first to discover it. It was later found to be present in many proteins of various functions and across many organisms. In this article, we review the first structures of this fold found 55 years ago and then examine how the population of this fold has grown in protein structure databases over the last 55 years. We also estimate the number of the Rossmann folds in various genomes through a database of homology models of proteins. At last, we discuss the Rossmann fold as an interesting protein design target.

2 The First Discovery of the Rossmann Fold

The Rossmann fold is the only domain structure that was named after its discoverer in the protein structure classification databases, CATH [1] and SCOP [2]. Dr. Michael Rossmann is one of the pioneers of structural biology. Besides the fold we will discuss here,

Dr. Rossmann is also well-known as the developer of the commonly used technique molecular replacement, which is used to solve the phase problem in X-ray crystallography. He has also solved many structures including hemoglobin and virus capsids. He started his research on crystallography of small molecules when he was a graduate student at the University of Glasgow, Scotland. Then he solved the structure of hemoglobin with Dr. Max Perutz at the University of Cambridge [3]. After he joined Purdue University in 1964 as a faculty member, he solved the structure of lactate dehydrogenase [4] in 1970, which would later be known to contain a Rossmann fold. This was the first protein structure that he solved at Purdue. A picture of the protein wire model he built at that time is shown in Fig. 1a. Figure 1b is the structure of lactate dehydrogenase in the PDB database (PDB ID: 3LDH). After the structure of lactate dehydrogenase was solved, a series of dehydrogenase structures were solved and published, including apo structures of lactate dehydrogenase [5] and D-glyceraldehyde-3-phosphate dehydrogenase [6]. From the series of structures, Dr. S.T. Rao and Rossmann identified a common set of the secondary structure elements in dinucleotide-binding proteins, which was the discovery of the Rossmann fold [7]. They compared crystal structures of a lactate dehydrogenase [5], two flavodoxin structures (one from Dr. Martha Ludwig of University of Michigan and the other from Dr. Lyle Jensen of University of Washington), subtilisin (PDB ID: 1SBT), and malate dehydrogenase (from Dr. Leonard J. Banaszak of Washington University of St. Louis) by minimizing the position difference of C α -atoms after superimposition of the structures. They found a common structural motif that is composed of alternating three β -strands and two α -helices, i.e., a $\beta\alpha\beta\alpha\beta$ structure. In a lactate dehydrogenase structure, 60 residues are involved in this super-secondary structure. The three strands are paired in parallel, connected by helices. The loops that connect the C-terminus of β -strand and the N-terminus of α -helix construct a hydrophobic pocket, specifically formed by Val27, which is at the loop between the first α -helix and the first β -strand, and Val54, which is on the loop between the second α -helix and the second β -strand. These two valines interact with the adenine ring of nicotinamide adenine dinucleotide (NAD) or other substrates (Fig. 1c).

Currently in the SCOP database [2], the Rossmann fold is defined as a repeat of the $\beta\alpha\beta\alpha\beta$ structure, which has an approximate twofold symmetry. The six β -strands form a sheet in an order of 321456 counted from the N-terminus to the C-terminus (Fig. 2). The repeated fold binds to a dinucleotide, a coenzyme that contains two distinct nucleotides, such as NAD or flavin adenine dinucleotide (FAD). In 1982, Schulz, Schirmer, and Pai found that a tight loop that connects the first β -strand and α -helix has a conserved sequence GXGXXG in FAD-binding domains [8] by



Fig. 1 The tertiary structure of lactate dehydrogenase. (a) A physical wire model, which is exhibited in the Hockmeyer Hall of Purdue University. (b) The structure in PDB with the oldest deposited date (PDB ID: 3LDH, deposited in 1977). The purple region is the $\beta\alpha\beta\alpha\beta$ structure that is identified by Rao and Rossmann. (c) The NAD-binding site (3LDH). Val27 and Val54, which form a hydrophobic pocket, are colored in red. NAD is shown in a ball-and-stick representation

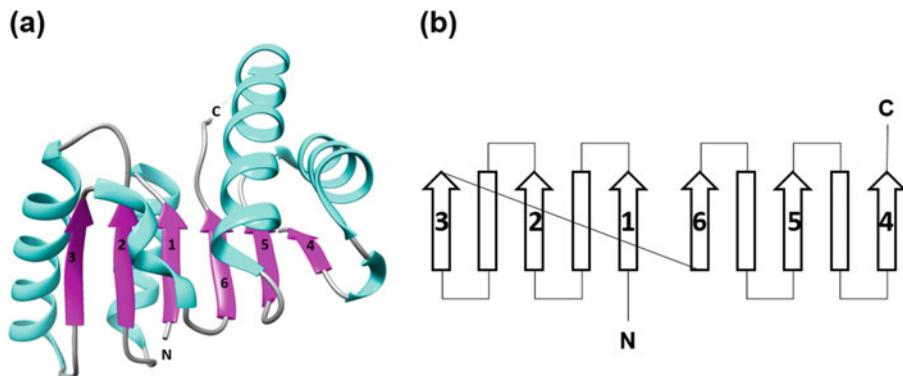


Fig. 2 The Rossmann fold definition in the SCOP database. (a) An example of the Rossmann fold (PDB ID: 2D4A). Helices are colored in cyan and strands are in magenta. (b) The two-dimensional model of the Rossmann fold. Arrows and squares are β -strands and α -helices, respectively. The numbering on the strands starts from the N-terminus of the protein

comparing three homologous protein sequences (pig heart lipoamide dehydrogenase, *p*-hydroxybenzoate hydroxylase, and D-amino acid oxidase). The loop with the consensus sequence makes a contact with negatively charged oxygens of two phosphate groups in both NAD-binding protein and FAD-binding proteins (Fig. 3) [9]. Israel Hanukoglu found that in nicotinamide adenine dinucleotide phosphate (NADP)-binding crystal structure, the last glycine of the loop is mutated to alanine [10]. It was revealed that the position determines binding coenzyme specificity by Purham and his colleagues [11].

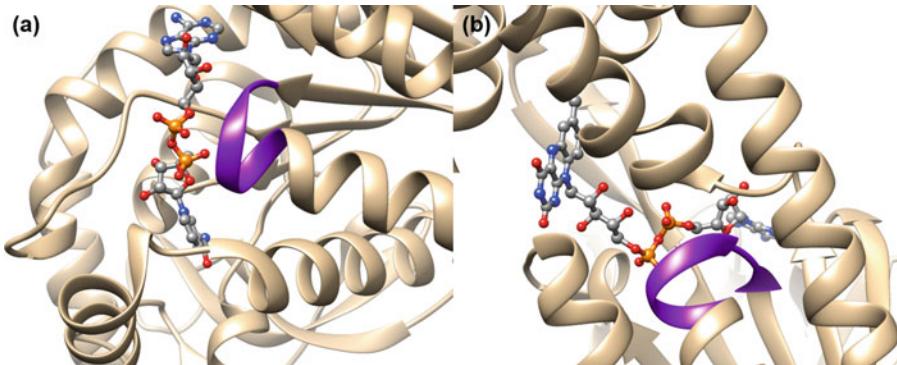


Fig. 3 Crystal structures of Rossmann fold proteins binding with cofactors. **(a)** NAD (PDB ID: 1I0Z); **(b)** FAD (PDB ID: 3GRS). The conserved GXGXXG motif is colored in purple, and the cofactors are represented in a ball-and-stick model

3 Database Entries of Structures of the Fold

Since the first lactate dehydrogenase solved in 1970, a number of Rossmann fold structures have been determined. We examined Rossmann fold entries in the SCOPe (a successor of SCOP) [12] and CATH [1] databases. The Rossmann fold is classified as one of the folds in SCOPe while as a topology in CATH.

SCOP classifies protein domain structures in a hierarchical level, class, fold, superfamily, and family. The latter two are sequence-based classifications. In SCOPe, Rossmann fold, named as “NAD(P)-binding Rossmann-fold domains,” is classified as one of a fold (c.2) under “ α and β proteins (α/β)” class, class c. The fold (c.2.1) has only one superfamily, with the same name as the fold. The condition for classifying domains as Rossmann fold is “three layers, $\alpha/\beta/\alpha$; parallel beta-sheet of six strands, order 321456.” The Rossmann fold is the fourth largest population in SCOPe in the fold level (5985 domains out of 246,157 domains, 2.4%) following immunoglobulin-like β -sandwich (b.1, 7.2%), TIM β/α -barrel (c.1, 4.5%), and N-terminal hydrolase-like (d.153, 2.6%). In the superfamily level, it is the third (2.4%) following immunoglobulin (b.1.1, 5.7%) and N-terminal nucleophile hydrolases (d.153.1, 2.6%) (classes of low-resolution protein structures, peptides, designed proteins, and artifacts were not considered in these statistics).

The Rossmann superfamily is composed of 13 families (proteins in the same family share sequence identity $>30\%$ or a lower sequence identity but perform the same function). Structures of 12 families are shown in Fig. 4a, except for c.2.1.0, classified as “not a true family” whose structures are classified as a Rossmann fold superfamily, but cannot be categorized to any of families by

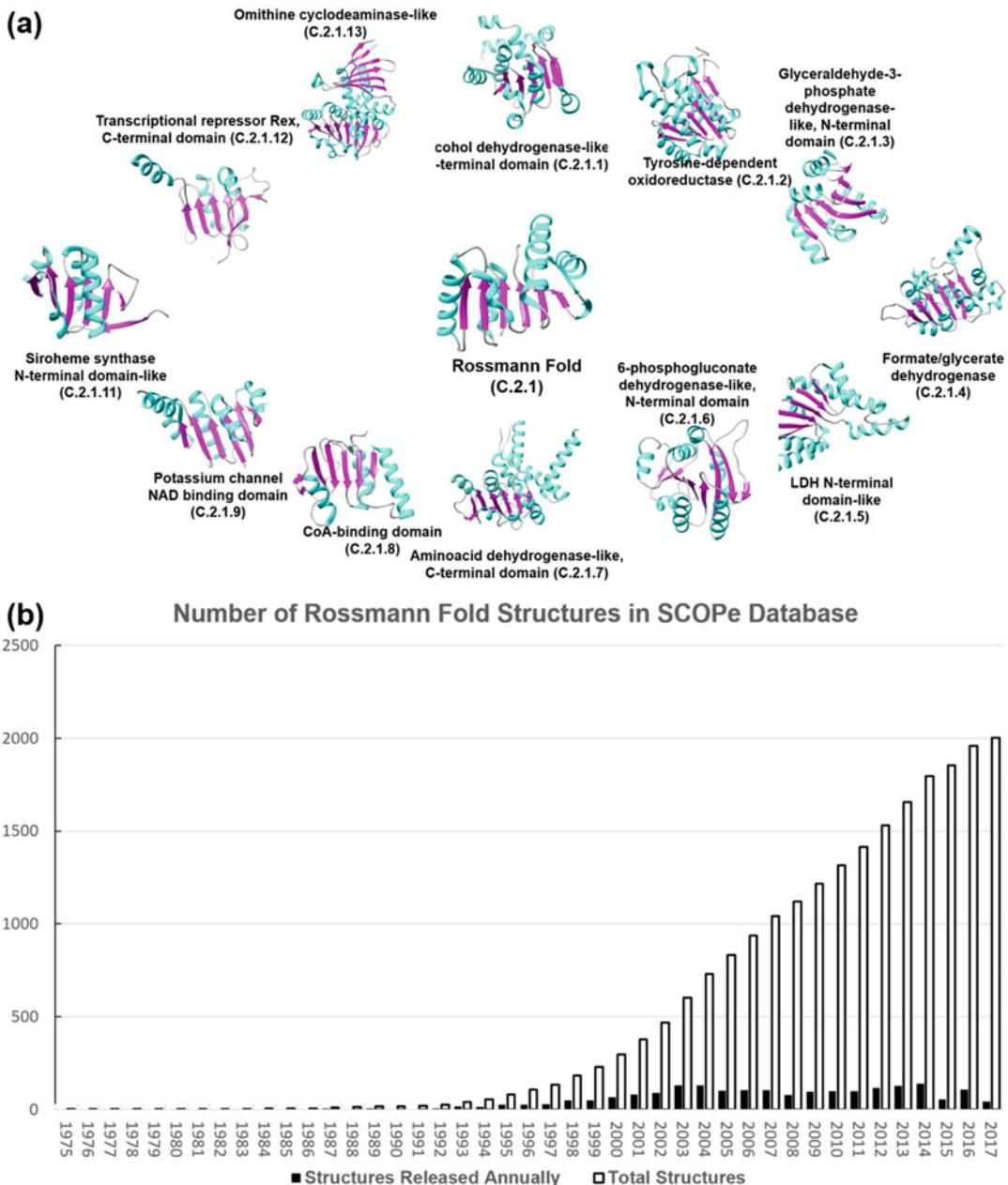


Fig. 4 The Rossmann fold in the SCOPe database. **(a)** Structures of 12 families of the Rossmann fold superfamily in SCOPe. Helices are colored in cyan, and strands are shown in magenta. **(b)** The growth of PDB structures of the Rossmann fold in the SCOPe database. Black bars are the number of PDB structures released in that year, and white bars are the total number of the structures in the database

sequence identity or protein function. Although all families share 321456 β -sheet as a core structure, there is a small difference in each family. For example, family c.2.1.2, tyrosine-dependent oxidoreductases, has an extra parallel seventh strand, forming a β -sheet order of 3214567. On the other hand, for family c.2.1.6, 6-phosphogluconate dehydrogenase-like, N-terminal domain, β -sheet is extended to eight strands with an order of 32145678, and the extra strands are antiparallel to the core six strands. The growth of PDB structures of the Rossmann fold in SCOPe released every year is plotted in Fig. 4b. As of the SCOPe version 2.07 (2018-09-06), the Rossmann fold superfamily has 2002 PDB structures.

In the SCOPe database, there are six folds that are classified as “Rossmann-like folds.” They are nucleotide-binding domain (c.4), MurCD N-terminal domain (c.5), nucleoside phosphorylase/phosphoribosyltransferase catalytic domain (c.27), cytochrome/phosphorylase N-terminal domain (c.28), DHS-like NAD-/FAD-binding domain (c.31), and GckA/TtuD-like (c.118). The difference between the Rossmann fold and the Rossmann-like fold is the number of β -strands that form the β -sheet. The Rossmann-like folds have five parallel strands with an order of 32145, a combination of $\beta\alpha\beta\alpha\beta$ structure and $\beta\alpha\beta$ structure. Just as in the Rossmann fold, the Rossmann-like fold contains the tight loop that connects the first strand and the first helix of $\beta\alpha\beta\alpha\beta$ motif interacts with negative oxygens of phosphate of a ligand molecule. Their structures are shown in Fig. 5a. As of the SCOPe version 2.07 (2018-09-06), 347 PDB structures belong to the six Rossmann-like fold superfamilies. The growth of the structures of the Rossmann-like fold is shown in Fig. 5b.

CATH is another protein structure classification database [1]. CATH classifies structures into a slightly different hierarchy than SCOP, which has four main levels: Class (C), which concerns the secondary structure content, Architecture (A), which concerns the spatial arrangement of the secondary structure elements, Topology (T), which corresponds to the fold level in SCOP, and Homology (H), a sequence identity-based classification where proteins need to have over 25% sequence identity to be in the same H classification. C, T, and H levels are automatically assigned by a protein structure comparison algorithm by Michie et al. [13] and the SSAP program [14]. Architecture is manually assigned after C, T, and H levels are determined. In CATH, the Rossmann fold is classified in the $\alpha\beta$ class (labeled as 3), the architecture class of 40, which is three-layer ($\alpha\beta\alpha$) sandwich, and the topology class of 50; thus, the classification ID is 3.40.50. Comparing with CATH and SCOPe classification of Rossmann fold, the Rossmann fold topology in CATH includes not only the Rossmann fold in SCOPe, but also the Rossmann-like folds. CATH classifies the domain to 3.40.50 if the $\beta\alpha\beta\alpha\beta$ motif is included in the protein

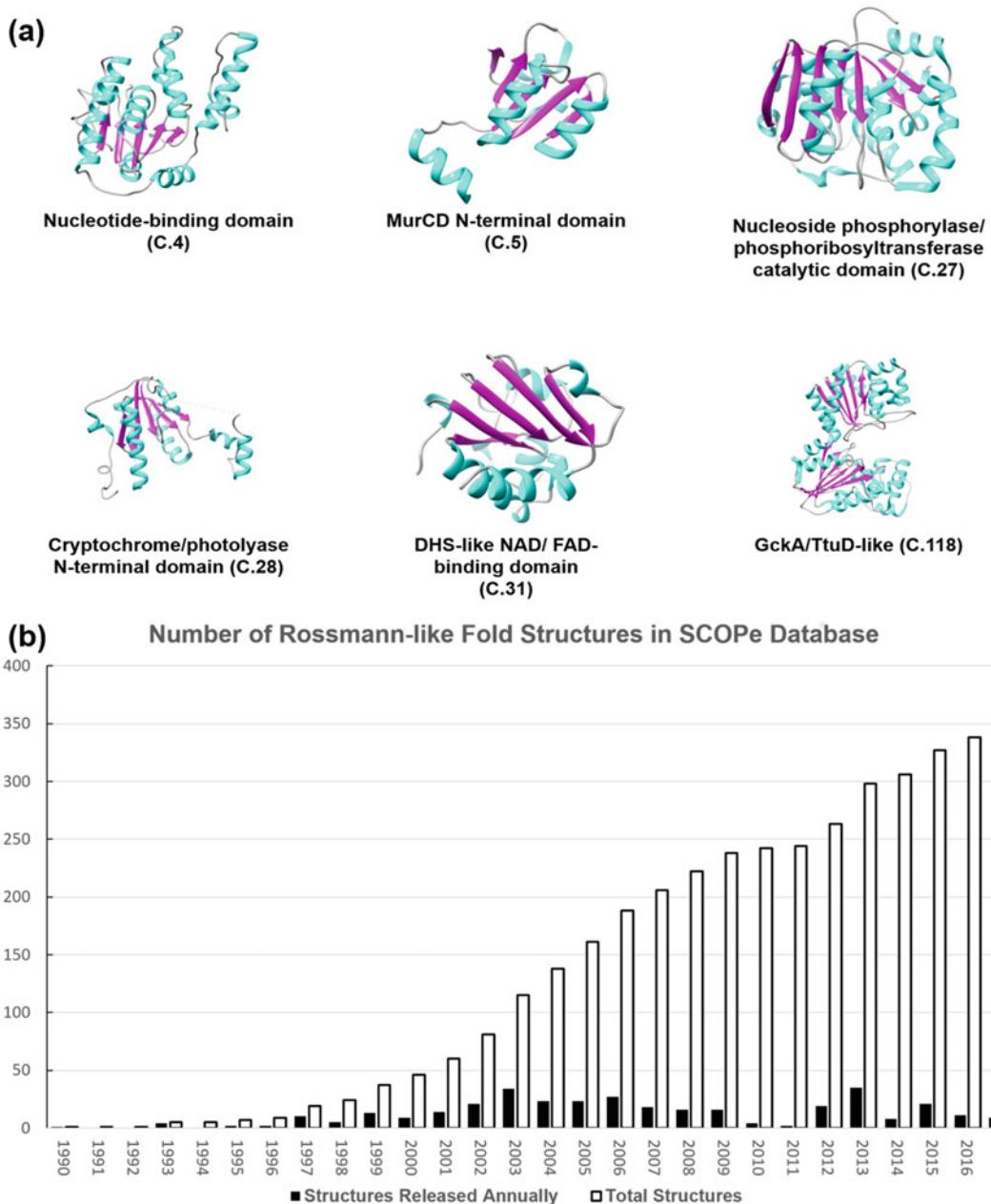


Fig. 5 Rossmann-like superfamilies in SCOPe. **(a)** The tertiary structures of six Rossmann-like superfamilies in SCOPe. **(b)** The growth of Rossmann-like fold structures in SCOPe

domain structure not restricting it to structures with a twofold symmetry, as shown in the center of Fig. 6a.

In the topology level, the Rossmann fold has the largest number of domains (52,880 domains out of 434,857 domains, 12.2%). The Rossmann fold topology is composed of 232 homologous

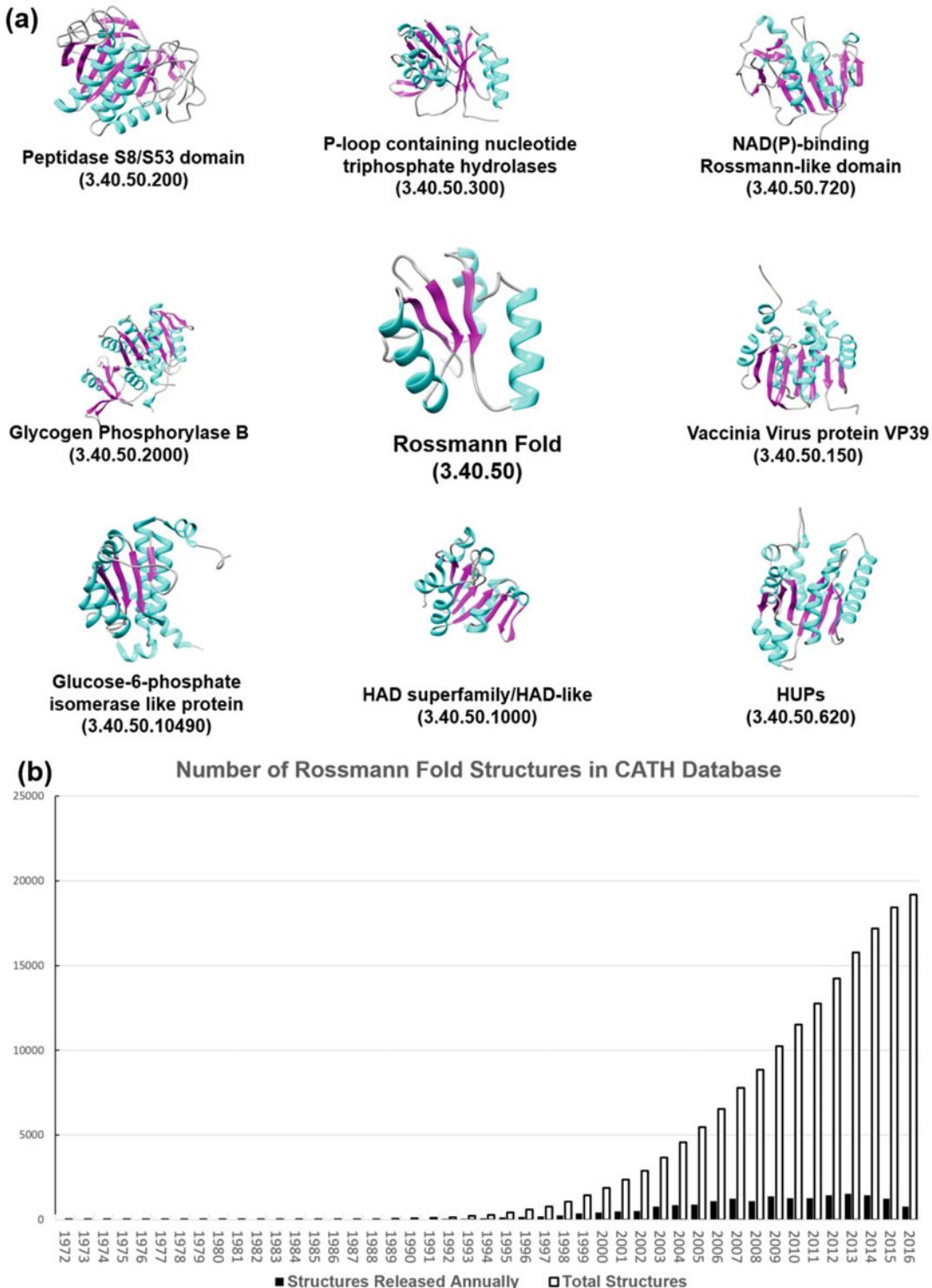


Fig. 6 The Rossmann fold in CATH. (a) Structures that belong to the Rossmann fold topology. (b) The growth of Rossmann fold structures in the CATH database

superfamilies (H level). The eight largest superfamilies of the Rossmann fold are illustrated in Fig. 6a. The main difference across the homologous superfamilies is the number of β -strands forming the β -sheet and their directions. For example, Vaccinia virus protein (3.40.50.150) VP39 has seven β -strands with one strand antiparallel, whereas glucose-6-phosphate isomerase-like protein (3.40.50.10490) has five parallel β -strands. As of the latest update of the CATH database (2018-09-06), the number of PDB entries that belong to the Rossmann fold (3.40.50) has reached 19,184. Figure 6b shows the growth curve.

4 Rossmann Folds in Genomes

The number of Rossmann fold proteins in genomes has been estimated a couple of times by using bioinformatics predictions. Wolf et al. estimated the protein fold distribution on a genomic scale in 1999 [15]. They assigned protein structures to 13 complete genome sequences (*H. influenzae*, *M. genitalium*, *M. pneumoniae*, *Synechocystis* sp., *H. pylori*, *E. coli*, *B. subtilis*, *B. burgdorferi*, *A. aeolicus*, *M. jannaschii*, *M. thermoautotrophicum*, *A. fulgidus*, and *S. cerevisiae* and *C. elegans*), which were available at that time (April 1998) using a sequence database search method, PSI-BLAST [16] ran against SCOP ver. 1.35. A fold was assigned to a protein if the E-value was less than or equal to 10^{-2} . They found that P-loop NTPases are the most abundant fold in all the genomes, and distribution of top 30 folds follows an exponential distribution. The Rossmann fold had a proportion of 3.6% (6th), 3.5% (8th), and 3.2% (9th) in bacterial, archaeal, and eukaryotic genomes, respectively. Although their aim was to assign protein folds to the each protein in the genomes, the percentages of the fold assignment in a genome ranged from 19.2 to 39.0% leaving 60% or more of the proteins unassigned.

Three years later, Gerstein and his colleagues also performed a fold assignment for 20 complete genomes [17]. Six more genomes, *C. pneumoniae*, *C. trachomatis*, *M. tuberculosis*, *P. horikoshii*, *R. prowazekii*, and *T. pallidum*, were added to the 14 genomes which Wolf et al. have analyzed. PSI-BLAST was used for the assignment, which was ran against SCOP ver. 1.39 with an E-value cutoff of 10^{-4} . The assignment coverages ranged from 17.6 to 34.6%. In their assignment, the Rossmann fold did not appear in three genomes, *C. elegans*, *B. burgdorferi*, and *T. pallidum*. On the other hand, in *M. tuberculosis* and *B. subtilis*, the Rossmann fold was the most abundant fold, 16% in *M. tuberculosis* and 14% for *B. subtilis*. In the remaining 15 genomes, the proportion of the Rossmann fold was 4–12%.

In 2004, Kihara and Skolnick predicted protein folds of five complete genomes, *M. genitalium*, *E. coli*., *B. subtilis*, *A. aeolicus*,

Table 1**The number of proteins with a structure model that have the Rossmann fold in five genomes**

Genomes	The number of proteins	The number of proteins with a model	The proteins of the Rossmann fold
<i>Homo Sapiens</i>	170,418	140,104	21,809 (15.6%)
<i>Mus musculus</i>	57,825	52,444	7117 (13.6%)
<i>Canis familiaris</i>	37,262	34,447	4915 (14.3%)
<i>Caenorhabditis elegans</i>	27,954	24,011	3587 (14.9%)
<i>Drosophila melanogaster</i>	21,986	19,550	3023 (15.5%)

We counted the number of proteins with a structure model built from a template structure of the Rossmann fold. The percentage is computed relative to the number of proteins with a model.

and *S. cerevisiae*, using a template-based structure prediction method (threading), PROSPECTOR_Q [18]. PROSPECTOR_Q searches structures that fit to a query protein sequence in a similar way as sequence database search methods, e.g., PSI-BLAST, but uses structure information, which is predicted secondary structure information and predicted residue-residue contacts, in addition to sequence similarity information from FASTA [19]. Structures in CATH at the topology levels were used as a reference dataset to be searched. The coverage of the fold assignment was higher than the two previous methods, 73–85%. In their results, the Rossmann fold was the most abundant topology in all five genomes, around 18% except for *S. cerevisiae* (12.8%).

In Table 1, we have newly analyzed the number of the Rossmann folds in genomes in ModBase [20], which is a database of annotated homology models of proteins in genomes. In ModBase, structure models were generated by a modeling pipeline called ModPipe, which uses PSI-BLAST to find template structures for modeling and uses the Modeller [21] program to build models. We searched the number of the Rossmann folds in five genomes, *Homo Sapiens*, *M. musculus*, *C. familiaris*, *C. elegans*, and *D. melanogaster*, which have over 10,000 proteins that have structure models. The number of Rossmann fold proteins is summarized in Table 1. The percentage of the Rossmann folds among proteins with models in the five genomes was around 14%, slightly lower than the estimate by Kihara and Skolnick [18].

5 The Rossmann Fold and Protein Engineering

Since the Rossmann fold binds nucleoside cofactors and catalyzes various reactions, it is considered an interesting target for protein

engineering. The focus of this engineering has been to change the specificity of cofactors that bind to enzymes [22].

The first protein engineering performed on the Rossmann fold was by Scrutton et al. [11] in 1990, who mutated the last glycine of the GXGXXG motif of glutathione reductase to alanine so that the engineered protein binds to NADP, but not NAD, the protein's original cofactor of the protein. In 2015 Gerth and her colleagues attempted to modify the behavior of primary-secondary alcohol dehydrogenase to use NADPH as a cofactor in contrast to the original NADH cofactor. Based on observations of the conservation and the difference between structures of NADPH-dependent enzymes and NADH-dependent enzymes, they introduced a quadruple mutant, G198D/S199 V/P201E/Y218A, which significantly reduced the enzymatic activity upon NADPH binding while showing the catalytic activity upon binding of NADH. The observed change of the enzymatic activity was explained by the interactions between side chains and NADPH [23]. Brinkmann-Chen et al. conducted a thorough study on cofactor specificity of ketol-acid reductoisomerase [24]. By comparing a few hundred sequences and seven structures of the enzyme, particularly around the GXGXXG motif, they found "DDV" switch residues that change the cofactor specificity. The native enzyme is activated upon binding NADPH. They found that two mutations to D from S make up for missing phosphates in NAD and substitution to V from I makes a deeper pocket for an adenine fragment, which resulted in a higher catalytic efficiency with NADH than the wild-type enzyme with NADPH.

6 Summary

The Rossmann fold is probably one of the best known folds because of its abundance and the functional divergence of proteins that adopt this fold. Here we briefly reviewed the history, the database entries, and the population of this fold. We also introduced works that used the Rossmann fold as a target of engineering, which lead us to a better understanding of the ways to control compound binding specificity of proteins.

Acknowledgments

The authors are thankful for Lyman Monroe for proofreading the manuscript. This work was partly supported by the National Institute of General Medical Sciences of the NIH (R01GM123055) and the National Science Foundation (DMS1614777, CMMI1825941).

References

1. Dawson NL, Lewis TE, Das S, Lees JG, Lee D, Ashford P, Orengo CA, Sillitoe I (2017) CATH: an expanded resource to predict protein function through structure and sequence. *Nucleic Acids Res* 45(Database issue): D282–D295. <https://doi.org/10.1093/nar/gkw1098>
2. 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(7):536–540. [https://doi.org/10.1016/S0022-2836\(05\)80134-2](https://doi.org/10.1016/S0022-2836(05)80134-2)
3. Perutz MF, Rossmann MG, Cullis AF, Muirhead H, Will G, North AC (1960) Structure of haemoglobin: a three-dimensional Fourier synthesis at 5.5-Å resolution, obtained by X-ray analysis. *Nature* 185(4711):416–422. <https://doi.org/10.1038/185416a0>
4. Adams MJ, Ford GC, Koekeok R, Lentz PJ, McPherson A, Rossmann MG, Smiley IE, Schevitz RW, Wonacott AJ (1970) Structure of lactate dehydrogenase at 2.8 Å resolution. *Nature* 227:1098–1103. <https://doi.org/10.1038/2271098a0>
5. Rossmann MG, Adams MJ, Buehner M, Ford GC, Hackert ML, Lentz PJ, McPherson A, Schevitz RW, Smiley IE (1972) Structural constraints on possible mechanisms of lactate dehydrogenase as shown by high resolution studies of the apoenzyme and a variety of enzyme complexes. *Cold Spring Harb Symp Quant Biol* 36:176–191
6. Buehner M, Ford GC, Moras D, Olsen KW, Rossmann MG (1973) D-glyceraldehyde-3-phosphate dehydrogenase: three-dimensional structure and evolutionary significance. *Proc Natl Acad Sci U S A* 70(11):3052–3054. <https://doi.org/10.1073/pnas.70.11.3052>
7. Rao ST, Rossmann MG (1973) Comparison of super-secondary structures in proteins. *J Mol Biol* 76(2):241–256. [https://doi.org/10.1016/0022-2836\(73\)90388-4](https://doi.org/10.1016/0022-2836(73)90388-4)
8. Schulz GE, Schirmer RH, Pai EF (1982) FAD-binding site of glutathione reductase. *J Mol Biol* 160(2):287–308. [https://doi.org/10.1016/0022-2836\(82\)90177-2](https://doi.org/10.1016/0022-2836(82)90177-2)
9. Hanukoglu I (2015) Proteopedia: Rossmann fold: a beta-alpha-beta fold at dinucleotide binding sites. *Biochem Mol Biol Educ* 43(3):206–209. <https://doi.org/10.1002/bmb.20849>
10. Hanukoglu I, Gutfinger T (1989) cDNA sequence of adrenodoxin reductase. Identification of NADP-binding sites in oxidoreductases. *Eur J Biochem* 180(2):479–484. <https://doi.org/10.1111/j.1432-1033.1989.tb14671.x>
11. Scrutton NS, Berry A, Perham RN (1990) Redesign of the coenzyme specificity of a dehydrogenase by protein engineering. *Nature* 343(6253):38–43. <https://doi.org/10.1038/343038a0>
12. Fox NK, Brenner SE, Chandonia JM (2014) SCOPe: structural classification of proteins-extended, integrating SCOP and ASTRAL data and classification of new structures. *Nucleic Acids Res* 42(Database issue): D304–D309. <https://doi.org/10.1093/nar/gkt1240>
13. Michie AD, Orengo CA, Thornton JM (1996) Analysis of domain structural class using an automated class assignment protocol. *J Mol Biol* 262(2):168–185. <https://doi.org/10.1006/j.bi.1996.0506>
14. Orengo CA, Brown NP, Taylor WR (1992) Fast structure assignment for protein databank searching. *Proteins* 14(2):139–167. <https://doi.org/10.1002/prot.340140203>
15. Wolf YI, Brenner SE, Bash PA, Koonin EV (1999) Distribution of protein folds in the three superkingdoms of life. *Genome Res* 9(1):17–26. <https://doi.org/10.1101/gr.9.1.17>
16. Altschul SF, Madden TL, Schaffer AA, Zhang J, Zhang Z, Miller W, Lipman DJ (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res* 25(17):3389–3402. <https://doi.org/10.1093/nar/25.17.3389>
17. Hegyi H, Lin J, Greenbaum D, Gerstein M (2002) Structural genomics analysis: characteristics of atypical, common, and horizontally transferred folds. *Proteins* 47(2):126–141. <https://doi.org/10.1002/prot.10078>
18. Kihara D, Skolnick J (2004) Microbial genomes have over 72% structure assignment by the threading algorithm PROSPECTOR_Q. *Proteins* 55(2):464–473. <https://doi.org/10.1002/prot.20044>
19. Pearson WR, Lipman DJ (1988) Improved tools for biological sequence comparison. *Proc Natl Acad Sci U S A* 85(85):2444–2448. <https://doi.org/10.1073/pnas.85.8.2444>
20. Pieper U, Eswar N, Davis FP, Braberg H, Madhusudhan MS, Rossi A, Marti-Renom M, Karchin R, Webb BM, Eramian D, Shen MY, Kelly L, Melo F, Sali A (2006) MODBASE: a database of annotated comparative proteins

- structure models and associated resources. Nucleic Acids Res 34(Database issue): D291–D295. <https://doi.org/10.1093/nar/gkj059>
21. Webb B, Sali A (2016) Comparative protein structure modeling using modeller. Curr Protoc Bioinformatics 54:5.6.1–5.6.37. <https://doi.org/10.1002/cpb.3>
22. Li Y, Cirino PC (2014) Recent advances in engineering proteins for biocatalysis. Biotechnol Bioeng 111(7):1273–1287. <https://doi.org/10.1002/bit.25240>
23. Maddock DJ, Patrick WM, Gerth ML (2015) Substitutions at the cofactor phosphate-binding site of a clostridial alcohol dehydrogenase lead to unexpected changes in substrate specificity. Protein Eng Des Sel 28 (8):251–258. <https://doi.org/10.1093/protein/gzv028>
24. Brinkmann-Chen S, Flock T, Cahn JKB, Snow CD, Brustad EM, McIntosh JA, Meinholt P, Zhang L, Arnold FH (2013) General approach to reversing ketol-acid reductoisomerase cofactor dependence from NADPH to NAD. Proc Natl Acad Sci U S A 110(27):10946–10951. <https://doi.org/10.1073/pnas.1306073110>



Chapter 2

Advances in Protein Super-Secondary Structure Prediction and Application to Protein Structure Prediction

Elijah MacCarthy, Derrick Perry, and Dukka B. KC

Abstract

Due to the advancement in various sequencing technologies, the gap between the number of protein sequences and the number of experimental protein structures is ever increasing. Community-wide initiatives like CASP have resulted in considerable efforts in the development of computational methods to accurately model protein structures from sequences. Sequence-based prediction of super-secondary structure has direct application in protein structure prediction, and there have been significant efforts in the prediction of super-secondary structure in the last decade. In this chapter, we first introduce the protein structure prediction problem and highlight some of the important progress in the field of protein structure prediction. Next, we discuss recent methods for the prediction of super-secondary structures. Finally, we discuss applications of super-secondary structure prediction in structure prediction/analysis of proteins. We also discuss prediction of protein structures that are composed of simple super-secondary structure repeats and protein structures that are composed of complex super-secondary structure repeats. Finally, we also discuss the recent trends in the field.

Key words Protein structure prediction, Free modeling, Template-based modeling, Secondary structure prediction, Simple secondary structure, Complex super-secondary structure, Proteins with simple super-secondary structure repeats, Proteins with complex super-secondary structure repeats

1 Introduction

Protein sequences are unique, and this sequence allows the protein to fold into a specific three-dimensional structure [1]. The elucidation of structure of protein is essential in understanding its biological functions. In that regard, various structural genomics initiative and efforts from various researchers have led to the deposition of large number of protein structures. In August 2015, there were about 100 thousand [2, 3] structures deposited in PDB. As of April 2018, the number of structures in the database has increased to about 130 thousand. In August 2015, there were about 50 million [2, 3] sequences in UNIPROT, and as of April 2018, the number is approximately 114 million which suggests that the gap between the sequence space and structure space is ever increasing

and that computational approaches are required to fill this gap. We highlight some important methods for protein structure prediction in Subheading 2. For a thorough review on recent advances on protein structure prediction, refer to KC [4]. We also discuss the fragment assembly approach and relationship between fragments and super-secondary structure.

Computational methods for prediction of protein structures use information of secondary and super-secondary structures [5–11]. Thus, the prediction/assignment of secondary and super-secondary structures of a given protein sequence is of paramount importance for the prediction of protein structure. Please refer to [12] for review on methods for prediction of secondary structures.

Super-secondary structures are mostly a combination of two secondary structures and a turn and are considered as a bridge between secondary and tertiary structures. They are also referred to as structural motifs, and some main motifs include α -turn- α , β -hairpins, and coiled coils. Aside from these major motifs, there are others that are classified into three categories, helix, sheet, and mix. We will elaborate on these in more detail in Subheading 3. In the mid-1900s, Anfinsen demonstrated through experiments that a protein's tertiary structure is encoded in its amino acid sequence, resulting in the design of methods that predict tertiary structure from protein sequence [5, 13–15]. In the primary structure, the distribution of amino acids, whether hydrophilic or hydrophobic is pivotal in determining the tertiary structure of the protein [1]. The primary structure comprises a sequence of amino acids in a polypeptide chain, and these are joined together by peptide bonds [5]. Peptide is a molecule of two or more amino acid residues and polypeptides are larger peptides. The peptide bond that holds the amino acid residues together is a chemical bond formed by the interaction between the different residues in the peptide molecule. Residues from the carboxyl group react with residues from the amino group of another residue, and this releases the chemical bond. Oxygen atoms exist within the amino group, and hydrogen atoms also exist within the carboxyl group. Thus, the presence of hydrogen bond patterns between these atoms introduces the next level in the protein hierarchy, the protein secondary structure [1]. In this regard, prediction of secondary structure as well as super-secondary structure is an important step in prediction of protein structure. In Subheading 3, we briefly describe assignment of protein secondary structure followed by description of protein super-secondary structure and types of super-secondary structure: simple super-secondary structure and complex super-secondary structure. In Subheading 4, we describe methods for prediction of simple super-secondary structure (including coiled-coil prediction in details), methods for prediction of complex super-secondary structures, methods for prediction/analysis of proteins that are composed of repeats of simple super-secondary structure, and methods

for prediction/analysis of protein structures that are composed of repeats with complex super-secondary structures. We also briefly discuss on the symmetry aspects of these repeats. Finally, we will conclude the chapter by discussing applications of super-secondary structure prediction in protein structure prediction focusing on methods that use super-secondary structure information, methods that use super-secondary structures, and recent trends in prediction of protein structures using super-secondary structures. We also provide a set of notes that summarizes the chapter.

2 Recent Advances in Protein Structure Prediction Methods

Methods for protein structure prediction have been classified into three categories [16]: comparative modeling (CM or homology modeling) [17], threading [18], and free-modeling (FM or ab initio) [19] approaches. Qualitatively, threading and CM can be combined in one group: template-based modeling (TBM). In that regard, protein structure prediction approaches can be categorized into two types: template-based modeling and template-free modeling (FM). TBM approaches make use of structural templates, and FM approaches attempt to predict the structure from first principle and without structural templates [20].

The most successful methods for such *free modeling* prediction from sequence are based on fragment assembly [21, 22]. These methods that use fragment assembly are in a way very relevant to our discussion as these fragments have length similar to super-secondary structures. One of the important observations in current protein structure prediction arena is that the success rate of these methods is highly correlated to the evolutionary distance between target (query) sequence and template structure [23, 24]. If the query sequence has a sequence identity >50% to the template(s), TBM methods are able to produce models with root-mean-squared distance (RMSD) as low as 1.0 Å. Similarly, if the target sequence has sequence identity between 30 and 50% to the templates, the methods are able to produce models of around 85% of the core regions as low as 3.5 Å [23]. Furthermore, if the query sequence has a sequence identity below 30% (“the twilight zone”) [25], the accuracy of the model decreases quite drastically. Next, we discuss most recent advances in TBM approaches and template-free modeling approaches (*see Note 1*).

2.1 Recent Advances in Template-Based Modeling (TBM) Approaches

We can possibly track the origin of TBM to the 1969 work of Browne and colleagues where they constructed structural models of the bovine alpha-lactalbumin using the solved hen egg-white lysozyme structure as a template [26]. Among many developments, three major developments have been attributed to the improvement of TBM approaches [27]: (1) the development of PSI-BLAST

[28] and the consequent profile-to-profile alignment techniques [9, 18, 29] that significantly increased the accuracy of template identification, (2) the development of composite structure assembly simulation methods [30, 31], and (3) the rapid accumulation of experimental sequence and structure databases. Here, we discuss some of the most successful TBM approaches: MODELLER [32], ModBase [33], and I-TASSER.

2.1.1 MODELLER

MODELLER [32] developed at Andrej Sali’s lab is one of the most widely used TBM approaches for protein structure prediction. MODELLER’s protocol generally consists of four steps: (1) searching for structures related to target sequence, (2) aligning target sequence and the template(s), (3) model building, and (4) evaluation of the model. In addition, MODELLER implements TBM by satisfaction of spatial restraints collected from various sources [32].

2.1.2 ModBase

The ModBase [33] is a database and other associated resources for comparative protein structure models and is also developed at Andrej Sali’s lab at UCSF. The models of ModBase are calculated using ModPipe [34], a pipeline for comparative protein modeling that relies on number of modules of MODELLER [32] and other various sequence-sequence [35], sequence-profile [28], and profile-profile [29, 36] methods for sequence-structure alignment. It is important to note that other external databases such as the Protein Model Portal (PMP) [32] also provide access to ModBase models.

2.1.3 I-TASSER

Another important TBM approach is I-TASSER [37] developed by Yang Zhang’s group at the University of Michigan. It is a composite TBM approach and has been consistently ranked as one of the most successful TBM protein structure prediction approaches in CASP assessments. For a target sequence, a meta-threading approach called LOMETS [30, 31] is first utilized to identify structural templates. The continuous fragments are then excised from the templates in the regions that are aligned by threading. Replica-exchange Monte Carlo simulations are then used to reassemble full-length models from these fragments. Subsequently, the structure trajectories are clustered to identify the low free-energy states. The models from the low-energy conformations are further refined by atomic-level simulations to obtain the final model. For interested readers, a retrospective report of the I-TASSER pipeline can be found in Yang et al. [27]. For CASP 11, the Zhang’s group proposed a new I-TASSER [27] pipeline that combined template identification by meta-threading programs, followed by the QUARK ab initio approach to generate initial full-length models under strong constraints from template alignments.

2.2 Recent Advances in Free Modeling (FM) Approaches

Template-free modeling (FM) approaches seek to construct structural models for protein sequences that do not have a detectable template. These methods are also termed as FM ab initio or de novo methods. Among the FM approaches, fragment assembly approach has become one of the most important approaches. Here, we describe some important fragment assembly-based FM methods. It also has to be noted that the fragments can be related to supersecondary structures.

2.2.1 Fragment Assembly Approach

The fragment-assembly approach refers to modeling methods that assemble models of proteins using fragments from known protein structures. Bowie and Eisenber's work on assembling a protein structure using small 9-mer fragments from other PDB proteins [38] is often recognized as the foundational work for fragment assembly-based approach. Later, David Baker's group adopted this idea in the development of ROSETTA [21]. Subsequent works from David Baker's group Bradley et al. [39] and other works (I-TASSER [7], QUARK [22]) from Yang Zhang's group championed the method of fragment assembly.

ROSETTA

ROSETTA, originally developed by David Baker's group at the University of Washington but now co-developed by more than 44 laboratories, is undoubtedly one of the most actively developed tools for protein structure prediction.

As in any other structure prediction tools, the two main steps in ROSETTA are (1) conformational sampling and (2) ranking of the models.

For conformational sampling, ROSETTA uses sampling for both backbone and side chain. Additionally, the backbone conformational sampling is divided into large backbone conformational sampling and local backbone refinement. Conformations of nine- or three-amino acid peptide fragments are used for modeling the large backbone conformational sampling, and Metropolis Monte Carlo sampling of ϕ and ψ is performed for modeling local backbone refinement. The side chain is modeled as centroid. Refer to Rohl et al. [40] for details of ROSETTA. For ranking the models, ROSETTA uses a knowledge-based energy function that includes solvation, electrostatics, hydrogen bonding, and steric clashes [40]. Most recently, ROSETTA approach was able to successfully predict the structure of a 250 residue long protein that did not have any templates.

QUARK

QUARK [22] developed at Yang Zhang's lab at the University of Michigan is another FM approach that has been consistently ranked as one of the top FM approaches. QUARK breaks the query sequences into fragments of 1–20 residues (many of them belong to the super-secondary structures category) where multiple fragment residues are retrieved at each position from unrelated

experimental structures. Then, these fragments are assembled into a full length using replica-exchange Monte Carlo simulations guided by a composite knowledge-based force field [22].

Based on a benchmark study [22], QUARK was able to correctly fold 31% of the mid-sized proteins (100–150 residues) with a TM-score >0.5. It was also noted that the cases where QUARK was not able to perform well were the proteins where the structural topology was complex, such as β -proteins of complicated strand arrangement.

In CASP11, Yang Zhang's group also developed a TBM version of QUARK called QUARK-TBM, which is an extension of QUARK to template-based modeling. This intermediate step was added to the pipeline that improved the quality of the output models for TBM targets. In this pipeline, in addition to the spatial restraints from the threading templates, the restraints are also taken from the full-length models generated from QUARK-TBM. However, it should be noted that, due to its high computational cost, QUARK-TBM is only used for domains that are less than 300 residues in length.

2.2.2 Fragment Generation and Fragment Library

As discussed above, fragment-based approaches are one of the most successful approaches for free modeling, and they rely on accurate and reliable fragment libraries. Hence, accurate fragment library generation is important to the overall success of fragment-based approaches [41]. The quality of the fragments, hence, is tied to the success of the fragment-based approaches. There are various approaches for generating fragments NNMake (ROSETTA's method for fragment library generation) [42], FRAGFold [43], HHFrag [44], SAFFrag [45], and others.

Recently, some studies have been performed with the aim of designing a better fragment library for improving FM of protein structure. In this regard, De Oliveira et al. developed Flib [42], a novel method to build better fragment library. Scoring fragments based on the predicted secondary structure of the fragment (essentially alpha helical fragments being predicted more accurately), Flib on a validation set of 41 proteins performed better than two state-of-the-art methods NNMake and HHFrag.

In addition, not only the length of the fragment but also the number of fragments used per position is equally important for the success of fragment assembly-based approaches. In this regard, Xu and Zhang [46] performed systematic analysis of length of the fragments, number of fragments per position, and how these factors affect the precision of the library and showed that this new fragment library developed based on the findings of this analysis performs better than the existing fragment libraries for the ab initio structure prediction.

Based on this study, it was also concluded that the optimal fragment length for structural assembly is around 10 and at least 100 fragments per position are required for reliable structure

Table 1

Size of fragments and relationship to super-secondary structures in some popular protein structure prediction algorithm

	Rosetta	I-TASSER	QUARK	Smotif	Fragfold
Size of fragments	3- and 9-residue fragments	Segments of 2–4 consecutive or nonconsecutive secondary structural elements	1–20 residue fragment	Super-secondary motifs by SSE orientation	Super-secondary structural fragments
Description	SS and SSS coverage	SEGMER detects SSS in algorithm	Many of the fragment belong to SSS	SSS motifs	SSS

prediction. This is also in alignment with the fact that 9-mer fragment is the most popular choice for the size of fragments [41, 47]. In the case of ROSETTA, its fragment libraries contain 200 fragments per position, and these fragments are typically 3 and 9 residues long [42].

2.2.3 Fragment and Super-Secondary Structure

In 2017, Mackenzie and Grigoryan [48] reviewed the motif libraries and summarized recent achievements in this arena, focusing on subdomain level structural patterns and their applications to protein design and structure prediction. Especially, relevant to protein super-secondary structures are the motif libraries of Rosetta (3- and 9- residue fragments), QUARK (1–20 residue fragments), Smotifs (super-secondary structure itself), TERMs (motifs) and I-TASSER(SEGMER generated super-secondary structure motifs).

Similar to the work of Mackenzie and Grigoryan [48], we briefly summarize the relationship between fragments and super-secondary structure in the context of various protein structure prediction methods. The summary is depicted in Table 1.

3 Protein Super-Secondary Structures and Types of Super-Secondary Structure

In this section, we briefly discuss protein secondary structures, recent methods for assignments of secondary structure, super-secondary structure, and types of super-secondary structures.

3.1 Assignment of Protein Secondary Structures

Super-secondary structures serve as a bridge between secondary and tertiary structures. Combination of two or more secondary structures and a turn basically defines a super-secondary structure. Some super-secondary structures can be associated with particular functions like DNA binding, whereas others have no biological

functions on their own, except that they are part of larger structural and functional assemblies. Super-secondary structures include α -helix hairpins, β -hairpins, coiled coils, Greek key motif, Rossmann fold, α -turn- α , α -loop- α , β - α - β , and others. Since supersecondary structures are made up of secondary structures, here we briefly review the secondary structure and secondary structure prediction methods.

In predicting protein secondary structures, the main aim is to tell whether amino acid residues in the protein fall under α -helices, β -strands, or neither. The α -helices are denoted by **H**, the β -strands by **E**, and those that are neither α -helices nor β -strands by **C** [5, 15]. In some other literature, the third group, **C**, is referred to as the coil region [49]. These structures are assigned from the three-dimensional atomic coordinates of a protein structure [5, 15] and are used in a variety of applications, including classification of protein folds, visualization [5, 50–55], and structural alignment. Due to their usefulness, numerous methods have been designed for their extraction.

Pauling and Corey [56] proposed the existence of helical and sheet conformations in 1951 even before the first structure determined by X-ray crystallography in 1958. Since then, more than 66 years have passed, and a plethora of methods have been developed. The first of such approaches dates back to the 1970s by Levitt Greer of Columbia University [5, 57]. Others include STRIDE [58], DSSP [59], DEFINE [60], P-CURVE [61], P-SEA [62], SKSP [63], PROSIGN [64], SABA [65], KAKSI [66], XTLSSTR [67], SECSTR [68], PALSSE [69], SKSP [63], and Segno [70]. Yang et al. [12] observed that as of August 2016, there were 266 methods for the prediction of protein secondary structure. Among those 266 methods, DSSP [59] (which assigns secondary structure into eight states) seems to be most commonly used. One thing to note is that the accuracy of the methods depends upon the definition of secondary structure. The mostly used convention seems to be helix is designated as G (310 helix), H (α -helix), and I (β -helix), sheet as B (isolated bridge) and E (extended sheet), and all other states designated as a coil. The most recent methods use various deep learning techniques like deep neural network and convolution neural field network [71]. It has been observed that the prediction accuracy of the secondary structure prediction methods is slow but increasing.

One of the things that is lacking in the field is that individual groups report their prediction accuracy, and there are only few works that compare the accuracy of the methods on the same independent test set. One of them is the work by Yang et al. [12] where they created a robust independent benchmark dataset consisting of 115 proteins with sequences ranging from 43 to 1085 residues and then compared Jpred4, SCORPION, Porter4.0, PSIPRED3.3, SPINE X, SPIDER2, and DeepCNF

[71]. Comparison based on the 115 proteins as well as CASP 12 results showed that DeepCNF [71] performs slightly better than other compared approaches.

In conclusion, the secondary structure prediction has reached an accuracy of about 84%, and it has been reported that one of the obstacles for improvement in prediction accuracy is due to the difficulty in capturing nonlocal interactions between the residues while using window-based approaches [12].

3.2 Protein Super-Secondary Structures

Super-secondary structures are bridge between secondary structure and tertiary structure and are combination of numerous secondary structure elements. They normally are comprised of secondary structures and a turn. Thus, the assignment of protein super-secondary structures depends on the correct assignment of secondary structures. Super-secondary protein structures can be classified into three broad categories based on either they are made up of mainly helix or sheet or mix.

The helix based super-secondary structures include α -turn- α , helix-loop-helix (HlH), helix-hairpin-helix (HhH), helix corner (α - α corner), and EF hand. The sheet-based super-secondary structures are β -hairpin, β - β corner (β -corner), and Greek key motif. The mix super-secondary structures comprise both α and β . Some mix super-secondary structures include β - α - β and Rossmann fold (these have a complex structure and consist of α -helices and β -sheets that are connected by $\beta\alpha\beta$ motifs.)

3.3 Types of Super-Secondary Structures

In proteins, if two secondary structure units are connected by a polypeptide (loop) with a specific arrangement of geometry, the resulting structure is referred to as a super-secondary structure or motif. We can define super-secondary structures as combinations of α -helices and β -structures connected through loops that form patterns that are present in many different protein structures. These folding patterns are stabilized through the same kind of linkages than the tertiary level. Sometimes the term “motif” is used to describe these super-secondary structures. Based on the complexity of super-secondary structures, they can be classified as simple super-secondary structure and complex super-secondary structure (see Note 2).

3.3.1 Simple Super-Secondary Structure

These structures can be relatively simple, as α - α (two α -helices linked by a loop) and β - β (two beta-strands linked by a loop). There are four kinds of simple super-secondary structures, namely, α -loop- α , β -loop- α , β -turn- β , and β -loop- β . These motifs play an important role in protein folding and stability because a large number of motifs exist in protein spatial structure.

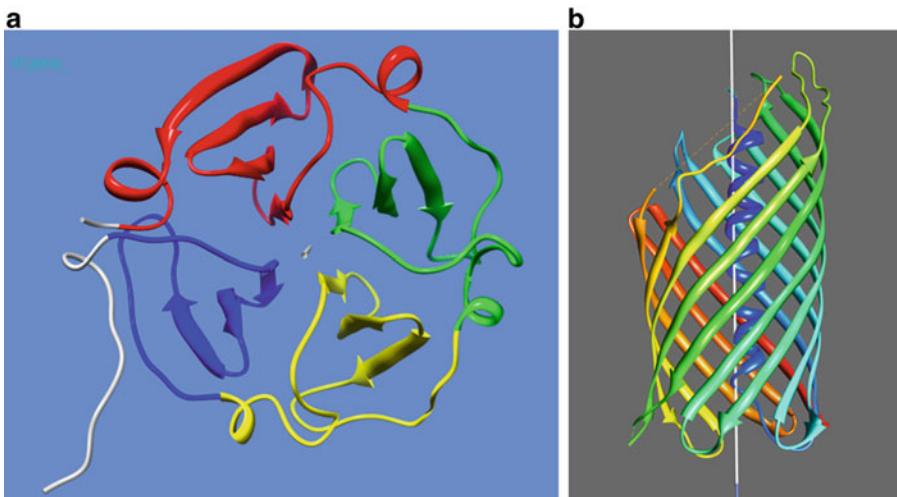


Fig. 1 (a) A four-bladed β -propeller and (b) β -barrel. The PDB code is shown in the figure. The axis of symmetry is also shown in each figure. SymD program was used to detect the axis of symmetry. For β -propeller, repeating units (complex super-secondary structures) are colored with different colors

3.3.2 Complex Super-Secondary Structure

Two or more simple super-secondary structures further fold into a complex super-secondary structure [72]. Some of the complex super-secondary structures include Rossmann fold (consist of $\beta\alpha\beta\alpha\beta$), Greek key ($\beta\beta\beta$), strand-loop-helix-loop-strand ($\beta\alpha\beta$), and β -meander ($\beta\beta\beta$). Two or more consecutive anti-parallel β -strands linked together by hairpin loops form the β -meander motif. It is common in β -sheets, found in several structural architectures including β -barrels and β -propellers. Figure 1 shows a four-bladed β -propeller and a β -barrel. The symmetry axis is also shown in the figure. The symmetry was determined by symmetry detection algorithm (SymD) [73].

4 Super-Secondary Structure Prediction Methods

In this section, we describe various super-secondary structure prediction methods. As mentioned in Subheading 3, combinations of two or more secondary structures and a turn are easily available in protein structures. They can be associated with particular functions like DNA binding. Aside from that, others have no biological functions on their own, except that they are part of larger structural and functional assemblies. The three main super-secondary structures are α -turn- α , β -hairpins, and coiled coils. We will look into details predictors for the other super-secondary structures as well. Super-secondary structure predictors are designed for specific types of protein super-secondary structures; for instance, the SpiroCoil, LogiCoil, and MultiCoil2 predictors predict only coiled coils [5]. Super-secondary structure predictors in most cases rely on predictions from secondary structure predictors when predicting

Table 2

Prediction methods discussed in the chapter. Various computational tools discussed in the chapter, WS stands for web server, NA not available, and SA stand alone

Type	Method	Year last published	Availability	Group	Reference
Template-based modeling	MODELLER	1989	WS/SA	Sali	[133]
	ModBase	1998	WS/SA	Sali	[134]
	I-TASSER	2007	WS/SA	Zhang	[135]
Template-free modeling	Rosetta	2013	WS/SA	Baker	[23]
	QUARK	2011	WS	Zhang	[136]
Coiled coil prediction	MultiCoil2	2011	WS/SA	Berger	[37]
	SCORER 2.0	2011	WS	Woolfson	[137]
	RFCoil	2015	WS	Li	[85]
	LogiCoil	2013	WS	Woolfson	[138]
	AAFreqCoil	2015	WS	Wang	[86]
	ACCORD	2015	NA	Song	[91]
	Waggawagga	2015	WS	Simm	[87]
	CCBuilder 2.0	2018	WS	Wood	[93]
Complex SSS (Rossmann fold)	Cofactory	2014	WS	Peterson	[104]
$\beta\alpha\beta$ (complex SSS)	Sun et al.	2016	WS	Hu	[103]
Symmetry detection	SymD	2014	WS	BK Lee	[73]
Beta-barrel	BBP	2012	NA	Tran	[112]
	BETAWARE	20,132,013	WS	Savojardo	[113]
Helix-turn-helix	Thornton et al.	2005	WS	Thornton	[55]
Multiple super-secondary structures	Zou et al.	2011	NA	Zou	[95]
Multiple super-secondary structures	Feng and Kou	2015	NA	Kou	[96]

super-secondary structures. An example is the BhairPred super-secondary predictor. This makes use of secondary predictions from the secondary predictor PSIPRED.

Elements of secondary structure and super-secondary structure can then combine to form the full three-dimensional fold of a protein. As the information gained can be used in the tertiary structure prediction, protein super-secondary structure prediction is a key step in the hierarchical approach to derive the protein tertiary structure [74]. Table 2 lists all the prediction methods presented in the chapter. Based on the types of methods, we classify these methods as methods for simple super-secondary structures, methods for multiple super-secondary structures, methods for complex super-secondary structures, and methods for repeated simple super-secondary and repeated complex super-secondary structures.

4.1 Prediction Methods for Simple Super-Secondary Structures

4.1.1 β -Hairpin Prediction

Here, we describe methods for prediction of various types of simple super-secondary structures. It can be observed that not much development has happened in the arena of prediction methods for simple super-secondary structures (*see Note 3*). Here, we describe a few recent methods in this category.

β -Hairpin super-secondary structures which are also known as strand-loop-strand motifs are one of the simplest super-secondary structures and occur frequently in globular proteins. The structure looks like a hairpin, thus the name β -hairpin. β -Hairpin comprises two antiparallel beta-strands connected by a hairpin bend, that is, β -turn (bovine trypsin inhibitor).

Some of the early computational methods to predict β -hairpins are the method from Janet Thornton's group by Cruz et al. [5, 75]. They developed artificial neural network-based approach to predict β -hairpins in 534 proteins. In 2004, Kuhn et al. [72] developed another neural network-based approach which achieved a classification accuracy of around 75%.

Since then, various predictors for predicting β -hairpins have been developed. Some of these predictors are BhairPred [5, 76], method by de la Cruz et al. [5, 75], method by Hu et al. [5, 75], method by Zou et al. [5, 77], and method by Xia et al. [5, 78], among others.

Xia et al. [79] proposed a β -hairpin prediction in proteins using support vector machine. In addition to using autocorrelation as a feature, they also used secondary structure information and residue conformation propensity to improve the prediction of β -hairpin. Please refer to Chen and Kurgan [80] for summary of β -hairpin methods developed until 2011.

More recently, in 2017 Li et al. [81] proposed a SVM-based algorithm for identifying the β -hairpin in *enzyme protein*. This approach attempted to predict β -hairpin motifs in an important class of proteins with catalytic functions. Based on the observation that β -hairpin is simple arrangement of the β -strand, and a cooperative interaction between the two strands of the β -hairpin loop often plays important role in ligand binding of enzyme, this method focused on the prediction of β -hairpin in enzyme proteins. The method used 15 amino acid residues as the length of the pattern and was trained on a non-redundant enzyme β -hairpin dataset containing 2818 β -hairpin and 1098 non- β -hairpins. Independent testing of the method produced a Matthews's Correlation Coefficient of 0.7.

Recently, YongE and GaoShan also utilized chemical shifts as a feature in the prediction of beta-hairpin motif [82]. Chemical shift usage in the prediction of protein structure is gaining steam recently (*see Note 6*). The proposed method [82] uses a quadratic discriminant-based approach to identify β -hairpins that used chemical shifts experimental data from NMR. The Matthew's correlation coefficient of this method was around 0.85.

4.1.2 Helix-Turn-Helix (α -Turn- α) Prediction

Helix-turn-helix (α -turn- α) motifs are an important structural motif of DNA-binding proteins. Janet Thorntons's group in 2005 [83] developed a web service to determine if a protein structure has a helix-turn-helix structural motif. This method determines whether a protein has a HTH motif or not by scanning the protein chain with a set of seven structural templates and determining the regions of the protein with the smallest RMSD. The features were combined using a linear predictor.

4.1.3 Prediction of Coiled Coils

Coiled coils are another type of simple super-secondary structures. Essentially, this super-secondary structure is formed when the α -helices twist around each other to form rope-like structures. These are two helices winding around each other in a super coil, and these are found in fibrinogen. They mainly contain a repeated pattern of hydrophobic (H) and charged amino acid residues (polar, P, residues), HPPHPP, and they are most frequently observed as the protein-protein interaction motifs in nature. Majority of coiled coils have at least four heptad repeats [84]. In addition, coiled-coil domains (CCDs) have been estimated to be about 3% of protein-encoding regions in all genes and, thus, are attractive and tractable targets for rational protein design [84].

In addition to coiled-coil predictors that detect coiled-coil structures in proteins, there are predictors that predict the oligomerization state of the coiled coil. Among many predictions for various super-secondary structures, coiled-coil prediction remains one of the most active fields of research. Some coiled-coil detection methods include COILS [28], PCOILS [18], Paircoil2 [29], SOSUICoil [9], MARCOIL [30], CCHMM_PROF [31], SpiriCoil [32], RFCoil [85], AAFreqCoil [86], and WaggaWagga [87]. Some methods that can predict coiled-coil oligomeric state predictions are SCORER 2.0 [33], LOGICOIL [34], PrOCoil [35], and RFCoil [85]. Multicoil2 [37] can detect both coiled-coil region and predict coiled-coil oligomeric state. Please refer to Li et al. [88] for a comprehensive review of these methods.

We briefly describe some of the recent coiled-coil detection methods especially, focusing on the ones that are developed after 2014. For a list of coiled-coil methods, refer to [80]. As the detection of coiled coil from sequence is relatively mature, most of the recent methods are focused on prediction of oligomeric state of coiled coil. Since there are a lot of recent methods developed in coiled-coil predictions, we describe these methods and also various aspects of coiled-coil prediction in detail (*see Note 4*).

Methods for Coiled-Coil Detection As Well As Oligomerization State Prediction

1. MultiCoil2

Amy Keating's and Bonnie Berger's group developed Multicoil2 [37] in 2011. The Multicoil2 algorithm predicts both the detection of coiled coils and oligomerization state of the

coiled coils. It has to be noted that this method only focuses on discriminating between dimers and trimers. Multicoil2 combines the correlations of the original Multicoil [89] with hidden Markov model in a Markov random field framework. Essentially, multinomial logical regression is used to combine sequence features. In addition to oligomeric state predictions, Multicoil2 can also detect coiled coil. The Multicoil2 program and other resources are available at <http://multicoil2.csail.mit.edu>.

Methods for Prediction of Oligomeric State of Coiled Coil

1. RFCOIL

In 2015, Li et al. developed RFCOIL [85] to predict oligomerization state predictor of coiled-coil super-secondary structures. This method uses various amino acid indices and combines them as features in a random forest framework to predict oligomeric states of coiled-coil regions. Upon comparison with four existing predictors LOGICOIL [34], PrOCOIL [35], SCORER2.0 [33], and MULTICOIL2 [37], RFCOIL outperformed these methods. RFCOIL is available at <http://protein.cau.edu.cn/RFCOIL/>.

2. AAFreqCoil

AAFreqCoil [86] is another predictor that distinguishes between parallel dimeric and trimeric coiled coils. This predictor was developed in 2015 by Wang et al. Since only two major oligomeric states have been established until date, this predictor classifies the two, parallel dimeric and trimeric coiled-coil structures. It has been demonstrated to have a competitive performance when benchmarked based on jackknife and ten-fold cross-validations. The AAfreqCoil algorithm is available at <http://genomics.fzu.edu.cn/AAFreqCoil/>.

3. SCORER 2.0

In 2011, Woolfson's group [33] developed a coiled-coil oligomeric state predictor to distinguish between parallel dimers and trimers. SCORER 2.0 is an upgraded version of SCORER algorithm that uses a log-likelihood ratio to calculate the relative likelihood of a test coiled-coil sequence to be a dimer or a trimer. The SCORER 2.0 is available at <http://coiledcoils.chm.bris.ac.uk/Scorer>.

4. LOGICOIL

In 2013, Vincent et al. [34] developed a multi-state prediction of coiled-coil oligomeric state prediction called LOGICOIL. This predictor mainly targets antiparallel dimer, parallel dimer, trimer, and tetramer. LOGICOIL uses Bayesian variable selection and multinomial probit regression for prediction by defining the problem as a multi-class classification problem. LOGICOIL is available at: <http://coiledcoils.chm.bris.ac.uk/LOGICOIL/>.

Predicting Relative Orientation of α -Helix in a CC Domain

Coiled-coil (CC) super-secondary structure is an important unit of proteins. CCs can form different oligomeric complexes. The major oligomeric state of CCs is a dimer, and the helix orientation in complexes can be either parallel or antiparallel. To understand the molecular function of CC-containing proteins, it is crucial to determine the relative orientation of each α -helix in a CC domain. Amy Keating's group 2008 [90] did a comprehensive study to compare the performance of existing computational methods for predicting helix orientation for coiled-coil dimers. This work concluded that several different types of computational approaches are capable of discriminating parallel from antiparallel coiled-coil helix alignments with reasonable accuracy.

Experimental Approaches to Determine the Relative Orientation of α -Helix in a CC Domain

Along with computational approaches, there have been some experimental approaches to identify the orientation of helix by introducing a cysteine residue at an appropriate site in a helix and then analyzing the resulting disulfide bridge. This approach has limitations like inability to form proper disulfide bond among others.

In this regard, recently Kim et al. [91] developed ACCORD (Assessment tool for homodimeric Coiled-coil Orientation Decision), a biochemical method using a fusion tag for assessing differences between parallel and antiparallel CC homodimers. When tested on 15 different CC proteins with known structures, the ACCORD was able to correctly identify the orientations of these proteins. Furthermore, ACCORD was able to determine the orientation of an unknown CC domain that was later confirmed by X-ray crystallography and small angle X-ray scattering. It might be a good idea to use ACCORD in conjunction with other computational approaches. It has to be noted that ACCORD might not be able to assess antiparallel short CCs.

Comparative Performance of Methods for Coiled-Coil Prediction

There have been a few efforts [88, 92] to analyze the comparative performance of methods for coiled-coil prediction. Gruber et al. [92] in 2006 compared commonly used coiled-coil prediction methods against a database derived from proteins of known structure and observed two older programs COILS and PairCoil/Multi-coil are significantly outperformed by two recent developments: Marcoil, a program built on hidden Markov models, and PCOILS, a new COILS version that uses profiles as inputs, and to a lesser extent by a PairCoil update, PairCoil2.

Overall, Marcoil provided a slightly better performance over the reference database than PCOILS and is considerably faster, but it is sensitive to highly charged false positives, whereas the weighting option of PCOILS allows the identification of such sequences.

More recently, Li et al. [88] compared 12 sequence-based bioinformatics tools to predict coiled coil. These methods included COILS [28], PCOILS [18], Paircoil2 [29], SOSUICoil [9],

MARCOIL [30], CCHMM_PROF [31], SpiriCoil [32], SCORER 2.0 [33], LOGICOIL [34], PrOCoil [35], RFCoil [36], and Multicoil2 [37]. As eluded above, Li et al. categorized the tools into two classes: coiled-coil detection and coiled-coil oligomeric state prediction.

Based on independent test analysis, it was observed that LOGICOIL achieved the overall highest AUC values for predicting parallel dimeric and trimeric coiled coils. For the coiled-coil detection, Multicoil2 [37] achieved the highest AUC values. It was also noted that the prediction of coiled coil was quite inconsistent among the compared methods highlighting the fact that developing a consensus-based approach could be difficult. In addition, based on this study, we can conclude that the newer methods are outperforming the older methods in general and that the coiled-coil prediction is still a challenging task. It would also be interesting to compare the prediction of orientation of helix of these methods.

Comparative Visualization of Coiled Coil: Waggawagga

More recently, Simm et al. [87] developed Waggawagga that allows the comparative analysis of six coiled-coil predictions (Marcoil, Multicoil [89], Multicoil2, Ncoils, Paircoil, Paircoil2) and three oligomerization state prediction programs (Scorer, PrOCoil, and LOGICOIL). The tools represent coiled coil using helical-wheel and helical-net representation. The method is available as a web server at <https://waggawagga.motorprotein.de/>.

In addition, Multicoil2 distinguishes dimers, trimers, and non-coiled-coil oligomerization states. These tools can be run in any combination against single or multiple query sequences.

Building, Designing, and Assessing Coiled Coil: CCBuilder 2.0

Wood et al. in 2018 developed CCBuilder 2.0 [93], a web-based application that generates all-atom model (backbone and side chain) of coiled coils in various conformations. The webserver also provides scores that are the feasibility measure of backbone and other force field terms. In addition, the builder is also able to build home- and hetero-oligomeric coiled coils in parallel and antiparallel conformation. CCBuilder2.0 generates coiled-coil backbones, builds side chains onto these frameworks, and provides a range of metrics to measure the quality of the models. This tool can prove to be valuable for designing as well as engineering novel proteins.

4.1.4 Helix Capping and β -Turn Prediction from NMR Chemical Shifts

Shen et al. [94] presented an empirical method for identification of distinct structural motifs in proteins on the basis of experimentally determined backbone and $^{13}\text{C}^\beta$ chemical shifts. Elements identified include the N-terminal and C-terminal helix capping motifs and five types of β -turns: I, II, I', II', and VIII. Using a database of proteins of known structure, the NMR chemical shifts, together with the PDB-extracted amino acid preference of the helix capping and β -turn motifs are used as input data for training an artificial

neural network algorithm, which outputs the statistical probability of finding each motif at any given position in the protein. The trained neural networks, contained in the MICS (motif identification from chemical shifts) program, also provide a confidence level for each of their predictions, and values ranging from 0.7 to 0.9 for the Matthews correlation coefficient of its predictions far exceed those attainable by sequence analysis.

4.1.5 Database of Simple Super-Secondary Structure Elements: ArchDB2014

Various super-secondary structure prediction methods [95, 96] use ArchDB as a database to extract training and test data. In this regard, we will briefly describe the ArchDB2014 [97] which is a newer version of the dataset. ArchDB2014 consists of ten loop types: alpha-alpha (HH), alpha-beta (HE), beta-alpha (EH), beta-beta hairpin (BN), beta-beta link (BK), beta-helix₃₁₀ (EG), helix₃₁₀-beta (GE), helix₃₁₀-helix (GH), helix-helix₃₁₀ (HG), and helix₃₁₀-helix₃₁₀ (GG). This dataset will possibly help in the development of computational methods for identification of various super-secondary structures including secondary structure with 3₁₀ helices.

4.2 Prediction Methods for (Simple) Multiple Super-Secondary Structures

In addition to methods that predict one type of super-secondary structures, there are few methods that predict multiple super-secondary structures. One of the methods that predicts multiple super-secondary structures is by Zou et al. [95]. According to the regular secondary structures connected by loops, super-secondary structures are divided into β-β (β-loop-beta), β-α (β-loop-alpha), α-β (α-loop-beta), and α-α (α-loop-alpha). This method uses other sequence-based features including PseAAC (pseudo-amino acid composition) in the SVM as well as IDQD (increment of diversity combined with quadratic analysis) and predicts the four different types of super-secondary structures quite well.

Recently, Feng and Kou [96] developed a quadratic discriminant-based algorithm to predict five kinds of *simple* super-secondary structures using chemical shifts. By analyzing the statistical distribution of the chemical shifts in these five simple super-secondary structures (namely, 110 α-loop-α, 93 α-loop-β, 110 β-loop- α, 75 β-loop-β-hairpin, and 157 β-loop-β-link in a set of 123 proteins) and then combining the chemical shift as features with quadratic discriminant analysis, they developed a method for predicting these five different types of simple super-secondary structures.

4.3 Prediction Methods for Complex Super-Secondary Structures

Prediction of complex super-secondary structures is a necessary step in the study of protein tertiary structures [98]. When complex super-secondary structures are predicted accurately, it helps in recognizing tertiary structures and functions of proteins in general. Since super-secondary structures are formed by connecting two or more secondary structures, complex super-secondary structures are

also formed when two or more super-secondary structures fold into a single motif [99]. The accurate prediction of oligomeric state coiled-coil structures has helped in the general structure and function of influenza virus hemagglutinin [99, 100]. Also, heptad repeat units consistent with α helical coiled-coil conformations are located in amino acid sequences of paramyxovirus which is present in measles viruses [99–101]. Thus, we see here how knowledge of super-secondary structures has influenced the identification and function of proteins. Similarly, knowledge about complex super-secondary structures such as $\beta\alpha\beta$ helps significantly in tertiary structure prediction since many functional and active sites often occur in the polypeptides of $\beta\alpha\beta$. Some examples of complex super-secondary structures include $\beta\alpha\beta$, Rossmann fold, β -meander, four helix bundle ($\alpha\alpha\alpha\alpha$), and Greek key ($\beta\beta\beta\beta$). Methods that predict complex super-secondary structures are very important; thus, we present a brief description of a few.

4.3.1 Prediction of $\beta\alpha\beta$ Complex Super-Secondary Structure

$\beta\alpha\beta$ is an important complex super-secondary structure in proteins. Many functional sites and active sites often occur in polypeptides of $\beta\alpha\beta$ motifs. Therefore, the accurate prediction of $\beta\alpha\beta$ motifs is very important to recognizing protein tertiary structure and the study of protein function.

Computational prediction of $\beta\alpha\beta$ started in 1983 when Janet Thornton's group predicted $\beta\alpha\beta$ motifs with accuracy of 70% [102]. Recently, Sun et al. [103] in 2016 developed a SVM-based method for prediction of the complex super-secondary structure, $\beta\alpha\beta$. Initially, a $\beta\alpha\beta$ motif dataset was constructed from SCOP database resulting in 1363 protein chains that contained at least one $\beta\alpha\beta$ motif resulting in 4277 $\beta\alpha\beta$ motifs where the length of loop- α -loop varied from 10 to 26 amino acids. Then, using the sequence-based features, functional information, and predicted structure-based information, a SVM-based machine learning algorithm was trained. The prediction accuracy for this was reported to be 81.7% for fivefold cross-validation and 76.7% for independent test.

4.3.2 Prediction of Rossmann Fold and Cofactor Specificity: Complex Super-Secondary Structure

Rossmann fold is a type of complex super-secondary structure. Geertz-Hansen et al. in 2014 developed Cofactory, a sequence based predictor for cofactor binding Rossmann folds [104]. This method first identifies cofactor binding Rossmann folds and then predicts the specificity for the cofactors. The identification of cofactor-binding Rossmann fold is performed using hidden Markov models, and the cofactor specificity prediction artificial neural network is used for specificity prediction. The Cofactory method is available at <http://www.cbs.dtu.dk/services/Cofactory>.

4.4 Prediction/ Analysis of Protein Structures Composed of Repeats of Simple Super-Secondary Structure

Repeats are ubiquitous phenomena in proteins. The repeated units can be classified as short, intermediate, or long based on the number of residues. Repeats that fold into single domains yield a hierarchy of structural complexity, from fibrous domains made by the repetition of patterns only a few amino acids long (collagen, coiled coils, β -helices), to solenoid domains formed by the repetition of simple super-secondary structures ($\alpha\alpha$ -hairpins, tetratricopeptide and HEAT repeats; $\beta\beta$ -hairpins, choline-binding domains; $\beta\alpha$ -hairpins, leucine-rich repeats), to globular domains formed by the repetition, frequently in interleaved form, of complex super-secondary structure units ($\beta\beta\alpha\beta$, cradle-loop barrels; $\beta\alpha\beta$, ferredoxins). Toroids are intermediate in complexity between solenoids and globular proteins, as they are usually formed by simple, non-interleaved super-secondary structure units but fold into a closed, rather than open, structure ($\alpha\alpha$ -hairpins, protein prenyltransferases; $\beta\beta$ -hairpins, porins; $\beta\alpha$ -hairpins, TIM barrels).

In this regard, reuse of super-secondary structure subunits is also a recurrent theme in protein evolution. This reuse can form tandem repeats with both closed (β -propeller, β -trefoils) and open structures (ARM, ankyrin repeats) [105]. It has to be noted here that symmetry detection algorithm SymD can detect/analyze whether a structure is closed or open.

Repeated use of small super-secondary structure element (SSE) subunits is a recurrent feature in protein evolution [106]. These subunits can duplicate to form interleaved, globular domains or can form higher-order tandem repeat in protein (TRP) structures. TRP structures can adopt closed (i.e., where both termini of the TRP are near in space) repeats such as the β -propellers or β -trefoils or open repeats (i.e., where both termini are distant in space) such as the ARM or ankyrin repeats [107]. In addition, the TRP can form solenoid domains by the repetition of simple super-secondary structures ($\alpha\alpha$ -hairpins, tetratricopeptide and HEAT repeats; $\beta\beta$ -hairpins, choline-binding domains, porins; $\beta\alpha$ -hairpins, TIM barrels) [108]. Some of these structures are open while some are closed. SymD [73] program developed by us can be used to analyze the repeats as well as detect the symmetry in these proteins.

4.5 Prediction/ Analysis of Protein Structures Composed of Repeats of Complex Super-Secondary Structure

There are few protein classes that are made up of repeats of complex super-secondary structures occur frequently in protein universe. One of the major classes of this type of protein is transmembrane protein (TMP). There are two major types of TMP: α -helix bundles (long stretches of polar amino acids, fold into transmembrane alpha-helices, e.g., cell surface receptors, ion channels, active and passive transporters) and beta-barrels (anti-parallel β -sheets rolled into a cylinder form, e.g., outer membrane of Gram-negative bacteria). Here we shortly describe prediction methods for TMP proteins.

4.5.1 β -Propellers

β -Propellers are toroidal folds, in which repeated, four-stranded beta-meanders are arranged in a circular and slightly tilted fashion, like the blades of a propeller with varying numbers of repeats. For a comprehensive review of β -propeller fold and its evolution, refer to Chaudhuri et al. [108]. A four-bladed B-propeller is shown in Fig. 1.

4.5.2 β -Barrels

These TMP proteins are created by a succession of antiparallel-paired β -strands forming a channel; in that regard, a β -barrel can be considered as a self-closed β -sheet. The observed structures are formed by 8–22 β -strands that incline at an angle of 20° to 45° with respect to the barrel axis. Each of these β -strands comprises about 9–11 residues. While 8 appears to be the lower bound on the number of necessary β -strands to form a channel [109], the upper bound of 22 is only obtained by experimental observation [99]. The β -barrels are usually constituted by an even number of β -strands, which allows an antiparallel pairing at the barrel closure. In TMB proteins, β -barrels are found more commonly in such a way that β -strands are paired in antiparallel manner to each other. The most popular super-secondary structures are those containing disjoint Greek key motifs.

4.5.3 Membrane Proteins

Solved crystal structures of membrane protein (MPs) structures lag other types of proteins due to various difficulties associated with the structure determination. In this regard, prediction of membrane proteins is an important topic in protein structure prediction. Lehman et al. [109] recently did an exhaustive review of computational tools for modeling of membrane proteins. Moreover, in 2017, Venko et al. [110] also reviewed computational approaches for structure prediction of membrane transporters.

As discussed earlier, membrane proteins can be broadly classified as β -barrels membrane protein and α -helical membrane proteins, and majority of the MPs belong to α -helical bundles. Transfold [111] is one of the earlier methods that predicted the β -barrel structure based on a sequence and predicted the super-secondary structure of MP β -barrels such as secondary structure, TM topology, residue contacts, side-chain orientation, and strand angles with respect to the membrane. Another predictor, the BBP (BetaBarrelPredictor) [112] uses a graph-theoretic approach to classify β -barrels and model their super-secondary structure. Another predictor BETAWARE [113] is a machine learning-based tool to detect and predict TMP in prokaryotes.

4.6 Symmetry of SSS Elements, LSSP, and Super-Secondary Structure

Analysis of protein structures composed of repeated complex super-secondary structure is an important task in structural biology. In this regard, we recently developed a symmetry detection program SymD [73] that can detect symmetry in proteins and can be used to analyze the symmetry in various types of proteins. In addition,

SymD can also be used to find repeating units (manuscript in preparation) in proteins. These repeating units often correspond to super-secondary structure and/or structural alphabets. SymD is available at <https://synd.nci.nih.gov/>. BK Lee called these repeating units as LSSP (*locally structural similar pieces*).

Mackenzie et al. [114] decomposed the known protein structures into so-called tertiary structural motif (TERMs). A TERM is defined as a compact backbone fragment of secondary, tertiary, and quaternary environments around a given residue, comprising one or more disjoint segments (three on average). Based on their analysis, they observed that only <600 TERMs are sufficient to describe 50% of the PDB at sub-Angstrom resolution.

4.7 Representation and Visualization of Super-Secondary Structure

Another important aspect of super-secondary structure prediction/analysis is representation and visualization of super-secondary structure. In this regard, one of the ways to represent protein super-secondary structure (SSS) is the use of protein graph. In a protein graph (PG), the vertices represent secondary structure elements, and the edges represent the spatial neighborhood. There are a few tools for visualization of SSS topologies like HERA and PROMOTIF. The database PTGL [115] also generates SSS cartoons based on the graph-theoretic description. Readers are advised to refer to Koch and Schafer [116] for details of protein graph and other aspects of describing and visualizing the super-secondary structures.

4.8 Chemical Shift as Emerging Feature for Protein Super-Secondary Structure Prediction

The existing approaches for predicting super-secondary structure primarily use amino acid sequence features. Chemical shift is an emerging feature for protein super-secondary structure prediction. Just like methods for secondary structure assignment that use chemical shift TALOS+[117] and TALOS-N [118], recently Hafsa et al. developed a webserver called CSI3.0 [119] (Chemical Shift Index or CSI3.0) to identify the location of secondary and super-secondary structure using nuclear magnetic resonance (NMR) backbone chemical shifts and the protein sequence data. The CSI3.0 is available at <http://csi3.wishartlab.com>. Moreover, we expect to see a lot of new methods for prediction/analysis of complex super-secondary structure (see Notes 3–5).

5 Application of Super-Secondary Structure Prediction

Protein super-secondary structure prediction is a very important step in the general protein structure prediction pipeline. Besides, there are some prediction tools that explicitly use super-secondary structures. Super-secondary structures have been used in the

modeling of intermediate filaments and prediction of protein-protein interactions, among others. In this section, we consider super-secondary structure prediction and its application in structure prediction and analysis.

5.1 Protein Structure Prediction Using Super-Secondary Structure Information

5.1.1 I-TASSER

In this section, we describe various protein structure prediction methods that use super-secondary structure information.

In I-TASSER [7], the query sequence of the target protein is first threaded through the PDB library by LOMETS [30, 31], a meta-threading approach containing multiple individual threading programs to identify possible template structures. In addition, due to the fact that it is often difficult to identify distant-homology templates from global sequence threading, I-TASSER also employs SEGMER [120] to identify substructure motifs by segmental threading which in turn correspond to super-secondary structures. The target sequence is split into segments of two to four consecutive or nonconsecutive secondary structural elements (super-secondary structures), which are then threaded through the PDB to identify appropriate substructure motifs by MUSTER. Then, spatial restraints derived from high-scoring motifs are incorporated into the global template-based restraints from LOMETS to guide the I-TASSER simulations.

The online webserver for SEGMER is available at <https://zhanglab.ccmb.med.umich.edu/SEGMER/>. SEGMER is a segmental threading algorithm that recognizes substructure motifs from the PDB library. The server first splits the target sequence into segments that consist of 2–4 consecutive (or non-consecutive) secondary structures that mostly correspond to super secondary structure.

5.1.2 QUARK

QUARK [22] breaks the query sequences into fragments of 1–20 residues (many of them belong to the super-secondary structures category) where multiple fragment residues are retrieved at each position from unrelated experimental structures. Then, these fragments are assembled into a full length using replica-exchange Monte Carlo simulations guided by a composite knowledge-based force field [22].

5.1.3 Rosetta

Rosetta[40] tries to identify a set of fragments (three or nine continuous residues) from templates which are in turn used to assemble the overall structures. These fragments sometime correspond to super-secondary structure elements.

5.1.4 CCFOLD: Prediction of Intermediate Filament Protein Classes

Recently, Guzenko and Strelkov developed a threading-based algorithm that produces coiled-coil models called CCFold [121]. CCFold picks multiple CC fragments for short overlapping segments of the input sequence. Finally, the fragments are scored based on the statistics obtained from known CC folds, and then the

best-scoring fragments are merged together using an optimization procedure. When benchmarked against other method, this method was able to produce better results in terms of both local and global similarity measures.

Additionally, CCFold was also used to model intermediate filaments. By combining CCFold with Rosetta folding, the authors also generated representative dimer models for all intermediate filaments. The method is available as webserver at <http://parm.kuleuven.be/Biocystallography/cc>.

5.2 Protein-Structure Prediction Using Super-Secondary Structures

5.2.1 Smotif (Super-Secondary Structure)-Based Approach

In this section, we describe protein structure prediction methods that use super-secondary structures.

Another seminal work in the application of super-secondary structure is the work by Andras Fiser's group. This group uses super-secondary structure motifs (Smotifs) to predict protein structure. In their work, super-secondary structure motifs (Smotifs) are defined systematically as two secondary structures with a connecting loop. In that regard, they have built a library containing clusters of Smotifs with similar internal geometry and observed that new folds discovered during the last decade did not require the emergence of new Smotifs but are simply a consequence of novel combinations of existing Smotifs [122].

This observation presents a hypothesis according to which it should be possible to build any new structure of a known or yet-to-be-discovered fold by combining existing Smotifs from already known structures. The library of Smotifs is a backbone-only, geometrically defined fragment library, which means that for practical modeling applications, a relation needs to be made between the target protein and specific fragments in the library.

Given these properties, Smotifs can be used as basic building blocks for sampling native-like protein structures and replacing smaller fragment libraries. Fiser's group developed a structure prediction using Smotifs for template-free modeling called template-free modeling (SmotifTF) [123]. Recently, they also developed chemical shift-guided Smotif assembly program called SmotifCS [124]. It was shown for a set of proteins that these two methods performed in par with current state-of-the-art software such as I-TASSER [7], HHpred [29], and Rosetta [40] but were reliant on Smotif libraries specifically generated for the chosen target, making the libraries non-universal. In the case of SmotifTF [123], the Smotif libraries are generated from *homologous protein* structures, while the SmotifCS [124] approach selects the Smotifs that locally match the experimentally observed chemical shifts. The size of the protein that could be solved using this approach is around 110 residues.

5.2.2 Predicting Structure Using Super-Secondary Structural Fragments

Another seminal work in the application of super-secondary structure for protein structure prediction is FRAGFOLD[125] by Jones' group. The origin of the FRAGFOLD can be attributed to the 1997 work of Jones [126]. In this work, an ab initio prediction approach was developed for pig NK-lysin protein based on the assembly of recognized super-secondary structural fragments. Later in 2001, and in 2003 [43], a general purpose protein tertiary structure prediction method called FRAGFOLD was proposed based on the assembly of super-secondary structural fragments that are taken from high-resolution protein structures using a simulated annealing [125]. Essentially, using super-secondary structural fragments from a library of high-quality protein structures and deriving pairwise potentials from these structures, the group was able to perform quite well in CASP structure prediction. The super-secondary structures defined in the original FRAGFOLD [43] are α -hairpin, α -corner, β -hairpin, β -corner, β - α - β unit, and split β - α - β unit. Since then numerous variants of FRAGFOLD algorithm have been developed.

5.2.3 FRAGFOLD Combined with Contact Prediction for Globular Proteins

As with other approaches, one of the recent trends in protein structure prediction is the use of predicted residue contacts to boost the quality of the protein structure prediction. In this regard, Jones group also in 2014 [127] combined FRAGFOLD with PSICOV [128], a contact prediction approach, and developed a method for globular proteins that could improve the protein structure prediction. On a dataset of 150 diverse globular proteins, the method was able to predict structure with reasonable predictions.

5.2.4 De Novo Prediction of Transmembrane Proteins Using Fragment Assembly and Contact Prediction

In 2003, Jones Group developed FILM [129], a FRAGFOLD-based approach to assemble folds from super-secondary structural fragments using simulated annealing. Furthermore, in 2012, the same group developed FILM3 [130], a de novo prediction approach for transmembrane proteins based on FRAGFOLD and using contacts predicted using a correlated mutation-based approach called PSICOV [128] as scoring function. This method is able to generate reasonable models for 28 membrane proteins with diverse range of topologies.

5.3 Protein–Protein Interaction Prediction

Protein–protein interactions are essential in many biological functions, for instance, in regulating enzymatic activity and mediating the assembly of protein complexes. Based on the observation that the physical interaction between proteins is sometimes mediated by matching coiled-coil regions, Mier et al. [131] developed a protein–protein interaction prediction approach using coiled-coil evolution patterns.

This approach is based on the hypothesis that the emergence of the interacting CCs of interaction partners are correlated and that the presence or absence of correlated CC in diverse species can be

used as a predictor of the interaction. Coiled-coil protein super-secondary structures sometimes mediate protein-protein interactions. Hence, they are directly related when it comes to determining the function of proteins. Thus, the evolutionary correlation of coiled coils in two proteins can be used as a hint that these two proteins interact.

5.4 Recent Trends in Prediction of Protein Structures Using Super-Secondary Structure

5.4.1 Modeling Proteins Using Super-Secondary Structure Library and NMR Chemical Shift

5.4.2 Protein Structure Determination by Assembling Super-Secondary Structure Motif Using Pseudocontact Shifts

In this section, we describe a few recent trends in prediction of protein structures using super-secondary structure (also see Note 6).

As discussed earlier, another seminal work in the application of super-secondary structure for protein structure prediction is the work by Andras Fiser's group. This group uses super-secondary structure motifs (Smotifs) to predict protein structure. In the absence of any sequence similarity signal, limited experimental data can be used to relate the backbone conformations. In this regard, this group presented a modeling algorithm that combines an exhaustive Smotif library and NMR chemical shift patterns [124]. In a test of 102 proteins with unique folds, the approach was able to generate homology quality models for 90 proteins.

Pilla et al. [132] developed DINGOPCS (3-Dimensional assembly of Individual Smotifs to Near-native Geometry as orchestrated by Pseudocontact Shifts), a computational and nuclear magnetic resonance hybrid approach for assembling the 3D structure of a protein from its constituent super-secondary structural motifs (Smotifs) and pseudocontact shifts (PCs) restraints for backbone amide from NMR. Based on the benchmark, this method was able to identify near-native Smotifs for nine out of ten cases for proteins ranging from 100 to 220 residues with various topologies.

6 Notes

1. The Critical Assessment of protein Structure Prediction (CASP) experiments have been a great platform in promoting and assessing the performance of different structure prediction methods. The most recent (as of April 2018) is CASP12, and significant advances have been recorded. There has been a twofold improvement in contact accuracy prediction using new methods for three-dimensional contact predictions. This has resulted in dramatic improvements in model accuracy for proteins where no templates are available. Furthermore, there has been an overall improvement in accuracy for models based on structural templates.
2. Methods for prediction of super-secondary structures broadly can be classified as (a) methods for prediction of simple super-

secondary structures and (b) methods for prediction of complex super-secondary structures. It can also be noted that some protein structures are made up of repeats of simple super-secondary structures and some protein structures are made up of repeats of complex super-secondary structures.

3. In the last few years, not much development has been observed in the arena of simple super-secondary structure prediction, except for coiled coil. This can be partly attributed to the fact that accuracy of methods for prediction of simple super-secondary structure is quite acceptable.
4. Recently, most development in super-secondary structure prediction is in coiled-coil arena and complex super-secondary structure prediction. About eight sequence-based coiled-coil predictors have been developed over the past few years. The most cited coiled-coil predictor currently is LogiCoil and has about 52 citations. Most of the focus of these predictors is in the prediction of the oligomerization state of the coiled coils.
5. There is still a lot of room for improvements in the prediction accuracy of complex super-secondary structure prediction methods, and significant interest and development in the prediction of complex super-secondary structure prediction can be expected.
6. As in other areas of structure prediction, use of extra information (e.g., chemical shift obtained from NMR or some other experimental techniques) is an emerging theme to improve the performance accuracy of the methods for the prediction of super-secondary structures as well as prediction of protein structures.

Funding

D.B.K.C. is partly supported by a start-up grant from the Department of Computational Science and Engineering at North Carolina A&T State University. D.B.K.C. is also partly supported by NSF grant no. 1564606 and NSF grant no. 1647884.

References

1. Dorn M, e Silva MB, Buriol LS, Lamb LC (2014) Three-dimensional protein structure prediction: methods and computational strategies. *Comput Biol Chem* 53:251–276
2. Kc DB (2016) Recent advances in sequence-based protein structure prediction. *Brief Bioinform* 18:1021–1032
3. Pruitt KD, Tatusova T, Klimke W, Maglott DR (2008) NCBI reference sequences: current status, policy and new initiatives. *Nucleic Acids Res* 37:D32–D36
4. Kc DB (2017) Recent advances in sequence-based protein structure prediction. *Brief Bioinform* 18:1021–1032
5. Chen K, Kurgan L (2012) Computational prediction of secondary and supersecondary structures, *Protein supersecondary structures*. Springer, New York, pp 63–86

6. Yang Y, Faraggi E, Zhao H, Zhou Y (2011) Improving protein fold recognition and template-based modeling by employing probabilistic-based matching between predicted one-dimensional structural properties of query and corresponding native properties of templates. *Bioinformatics* 27:2076–2082
7. Roy A, Kucukural A, Zhang Y (2010) I-TASSER: a unified platform for automated protein structure and function prediction. *Nat Protoc* 5:725
8. Faraggi E, Yang Y, Zhang S, Zhou Y (2009) Predicting continuous local structure and the effect of its substitution for secondary structure in fragment-free protein structure prediction. *Structure* 17:1515–1527
9. Wu S, Zhang Y (2008) MUSTER: improving protein sequence profile-profile alignments by using multiple sources of structure information. *Proteins* 72:547–556
10. Zhou H, Skolnick J (2007) Ab initio protein structure prediction using chunk-TASSER. *Biophys J* 93:1510–1518
11. Skolnick J (2006) In quest of an empirical potential for protein structure prediction. *Curr Opin Struct Biol* 16:166–171
12. Yang Y, Gao J, Wang J, Heffernan R, Hanson J et al (2018) Sixty-five years of the long march in protein secondary structure prediction: the final stretch? *Brief Bioinform* 19(3):482–494
13. Anfinsen CB, Haber E, Sela M, White F (1961) The kinetics of formation of native ribonuclease during oxidation of the reduced polypeptide chain. *Proc Natl Acad Sci* 47:1309–1314
14. Anfinsen CB (1973) Principles that govern the folding of protein chains. *Science* 181:223–230
15. Singh M (2006) Predicting protein secondary and supersecondary structure. In: Aluru S (ed) *Handbook of computational molecular biology*. Chapman and Hall/CRC Press, Boca Raton, pp 29.1–29.29
16. Zhang Y (2008) Progress and challenges in protein structure prediction. *Curr Opin Struct Biol* 18:342–348
17. Sali A, Blundell TL (1993) Comparative protein modelling by satisfaction of spatial restraints. *J Mol Biol* 234:779–815
18. Bowie JU, Luthy R, Eisenberg D (1991) A method to identify protein sequences that fold into a known three-dimensional structure. *Science* 253:164–170
19. Liwo A, Lee J, Ripoll DR, Pillardy J, Scheraga HA (1999) Protein structure prediction by global optimization of a potential energy function. *Proc Natl Acad Sci U S A* 96:5482–5485
20. Kryshtafovych A, Fidelis K, Moult J (2011) CASP9 results compared to those of previous CASP experiments. *Proteins* 79(Suppl 10):196–207
21. Simons KT, Kooperberg C, Huang E, Baker D (1997) Assembly of protein tertiary structures from fragments with similar local sequences using simulated annealing and Bayesian scoring functions. *J Mol Biol* 268:209–225
22. Xu D, Zhang Y (2012) Ab initio protein structure assembly using continuous structure fragments and optimized knowledge-based force field. *Proteins* 80:1715–1735
23. Baker D, Sali A (2001) Protein structure prediction and structural genomics. *Science* 294:93–96
24. Tramontano A, Morea V (2003) Assessment of homology-based predictions in CASP5. *Proteins* 53(Suppl 6):352–368
25. Rost B (1999) Twilight zone of protein sequence alignments. *Protein Eng* 12:85–94
26. Browne WJ, North AC, Phillips DC, Brew K, Vanaman TC et al (1969) A possible three-dimensional structure of bovine alpha-lactalbumin based on that of hen's egg-white lysozyme. *J Mol Biol* 42:65–86
27. Yang J, Zhang W, He B, Walker SE, Zhang H et al (2016) Template-based protein structure prediction in CASP11 and retrospect of I-TASSER in the last decade. *Proteins* 84(Suppl 1):233–246
28. Altschul SF, Madden TL, Schaffer AA, Zhang J, Zhang Z et al (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res* 25:3389–3402
29. Soding J (2005) Protein homology detection by HMM-HMM comparison. *Bioinformatics* 21:951–960
30. Ginalski K, Elofsson A, Fischer D, Rychlewski L (2003) 3D-Jury: a simple approach to improve protein structure predictions. *Bioinformatics* 19:1015–1018
31. Wu S, Zhang Y (2007) LOMETS: A local meta-threading-server for protein structure prediction. *Nucleic Acids Res* 35:3375–3382
32. Webb B, Sali A (2014) Comparative protein structure modeling using MODELLER. *Curr Protoc Bioinformatics* 47:5.6.1–5.6.32
33. Pieper U, Webb BM, Dong GQ, Schneidman-Duhovny D, Fan H et al (2014) ModBase, a database of annotated comparative protein structure models and associated resources. *Nucleic Acids Res* 42:D336–D346

34. Eswar N, John B, Mirkovic N, Fiser A, Ilyin VA et al (2003) Tools for comparative protein structure modeling and analysis. *Nucleic Acids Res* 31:3375–3380
35. Smith TF, Waterman MS (1981) Identification of common molecular subsequences. *J Mol Biol* 147:195–197
36. Remmert M, Biegert A, Hauser A, Soding J (2012) HHblits: lightning-fast iterative protein sequence searching by HMM-HMM alignment. *Nat Methods* 9:173–175
37. Zhang Y, Skolnick J (2004) Automated structure prediction of weakly homologous proteins on a genomic scale. *Proc Natl Acad Sci U S A* 101:7594–7599
38. Bowie JU, Eisenberg D (1994) An evolutionary approach to folding small alpha-helical proteins that uses sequence information and an empirical guiding fitness function. *Proc Natl Acad Sci U S A* 91:4436–4440
39. Bradley P, Misura KM, Baker D (2005) Toward high-resolution de novo structure prediction for small proteins. *Science* 309:1868–1871
40. Rohl CA, Strauss CE, Misura KM, Baker D (2004) Protein structure prediction using Rosetta. *Methods Enzymol* 383:66–93
41. Holmes JB, Tsai J (2004) Some fundamental aspects of building protein structures from fragment libraries. *Protein Sci* 13:1636–1650
42. Gront D, Kulp DW, Vernon RM, Strauss CE, Baker D (2011) Generalized fragment picking in Rosetta: design, protocols and applications. *PLoS One* 6:e23294
43. Jones DT, McGuffin LJ (2003) Assembling novel protein folds from super-secondary structural fragments. *Proteins* 53(Suppl 6):480–485
44. Kalev I, Habeck M (2011) HHfrag: HMM-based fragment detection using HHpred. *Bioinformatics* 27:3110–3116
45. Shen Y, Picord G, Guyon F, Tuffery P (2013) Detecting protein candidate fragments using a structural alphabet profile comparison approach. *PLoS One* 8:e80493
46. Xu D, Zhang Y (2013) Toward optimal fragment generations for ab initio protein structure assembly. *Proteins* 81:229–239
47. Bystroff C, Simons KT, Han KF, Baker D (1996) Local sequence-structure correlations in proteins. *Curr Opin Biotechnol* 7:417–421
48. Mackenzie CO, Grigoryan G (2017) Protein structural motifs in prediction and design. *Curr Opin Struct Biol* 44:161–167
49. Moult J (2005) A decade of CASP: progress, bottlenecks and prognosis in protein structure prediction. *Curr Opin Struct Biol* 15:285–289
50. 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
51. Orengo CA, Michie A, Jones S, Jones DT, Swindells M et al (1997) CATH—a hierachic classification of protein domain structures. *Structure* 5:1093–1109
52. Andreeva A, Howorth D, Chandonia JM, Brenner SE, Hubbard TJ et al (2007) Data growth and its impact on the SCOP database: new developments. *Nucleic Acids Res* 36:D419–D425
53. Cuff AL, Sillitoe I, Lewis T, Clegg AB, Rentzsch R et al (2010) Extending CATH: increasing coverage of the protein structure universe and linking structure with function. *Nucleic Acids Res* 39:D420–D426
54. Kolodny R, Honig B (2006) VISTAL—a new 2D visualization tool of protein 3D structural alignments. *Bioinformatics* 22:2166–2167
55. Moreland JL, Gramada A, Buzko OV, Zhang Q, Bourne PE (2005) The Molecular Biology Toolkit (MBT): a modular platform for developing molecular visualization applications. *BMC Bioinformatics* 6:21
56. Eisenberg D (2003) The discovery of the α -helix and β -sheet, the principal structural features of proteins. *Proc Natl Acad Sci* 100:11207–11210
57. Levitt M, Greer J (1977) Automatic identification of secondary structure in globular proteins. *J Mol Biol* 114:181–239
58. Frishman D, Argos P (1995) Knowledge-based protein secondary structure assignment. *Proteins* 23:566–579
59. Kabsch W, Sander C (1983) Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers* 22:2577–2637
60. Richards FM, Kundrot CE (1988) Identification of structural motifs from protein coordinate data: secondary structure and first-level supersecondary structure. *Proteins* 3:71–84
61. 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
62. Labesse G, Colloc'h N, Pothier J, Mornon J-P (1997) P-SEA: a new efficient assignment of secondary structure from $\text{C}\alpha$ trace of proteins. *Bioinformatics* 13:291–295
63. Zhang W, Dunker AK, Zhou Y (2008) Assessing secondary structure assignment of

- protein structures by using pairwise sequence-alignment benchmarks. *Proteins* 71:61–67
64. Hosseini S-R, Sadeghi M, Pezeshk H, Eslahchi C, Habibi M (2008) PROSIGN: a method for protein secondary structure assignment based on three-dimensional coordinates of consecutive C α atoms. *Comput Biol Chem* 32:406–411
 65. Park S-Y, Yoo M-J, Shin J-M, Cho K-H (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
 66. Martin J, Letellier G, Marin A, Taly J-F, de Brevern AG et al (2005) Protein secondary structure assignment revisited: a detailed analysis of different assignment methods. *BMC Struct Biol* 5:17
 67. King SM, Johnson WC (1999) Assigning secondary structure from protein coordinate data. *Proteins* 35:313–320
 68. Fodje M, Al-Karadaghi S (2002) Occurrence, conformational features and amino acid propensities for the π -helix. *Protein Eng Des Sel* 15:353–358
 69. Majumdar I, Krishna SS, Grishin NV (2005) PALSSE: a program to delineate linear secondary structural elements from protein structures. *BMC Bioinformatics* 6:202
 70. Cubellis MV, Cailliez F, Lovell SC (2005) Secondary structure assignment that accurately reflects physical and evolutionary characteristics. *BMC Bioinformatics* 6:S8
 71. Wang S, Peng J, Ma J, Xu J (2016) Protein secondary structure prediction using deep convolutional neural fields. *Sci Rep* 6:18962
 72. Kuhn M, Meiler J, Baker D (2004) Strand-loop-strand motifs: prediction of hairpins and diverging turns in proteins. *Proteins* 54:282–288
 73. Tai CH, Paul R, Dukka KC, Shilling JD, Lee B (2014) SymD webserver: a platform for detecting internally symmetric protein structures. *Nucleic Acids Res* 42:W296–W300
 74. de la Cruz X, Hutchinson EG, Shepherd A, Thornton JM (2002) Toward predicting protein topology: an approach to identifying beta hairpins. *Proc Natl Acad Sci U S A* 99:11157–11162
 75. de la Cruz X, Hutchinson EG, Shepherd A, Thornton JM (2002) Toward predicting protein topology: An approach to identifying β hairpins. *Proc Natl Acad Sci* 99:11157–11162
 76. Kumar M, Bhasin M, Natt NK, Raghava G (2005) BhairPred: prediction of β -hairpins in a protein from multiple alignment information using ANN and SVM techniques. *Nucleic Acids Res* 33:W154–W159
 77. Zou D, He Z, He J (2009) β -Hairpin prediction with quadratic discriminant analysis using diversity measure. *J Comput Chem* 30:2277–2284
 78. Xia J-F, Wu M, You Z-H, Zhao X-M, Li X-L (2010) Prediction of β -hairpins in proteins using physicochemical properties and structure information. *Protein Pept Lett* 17:1123–1128
 79. Xia JF, Wu M, You ZH, Zhao XM, Li XL (2010) Prediction of beta-hairpins in proteins using physicochemical properties and structure information. *Protein Pept Lett* 17:1123–1128
 80. Chen K, Kurgan L (2013) Computational prediction of secondary and supersecondary structures. *Methods Mol Biol* 932:63–86
 81. Li D, Hu X, Liu X, Feng Z, Ding C (2017) Using feature optimization-based support vector machine method to recognize the beta-hairpin motifs in enzymes. *Saudi J Biol Sci* 24:1361–1369
 82. Yong EF, GaoShan K (2015) Identify beta-hairpin motifs with quadratic discriminant algorithm based on the chemical shifts. *PLoS One* 10:e0139280
 83. Ferrer-Costa C, Shanahan HP, Jones S, Thornton JM (2005) HTQuery: a method for detecting DNA-binding proteins with a helix-turn-helix structural motif. *Bioinformatics* 21:3679–3680
 84. Fletcher JM, Boyle AL, Bruning M, Bartlett GJ, Vincent TL et al (2012) A basis set of de novo coiled-coil peptide oligomers for rational protein design and synthetic biology. *ACS Synth Biol* 1:240–250
 85. Li C, Wang XF, Chen Z, Zhang Z, Song J (2015) Computational characterization of parallel dimeric and trimeric coiled-coils using effective amino acid indices. *Mol BioSyst* 11:354–360
 86. Wang X, Zhou Y, Yan R (2015) AAFreqCoil: a new classifier to distinguish parallel dimeric and trimeric coiled-coils. *Mol BioSyst* 11:1794–1801
 87. Simm D, Hatje K, Kollmar M (2015) Wagga-wagga: comparative visualization of coiled-coil predictions and detection of stable single alpha-helices (SAH domains). *Bioinformatics* 31:767–769
 88. Li C, Ching Han Chang C, Nagel J, Porebski BT, Hayashida M et al (2016) Critical evaluation of in silico methods for prediction of

- coiled-coil domains in proteins. *Brief Bioinform* 17:270–282
89. Wolf E, Kim PS, Berger B (1997) MultiCoil: a program for predicting two- and three-stranded coiled-coils. *Protein Sci* 6:1179–1189
 90. Apgar JR, Gutwin KN, Keating AE (2008) Predicting helix orientation for coiled-coil dimers. *Proteins* 72:1048–1065
 91. Kim BW, Jung YO, Kim MK, Kwon DH, Park SH et al (2017) ACCORD: an assessment tool to determine the orientation of homodimeric coiled-coils. *Sci Rep* 7:43318
 92. Gruber M, Soding J, Lupas AN (2006) Comparative analysis of coiled-coil prediction methods. *J Struct Biol* 155:140–145
 93. Wood CW, Woolfson DN (2018) CCBUILDER 2.0: powerful and accessible coiled-coil modeling. *Protein Sci* 27:103–111
 94. Shen Y, Bax A (2012) Identification of helix capping and b-turn motifs from NMR chemical shifts. *J Biomol NMR* 52:211–232
 95. Zou D, He Z, He J, Xia Y (2011) Supersecondary structure prediction using Chou's pseudo amino acid composition. *J Comput Chem* 32:271–278
 96. Kou G, Feng Y (2015) Identify five kinds of simple super-secondary structures with quadratic discriminant algorithm based on the chemical shifts. *J Theor Biol* 380:392–398
 97. Bonet J, Planas-Iglesias J, Garcia-Garcia J, Marin-Lopez MA, Fernandez-Fuentes N et al (2014) ArchDB 2014: structural classification of loops in proteins. *Nucleic Acids Res* 42:D315–D319
 98. Sun L, Hu X, Li S, Jiang Z, Li K (2016) Prediction of complex super-secondary structure $\beta\alpha\beta$ motifs based on combined features. *Saudi J Biol Sci* 23:66–71
 99. Chambers P, Pringle CR, Easton AJ (1990) Heptad repeat sequences are located adjacent to hydrophobic regions in several types of virus fusion glycoproteins. *J Gen Virol* 71:3075–3080
 100. Edoh K, MacCarthy E (2018) Network and equation-based models in epidemiology. *Int J Biomath* 11:1850046
 101. Smith RK, Archibald A, MacCarthy E, Liu L, Luke NS (2016) A mathematical investigation of vaccination strategies to prevent a measles epidemic. *N C J Math Stat* 2:29–44
 102. Taylor WR, Thornton JM (1983) Prediction of super-secondary structure in proteins. *Nature* 301:540–542
 103. Sun L, Hu X, Li S, Jiang Z, Li K (2016) Prediction of complex super-secondary structure betaalphabeta motifs based on combined features. *Saudi J Biol Sci* 23:66–71
 104. Geertz-Hansen HM, Blom N, Feist AM, Brunak S, Petersen TN (2014) Cofactory: sequence-based prediction of cofactor specificity of Rossmann folds. *Proteins* 82:1819–1828
 105. Schaeffer RD, Liao Y, Cheng H, Grishin NV (2017) ECOD: new developments in the evolutionary classification of domains. *Nucleic Acids Res* 45:D296–D302
 106. Andrade MA, Perez-Iratxeta C, Ponting CP (2001) Protein repeats: structures, functions, and evolution. *J Struct Biol* 134:117–131
 107. Schaeffer RD, Kinch LN, Liao Y, Grishin NV (2016) Classification of proteins with shared motifs and internal repeats in the ECOD database. *Protein Sci* 25:1188–1203
 108. Chaudhuri I, Soding J, Lupas AN (2008) Evolution of the beta-propeller fold. *Proteins* 71:795–803
 109. Koehler Leman J, Ulmschneider MB, Gray JJ (2015) Computational modeling of membrane proteins. *Proteins* 83:1–24
 110. Venko K, Roy Choudhury A, Novic M (2017) Computational approaches for revealing the structure of membrane transporters: case study on bilitranslocase. *Comput Struct Biotechnol J* 15:232–242
 111. Waldspuhl J, Berger B, Clote P, Steyaert JM (2006) transFold: a web server for predicting the structure and residue contacts of transmembrane beta-barrels. *Nucleic Acids Res* 34:W189–W193
 112. Tran Vdu T, Chassagnet P, Sheikh S, Steyaert JM (2012) A graph-theoretic approach for classification and structure prediction of transmembrane beta-barrel proteins. *BMC Genomics* 13(Suppl 2):S5
 113. Savojardo C, Fariselli P, Casadio R (2013) BETAWARE: a machine-learning tool to detect and predict transmembrane beta-barrel proteins in prokaryotes. *Bioinformatics* 29:504–505
 114. Mackenzie CO, Zhou J, Grigoryan G (2016) Tertiary alphabet for the observable protein structural universe. *Proc Natl Acad Sci U S A* 113:E7438–E7447
 115. May P, Barthel S, Koch I (2004) PTGL—a web-based database application for protein topologies. *Bioinformatics* 20:3277–3279
 116. Koch I, Schafer T (2018) Protein supersecondary structure and quaternary structure topology: theoretical description and application. *Curr Opin Struct Biol* 50:134–143
 117. Shen Y, Delaglio F, Cornilescu G, Bax A (2009) TALOS+: a hybrid method for

- predicting protein backbone torsion angles from NMR chemical shifts. *J Biomol NMR* 44:213–223
118. Shen Y, Bax A (2013) Protein backbone and sidechain torsion angles predicted from NMR chemical shifts using artificial neural networks. *J Biomol NMR* 56:227–241
119. Hafsa NE, Wishart DS (2014) CSI 2.0: a significantly improved version of the Chemical Shift Index. *J Biomol NMR* 60:131–146
120. Wu S, Zhang Y (2010) Recognizing protein substructure similarity using segmental threading. *Structure* 18:858–867
121. Guzenko D, Strelkov SV (2017) CCFold: rapid and accurate prediction of coiled-coil structures and application to modelling intermediate filaments. *Bioinformatics* 34:215–222
122. Fernandez-Fuentes N, Dybas JM, Fiser A (2010) Structural characteristics of novel protein folds. *PLoS Comput Biol* 6:e1000750
123. Vallat B, Madrid-Aliste C, Fiser A (2015) Modularity of protein folds as a tool for template-free modeling of structures. *PLoS Comput Biol* 11:e1004419
124. Menon V, Vallat BK, Dybas JM, Fiser A (2013) Modeling proteins using a supersecondary structure library and NMR chemical shift information. *Structure* 21:891–899
125. Jones DT (2001) Predicting novel protein folds by using FRAGFOLD. *Proteins* 45 (Suppl 5):127–132
126. Jones DT (1997) Successful ab initio prediction of the tertiary structure of NK-lysin using multiple sequences and recognized supersecondary structural motifs. *Proteins* 1:185–191
127. Kosciolek T, Jones DT (2014) De novo structure prediction of globular proteins aided by sequence variation-derived contacts. *PLoS One* 9:e92197
128. Jones DT, Buchan DW, Cozzetto D, Pontil M (2012) PSICOV: precise structural contact prediction using sparse inverse covariance estimation on large multiple sequence alignments. *Bioinformatics* 28:184–190
129. Pellegrini-Calace M, Carotti A, Jones DT (2003) Folding in lipid membranes (FILM): a novel method for the prediction of small membrane protein 3D structures. *Proteins* 50:537–545
130. Nugent T, Jones DT (2012) Accurate de novo structure prediction of large transmembrane protein domains using fragment-assembly and correlated mutation analysis. *Proc Natl Acad Sci U S A* 109:E1540–E1547
131. Mier P, Alanis-Lobato G, Andrade-Navarro MA (2017) Protein-protein interactions can be predicted using coiled-coil co-evolution patterns. *J Theor Biol* 412:198–203
132. Pilla KB, Otting G, Huber T (2017) Protein structure determination by assembling supersecondary structure motifs using pseudocontact shifts. *Structure* 25:559–568
133. Fiser A, Šali A (2003) Modeller: generation and refinement of homology-based protein structure models. *Methods Enzymol* 374:461–491
134. Sánchez R, Šali A (1999) ModBase: a database of comparative protein structure models. *Bioinformatics* 15:1060–1061
135. Zhang Y (2008) I-TASSER server for protein 3D structure prediction. *BMC Bioinformatics* 9:40
136. Xu D, Zhang J, Roy A, Zhang Y (2011) Automated protein structure modeling in CASP9 by I-TASSER pipeline combined with QUARK-based ab initio folding and FG-MD-based structure refinement. *Proteins* 79:147–160
137. Armstrong CT, Vincent TL, Green PJ, Woolfson DN (2011) SCORER 2.0: an algorithm for distinguishing parallel dimeric and trimeric coiled-coil sequences. *Bioinformatics* 27:1908–1914
138. Vincent TL, Green PJ, Woolfson DN (2012) LOGICOIL—multi-state prediction of coiled-coil oligomeric state. *Bioinformatics* 29:69–76



Chapter 3

Automated Family-Wide Annotation of Secondary Structure Elements

Adam Midlik, Ivana Hutařová Vařeková, Jan Hutař, Taraka Ramji Moturu, Veronika Navrátilová, Jaroslav Koča, Karel Berka, and Radka Svobodová Vařeková

Abstract

Secondary structure elements (SSEs) are inherent parts of protein structures, and their arrangement is characteristic for each protein family. Therefore, annotation of SSEs can facilitate orientation in the vast number of homologous structures which is now available for many protein families. It also provides a way to identify and annotate the key regions, like active sites and channels, and subsequently answer the key research questions, such as understanding of molecular function and its variability.

This chapter introduces the concept of SSE annotation and describes the workflow for obtaining SSE annotation for the members of a selected protein family using program SecStrAnnotator.

Key words Annotation, Secondary structure, Secondary structure elements, Protein family, Protein domain, SecStrAnnotator, Structural alignment, Secondary structure assignment

1 Introduction

1.1 Background and Motivation

Protein structural data represent a highly valuable source of information, and important research results have been discovered based on them. All the data (currently ~140,000 entries) are accessible to the research community via Protein Data Bank [1], and the number of structures is continuously growing.

In the past, the newly determined structures differed from other available proteins, because only a few isolated islands in the chemical space of proteins were mapped. With the increasing number of known structures, protein families started to emerge, consisting of structurally and functionally similar proteins. Nowadays, more and more structures (which originate from various organisms, contain different ligands, or have various mutations) are being collected in each family. This trend is nicely demonstrated on five different protein families, mentioned in Table 1.

Table 1**Number of biomacromolecular structures in Protein Data Bank for selected protein families**

Protein family	CATH code	Number of PDB entries in years			
		1990	2000	2010	2018
Globins	1.10.490.10	40	319	797	1090
Cytochrome P450	1.10.630.10	2	59	376	728
NADP-dependent oxidoreductase	3.20.20.100	0	21	197	353
Apoptosis regulator Bcl-2	1.10.437.10	0	7	79	133
Bulb-type lectin	2.90.10.10	0	7	16	30

To characterize these families, several databases focused on classification of protein structures based on their similarity have been developed (CATH [2], SCOPe [3]). With the vast amount of structural data about each protein family, the systematic study of larger datasets, as opposed to the study of individual structures, is gaining importance. Based on these data, it is possible to reach interesting and important research results—from understanding biomacromolecular functions and mechanisms of their action to the classification of types of diseases or the rational development of novel drugs.

But such studies can hardly rely on bare structural data; an additional layer of information is necessary. This new layer is *annotation*—assigning a name or any potentially biologically relevant information to a structure or its part. The annotated part can range from a single atom or residue through a secondary structure element or a functionally important region to a whole protein.

This chapter will focus on the annotation of secondary structure elements (SSEs). Each protein family has a set of characteristic SSEs with a well-defined arrangement, which is consistent even if the proteins originate from different species or perform different functions. Hence, the SSEs can serve as landmarks which enable easier orientation in the protein structures.

Annotation of SSEs has a long tradition in some protein families. For example, the nomenclature of helices and sheets in cytochrome P450 (CYP) family is well established [4, 5] and proves to be particularly useful when comparing existing structures, describing new ones, or generalizing observations over the whole family (*see* Fig. 1a). Furthermore, SSEs can be used as a reference to describe the position of other key regions, such as catalytic sites, channels, or protein-protein interfaces. A nice illustrative example is again the CYP family, with a well-established classification of multiple different channels based on their position relative to the traditionally named SSEs [6] (*see* Fig. 1b).

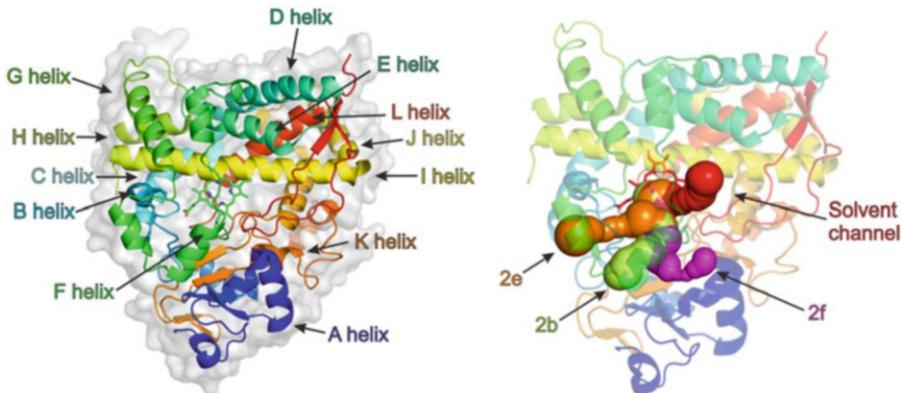


Fig. 1 Illustrative annotation of (a) secondary structure elements and (b) channels in a member of CYP family

Even in families that have no such traditional nomenclature, annotation of SSEs can still be valuable, because it provides the correspondence between the SSEs of individual members of the family—mutually corresponding SSEs are simply annotated by the same name, albeit arbitrarily created. This makes it possible to study the SSEs in the context of the family and describe the general anatomy of the family—the occurrence of the individual SSEs, their typical length, position, amino acid composition, and the variability of these properties and their relation to the function.

Visualization of protein structures can also benefit from SSE annotation. Currently available tools for the generation of 2D topology diagrams (e.g., HERA [7], PROMOTIF [8], Pro-origami [9]) treat each structure separately and do not take protein similarity into account. As a result, two structurally similar proteins (or even two structures of the same protein) can yield entirely dissimilar diagrams. SSE annotation can be used to modify the generation of topology diagrams in such way that the resulting diagrams would place conserved SSEs to similar positions within the whole family [10]. This highlights the parts of each structure which diverge from the general anatomy.

In this chapter, we describe methods for automated annotation of SSEs in protein structures. Our approach is template-based, meaning that a template annotation of one protein from the family is provided to the algorithm. The overall procedure therefore consists of three main stages: preparing the data for a selected protein family, preparing the template annotation, and running the annotation algorithm on each member of the family.

1.2 Terminology

To avoid later confusion, we provide a summary of the basic terms that will be used throughout the text.

Secondary structure element (SSE) is a contiguous region of a protein chain exhibiting some secondary structure pattern. SSEs can be coarsely divided into three types: *helix*, β -*strand* (or simply

strand), and *loop*. In this work, we focus only on the first two types. Further classification into different subtypes (α -helix, β_{10} -helix, etc.) is not necessary for the annotation purposes. SSE types are typically abbreviated as H (helix) and E (strand).

Each SSE within a structure can be identified by its position—*chain* identifier, *start* (index of its first residue), and *end* (index of its last residue)—and its *type*.

The situation gets more complicated in the case of β -strands because they are mutually connected via hydrogen bonds—we call this β -connectivity. The term β -ladder refers to a set of backbone-backbone hydrogen bonds between two particular β -strands. A single hydrogen bond is not considered a β -ladder. Each β -ladder can be classified as either *parallel* or *antiparallel*, based on the relative orientation of the two β -strands.

We define β -graph as an undirected edge-labelled graph whose vertices correspond to the β -strands in a structure and edges correspond to the β -ladders. The label of each edge denotes the type of the β -ladder (parallel or antiparallel).

The term β -sheet refers to a set of β -strands which are connected by β -ladders. Using the notion of β -graph, a β -sheet is defined as a connected component in β -graph. A β -sheet can contain β -strands from more than one chain.

Secondary structure assignment (SSA) consists of the set of SSEs found in a protein structure (each one described by its chain, start, end, and type) and optionally the β -graph. SSA can also refer to the process by which the SSEs and the β -graph are found.

Secondary structure annotation is assignment of names to some (or all) SSEs in a protein structure.

Protein family is a set of structurally similar *protein domains*. Each of these domains can be either a whole protein chain or only its part (in multidomain proteins).

2 Materials

2.1 Databases

1. **PDBe:** Protein Data Bank in Europe (PDBe) is one of the members of the Worldwide Protein Data Bank (wwPDB) [11] which maintains and provides access to the global repository of macromolecular structure models, the Protein Data Bank (PDB). Apart from the access to the structural data, PDBe provides a range of related services and tools. SIFTS (structure integration with function, taxonomy, and sequence [12]) provides cross-references to other biological databases, such as UniProt, CATH, or Pfam. PDBe REST API is a programmatic way to obtain information from the PDBe services.

<http://www.ebi.ac.uk/pdbe/>

2. **CATH:** The CATH database [2] provides structure-based hierarchical classification of protein domains found in the protein structures from PDB. CATH uses a four-level structural hierarchy, whose bottom level, homologous superfamily, corresponds to a demonstrable evolutionary relationship between domains.

<http://www.cathdb.info/>

3. **Pfam:** The Pfam database [13] classifies protein domains into families based on their sequence similarity. Each protein family is represented by a multiple sequence alignment and a hidden Markov model (HMM).

<https://pfam.xfam.org/>

2.2 Tools

1. **PyMOL:** PyMOL [14] is a commonly used molecular visualization tool. It is typically operated from graphical user interface (GUI), but it also supports interpretation of scripts from command line, without GUI. Besides other functionality, it provides commands for structural alignment and superimposition.

<https://pymol.org/>

2. **DSSP:** Define secondary structure of proteins (DSSP) [15] is a well-established algorithm for secondary structure assignment based on hydrogen bond patterns.

<https://swift.cmbi.umcn.nl/gv/dssp/index.html>

3 Methods

In this section, we will describe all steps which are necessary to obtain secondary structure annotations for a selected protein family. The annotation procedure contains three stages (*see* Fig. 2). First, we must obtain the list of domains that belong to the protein family and download their structures. The second step is the choice of the template domain and obtaining its annotation. Third, we run annotation algorithm on each domain in the family. The annotation algorithm is implemented in a program called *SecStrAnnotator* and itself consists of three steps: structural alignment, secondary structure assignment, and matching the template SSEs with the query SSEs. We will discuss each of these stages in more detail.

The individual steps of the procedure will be demonstrated on the cytochrome P450 family (CYPs). We will reference some scripts in the text, which can be used for easier automation of the workflow. All these scripts are written in programming language Python3 and are available on our website (<https://webchem.ncbr.muni.cz/Wiki/SecStrAnnotator>) together with SecStrAnnotator software.

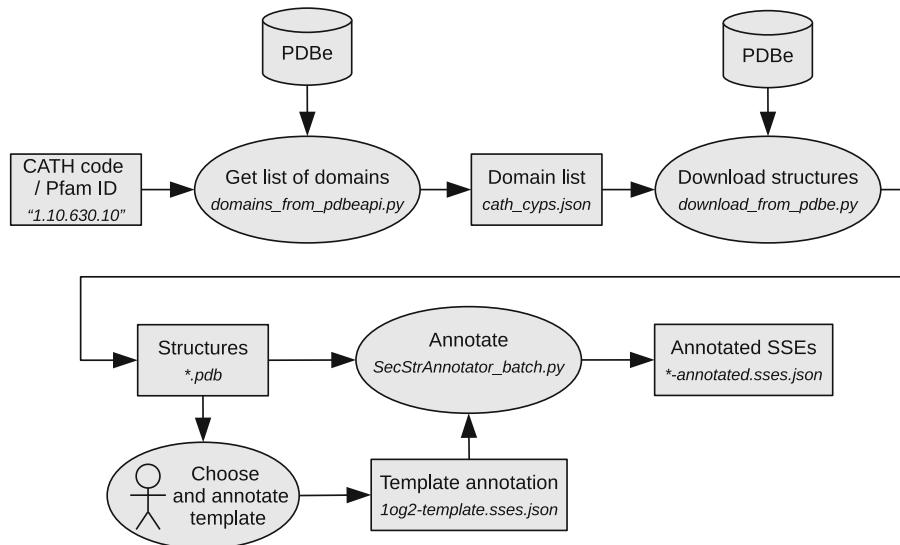


Fig. 2 The overall workflow of SSE annotation performed on a protein family. File names are illustrative (based on CYP family)

3.1 Obtaining the Structures

The list of members of the selected protein family can be acquired from several databases. We will mention two of them, namely, CATH [2] and Pfam [13].

To identify a particular domain, it is necessary to specify the PDB identifier of the structure (PDB ID), chain identifier within the structure, and residue range (or ranges) within the chain. Our notation is best demonstrated on examples:

- Domain 1tqnA00, as defined in CATH, is represented as (1tqn A 28:499), meaning that it is located in the structure with PDB ID 1tqn, in chain A, and it spans residues from 28 to 499.
- Similarly, 1h9rA01 is described as (1h9r A 123:182,255:261), meaning that it is in PDB structure 1h9r and in chain A and consists of two segments, containing residues 123–182 and 255–261.

We use a colon in the residue ranges to avoid confusion between a dash and a minus symbol. In case there are no residues on the chain before or after the domain, the residue numbers in the range can be omitted, e.g., (1tqn A 28:), (1tqn A :499), or (1tqn A :), where the last notation represents the whole chain.

The residue numbers and chain identifier conform to the numbering scheme used in PDB files (auth_* numbering scheme). This corresponds to fields `_atom_site.auth_seq_id` and `_atom_site.auth_asym_id` in mmCIF files, rather than `_atom_site.label_seq_id` and `_atom_site.label_asym_id` (label_* numbering scheme). It also corresponds to `author_residue_number` and `chain_id` in PDBe REST API.

It is important to be aware of which numbering scheme is used in each moment, because some software tools use `auth_*` while others use `label_*` (see Note 1).

3.1.1 List of Domains from CATH

Protein family corresponds to the term *homologous superfamily* used in CATH. At CATH website it is possible to find the selected homologous superfamily and get its CATH code (a four-part numeric identifier such as 1.10.630.10). Alternatively, a cross-reference from a particular structure in PDBe can be used.

The list of domains can be obtained programmatically using PDBe REST API, specifically *SIFTS mapping* call. An example of such API call is as follows:

```
GET: http://www.ebi.ac.uk/pdbe/api/mappings/1.10.630.10
```

The server response is in a convenient and easy-to-process JSON format. The response can be shown directly in a web browser on the PDBe REST API documentation page (<https://www.ebi.ac.uk/pdbe/api/doc/sifts.html>).

A script can be used to call the API, extract all needed information from the response, and write it out into a simplified JSON file (see Note 2). Here is an example of calling the script from command line:

```
python3 domains_from_pdbeapi.py 1.10.630.10 > cath_cyps.json
```

The output of the script is a list of domains for each PDB ID in JSON format. Each domain is represented by a three-element array containing domain name, chain identifier, and residue range. The following example of an output contains two domains in PDB entry 1bu7 and one domain in 1tqn:

```
{
  "1bu7": [[{"1bu7A00", "A", "1:455"}, {"1bu7B00", "B", "1:455"}]],
  "1tqn": [{"1tqnA00", "A", "28:499"}]
}
```

According to CATH convention, domain name 1tqnA00 consists of the PDB identifier 1tqn, chain identifier A, and the number of the domain within the chain (00 is typically used when there is only one domain in the chain; otherwise the domains are numbered 01, 02, etc.). Domain 1tqnA00 would be described as (1tqn A 28:499) in our notation.

3.1.2 List of Domains from Pfam

The website of Pfam database provides a search tool for finding the family of interest. Alternatively, it can be navigated to by a cross-reference from a particular structure in PDBe. Once the page of the

family is found, its Pfam ID (a string such as “p450” or “Piwi”) and Pfam accession (such as PF00067 or PF02171) can be obtained.

To obtain the list of domains, PDBe REST API can be used again. The API call is constructed in the same way as with CATH code:

GET: <http://www.ebi.ac.uk/pdbe/api/mappings/PF00067>

However, the structure of the response is slightly different than in the case of CATH code; among other things, the domain names are not present. Our script (*see Note 3*) supports both types of response and constructs the missing domain names in a CATH-like manner. The script can be used with Pfam accession code in the same way as with CATH code:

```
python3 domains_from_pdbeapi.py PF00067 > pfam_cyps.json
```

3.1.3 Structures from PDBe

There are many online servers providing PDB structural data. We use PDBe. All needed structures can be downloaded at once, in PDB file format, using script `download_from_pdbe.py`. An example of calling the script:

```
python3 download_from_pdbe.py cath_cyps.json structure_directory
```

3.2 Choice and Annotation of the Template

Our approach to SSE annotation is template-based, meaning that an annotated template domain must be provided to the algorithm. The algorithm then tries to find the annotation of the query domains which well reflects the template annotation. Thus, two tasks are crucial in order to obtain useful annotations for a protein family: selection of the template domain from all domains in the family and preparing the annotation file for this template domain. Our current approach does not include any automated method for fulfilling these tasks; therefore, they must be performed manually. The situation strongly depends on whether an annotation for the selected template domain is available (from literature or other sources). If so, it can be used as the template annotation (possibly with some refinements, described in Subheading 3.2.2). If there is no such annotation, then it must be created from scratch (described in Subheading 3.2.3).

3.2.1 Choice of the Template Domain

There are several requirements for the template domain. An important requirement is availability of SSE annotation in literature. In case of CYPs, the SSE nomenclature is well established, and we based our template annotation on the structure of human CYP 2C9 (PDB ID 1og2) as described by Rowland [5] (*see* Subheading 3.2.2). Unfortunately, sometimes there is no annotation available

in literature. In that case it must be obtained by other procedures, described in *see* Subheading 3.2.3.

The template domain should be a representative of the whole family, so it should be a “typical” or “average” structure rather than an unusual structure which diverges greatly from the rest of the family. If possible, it should contain all the SSEs that are characteristic for the family.

When there are several candidates for the template domain, the quality of the structures should be taken into account—resolution, R values, coverage (i.e., what fraction of residues of the domain are included in the model), and other quality metrics provided by wwPDB structure validation report [16].

In case that a candidate structure contains ligands or other protein chains, it should be checked that these do not induce nonstandard conformation of our domain of interest, as this could have a negative effect on the outcome of the annotation procedure.

An appropriate strategy is also to try out multiple alternatives for the template domain and select the one which performs best. In cases of families with high structural diversity, it might be necessary to divide the family into a few more uniform subgroups and use a separate template for each of them. CATH S35 sequence clusters can serve as a guide to this division.

3.2.2 Refinement of Existing Template Annotation

If some annotation is available, it can be used as it stands (after converting into the required format, described in Subheading 3.2.4). However, it may be appropriate to apply some modifications to this annotation. We will demonstrate these modifications on the annotation of CYP 2C9 (PDB ID 1og2, domain 1og2A00). Individual modification steps are shown in Table 2. For the sake of simplicity, only a few illustrative SSEs are shown (the complete annotation contains more than 30 SSEs).

1. The starting point is the annotation obtained from literature or another source. In case of 1og2, we obtained it from ref. 5. This annotation is shown in column *Original* of Table 2.
2. The best results will be obtained if the secondary structure assignment (SSA; *see* Subheading 1.2) of the template and query domains are obtained by the same method. Therefore, we advise running SSA algorithm on the template domain and making sure that the template annotation is consistent with it. SSA can be run easily by SecStrAnnotator with option --onlyssa. Furthermore, the resulting SSA file will be in file format described in Subheading 3.2.4, so it is easier to add SSE names into this file than creating the file manually.

In case of 1og2, we shifted boundaries of helices B, C, and J' by a few residues to make them consistent with the SSA

Table 2
The process of refinement of the template annotation for 1og2

Label	Original →	Consistent SSA →	Additional SSEs →	Artificial SSEs
A	50–61	50–61	50–61	50–61
B	80–89	80–90	80–90	80–90
B''			91–94	91–94
C	117–131	118–131	118–131	118–131
J'	339–342	339–345	339–345	339–345
β2.1			374–374	374–374
β2.2			381–381	381–381
β4.1	472–473	472–473	472–473	472–473
β4.2	478–479	478–479	478–479	478–479
β4.3				462–462

SSEs added or changed in each step are shown in bold

algorithm used in SecStrAnnotator. The result is shown in column *Consistent SSA* of Table 2.

3. Sometimes it can be discovered that some SSEs are frequently present in members of a protein family but are not included in the annotation that was obtained from literature. This can be a reason to add these SSEs to the template annotation.

This is illustrated by a short helix found between helices B and C of 1og2. We found out that it is present in more than 80% members of the CYP family and gave it a name B'' (B' is already used). Sheet β2, consisting of strands β2.1 and β2.2, belongs to SSEs with traditionally established names but was missing in the annotation from ref. 5. Column *Additional SSEs* in Table 2 shows the template annotation after adding B'' and sheet β2.

4. Some SSEs are typical for a protein family (thus worth being annotated) but do not occur in all its members. It is not always possible to find a template domain which would contain all SSEs that we want to annotate in the family. In such case we are forced to use a little trick and add the missing SSEs to the template annotation artificially (even though it is in contradiction with point 2).

As an example, in some CYPs, sheet β4 consists of three strands (β4.1, β4.2, β4.3), while in others, including 1og2, it is formed only by two strands (β4.1, β4.2). We artificially added β4.3 to the template annotation, in order to allow it to be annotated in those CYPs where it is really present. We

determined approximate position of this artificial β 4.3 based on a few CYPs where β 4.3 is present. The annotation after this step is shown in column *Artificial SSEs* of Table 2.

3.2.3 Creating Template Annotation from Scratch

If there is no available annotation for any member of the family and no naming convention for the SSEs in the family, the template annotation must be created from scratch. This is a nontrivial task, and we have not yet developed any rigorous algorithm to fulfil it. Therefore, we will only describe an intuitive manual method:

1. Run SecStrAnnotator on the template domain with option `--onlyssa`. This will produce secondary structure assignment, which can be used as a template annotation. The individual SSEs will be labelled sequentially and prefixed by the SSE type (e.g., H0, H1, E2, E3, etc.).
2. Try to annotate the family (or a sample of it) using the template annotation from 1 and inspect the results. If there is some unannotated SSE frequently occurring between two particular annotated SSEs, add it to the template annotation (it will be an artificial SSE).
3. If some SSE from the template annotation occurs very rarely in the family, remove it from the template annotation.

Steps 2 and 3 can be performed repeatedly until a satisfactory template annotation is obtained (*see Note 4*). In some cases, it might turn out that the selected template domain is not appropriate, and another domain will serve as a better template.

The procedure can be illustrated on an example of GPCR family (CATH code 1.20.1070.10), shown in Table 3. We randomly selected domain 3pdsA01 as the template domain. SSA yielded eight helices, automatically labelled H0 to H7 (column *Original* in Table 3). After annotation of some other members of the family, we found a two-strand β -sheet occurring between helices H1 and H2 in around 50% of the structures, so we added two artificial strands in the corresponding position (column *Artificial SSEs*). On the other hand, helix H4 was found in less than 20% of the structures, so we removed it from the template annotation (column *Removed SSEs*). Just for transparency, we assigned labels A to G to the remaining seven helices and β 1.1, β 1.2 to the strands (column *Annotation*).

3.2.4 SecStrAnnotator Annotation Format

The remaining task is to convert the template annotation to the format required by SecStrAnnotator. The SecStrAnnotator annotation format is a JSON file, and its structure is illustrated in Fig. 3 (*see Note 5*).

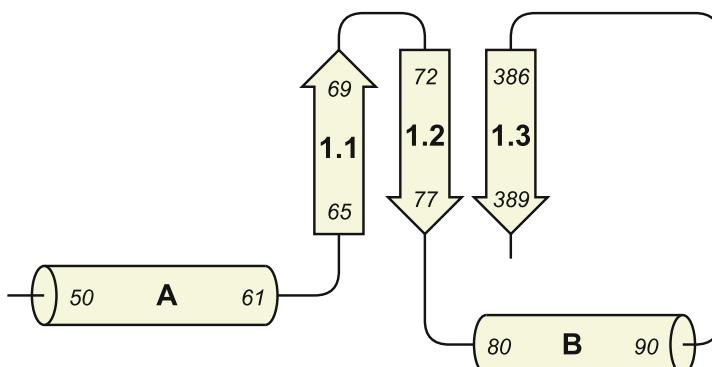
The annotation file contains an object with key-value pairs corresponding to PDB IDs (keys) and annotation data objects (values). Typically, there is only one key-value pair (i.e., annotation

Table 3
The process of creating the template annotation for domain 3pdsA01

Label	Original →	Artificial SSEs →	Removed SSEs →	Annotation
H0	30–61	30–61	30–61	A
H1	67–96 97–97 101–101	67–96 97–97 101–101	67–96 97–97 101–101	B β1.1 β1.2
H2	103–136	103–136	103–136	C
H3	147–171	147–171	147–171	D
H4	179–187	179–187		
H5	197–229	197–229	197–229	E
H6	267–299	267–299	267–299	F
H7	305–329	305–329	305–329	G

SSEs added or changed in each step are shown in bold

a



b

```
{
  "log2": {
    "comment": "Example template for demonstration of annotation format.",
    "secondary_structure_elements": [
      { "label": "A", "chain_id": "A", "start": 50, "end": 61, "type": "H" },
      { "label": "1.1", "chain_id": "A", "start": 65, "end": 69, "type": "E" },
      { "label": "1.2", "chain_id": "A", "start": 72, "end": 77, "type": "E" },
      { "label": "B", "chain_id": "A", "start": 80, "end": 90, "type": "H" },
      { "label": "1.3", "chain_id": "A", "start": 386, "end": 389, "type": "E" }
    ],
    "beta_connectivity": [
      [ "1.1", "1.2", -1 ],
      [ "1.2", "1.3", 1 ]
    ]
  }
}
```

Fig. 3 (a) Topology diagram of an example domain. **(b)** Annotation of the example domain in SecStrAnnotator format. The domain contains two helices, named A and B, and a β-sheet consisting of three strands, named 1.1, 1.2, and 1.3. Strands 1.1 and 1.2 are connected by an antiparallel β-ladder, strands 1.2 and 1.3 by a parallel β-ladder. All the SSEs are located on chain A of structure 1og2

of one structure) in one file. The annotation data object contains keys `secondary_structure_elements` and `beta_connectivity`. Additional information, such as `comment`, can be included but will be ignored by `SecStrAnnotator`.

The value of `secondary_structure_elements` must be an array of objects, each object describing a single SSE. Each of these objects should contain the following:

- `label`—name of the SSE, unique within the domain,
- `chain_id`—chain identifier,
- `start`, `end`—residue number of the first and the last residue in the SSE,
- `type`—type of the SSE. The value can be “H” or “h” for a helix and “E” or “e” for a β -strand. More detailed distinction can be made using DSSP convention [15], i.e., “G” for β_{10} -helix, “H” for α -helix, “I” for π -helix, “B” for β -strand (with one residue), and “E” for β -strand (with at least two residues); nevertheless, `SecStrAnnotator` will consider them as equivalent to “H” or “E”.

The necessity of the `beta_connectivity` section in the template annotation depends on the choice of matching algorithm used in `SecStrAnnotator`. The default algorithm (MOM) takes connectivity of β -strands into account; therefore, the section is required. When using the alternative algorithm (DP), which ignores β -connectivity, this section can be omitted.

The value of `beta_connectivity` contains an array of connection items. Each connection item itself is an array containing two strings and one number and bears information about two β -strands connected by a β -ladder. The two strings are labels of the two connected strands, and the number describes the relative orientation of the strands (1 for parallel, -1 for antiparallel).

3.3 Running `SecStrAnnotator`

In the previous steps, we described how to obtain a list of domains belonging to a protein family, their structures, and an annotation of one of these domains (template domain). Now the annotation algorithm, implemented in `SecStrAnnotator`, can be executed on each domain.

`SecStrAnnotator` finds annotation for a query domain Q based on the template domain T . Thus, the input consists of the structure of T , structure of Q , and annotation of T . The algorithm consists of three steps. First, it will perform structural alignment of T and Q so that their corresponding SSEs are located close to each other. Then, it will run secondary structure assignment (SSA) on domain Q . Finally, it will match the template SSEs to the query SSEs, and for each annotated SSE in T , it will select the corresponding SSE in Q .

`SecStrAnnotator` is implemented in C# programming language. It can be downloaded from our website (<https://webchem.ncbr.muni.cz/Wiki/SecStrAnnotator>) together with

`SecStrAnnotator_batch.py`, which is a wrapper for running `SecStrAnnotator` on a batch of domains and collecting the results into one annotation file (*see Notes 6 and 7*).

On Windows, `SecStrAnnotator` is executed from command line using the following syntax:

```
SecStrAnnotator.exe [OPTIONS] DIRECTORY TEMPLATE QUERY
```

On Linux, it can be executed using Mono (requires installing `mono-devel` package):

```
mono SecStrAnnotator.exe [OPTIONS] DIRECTORY TEMPLATE QUERY
```

The argument `DIRECTORY` is the directory containing all the input files. The output files will also be saved to this directory. The remaining arguments `TEMPLATE` and `QUERY` describe the domains T and Q . Acceptable formats for these arguments are `PDB` (e.g., `1tqn`) or `PDB,CHAIN` (e.g., `1tqn,A`) or `PDB,CHAIN,RANGES` (e.g., `1tqn,A,:`). For example, the domain in ranges `123:183, 252:261` on chain `B` in `1h9r` will be described as `1h9r, B, 123:183,252:261`.

The following input files must exist:

- `DIRECTORY/TEMPLATEPDB.pdb` (structure of T in PDB format).
- `DIRECTORY/QUERYPDB.pdb` (structure of Q in PDB format).
- `DIRECTORY/TEMPLATEPDB-template.sses.json` (annotation of T in format described in Subheading 3.2.4).

The output files will be:

- `DIRECTORY/QUERYPDB-detected.sses.json` (SSA of Q).
- `DIRECTORY/QUERYPDB-annotated.sses.json` (annotation of Q).

`SecStrAnnotator` has dependencies on other programs (PyMOL, optionally DSSP) and scripts (`script_align.py`, `script_session.py`). These auxiliary files need to be available in the system, and their location must be specified in the configuration file `SecStrAnnotator_config.json`. The configuration file itself must be in the same directory as `SecStrAnnotator.exe`. Modification of the configuration file might be necessary for successful execution (mainly setting the location of PyMOL on Windows).

Options

The following is an enumeration of the most important command line options. Default values (printed in bold) have been selected to be the most appropriate and robust.

- **--help**
Prints help message and returns.
- **--align METHOD**
Specifies structural alignment method, METHOD is one of align, super, **cealign**, none (more in Subheading 3.3.1).
- **--ssa METHOD**
Specifies secondary structure assignment method, METHOD is one of file, dssp, hbond, geom-dssp, **geom-hbond** (more in Subheading 3.3.2).
- **--onlyssa**
Changes the behavior of SecStrAnnotator so that it only runs SSA (no alignment and matching step). In this case it is executed: SecStrAnnotator.exe [OPTIONS] DIRECTORY QUERY (i.e., TEMPLATE argument is skipped).
- **--limit LIMIT**
Specifies the value of parameter r_0 (in angstroms) in geometrical SSA method, default 1.0 (more in Subheading 3.3.2).
- **--matching METHOD**
Specifies matching method, METHOD is one of dp, mom (more in Subheading 3.3.3).
- **--soft**
Switches on the soft matching variant in MOM algorithm (MOM-soft, more in Subheading 3.3.3).
- **--session**
Creates a PyMOL session visualizing the resulting annotation (see Note 8).

3.3.1 Structural Alignment

Structural alignment is realized by calling PyMOL. Option --align is used to select which PyMOL's command will be used for alignment:

- *align* (fastest but sequence dependent),
- *super* (slower, sequence independent),
- *cealign* (slowest but very robust, sequence independent CE algorithm [17]).

The default method is *cealign*, and it is preferred unless some performance issues are encountered.

3.3.2 Secondary Structure Assignment (SSA)

SecStrAnnotator allows several methods of SSA to be used (see Note 9). They can be selected by option --ssa. The default and recommended method is *geom-hbond*. However, other methods can be used:

- *file*: The SSA is simply loaded from file DIRECTORY/QUERYPDB .sses.json in format described in Subheading 3.2.4.

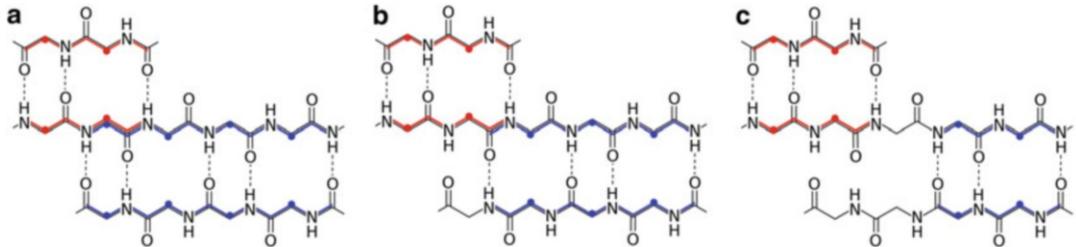


Fig. 4 Marginal situations for relative position of two antiparallel β -ladders: (a) the two ladders share four backbone atoms (one residue) and thus constitute single three-strand β -sheet; (b) the ladders share two backbone atoms (no residue) and are treated differently by DSSP (as two separate sheets) and our algorithm (as one sheet); (c) the ladders share no backbone atoms (no residue) and thus constitute two independent β -sheets. Backbone atoms belonging to each ladder are indicated by red and blue lines. Residues belonging to each ladder are indicated by a red or blue bead on their C^α atom

- *dssp*: DSSP program is executed on the query structure.
- *hbond*: Our modified built-in implementation of DSSP algorithm [15]. The most notable modification is in distinguishing between two β -ladders sharing one strand (thus constituting one sheet) and two independent β -ladders (in two different sheets). The difference between DSSP and our approach can be explained as follows: a backbone atom belongs to a ladder if it lies on a cycle formed by covalent bonds and hydrogen bonds of the ladder. A residue belongs to a ladder if its C^α atom belongs to the ladder (the original paper [15] uses different but equivalent formulation). DSSP considers two ladders to share a strand if they share at least one residue. We consider two ladders to share a strand if they share at least one backbone atom—this is more natural and also consistent with some other software, such as HERA [7] and PROMOTIF [8]. The marginal situations are shown in Fig. 4.
- *geom-hbond*: This is a combined method— β -strands are assigned by *hbond*, while helices are assigned using a geometrical method similar to ref. 18, described in the following text. This method is based purely on the geometry of protein backbone. 3×4 matrix \mathbf{Q}_i contains the coordinates of C^α atoms of residues i , $i + 1$, $i + 2$, and $i + 3$ in a chain. \mathbf{H} denotes “ideal” coordinates of four consecutive C^α in α -helix. This “ideal” coordinates were obtained from α -helices in experimental protein structures. $RMSD_i^H$ is defined as the RMSD between \mathbf{Q}_i and \mathbf{H} (after superimposition). If two or more consecutive values $RMSD_j^H \dots RMSD_k^H$ are below the threshold r_0 , then residues $j + 1$ to $k + 2$ are assigned as a helix. The parameter r_0 can be adjusted using option `--limit`. Lower values of r_0 will lead to stricter assignment with shorter and more regular helices, whereas higher values will tend to assign longer helices which may be curved and contain irregularities (kinks). Its default value 1.0 Å

is quite tolerant to irregularities (compared to DSSP), which is suitable for the purposes of annotation (kinked helices are usually annotated as one helix rather than being divided into two shorter helices). The algorithm does not distinguish between different types of helices (α , β_{10} , π).

- *geom-dssp*: Another combined method—uses DSSP for β -strands and the geometrical method for helices.

In order to allow easy comparison between SSEs, they are simplified to *line segments*, i.e., start-point and end-point of each SSE is calculated. This is done in by an algorithm related to our geometrical SSA method. Besides the ideal helix geometry \mathbf{H} , it uses $\mathbf{a}_\mathbf{H}$, a unit-length (column) vector with the direction of the axis of the ideal helix \mathbf{H} . The axis vector of a real helix spanning residues j to k is then calculated as

$$\mathbf{a} = \sum_{j \leq i \leq k-3} \mathbf{R}_i \mathbf{a}_\mathbf{H} \quad (1)$$

where \mathbf{R}_i is the rotation matrix of superimposition of \mathbf{H} onto \mathbf{Q}_i . Center of the real helix is calculated as

$$\mathbf{c} = \frac{1}{k-j+1} \sum_{j \leq i \leq k} \mathbf{r}_i \quad (2)$$

where \mathbf{r}_i is the position of C^α atom of residue i . The axis of the helix is the straight line p which passes through \mathbf{c} and has direction \mathbf{a} . The start-point \mathbf{u} and end-point \mathbf{v} of the helix are then calculated as the projection of its first and last C^α atom onto p :

$$\mathbf{u} = \mathbf{c} + \frac{(\mathbf{r}_j - \mathbf{c}) \cdot \mathbf{a}}{\mathbf{a} \cdot \mathbf{a}} \mathbf{a} \quad (3)$$

$$\mathbf{v} = \mathbf{c} + \frac{(\mathbf{r}_k - \mathbf{c}) \cdot \mathbf{a}}{\mathbf{a} \cdot \mathbf{a}} \mathbf{a} \quad (4)$$

The calculation of the line segment is illustrated in Fig. 5. Line segments for β -strands are calculated in the same manner, except \mathbf{E} and $\mathbf{a}_\mathbf{E}$ (C^α coordinates and axis vector of an ideal strand) are used

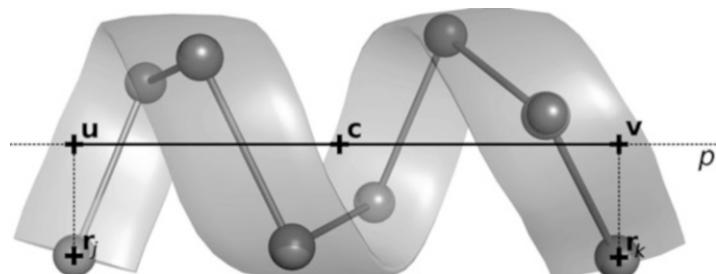


Fig. 5 Calculation of line segment \mathbf{uv} for a helix

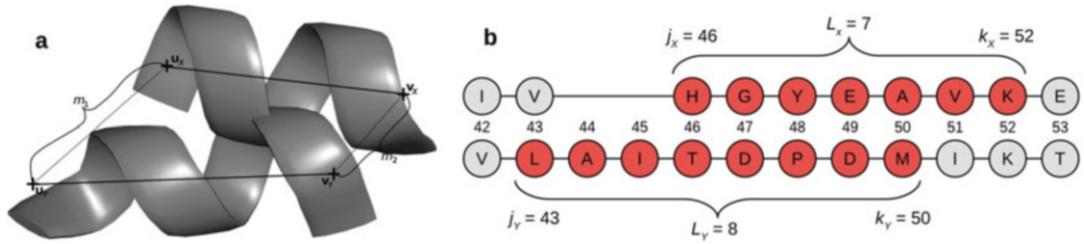


Fig. 6 Calculation of metric μ between two helices, X and Y . (a) Spatial part of μ : the first term in Eq. (5) is calculated as $0.5 \times (m_1 + m_2) = 0.5 \times (5.5 + 3.3) = 4.4$. (b) Structural alignment-based part of μ : the second term is calculated as $0.5 \times (|j_X - j_Y| + |k_X - k_Y|) = 2.5$ and the third term as $10 \times |L_X - L_Y| / \sqrt{(7 \times 8 + 9^2)} = 0.85$

instead of \mathbf{H} and $\mathbf{a_H}$. This calculation is used regardless of which methods was used for SSA.

3.3.3 SSE Matching

This is the core part of algorithm. The goal is to find the optimal matching between the SSEs of the domains T and Q . Subsequently, each matched SSEs in Q can be annotated by the same label as the SSE in T it was matched to.

Before the optimal matching can be found, it is necessary to have a measure of similarity between two SSEs, X and Y , belonging to T and Q , respectively. For this purpose, we define metric μ :

$$\begin{aligned} \mu(X, Y) = & c_1 (\|\mathbf{u}_X - \mathbf{u}_Y\| + \|\mathbf{v}_X - \mathbf{v}_Y\|) + c_2 (|j_X - j_Y| + |k_X - k_Y|) \\ & + c_3 |L_X - L_Y| / \sqrt{L_X L_Y + c_4^2} \end{aligned} \quad (5)$$

where $\mathbf{u}_X \mathbf{v}_X$ ($\mathbf{u}_Y \mathbf{v}_Y$) is the line segment for X (Y), j_X , k_X (j_Y , k_Y) are the positions of the first and last residue of X (Y) in the structural alignment of T and Q , and L_X (L_Y) is the number of residues in X (Y). The values of parameters c_1 to c_4 have been optimized to $c_1 = 0.5$, $c_2 = 0.5$, $c_3 = 10$, and $c_4 = 9$. Higher values of $\mu(X, Y)$ mean bigger difference between X and Y . Calculation of metric μ is illustrated in Fig. 6.

SecStrAnnotator allows choice from two different *matching algorithms*. DP algorithm is fast but ignores β -connectivity. MOM algorithm includes β -connectivity but might be slower under some circumstances (see Note 10). The default algorithm is MOM.

DP algorithm (standing for *dynamic programming*) ignores the β -connectivity of the structures and therefore can take advantage of dynamic programming technique [19]. The algorithm is very similar to the well-known Needleman-Wunsch algorithm used for sequence alignment in bioinformatics [20]. Let's denote $X = (X_i)_{1 \leq i \leq m}$, the sequence of template SSEs, i.e., the annotated

SSEs in the template domain T , in the same order as they appear in the primary structure. Similarly, $\Upsilon = (\Upsilon_i)_{1 \leq i \leq n}$ is the sequence of query SSEs, i.e., all SSEs found in the query domain Q .

The score S for matching SSE X_i with SSE $\Upsilon_{i'}$ is defined:

$$\begin{aligned} S(i, i') &= K - \mu(X_i, \Upsilon_{i'}) && \text{for } X_i, \Upsilon_{i'} \text{ of the same SSE type} \\ S(i, i') &= 0 && \text{for } X_i, \Upsilon_{i'} \text{ of different SSE types} \end{aligned}$$

where SSE type refers to two-class distinction (helix vs. strand). The default value of parameter K is set to 30 (see **Note 11**). Higher values of S indicate more similar SSEs.

The goal of the algorithm is to find a matching $M \subseteq \{1 \dots m\} \times \{1 \dots n\}$ which:

- (a) Preserves the order of SSEs:
 $i < j \Leftrightarrow i' < j'$ for each $(i, i') \in M, (j, j') \in M$
- (b) Matches only SSEs with positive score:
 $S(i, i') > 0$ for each $(i, i') \in M$
- (c) Maximizes the total score:
 $S_{\text{total}} = \sum_{(i, i') \in M} S(i, i').$

The optimal matching M is then found using the dynamic programming technique. The computational complexity is $O(mn)$.

MOM algorithm (standing for *mixed ordered matching*) takes the β -connectivity into account. Unlike DP algorithm, matching helix-to-helix and strand-to-strand, MOM matches helix-to-helix and ladder-to-ladder. Therefore, it requires slightly different formulation of the problem than DP algorithm. Besides X and Υ , we will define the set of template helices $H_X = \{i | X_i \text{ is a helix}\}$, the set of template strands $E_X = \{p | X_p \text{ is a strand}\}$, and the set of template ladders $L_X = \{pq | p, q \in E_X \wedge p < q \wedge X_p \text{ forms a ladder with } X_q\}$. Note that ladders are formally expressed as tuples, e.g., (p, q) , but for better readability, we use shortened notation pq . Sets H_Υ , E_Υ , and L_Υ are defined analogously for the query domain. In the following text, we will keep using indices i and j exclusively for helices and p, q, u , and v for strands; prime symbol will be used with query SSEs.

The goal is to find the optimal matching $M \subseteq H_X \times H_\Upsilon \cup L_X \times L_\Upsilon$, thence the word *mixed* in the name of the algorithm. There are the same three requirements for the matching M as in the case of DP algorithm; however, it is more complicated to express them formally. The matching M must:

- (a) Preserve the order of SSEs:

$$\begin{aligned} o(i, j, i', j') &\text{ for each } (i, i') \in M, (j, j') \in M \\ o(i, p, i', p') \wedge o(i, q, i', q') &\text{ for each } (i, i') \in M, (pq, p'q') \in M \end{aligned}$$

$$o(p, u, p', u') \wedge o(p, v, p', v') \wedge o(q, u, q', u') \wedge o(q, v, q', v') \text{ for each } \\ (pq, p'q') \in M, (uv, u'v') \in M$$

where o is the order consistency predicate, defined as $o(i, j, i', j')$:
 $i < j \Leftrightarrow i' < j'$ (in human words: i with j goes in the same order as i' with j').

(b) Match only SSEs with positive score:

$$S(i, i') > 0 \quad \text{for each } (i, i') \in M \\ S(p, p') > 0 \wedge S(q, q') > 0 \quad \text{for each } (pq, p'q') \in M$$

(c) Maximize the total score:

$$S_{\text{total}} = \sum_{(i, i') \in M} S(i, i') + \sum_{(pq, p'q') \in M} [S(p, p') + S(q, q')]$$

The problem of finding the matching M can now be easily reduced to the problem of finding a maximum-weight clique in a weighted graph (a clique is a subset of vertices all adjacent to each other). The vertices of the graph are all helix-to-helix and ladder-to-ladder matches with positive score (i.e., fulfilling criterion (b)). The edges of the graph connect only those pairs of vertices which preserve the order of SSEs (i.e., fulfilling criterion (a)). The weight of each vertex is simply the score S , and a clique with maximum total weight is to be found (i.e., fulfilling criterion (c)).

The maximum-weight clique problem can be solved by a backtracking algorithm, which systematically enumerates all inclusion-maximal cliques and selects the one with the best total weight (known as Bron-Kerbosch algorithm [21]). The algorithm can be further improved by using branch-and-bound technique—this means that the current maximum (the best weight found so far) is remembered, and any branch of the algorithm whose maximum expected weight is lower than the current maximum is evaluated as non-perspective and thus is ignored. This modification significantly improved the running time of the algorithm in most tested cases, yet a good worst-case computational complexity cannot be guaranteed. The algorithm will always find an optimal solution. In theoretical case of two cliques having exactly the same weight, the algorithm will find only one of them; nevertheless, in practice this is extremely unlikely. The theoretical computational complexity of MOM algorithm is exponential; however, in most cases the running times are very close to those of DP.

MOM-soft is a slight modification of the MOM algorithm, which allows two ladders which share a strand (see Fig. 4a) to be matched to two ladders whose strands are close to each other (see Fig. 4c)—this is a kind of variation that often occurs in protein

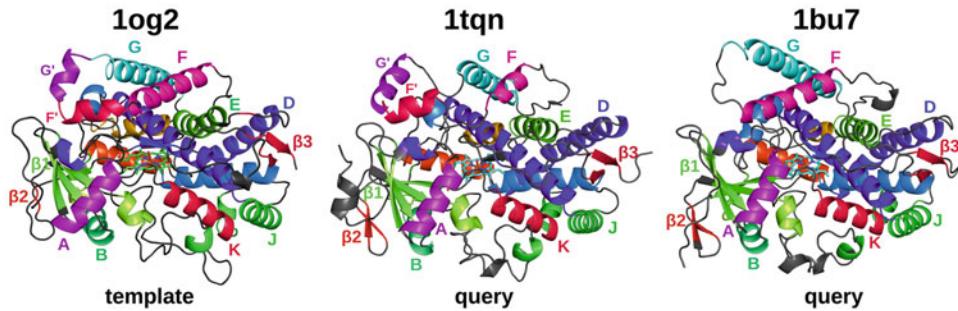


Fig. 7 Example of annotation of two query cytochrome P450 protein domains annotated with 1og2 used as template. For transparency, only visible SSEs are labelled

families, so it may be desirable to allow such matching. The only difference from MOM is that the order consistency predicate is defined as

$$\begin{aligned} o(i, j, i', j') : & (i < j \wedge i' < j') \vee (i > j \wedge i' > j') \vee (i = j \wedge |i' - j'| \leq 1) \\ & \vee (|i - j| \leq 1 \wedge i' = j') \end{aligned}$$

(see Note 12). MOM-soft is switched on by option `--soft`.

The final step, after running MOM or DP matching algorithm, is the transfer of SSE labels from the template to the query domain. In case of MOM, this means that for each pair of matched helices $X_i, Y_{i'}^*$ (i.e., for each $(i, i') \in M$), helix $Y_{i'}$ is assigned the same label as X_i has. Similarly, for each pair of matched ladders $X_p X_q, Y_{p'} Y_{q'}^*$ (i.e., for each $(pq, p'q') \in M$), strand $Y_{p'}$ gets the same label as X_p , and strand $Y_{q'}$ gets the same label as X_q . In case of DP, the situation is more straightforward: both the helices and the strands are annotated in the same way as the helices in MOM. Finally, all template SSEs which have been annotated are written into the output file with their newly assigned labels (see Note 13).

The results of the annotation algorithm are illustrated on two protein domains in Fig. 7.

4 Notes

1. The `auth_*` numbering scheme allows use of insertion codes, so in some situations, the residue numbering is not straightforward, e.g., 85, 86, 86A, 86B, 87, 88, etc. This complicates the situation even more. The current version of SecStrAnnotator does not support insertion codes and will raise an error if it encounters any. An ugly fix to this is running SecStrAnnotator with `--ignoreinsertions`; however, the structure will then not correspond to the real structure because all residues with insertion codes will be ignored. A better solution is converting an mmCIF file to a PDB file in such way that the `label_*`

numbering is used instead of `auth_*`. This can be done with help of PyMOL, using script `cif_to_pdb_with_label_numbering.py`. Then it is necessary that the domain residue ranges be in the `label_*` numbering as well (can be obtained by running `domains_from_pdbeapi.py` with `--numbering label`).

2. An alternative way of obtaining the list of domains is to download CATH classification file and filter the domains which belong to the selected homologous superfamily. Nevertheless, this has the disadvantage of including obsolete PDB entries in the list (whereas PDBe API excludes obsolete structures).
3. The Pfam database also provides its own API, which might be used as an alternative to PDBe API (with output in XML or tab-delimited format).
4. In the described procedure, the words “frequently” and “rarely” are very subjective and may also depend on the purpose for which the annotation is performed. In some situations, it may be desirable to have a rich template annotation with some SSEs occurring only in a small fraction of the family members. In other cases, it will be suitable to include only the most frequently occurring SSEs in the template annotation even if many rarer SSEs will then stay unannotated.
5. JSON files are sometimes not nicely formatted (without new lines and indentation) and are hard to read. Some web browsers (e.g., Firefox) can visualize such files in a human-friendly interactive form (although installation of extensions may be necessary).
6. SecStrAnnotator cannot guarantee 100% correctness of the provided annotations. Due to the diversity between the structures, there exist twilight-zone cases, in which it is unclear what the correct annotation should be. Therefore, even determining the error rate is very subjective. We performed a manual validation for CYP and GPCR families, and we can claim that the ratio of incorrectly annotated SSEs was under 3% and 0.5%, respectively.
7. Running time of SecStrAnnotator on one domain is typically a few seconds, depending on the size of the structures and available hardware. `SecStrAnnotator_batch.py` can reduce the overall running time for the whole family by running on several CPU cores in parallel (option `--threads`).
8. Visual inspection of the resulting annotation in the automatically created PyMOL session is a simple way of checking the results for possible wrongly annotated SSEs. However, this becomes less convenient as the number of annotated domains gets bigger. Then performing statistics and detection of outliers can be used to uncover wrong annotations.

9. There are many possible criteria for defining secondary structure, and therefore many different SSA methods have been developed [22, 23]. We use *geom-hbond* as the default method because it focuses on the overall shape of helices instead of the details of hydrogen bonding patterns, which are not relevant for the annotation. On the other hand, hydrogen bond approach is used for β -strands, for two reasons: first, connections between strands are vital (a β -strand is not a β -strand without being bound to another β -strand by a ladder), and second, occurrence of β -bulges significantly disrupts the shape of β -strands, so it is hard to describe their geometry universally.
10. The computational complexity of DP algorithm is quadratic with respect to the number of SSEs in T and Q . The theoretical computational complexity of MOM algorithm is exponential; however, in most cases the running times are very close to those of DP. Therefore, MOM is preferred unless serious performance issues are encountered. For structures without β -strands the two algorithms will give identical matching.
11. The value of parameter K can be adjusted using option `--maxmetric`. Decreasing K will result in stricter matching – only very similar SSEs will be allowed to be matched together; the resulting annotation will therefore tend to contain less annotated SSEs but will also be less likely to contain wrong annotations. Increasing K will make the algorithm more tolerant to differences between matched SSEs. This might be necessary in protein families with higher structural diversity. However, it should be done with precaution because too high values of K will cause the algorithm to maximize the number of matched SSEs without focusing on their similarity. High values of K can also slow down MOM algorithm. The option `--maxmetric` can also be used to define K as a linear function of the lengths of SSEs X_i and Y_i and so to put more importance on matching longer SSEs. It is possible to find out the score S for a pair of SSEs from output file `score_matrix.tsv` (when `SecStrAnnotator` is run with `--verbose`). Values of metric μ for matched SSE pairs are included in the output annotation file (field `metric_value`).
12. There is also a constraint that each helix (ladder) can be matched to at most one helix (ladder) – this constraint was not mentioned in DP and pure MOM, because it was a logical consequence of the other constraints.
13. Although obtaining the SSE annotations is the ultimate goal in this chapter, it is often advisable to perform additional statistics on the results over a larger set of proteins from the protein family. Analysis of the distribution of length of the individual

SSEs can provide overview of the general SSE anatomy of the protein family. It can also detect outliers, which may be due to erroneous annotations. Finally, it can help uncover interesting features, like correlation of the structure with source organism or function of the protein.

Acknowledgments

This work was supported by ELIXIR CZ research infrastructure project (MEYS) [LM2015047 to A.M., I.H.V., J.H., K.B., and R.S.V.]; Ministry of Education, Youth and Sports of the Czech Republic under the project CEITEC 2020 [LQ1601 to A.M., R.S.V., and J.K.]; ELIXIR-EXCELERATE project, which received funding from the European Union's Horizon 2020 research and innovation program [676559]; ELIXIR-CZ: Budování kapacit [CZ.02.1.01/0.0/0.0/16_013/0001777]; Ministry of Education, Youth and Sports of the Czech Republic [project CZ.02.1.01/0.0/0.0/16_019/0000754 to V.N. and K.B.]; and Palacky University Olomouc [IGA_PrF_2018_032 to V.N.]. A.M. is a “Brno Ph.D. Talent” scholarship holder funded by Brno City Municipality.

References

1. Berman HM, Westbrook J, Feng Z, Gilliland G, Bhat TN, Weissig H et al (2000) The protein data bank. Nucleic Acids Res 28(1):235–242. <https://doi.org/10.1093/nar/28.1.235>
2. Sillitoe I, Lewis TE, Cuff A, Das S, Ashford P, Dawson NL et al (2015) CATH: comprehensive structural and functional annotations for genome sequences. Nucleic Acids Res 43(D1):D376–D381. <https://doi.org/10.1093/nar/gku947>
3. Fox NK, Brenner SE, Chandonia JM (2014) SCOPe: structural classification of proteins—extended, integrating SCOP and ASTRAL data and classification of new structures. Nucleic Acids Res 42(D1):D304–D309. <https://doi.org/10.1093/nar/gkt1240>
4. Poulos TL, Finzel BC, Howard AJ (1987) High-resolution crystal structure of cytochrome P450cam. J Mol Biol 195(3):687–700. [https://doi.org/10.1016/0022-2836\(87\)90190-2](https://doi.org/10.1016/0022-2836(87)90190-2)
5. Rowland P, Blaney FE, Smyth MG, Jones JJ, Leydon VR, Oxbrow AK et al (2006) Crystal structure of human cytochrome P450 2D6. J Biol Chem 281(11):7614–7622. <https://doi.org/10.1074/jbc.M511232200>
6. Cojocaru V, Winn PJ, Wade RC (2007) The ins and outs of cytochrome P450s. Biochim Biophys Acta 1770(3):390–401. <https://doi.org/10.1016/j.bbagen.2006.07.005>
7. Hutchinson EG, Thornton JM (1990) HERA—a program to draw schematic diagrams of protein secondary structures. Proteins 8(3):203–212. <https://doi.org/10.1002/prot.340080303>
8. Hutchinson EG, Thornton JM (1996) PROMOTIF—a program to identify and analyze structural motifs in proteins. Protein Sci 5(2):212–220. <https://doi.org/10.1002/pro.5560050204>
9. Stivala A, Wybrow M, Wirth A, Whisstock JC, Stuckey PJ (2011) Automatic generation of protein structure cartoons with Pro-origami. Bioinformatics 27(23):3315–3316. <https://doi.org/10.1093/bioinformatics/btr575>
10. Svobodova Varekova R, Midlik A, Hutarova Varekova I, Hutar J, Navratilova V, Koca J et al (2018) Secondary structure elements—annotations and schematic 2D visualizations stable for individual protein families. Biophys J 114(3):46a–47a. <https://doi.org/10.1016/j.bpj.2017.11.307>

11. Berman H, Henrick K, Nakamura H (2003) Announcing the worldwide Protein Data Bank. *Nat Struct Biol* 10(12):980–980. <https://doi.org/10.1038/nsb1203-980>
12. Velankar S, Dana JM, Jacobsen J, van Ginkel G, Gane PJ, Luo J et al (2013) SIFTS: structure integration with function, taxonomy and sequences resource. *Nucleic Acids Res* 41 (D1):D483–D489. <https://doi.org/10.1093/nar/gks1258>
13. Finn RD, Coggill P, Eberhardt RY, Eddy SR, Mistry J, Mitchell AL et al (2016) The Pfam protein families database: towards a more sustainable future. *Nucleic Acids Res* 44(D1): D279–D285. <https://doi.org/10.1093/nar/gkv1344>
14. The PyMOL Molecular Graphics System, Version 2.0 Schrodinger, LLC
15. Kabsch W, Sander C (1983) Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers* 22(12):2577–2637. <https://doi.org/10.1002/bip.360221211>
16. Gore S, Sanz Garcia E, Hendrickx PMS, Gutmanas A, Westbrook JD, Yang H et al (2017) Validation of structures in the Protein Data Bank. *Structure* 25(12):1916–1927. <https://doi.org/10.1016/j.str.2017.10.009>
17. Shindyalov IN, Bourne PE (1998) Protein structure alignment by incremental combinatorial extension (CE) of the optimal path. *Protein Eng* 11(9):739–747. <https://doi.org/10.1093/protein/11.9.739>
18. Mitchell EM, Artymiuk PJ, Rice DW, Willett P (1990) Use of techniques derived from graph theory to compare secondary structure motifs in proteins. *J Mol Biol* 212(1):151–166. [https://doi.org/10.1016/0022-2836\(90\)90312-A](https://doi.org/10.1016/0022-2836(90)90312-A)
19. Eddy SR (2004) What is dynamic programming? *Nat Biotechnol* 22:909. <https://doi.org/10.1038/nbt0704-909>
20. 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(3):443–453. [https://doi.org/10.1016/0022-2836\(70\)90057-4](https://doi.org/10.1016/0022-2836(70)90057-4)
21. Bron C, Kerbosch J (1973) Algorithm 457: finding all cliques of an undirected graph. *Commun ACM* 16(9):575–577. <https://doi.org/10.1145/362342.362367>
22. Anderson CA, Rost B (2009) Secondary structure assignment. In: Gu J, Bourne PE (eds) Structural bioinformatics, 2nd edn. Wiley, Hoboken
23. Cao C, Xu ST, Wang LC (2015) An algorithm for protein helix assignment using helix geometry. *PLoS One* 10(7):20. <https://doi.org/10.1371/journal.pone.0129674>



Chapter 4

Computational Prediction of Secondary and Supersecondary Structures from Protein Sequences

Christopher J. Oldfield, Ke Chen, and Lukasz Kurgan

Abstract

Many new methods for the sequence-based prediction of the secondary and supersecondary structures have been developed over the last several years. These and older sequence-based predictors are widely applied for the characterization and prediction of protein structure and function. These efforts have produced countless accurate predictors, many of which rely on state-of-the-art machine learning models and evolutionary information generated from multiple sequence alignments. We describe and motivate both types of predictions. We introduce concepts related to the annotation and computational prediction of the three-state and eight-state secondary structure as well as several types of supersecondary structures, such as β hairpins, coiled coils, and α -turn- α motifs. We review 34 predictors focusing on recent tools and provide detailed information for a selected set of 14 secondary structure and 3 supersecondary structure predictors. We conclude with several practical notes for the end users of these predictive methods.

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*, which concerns regular local substructures including α -helices and β -strands that were first postulated by Pauling and coworkers [1, 2]; and *tertiary structure*, which is the three-dimensional structure of a protein molecule. Supersecondary structure (SSS) bridges the two latter levels and concerns specific combinations/geometric arrangements of just a few secondary structure elements. Common supersecondary structures include α -helix hairpins, β hairpins, coiled coils, Greek key, and β - α - β , α -turn- α , α -loop- α , and Rossmann motifs. The secondary and SSS elements are combined together, with the help of various types of coils, to form the tertiary structure. An example that displays the

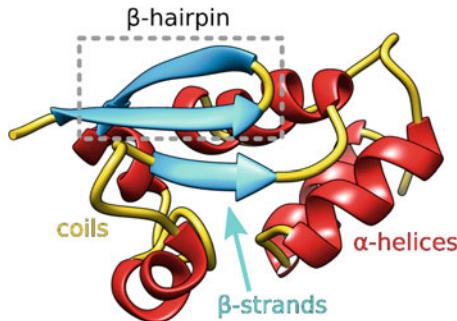


Fig. 1 Cartoon representation of the tertiary structure of the T1 domain of human renal potassium channel Kv1.3 (PDB code: 4BGC). Secondary structures are color-coded: α -helices (red), β -strands (blue), and coils (yellow). The β hairpin supersecondary structure motif, which consists of two β -strands and the coil between them, is denoted using the dotted rectangle

secondary structures and the β hairpin supersecondary structure is given in Fig. 1.

In the 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 May 2018, about 140,000 structures for 44,000 distinct protein sequences are included in the Protein Data Bank (PDB) [4, 5] when compared with 110.3 million nonredundant protein sequences in the RefSeq database [6, 7]. Moreover, the experimental determination of protein structure is relatively expensive and time-consuming and cannot keep up with the rapid accumulation of the sequence data [8–14]. One successful way to predict the tertiary structure is to proceed in a stepwise fashion. First, we predict how the sequence folds into the secondary structure and then how these secondary structure elements come together to form SSSs, and finally the information about the secondary and supersecondary structures is used to help in computational determination of the full three-dimensional molecule [15–22].

The last three decades observed strong progress in the development of accurate predictors of the secondary structure, with predictions with about 82% accuracy [23]. This number has climbed in recent years to 84%, which is approaching the estimated accuracy limit of approximately 88% [24]. Besides being useful for the prediction of the tertiary structure, the secondary structure predicted from the sequence is widely adopted for analysis and prediction of numerous structural and functional characteristics of proteins. These applications include computation of multiple alignment [25], target selection for structural genomics [26–28], and

prediction of protein-nucleic acids interactions [29–32], protein-ligand interactions [33–35], residue depth [36, 37], beta-turns [38], structural classes and folds [39–43], residue contacts [44, 45], disordered regions [46–51], disordered linker regions [52], disordered protein-binding regions [53, 54], and folding rates and types [55–57], to name selected few. Secondary structure predictors enjoy strong interest, which could be quantified by the massive workloads that they handle. For instance, the web server of the arguably one of the most popular methods, PSIPRED, was reported already in 2005 to receive over 15,000 requests per month [58]. Another indicator is the fact that many of these methods receive high citations counts. A review [59] reported that 7 methods were cited over 100 times, and two of them, PSIPRED [58, 60, 61] and PHD [62, 63], were cited over 1300 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 the 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 [64, 65], microbial pathogens [66], and synthases [67], simulation of protein folding [68], analysis of relation between coiled coils and disorder [69], genome-wide studies of protein structure [70, 71], and prediction of protein domains [72]. 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 group [73] and the BhairPred method [74], 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 [75–77] and the utilization of consensus-based approaches [78, 79]. Several 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 [23, 80, 81]. Most recent reviews cover many of the current state-of-the-art secondary structure prediction methods, but they lack the coverage of the supersecondary structure predictors [24, 82]. The SSS prediction area has been reviewed less extensively. The β hairpin and coiled coil predictors, as well as the secondary structure predictors, were overviewed in 2006 [83], and a comparative analysis of the coiled coil predictors was presented in the same year [84]. A recent review provides an in depth guide to the prediction of coiled coils [85]. To the best of our knowledge, there were only two surveys

that covered both secondary and supersecondary structure predictors [86, 87]. This chapter extends our review from 2013 on the predictors of secondary and supersecondary structures [87] by including the most recent advancements and methods in these active areas of research. We summarize a comprehensive set of 34 recent secondary structure and SSS predictors, with 17 methods for each type of predictions. We also demonstrate how the prediction of the secondary structure is used to implement a SSS predictor and provide several practical notes for the end users.

2 Materials

2.1 Assignment of Secondary Structure

Secondary structure, which is assigned from experimentally determined protein structure, is used for a variety of applications, including visualization [88–90] and classification of protein folds [91–96], and as a ground truth to develop and evaluate the secondary and SSS predictors. Several annotation protocols were developed over the last few decades. The first implementation was done in the late 1970s by Levitt and Greer [97]. This was followed by Kabsch and Sander who developed a method called Dictionary of Protein Secondary Structure (DSSP) [98], which is based on the detection of hydrogen bonds defined by an electrostatic criterion. Many other secondary structure assignment methods have been developed, including (in chronological order): DEFINE [99], P-CURVE [100], STRIDE [101], P-SEA [102], XTLSSTR [103], SECSTR [104], KAKSI [105], Segno [106], PALSSE [107], SKSP [108], PROSIGN [109], SABA [110], PSSC [111], PCASSO [112], and SACF [113]. Moreover, the 2Struc web server provides an integrated access to multiple annotation methods and enables convenient comparison between different assignment protocols [114].

DSSP remains the most widely used protocol [105], 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) [115] and the EValuation of Automatic protein structure prediction (EVA) continuous benchmarking project [116]. DSSP determines secondary structures based on 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) β_10 helix, where the carboxyl group of a given amino acid forms a hydrogen bond with amide group of the residue

three positions down in the sequence forming a tight, right-handed helical structure with three 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 four positions away.
- I: (5-turn) π -helix, where the hydrogen bonding occurs between residues spaced five positions away. Most of the π -helices are right-handed.
- E: extended strand, where two 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 i th 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 <https://swift.cmbi.umcn.nl/gv/dssp/>.

2.2 Assignment of Supersecondary Structures

SSS is composed of several adjacent secondary structure elements. Therefore, the assignment of SSS relies on the assignment of the secondary structure. Among more than a dozen types of SSSs, 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 [74]. The coiled coil motif mediates the oligomerization of a large number of proteins, are involved in regulation of gene

expression, and serve as molecular spacers [117, 118]. The α -turn- α (helix-turn-helix) motif is instrumental for DNA binding and transcription regulation [119, 120]. The β hairpin, coiled coil, and α -turn- α motifs are defined as follows:

- A β hairpin motif contains two strands that are adjacent in the primary structure, oriented in an antiparallel arrangement, and linked by a short loop.
- A coiled coil motif is built by two or more α -helices that wind around each other to form a supercoil.
- An α -turn- α motif is composed of two α -helices joined by a short turn structure.

β hairpins are commonly annotated by PROMOTIF program [121], which also assigns several other SSS types, e.g., psi-loop and β - α - β motifs. Similar to DSSP, the PROMOTIF program assigns SSS based on the distances and hydrogen bonding between the residues. Coiled coils are usually assigned with the SOCKET program [122], which locates/annotates coiled coil interactions based on the distances between multiple helical chains. 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 user convenience, certain supersecondary structures, such as the coiled coils and β - α - β motifs, can be accessed, analyzed, and visualized using specialized databases like CCPLUS [123] and TOPS [124]. 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 chirality of selected SSSs, e.g., β hairpins and β - α - β motifs.

2.3 Multiple Sequence Alignment

Multiple sequence alignments were introduced to prediction of secondary structure in the early 1990s [125]. Using multiple sequence alignment information rather than only protein sequence has led to a 10% accuracy improvement in secondary structure prediction [125]. Multiple sequence alignments are also often used in the prediction of SSS [74, 83, 84]. The strength of including multiple sequence alignment information in prediction is the evolutionary information they contain, which is much richer (or accessible) than a single sequence. Multiple sequence alignments can be obtained 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

Query protein	... E R V V I N I S G L R F E T Q L K T L - Q F P E ...
Q61923	... E R L V I N I S G L R F E T Q L R T L S L F P D ...
Q8I4B0	... Q I V T I N V S G M R F Q T F E S T L S R Y P N ...
P17972	... N R V V L N V G G I R H E T Y K A T L K K I P A ...
Q63881	... A L I V L N V S G T R F Q T W Q D T L E R Y P D ...
Q01956	... G K I V I N V G G V R H E T Y R S T L R T L P G ...
P97557	... D C L T V N V G G S R F V L S Q Q A L S C F P H ...
Q0P583	... D S F T V N V G G S R F V L S Q Q A L S C F P H ...
O18868	... R R V R L N V G G L A H E V L W R T L D R L P R ...

Fig. 2 Multiple sequence alignment between the input (query) sequence, which is a fragment of the T1 domain of human renal potassium channel Kv1.3 shown in Fig. 1, and similar sequences. 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 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 glutamic acid (E), glutamine (Q), and valine (V) are 5, 2, and 2, respectively. Therefore, this position can be represented by a 20-dimensional vector (0, 0, 0, 5/9, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2/9, 0, 0, 0, 2/9, 0, 0), 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. A multiple sequence alignment profile is composed of these 20-dimensional vectors for each position in the input protein chain and is a common representation of a multiple sequence alignment.

The PSI-BLAST (Position-Specific Iterated BLAST) [126] 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 position-specific scoring matrix (PSSM), which is similar to a profile discussed above with the exception that values are the log-odds of observing a given residue. Another query against the sequence database is run using this first PSSM, and a larger group of sequences is found. This larger group of sequences is used to construct another PSSM, 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 the late 1990s, PSI-BLAST is commonly used for the generation of PSSMs that are often used directly in the prediction of secondary and

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

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. 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 , where $p_{i,j}$ indicates the probability that the j th amino acid type (in columns) occurs at i th position in the input chain (in rows), is generated. The position-specific scoring matrix M is defined as $m_{i,j} = \log(p_{i,j}/b_j)$, where b_j is the background frequency of the j th amino acid type

supersecondary structures. 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. Many secondary structure predictors use a sliding window approach in which a local stretch of residues around a central position in the window is used to predict the secondary structure state at the central position. Moreover, as one of the first steps in the prediction protocol, many 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. Early predictors

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

Name	Year last published	Prediction model	States	Availability
MUFOLD-SS	2018	Deep neural network	8	SP
SPIDER3	2017	Bidirectional recurrent neural network	3	WS + SP
RaptorX	2016	Deep conditional neural fields	8	WS
Jpred	2015	Neural network	3	WS + API
SCORPION	2014	Neural network	8	WS
PSIPRED	2013	Neural network	3	WS + SP + API
PORTER	2013	Bidirectional recurrent neural network	3	WS + SP
SPARROW	2012	Quadratic model + neural network	3	SP
Frag1D	2010	Scoring function	3	WS + SP
DISSPred	2009	Support vector machine + clustering	3	WS
PCI-SS	2009	Parallel cascade identification	3	WS + API
PROTEUS	2008	Neural network	3	WS + SP
OSS-HMM	2006	Hidden Markov model	3	SP
YASSPP	2006	Support vector machine	3	WS
YASPIN	2005	Neural network + hidden Markov model	3	WS
SABLE	2005	Neural network	3	WS + SP
SSpro	2005	Neural network	8	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), a web server (WS), and/or an application programmer’s interface (API) is available. The methods are sorted by the year of their last publication in the descending order.

were implemented based on a relatively simple statistical analysis of composition of the input sequence. Modern methods adopt sophisticated machine learning-based classifiers to represent the relation between the input sequence (or more precisely between evolutionary information encoded in its PSSM) and the secondary structure states. In the 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), different scales of the networks (e.g., regular and deep), and different sizes of the sliding windows.

Prediction methods are provided to the end users as standalone applications and/or as web servers. Standalone programs are

suitable for higher-volume (for a large 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 [23] 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 17 methods in the reverse chronologic order: MUFOLD-SS [127], SPIDER3 [128], RaptorX [129, 130], Jpred [131–134], SCORPION [135], PSIPRED [58, 60, 61, 136], PORTER [137–139], SPARROW [140], Frag1D [141], DISSPred [142], PCI-SS [143], PROTEUS [131, 144, 145], OS-HMM [146], YASSPP [147], YASPIN [148], SABLE [149], and SSpro [150, 151]. This list is limited to predictors published since 2005 and that have standalone programs or websites available at the time of this writing. Older methods were comprehensively reviewed in [75–77]. Note that only 4 of the 17 methods (MUFOLD-SS, RaptorX, SCORPION, and SSpro) predict the 8-state secondary structure, and these methods also provide the 3-state predictions.

Below, we discuss in detail 14 methods that are listed in reverse chronologic order. These methods 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 observe 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 comparative studies (*see Note 2*) [23, 59]. Moreover, research shows that improved predictive performance could be obtained by post-processing of the secondary structure predictions (*see Note 3*) [152].

3.1.1 SPIDER3

The SPIDER series of predictors have been developed by the Zhou group at the Griffith University. In particular, SPIDER3 [128] is inspired by its predecessors that have come from the same lab: SPIDER2 [153, 154] and SPINE X [155]. SPIDER3 is designed to consider long-range sequence information to improve secondary structure prediction accuracy. Common neural network

architectures require a fixed size input window, and the network is applied to the sequence repeatedly over a sliding window. In contrast, bidirectional recurrent neural networks (BRNN) used in SPIDER3, in a sense, consider the entire sequence simultaneously, with network state being shared along the sequence [156]. Moreover, use of specialized long short-term memory (LSTM) network nodes improves the flow of information between distant sequence positions [157]. The authors demonstrate that the LSTM-BRNN improves prediction accuracy, particularly for residues with many long-range contacts. SPIDER3 achieves a Q3 score of nearly 84% on an independent test dataset.

Inputs: hidden Markov model profiles and PSSM generated from the input protein sequence using HHBlits [158] and PSI-BLAST, respectively.

Architecture: LSTM-BRNN.

Availability: <http://sparks-lab.org/server/SPIDER3/>.

3.1.2 RaptorX

RaptorX [129, 130] was developed by the Xu group at the University of Chicago. Like SPIDER3, this predictor also considers long-range sequence information but through the use of deep convolutional neural networks (DeepCNF). For each layer of the network, inputs are taken from the previous layer from neighboring positions. This architecture considers information from distant sequence positions, where the depth of the network determines the maximum distance considered. RaptorX uses a window size of 11 residues and 5 layers, resulting in an effective window size of 51 residues. On an independent test set, the authors find RaptorX to outperform all other tested methods.

Inputs: PSSM generated from the input protein sequence using PSI-BLAST.

Architecture: deep convolutional neural network.

Availability: <http://raptorgx2.uchicago.edu/StructurePropertyPred/predict/>.

3.1.3 Jpred

Jpred was originally developed in the late 1990s by Barton group at the University of Dundee [132]. This method was updated a few times, with the most recent version Jpred 4 [134]. Similar to PSIPRED, Jpred was demonstrated to provide about 82% accuracy for the three-state secondary structure prediction [134]. 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 [159].

Inputs: hidden Markov model profiles and PSSM generated from the input protein sequence using HMMer [160] and PSI-BLAST, respectively.

Architecture: ensemble of neural networks.

Availability: <http://www.compbio.dundee.ac.uk/jpred/>.

3.1.4 SCORPION

The Li group of Old Dominion University developed the SCORPION method to take advantage of local sequence features for the prediction of secondary structure [135]. This is accomplished by using statistics derived from residue pairs and triplets at defined sequence distances within a short local window. The other key aspect of this method is its stacked architecture. Stacking is an approach where the output of the first predictor is used for the input to the second predictor, second to third, etc. SCORPION uses a series of three stacked neural networks to refine secondary structure predictions. By the authors' assessment, SCORPION shows superior accuracy to other prediction methods, in both Q3 (three-state accuracy) and Q8 (eight-state accuracy) measures. Though improvement in Q3 accuracy over PSIPRED is small, SCORPION shows a large improvement in segment-based accuracy, indicating a better match to secondary structure elements.

Inputs: PSSM generated from the input protein sequence using PSI-BLAST.

Architecture: three stacked neural networks.

Availability: three-state prediction, <http://hpcr.cs.odu.edu/c3scorpion/>; eight-state prediction, <http://hpcr.cs.odu.edu/c8scorpion/>.

3.1.5 PSIPRED

PSIPRED is one of the most popular prediction methods (*see Note 4*); for example, it received the largest number of citations as shown in [23, 59]. This method was developed in the late 1990s by Jones group at the University College London [60] and was later improved and updated in 2020 and 2013 [61, 136]. 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 three-state secondary structure predictions with 81% accuracy [23, 61]. 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/>.

3.1.6 PORTER

This predictor was developed by Pollastri group at the University College Dublin [138]. 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 [137]. The original and the homology-enhanced versions were recently shown to provide 79% [23] and 83% accuracy [59], respectively. PORTER is a part of a comprehensive predictive platform called DISTILL [161], which also incorporates predictors of relative solvent accessibility, residue-residue contact density, contacts

maps, subcellular localization, and tertiary structure. The most recent upgrades to PORTER in 2009 [162] and 2013 [139] expanded the training set and architecture, resulting in a significant increase in performance.

Inputs: PSSM generated from the input protein sequence using PSI-BLAST.

Architecture: ensemble of recurrent neural networks.

Availability: <http://distill.ucd.ie/porter/>.

3.1.7 *Frag1D*

The idea behind Frag1D, created by the Hovmöller group at Stockholm University, is that similar sequence fragments will have similar structures [141]. A database of fragments from known structures is compared to fragments of the query protein, where the most similar segments are used to predict secondary structure. Fragment comparison is scored using profile-profile comparison, augmenting sequence-derived profiles with structure-derived profiles. For the query sequence, structure profiles are unknown and are approximated through an iterative procedure of fragment matching. By the authors' evaluation, this method has comparable performance to PSIPRED.

Inputs: sequence profiles generated from the input protein sequence using PSI-BLAST.

Architecture: scoring function.

Availability: <http://frag1d.bioshu.se>.

3.1.8 *DISSPred*

The DISSPred approach was recently introduced by Hirst group at the University of Nottingham [142]. Similar to SPIDER3 and its predecessors, this method predicts both the three-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% [142].

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/>.

3.1.9 *PCI-SS*

PCI-SS [143] is a unique approach to secondary structure prediction, developed by the Green group at the Carleton University. The approach, known as parallel cascade identification (PCI), progressively refines predictions using a series of linear/nonlinear function layers. The parameter space of PCI contains discrete components, intractable to conventional training methods, so the authors employ genetic algorithm for model training. The authors find that this method performs comparably to other, more conventional, methods.

Inputs: PSSM generated from the input protein sequence using PSI-BLAST.

Architecture: parallel cascade identification.

Availability: <http://bioinf.sce.carleton.ca/PCISS>.

3.1.10 PROTEUS

This secondary structure prediction approach was developed by Wishart group at the University of Alberta [145]. PROTEUS is a consensus-based method, in which outputs of three secondary structure predictors, namely PSIPRED, Jnet [163], and an in-house TRANSSEC [145], 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 by both the authors [144] and in a comparative survey [23]. This predictor was incorporated into an integrated system called PROTEUS2, which additionally offers prediction of signal peptides, transmembrane helices and strands, and tertiary structure [144].

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/>.

3.1.11 YASSPP

YASSPP was designed by Karypis lab at the University of Minnesota in 2005 [147]. Rather than typically used neural network classifiers, YASSPP instead utilizes multiple support vector machine learners. This method was shown to provide similar predictive quality to PSIPRED [147].

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/>.

3.1.12 YASPIN

The YASPIN method was developed by Heringa lab at the Vrije Universiteit in 2004 [148]. 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 [148]. The predictive performance of YASPIN was evaluated using EVA benchmark and two comparative assessments [23, 61], 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 first level and hidden Markov model in the second level.

Availability: <http://www.ibi.vu.nl/programs/yaspinwww/>.

3.1.13 SABLE

The SABLE predictor was developed by Meller group at the University of Cincinnati [149]. The web server that implements this method was used close to 200,000 times since it became operational in 2003. Two recent comparative studies [23, 61] 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/>.

3.1.14 SSpro

SSpro was introduced in early 2000 by the Baldi group at the University of California, Irvine [151]. Its version 4.5 [150] utilizes homology modeling, which is based on alignment to known tertiary structures from PDB, and achieves over 82% accuracy [23]. The SSpro 4.0 was also ranked as one of the top secondary structure prediction servers in the EVA benchmark [164]. SSpro's most recent version 5.2 is part of a comprehensive prediction center called SCRATCH, which also includes predictions of secondary structure in eight states using SSpro8 [151] and prediction of solvent accessibility, intrinsic disorder, contact numbers and contact maps, domains, disulfide bonds, B-cell epitopes, solubility upon overexpression, antigenicity, viral capsid and tail proteins, 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/>.

3.2 Supersecondary Structure Prediction Methods

Since SSS predictors are designed for a specific type of the supersecondary structures, e.g., SpiroCoil only predicts the coiled coils [70], the prediction of 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. The architectures of the methods for the prediction of different types of SSSs vary more substantially when compared with the fairly uniform architectures of the modern secondary structure predictors that primarily rely on the neural networks.

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 [73]. More recent β hairpin predictors use the predicted secondary structure

and some sequence-based descriptors to represent the predicted sequence [74, 165–169]. Moreover, several types of prediction algorithms, such as neural networks, support vector machines, quadratic discriminant functions, and random forests, were used for the prediction of the β hairpin motifs. The most recent predictor, STARpdb-beta hairpin [170], is based on a simple alignment into structurally annotated proteins collected from PDB.

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 position-specific scoring matrix derived from a statistical analysis of a coiled coil database [94]. More recent studies utilize hidden Markov models and a PSSM profile to represent the input sequence [70, 171–175]. Predictive quality of these tools was empirically evaluated in a recent comparative review [85] (*see Note 5*).

The initial study on the prediction of the α -turn- α motif was also based on scoring similarity between the predicted sequence and the α -turn- α structure library [176]. Subsequently, the method uses a pattern dictionary developed from known α -turn- α structures [177, 178], where predictions are made directly from pattern similarity [177] or using a classifier over pattern occurrences [178].

Table 2 summarizes 17 supersecondary structure prediction methods, including 7 β hairpin predictors (in chronological order) (method by de la Cruz et al. [73], BhairPred [74], methods by Hu et al. [168], Zou et al. [167], Xia et al. [166], and Jia et al. [165], and the STARpdb-beta hairpin method [170]); 7 coiled coil predictors (MultiCoil2 [179, 180], MARCOIL [175], PCOILS [174], bCIPA [173], Paircoil2 [172], CCHMM_PROF [171], and SpiriCoil [70]); and 3 α -turn- α predictors (method by Dodd and Egan [176], GYM [177], and method by Xiong et al. [178]). Older coiled coil predictors were reviewed in [84].

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.

3.2.1 BhairPred

The BhairPred predictor was developed by Raghava group at the Institute of Microbial Technology, India, in 2005 [74]. The predictions are performed using a support vector machine-based model, 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 [181]. BhairPred was shown to provide predictions with accuracy in the 71–78% range on two independent test sets [74].

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

Supersecondary structure type	Name (<i>authors</i>)	Year last published	Prediction model	Availability
β hairpin	STARpdb-beta hairpin	2016	Sequence similarity	WS
	Jia <i>et al.</i>	2011	Random forest	NA
	Xia <i>et al.</i>	2010	Support vector machine	NA
	Zou <i>et al.</i>	2009	Increment of diversity + quadratic discriminant analysis	NA
	Hu <i>et al.</i>	2008	Support vector machine	NA
	BhairPred	2005	Support vector machine	WS
	de la Cruz <i>et al.</i>	2002	Neural network	NA
Coiled coil	MultiCoil2	2011	Markov random field	WS + SP
	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
α -turn- α	Xiong <i>et al.</i>	2009	Support vector machine	NA
	GYM	2002	Statistical method	WS
	Dodd <i>et al.</i>	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) is available. NA denotes that neither SP nor WS 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

Inputs: PSSM generated from the input protein sequence using PSI-BLAST, three-state secondary structure predicted using PSIPRED, and solvent accessibility predicted with NETASA.

Architecture: support vector machine.

Availability: <http://www.imtech.res.in/raghava/bhairpred/>.

3.2.2 MultiCoil2

The MultiCoil2 prediction method [179] extends the MultiCoil method [180], which in turn is an extension of the Paircoil method [182]. The Paircoil method is based on the pairwise residue statistics of positions within the heptapeptide coiled coil repeat, calculated from a set of known coiled coils. This idea is based on the structural constraints of coiled coil packing and the compensatory nature of neighboring positions with the α -helix. MultiCoil extends this idea to simultaneously predicting two- or three-way coiled coils by employing a separate set of statistics for each. MultiCoil2 improves this approach by replacing the simple scoring scheme of previous methods with a Markov random field. This improvement yields a substantial improvement over both Paircoil and Multicoil when trained on the same dataset [179]; MultiCoil2 correctly

identifies nearly 92% of coiled coil residues with only 0.3% of non-coiled coils incorrectly identified, according to the authors' assessment [179] (*see Note 6*).

Inputs: protein sequence.

Architecture: Markov random field.

Availability: <http://cb.csail.mit.edu/cb/multicoil2/cgi-bin/multicoil2.cgi>.

3.2.3 GYM

The GYM prediction method is based on mining patterns from known helix-turn-helix examples and matching those patterns from novel sequences [177]. Patterns are defined as two or more residues occurring at the same positions within different helix-turn-helix examples. These patterns are discovered with a novel algorithm which ensures that they are maximal (i.e., not a portion of another pattern) and occur with a minimum defined frequency. When matching patterns to novel sequences, the GYM2 method uses the BLOSUM62 matrix to weight patterns by similarity, rather than strict matching.

Inputs: protein sequence.

Architecture: motif scoring.

Availability: <http://users.cis.fiu.edu/~giri/bioinf/GYM2/prog.html>.

3.3 Supersecondary Structure Prediction by Using Predicted Secondary Structure

Since supersecondary structure is composed of several adjacent secondary structure elements, the prediction of the secondary structures should be a useful input to predict SSS (*see Note 7*). Two SSS predictors, BhairPred [74] and the method developed by Thornton group [73], have utilized the predicted secondary structure for the identification of β hairpins. Following, we discuss the latter method to demonstrate how the predicted secondary structure is used for the prediction of the SSS. This method consists of five steps:

- Step 1. Predict secondary structure for a given input sequence using the PHD method [62].
- 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.

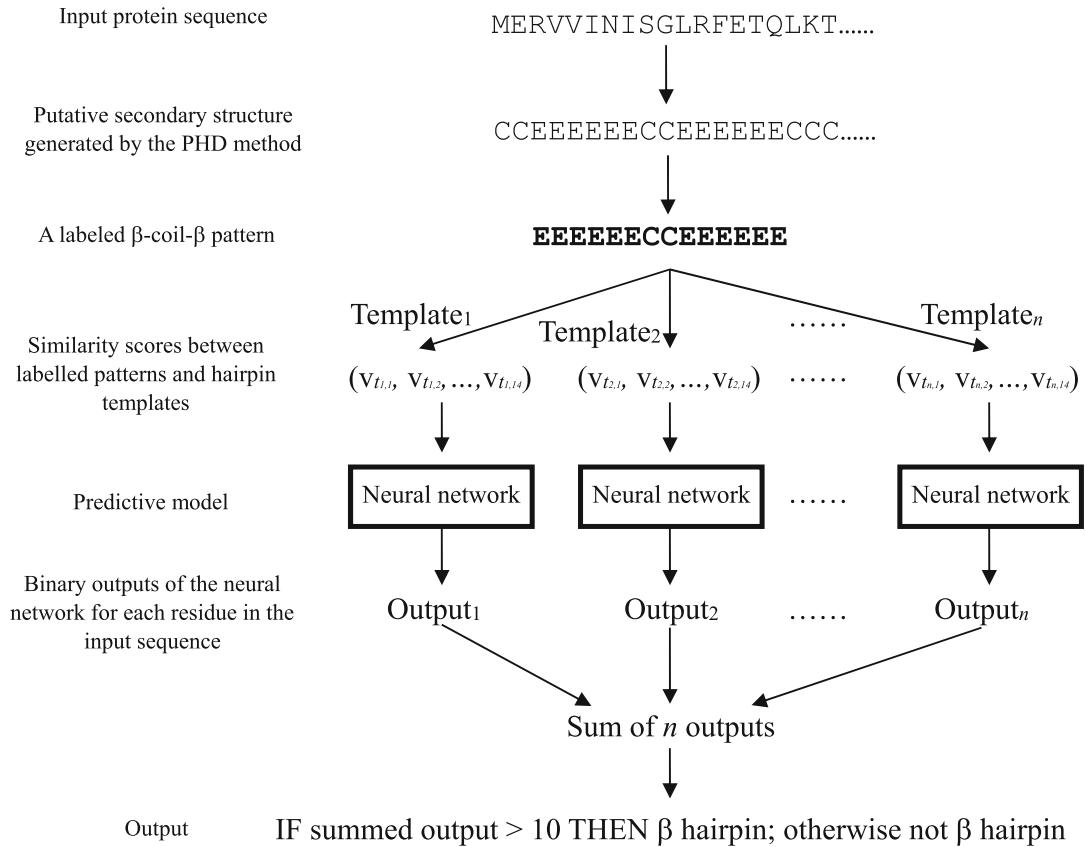


Fig. 4 The architecture of the β hairpin predictor proposed by the Thornton group. The prediction concerns the input (query) sequence, which is a fragment of chain A of the AF1521 protein shown in Fig. 1

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 one for more than ten template hairpins.

The working of the de la Cruz et al. method developed by the Thornton group [73] is illustrated in Fig. 4.

4 Notes

1. The predictive quality of the secondary structure predictors was empirically compared in several large-scale, worldwide initiatives including CASP [115], Critical Assessment of Fully

Automated Structure Prediction (CAFASP) [183], and EVA [116, 164]. 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 2008.

2. A large-scale comparative analysis [23] has revealed a number of interesting and practical observations concerning structure prediction. The accuracy of the three-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. A recent comparative analysis [24] finds that accuracy has climbed to about 84%. Further, they find that errors most commonly confuse helices and coils and strands and coils. Prediction errors for helices and strands are most frequent at the ends of elements, whereas coils show little location bias in errors. Errors are also elevated for residues with many long-range contacts, relative to residues with few long-range contacts.
3. As shown in [23], the current secondary structure predictors are characterized by several drawbacks, which motivate further research in this area. Depending on the predictor, they confuse between 1 and 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 β_{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 May 2018) count of citations in the ISI Web of Knowledge to the paper that describes the original PSIPRED algorithm [60] is close to 3187, which demonstrates the broad usage of this method.
5. A recent assessment evaluated all coiled coil predictors listed in Table 2 [85]. In general, MultiCoil2 and CCHMM_PROF were found to have the highest prediction performance. Due to its architecture, MultiCoil2 cannot detect coiled coils shorter than 21 amino acids, and on a length restricted set,

MultiCoil2 has the best performance. However, on an unrestricted set, CCHMM_PROF has the best performance.

6. Prediction of the supersecondary structures could be potentially improved by utilizing a consensus of different approaches. As shown in a comparative analysis of coiled coil predictors [84], 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. However, another comparative study highlights some problem cases for consensus prediction and instead calls for further predictor development [85].
7. The major obstacle to utilize the predicted secondary structure in the prediction of the supersecondary structures, which was observed in the 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 [73]. The use of the native rather than the predicted secondary structure was shown to lead to a significant improvement in the prediction of the supersecondary structures [74].

Acknowledgments

This work was supported by the Qimonda Endowment funds to L.K.

References

1. Pauling L, Corey RB (1951) The pleated sheet, a new layer configuration of polypeptide chains. *Proc Natl Acad Sci* 37 (5):251–256
2. Pauling L, Corey RB, Branson HR (1951) The structure of proteins: two hydrogen-bonded helical configurations of the polypeptide chain. *Proc Natl Acad Sci* 37(4):205–211
3. Anfinsen CB (1973) Principles that govern the folding of protein chains. *Science* 181 (4096):223–230
4. Berman HM (2000) The protein data bank. *Nucleic Acids Res* 28(1):235–242
5. Burley SK, Berman HM, Kleywegt GJ, Markley JL, Nakamura H, Velankar S (2017) Protein data bank (PDB): the single global macromolecular structure archive. *Methods Mol Biol* 1607:627–641
6. Pruitt KD, Tatusova T, Klimke W, Maglott DR (2009) NCBI Reference Sequences: current status, policy and new initiatives. *Nucleic Acids Res* 37(Database):D32–D36
7. O’Leary NA, Wright MW, Brister JR, Ciufo S, Haddad D, McVeigh R, Rajput B, Robbertse B, Smith-White B, Ako-Adjei D, Astashyn A, Badretdin A, Bao Y, Blinkova O, Brover V, Chetvernin V, Choi J, Cox E, Ermolaeva O, Farrell CM, Goldfarb T, Gupta T, Haft D, Hatcher E, Hlavina W, Joardar VS, Kodali VK, Li W, Maglott D, Masterson P, McGarvey KM, Murphy MR, O’Neill K, Pujar S, Rangwala SH, Rausch D,

- Riddick LD, Schoch C, Shkeda A, Storz SS, Sun H, Thibaud-Nissen F, Tolstoy I, Tully RE, Vatsan AR, Wallin C, Webb D, Wu W, Landrum MJ, Kimchi A, Tatusova T, DiCuccio M, Kitts P, Murphy TD, Pruitt KD (2016) Reference sequence (RefSeq) database at NCBI: current status, taxonomic expansion, and functional annotation. *Nucleic Acids Res* 44(D1):D733–D745
8. Gronwald W, Kalbitzer HR (2010) Automated protein NMR structure determination in solution, *Methods in molecular biology*. Humana Press, Totowa
 9. Chayen NE (2009) High-throughput protein crystallization. *Adv Protein Chem Struct Biol* 77:1–22
 10. Zhang Y (2009) Protein structure prediction: when is it useful? *Curr Opin Struct Biol* 19(2):145–155
 11. Ginalski K (2006) Comparative modeling for protein structure prediction. *Curr Opin Struct Biol* 16(2):172–177
 12. Mizianty MJ, Fan X, Yan J, Chalmers E, Woloschuk C, Joachimiak A, Kurgan L (2014) Covering complete proteomes with X-ray structures: a current snapshot. *Acta Crystallogr D Biol Crystallogr* 70(Pt 11):2781–2793
 13. Gao J, Wu Z, Hu G, Wang K, Song J, Joachimiak A, Kurgan L (2018) Survey of predictors of propensity for protein production and crystallization with application to predict resolution of crystal structures. *Curr Protein Pept Sci* 19(2):200–210
 14. Grabowski M, Niedzialkowska E, Zimmerman MD, Minor W (2016) The impact of structural genomics: the first quindecennial. *J Struct Funct Genom* 17(1):1–16
 15. Yang Y, Faraggi E, Zhao H, Zhou Y (2011) Improving protein fold recognition and template-based modeling by employing probabilistic-based matching between predicted one-dimensional structural properties of query and corresponding native properties of templates. *Bioinformatics* 27(15):2076–2082
 16. Roy A, Kucukural A, Zhang Y (2010) I-TASSER: a unified platform for automated protein structure and function prediction. *Nat Protoc* 5(4):725–738
 17. Faraggi E, Yang Y, Zhang S, Zhou Y (2009) Predicting continuous local structure and the effect of its substitution for secondary structure in fragment-free protein structure prediction. *Structure* 17(11):1515–1527
 18. Wu S, Zhang Y (2008) MUSTER: improving protein sequence profile-profile alignments by using multiple sources of structure information. *Proteins* 72(2):547–556
 19. Zhou H, Skolnick J (2007) Ab initio protein structure prediction using chunk-TASSER. *Biophys J* 93(5):1510–1518
 20. Skolnick J (2006) In quest of an empirical potential for protein structure prediction. *Curr Opin Struct Biol* 16(2):166–171
 21. Zhang W, Yang J, He B, Walker SE, Zhang H, Govindarajoo B, Virtanen J, Xue Z, Shen HB, Zhang Y (2016) Integration of QUARK and I-TASSER for ab initio protein structure prediction in CASP11. *Proteins* 84(Suppl 1):76–86
 22. Czaplewski C, Karczynska A, Sieradzan AK, Liwo A (2018) UNRES server for physics-based coarse-grained simulations and prediction of protein structure, dynamics and thermodynamics. *Nucleic Acids Res* 46(W1):W304–W309
 23. Zhang H, Zhang T, Chen K, Kedarisetti KD, Mizianty MJ, Bao Q, Stach W, Kurgan L (2011) Critical assessment of high-throughput standalone methods for secondary structure prediction. *Brief Bioinform* 12(6):672–688
 24. Yang Y, Gao J, Wang J, Heffernan R, Hanson J, Paliwal K, Zhou Y (2018) Sixty-five years of the long march in protein secondary structure prediction: the final stretch? *Brief Bioinform* 19(3):482–494
 25. Pei J, Grishin NV (2007) PROMALS: towards accurate multiple sequence alignments of distantly related proteins. *Bioinformatics* 23(7):802–808
 26. Mizianty MJ, Kurgan L (2011) Sequence-based prediction of protein crystallization, purification and production propensity. *Bioinformatics* 27(13):i24–i33
 27. Slabinski L, Jaroszewski L, Rychlewski L, Wilson IA, Lesley SA, Godzik A (2007) XtalPred: a web server for prediction of protein crystallizability. *Bioinformatics* 23(24):3403–3405
 28. Wang H, Feng L, Webb GI, Kurgan L, Song J, Lin D (2017) Critical evaluation of bioinformatics tools for the prediction of protein crystallization propensity. *Brief Bioinform*. <https://doi.org/10.1093/bib/bbx1018>
 29. Zhang T, Zhang H, Chen K, Ruan J, Shen S, Kurgan L (2010) Analysis and prediction of RNA-binding residues using sequence, evolutionary conservation, and predicted secondary structure and solvent accessibility. *Curr Protein Pept Sci* 11(7):609–628
 30. Yan J, Kurgan L (2017) DRNApred, fast sequence-based method that accurately

- predicts and discriminates DNA- and RNA-binding residues. *Nucleic Acids Res* 45(10):e84
31. Yan J, Friedrich S, Kurgan L (2016) A comprehensive comparative review of sequence-based predictors of DNA- and RNA-binding residues. *Brief Bioinform* 17(1):88–105
 32. Peng Z, Kurgan L (2015) High-throughput prediction of RNA, DNA and protein binding regions mediated by intrinsic disorder. *Nucleic Acids Res* 43(18):e121
 33. Pulim V, Bienkowska J, Berger B (2008) LTHREADER: prediction of extracellular ligand-receptor interactions in cytokines using localized threading. *Protein Sci* 17(2):279–292
 34. Fischer JD, Mayer CE, Söding J (2008) Prediction of protein functional residues from sequence by probability density estimation. *Bioinformatics* 24(5):613–620
 35. Chen K, Mizianty MJ, Kurgan L (2012) Prediction and analysis of nucleotide-binding residues using sequence and sequence-derived structural descriptors. *Bioinformatics* 28(3):331–341
 36. Song J, Tan H, Mahmood K, Law RHP, Buckle AM, Webb GI, Akutsu T, Whisstock JC (2009) Prodepth: predict residue depth by support vector regression approach from protein sequences only. *PLoS One* 4(9):e7072
 37. Zhang H, Zhang T, Chen K, Shen S, Ruan J, Kurgan L (2008) Sequence based residue depth prediction using evolutionary information and predicted secondary structure. *BMC Bioinformatics* 9(1):388
 38. Zheng C, Kurgan L (2008) Prediction of beta-turns at over 80% accuracy based on an ensemble of predicted secondary structures and multiple alignments. *BMC Bioinformatics* 9:430
 39. Mizianty MJ, Kurgan L (2009) Modular prediction of protein structural classes from sequences of twilight-zone identity with predicting sequences. *BMC Bioinformatics* 10(1):414
 40. Kurgan L, Cios K, Chen K (2008) SCPRED: accurate prediction of protein structural class for sequences of twilight-zone similarity with predicting sequences. *BMC Bioinformatics* 9(1):226
 41. Chen K, Kurgan L (2007) PFRES: protein fold classification by using evolutionary information and predicted secondary structure. *Bioinformatics* 23(21):2843–2850
 42. Kong L, Zhang L (2014) Novel structure-driven features for accurate prediction of protein structural class. *Genomics* 103(4):292–297
 43. Kurgan LA, Zhang T, Zhang H, Shen S, Ruan J (2008) Secondary structure-based assignment of the protein structural classes. *Amino Acids* 35(3):551–564
 44. Xue B, Faraggi E, Zhou Y (2009) Predicting residue-residue contact maps by a two-layer, integrated neural-network method. *Proteins* 76(1):176–183
 45. Cheng J, Baldi P (2007) Improved residue contact prediction using support vector machines and a large feature set. *BMC Bioinformatics* 8(1):113
 46. Mizianty MJ, Stach W, Chen K, Kedarisetti KD, Disfani FM, Kurgan L (2010) Improved sequence-based prediction of disordered regions with multilayer fusion of multiple information sources. *Bioinformatics* 26(18):i489–i496
 47. Mizianty MJ, Zhang T, Xue B, Zhou Y, Dunker A, Uversky VN, Kurgan L (2011) In-silico prediction of disorder content using hybrid sequence representation. *BMC Bioinformatics* 12(1):245
 48. Schlessinger A, Punta M, Yachdav G, Kajan L, Rost B (2009) Improved disorder prediction by combination of orthogonal approaches. *PLoS One* 4(2):e4433
 49. Mizianty MJ, Peng ZL, Kurgan L (2013) MFDp2: Accurate predictor of disorder in proteins by fusion of disorder probabilities, content and profiles. *Intrinsically Disord Proteins* 1(1):e24428
 50. Mizianty MJ, Uversky V, Kurgan L (2014) Prediction of intrinsic disorder in proteins using MFDp2. *Methods Mol Biol* 1137:147–162
 51. Walsh I, Martin AJ, Di Domenico T, Vullo A, Pollastri G, Tosatto SC (2011) CSpritz: accurate prediction of protein disorder segments with annotation for homology, secondary structure and linear motifs. *Nucleic Acids Res* 39(Web Server issue):W190–W196
 52. Meng F, Kurgan L (2016) DFLpred: high-throughput prediction of disordered flexible linker regions in protein sequences. *Bioinformatics* 32(12):i341–i350
 53. Yan J, Dunker AK, Uversky VN, Kurgan L (2016) Molecular recognition features (MoRFs) in three domains of life. *Mol BioSyst* 12(3):697–710
 54. Sharma R, Raicar G, Tsunoda T, Patil A, Sharma A (2018) OPAL: prediction of MoRF regions in intrinsically disordered protein sequences. *Bioinformatics* 34(11):1850–1858

55. Zhang H, Zhang T, Gao J, Ruan J, Shen S, Kurgan L (2010) Determination of protein folding kinetic types using sequence and predicted secondary structure and solvent accessibility. *Amino Acids* 42(1):271–283
56. Gao J, Zhang T, Zhang H, Shen S, Ruan J, Kurgan L (2010) Accurate prediction of protein folding rates from sequence and sequence-derived residue flexibility and solvent accessibility. *Proteins* 78(9):2114–2130
57. Jiang Y, Iglicinski P, Kurgan L (2009) Prediction of protein folding rates from primary sequences using hybrid sequence representation. *J Comput Chem* 30(5):772–783
58. Bryson K, McGuffin LJ, Marsden RL, Ward JJ, Sodhi JS, Jones DT (2005) Protein structure prediction servers at University College London. *Nucleic Acids Res* 33(Web Server):W36–W38
59. 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
60. Jones DT (1999) Protein secondary structure prediction based on position-specific scoring matrices. *J Mol Biol* 292(2):195–202
61. Buchan DWA, Ward SM, Lobley AE, Nugent TCO, Bryson K, Jones DT (2010) Protein annotation and modelling servers at University College London. *Nucleic Acids Res* 38 (Web Server):W563–W568
62. Rost B (1996) PHD: predicting one-dimensional protein structure by profile-based neural networks. *Methods Enzymol* 266:525–539
63. Rost B, Yachdav G, Liu J (2004) The Predict-Protein server. *Nucleic Acids Res* 32(Web Server):W321–W326
64. O'Donnell CW, Waldspühl J, Lis M, Halfmann R, Devadas S, Lindquist S, Berger B (2011) A method for probing the mutational landscape of amyloid structure. *Bioinformatics* 27(13):i34–i42
65. Bryan AW, Menke M, Cowen LJ, Lindquist SL, Berger B (2009) BETASCAN: probable β -amyloids identified by pairwise probabilistic analysis. *PLoS Comput Biol* 5(3):e1000333
66. Bradley P, Cowen L, Menke M, King J, Berger B (2001) BETAWRAP: successful prediction of parallel β -helices from primary sequence reveals an association with many microbial pathogens. *Proc Natl Acad Sci* 98 (26):14819–14824
67. Hornung T, Volkov OA, Zaida TMA, Delannoy S, Wise JG, Vogel PD (2008) Structure of the cytosolic part of the subunit b-dimer of *Escherichia coli* FOF1-ATP synthase. *Biophys J* 94(12):5053–5064
68. Sun ZR, Cui Y, Ling LJ, Guo Q, Chen RS (1998) Molecular dynamics simulation of protein folding with supersecondary structure constraints. *J Protein Chem* 17(8):765–769
69. Szappanos B, Süveges D, Nyitrai L, Perczel A, Gáspári Z (2010) Folded-unfolded cross-predictions and protein evolution: the case study of coiled-coils. *FEBS Lett* 584 (8):1623–1627
70. Rackham OJL, Madera M, Armstrong CT, Vincent TL, Woolfson DN, Gough J (2010) The evolution and structure prediction of coiled coils across all genomes. *J Mol Biol* 403(3):480–493
71. Gerstein M, Hegyi H (1998) Comparing genomes in terms of protein structure: surveys of a finite parts list. *FEMS Microbiol Rev* 22 (4):277–304
72. Reddy CCS, Shameer K, Offmann BO, Sowdhamini R (2008) PURE: a webserver for the prediction of domains in unassigned regions in proteins. *BMC Bioinformatics* 9(1):281
73. de la Cruz X, Hutchinson EG, Shepherd A, Thornton JM (2002) Toward predicting protein topology: an approach to identifying β hairpins. *Proc Natl Acad Sci* 99 (17):11157–11162
74. Kumar M, Bhasin M, Natt NK, Raghava GPS (2005) BhairPred: prediction of β -hairpins in a protein from multiple alignment information using ANN and SVM techniques. *Nucleic Acids Res* 33(Web Server):W154–W159
75. Barton GJ (1995) Protein secondary structure prediction. *Curr Opin Struct Biol* 5 (3):372–376
76. Heringa J (2000) Computational methods for protein secondary structure prediction using multiple sequence alignments. *Curr Protein Pept Sci* 1(3):273–301
77. Rost B (2001) Protein secondary structure prediction continues to rise. *J Struct Biol* 134(2–3):204–218
78. Albrecht M, Tosatto SCE, Lengauer T, Valle G (2003) Simple consensus procedures are effective and sufficient in secondary structure prediction. *Protein Eng Des Sel* 16 (7):459–462
79. Yan J, Marcus M, Kurgan L (2014) Comprehensively designed consensus of standalone secondary structure predictors improves Q3 by over 3%. *J Biomol Struct Dyn* 32(1):36–51
80. Rost B (2009) Prediction of protein structure in 1D—secondary structure, membrane

- regions, and solvent accessibility. Structural bioinformatics, 2nd edn. Wiley, New York
81. Pirovano W, Heringa J (2010) Protein secondary structure prediction. *Methods Mol Biol* 609:327–348
82. Meng F, Kurgan L (2016) Computational prediction of protein secondary structure from sequence. *Curr Protoc Protein Sci* 86:2.3.1–2.3.10
83. Singh M (2006) Predicting protein secondary and supersecondary structure, Chapman & Hall/CRC Computer & Information Science Series. Chapman and Hall/CRC, New York
84. Gruber M, Söding J, Lupas AN (2006) Comparative analysis of coiled-coil prediction methods. *J Struct Biol* 155(2):140–145
85. Li C, Ching Han Chang C, Nagel J, Porebski BT, Hayashida M, Akutsu T, Song J, Buckle AM (2016) Critical evaluation of in silico methods for prediction of coiled-coil domains in proteins. *Brief Bioinform* 17(2):270–282
86. Ho HK, Zhang L, Ramamohanarao K, Martin S (2013) A survey of machine learning methods for secondary and supersecondary protein structure prediction. *Methods Mol Biol* 932:87–106
87. Chen K, Kurgan L (2013) Computational prediction of secondary and supersecondary structures. *Methods Mol Biol* 932:63–86
88. Kolodny R, Honig B (2006) VITAL—a new 2D visualization tool of protein 3D structural alignments. *Bioinformatics* 22(17):2166–2167
89. Moreland JL, Gramada A, Buzko OV, Zhang Q, Bourne PE (2005) BMC Bioinformatics 6(1):21
90. Porollo AA, Adamczak R, Meller J (2004) POLYVIEW: a flexible visualization tool for structural and functional annotations of proteins. *Bioinformatics* 20(15):2460–2462
91. 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(4):536–540
92. Orengo CA, Michie AD, Jones S, Jones DT, Swindells MB, Thornton JM (1997) CATH—a hierarchic classification of protein domain structures. *Structure* 5(8):1093–1109
93. Andreeva A, Howorth D, Chandonia JM, Brenner SE, Hubbard TJP, Chothia C, Murzin AG (2007) Data growth and its impact on the SCOP database: new developments. *Nucleic Acids Res* 36(Database):D419–D425
94. Cuff AL, Sillitoe I, Lewis T, Clegg AB, Rentzsch R, Furnham N, Pellegrini-Calace M, Jones D, Thornton J, Orengo CA (2011) Extending CATH: increasing coverage of the protein structure universe and linking structure with function. *Nucleic Acids Res* 39(Database):D420–D426
95. Sillitoe I, Dawson N, Thornton J, Orengo C (2015) The history of the CATH structural classification of protein domains. *Biochimie* 119:209–217
96. Andreeva A, Howorth D, Chandonia JM, Brenner SE, Hubbard TJ, Chothia C, Murzin AG (2008) Data growth and its impact on the SCOP database: new developments. *Nucleic Acids Res* 36(Database issue):D419–D425
97. Levitt M, Greer J (1977) Automatic identification of secondary structure in globular proteins. *J Mol Biol* 114(2):181–239
98. Kabsch W, Sander C (1983) Dictionary of protein secondary structure: Pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers* 22(12):2577–2637
99. Richards FM, Kundrot CE (1988) Identification of structural motifs from protein coordinate data: secondary structure and first-level supersecondary structure. *Proteins Struct Funct Genet* 3(2):71–84
100. Sklenar H, Etchebest C, Lavery R (1989) Describing protein structure: a general algorithm yielding complete helicoidal parameters and a unique overall axis. *Proteins Struct Funct Genet* 6(1):46–60
101. Frishman D, Argos P (1995) Knowledge-based protein secondary structure assignment. *Proteins Struct Funct Genet* 23(4):566–579
102. Labesse G, Colloc'h N, Pothier J, Mornon JP (1997) P-SEA: a new efficient assignment of secondary structure from $\text{C}\alpha$ trace of proteins. *Bioinformatics* 13(3):291–295
103. King SM, Johnson WC (1999) Assigning secondary structure from protein coordinate data. *Proteins Struct Funct Genet* 35(3):313–320
104. Fodje MN, Al-Karadaghi S (2002) Occurrence, conformational features and amino acid propensities for the π -helix. *Protein Eng Des Sel* 15(5):353–358
105. Martin J, Letellier G, Marin A, Taly J-F, de Brevern AG, Gibrat J-F (2005) BMC Struct Biol 5(1):17
106. Cubellis M, Cailliez F, Lovell SC (2005) Secondary structure assignment that accurately reflects physical and evolutionary characteristics. *BMC Bioinformatics* 6(Suppl 4):S8
107. Majumdar I, Krishna SS, Grishin NV (2005) PALSSe: a program to delineate linear

- secondary structural elements from protein structures. *BMC Bioinformatics* 6(1):202
108. Zhang W, Dunker AK, Zhou Y (2008) Assessing secondary structure assignment of protein structures by using pairwise sequence-alignment benchmarks. *Proteins* 71(1):61–67
109. Hosseini S-R, Sadeghi M, Pezeshk H, Eslahchi C, Habibi M (2008) PROSIGN: A method for protein secondary structure assignment based on three-dimensional coordinates of consecutive C^α atoms. *Comput Biol Chem* 32(6):406–411
110. Park S-Y, Yoo M-J, Shin J-M, Cho K-H (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(2):118–122
111. Zacharias J, Knapp EW (2014) Protein secondary structure classification revisited: processing DSSP information with PSSC. *J Chem Inf Model* 54(7):2166–2179
112. Law SM, Frank AT, Brooks CL 3rd (2014) PCASSO: a fast and efficient Calpha-based method for accurately assigning protein secondary structure elements. *J Comput Chem* 35(24):1757–1761
113. Cao C, Wang GS, Liu A, Xu ST, Wang LC, Zou SX (2016) A new secondary structure assignment algorithm using C-alpha backbone fragments. *Int J Mol Sci* 17(3):333
114. Klose DP, Wallace BA, Janes RW (2010) 2Struc: the secondary structure server. *Bioinformatics* 26(20):2624–2625
115. Moult J, Pedersen JT, Judson R, Fidelis K (1995) A large-scale experiment to assess protein structure prediction methods. *Proteins Struct Funct Genet* 23(3):ii–iv
116. Koh IYY (2003) EVA: evaluation of protein structure prediction servers. *Nucleic Acids Res* 31(13):3311–3315
117. Parry DAD, Fraser RDB, Squire JM (2008) Fifty years of coiled-coils and α-helical bundles: a close relationship between sequence and structure. *J Struct Biol* 163(3):258–269
118. Truebestein L, Leonard TA (2016) Coiled-coils: the long and short of it. *BioEssays* 38(9):903–916
119. Pellegrini-Calace M (2005) Detecting DNA-binding helix-turn-helix structural motifs using sequence and structure information. *Nucleic Acids Res* 33(7):2129–2140
120. Aravind L, Anantharaman V, Balaji S, Babu MM, Iyer LM (2005) The many faces of the helix-turn-helix domain: transcription regulation and beyond. *FEMS Microbiol Rev* 29(2):231–262
121. Hutchinson EG, Thornton JM (1996) PROMOTIF-A program to identify and analyze structural motifs in proteins. *Protein Sci* 5(2):212–220
122. Walshaw J, Woolfson DN (2001) SOCKET: a program for identifying and analysing coiled-coil motifs within protein structures. *J Mol Biol* 307(5):1427–1450
123. Testa OD, Moutevelis E, Woolfson DN (2009) CC+: a relational database of coiled-coil structures. *Nucleic Acids Res* 37(Database):D315–D322
124. Michalopoulos I (2004) TOPS: an enhanced database of protein structural topology. *Nucleic Acids Res* 32(90001):D251–D254
125. Rost B, Sander C (1993) Improved prediction of protein secondary structure by use of sequence profiles and neural networks. *Proc Natl Acad Sci* 90(16):7558–7562
126. Altschul S (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res* 25(17):3389–3402
127. Fang C, Shang Y, Xu D (2018) MUFOOLD-SS: new deep inception-inside-inception networks for protein secondary structure prediction. *Proteins* 86(5):592–598
128. Heffernan R, Yang Y, Paliwal K, Zhou Y (2017) Capturing non-local interactions by long short-term memory bidirectional recurrent neural networks for improving prediction of protein secondary structure, backbone angles, contact numbers and solvent accessibility. *Bioinformatics* 33(18):2842–2849
129. Wang S, Li W, Liu S, Xu J (2016) RaptorX-Property: a web server for protein structure property prediction. *Nucleic Acids Res* 44(W1):W430–W435
130. Wang S, Peng J, Ma J, Xu J (2016) Protein secondary structure prediction using deep convolutional neural fields. *Sci Rep* 6:18962
131. Cole C, Barber JD, Barton GJ (2008) The Jpred 3 secondary structure prediction server. *Nucleic Acids Res* 36(Web Server):W197–W201
132. Cuff JA, Clamp ME, Siddiqui AS, Finlay M, Barton GJ (1998) JPred: a consensus secondary structure prediction server. *Bioinformatics* 14(10):892–893
133. Cuff JA, Barton GJ (2000) Application of multiple sequence alignment profiles to improve protein secondary structure prediction. *Proteins Struct Funct Genet* 40(3):502–511
134. Drozdetskiy A, Cole C, Procter J, Barton GJ (2015) JPred4: a protein secondary structure

- prediction server. *Nucleic Acids Res* 43(W1): W389–W394
135. Yaseen A, Li Y (2014) Context-based features enhance protein secondary structure prediction accuracy. *J Chem Inf Model* 54(3):992–1002
136. Buchan DW, Minneci F, Nugent TC, Bryson K, Jones DT (2013) Scalable web services for the PSIPRED Protein Analysis Workbench. *Nucleic Acids Res* 41(Web Server issue):W349–W357
137. Pollastri G, Martin AJM, Mooney C, Vullo A (2007) Accurate prediction of protein secondary structure and solvent accessibility by consensus combiners of sequence and structure information. *BMC Bioinformatics* 8(1):201
138. Pollastri G, McLysaght A (2005) Porter: a new, accurate server for protein secondary structure prediction. *Bioinformatics* 21(8):1719–1720
139. Mirabello C, Pollastri G (2013) Porter, PaleAle 4.0: high-accuracy prediction of protein secondary structure and relative solvent accessibility. *Bioinformatics* 29(16):2056–2058
140. Bettella F, Rasinski D, Knapp EW (2012) Protein secondary structure prediction with SPARROW. *J Chem Inf Model* 52(2):545–556
141. Zhou T, Shu N, Hovmöller S (2010) A novel method for accurate one-dimensional protein structure prediction based on fragment matching. *Bioinformatics* 26(4):470–477
142. Kountouris P, Hirst JD (2009) Prediction of backbone dihedral angles and protein secondary structure using support vector machines. *BMC Bioinformatics* 10(1):437
143. Green JR, Korenberg MJ, Aboul-Magd MO (2009) PCI-SS: MISO dynamic nonlinear protein secondary structure prediction. *BMC Bioinformatics* 10:222–222
144. Montgomerie S, Cruz JA, Shrivastava S, Arndt D, Berjanskii M, Wishart DS (2008) PROTEUS2: a web server for comprehensive protein structure prediction and structure-based annotation. *Nucleic Acids Res* 36(Web Server):W202–W209
145. Montgomerie S, Sundararaj S, Gallin WJ, Wishart DS (2006) Improving the accuracy of protein secondary structure prediction using structural alignment. *BMC Bioinformatics* 7:301
146. Martin J, Gibrat JF, Rodolphe F (2006) Analysis of an optimal hidden Markov model for secondary structure prediction. *BMC Struct Biol* 6:25
147. Karypis G (2006) YASSPP: better kernels and coding schemes lead to improvements in protein secondary structure prediction. *Proteins* 64(3):575–586
148. Lin K, Simossis VA, Taylor WR, Heringa J (2005) A simple and fast secondary structure prediction method using hidden neural networks. *Bioinformatics* 21(2):152–159
149. Adamczak R, Porollo A, Meller J (2005) Combining prediction of secondary structure and solvent accessibility in proteins. *Proteins* 59(3):467–475
150. Cheng J, Randall AZ, Sweredoski MJ, Baldi P (2005) SCRATCH: a protein structure and structural feature prediction server. *Nucleic Acids Res* 33(Web Server):W72–W76
151. Pollastri G, Przybylski D, Rost B, Baldi P (2002) Improving the prediction of protein secondary structure in three and eight classes using recurrent neural networks and profiles. *Proteins Struct Funct Genet* 47(2):228–235
152. Madera M, Calmus R, Thiltgen G, Karplus K, Gough J (2010) Improving protein secondary structure prediction using a simple k-mer model. *Bioinformatics* 26(5):596–602
153. Yang Y, Heffernan R, Paliwal K, Lyons J, Dehzangi A, Sharma A, Wang J, Sattar A, Zhou Y (2017) SPIDER2: a package to predict secondary structure, accessible surface area, and main-chain torsional angles by deep neural networks. *Methods Mol Biol* 1484:55–63
154. Heffernan R, Paliwal K, Lyons J, Dehzangi A, Sharma A, Wang J, Sattar A, Yang Y, Zhou Y (2015) Improving prediction of secondary structure, local backbone angles, and solvent accessible surface area of proteins by iterative deep learning. *Sci Rep* 5:11476
155. Faraggi E, Zhang T, Yang Y, Kurgan L, Zhou Y (2012) SPINE X: improving protein secondary structure prediction by multistep learning coupled with prediction of solvent accessible surface area and backbone torsion angles. *J Comput Chem* 33(3):259–267
156. Schuster M, Paliwal KK (1997) Bidirectional recurrent neural networks. *IEEE Trans Signal Process* 45(11):2673–2681
157. Hochreiter S, Schmidhuber J (1997) Long short-term memory. *Neural Comput* 9(8):1735–1780
158. Remmert M, Biegert A, Hauser A, Soding J (2011) HHblits: lightning-fast iterative protein sequence searching by HMM-HMM alignment. *Nat Methods* 9(2):173–175
159. Lupas A, Van Dyke M, Stock J (1991) Predicting coiled coils from protein sequences. *Science* 252(5009):1162–1164

160. Eddy SR (1998) Profile hidden Markov models. *Bioinformatics* 14(9):755–763
161. Baú D, Martin AJM, Mooney C, Vullo A, Walsh I, Pollastri G (2006) Distill: a suite of web servers for the prediction of one-, two- and three-dimensional structural features of proteins. *BMC Bioinformatics* 7(1):402
162. Mooney C, Pollastri G (2009) Beyond the Twilight Zone: automated prediction of structural properties of proteins by recursive neural networks and remote homology information. *Proteins* 77(1):181–190
163. Cuff JA, Barton GJ (2000) Application of multiple sequence alignment profiles to improve protein secondary structure prediction. *Proteins* 40(3):502–511
164. Eyrich VA, Marti-Renom MA, Przybylski D, Madhusudhan MS, Fiser A, Pazos F, Valencia A, Sali A, Rost B (2001) EVA: continuous automatic evaluation of protein structure prediction servers. *Bioinformatics* 17(12):1242–1243
165. Jia S-C, Hu X-Z (2011) Using random forest algorithm to predict β -hairpin motifs. *Protein Pept Lett* 18(6):609–617
166. Xia J-F, Wu M, You Z-H, Zhao X-M, Li X-L (2010) Prediction of β -hairpins in proteins using physicochemical properties and structure information. *Protein Pept Lett* 17(9):1123–1128
167. Zou D, He Z, He J (2009) β -Hairpin prediction with quadratic discriminant analysis using diversity measure. *J Comput Chem* 30(14):2277–2284
168. Hu XZ, Li QZ (2008) Prediction of the β -hairpins in proteins using support vector machine. *Protein J* 27(2):115–122
169. Kuhn M, Meiler J, Baker D (2004) Strand-loop-strand motifs: Prediction of hairpins and diverging turns in proteins. *Proteins* 54(2):282–288
170. Singh H, Raghava GPS (2016) BLAST-based structural annotation of protein residues using Protein Data Bank. *Biol Direct* 11:4
171. Bartoli L, Fariselli P, Krogh A, Casadio R (2009) CCHMM_PROF: a HMM-based coiled-coil predictor with evolutionary information. *Bioinformatics* 25(21):2757–2763
172. McDonnell AV, Jiang T, Keating AE, Berger B (2006) Paircoil2: improved prediction of coiled coils from sequence. *Bioinformatics* 22(3):356–358
173. Mason JM, Schmitz MA, Muller KM, Arndt KM (2006) Semirational design of Jun-Fos coiled coils with increased affinity: universal implications for leucine zipper prediction and design. *Proc Natl Acad Sci* 103(24):8989–8994
174. Gruber M, Soding J, Lupas AN (2005) REPPER—repeats and their periodicities in fibrous proteins. *Nucleic Acids Res* 33(Web Server):W239–W243
175. Delorenzi M, Speed T (2002) An HMM model for coiled-coil domains and a comparison with PSSM-based predictions. *Bioinformatics* 18(4):617–625
176. Dodd IB, Egan JB (1990) Improved detection of helix-turn-helix DNA-binding motifs in protein sequences. *Nucleic Acids Res* 18(17):5019–5026
177. Narasimhan G, Bu C, Gao Y, Wang X, Xu N, Mathee K (2002) Mining protein sequences for motifs. *J Comput Biol* 9(5):707–720
178. Xiong W, Li T, Chen K, Tang K (2009) Local combinational variables: an approach used in DNA-binding helix-turn-helix motif prediction with sequence information. *Nucleic Acids Res* 37(17):5632–5640
179. Trigg J, Gutwin K, Keating AE, Berger B (2011) Multicoil2: predicting coiled coils and their oligomerization states from sequence in the twilight zone. *PLoS One* 6(8):e23519
180. Wolf E, Kim PS, Berger B (1997) MultiCoil: a program for predicting two-and three-stranded coiled coils. *Protein Sci* 6(6):1179–1189
181. Ahmad S, Gromiha MM (2002) NETASA: neural network based prediction of solvent accessibility. *Bioinformatics* 18(6):819–824
182. Berger B, Wilson DB, Wolf E, Tonchev T, Milla M, Kim PS (1995) Predicting coiled coils by use of pairwise residue correlations. *Proc Natl Acad Sci U S A* 92(18):8259–8263
183. Fischer D, Barret C, Bryson K, Elofsson A, Godzik A, Jones D, Karplus KJ, Kelley LA, MacCallum RM, Pawowski K, Rost B, Rychlewski L, Sternberg M (1999) CAFASP-1: critical assessment of fully automated structure prediction methods. *Proteins Suppl* 3:209–217



Chapter 5

StackSSSPred: A Stacking-Based Prediction of Supersecondary Structure from Sequence

Michael Flot, Avdesh Mishra, Aditi Sharma Kuchi, and Md Tamjidul Hoque

Abstract

Supersecondary structure (SSS) refers to specific geometric arrangements of several secondary structure (SS) elements that are connected by loops. The SSS can provide useful information about the spatial structure and function of a protein. As such, the SSS is a bridge between the secondary structure and tertiary structure. In this chapter, we propose a stacking-based machine learning method for the prediction of two types of SSSs, namely, β -hairpins and β - α - β , from the protein sequence based on comprehensive feature encoding. To encode protein residues, we utilize key features such as solvent accessibility, conservation profile, half surface exposure, torsion angle fluctuation, disorder probabilities, and more. The usefulness of the proposed approach is assessed using a widely used threefold cross-validation technique. The obtained empirical result shows that the proposed approach is useful and prediction can be improved further.

Key words Supersecondary structure prediction, Beta-hairpins, Beta-alpha-beta, Stacking, Machine learning, Sequence-based prediction

1 Introduction

A protein macromolecule is a linear chain of amino acid residues linked together by peptide bonds. Protein structure can be described in terms of four different hierarchies of structural and folding patterns-based complexities: *a primary structure* is the sequence of amino acid chain only that makes up a polypeptide chain without any structural information; *secondary structure* concerns regular, repeated local three-dimensional (3D) segments of proteins including α -helix and β -strand; *tertiary structure* is the global 3D structure of a protein molecule; and *quaternary structure* describes the way in which the different tertiary subunits are packed together to form the structure of a protein complex [1]. The *supersecondary structures* bridge the secondary structure and the tertiary

Michael Flot and Avdesh Mishra contributed equally to this work.

structure of a protein. Secondary structure elements connected by a polypeptide (loop) in specific geometric arrangements are called motifs or supersecondary structures [2]. These supersecondary structures (SSSs) can provide information about the spatial structure of a protein. Some of the most commonly occurring SSSs include α -helix hairpins, β -hairpins, β - α - β , coiled coils, Greek key, α -loop- α , α -turn- α , and Rossmann motifs. Accurate knowledge of 3D structure of a protein provides insight on a protein's function, which is crucial in effective design and development of drug.

The classic work of Anfinsen, in the 1950s, on the enzyme ribonuclease revealed the relation between the amino acid sequence of a protein and its conformation. Through his experiments, Anfinsen showed that the information needed for a protein to obtain its 3D structure is contained in its amino acid sequence. Nevertheless, prediction of 3D structure of protein from sequence remains as one of the greatest challenge for the scientific community [3, 4]. Investigators are exploring two fundamentally different approaches of predicting the 3D structure from amino acid sequence. The first is ab initio or de novo protein structure prediction (*aiPSP*), which attempts to build the structure from the sequence of amino acid residues without prior knowledge about similar sequences in known protein structure database [5–12]. Computational methods have employed that attempt to minimize the free energy of a structure with a given sequence or to simulate the folding process. Molecular dynamics (MD) is an example of the ab initio method that performs simulation of the protein folding process. MD has been successfully applied for the prediction of small proteins and peptides as well as for the refinement of the structures (both small and large proteins) by minimizing the energy, to some extent [13, 14]. The second approach is dependent on the availability of similar templates in the protein database and is commonly known as homology modelling [15–19]. Amino acid sequence of a known structure or fragments is scanned for sequence similarity with the sequence of the target protein with unknown structures, and if a significant match is detected, the known structural knowledge is applied to construct the final model. Moreover, the prediction of tertiary structure of a protein can also be achieved by proceeding in a hierachal fashion. First, the secondary structure of the protein is predicted from the amino acid sequence, then the supersecondary structures are derived from the secondary structure elements, and finally the information about the secondary and SSSs is used to computationally determine the 3D shape of the protein molecule [19–24].

The past decade has witnessed tremendous progress in the development of accurate predictors of secondary structure. Some of the recent and successful predictors of secondary structures include SSpro [25], Spider 3 [26], Spider 2 [27], and SPINE X [28]. As reported, SSpro achieved the highest accuracy of 92.9% for secondary structure prediction by combining sequence similarity

and sequence-based structural similarity. In addition to 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 properties of proteins. These properties include the prediction of RNA-binding proteins [29], DNA-binding protein and their binding sites [30], protein-peptide interactions [31], protein-carbohydrate interaction [32], residue contacts [8, 33], disorder region [34, 35], accessible surface of amino acids [36], target selection for structural genomics [37, 38], and more.

In the past, many attempts have been made in predicting individual SSSs types and several effective computational prediction methods have been proposed in the literature for analyzing them, such as β hairpins [39, 40], β - α - β [2, 41], coiled coils [42, 43], and helix-turn-helix motifs [44–46]. Many of the SSS prediction methods capitalize on the fact that the prediction of secondary structure provides useful information for the prediction of SSS [2, 47]. Predicted SSSs are useful features for various applications, such as simulation of protein folding [48], analysis of the relation between coiled coils and disorder regions [49], study and identification of many functional and active sites [2], analysis of amyloids [50], genome-wide studies of protein structure [51, 52], and prediction of protein domains [53].

In this chapter, we present a machine learning (ML) approach for the prediction of SSSs directly from the sequence of amino acids instead of following the traditional hierarchical approach of first predicting the secondary structures and then utilizing the predicted SS types (labels) to predict the SSSs. We implement several ML methods along with a recently studied [31, 54] Stacking-based ML predictor for two different types of supersecondary structures β -hairpins and β - α - β . The Stacking-based ML approach combines the information of several different ML algorithms to generate a new prediction model. It provides a scheme for minimizing the generalization error rate of one or more predictive models. The utility of the proposed approach is fast assessed by threefold cross-validation approach. The results obtained from extensive examination shows that the proposed approach is time-consuming, yet very promising. Along with detailed methodology and explanation of required tools, techniques, and resources, we provide useful notes to assist readers with the process of improving the prediction accuracy of the proposed method.

2 Materials

In this section, we describe the procedure for benchmark dataset preparation, tools necessary for class label assignment, aggregation and encoding of input features, machine learning algorithms, and the criteria to evaluate them.

2.1 Dataset

We built up a benchmark dataset [55] of protein sequences collected from the protein data bank [56] using an *Advanced Search* interface with the following specifications: (1) experimental method, X-ray; (2) molecule type, protein; (3) X-ray resolution, $< 1.5 \text{ \AA}$; (4) chain length, ≥ 40 ; and (5) sequence identity (cutoff), 30%. This resulted in 3474 proteins. The chains of these proteins were split into separate structures, and the sequence from these single-chain structures were extracted resulting in 5388 different sequences. To reduce bias from too many similar sequences, BLASTCLUST [57] was used to reduce sequence similarity to 25%. Keeping just the first of each cluster reduced the number of sequences to 3349.

Furthermore, we discarded the protein sequences with unknown amino acid, labelled with “X” character, because of the unavailability of the corresponding features. Structures with unknown coordinates of amino acids were removed as well, because the corresponding supersecondary structure of the amino acids could not be obtained. Moreover, to train the ML algorithms, several tools were used to generate features from the sequence (see Subheading 2.3). For some of the sequences, these tools failed to generate useful information. Such sequences were discarded from further consideration. We also discarded sequences where we found that the length of a sequence given by the tool’s output and the length of a FASTA sequence provided by the collected PDB files differed. Finally, this reduced the number of sequences to 3203.

In addition, if none of the amino acid residues in a protein sequence were labelled as either β -hairpin or $\beta\text{-}\alpha\text{-}\beta$, such sequences were discarded from their respective benchmark dataset. As a result, the β -hairpin dataset contains 2520 proteins, and the $\beta\text{-}\alpha\text{-}\beta$ dataset contains 1208 proteins.

2.2 Assignment of Supersecondary Structures

The SSS is composed of two secondary structure units connected by a polypeptide (loop) with a specific arrangement of geometry. Among more than a dozen types of the SSSs, the β -hairpins, coiled coils, α -turn- α , and $\beta\text{-}\alpha\text{-}\beta$ motifs received more attention due to the fact that they are present in a large number of protein structures and play an important role in many biological activities. In this study, we focus on the study of β -hairpins and $\beta\text{-}\alpha\text{-}\beta$ motifs. The second largest group of protein domains is the β -hairpins. They are found in diverse protein families, including enzymes, transporter proteins, antibodies, and viral coats [47]. The β -hairpin motif consists of two strands that are adjacent in primary structure, oriented in an anti-parallel direction, and linked by a short loop of two to five amino acids. On the other hand, $\beta\text{-}\alpha\text{-}\beta$ is a complex supersecondary structure in proteins and often appears in *Bacillus subtilis* proteases [58]. The study of $\beta\text{-}\alpha\text{-}\beta$ motifs is important because many functional as well as active sites often occur in the polypeptide of $\beta\text{-}\alpha\text{-}\beta$ motifs, including ADP-binding sites, FAD-binding sites,

NAD-binding sites, and more [59]. In this work, we used the PROMOTIF [60] program to generate annotations (labels) for two types of supersecondary structures β -hairpin and β - α - β predictors. PROMOTIF is a program like DSSP [61] as it uses the distances and hydrogen bonding between residues to assign supersecondary structures. The single-chain protein structures are passed to the PROMOTIF program to obtain the information about the residues which belong to β -hairpin or β - α - β motifs. Based on the outcome of the PROMOTIF program, if the residue belongs to the β -hairpin or β - α - β motif, the residue is labelled as “1” else “0,” respectively.

2.3 Feature Extraction

Feature extraction and encoding is an important step in the development of machine learning-based predictors. To create an effective machine learning-based method to predict β -hairpin and β - α - β motifs from sequence alone, we use various sequence and structure-based features. These features provide information about the chemical, structural, and flexibility profiles of the proteins. A set of features used in this study are listed in Table 1 and are briefly discussed below.

1. *Amino acid (AA)*: Twenty different standard amino acids were encoded using 20 different integers ranging from 1 to

Table 1
List of features used in SSS prediction

Feature category	Features count
Amino acid (AA)	1
Physiochemical properties (PP)	7
Position-specific scoring matrix	20
Secondary structure probabilities	6
Accessible surface area	1
Torsion angle (φ , ψ) fluctuation	2
Monogram	1
Bigram	20
Position-specific estimated energy	1
Terminal indicator (TI)	1
Disorder probability	1
Phi and psi torsion angles	2
Half sphere exposures	2
Total	65

20, which is a useful feature to capture the amino acid composition.

2. *Physicochemical properties (PP)*: Seven different physicochemical properties per amino acid, namely, steric parameter, polarizability, normalized van der Waals volume, hydrophobicity, isoelectric point, and helix and sheet probabilities, were collected from DisPredict2 [35] program. These features were originally reported in [62].
3. *Position-specific scoring matrix (PSSM)*: PSSM captures the conservation pattern using multiple sequence alignments and stores this pattern as a matrix of scores for each position in the alignment. High scores in this matrix represent more conserved positions, and scores close to zero or negative represent weakly conserved position. Thus, PSSM provides the evolutionary information in proteins. Evolutionary information is one of the most important kinds of information for protein functionality prediction in biological analysis and is widely used in such studies [34, 36, 63–66]. We executed three iterations of PSI-BLAST [67] against NCBI’s nonredundant database to generate PSSM of size sequence length × 20, which gave us 20 features per residue.
4. *Monogram (MG) and bigram (BG)*: The monogram (single feature) and bigram (20 features) were computed from PSSM by further extending the PSSM values to higher dimension. Both of these features were collected from DisPredict2 program. These features are found to be useful in protein fold recognition [68, 69] and various other applications such as disordered prediction [35] and protein-peptide binding [31].
5. *Local structural properties*: We collected a total of eleven predicted local structural features, which include three secondary structures probabilities for helix (H), beta (B), and coil (C) obtained from MetaSSpred [64] and three additional SS probabilities obtained from Spider 3 [26]; two torsion angles, phi (Φ) and psi (Ψ); one accessible surface area (ASA); and two half sphere exposure (HSE), namely, HSE-up and HSE-down. The torsion angles and HSE features were predicted using Spider 3 program. ASA was predicted using DisPredict2 which generates this feature from Spine X [28].
6. *Flexibility properties*: We include multiple flexibility properties of amino acids, which include two torsion angle fluctuations, dphi ($\nabla\Phi$) and dpsi ($\nabla\Psi$), and one disorder probability. The torsion angle features can be originally predicted using DAVAR [70]; however, all the above features were extracted from DisPredict2.
7. *Energy features*: Since many functional sites and active sites often occur in the polypeptide of β - α - β motifs, they play a

significant role in binding. The binding of protein and ligand involves formation and dissolution of atomic interactions that require change in free energy [71]. Thus, to capture the state of free energy contribution of residues, we include position-specific estimated energy (PSEE) which was also predicted using DisPredict2.

8. *Terminal region:* Often terminal residues of a protein show higher flexibility. Thus, to distinguish the terminal residues from others, we included terminal indicator feature by encoding five residues of N-terminal as $[-1.0, -0.8, -0.6, -0.4, -0.2]$ and C-terminal as $[+1.0, +0.8, +0.6, +0.4, +0.2]$, respectively, whereas the rest of the residues were labelled as 0.0.

Before using the features mentioned above into the classifier, different sized sliding windows were evaluated. This technique is used to incorporate neighboring information for each residue. Sliding windows work by aggregating information on both sides of the target residue. For example, if window size 11 is chosen, the target residue will be at 6th location with 65 features, and for 5 residues before and 5 residues after the target residue will have (10×65) or, 650 features, totaling $(65 + 650)$ or, 715 features per residue.

2.4 Machine Learning Algorithms

In this study, we explored five different potential machine learning algorithms for the prediction of two types of SSSs: β -hairpin and β - α - β . The implemented algorithms are briefly discussed below. All of the classifiers used in our study are built and tuned using scikit-learn [72].

1. *K Nearest Neighbor (KNN) Classifier:* The KNN algorithm compares an input to the K closest training examples [73]. A majority vote coming from the most similar neighbors in the training set decides the classification. We used Euclidean distance as a metric for finding the nearest neighbors. As the idea of learning a model using KNN is simple, this method is computationally cheap. For all our experiments with KNN method, the value of K was set to 7 and all the neighbors were weighted uniformly.
2. *Extra Tree (ET) Classifier:* The extremely randomized tree or ET [74] is one of the ensemble methods, which constructs randomized decision trees from the original learning sample and uses above-average decision to improve the predictive accuracy and control over-fitting. We constructed the ET model with 1000 trees, and the quality of a split was measured by Gini impurity index.
3. Gradient Boosting Classifier (GBC): The GBC works by combining weak learners into a single learner in an iterative fashion

[75]. We applied 1000 boosting stages where a regression tree was fit on the negative gradient of the deviance loss function. In our implementation, the learning rate was set to 0.1 and the maximum depth of each regression tree was set to 3. GBC overcomes over-fitting with higher number of boosting stages, and we observed that 1000 stages were giving competitive performance for this application.

4. *Logistic Regression (LogReg)*: We used LogReg [76] with L2 regularization for the prediction of SSSs. LogReg measures the relationship between the dependent variable, which is categorical (in our case: whether an amino acid belongs to SSSs type or not), and one or more independent variables by generating an estimation probability using logistic regression. It utilizes the sigmoid function to predict the output [77].
5. *Random Decision Forest (RDF)*: The RDF [77, 78] operates by constructing a multitude of decision trees, each of which is trained on a random sub-sample of the training data. The sub-sample used for creating a decision tree is constructed from a given set of observation of training data by taking n observations at random and with replacement. This technique of sub-samples creation is also known as Bootstrap sampling. Then, the predictions from individual decision trees are aggregated to provide a single prediction. For classification, the single prediction is made by computing the mode (the value that appears most often) of the classes (in our proposed case: whether an amino acid belongs to SSSs type or not). We used bootstrap samples to construct 1000 trees in the forest.

2.5 Performance Metrics

To build the proof-of-concept of Stacking versus non-Stacking approach fast, we used threefold cross-validation (FCV) [76, 79, 80] to compare and evaluate the performance of each predictor. FCV is performed in folds, where the data is divided into m parts, which are each of about equal size. While a fold is set aside for testing, the other ($m - 1$) folds are used to train the classifier. This process is repeated until each fold has been set aside once for testing and then the m estimates of error are combined to find the average. We employed various performance measures listed in Table 2 to test the predictive ability of various predictors. The majority of the metrics listed in the table are computed from the true positive (TP), false positive (FP), true negative (TN), and false negative (FN) metrics. TP refers to the number of instances that are correctly predicted as positive. FP refers to the number of instances that are incorrectly predicted as positive. TN refers to the number of instances that are correctly labelled as negative. FN refers to the number of instances that are incorrectly labelled as negative. Recall is defined as proportion of real positive cases that are correctly predicted positive. Similarly, precision is defined as proportion of predicted positive cases that are correctly real positives. Likewise,

Table 2
Name and definition of the evaluation metric

Name of metric	Definition
True positive (TP)	Correctly predicted supersecondary structures
True negative (TN)	Correctly predicted non-supersecondary structures
False positive (FP)	Incorrectly predicted supersecondary structures
False negative (FN)	Incorrectly predicted non-supersecondary structures
Recall/sensitivity	$True\ Positive\ Rate\ (TPR) = \frac{TP}{TP+FN}$
Specificity	$True\ Negative\ Rate\ (TNR) = \frac{TN}{FP+TN}$
Fallout (or overprediction) rate	$False\ Positive\ Rate\ (FPR) = \frac{FP}{FP+TN}$
Miss rate	$False\ Negative\ Rate\ (FNR) = \frac{FN}{FN+TP}$
Accuracy (ACC)	$\frac{TP+TN}{FP+FP+TN+FN}$
Balanced accuracy (mean of specificity and recall)	$\frac{1}{2} \left(\frac{TP}{TP+FN} + \frac{TN}{TN+FP} \right)$
Precision	$\frac{TP}{TP+FP}$
F1 score (harmonic mean of precision and recall)	$\frac{2TP}{2TP+FP+FN}$
Matthews correlation coefficient (MCC)	$\frac{(TP \times TN) - (FP \times FN)}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$

F1 score is defined as the harmonic mean of recall and precision. The miss rate and fallout rate measure two complementary types of incorrect predictions. The miss rate is defined as proportion of real positive cases that occur as predicted negative. Similarly, fallout rate is defined as proportion of real negative cases that are correctly predicted positive. Furthermore, *Matthews correlation coefficient* (MCC) measures the degree of overlap between the predicted labels and true labels of all the samples in the benchmark dataset. Lastly, the balanced accuracy is defined as mean of recall and specificity.

3 Methods

In this section, we discuss the implementation of Stacking-based machine learning approach for the prediction of two types of supersecondary structures: β -hairpin and β - α - β only from the sequence. We first discuss the results of individual classifiers and, subsequently, report the performance of various stacked predictors based on the benchmark dataset.

3.1 Stacking Framework

We applied the Stacking technique [81] to deal with the sequence-based supersecondary structure prediction problem. Stacking is an ensemble technique which minimizes the generalization error by combining information from multiple predictive models to generate a new model. Because Stacking minimizes the generalization error rate of one or more predictive models, it has been successfully applied in several ML tasks [82–86] and recently has been shown to work well with the prediction of protein-peptide binding sites [31] and prediction of DNA-binding proteins [54].

The Stacking method uses a two-tier learning framework. The first (i.e., base) tier consists of a collection of classifiers called base-learners. In the second (i.e., meta) tier, the outputs of the base-level learners are combined with the original input vector and fed to another classifier called a meta-learner. This method considers the fact that different base-learners can react to certain regions of the feature space poorly due to the no-free-lunch theorem [87]. Thus, using meta-learner in the second tier, the outputs of the base classifiers are combined with the aim of reducing the generalization error. For better performance, it is desirable to choose classifiers those are highly uncorrelated to each other [31] or are different from each other based on their underlying operating principle as the base classifiers [54]. As Stacking combines the outputs from the first tier in the second tier, this makes the Stacking technique different from other ensemble methods like bagging and boosting as these techniques apply weighted average or majority vote to form a final prediction.

The base and meta-classifiers used in the Stacking framework for this experiment include (a) Logistic Regression (LogReg) [23, 76], (b) Extra Trees (ET) [74], (c) Random Decision Forest (RDF) [78], (d) K Nearest Neighbor (KNN) [73], and (e) Gradient Boosting Classifier (GBC) [75]. These algorithms and their configurations are briefly discussed in Subheading 2.4. For each algorithm, feature window size which results in best accuracy was identified, and then, the classifiers with their respective best window sizes were used in the Stacking framework.

In our implementation of Stacking framework, we explored four different classifiers KNN, ET, GBC, and RDF as both meta- and base classifiers. While one of the four methods was used as the meta-learner, the rest of the methods were used as the base-learners. We dropped LogReg classifier out from Stacking because it took longer to train this classifier on our benchmark dataset. The combinations of the stacked model (SM) separately assessed for both β -hairpin and β - α - β in this study are:

1. **SM1:** includes KNN, GBC, and RDF as base-learners and ET as meta-learner.
2. **SM2:** includes ET, GBC, and RDF as base-learners and KNN as meta-learner.

3. **SM3:** includes ET, KNN, and RDF as base-learners and GBC as meta-learner.
4. **SM4:** includes ET, KNN, and GBC as base-learners and RDF as meta-learner.

The output probabilities (probability p belonging to β -hairpin or $\beta\text{-}\alpha\text{-}\beta$ and probability $(1 - p)$ not belonging to β -hairpin or $\beta\text{-}\alpha\text{-}\beta$) generated by the respective base classifiers are combined with the original windowed feature vector to train a new meta-classifier. In our implementation, we found that a window size of 11 gives the highest performance for each of the classifier. Thus, in Stacking all the base-learners were trained using best window size feature vector of (65×11) or 715 and the meta-learners were trained using best window size features plus six additional probability features, resulting into feature vector of $(65 \times 11 + 6)$ or 721. The general framework of our Stacking-based predictor is shown in Fig. 1.

3.2 Results

In this section, we first present the results obtained from best window size selection experiment and then provide the comparative results of individual classifiers obtained on best window size and, subsequently, report the performance of the stacked predictors on the benchmark dataset.

3.2.1 Window Selection

In this experiment, we searched for a suitable size of the sliding window (W) that determines the number of residues around a target residue, which could belong to SSS types of either β -hairpin or $\beta\text{-}\alpha\text{-}\beta$. To select the optimal window size, we designed five different models using GBC classifier with five different window sizes: 1, 3, 7, 11, and 13. The models were trained and validated using threefold cross-validation on the benchmark dataset. Figure 1 illustrates the overall accuracy obtained for all the window sizes for both $\beta\text{-}\alpha\text{-}\beta$ and β -hairpin SSS types while using GBC classifier.

From Fig. 2, we observed that the overall accuracy of the model increased drastically from window size 1 to 11, whereas, after window size 11, the increment is only after two decimal places. Thus, we selected window size of 11 as the best window size. All the methods used in the Stacking were trained on window size 11 feature vector.

3.2.2 Analysis and Evaluation of Individual Machine Learning Algorithms

Here, we analyze the performance of five individual classifiers, Log-Reg, ET, KNN, RDF, and GBC. The performance metrics of the classifiers were obtained by performing threefold cross-validation. The predicted annotations of every residue were compared against the actual annotations obtained from the PROMOTIF program.

From Table 3, we observe that the GBC gives an outstanding balanced as well as overall accuracy compared to other methods for

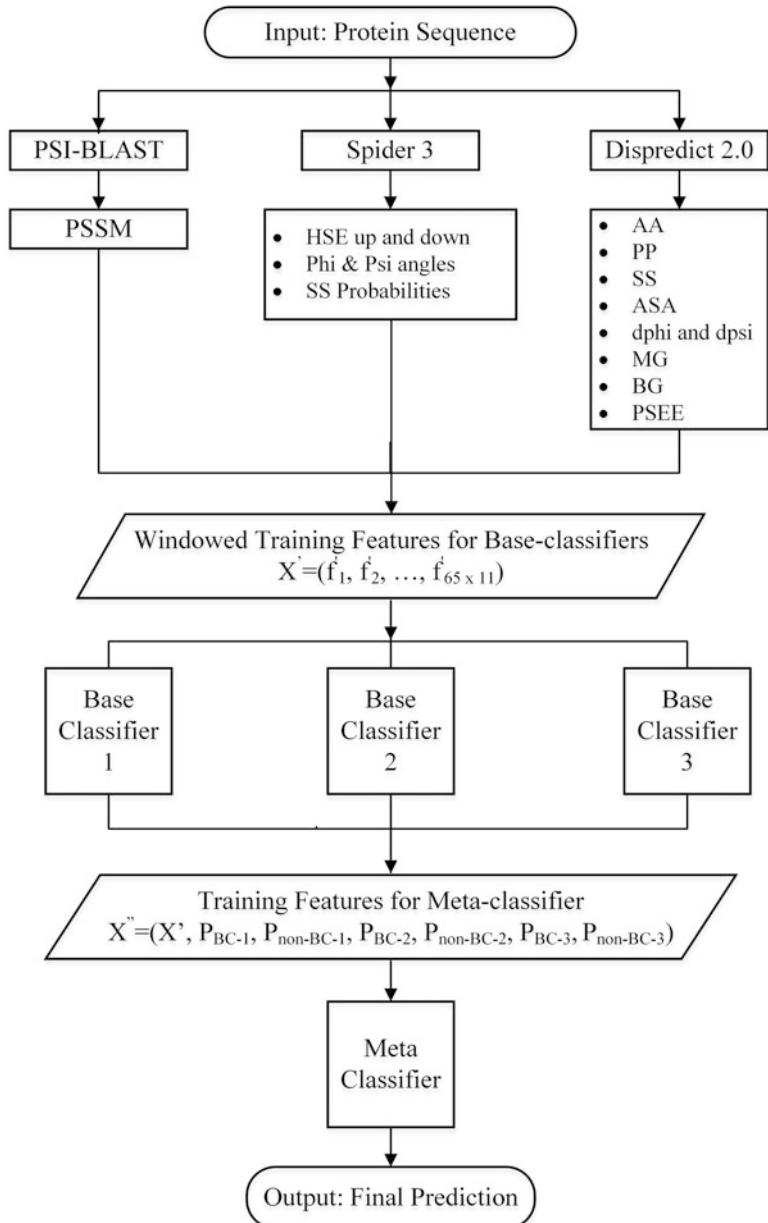


Fig. 1 Flowchart describing the Stacking prediction framework. This framework was applied for both β -hairpin and β - α - β separately

the prediction of beta-alpha-beta SSS. The ET method resulted in the best recall or sensitivity of 0.705 and FNR or miss rate of 0.295. However, based on the rest of the performance measures, ET performed less accurately than the GBC. In addition, based on specificity, balanced accuracy, overall accuracy, FPR, precision, F1 score, and MCC, the GBC outperformed other methods. It is also evident that the RDF is the second-best method based on balanced

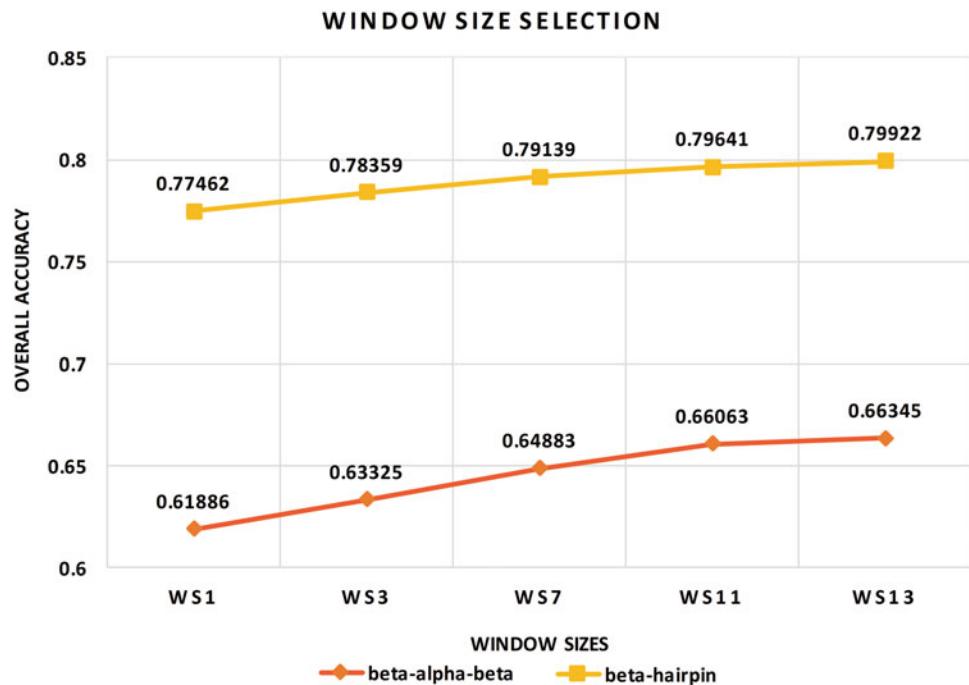


Fig. 2 Performance comparison of different window sizes using GBC method for both beta-alpha-beta and beta-hairpin SSS types. The accuracies of different window sizes are compared and used to decide the best window size

Table 3

Comparison of individual ML methods on predicting β - α - β SSS on the benchmark dataset using threefold cross-validation and feature vector with window size 11

Metric/method	LogReg	ET	KNN	RDF	GBC
Sensitivity	0.679	0.705	0.603	0.704	0.695
Specificity	0.576	0.595	0.550	0.608	0.626
Balanced accuracy	0.627	0.650	0.577	0.656	0.661
Overall accuracy	0.627	0.650	0.577	0.656	0.661
FPR/fallout rate	0.424	0.405	0.450	0.392	0.374
FNR/miss rate	0.321	0.295	0.397	0.296	0.305
Precision	0.615	0.635	0.573	0.642	0.650
F1 score	0.645	0.668	0.588	0.672	0.672
MCC	0.256	0.302	0.154	0.313	0.322

Bold indicates best score

Table 4

Comparison of individual ML methods on predicting β -hairpin SSS on the benchmark dataset using threefold cross-validation and feature vector with window size 11

Metric/method	LogReg	ET	KNN	RDF	GBC
Sensitivity	0.990	0.992	0.947	0.989	0.972
Specificity	0.141	0.179	0.174	0.196	0.259
Balanced accuracy	0.566	0.585	0.560	0.592	0.616
Overall accuracy	0.781	0.791	0.756	0.793	0.796
FPR/fallout rate	0.859	0.821	0.826	0.804	0.741
FNR/miss rate	0.010	0.008	0.053	0.011	0.028
Precision	0.779	0.787	0.778	0.790	0.800
F1 score	0.872	0.877	0.854	0.878	0.878
MCC	0.282	0.335	0.190	0.344	0.358

Bold indicates best score

Table 5

Results of various stacked models for the prediction of β - α - β SSS

Method/ metric	Sensitivity	Specificity	Balanced accuracy	Overall accuracy	Fallout rate	Miss rate	Precision	F1 score	MCC
SM1	0.709	0.609	0.659	0.659	0.391	0.291	0.644	0.675	0.319
SM2	0.641	0.583	0.612	0.612	0.417	0.359	0.606	0.623	0.224
SM3	0.693	0.620	0.657	0.657	0.380	0.307	0.646	0.669	0.314
SM4	0.707	0.610	0.658	0.658	0.391	0.293	0.644	0.674	0.318

Bold indicates best overall accuracy

and overall accuracy. Likewise, ET stands third, LogReg stands fourth, and KNN is the least performing method based on the balanced and overall accuracies.

From Table 4, GBC provides an outstanding balanced and overall accuracy compared to other methods for the prediction of β -hairpin SSS. The ET method gave best recall or sensitivity of 0.991 and FNR of 0.010. However, based on the rest of the performance metrics, ET performed less accurately than the GBC. In addition, except the performance metrics sensitivity, FNR, and F1 score, the GBC showed better performance than other methods based on the rest of the measurements. Furthermore, we can see that RDF is the second-best method based on balanced and overall accuracy. Likewise, ET stands third, LogReg stands fourth, and KNN is the least performing method based on the balanced and overall accuracies.

Table 6
Results of various stacked models for the prediction of β -hairpin SSS

Method/ metric	Sensitivity	Specificity	Balanced accuracy	Overall accuracy	Fallout rate	Miss rate	Precision	F1 score	MCC
SM1	0.984	0.221	0.603	0.796	0.779	0.016	0.794	0.879	0.355
SM2	0.953	0.243	0.597	0.777	0.758	0.047	0.793	0.866	0.285
SM3	0.965	0.269	0.617	0.793	0.731	0.035	0.801	0.876	0.348
SM4	0.977	0.242	0.610	0.796	0.758	0.023	0.797	0.878	0.355

Bold indicates best overall accuracy

Table 7
Comparison of overall accuracies obtained by stacked models and the individual methods for the prediction of β - α - β SSSs

Machine learning method	Individual	Meta-learner
ET	0.650	0.659
KNN	0.577	0.612
GBC	0.661	0.657
RDF	0.656	0.658

Next, we selected four of the ML techniques including GBC, RDF, KNN, and ET to use in our Stacking approach. While one method was selected as the meta-learner, the others were selected as the base-learners. Table 5 shows the performance comparison of the combination of stacked models SM1, SM2, SM3, and SM4 used for the prediction of β - α - β SSSs using threefold cross validation on benchmark dataset. It can be seen from the table that the SM1, which consists of ET as meta-learner, provides the highest overall accuracy of 0.659 followed by SM4, SM3, and SM2, respectively. Furthermore, except SM2 the overall accuracies of all other stacked models are close to each other with differences after two decimal places.

Similarly, Table 6 shows the performance comparison of the combination of stacked models SM1, SM2, SM3, and SM4 used for the prediction of β -hairpin SSSs using threefold cross-validation on benchmark dataset. It can be observed from Table 6 that the SM4, which consists of RDF as meta-learner, provides the highest overall accuracy of 0.796 followed by SM1, SM3, and SM2, respectively. Furthermore, except SM2 the overall accuracies of all other stacked models are close to each other with differences after two decimal places.

Table 8
Comparison of overall accuracies obtained by stacked models and the individual methods for the prediction of β -hairpin SSSs

Machine learning method	Individual	Meta-learner
ET	0.791	0.796
KNN	0.756	0.777
GBC	0.796	0.793
RDF	0.793	0.796

Next, in Table 7, we show the comparison of overall accuracies achieved by individual methods with the accuracies obtained while the respective methods were used as the meta-learner in the Stacking framework for the prediction of β - α - β SSSs. It is evident from the table that, while the methods are used as the meta-learner in stacking, they yield better accuracy compared to while the respective methods are used independently. For example, while KNN was separately used for the prediction of β - α - β SSSs, we achieved an overall accuracy of 0.577 or 57.67%. However, while KNN was used as the meta-learner, we achieved an overall accuracy of 0.612 or 61.2%, which is 6% higher than while KNN was used separately. This indicates that the Stacking-based methods can be useful in predicting the supersecondary structures.

Similarly, in Table 8, we show the comparison of overall accuracies achieved by individual methods with the accuracies obtained while the respective methods were used as the meta-learner in the Stacking framework for the prediction of β -hairpin SSSs. Table 8 also shows that Stacking yielded better accuracy compared to individual methods for the prediction of β -hairpins except for Stacking model in which GBC was used as a meta-learner. In case of GBC, the accuracy seems to slightly decrease. This decrease in accuracy is negligible however and occurs after the second decimal place. Moreover, the results for all other cases indicate that Stacking resulted in better accuracy compared to the individual methods in this study.

4 Notes

1. The Stacking-based machine learning predictors have been utilized in various bioinformatics applications [31, 54, 84, 86, 88]. Among others, Iqbal et al. recently proposed a Stacking framework, called PBRpredict-Suite, to predict peptide-binding residues of receptor proteins from sequence [31]. They first compared six predictors to find the best

predictor (SVM) and the two predictors least correlated with it (GBC and KNN) to use as base-learners. These base-learners' probability outputs were then used to train a logistic regression-based meta-learner. As reported, PBRpredict-Suite provides the best accuracy of 80.4% for the prediction of protein-peptide binding residues. Moreover, very recently, Mishra et al. applied Stacking to develop a predictor, called StackDPPred, to predict DNA-binding proteins from sequence [54]. They combined machine learning methods SVM, Logistic Regression, KNN, and RDF which are different from each other based on their underlying operating principle at the base layer. Next, to enrich the meta-learner (SVM), the original feature vector was combined with the base predictor probabilities. As reported, StackDPPred provided an accuracy of over 89% on benchmark dataset and an accuracy of 86.5 and 85.95% on two different independent test datasets, respectively.

2. Stacking-based predictors developed recently [31, 54] use SVM either as a base-learner or as the meta-learner, whereas in this application of supersecondary structure prediction, we were unable to use SVM because of the time constraint as it took longer to train. The SVM has been proven to be a very useful machine learning algorithm for various bioinformatics applications [32, 35, 36, 89]. Therefore, using SVM as either a base-learner or meta-learner could significantly improve the accuracy of supersecondary structure prediction. We propose to use SVM as a learner in our Stacking application in our future work on supersecondary structure prediction. SVM is a fast learner; however, its optimization using grid search for RBF kernel parameters is often found impractically slow, especially for the larger dataset.
3. Feature ranking and selection techniques have also been successfully applied in numerous applications including bioinformatics to improve the accuracy of the machine learning-based predictors [29, 32, 90]. Identifying relevant features and removing unimportant or redundant features can reduce computation time and improve results. In our future work, we will implement various feature ranking and selection techniques to improve the accuracy of our current supersecondary structure prediction approach.
4. Here, we present a review of some recently developed supersecondary structure prediction methods. In one of the recent work, Sun et al. developed a predictor which uses statistical approach and SVM to predict β - α - β motifs [2] from sequence but uses predicted secondary structure labels to predict β - α - β motifs. Similarly, Jia et al. proposed a predictor which also uses statistical approach and RDF to predict β - α - β motifs [41]. In this work, the authors used DSSP and PROMOTIF software to

obtain the secondary structure and supersecondary structure labels. Additionally, they performed a statistical analysis on β - α - β and non- β - α - β motifs and only selected the motifs that contain loop-helix-loop length from 10 to 26 amino acids. One major difference between these approaches and the method proposed in this study is that we predict any length of β - α - β motifs, while other methods only select the motifs that contain loop-helix-loop length from 10 to 26.

5. For intermediate steps, a more reliable SS prediction could improve the accuracy of SSS prediction: instead of utilizing single SS assignment method such as DSSP [61], the consensus-based SS label assignment would generate better SS assignment. For example, utilizing consensus of DSSP [61], STRIDE [91], and KASKI [92] methods, where final class label is generated based on majority vote from the assignment of these programs, could improve the SS assignment which could subsequently lead to better SSS prediction. This leverages the fact that each assignment software approaches this problem in different ways. Combining multiple methods should theoretically correct inaccuracies that can come from using single-label generation method.
6. In real-world applications, computing resources and computing time are important factors in deciding which machine learning algorithm to use. It is common to weigh predictive accuracy against computational time to decide which method to use. In our application, we found that the SVM and LogReg were the two methods which took longer to run. Thus, due to time constraints, we discarded these methods for the prediction of supersecondary structures in this work. We look forward to implementing these methods in our future work on SSS prediction.
7. For the readers' convenience, links to the software necessary for feature and annotation collection are provided below:
 - (a) Dispredict2: <http://cs.uno.edu/~tamjid/Software.html>
 - (b) PSI-BLAST: <https://blast.ncbi.nlm.nih.gov/Blast.cgi>
 - (c) Spider 3: <http://sparks-lab.org/server/SPIDER3/>
 - (d) PROMOTIF: <http://www.img.bio.uni-goettingen.de/ms-www/internal/manuals/promotif/promotif.html>
8. While predictors for a few major class of SSSs have been proposed and tested, many have not been approached due to the limited scope of this article. Some interesting structures for further research are listed below:
 - (a) α -Helix hairpins
 - (b) Psi loop

- (c) Greek key
 (d) Rossmann motifs
9. For our future work, we intend to improve the accuracy of Stacking-based prediction for supersecondary structures using various feature ranking and selection technique as well as including SVM, LogReg, and other useful machine learning methods in our Stacking framework. We also intend to develop Stacking-based machine learning predictors for coiled coil, Psi loop, and other SSS types. Furthermore, we plan to develop a Stacking-based software suit (tool) to predict multiple types of SSSs through a single complex framework. Finally, we will explore deep learning-based techniques for the prediction of supersecondary structures.

Acknowledgment

The authors gratefully acknowledge the Louisiana Board of Regents through the Board of Regents Support Fund LEQSF (2016-19)-RD-B-07.

References

- Chen K, Kurgan L (2012) Computational prediction of secondary and supersecondary structures. In: Kister A (ed) Protein supersecondary structures, vol 932. Humana Press, Totowa, NJ
- Sun L, Hu X, Li S, Jiang Z, Li K (2016) Prediction of complex super-secondary structure $\beta\alpha\beta$ motifs based on combined features. Saudi J Biol Sci 23(1):66–71
- Baker D, Sali A (2001) Protein structure prediction and structural genomics. Science 294 (5540):93–96
- Skolnick J, Fetrow JS, Kolinski A (2000) Structural genomics and its importance for gene function analysis. Nat Biotechnol 18:283–287
- Bhattacharya D, Cao R, Cheng J (2016) UniCon3D: de novo protein structure prediction using united-residue conformational search via stepwise, probabilistic sampling. Bioinformatics 32(18):2791–2799
- Bhattacharya D, Cheng J (2013) i3Drefine software for protein 3D structure refinement and its assessment in CASP10. PLoS One 8(7): e69648
- Bradley P, Misura KMS, Baker D (2005) Toward high-resolution de novo structure prediction for small proteins. Science 309 (5742):1868–1871
- Cao R, Bhattacharya D, Adhikari B, Li J, Cheng J (2015) Large-scale model quality assessment for improving protein tertiary structure prediction. Bioinformatics 31(12): i116–i123
- Jauch R, Yeo HC, Kolatkar PR, Clarke ND (2007) Assessment of CASP7 structure predictions for template free targets. Proteins 69 (S8):57–67
- Klepeis JL, Wei Y, Hecht MH, Floudas CA (2005) Ab initio prediction of the three-dimensional structure of a de novo designed protein: a double-blind case study. Proteins 58(3):560–570
- Liwo A, Khalili M, Scheraga HA (2005) Ab initio simulations of protein-folding pathways by molecular dynamics with the united-residue model of polypeptide chains. Proc Natl Acad Sci U S A 102(7):2362–2367
- Wu S, Skolnick J, Zhang Y (2007) Ab initio modeling of small proteins by iterative TAS-SER simulations. BMC Biol 5:17
- He X, Zhu Y, Epstein A, Mo Y (2018) Statistical variances of diffusional properties from ab initio molecular dynamics simulations. npj Comput Mater 4(1):18. <https://doi.org/10.1038/s41524-018-0074-y>

14. Magnan CN, Baldi P (2015) Molecular dynamics simulations advances and applications. *Adv Appl Bioinforma Chem* 8:37–47
15. Ginalski K, Pas J, Wyrwicz LS, Mv G, Bujnicki JM, Rychlewskia L (2003) ORFeus: detection of distant homology using sequence profiles and predicted secondary structure. *Nucleic Acids Res* 31(13):3804–3807
16. Jones DT (1999) GenTHREADER: an efficient and reliable protein fold recognition method for genomic sequences. *J Mol Biol* 287(4):797–815
17. Karplus K, Barrett C, Hughey R (1998) Hidden Markov models for detecting remote protein homologies. *Bioinformatics* 14 (10):846–856
18. Skolnick J, Kihara D, Zhang Y (2004) Development and large scale benchmark testing of the PROSPECTOR 3.0 threading algorithm. *Proteins* 56:502–518
19. Wu S, Zhang Y (2008) MUSTER: improving protein sequence profile-profile alignments by using multiple sources of structure information. *Proteins* 72(2):547–556
20. Yang Y, Faraggi E, Zhao H, Zhou Y (2011) Improving protein fold recognition and template-based modeling by employing probabilistic-based matching between predicted one-dimensional structural properties of query and corresponding native properties of templates. *Bioinformatics* 27 (15):2076–2082
21. Yang J, Yan R, Roy A, Xu D, Poisson J, Zhang Y (2015) The I-TASSER Suite: protein structure and function prediction. *Nat Methods* 12:7–8
22. Faraggi E, Yang Y, Zhang S, Zhou Y (2010) Predicting continuous local structure and the effect of its substitution for secondary structure in fragment-free protein structure prediction. *Structure* 17(11):1515–1527
23. Szilágyi A, Skolnick J (2006) Efficient prediction of nucleic acid binding function from low-resolution protein Structures. *J Mol Biol* 358(3):922–933
24. Zhou H, Skolnick J (2007) Ab initio protein structure prediction using chunk-TASSER. *Biophys J* 93(5):1510–1518
25. Magnan CN, Baldi P (2014) SSpro/ACCpro 5: almost perfect prediction of protein secondary structure and relative solvent accessibility using profiles, machine learning and structural similarity. *Bioinformatics* 30(18):2592–2597
26. Heffernan R, Yang Y, Paliwal K, Zhou Y (2017) Capturing non-local interactions by long short term memory bidirectional recurrent neural networks for improving prediction of protein secondary structure, backbone angles, contact numbers, and solvent accessibility. *Bioinformatics* 33(18):2842–2849
27. Heffernan R, Paliwal K, Lyons J, Dehzangi A, Sharma A, Wang J, Sattar A, Yang Y, Zhou Y (2015) Improving prediction of secondary structure, local backbone angles, and solvent accessible surface area of proteins by iterative deep learning. *Sci Rep* 5:11476
28. Faraggi E, Zhang T, Yang Y, Kurgan L, Zhou Y (2012) SPINE X: improving protein secondary structure prediction by multi-step learning coupled with prediction of solvent accessible surface area and backbone torsion angles. *J Comput Chem* 33(3):259–267
29. Zhang X, Liu S (2017) RBPPred: predicting RNA-binding proteins from sequence using SVM. *Bioinformatics* 33(6):854–862
30. Chowdhury SY, Shatabda S, Dehzangi A (2017) iDNAProt-ES: identification of DNA-binding proteins using evolutionary and structural features. *Sci Rep* 7:14938
31. Iqbal S, Hoque MT (2018) PBRpredict-Suite: a suite of models to predict peptide-recognition domain residues from protein sequence. *Bioinformatics* 34(19):3289–3299
32. Taherzadeh G, Zhou Y, Liew AW-C, Yang Y (2016) Sequence-based prediction of protein-carbohydrate binding sites using support vector machines. *J Chem Inf Model* 56 (10):2115–2122
33. Eickholt J, Cheng J (2012) Predicting protein residue-residue contacts using deep networks and boosting. *Bioinformatics* 28 (23):3066–3072
34. Iqbal S, Hoque MT (2015) DisPredict: a predictor of disordered protein using optimized RBF kernel. *PLoS One* 10(10):e0141551
35. Iqbal S, Hoque MT (2016) Estimation of position specific energy as a feature of protein residues from sequence alone for structural classification. *PLoS One* 11(9):e0161452
36. Iqbal S, Mishra A, Hoque T (2015) Improved prediction of accessible surface area results in efficient energy function application. *J Theor Biol* 380:380–391
37. Mizianty MJ, Kurgan L (2011) Sequence-based prediction of protein crystallization, purification and production propensity. *Bioinformatics* 27(13):i24–i33
38. Slabinski L, Jaroszewski L, Rychlewski L, Wilson IA, Lesley SA, Godzik A (2007) XtalPred: a web server for prediction of protein crystallizability. *Bioinformatics* 23(24):3403–3405
39. Jia S-C, Hu X-Z (2011) Using random forest algorithm to predict β -hairpin motifs. *Protein Pept Lett* 18(6):609–617

40. 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(3):915–921
41. Sun L, Hu X (2013) Recognition of beta-alpha-beta motifs in proteins by using Random Forest algorithm. Paper presented at the sixth International Conference on Biomedical Engineering and Informatics, Hangzhou, China
42. Mahrenholz CC, Abfalter IG, Bodenhofer U, Volkmer R, Hochreiter S (2011) Complex networks govern coiled-coil oligomerization—predicting and profiling by means of a machine learning approach. *Mol Cell Proteomics* 10(5): M110.004994
43. Bartoli L, Fariselli P, Krogh A, Casadio R (2009) CCHMM_PROF: a HMM-based coiled-coil predictor with evolutionary information. *Bioinformatics* 25(21):2757–2763
44. Pellegrini-Calace M, Thornton JM (2005) Detecting DNA-binding helix-turn-helix structural motifs using sequence and structure information. *Nucleic Acids Res* 33(7):2129–2140
45. Dodd IB, Egan JB (1990) Improved detection of helix-turn-helix DNA-binding motifs in protein sequences. *Nucleic Acids Res* 18(17):5019–5026
46. Ferrer-Costa C, Shanahan HP, Jones S, Thornton JM (2005) HTQuery: a method for detecting DNA-binding proteins with a helix-turn-helix structural motif. *Bioinformatics* 21(18):3679–3680
47. Kumar M, Bhasin M, Natt NK, Raghava GPS (2005) BhairPred: prediction of β -hairpins in a protein from multiple alignment information using ANN and SVM techniques. *Nucleic Acids Res* 33(Web Server issue):W154–W159
48. Sun ZR, Cui Y, Ling LJ, Guo Q, Chen RS (1998) Molecular dynamics simulation of protein folding with supersecondary structure constraints. *J Protein Chem* 17(8):765–769
49. Szappanos B, Süveges D, Nyitrai L, Perczel A, Gáspári Z (2010) Folded-unfolded cross-predictions and protein evolution: the case study of coiled-coils. *FEBS Lett* 584(8):1623–1627
50. O'Donnell CW, Waldspühl J, Lis M, Halfmann R, Devadas S, Lindquist S, Berger B (2011) A method for probing the mutational landscape of amyloid structure. *Bioinformatics* 27(13):i34–i42
51. Rackham OJL, Madera M, Armstrong CT, Vincent TL, Woolfson DN, Gough J (2010) The evolution and structure prediction of coiled coils across all genomes. *J Mol Biol* 403(3):480–493
52. Gerstein M, Hegyi H (1998) Comparing genomes in terms of protein structure: surveys of a finite parts list. *FEMS Microbiol Rev* 22(4):277–304
53. Reddy CC, Shameer K, Offmann BO, Sowdhamini R (2008) PURE: a webserver for the prediction of domains in unassigned regions in proteins. *BMC Bioinformatics* 9:281
54. Mishra A, Pokhrel P, Hoque MT (2018) StackDPPred: a stacking based prediction of DNA-binding protein from sequence. http://cs.uno.edu/~tamjid/TechReport/StackDPPred_TR2018_2.pdf
55. Flot M, Mishra A, Kuchi AS, Hoque MT (2018) Benchmark data for supersecondary structure prediction only from sequence. University of New Orleans. http://cs.uno.edu/~tamjid/Software/StackSSSPred/code_data.zip. Accessed June 2018
56. Berman HM, Westbrook J, Feng Z, Gilliland G, Bhat TN, Weissig H, Shindyalov IN, Bourne PE (2000) The protein data bank. *Nucleic Acids Res* 28(1):235–242
57. Camacho C, Coulouris G, Avagyan V, Ma N, Papadopoulos J, Bealer K, Madden TL (2009) BLAST+: architecture and applications. *BMC Bioinformatics* 10:421
58. Blundell TL, Sibanda BL, Sternberg MJE, Thornton JM (1987) Knowledge-based prediction of protein structures and the design of novel molecules. *Nature* 326:347–352
59. 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(1):101–107
60. Hutchinson EG, Thornton JM (1996) PROMOTIF—a program to identify and analyze structural motifs in proteins. *Protein Sci* 5(2):212–220
61. Kabsch W, Sander C (1983) Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers* 22:2577–2637
62. Meiler J, Müller M, Zeidler A, Schmäschke F (2001) Generation and evaluation of dimension-reduced amino acid parameter representations by artificial neural networks. *J Mol Model* 7:360–369
63. Biswas AK, Noman N, Sikder AR (2010) Machine learning approach to predict protein phosphorylation sites by incorporating evolutionary information. *BMC Bioinformatics* 11:273
64. Islam N, Iqbal S, Katebi AR, Hoque MT (2016) A balanced secondary structure predictor. *J Theor Biol* 389:60–71

65. Kumar M, Gromiha MM, Raghava GP (2007) Identification of DNA-binding proteins using support vector machines and evolutionary profiles. *BMC Bioinformatics* 8:463
66. Verma R, Varshney GC, Raghava GPS (2010) Prediction of mitochondrial proteins of malaria parasite using split amino acid composition and PSSM profile. *Amino Acids* 39(1):101–110
67. Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ (1990) Basic local alignment search tool. *J Mol Biol* 215:403–410
68. Paliwal KK, Sharma A, Lyons J, Dehzangi A (2014) A tri-gram based feature extraction technique using linear probabilities of position specific scoring matrix for protein fold recognition. *IEEE Trans Nanobioscience* 13(1):44–50
69. Sharma A, Lyons J, Dehzangi A, Paliwal KK (2013) A feature extraction technique using bi-gram probabilities of position specific scoring matrix for protein fold recognition. *J Theor Biol* 320:41–46
70. Zhang T, Faraggi E, Zhou Y (2010) Fluctuations of backbone torsion angles obtained from NMR-determined structures and their prediction. *Proteins* 78:3353–3362
71. London N, Movshovitz-Attias D, Schueler-Furman O (2010) The structural basis of peptide-protein binding strategies. *Structure* 18(2):188–199
72. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E (2011) Scikit-learn: machine learning in Python. *J Mach Learn Res* 12:2825–2830
73. Altman NS (1992) An introduction to kernel and nearest-neighbor nonparametric regression. *Am Stat* 46:175–185
74. Geurts P, Ernst D, Wehenkel L (2006) Extremely randomized trees. *Mach Learn* 63 (1):3–42
75. Friedman JH (2002) Stochastic gradient boosting. *Comput Stat Data Anal* 38 (4):367–378. [https://doi.org/10.1016/S0167-9473\(01\)00065-2](https://doi.org/10.1016/S0167-9473(01)00065-2)
76. Hastie T, Tibshirani R, Friedman J (2009) The elements of statistical learning, Springer series in statistics, 2nd edn. Springer, New York
77. Freedman DA (2009) Statistical models: theory and practice. Cambridge University Press, Cambridge
78. Ho TK (1995) Random decision forests. Paper presented at the Document Analysis and Recognition, 1995. Proceedings of the Third International Conference, Montreal, Quebec, Canada
79. Duda RO, Hart PE, Stork DG (2000) Pattern classification. Wiley, Hoboken, NJ
80. Bishop C (2009) Pattern recognition and machine learning. Information science and statistics. Springer, New York
81. Wolpert DH (1992) Stacked generalization. *Neural Netw* 5(2):241–259
82. Frank E, Hall M, Trigg L, Holmes G, Witten IH (2004) Data mining in bioinformatics using Weka. *Bioinformatics* 20(15):2479–2481
83. Ginsburga GS, McCarthy JJ (2001) Personalized medicine: revolutionizing drug discovery and patient care. *Trends Biotechnol* 19 (12):491–496
84. Nagi S, Bhattacharyya DK (2013) Classification of microarray cancer data using ensemble approach. *Netw Model Anal Health Inform Bioinform* 2(3):159–173
85. Hu Q, Merchante C, Stepanova AN, Alonso JM, Heber S (2015) A stacking-based approach to identify translated upstream open reading frames in *Arabidopsis Thaliana*. Paper presented at the International Symposium on Bioinformatics Research and Applications
86. Verma A, Mehta S (2017) A comparative study of ensemble learning methods for classification in bioinformatics. Paper presented at the seventh International Conference on Cloud Computing, Data Science & Engineering—Confluence, Noida, India
87. Wolpert DH, Macready WG (1997) No free lunch theorems for optimization. *IEEE Trans Evolut Comput* 1(1):67–82
88. Frank E, Hall M, Trigg L, Holmes G, Witten IH (2004) Data mining in bioinformatics using Weka. *Bioinformatics* 20:2479–2481
89. Guruge I, Taherzadeh G, Zhan J, Zhou Y, Yang Y (2018) B-factor profile prediction for RNA flexibility using support vector machines. *J Comput Chem* 39:407–411
90. Anne C, Mishra A, Hoque MT, Tu S (2018) Multiclass patent document classification. *Artif Intell Res* 7(1):1
91. Heinig M, Frishman D (2004) STRIDE: a web server for secondary structure assignment from known atomic coordinates of proteins. *Nucleic Acids Res* 32(Web Server issue):W500–W502
92. Martin J, Letellier G, Marin A, Taly J-F, AGD B, Gibrat J-F (2005) Protein secondary structure assignment revisited: a detailed analysis of different assignment methods. *BMC Struct Biol* 5:17



Chapter 6

Information-Theoretic Inference of an Optimal Dictionary of Protein Supersecondary Structures

Arun S. Konagurthu, Ramanan Subramanian, Lloyd Allison, David Abramson, Maria Garcia de la Banda, Peter J. Stuckey, and Arthur M. Lesk

Abstract

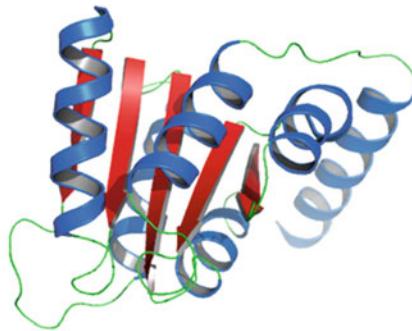
We recently developed an unsupervised Bayesian inference methodology to automatically infer a dictionary of protein supersecondary structures (Subramanian et al., IEEE data compression conference proceedings (DCC), 340–349, 2017). Specifically, this methodology uses the information-theoretic framework of minimum message length (MML) criterion for hypothesis selection (Wallace, Statistical and inductive inference by minimum message length, Springer Science & Business Media, New York, 2005). The best dictionary of supersecondary structures is the one that yields the most (lossless) compression on the source collection of folding patterns represented as tableaux (matrix representations that capture the essence of protein folding patterns (Lesk, *J Mol Graph.* 13:159–164, 1995)). This book chapter outlines our MML methodology for inferring the supersecondary structure dictionary. The inferred dictionary is available at <http://lcb.infotech.monash.edu.au/proteinConcepts/scop100/dictionary.html>.

Key words Minimum message length, MML, Tableau representation, Protein folding pattern, Supersecondary structure

1 Introduction

The tableau representation introduced by Lesk [1] provides a compact and powerful way to capture the essence of protein folding patterns (*see* Fig. 1). A tableau is a two-dimensional representation of a folding pattern constructed from the protein coordinate data, capturing (*see* Fig. 1b):

1. The order of secondary structural elements along the protein chain, represented as a string over a {‘H’ = helix, ‘E’ = strand} alphabet.
2. The relative orientation of every pair of secondary structural elements, represented by a symmetric matrix of angles within the range $(-180^\circ, 180^\circ]$.

a**b**

	E1										
H2	-164.6°	H2									
E3	-20.5°	-168.4°	E3								
H4	-169.1°	25.5°	152.7°	H4							
E5	-13.9°	151.3°	-34.2°	-169.2°	E5						
H6	128.1°	-67.0°	108.7°	-44.0°	141.7°	H6					
E7	-53.3°	112.4°	-73.5°	133.1°	-39.4°	-171.6°	E7				
H8	126.3°	-69.0°	107.8°	-44.9°	139.5°	-5.6°	166.2°	H8			
E9	-26.8°	137.8°	-45.6°	161.6°	-14.3°	154.3°	-29.0°	153.1°	E9		
H10	147.8°	-47.3°	131.7°	-21.8°	157.9°	-25.7°	148.5°	-24.8°	170.4°	H10	
H11	58.9°	-131.4°	38.7°	-117.2°	71.9°	-75.1°	109.4°	-76.2°	80.4°	-100.7°	

Fig. 1 (a) Cartoon representation of the ASTRAL SCOPe (v2.05) domain (chain A) of Ribose-5-phosphate isomerase B from *Clostridium thermocellum* (SCOP domain ID: d3he8a_). Secondary structures were assigned using SST [2]. Helices are shown in blue, while strands of sheet are shown in red. (b) The tableau representation of d3he8a_ capturing the order of helices (H) and strands (E) numbered, in order, from 1 to 11, the pairwise orientation angles of the secondary structural elements (shown as a lower triangular matrix), and the corresponding contact information (shown as entries of the matrix with bold font angles, when the corresponding secondary structural elements are in contact)

3. The interactions between pairs of secondary structural elements, represented by a binary contact matrix.

Supersecondary structures arise out of contacts between a local assembly of secondary structure elements in the amino acid sequence. A sub-tableau compactly captures the geometry of any supersecondary structure. Thus, this representation provides a mathematical and computable definition for any supersecondary structure, thereby allowing us to utilize this representation in our lossless-compression-based inference methodology to identify a dictionary of supersecondary structures automatically [3]. This inference procedure is outlined below (*see Subheading 3*).

2 Materials

1. A source collection containing 51,368 domains from ASTRAL SCOP (v2.05) database [4]. No two domains in our collection share the same amino acid sequence.

2. SST [2] is used to assign secondary structure and construct tableau representations.
3. The inference method is implemented in the C++ programming language and parallelized using OpenMPI.

3 Methods

3.1 Construction of Tableau

The tableau representation of a given protein structure is constructed from its 3D coordinate data as follows [1, 5–7]:

1. Identification of the secondary structural elements: The 3D protein coordinates are delineated into secondary structural elements (*see Note 1*). All identified helices are labeled “H” not distinguishing α , 310, and π helices (or their handedness). Strands of sheet are labeled “E.” This yields a string of secondary structural elements (*see Fig. 1b*).
2. Identification of relative orientation between secondary structural elements: For every pair of secondary structural elements identified, their relative orientation is specified as an angle $-180^\circ < \Omega \leq 180^\circ$ (*see Fig. 1b*), computed as follows:
 - (a) Represent each of the two secondary structural elements as a vector. For a helix, this vector corresponds to the helical axis. For a strand, it is represented by the least-squares line through its C_α atoms. Each vector is directed from the N-terminus to C-terminus of the secondary structural element.
 - (b) Compute the mutual perpendicular between these two vectors.
 - (c) Compute the relative orientation angle as the shortest rotation about the mutual perpendicular computed above (*see Note 2*).
3. Identification of contacts between secondary structural elements: A tableau representation also records the contact information between secondary structures. Two secondary structural elements are defined to be in contact if at least one pair of residues between the elements is in contact. Two residues are said to be in contact if there exists at least one pair of atoms between the two residues that is 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 3*).

3.2 The “Concept” of Supersecondary

Structure

1. A tableau is constructed for each domain in our source database (*see Subheading 2*).
2. This yields a source collection of tableaux, denoted by the set: $T = \{\tau_1, \tau_2, \dots, \tau_{|T|}\}$ of size $|T|$.

3. In this work, a candidate supersecondary structure takes the form of a contiguous sub-tableau, referred to as a (candidate) concept.
4. A concept, denoted by c , can be instantiated by selecting a source tableau $\tau_x \in T$ and specifying a continuous range of indices $[i, j]$, such that $1 \leq i < j \leq |\tau_x|$, and each secondary structural element in this range has at least one contact with other elements in the range $[i, j]$.

3.3 The Candidate Dictionary of “Concepts”

1. A candidate dictionary of supersecondary structural concepts is denoted by the set $C = \{c_1, c_2, \dots, c_{|C|}\}$.
2. A candidate dictionary can contain any number of concepts ($|C|$), where each concept (i.e., sub-tableaux) $c_y \in C$ is composed of an arbitrary number of secondary structural elements ($|c_y| \geq 2$).
3. In this framework, any dictionary that can be constructed from the source collection T , is a potential candidate to compress losslessly (i.e., explain) the entire source collection.

3.4 Minimum Message Length (MML) Methodology

3.4.1 Bayesian and Information-Theoretic Framework

1. Let H be a hypothesis on some observed data D . From Bayes theorem (expressed via the product formula of probabilities), we know that $\Pr(H \& D) = \Pr(H) \Pr(D | H) = \Pr(D) \Pr(H | D)$. See Note 4.
2. Independently, from Shannon’s information content [8], denoted by $I(E)$, we know that the measure (in bits) of the shortest lossless code to represent any event E is $I(E) = -\log_2(\Pr(E))$.
3. Translating Bayes theorem into terms involving Shannon’s information content (by multiplying both sides of the Bayes theorem by $-\log_2(\Pr(.))$), we get: $I(H \& D) = I(H) + I(D | H) = I(D) + I(H | D)$. See Note 5.
4. This can be understood as an information communication process between an imaginary transmitter-receiver pair [9, 10]:
 - (a) The transmitter wants to send the observed data D over a two-part losslessly encoded message to the receiver.
 - (b) In the first part of the message, the transmitter encodes and communicates the hypothesis H to the receiver, taking $I(H)$ bits to state that hypothesis.
 - (c) In the second part, the transmitter communicates the details of the observed data D given the stated hypothesis H , taking $I(D | H)$ bits to state those details.
 - (d) Thus, the best hypothesis in this framework is the one that yields the shortest two-part lossless message to communicate the observed data D losslessly over the hypothesis H .

3.4.2 Formulating the Problem of Inference of the Dictionary of Supersecondary Structural Concepts in the Language of MML

1. In the context of the MML framework [9, 10], a candidate dictionary C (*see* Subheading 3.3) is seen as an attempt to explain (lossless compress) the source collection of tableaux T (*see* Subheading 3.2). That is, in this problem, C forms the hypothesis H , while D forms the observed data T .
2. Formally, for any candidate dictionary C and source collection T , the two-part message length (in bits) is $I(C \& T) = I(C) + I(T|C)$.
3. In the MML framework, this two-part message length is compared against the (single-part) null model message length, which constitutes the baseline. The null model message length is the length of the message required to encode the observed data as is (uncompressed), without the support of any hypothesis. The null model message length is denoted as $I_{\text{null}}(T)$.
4. Thus, the quality of any inferred dictionary C of concepts is measured as the lossless compression obtained by encoding the source collection T using C , compared against the baseline null model message length. That is, the compression gained = $I_{\text{null}}(T) - I(C \& T)$.
5. This yields an inference problem where the best dictionary C^* is the one that minimizes the two-part message length (which is same as the one that maximizes the total lossless compression gained).
6. Therefore, the best dictionary C^* achieves an implicit trade-off between the first part (the complexity of C^*) and the second part (the fidelity of how well C^* explains the source data T).
7. If the two-part message length is worse than the null model length (i.e., $I_{\text{null}}(T) - I(C \& T) \leq 0$), the dictionary C is rejected.
8. Addressing this inference problem requires the construction of:
 - (a) A method to estimate the null model message length, $I_{\text{null}}(T)$, for any given collection T (*see* Subheading 3.4.3).
 - (b) A method to estimate the dictionary model encoding length, $I(C \& T)$ for any given dictionary C and collection T (*see* Subheading 3.4.4).
 - (c) A search method for an optimal dictionary, that is, one that maximizes compression, as per the above stated objective (*see* Subheading 3.4.5).

3.4.3 Computation of $I_{\text{null}}(T)$

1. Our null encoding of the source collection T of tableaux contains:
 - (a) An encoding of the size of the source collection, $|T|$, over a lossless variable length integer code [10].

- (b) The null encoding of the information in each tableau $\tau_x \in T$ (see Note 6).
2. Thus, using this encoding, the message length involved in the null encoding is of the form
- $$I_{\text{null}}(T) = I_{\text{integer}}(|T|) + \sum_{k=1}^{|T|} I_{\text{null}}(\tau_x).$$

3.4.4 Computation of $I(C \& T)$

1. $I(C \& T) = I(C) + I(T|C)$ —see Subheading 3.4.2.
2. $I(C)$ captures the lossless encoding length of all concepts in any stated dictionary C . (Subheading 3.4.5 discusses the search of an optimal dictionary C^* for a given source collection of tableaux T .)
 - (a) Each concept $c_y \in C$ is a sub-tableau (see Subheading 3.2).
 - (b) Encoding of each c_y uses the null encoding of any tableau (see Subheading 3.4.3).
3. The $I(T|C)$ term captures the lossless encoding length of any given source collection T being explained by the stated dictionary of concepts C .
4. Specifically, the $I(T|C)$ term contains:
 - (a) An encoding of the size of the source collection, $|T|$, over a lossless variable length integer code [10].
 - (b) The sum of the encoding lengths required to explain each tableau $\tau_x \in T$ using C .
 - (c) Thus, $I(T|C) = I_{\text{integer}}(|T|) + \sum_{k=1}^{|T|} I(\tau_x|C)$.
5. To compute the $I(\tau_x|C)$ term in the above equation:
 - (a) Each tableau τ_x is partitioned into nonoverlapping regions of variable sizes (see Fig. 2).
 - (b) Any partition of $p(\tau_x)$ specifies an increasing sequence of integer indices $1 \equiv z_0 < z_1 < \dots < z_{|p(\tau_x)|} \equiv \tau_x | + 1$. Each successive pair of indices (z_i, z_{i+1}) defines a nonoverlapping region in the tableau τ_x (see Fig. 2 and Note 7).
 - (c) Each nonoverlapping region is a sub-tableau τ_x . Thus, each such region can be assigned to either the null model or to any $c_y \in C$ if the secondary structural string defining c_y is the same as the string of secondary structures in that region of τ_x .
 - (d) When the region is assigned to the null model, the sub-tableau information of that region of τ_x is explained using the null encoding (see Subheading 3.4.3).
 - (e) When the region is assigned to some concept $c_y \in C$, the sub-tableau information of that region of τ_x is explained as deviations from the corresponding (sub-)tableau information of concept c_y (see Subheading 3.2).

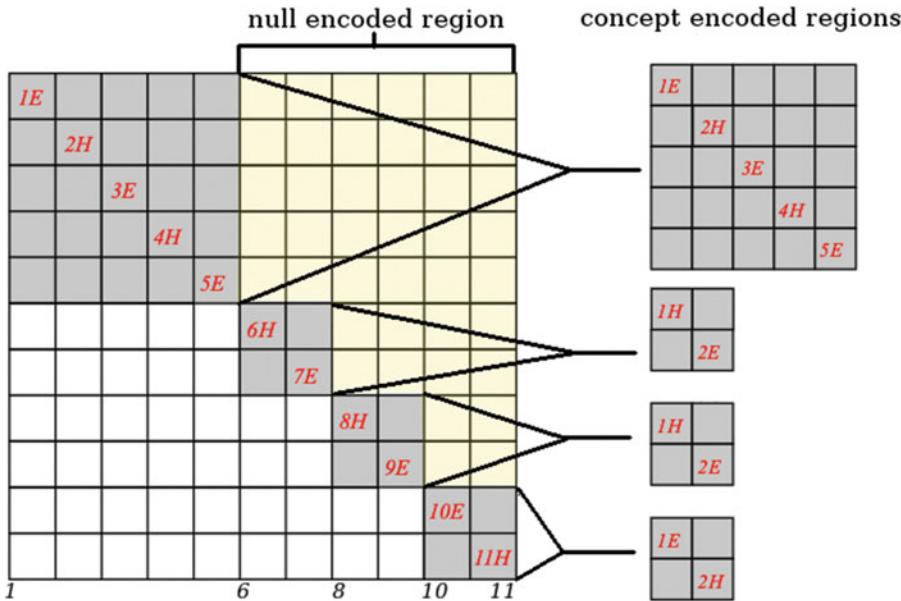


Fig. 2 An illustration of a candidate partition of a tableau of size 11 into 4 nonoverlapping regions. The regions (1,5), (6,7), (8,9), and (10,11) are shown in gray color. Each region defines a sub-tableau assigned to any one of the concepts in the stated dictionary C of similar size and secondary structural string (or can potentially be assigned to a null concept). The remaining parts of the tableau (shown in light yellow) are stated using the null encoding method. If the assigned concepts fit the regions well, their lossless encoded length becomes short. The best partition that minimizes the statement length over all possible partitions on any tableau can be solved using a one-dimensional dynamic programming approach

- (f) Thus, the collection of such assigned nonoverlapping regions defines a block along the main diagonal of tableau τ_x . Regions outside these blocks are defined using the null encoding (see Fig. 2).
- (g) A (best) dissection of any tableau τ_x using any stated dictionary of concepts C that minimizes the lossless encoding length of explaining τ_x using the above method of partitioning can be computed using a one-dimensional dynamic programming approach (see Note 6).

3.4.5 Searching for the Best Dictionary of Secondary Structural Elements

1. A simulated annealing (SA) heuristic is used to search for a dictionary of concepts that maximizes the $I_{\text{null}}(T) - I(C \& T)$ objective.
2. The search space is the set of all possible candidate dictionaries that could be derived from the source collection T .
3. The method starts from an empty dictionary state, where each tableau $\tau_x \in T$ is losslessly encoded using a null model, without any support of the concepts in the dictionary.
4. Proceeding from this starting dictionary state, the heuristic iteratively explores the local neighborhood of the current

dictionary state (C_{current}) using the following perturbation primitives:

- (a) Add concept: Create a concept randomly derived from the source collection (by randomly choosing first a $\tau_x \in T$, and then a $c_y \in C$ from τ_x), and add c_y into C_{current} .
 - (b) Remove concept: Randomly choose a concept from C_{current} , and remove that concept.
 - (c) Perturb concept length: Choose a concept randomly from C_{current} and extend/shorten it, in the context of its loci within its source tableau.
 - (d) Perturb concept flexibility: Increase/decrease the current value of the statistical parameter controlling the flexibility/plasticity of a randomly chosen concept in C_{current} (see **Note 6**).
 - (e) Swap concept with usage: Choose a concept randomly from C_{current} , and swap it with a randomly chosen region in the collection that is currently encoded by that concept within the source collection T .
5. At each iteration of SA, one of the above five perturbations is chosen uniformly at random, to yield a perturbed dictionary state, $C_{\text{perturbed}}$.
 6. If the two-part message length of $I(C_{\text{perturbed}} \& T)$ decreases compared to that of $I(C_{\text{current}} \& T)$, the perturbation is accepted with a probability of 1. Otherwise, it is accepted with a probability defined by $2^{-\Delta I/\text{temp}}$, under the Metropolis criterion, where:
 - (a) $\Delta I = I(C_{\text{perturbed}} \& T) - I(C_{\text{current}} \& T)$ bits.
 - (b) temp is the temperature parameter used in the cooling schedule defined below.
 7. The cooling schedule involves starting with a temperature of 5000 and decreasing each temperature step by a factor of 0.88.
 8. At each temperature step, 50,000 random perturbations are performed unless the temperature is below 10, where the number of perturbations is increased to 500,000 per temperature step.
 9. When the temperature reaches below 0.1, the search for the dictionary stops and the current state of the dictionary is reported as the best found (see **Note 7**).
1. Lossless compression of a source collection containing 51,368 tableaux (see Subheading 2) using the algorithm described above (see Subheading 3) to infer an optimal dictionary of supersecondary structures described resulted in a dictionary containing 4487 supersecondary structural concepts.

3.4.6 Automatically Inferred Dictionary

2. The complete inferred dictionary is available at: <http://lcb.infotech.monash.edu.au/proteinConcepts/scop100/dictionary.html>.

4 Notes

1. The assignment of secondary structure is carried out using SST [2], a Bayesian method to infer secondary structural assignment of proteins using its 3D coordinate data. Although SST differentiates between 13 types of secondary structures, in the tableau representation, the secondary structural states assigned by SST are reduced to 2 types: helix (of any type) and strand-of-sheet (of any type).
2. Clockwise rotations give positive relative orientation angles and anticlockwise rotations give negative angles.
3. A constant of 1.5 \AA is used in our work.
4. For hypothesis selection problems in the Bayesian framework, a hypothesis is sought that maximizes the joint probability, $\Pr(H \& D)$ (or equivalently maximizes the posterior probability, $\Pr(H | D)$, since $\Pr(D)$ is the same across all hypotheses).
5. Maximizing the joint/posterior probability translates to minimizing the message length $I(H \& D)$.
6. See ref. 3 for full technical details of the entire inference methodology, statistical models used for encoding, and search.
7. The total number of possible partitions for any tableau τ of size n (i.e., containing n secondary structural elements) is $2^n - 1$.

References

1. Lesk AM (1995) Systematic representation of protein folding patterns. *J Mol Graph* 13:159–164
2. Konagurthu AS, Lesk AM, Allison L (2012) Minimum message length inference of secondary structure from protein coordinate data. *Bioinformatics* 28(12):i97–i105
3. Subramanian R, Allison L, Stuckey PJ, Garcia De La Banda M, Abramson D, Lesk AM, Konagurthu AS (2017) Statistical compression of protein folding patterns for inference of recurrent substructural themes. In: IEEE data compression conference proceedings (DCC), pp 340–349
4. Fox NK, Brenner SE, Chandonia JM (2013) SCOPe: Structural Classification of Proteins—extended, integrating SCOP and ASTRAL data and classification of new structures. *Nucleic Acids Res* 42(D1):D304–D309
5. Kamat AP, Lesk AM (2007) Contact patterns between helices and strands of sheet define protein folding patterns. *Proteins* 66:869–876
6. Konagurthu AS, Lesk AM (2010) Cataloging topologies of protein folding patterns. *J Mol Recognit* 23(2):253–257
7. Konagurthu AS, Stuckey PJ, Lesk AM (2008) Structural search and retrieval using a tableau representation of protein folding patterns. *Bioinformatics* 24(5):645–651
8. Shannon CE (1948) A mathematical theory of communication. *Bell Syst Tech J* 27:379–423 and 623–656
9. Wallace CS (2005) Statistical and inductive inference by minimum message length. Springer Science & Business Media, New York
10. Allison L (2018) Coding Ockham’s Razor. Springer, Cham



Chapter 7

Formation of Secondary and Supersecondary Structure of Proteins as a Result of Coupling Between Local and Backbone-Electrostatic Interactions: A View Through Cluster-Cumulant Scope

Adam Liwo, Adam K. Sieradzan, and Cezary Czaplewski

Abstract

The secondary structure of proteins results from both local and long-range interactions, the latter being primarily backbone hydrogen bonding. In this chapter, based on our recent work, we suggest that the striking regularity of secondary structure can be described, in a semi-analytical manner, in terms of Kubo cluster cumulants (corresponding to the expansion of the protein's potential of mean force) that originate from the coupling between the backbone-local and backbone-electrostatic interactions. This finding is illustrated by the analysis of the Protein Data Bank statistics. Examples demonstrating the importance of the coupling terms in coarse-grained treatment of proteins are also presented.

Key words Coarse graining, Potential of mean force, Kubo cluster cumulants, Local and electrostatic interactions, Coupling terms, UNRES force field

1 Introduction

The striking regularity of protein structures has since long inspired scientists to search the forces which determine them. Local interactions mainly of steric nature, categorized into the Ramachandran maps [1] and hydrogen bonding were discovered to determine protein-backbone regularity in early works of Pauling and Corey [2, 3]. About 14 years ago, Banavar and co-workers [4] developed a tube model of polypeptide backbone which, by varying tube thickness, could reproduce the helical and sheet structures, as well as the phase diagrams of these forms. These authors emphasized on three-body local interactions. Niemi and co-workers [5–7] developed a model based on the Landau Hamiltonian, which is also composed of local-interaction terms. Their model can reproduce not only regular but also loop structures. With this approach, by solving the Discrete Nonlinear Schrödinger Equation (DNLSE), it was

possible to identify regular and loop structure as dark-soliton solutions of this equation, and to describe conformational transitions in terms of soliton formation, destruction, and displacement [5–7].

Proper modeling of the interplay between local and backbone-hydrogen-bonding interactions also is crucial in the development of both atomistically-detailed [8] and coarse-grained [9] force fields for protein simulations. While developing our UNRES coarse-grained force field for proteins [10–13], we took much care to account for this balance. Coarse-grained force fields are different from the all-atom ones in that the prototype of a respective effective energy function is the potential of mean force of a system under study, in which the degrees of freedom not taken into account in the model are integrated out [10, 13, 14]. Consequently, the *correlation* or *multibody* terms that couple atomistically-detailed interactions are essential for these force fields to work in the *ab initio* mode (i.e., without using any input from experiment or from structural databases). While working on the development of UNRES, we identified the correlation terms that can be considered “formers” of the regular structures [10, 13]. In what follows we describe briefly the pertinent theory, the correlation terms responsible for regular-structure formation, their manifestation in the statistics derived from the Protein Data Bank and influence on the quality of the simulated structures of proteins.

2 Methods

2.1 Potential of Mean Force of a Coarse-Grained System and Its Expansion

In our approach to coarse-graining [10, 13], we assume that the effective energy function of a system is its potential of mean force (PMF), with all degrees of freedom that are omitted from the coarse-grained model averaged over. These omitted degrees of freedom include solvent degrees of freedom, side-chain rotation angles, and the dihedral angles λ or rotation of the peptide groups about the $C^\alpha \cdots C^\alpha$ virtual bonds [15] (Fig. 1). The solvent degrees of freedom are usually averaged over explicitly by using Monte Carlo (MC) or Molecular Dynamics (MD) simulations or implicitly by using the data from the Protein Data Bank (PDB) [16]. Thus, the variables describing the geometry of the polypeptide-water system are divided into two sets: the *primary* variables (\mathbf{X}), which describe the coarse-grained degrees of freedom, and the *secondary* variables (\mathbf{Y}) that are averaged over. In general, the PMF [$F(\mathbf{X})$] is expressed by Eq. 1

$$F(\mathbf{X}) = -\frac{1}{\beta} \ln \left\{ \frac{1}{V_Y} \int_{\Omega_Y} \exp[-\beta E(\mathbf{X}; \mathbf{Y})] dV_Y \right\} \quad (1)$$

where $V_Y = \int_{\Omega_Y} dV_Y$, $E(\mathbf{X}; \mathbf{Y})$ is the original (all-atom) energy function, Ω_Y is the region of the \mathbf{Y} subspace of variables over

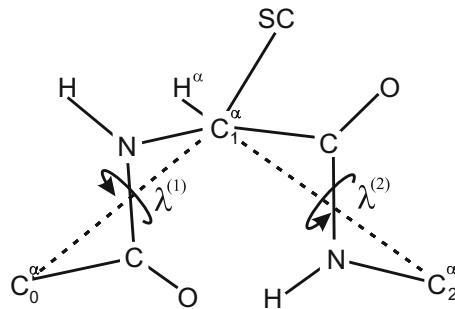


Fig. 1 Definition of the $\lambda^{(1)}$ and $\lambda^{(2)}$ angles for the rotation of the peptide groups about the $C^\alpha \cdots C^\alpha$ virtual-bond axes [15]. $\lambda^{(1)}$ is the angle for counterclockwise rotation of the peptide group located between C_0^α and C_1^α ; $\lambda^{(1)} = 0$ when the carbonyl-carbon atom of the peptide group is in the $C_0^\alpha \cdots C_1^\alpha \cdots C_2^\alpha$ plane and faces C_2^α . $\lambda^{(2)}$ is the angle for counterclockwise rotation of the peptide group located between C_1^α and C_2^α ; $\lambda^{(2)} = 0$ when the amide-nitrogen atom of the peptide group is in the $C_0^\alpha \cdots C_1^\alpha \cdots C_2^\alpha$ plane and faces C_0^α . Reproduced from J. Chem. Phys. 115, 2323 (2001), with the permission of AIP Publishing.

which the integration is carried out, V_Y is the volume of this region, and $\beta = 1/RT$, R being the universal gas constant and T the absolute temperature.

To identify the effective energy terms, the all-atom energy, $E(\mathbf{X}; \mathbf{Y})$ is expressed as a sum of *component energies*, each of which is either the sum of energies *within* a given unit or *between* given units, as given by Eq. 2. Given this partition, the PMF (Eq. 1) is decomposed into *factors*, each of which is a *Kubo's cluster-cumulant function* [17], as expressed by Eq. 3.

$$E(\mathbf{X}; \mathbf{Y}) = \sum_{i=1}^n \varepsilon_i(\mathbf{X}; \mathbf{z}_i) \quad (2)$$

where $\varepsilon_i(\mathbf{X}; \mathbf{z}_i)$ is the i th component energy, \mathbf{z}_i contains the secondary degrees of freedom on which ε_i depends, and n is the number of energy components.

$$\begin{aligned} F(\mathbf{X}) = & \sum_i f_i^{(1)}(\mathbf{X}) + \sum_{i < j} f_{ij}^{(2)}(\mathbf{X}) + \sum_{i < j < k} f_{ijk}^{(3)}(\mathbf{X}) + \dots \\ & + \sum_{i_1 < i_2 < \dots < i_n} f_{i_1, i_2, \dots, i_n}^{(n)}(\mathbf{X}) \end{aligned} \quad (3)$$

The factors are expressed by Eq. 4.

$$\begin{aligned}
f_{i_1 i_2 \dots i_k}^{(k)} &= \langle \langle \varepsilon_{i_1} \varepsilon_{i_2} \dots \varepsilon_{i_k} \rangle \rangle_f = \sum_{l=1}^k \sum_{\substack{i_{m_1} < i_{m_2} < \dots < i_{m_l} \\ m_l \in [1..k]}} (-1)^{k-l} F_{i_{m_1} i_{m_2} \dots i_{m_l}}^{(l)} \\
&= \sum_{l=1}^k \sum_{\substack{i_{m_1} < i_{m_2} < \dots < i_{m_l} \\ m_i \in [1..k]}} (-1)^{k-l} \langle \langle \varepsilon_{i_{m_1}} \varepsilon_{i_{m_2}} \dots \varepsilon_{i_{m_l}} \rangle \rangle
\end{aligned} \tag{4}$$

where

$$\begin{aligned}
F_{i_1, i_2, \dots, i_k}^{(k)}(\mathbf{X}) &\equiv \langle \langle \varepsilon_{i_1} \varepsilon_{i_2} \dots \varepsilon_{i_k} \rangle \rangle \\
&= -\frac{1}{\beta} \ln \left\{ \frac{1}{V_{y_I}} \int_{\Omega_I} \exp \left[-\beta \sum_{l=1}^k \varepsilon_{i_l}(\mathbf{X}; \mathbf{z}_{i_l}) \right] dV_{y_I} \right\}
\end{aligned} \tag{5}$$

(with V_{y_I} being the volume of the subspace spanned by variables $\mathbf{y}_{i_1}, \mathbf{y}_{i_2}, \dots, \mathbf{y}_{i_k}$) is the PMF containing only a subset of component interactions.

The factors of the first order, $f^{(1)}$ correspond to the PMF's of isolated units (such as, e.g., isolated amino-acid residues) or between isolated pairs of units (such as, e.g., pairs of interacting side chains), while factors of order 2 and higher correspond to the correlation (or multibody) terms. All factors depend on temperature, this dependence increasing with increasing the order of a factor because of the increasing order of the first term in generalized-cumulant expansion of this factor [10, 18]. In our approach, as opposed to other coarse-grained force fields, this temperature dependence is explicitly accounted for [18].

The factor expansion is truncated to achieve a compromise between the complexity of the force field and its ability to reproduce the structure and dynamics of the system; for the UNRES force field we found that the fourth-order expansion is sufficient [19]. In the neo-classical force fields, all long-range interactions are approximated by factors of order 1 (i.e., by the potentials of mean forces of isolated pairs of sites), while factors of order 2 only occur in the torsional potentials; these factors account for the coupling between the conformational states of the consecutive polymer units [10, 13]. The best way to obtain approximate analytical formulas for the factors is to use Kubo's generalized cumulant expansion [17], as given by Eq. 6.

$$\begin{aligned}
f_{i_1, i_2, \dots, i_k}^{(k)}(X) &\equiv \langle \langle \varepsilon_{i_1} \varepsilon_{i_2} \dots \varepsilon_{i_k} \rangle \rangle_f \\
&= -\frac{1}{\beta} \sum_{l=k}^{\infty} \sum_{\substack{m_1, m_2, \dots, m_k > 0 \\ m_1 + m_2 + \dots + m_k = l}} (-1)^l \langle \varepsilon_{i_1}^{m_1} \varepsilon_{i_2}^{m_2} \dots \varepsilon_{i_k}^{m_k} \rangle_c
\end{aligned} \tag{6}$$

where $\langle \epsilon_{i_1}^{m_1}, \epsilon_{i_2}^{m_2}, \dots, \epsilon_{i_k}^{m_k} \rangle_c$ is the generalized cumulant corresponding to factor $f_{i_1, i_2, \dots, i_k}^{(k)}$. The generalized cumulants are, in turn, expressed by the mixed energy moments, in which the products of powers of component energies are integrated over the secondary degrees of freedom. For example, for the lowest-order cumulants in the expansion of the first three “mixed” factors we have:

$$\langle \epsilon_i \rangle_c = \langle \epsilon_i \rangle \quad (7)$$

$$\langle \epsilon_i \epsilon_j \rangle_c = \langle \epsilon_i \epsilon_j \rangle - \langle \epsilon_i \rangle \langle \epsilon_j \rangle \quad (8)$$

$$\begin{aligned} \langle \epsilon_i \epsilon_j \epsilon_k \rangle_c &= \langle \epsilon_i \epsilon_j \epsilon_k \rangle - [\langle \epsilon_i \rangle \langle \epsilon_j \epsilon_k \rangle + \langle \epsilon_j \rangle \langle \epsilon_i \epsilon_k \rangle + \langle \epsilon_k \rangle \langle \epsilon_i \epsilon_j \rangle] \\ &\quad + 2 \langle \epsilon_i \rangle \langle \epsilon_j \rangle \langle \epsilon_k \rangle \end{aligned} \quad (9)$$

2.2 Energy Expansion in Secondary Degrees of Freedom

The appropriate expression for the energy in the degrees of freedom that are integrated out when passing to the coarse-grained representation is critical for the correctness of the derived formulas for the factors. Recently [13], we developed a general theory for the derivation of energy moments for the coarse-grained models of polymers, which is based on the fact that, in the absence of external potentials, the energy depends only on interatomic distances. The square of the distance between atom i of site I and atom j of site J is, in turn, expressed by the angles of collective rotation of the atoms about the virtual-bond axes, the distance between the centers, and the geometric parameters defining the orientation of the axes, as given by Eq. 10 [13]

$$\rho_{Ii;Jj}^2 = R_{IJ}^2 + d_{IiJj} + f_{IiJj}(\lambda_J) + f_{JjIi}(\lambda_I) - g_{IiJj}(\lambda_I, \lambda_J) \quad (10)$$

with

$$\begin{aligned} d_{IiJj} &= \delta_{Ii}^2 + \delta_{Jj}^2 + \epsilon_{Ii}^2 + \epsilon_{Jj}^2 - 2R_{IJ} \left[\delta_{Ii} \cos \theta_{IJ}^{(1)} - \delta_{Jj} \cos \theta_{IJ}^{(2)} \right] \\ &\quad - 2\delta_{Ii} \delta_{Jj} \cos \theta_{IJ}^{(12)} \end{aligned} \quad (11)$$

$$\begin{aligned} f_{IiJj}(\lambda_J) &= 2\epsilon_{Jj} \left[R_{IJ} \sin \theta_{IJ}^{(2)} \cos (\lambda_J + \varphi_{Jj} - \psi_{IJ}) \right. \\ &\quad \left. - \delta_{Jj} \sin \theta_{IJ}^{(12)} \cos (\lambda_J + \varphi_{Jj} - \Psi_{IJ}) \right] \end{aligned} \quad (12)$$

$$\begin{aligned} f_{JjIi}(\lambda_I) &= 2\epsilon_{Ii} \left[R_{JI} \sin \theta_{JI}^{(2)} \cos (\lambda_I + \varphi_{Ii} - \psi_{JI}) \right. \\ &\quad \left. - \delta_{Ii} \sin \theta_{JI}^{(12)} \cos (\lambda_I + \varphi_{Ii} - \Psi_{JI}) \right] \\ &= -2\epsilon_{Ii} \left[R_{JI} \sin \theta_{JI}^{(1)} \cos (\lambda_I + \varphi_{Ii} - \psi_{JI}) \right. \\ &\quad \left. + \delta_{Ii} \sin \theta_{JI}^{(12)} \cos (\lambda_I + \varphi_{Ii} - \Psi_{JI}) \right] \end{aligned} \quad (13)$$

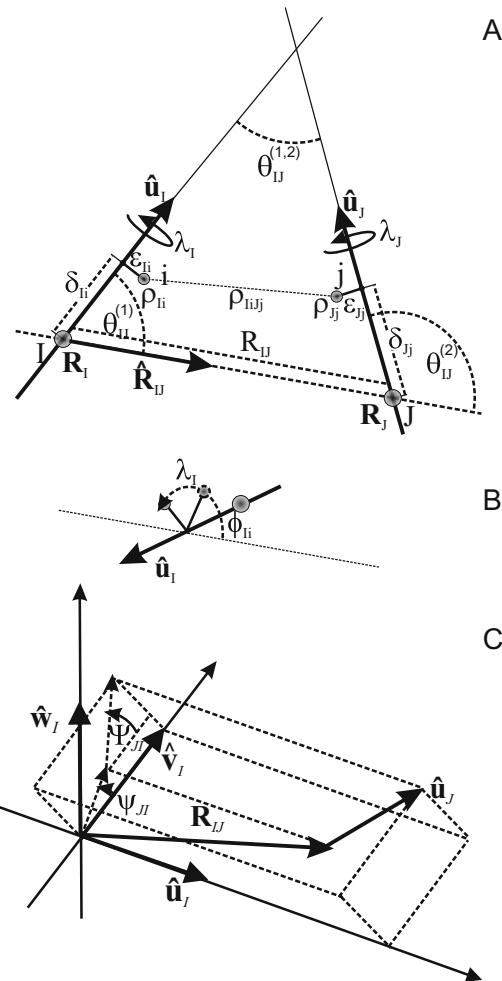


Fig. 2 Illustration of the definitions of the geometric parameters of Eqs. 11–14 to express the distance $\rho_{ii,jj}$ between atoms i with Cartesian coordinates ρ_{ii} and j with Cartesian coordinates ρ_{jj} of coarse-grained sites i (with Cartesian coordinates \mathbf{R}_i) and J (with the Cartesian coordinates \mathbf{R}_j), respectively (indicated in panel **a**). The centers of the sites are shown as larger shaded spheres and the atoms are shown as smaller shaded spheres in panels **(a)** and **(b)**. **(a)** Location of the sites in space. R_{IJ} is the distances between the centers of sites I and J , $\hat{\mathbf{u}}_i$ and $\hat{\mathbf{u}}_j$ are the unit vectors of the virtual-bond axes of the sites, and $\hat{\mathbf{R}}_{IJ}$ is the unit vector pointing from site I to site J . The angles $\theta_{IJ}^{(1)}$, $\theta_{IJ}^{(2)}$ and $\theta_{IJ}^{(12)}$ are the angles between vectors $\hat{\mathbf{u}}_i$ and $\hat{\mathbf{R}}_{IJ}$ between $\hat{\mathbf{u}}_j$ and $\hat{\mathbf{R}}_{IJ}$, and between $\hat{\mathbf{u}}_i$ and $\hat{\mathbf{u}}_j$, respectively. The projection of the position of atom i on the virtual-bond axis of site I is δ_{ii} and can be positive or negative depending on whether the atom is to the left or to the right of the center, and that of atom i is δ_{jj} . The distance of atom i from the virtual-bond axis is ε_{ii} and that of atom j is ε_{jj} . The angles λ_i and λ_j are the angles for counterclockwise rotation about the respective virtual-bond axes. **(b)** Illustration of the phase angle φ_{ii} of atom i of site I this angle defines the counterclockwise rotation of this atom about the virtual-bond axis for $\lambda_i = 0$. **(c)** Illustration of the angles Ψ_{JI} and Ψ_{IJ} , $\hat{\mathbf{u}}_i$ (the unit

$$\begin{aligned} \mathcal{G}_{IiJj}(\lambda_I, \lambda_J) = & \varepsilon_{II}\varepsilon_{JJ} \left\{ \left(1 - \cos \theta_{IJ}^{(12)}\right) \cos \left[(\lambda_I + \varphi_{Ii} - \Psi_{JI}) + (\lambda_J + \varphi_{Jj} - \Psi_{IJ}) \right] \right. \\ & \left. - \left(1 + \cos \theta_{IJ}^{(12)}\right) \cos \left[(\lambda_I + \varphi_{Ii} - \Psi_{JI}) - (\lambda_J + \varphi_{Jj} - \Psi_{IJ}) \right] \right\} \end{aligned} \quad (14)$$

The geometric variables are defined in Fig. 2 and described in detail in ref. 13. With the expansion of the energy in the Taylor series about R_{IJ} (the distance between the coarse-grained-site centers), all energy moments are computed as integrals over the powers of the trigonometric functions that occur in the f s and \mathcal{G} s of Eqs. 12–14, as given by Eq. 15 [13].

$$\begin{aligned} \langle \varepsilon_{k_1}^{i_1} \varepsilon_{k_2}^{i_2} \dots \varepsilon_{k_m}^{i_m} \rangle = & \frac{1}{V_{\mathbf{y}'}} \int_{\mathbf{y}_{I_1}}^{\mathbf{y}_{I_M}} \dots \int_{\mathbf{y}_{I_1}}^{\mathbf{y}_{I_M}} [\mathcal{E}(R_{I_1 J_1}, \dots, R_{I_n J_n}) \\ & + \sum_{\mu=1}^{\infty} \frac{1}{\mu!} \sum_{\nu_1, \nu_2, \dots, \nu_n} \sum_{\substack{k_1, \dots, k_n \\ \nu_1 + \dots + \nu_n = \mu}} \frac{\partial^\mu \mathcal{E}}{\partial (\rho_{I_1 i_{k_1}; J_1 j_{l_1}}^2)^{\nu_1} \dots \partial (\rho_{I_n i_{k_n}; J_n j_{l_n}}^2)^{\nu_n}} \Big|_{R_{I_1 J_1}, \dots, R_{I_n J_n}} \\ & \times \frac{1}{(2\pi)^M} \int_{-\pi}^{\pi} \dots \int_{-\pi}^{\pi} \prod_{m=1}^n d_{I_m i_{k_m} J_m j_{l_m}} + f_{I_m i_{k_m} J_m j_{l_m}}(\lambda_{J_m}) + f_{J_m j_{l_m} I_m i_{k_m}}(\lambda_{I_m}) \\ & \quad [-\mathcal{G}_{I_m i_{k_m} J_m j_{l_m}}(\lambda_{I_m}, \lambda_{J_m})]^{\nu_m} d\lambda_{I_1} \dots d\lambda_{I_M}] dV_{\mathbf{y}'} \end{aligned} \quad (15)$$

where \mathbf{y}_{I_k} denotes all fine-grained degrees of freedom of site I_k except for λ_{I_k} , $\mathbf{y}' = (\mathbf{y}'_{I_1}, \dots, \mathbf{y}'_{I_M})$, \mathcal{E} is the shorthand for $\varepsilon_{i_1}^{k_1} \varepsilon_{i_2}^{k_2} \dots \varepsilon_{i_m}^{k_m}$ and $i_{k_1}, i_{k_2}, \dots, i_{k_n}$ and $j_{l_1}, j_{l_2}, \dots, j_{l_n}$ denote the indices of the atoms. Only those energy moments are not zero, which contain even powers of the cosines of the angles λ . In particular, for mixed moments containing only the first powers of component energies, the terms with $f_{IpJq}f_{JrKs}$ or $f_{IpJq}\mathcal{G}_{JrKs}f_{K,L_u}$ will contribute to the expansion of the PMF. If $J = I + 1$, $K = I + 2$, and $L = I + 3$ these expressions contribute to the torsional and double-torsional potentials, respectively [13].

In order to better understand the meaning of the cumulants, it is advisable to derive approximate expressions to all-atom energy components by using the expression for interatomic distances given by Eqs. 10–14 and, subsequently, to derive the expressions for the moments based on the obtained expansions for the energy components. In our recent work [13], we demonstrated that expanding the electrostatic-interaction energy of two groups, each with zero net charge, up to the second order, results in obtaining the



Fig. 2 (continued) vector of the virtual-bond axis of site I , $\hat{\mathbf{v}}_I$, and $\hat{\mathbf{w}}_I$ are the unit vectors of the axes of the right-handed local Cartesian-coordinate system of site I . Ψ_{JI} and ψ_{JI} are the angles of counterclockwise rotation of the projections of $\hat{\mathbf{u}}_J$ and $\hat{\mathbf{R}}_{IJ}$, respectively, onto the $\hat{\mathbf{v}}_I, \hat{\mathbf{w}}_I$ plane from the $\hat{\mathbf{v}}_I$ axis. Reproduced from J. Chem. Phys. 146, 124106 (2017), with the permission of AIP Publishing

expression for the energy of the interaction of the dipoles of these groups, which rotate about the respective virtual-bond axes.

3 Results

The terms in the cumulant expansion that pertain to the coupling between the backbone-hydrogen-bonding (which are largely of electrostatic nature) and backbone-local interactions can generally be written as $U_{\text{corr}}^{(m)}$, where m is the number of energy components that occur in the respective term [10, 13]. The simplest case is $m = 3$; this means that the electrostatic interactions between two peptide groups are coupled with the local interactions involving two residues (preceding or succeeding in the chain), each involving one of the interacting peptide groups [10, 13]. By combining the coupling terms involving all pertinent local interactions, we obtain Eq. 16 [10, 13].

$$\begin{aligned} U_{\text{loc-el};IJ}^{(3)} &\equiv \langle (e_I + e_{I+1})E_{IJ}(e_J + e_{J+1}) \rangle_f \\ &\approx \frac{\beta^2}{6} [\langle e_I E_{IJ} e_J \rangle_c + \langle e_I E_{IJ} e_{J+1} \rangle_c + \langle e_{I+1} E_{IJ} e_J \rangle_c + \langle e_{I+1} E_{IJ} e_{J+1} \rangle_c] \\ &= \frac{\beta^2}{6R_{IJ}^3} [\mu_I \circ \mu_J - 3(\mu_I \circ \hat{\mathbf{R}}_{IJ})(\mu_J \circ \hat{\mathbf{R}}_{IJ})] \end{aligned} \quad (16)$$

where E_{IJ} is the electrostatic-interaction energy between units I and J ,

$$\begin{aligned} \mu_K &= \mu_K(\gamma_K, \theta_K, \theta_{K+1}) = \begin{pmatrix} \cos \gamma_K & -\sin \gamma_K \\ \sin \gamma_K & \cos \gamma_K \end{pmatrix} \mu_{2K}(\theta_K) + \mu_{1,K+1}(\theta_{K+1}), \\ K &= I, J \end{aligned} \quad (17)$$

where γ_k is the virtual-bond dihedral angle defined by $C_{K-1}^\alpha \cdots C_K^\alpha \cdots C_{K+1}^\alpha \cdots C_{K+2}^\alpha$ atoms, θ_k is the virtual-bond angle defined by the $C_{K-1}^\alpha \cdots C_K^\alpha \cdots C_{K+1}^\alpha$ atoms, and

$$\begin{aligned} \mu_{1K}(\theta_{K+1}) &= \mathbf{b}_{1,K+1}(\theta_{K+1}), \quad \mu_{2K}(\theta_K) \\ &= \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix} \mathbf{b}_{2K}(\theta_K) \end{aligned} \quad (18)$$

where the components of the vectors \mathbf{b}_{1K} and \mathbf{b}_{2K} , $K \in \{I, I+1, J, J+1\}$ (which depend on the respective virtual-bond angles θ) are those of the second-order expansion of the local-interaction-energy surface of a terminally-blocked amino-acid residue; detailed expressions for these components can be found in ref. 13. Thus, the $U_{\text{corr}}^{(3)}$ terms can be represented as the interactions of two fictitious dipoles located on the $C^\alpha \cdots C^\alpha$ virtual-bond axes; these dipoles are vector sums of components from the residues that share a virtual-bond

axes and, thus, depend on the virtual-bond-dihedral angles γ and the adjacent virtual-bond angles θ [13]. This finding entirely agrees with the concept of the interplay between local and hydrogen-bonding interactions in proteins; the Ramachandran maps of amino-acid residues determine the allowed regions of the ϕ and ψ backbone angles, which we convert to the virtual-bond angle θ and the $\lambda^{(1)}$ and $\lambda^{(2)}$ angles for rotation of the residue's peptide groups about the respective virtual-bond axes (Fig. 1). Because the allowed regions of the Ramachandran maps are quite restricted, so is the averaging over the λ angles. In the first approximation, the μ_1 and μ_2 components of the dipole moments of a given residue can be considered to correspond to the allowed orientations of the peptide groups of that residue. However, each interacting peptide group is shared between two consecutive residues; it is both the second peptide group of the preceding residue and the first peptide group of the succeeding residue. Therefore, its orientation is a compromise between the orientation forced by the Ramachandran map of the preceding and the succeeding residue, respectively and, consequently, the fictitious dipole moment of that peptide group is a vector sum of the the “dipole moments” resulting from the local-interaction pattern of the preceding and the succeeding residue.

The interpretation of the $U_{\text{corr}}^{(3)}$ coupling terms enables us to determine which local geometries of the $\text{C}^\alpha \cdots \text{C}^\alpha \cdots \text{C}^\alpha \cdots \text{C}^\alpha$ fragments (defined by the virtual-bond angles θ and virtual-bond-dihedral angles γ) result in the strongest long-range electrostatic interactions. Naturally, these are the geometries for which the two component dipole moments are aligned to produce a large net dipole moment. Consequently, the γ angle corresponding to the alignment depends on the adjacent virtual-bond angles θ . This dependence is plotted in Fig. 3, where the magnitudes of the components of the fictitious dipoles were taken from the ab initio energy surface of terminally-blocked alanine residue (which represents all residues except glycine and proline) calculated at the ab initio MP2/6-31G(d,p) level [13]. It can be seen from the Figure that the alignment angle γ depends primarily on the first θ angle of the four- C^α frame and that this dependence possesses a rapid jump at $\theta_1 \approx 120^\circ$. This suggests that the region with low θ and low but positive γ corresponds to helical and that with extended θ and $\gamma > 180^\circ$ or, equivalently $\gamma \leq -180^\circ$ corresponds to the β structures. Superpositions of the respective points derived from the PDB confirms that this is the case [13]. Of course, although it follows from the graph that the range of the γ angle is pretty much restricted to the values characteristic of helical or sheet structure (with β -helices being contained in the second region), the angles θ can still vary considerably. However, this variation becomes restricted if the long-range interactions between the total fictitious dipoles are considered; then only the angles close to 90° for helices

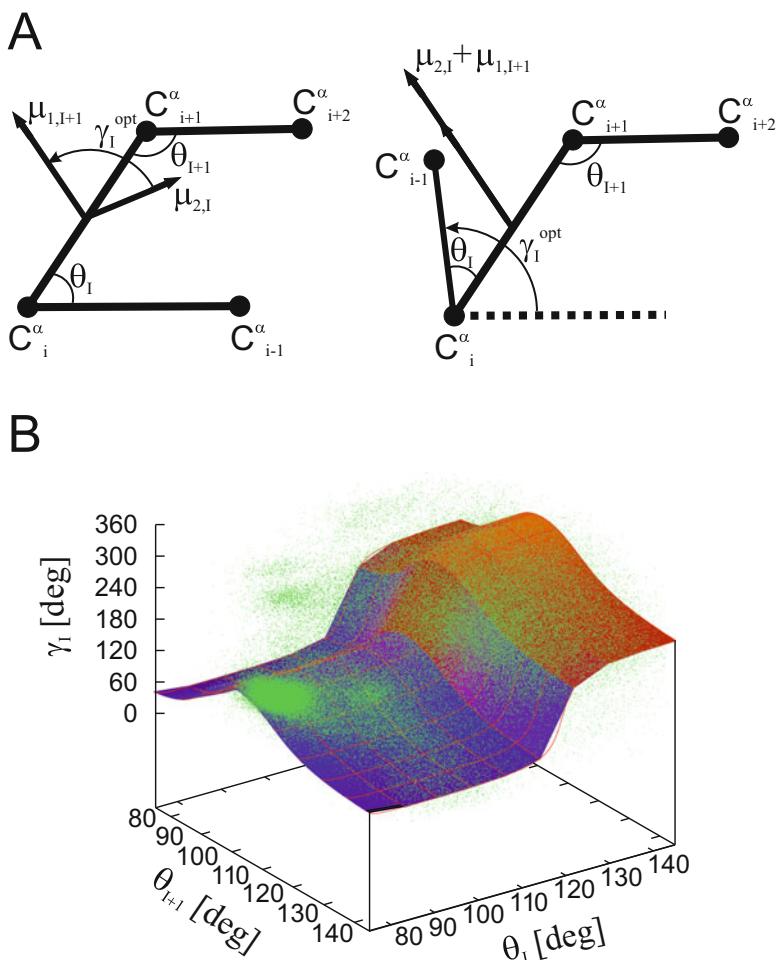


Fig. 3 (a) Illustration of the alignment of the fictitious dipoles $\mu_{1,I+1}$ and $\mu_{2,I}$ to produce the optimum value of the third-order correlation term (Eq. 98 of ref. 13). On the left side of the panel, the dipoles are in reference orientation corresponding to the virtual-bond-dihedral angle $\gamma = 0$ while, on the right side of the panel, these dipoles are aligned to produce the maximum total dipole, following the rotation of the $C_{i-1}^\alpha \dots C_i^\alpha \dots C_{i+1}^\alpha$ frame (and, consequently, $\mu_{2,I}$), by the angle γ_I^{opt} . The respective virtual-bond angles θ_I and θ_{I+1} are also shown in the figure. **(b)** A three-dimensional plot of the γ_I^{opt} vs. the θ_I and θ_{I+1} angles for the optimal alignment of peptide groups to achieve the most negative third-order correlation term accruing from the coupling between the backbone-local and backbone-electrostatic interactions (Eq. 98 of ref. 13) superposed on the scattered plot of the $(\theta_I, \theta_{I+1}, \gamma_I)$ angle triads extracted from the PDB. Reproduced from J. Chem. Phys. 146, 124106 (2017), with the permission of AIP Publishing

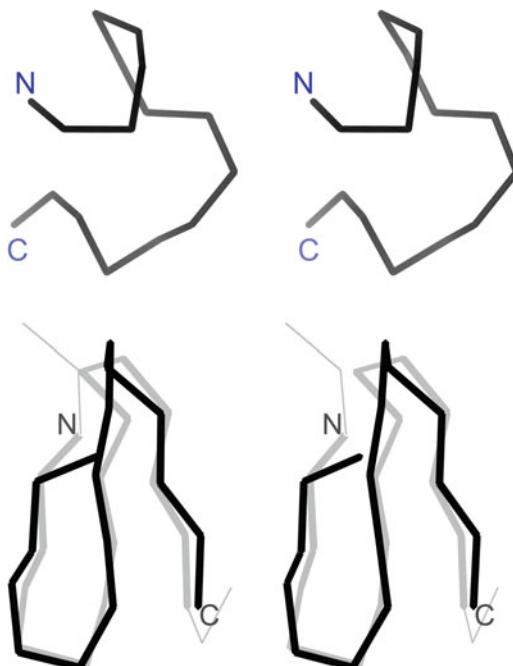


Fig. 4 Stereoview of the lowest-energy structures of betanova calculated with the UNRES force field without inclusion of the correlation terms (top) and stereoview of the superposition of the structure of betanova calculated with inclusion of the correlation terms (light gray lines) on the NMR structure (dark gray lines); the C^α RMS deviation being 1.2 Å (bottom). The N- and C-termini are indicated for tracing purposes. Reproduced from J. Chem. Phys. 115, 2323 (2001), with the permission of AIP Publishing

and extended for sheets and β -helices lead to geometries such that long-range interactions are efficient.

The importance of the correlation contributions is illustrated in Fig. 4 with the example of a small three-stranded anti-parallel β -sheet (betanova) [20], simulated with two variants of the UNRES coarse-grained force field developed in our laboratory [11, 12]. It can be seen (top panel) that no β -sheet structure can be obtained without the coupling terms. Conversely (bottom panel) a three-stranded β -sheet was obtained as the lowest-energy structure after the correlation contributions had been included [10]. Moreover, as illustrated in Fig. 5 with the example of a truncated mutant of the FBP28 WW domain [21], it is also important that the correlation terms depend on both virtual-bond-dihedral and virtual-bond angles, according to Eqs. 16–18. It can be seen that, with the potentials that we derived in our earlier work [10], which depended only on the virtual-bond-dihedral angles, the virtual-bond angles become too small, in order to maximize the interactions between the fictitious dipoles. Only after introducing appropriate dependence on the virtual-bond angles [13] (which

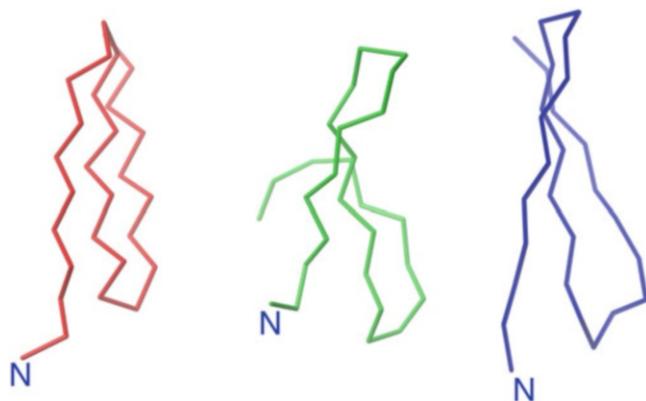


Fig. 5 Comparison of the experimental structure of the truncated FBP28 WW domain mutant (PDB: 2MWD; green, center) [21] with the structure calculated with the OPT-WTFSA-2 variant of the UNRES force field [22] that does not include the dependence of the correlation terms of backbone virtual-bond angles (left, red; the C^α -RMSD = 4.95 Å) and that calculated with the recent variant of UNRES (A. Liwo, A.K. Sieradzan, A. Lipska, C. Czaplewski, unpublished) that does include this dependence (blue, right, C^α -RMSD = 3.95 Å). The N-termini are marked for tracing purposes

follows from the Ramachandran maps) the virtual-bond angles in the calculated model become sufficiently large.

4 Conclusions

As shown in our earlier work [10, 13] and in this chapter, the interplay between backbone-electrostatic and backbone-local interactions (that pre-sculpt the energy surface of polypeptide chains to reduce the choice to that between the geometry of regular helix and sheet or β -helix structures) can be captured by a sum of third-order generalized Kubo cumulants corresponding to the coupling between the long-range electrostatic interactions and backbone-local interactions of the adjacent residues. Qualitatively good approximations of this term can be obtained at a relatively low cost, given the energy surfaces of terminally-blocked residues. In this work, we used alanine to represent all amino-acid residues except for glycine and proline (which were, consequently, removed from considerations). However, this treatment can be extended to distinguish more residue types. Moreover, it can also be used to estimate the regular structures of proteins with D-amino-acid residues or engineered residues, without doing expensive simulations. Of course, the result of such analysis indicates only the pre-sculpting of the energy landscape, while whether and what

kind of regular structure will ultimately form in a given chain fragment results from the totality of interactions.

Acknowledgments

This work was supported by grants UMO-2017/25/B/ST4/01026, UMO-2015/17/D/ST4/00509 and UMO-2017/26/M/ST4/00044 from the National Science Center of Poland (Narodowe Centrum Nauki). Calculations were carried out using the computational resources provided by (a) the supercomputer resources at the Informatics Center of the Metropolitan Academic Network (CI TASK) in Gdańsk, (b) the supercomputer resources at the Interdisciplinary Center of Mathematical and Computer Modeling (ICM), University of Warsaw (grant GA71-23), (c) the Polish Grid Infrastructure (PL-GRID), and (d) our 488-processor Beowulf cluster at the Faculty of Chemistry, University of Gdańsk.

References

1. Ramachandran GN, Sasisekharan V (1968) Conformation of polypeptides and proteins. *Adv Protein Chem* 23:283–437
2. Pauling L, Corey RB (1951) Configuration of polypeptide chains. *Nature* 168:550–551
3. Pauling L, Corey RB (1953) Stable configurations of polypeptide chains. *Proc R Soc B* 141:21–33
4. Hoang TX, Trovato A, Seno F, Banavar JR, Maritan A (2004) Geometry and symmetry presculpt the free-energy landscape of proteins. *Proc Natl Acad Sci U S A* 101:7960–7964
5. Molkenthin N, Hu S, Niemi JA (2011) Discrete nonlinear Schrödinger equation and polygonal solitons with applications to collapsed proteins. *Phys Rev Lett* 106:078102
6. Krokhotin A, Liwo A, Maisuradze GG, Niemi AJ, Scheraga HA (2014) Kinks, loops, and protein folding, with protein a as an example. *J Chem Phys* 140:4855735
7. Peng XB, Sieradzan AK, Niemi AJ (2016) Thermal unfolding of myoglobin in the Landau-Ginzburg-Wilson approach. *Phys Rev E* 92:062405
8. Maier JA, Martinez C, Kasavajhala K, Wickstrom L, Hauser KE, Simmerling C (2015) ff14sb: Improving the accuracy of protein side chain and backbone parameters from ff99sb. *J Chem Theory Comput* 11:3696–3713
9. Kmiecik S, Gront D, Kolinski M, Witeska L, Dawid AE, Kolinski A (2016) Coarse-grained protein models and their applications. *Chem Rev* 116:7898–7936
10. Liwo A, Czaplewski C, Pillardy J, Scheraga HA (2001) Cumulant-based expressions for the multibody terms for the correlation between local and electrostatic interactions in the united-residue force field. *J Chem Phys* 115:2323–2347
11. Liwo A, Czaplewski C, Ołdziej S, Rojas AV, Kaźmierkiewicz R, Makowski M, Murarka RK, Scheraga HA (2008) Simulation of protein structure and dynamics with the coarse-grained UNRES force field. In: Voth G (ed) Coarse-graining of condensed phase and biomolecular systems. Taylor & Francis Group, LLC, Boca Raton, pp 1391–1411
12. Liwo A, Baranowski M, Czaplewski C, Gołaś E, He Y, Jagiełła D, Krupa P, Maciejczyk M, Makowski M, Mozolewska MA, Niadzvedtski A, Ołdziej S, Scheraga HA, Sieradzan AK, Ślusarz R, Wirecki T, Yin Y, Zaborowski B (2014) A unified coarse-grained model of biological macromolecules based on mean-field multipole.multipole interactions. *J Mol Model* 20:2306
13. Sieradzan AK, Makowski M, Augustynowicz A, Liwo A (2017) A general method for the derivation of the functional forms of the effective energy terms in coarse-grained energy functions of polymers. I. Backbone potentials of coarse-grained polypeptide chains. *J Chem Phys* 146:124106

14. Ayton GS, Noid WG, Voth GA (2007) Multi-scale modeling of biomolecular systems: in serial and in parallel. *Curr Opin Struct Biol* 17:192–198
15. Nishikawa K, Momany FA, Scheraga HA (1974) Low-energy structures of two dipeptides and their relationship to bend conformations. *Macromolecules* 7:797–806
16. Berman HM, Westbrook J, Feng Z, Gilliland G, Bhat TN, Weissig H, Shindyalov IN, Bourne PE (2000) The protein data bank. *Nucl Acid Res* 28:235–242
17. Kubo R (1962) Generalized cumulant expansion method. *J Phys Soc Japan* 17:1100–1120
18. Liwo A, Khalili M, Czaplewski C, Kalinowski S, Ołdziej S, Wachucik K, Scheraga HA (2007) Modification and optimization of the united-residue (UNRES) potential energy function for canonical simulations. I. Temperature dependence of the effective energy function and tests of the optimization method with single training proteins. *J Phys Chem B* 111:260–285
19. Ołdziej S, Łagiewka J, Liwo A, Czaplewski C, Chinchio M, Nania M, Scheraga HA (2004) Optimization of the UNRES force field by hierarchical design of the potential-energy landscape. 3. Use of many proteins in optimization. *J Phys Chem B* 108, 16950–16959
20. Kortemme T, Ramirez-Alvarado M, Serrano L (1998) Design of a 20-amino acid, three-stranded β -sheet protein. *Science* 282:253–256.
21. Zhou R, Maisuradze GG, Sunol D, Todorovski T, Macias MJ, Xiao Y, Scheraga HA, Czaplewski C, Liwo A (2014) Folding kinetics of ww domains with the united residue force field for bridging microscopic motions and experimental measurements. *Proc Natl Acad Sci U S A* 111:18243–18248
22. Krupa P, Halis A, Źmudzińska W, Ołdziej S, Scheraga HA, Liwo A (2017) Maximum likelihood calibration of the UNRES force field for simulation of protein structure and dynamics. *J Chem Inf Model* 57:2364–2377



Chapter 8

Learning Organizations of Protein Energy Landscapes: An Application on Decoy Selection in Template-Free Protein Structure Prediction

Nasrin Akhter, Liban Hassan, Zahra Rajabi, Daniel Barbará,
and Amarda Shehu

Abstract

The protein energy landscape, which lifts the protein structure space by associating energies with structures, has been useful in improving our understanding of the relationship between structure, dynamics, and function. Currently, however, it is challenging to automatically extract and utilize the underlying organization of an energy landscape to link structural states it houses to biological activity. In this chapter, we first report on two computational approaches that extract such an organization, one that ignores energies and operates directly in the structure space and another that operates on the energy landscape associated with the structure space. We then describe two complementary approaches, one based on unsupervised learning and another based on supervised learning. Both approaches utilize the extracted organization to address the problem of decoy selection in template-free protein structure prediction. The presented results make the case that learning organizations of protein energy landscapes advances our ability to link structures to biological activity.

Key words Protein structure space, Energy landscape, Nearest neighbor graph, Communities, Basins, Community detection, Basin finding, Unsupervised and supervised learning, Decoy selection, Template-free protein structure prediction

1 Introduction

The tertiary structures in which the sequence of amino acids that constitute a protein molecule folds in three-dimensional space determine to a great extent the biological activities of a protein; the geometric and physicochemical complementary of molecular structures drives molecular docking events, making the ability of a protein to assume specific structures under physiological conditions essential to regulating interactions with molecular partners in the cell [1].

Algorithmic and hardware advances have resulted in an explosion of protein tertiary structure data. It is now possible to generate

thousands of tertiary structures for a (target) protein of interest, even when provided only with its amino acid sequence, in a matter of days, leveraging embarrassing parallelism in supercomputer architectures [2]. Some of the most visible computational methods able to do so are template-free protein structure prediction methods, such as Rosetta [3], Quark [4], and others [5, 6]. These methods operate under the umbrella of stochastic optimization, seeking local minima of some selected energy function that correspond to possibly biologically active/native structures in the vast structure space. Effectively, these methods probe the structure space (and the associated energy landscape that lifts the structure space with the additional dimension of energy) one structure at a time and yield diverse tertiary structures of a target protein.

The availability of diverse tertiary structures populating the structure space of a target protein presents an opportunity to analyze the probed space and possibly reveal, in an automated fashion, native structures. Though traditionally much of the literature on protein modeling (and template-free structure prediction, in particular) refers to one native structure, there is a growing consensus that the multiplicity of native structures cannot be ignored [2, 7–9]. We now know that proteins, like many other biological macromolecules, are intrinsically dynamic and undergo structural rearrangements to accommodate different molecular partners and so regulate their biological activities in the cell [1]. Lifting the structure space to additionally associate an energy with each computed structure reveals an energy landscape that is often rich in broad and deep wells/basins [9]; such basins correspond to thermodynamically stable structural states that are similarly long-lived and harnessed by a protein to participate in diverse cellular processes [10].

In principle, analysis of a computationally probed protein structure space or its energy landscape ought to reveal the structural states relevant for the possibly diverse menu of biological activities [11]. Doing so remains challenging. Revealing such states necessitates extracting the underlying organization of the structure space and/or the associated energy landscape. In addition, after such an organization is revealed, methods are needed to recognize in it states that are possibly active (or native). Such methods need to operate in the presence of approximations and errors that manifest themselves in possibly sparse and nonuniform distributions of structures sampled from the structure space guided by an energy function that may be inherently inaccurate and steer away from actual native structures.

Though the above task presents many challenges, in this chapter we relate some first steps to tackling it. We report on two computational approaches that extract organizations of protein structure spaces or associated energy landscapes. We then describe two complementary approaches, one based on unsupervised

learning and another based on supervised learning, that demonstrate the utility of the extracted organizations in the context of the problem of decoy selection in template-free protein structure prediction. The presented results make the case that learning organizations of protein energy landscapes advances our ability to link structures to biological activity and that whole data-driven approaches warrant further investigation.

The rest of this chapter is organized as follows. In Subheading 1.1, we provide some more formalisms, defining the notion of an energy landscape and properly linking it to the structure space, and then summarize pertinent research in the context of decoy selection in template-free protein structure prediction. The approaches proposed for extracting and utilizing the underlying organization of a protein structure or its associated energy landscape are described in Subheading 2, followed by evaluation in Subheading 3. The chapter concludes with some thoughts regarding the current shortcomings of this line of work and possible future remedies in Subheading 4.

1.1 Related Work

1.1.1 Energy Landscape

The concept of a landscape (or lifted space) is general, appears in many scientific disciplines [12–15], and can be defined as follows: a landscape consists of a set X of points, a neighborhood $N(X)$ defined on X , a distance metric on X , and a function $f: X \rightarrow R \geq 0$ that assigns a score (also known as a height or energy, depending on the application domain) to every point x in X . Every point in X is assigned a neighborhood by the neighborhood function N . In the context of decoy selection, points x are (decoy) tertiary structures, and the function f that scores the decoys is most often referred to as an energy function.

Protein energy landscapes (as probed via designed functions f) are multidimensional and multimodal, containing many components or elements, such as basins (or wells) and basin-separating barriers. A local (energy) minimum in the landscape is surrounded by a basin of attraction (becoming its focal minimum), which is the set of points on the landscape from which steepest descent/ascent converges to that focal optimum.

The decoy generation phase in template-free structure prediction methods samples points from an unknown, underlying structure space X , guided by a selected energy function f . The decoys are evaluated via f , and so the output is a set $\{x, f(x)\}$ of evaluated decoys x . In light of this, a computational approach can seek to elucidate the organization of the sampled X (the structure space) or the sampled energy landscape available as $\{x, f(x)\}$.

1.1.2 Decoy Selection

Decoy selection refers to the problem of identifying one or more native structures from the set of (decoy) structures computed by a template-free protein structure prediction method. The problem remains open [16, 17], garnering its own evaluation category in the Critical Assessment of protein Structure Prediction (CASP) series

of community-wide experiments [18]. CASP assesses progress in protein structure prediction and is structured as a competition, where participants submit blind predictions (what they assess to be native or near-native structures) of selected target proteins. The submitted structures are evaluated by independent assessors after one or more native structures per target are made available by recruited wet laboratories. The latest CASP assessment [19] shows that decoy selection remains a bottleneck.

Decoy selection methods can be grouped into single-model, bag-of-models, quasi-single, and machine learning (ML) methods. Single-model methods evaluate each generated/computed decoy via a physics- or a knowledge-based function to associate an energy/score with each decoy [20, 21]. These methods make use of a score threshold to filter out decoys that are putatively not near (similar to) the native structure (“the true answer”). Relying on a threshold, however, is shown to either miss native structures or allow the inclusion of too many nonnative ones [17, 22, 23]. In response, a popular approach is to ignore energy altogether and cluster decoys by structural similarity [24, 25], offering one or more of the top highest-populated clusters as prediction.

Cluster-based decoy selection methods implement the bag-of-models approach and leverage the premise that decoys are randomly distributed around the “true answer,” which a consensus-seeking method should reveal [26]. This is not a valid assumption, as template-free protein structure prediction methods conduct biased sampling of the structure space (to handle its size and dimensionality), guided by an energy function that often contains inherent biases manifest in invalidating of entire regions of the structure space [27]. Indeed, an active thrust of research in protein structure modeling concerns the design of more accurate energy functions [28]. Currently, cluster-based methods fail to pick up good decoys when applied to hard targets, where most decoys are very different from the known native structure(s) and are highly dissimilar and sparsely sampled [17].

Quasi single-model methods combine strategies from single-model and bag-of-models methods. They first select some high-quality structures to which then the rest of the decoys are compared [29]. These methods are shown to perform better than single-model and consensus-seeking methods [30, 31]. Finally, a recent but promising line of research leverages ML models, such as Support Vector Machines (SVM) [32], Neural Network [33], and Random Forest [34]. For instance, work in [35] extracts 94 features from a protein structure to build an SVM model for selecting native decoys. Ensemble learning has been shown to outperform SVM learning [36], with or without statistical features derived off decoys.

Though in their infancy, ML methods warrant further evaluation. Some of the efforts related in this chapter can be categorized as building ML models under the umbrella of unsupervised or supervised learning. However, as Subheading 2 describes, these models do not operate over single decoys; instead they operate over extracted subsets of decoys automatically extracted from the structure space or the energy landscape of a protein of interest.

2 Methods

Both computational approaches that we propose start by embedding computed tertiary structures in a graph, which is then subjected to techniques that utilize the structure of the graph to partition it into nonoverlapping groups of structures. These groups are then subjected either to unsupervised or supervised learning techniques to extract from them the ones most likely to contain biologically active/native structures, evaluating the selection in the context of decoy selection. Due to the evaluation setting, from now on, we will refer to the tertiary structures as decoys, even though the methods and techniques described here can extend beyond the specific context of decoy selection. We now proceed to describe each of the methods.

2.1 Embedding of Tertiary Structures in a Nearest Neighbor Graph

Let us refer to the set of computed tertiary structures of a protein (in our context, decoys), as Ω . Envision that this set consists of points sampled from a high-dimensional (protein) structure space. To encode the sample proximity (and thus, similarity) in this space, the set can be embedded in a nearest neighbor graph (nngraph) $G = (V, E)$. The vertex set V is populated with Ω (each decoy becoming a vertex). The edge set E is populated by inferring a local neighborhood structure over each decoy.

The proximity of two decoys is measured via root-mean-square deviation (RMSD), after each decoy is superimposed over some reference decoy (arbitrarily, chosen to be the first one, for instance, in Ω); the reason for the superimposition is so as to minimize differences due to rigid-body motions (translations and rotations in $SE(3)$) [37]. Superimposing all decoys to a reference decoy a priori to the pairwise RMSD computation, rather than conducting the superimposition over every pair of decoys under comparison, saves computational time (from linear to quadratic). Using RMSD to compute the distance between two decoys, a vertex $u \in V$ is connected to vertices $v \in V$ if $d(u, v) \leq \epsilon$, where ϵ is a user-defined parameter. Proximity query data structures (such as kd-trees, VP-trees, C-trees, and others) can be used to efficiently extract the nearest neighbors of a vertex (rather than rely on brute-force, all-pair comparisons).

The resulting nngraph may be disconnected, if ϵ is small and the decoys are the result of a sparse, nonuniform sampling of the structure space. This can be in part remedied by initializing ϵ to an initial value (ϵ_0) and then increasing it by δ_ϵ over a maximum of n_ϵ iterations, all the while controlling the density of the resulting nngraph via a specified maximum number (k) of nearest neighbors per vertex. However, the quality of the sampling dictates in large part the quality of the embedding and the rest of the analysis methods proposed here. In the context of decoy selection, embarrassing parallelism of software such as Rosetta ensures a large number of decoys, though their quality varies based on the difficulty of the protein target, as we relate in Subheading 3.

2.2 Extracting Organization Via Nngraph Embeddings

2.2.1 Identifying Communities in the Graph-Embedded Structure Space

The resulting nngraph can now be investigated for its organization via two different approaches.

The first approach does not consider the energy/score of each decoy embedded in the nngraph; that is, the nngraph is seen as a discrete representation of the sampled structure space. Under this treatment, community detection methods, borrowed from the domain of complex network analysis (such as social networks), can be readily utilized to detect communities. These methods effectively conduct clustering of the vertices in the nngraph, leveraging the distribution of edges over vertices in a community versus those outside. There are many community detection methods, but we select six representative, state-of-the-art ones, which we investigate for their ability to expose the underlying organization of the protein structure space.

Specifically, we utilize the Leading Eigenvector (LE) method, the Walktrap (WT) method, the Label Propagation (LP) method, the Louvain (Lo) method, the InfoMap (IM) method, and the Greedy Modularity Maximization (GMM) method. In summary, the LE method aims to maximize modularity over possible partitions of a graph by utilizing the eigenspectrum of the modularity matrix. In the WT method, random walks are used to capture similarities between vertices (or sets of vertices), and agglomerative approach is used to hierarchically combine two adjacent communities at a time. In the LP method, each vertex is given a unique label (thus starting with $|V|$ communities, and labels are iteratively propagated through the network, reaching a consensus on a unique label. The Lo method is based on modularity maximization and assigns vertices to communities based on modularity gain. In the IM method, communities are identified using random walks that analyze the information flow. In the GMM method, vertices are repeatedly joined together into two communities, whose modularity produces the largest increase, producing a dendrogram that

encodes good partitions based on high values of modularity. In the interest of brevity, we do not describe these methods in detail, but the interested reader can find a comprehensive review in ref. 38.

2.2.2 Identifying Basins in the Graph-EMBEDDED Energy Landscape

While the above approach only considers proximity of decoys in the structure space to detect an underlying organization, the graph embedding can be lifted in the energy landscape, additionally considering the energy/score of each decoy, as follows. Each vertex additionally contains the score of the corresponding decoy. Vertices that constitute local minima in the energy landscape are identified first. A vertex u in V is a local minimum if $\forall v \text{ in } V f(u) \leq f(v)$, where v in $N(u)$ ($N(u)$ denotes the 1 – neighborhood of u). The remaining vertices are then grouped into basins in the landscape as follows. Each vertex u is associated with a negative gradient estimated by selecting the edge (u, v) that maximizes the ratio $[f(u) - f(v)]/d(u, v)$, where $d(u, v)$ is the RMSD between the two corresponding decoys. From each vertex u that is not a local minimum, the negative gradient is then followed iteratively (following the edge that maximizes the above ratio) until a local minimum is reached. Vertices that reach the same local minimum are assigned to the basin associated with that minimum, which is considered the focal minimum of that basin. This approach of leveraging the nngraph to identify basins in the landscape is first described in [39] as part of the Structural Bioinformatics Library (SBL) suite of structure and structure ensemble analysis algorithms.

2.3 Unsupervised Learning for Decoy Selection

Let us generally refer to the communities or basins identified as above as groups G . Different measurements can be associated with a group. Two such measurements are size and energy. Size refers to the number of decoys/vertices in a group. The energy of a group can be defined in two different ways, either as the minimum or average energy/score over the decoys in the group.

Given any of these two measurements, identified groups can then be ranked. For instance, considering only size, the identified groups can be ranked in a descending sorted order, and an automatic unsupervised learning strategy can extract the top c groups and offer them as “prediction” for where native and near-native structures reside. We refer to this strategy as UL-S. Another ranking strategy can additionally consider energy, considering energy alone has long been proven ineffective, as summarized in Subheading 1. So, instead, in UL-S + E, we consider the top $l > c$ largest groups (in the sorted order), and then resort these l groups from lowest to highest energy, selecting the top c of them for prediction. Two more strategies can be devised, based on Pareto optimality in multi-objective optimization, recognizing the unclear interaction between the size and energy of a group.

Suppose we want to select optimally considering various conflicting criteria/objectives. In this scenario, Pareto-optimal

solutions are sought, as a single solution minimizing all conflicting objectives simultaneously is typically nonexistent; a Pareto-optimal solution cannot be improved in one objective without sacrificing the quality of at least one other objective, i.e., a solution S1 Pareto dominates another solution S2 if the following two conditions are satisfied: (1) for all optimization objectives, i.e., $\text{score}_i(S1) \geq \text{score}_i(S2)$, and (2) for at least one optimization objective, i.e., $\text{score}_i(S1) > \text{score}_i(S2)$.

One can now associate two additional quantities, Pareto Rank (PR) and Pareto Count (PC) with each group G. PR(G) is the number of groups that dominate G, and PC(G) is the number of communities that G dominates. So, two additional, Pareto-based strategies are proposed. In UL-PR, the identified groups are sorted by low to high PR values, and the top c communities in this sorted order are selected and analyzed. In UL-PR + PC, PC is additionally considered. Communities with the same PR value are sorted from high to low PCs, and the top c groups in this resulting sorted order are selected.

2.4 Supervised Learning for Decoy Selection

We consider both the classification and regression setting. The training and testing datasets consist of groups of decoys. Two settings are considered, when the groups are all communities identified as above or all basins also described above. In the classification setting, a group is labeled as either pure (Class 1) or impure (Class 0), given a user-defined threshold of purity (with purity measured over a group as described above). In the regression setting, the purity of a group is the actual output variable. In both settings, a group is reduced to a vector of four attributes, size, energy, PR, and PC, described above. In the case where models learn over basins, a fifth attribute is considered, basin stability, which relates to the measurement of whether a basin is shallow or deep relative to its nearest saddle [39]. Attributes are normalized to take values in the [0,1] range.

Most models struggle on classifying data with an imbalanced class distribution; this is the case on decoy datasets, where the number of what are deemed to be near-native decoys (those with IRMSD from a given native structure is within dist_thresh) is far outnumbered by the number of decoys deemed to be nonnative.

Since supervised learning models are generally designed to maximize overall prediction accuracy, these methods do not consider the imbalance data distribution. Rather, the majority class data are more focused, and the minority/rare class data are ignored and often misclassified. In our evaluation setting on decoy datasets, the number of what are deemed to be near-native decoys (those with IRMSD from a given native structure is within dist_thresh) can be far outnumbered by the number of decoys deemed to be nonnative. Three common approaches to address an imbalance class distribution are either to (1) modify the model being utilized,

(2) preprocess the data to compensate for the imbalance in the data distribution (notable, not all models are amenable to this change), or (3) select features carefully so as to be less affected by the imbalance class distribution. We adopt the algorithmic approach of using ensemble learning techniques and the data preprocessing approach of data under-sampling to address the imbalanced dataset issue.

We consider three representative ensemble learning methods [40] in the classification setting: Extreme Gradient Boosting (XGBoost), Balanced Bagging (BB), and Remove Tomeklinks (RT). XGBoost, is a fast, scalable implementation of the Gradient Boosting Machine (GBM), itself a boosting-based ensemble approach that adopts a gradient-descent based formulation [41]. Instead of reweighing the input samples in each iteration, GBM adds a weak learner to minimize an arbitrary differentiable loss function. GBM has been widely used in many machine learning applications with considerable success. XGBoost addresses the overfitting problem in GBM [42] and has rapidly become the most popular boosting technique in data mining and machine learning applications and is freely available via the XGBoost Python package. BB, also referred to as Bootstrap Aggregating, combines predictions of multiple predictors either via an averaging (regression) or voting (classification) scheme. To address class imbalance, BB resamples (under- or over-samples) each bootstrap training set to make them balanced before training is performed. The free Python package imbalanced-learn allows for both over-sampling and under-sampling of data. We opted for the under-sampling option (which greatly reduces overfitting) of the imbalanced-learn to balance our decoy datasets for BB. The RT method employs the concept of Tomeklinks, which are data samples that are each other's nearest neighbors but have different class labels. Removing Tomeklinks is an effective way to eliminate the unwanted class overlap and under-sample the data [43]. We utilize Python's scikit-learn's Nearest Neighbor classifier, which uses a kd-tree, to search for nearest neighbors in our implementation of removing Tomeklinks. Out of the three options of removing noisy data samples, we choose to remove both majority and minority classes that are nearest neighbors of each other to under-sample and balance our decoy datasets.

In the regression setting, we consider two representative methods: XGBoost, described above, and support vector regression (SVR). For SVR, we use Python's scikit-learn's SVR package, which implements an epsilon-support vector regression with a default radial basis function (rbf) kernel and free parameters of penalty and epsilon.

2.5 Evaluation for Decoy Selection

The quality of a subset of decoys (pulled from Ω) offered as “prediction,” whether the set is learned in an unsupervised or a supervised setting, can be evaluated based on the number of near-native

decoys contained in them via two main metrics: (1) n , the percentage of near-native decoys in a given subset relative to the overall number of near-native decoys in the entire decoy set Ω , and (2) p , the percentage of near-native decoys in a given subset over the number of decoys in that same subset. We note that purity penalizes a large subset that, due to its size, may contain a large number of true positives (near-native decoys) but also a high number of false positives (nonnative decoys). The reason for penalizing the number of false positives is by envisioning that a learning strategy may be utilized for further discovery. If a subset of decoys is presented to contain the true answer, but the majority of decoys in it are false positives, then the ratio of noise to signal is too high to be useful. On the contrary, a subset of decoys with more near-native decoys in them is a better “prediction,” as the likelihood of selecting a near-native decoys by drawing uniformly at random from it is higher when the number of false positives, nonnative decoys, is low. We emphasize that in the context of unsupervised learning, the top $c > 1$ groups are offered as prediction; in this setting, the decoys from these groups are combined (which we note via $G1 - c$), and the resulting subset of decoys is evaluated via the n and p metrics. In the context of supervised learning, the groups labeled as true are pulled together to form a subset of decoys (from the overall set Ω), and this subset is evaluated via the n and p metrics described above.

We note that key to the evaluation is the need to determine whether a decoy is considered near-native or nonnative; an RMSD threshold is utilized again. The selected threshold needs to allow populating the positive dataset (non-zero number of near-native decoys), which can then be used to evaluate a learning strategy. A threshold $dist_thresh$ is used and is set on a per-target basis, as there are protein targets on which the quality of generated decoys suffers greatly from either the size and/or fold of the protein under investigation; that is, the quality of the dataset can vary greatly depending on the protein target at hand. In our drawing of a list of representative target proteins, we consider targets that are easy, medium, and hard in their difficulty for the Rosetta ab initio structure prediction protocol. For instance, we include in our evaluation targets where Rosetta does not get close to 3 Å of the known native structure (the ground truth). Specifically, we consider the following thresholds: If the lowest IRMSD from a given native structure (overall decoys), to which we refer as min_dist , is ≤ 0.7 (these are considered easy cases), $dist_thresh$ is set to 2 Å. Otherwise, $dist_thresh$ is set to the minimum value that results in a non-zero number of near-native decoys populating the largest-size cluster obtained via leader clustering; the latter is used as a baseline in our comparison of community selection to cluster-based selection. For medium-difficulty proteins ($0.7 \text{ \AA} < min_dist < 2 \text{ \AA}$), $dist_thresh$ varies between 2 and 4.5 Å. We set $dist_thresh$ to 6 Å if $min_dist \geq 2 \text{ \AA}$ (these are the hard cases). This ensures a non-zero

number of near-native decoys to evaluate decoy selection strategies. A detailed analysis of the impact of this threshold on cluster-based decoy selection is related in our recent work [44], where we evaluate basins identified as described above for utilization in decoy selection.

2.6 Implementation Details

All in-house codes are implemented in Python. In the nngraph construction, $\delta_\epsilon = 0.2 \text{ \AA}$, $k = 20$, $n_\epsilon = 5$, and ϵ_0 are set so as not to exceed 900 K edges, varying in 0.5–2.2 Å for most of the test cases, with one particularly challenging case for Rosetta set to 6.0 Å. The nngraph construction takes between 26 min and 4.25 h on one CPU over decoy datasets of around 50,000 decoys for proteins ranging in length from 53 to 93 amino acids. We consider two proximity query data structures, kd-tree versus VP-tree; we select the kd-tree over the VP-tree for fast extraction of nearest neighbors, based on analysis of the time demands as a function of dimensionality (data not shown). The community detection methods take between 7 and 66 min using 8 cores and 1 GB memory per core. The basin detection approach takes between 26 min and 2.25 h on one CPU. In the unsupervised learning setting, we consider $1 \leq c \leq 3$.

In addition to the n and p metrics to evaluate performance, in the supervised learning setting, we consider standard metrics, such as accuracy, precision, recall, F-measure, and the number of hits, which indicates how many truly pure groups can be identified. In the classification setting, 15 rounds of boosting are used for XGBoost, using area under the curve (auc) as the evaluation metric. Two variants of XGBoost are considered: XGBoost with undersampling (XGBoost-us) and XGBoost with under-sampling and averaging (XGBoost-us-avg). In XGBoost-us, the data is under-sampled so that the number of true negatives is equal to the number of true positives (for instance, if the size of the true positive set is m , then the dataset size is $2m$). In XGBoost-us-avg, we have multiple subsets of true negatives which are the same size as the true positive set (for, instance, if the dataset size is n and the size of the true positive set is m , then, the number of true negative sets is $k = n/m$, and the total size of the training dataset is $(k + 1)m$; the total number of models is k , where each model is trained on a dataset of size $2m$ (m true positives and m true negatives)). To make a final prediction, predictions by each individual classifier trained on each of the k under-sampled dataset are averaged. In BB, we investigate different base learners, such as Random Forest, Decision tree, Gradient boosting, AdaBoost, Randomized decision tree, and a voting classifier. We report the best result obtained by using any of these base learners. In RT, we use kd-tree to compute the nearest neighbors and set this number to two. In SVR, in the regression setting, the free penalty and epsilon parameters are set to the default values of 1.0 and 0.01, respectively. In XGBoost for

regression, we use 15 rounds of boosting, using the gblinear booster with linear regression as the learning objective. The evaluation metric is set to both root-mean-square error (RMSE) and mean absolute error (MAE).

3 Results

Table 1 lists the ten protein targets selected for evaluation. The decoy ensemble of each target is generated from its amino acid sequence, running the Rosetta ab initio protocol [3] around 50,000 times in an embarrassingly parallel fashion in the Mason Argo supercomputing cluster to obtain ensembles of around 50,000 decoys per target; the actual ensemble sizes are shown in Column 5 in Table 1. Column 3 shows the Protein Data Bank [45] identifier (PDB id) for a known, crystallographic native structure of each protein. This structure is used to determine which decoys can be considered near-native (via the d thresh parameter). Table 1 divided the targets into three categories (easy, medium, and hard). This categorization is made evident by findings reported later, but it also emerges from analysis in terms of the lowest (l) RMSD overall decoys from the corresponding native structure. The lowest IRMSD over the decoys for each protein is listed in Column 6 in Table 1 (referred to as min_dist).

The findings are presented as follows. First, we demonstrate that two out of the six community detection methods are superior in terms of the qualities of the communities they detect. Second, we then evaluate the top communities by each of these two methods

Table 1

Column 2 shows the PDB ID of a known native structure for each test case

	PDB ID	Fold	Length (#aas)	 Ω 	min_dist (Å)
Easy	1. 1dtdb	$\alpha + \beta$	61	57,839	0.51
	2. 1tig	$\alpha + \beta$	88	52,099	0.60
	3. 1dtja	$\alpha + \beta$	74	53,526	0.68
Medium	4. 1hz6a	$\alpha + \beta$	64	57,474	0.72
	5. 1c8ca	β^*	64	53,322	1.08
	6. 1bq9	β	53	53,663	1.30
	7. 1sap	β	66	51,209	1.75
Hard	8. 2ezk	α	93	50,192	2.56
	9. 1aoy	α	78	52,218	3.26
	10. 1iusa	coil	62	60,360	5.53

Columns 3 and 4 show the fold (* indicates native structures with a predominant β fold and a short helix) and the length (number of amino acids), respectively. Column 5 shows the size of the decoy set Ω generated via Rosetta, and Column 6 shows the lowest IRMSD from the known native structure over the decoy ensemble

via the n and p metric in the unsupervised learning setting, comparing communities to basins. Our findings show that basins are of higher quality. Finally, we relate the performance of the various classification and regression models over identified basins.

3.1 Comparison of Community Detection Methods

We consider a comprehensive list of 15 recommended metrics to evaluate the quality of communities identified from a graph via community detection methods [46]. These metrics are scoring functions that mathematically formalize the community-wise connectivity structure of a given set of vertices and characterize high-scored sets as communities. Specifically, we consider the Fraction Over Median Degree (FOMD), the Max-ODF (Out Degree Fraction), the Flake-ODF, the Triangle(Triad) Participation Ratio, the Internal Edge Density, the Average Internal Degree, the Cut Ratio, the Expansion, the Edges Inside, the Conductance, the Normalized Cut, the Coverage, the Average ODF, the Modularity, and the Separability metric. A description of these metrics can be found in ref. 46.

In Fig. 1 we relate the comparison along two selected metrics that represent our findings. Specifically, the top panel of Fig. 1 shows the comparison along coverage, which compares of the number of intra-community edges to $|E(G)|$. Higher values mean that there are more edges inside the communities than edges linking different communities; ideally, communities are disconnected from one another, yielding a maximum coverage of 1. The comparison shows that three methods reach the higher coverage values: GMM, Lo, and LP. The bottom panel of Fig. 1 shows the comparison along cut ratio. Cut ratio measures the fraction of existing edges (out of all possible edges) leaving a community, and an average value (averaged overall communities) can be reported to compare different community detection methods. Lower scores correspond to better communities. Due to the wide range over the different community detection methods, we relate \log_{10} of this metric in the bottom panel of Fig. 1, which shows that the three methods reaching the lowest values are GMM, Lo, and LP. Many of the metrics (data not shown) point to these three methods as superior over others. In particular, as the bottom panel of Fig. 1, GMM and Lo are two of the best-performing methods. The communities that they identify are visualized for a selected target protein in Fig. 2.

3.2 Evaluation of Unsupervised Learning for Decoy Selection

We now restrict our evaluation in the context of unsupervised learning over communities detected via Lo or GMM or basins detected as described in Subheading 2 (we refer to that approach as BF for basin finder) with UL-S, UL-S + E, UL-PR, and UL-PR + PC, evaluating via the n and p the decoys over $G1 - x$, with $x \in \{1, 2, 3\}$. In the interest of space, we only relate evaluation along G1 and G1–3 and only relate results from UL-S + E (which is

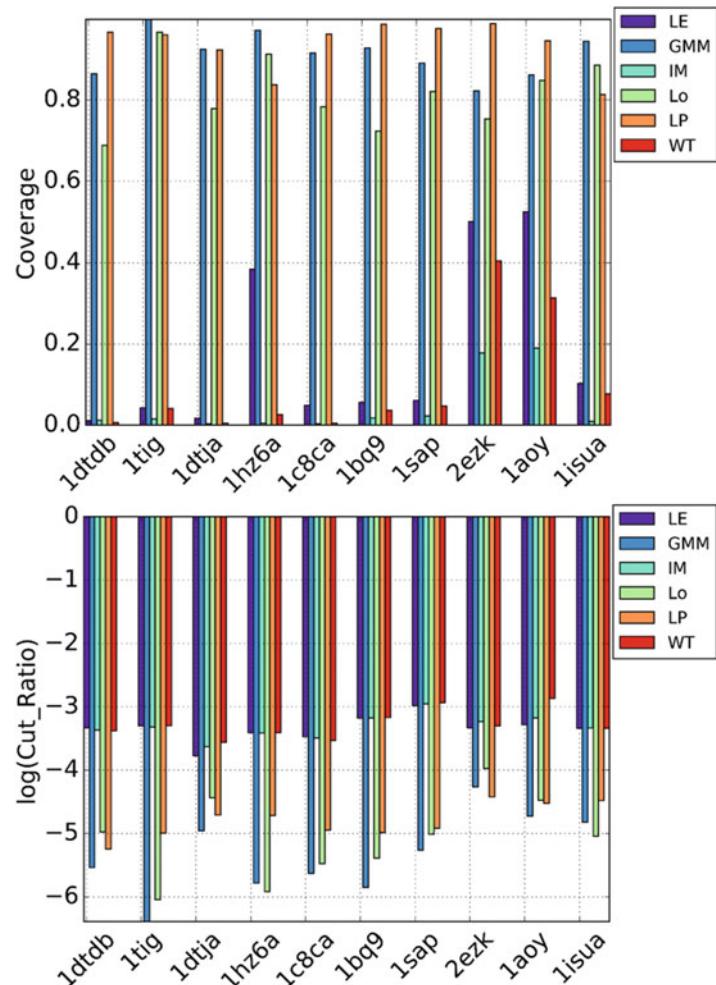


Fig. 1 Comparison of six community detection methods (encoded by different colors) on each of the ten datasets along coverage (top panel) and cut ratio (bottom panel)

the top or second top performing unsupervised learning strategy overall datasets, only rarely displaced from the top by UL-PR in a few datasets). Figure 3 shows the comparison along n , and Fig. 4 shows the comparison along p . Figure 3 shows no clear winners among community- or basin-based unsupervised learning, but Fig. 4, which compares purity, shows the superiority of learned basins over learned communities, suggesting that an unsupervised learning approach operating over basins is more likely to pick up purer subsets of decoys for decoy selection. In particular, these results suggest that ignoring the organization in the energy landscape (and conducting all analysis in the structure space) comes at the cost of allowing false positives in selected decoys. For this



Fig. 2 Communities identified via the Lo community detection method on the decoy dataset of the target protein with a known native structure under PDB id 1dtja are visualized via the Force Atlas 2 layout in Gephi [47]. This layout uses a force-driven, physics-inspired process, where nodes repel and edges attract, to flatten out a graph on a plane and visualize color-coded communities [48]

reason, the results related below evaluate supervised learning over identified basins.

3.3 Evaluation of Supervised Learning for Decoy Selection

Three settings are considered for the evaluation of classification methods: (1) training of a model on a single target protein (its decoy dataset) and testing on another single protein; (2) constructing a separate model for each of the three difficulty categories (easy, medium, hard), training a model on 60% of the decoys over all decoys combined over datasets in a category, and training the model on 40% of the remaining decoys in that category; (3) constructing a separate model for each of the three difficulty categories (easy, medium, hard), training a model on all but one of the proteins in a category, and testing it on the remaining protein in that category. Evaluation of regression methods is conducted only in the first setting.

Tables 2, 3, and 4 show performance in each of the three settings for the different classification methods. The best results consisting of high n and p are also highlighted in red and bold. The second best results where both n and p are also comparatively high are highlighted in blue and bold.

Table 2 shows the classification results in Setting 1. In this setting, we predict the n and p for one test protein given another protein as training data. Here, we investigate two categories: easy and medium. Our objective is to show the effect of mixing proteins in different categories as training and test data in the same model.

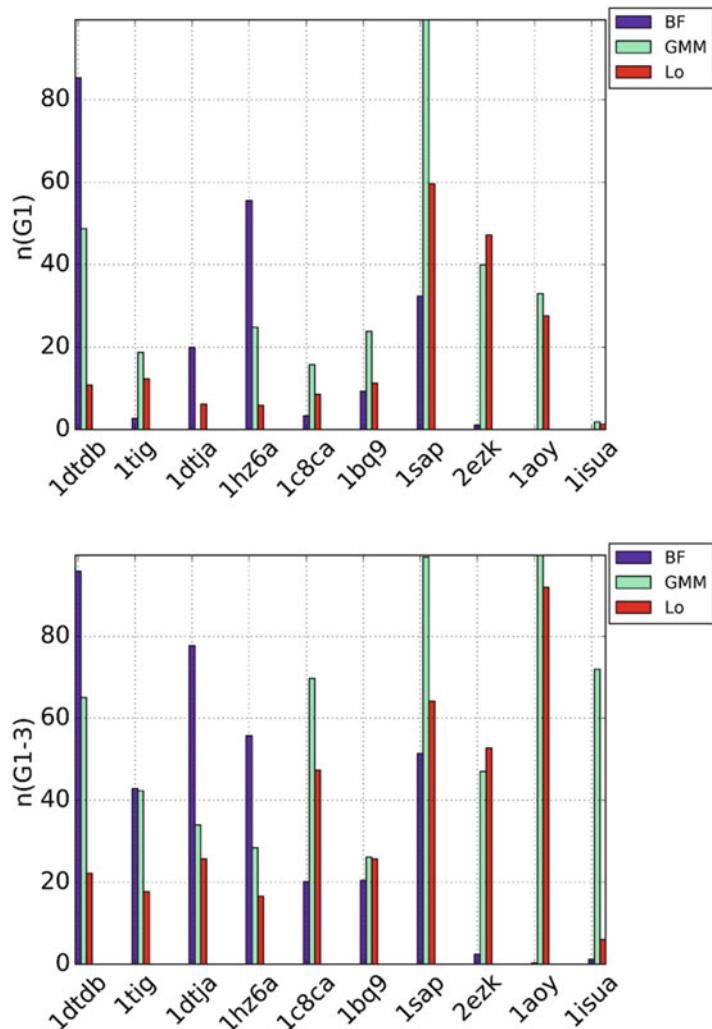


Fig. 3 Comparison of communities and basins in terms of n , selected by size and energy

Rows 1, 2, and 5 in Table 2 show the results when we train and test the models on proteins of the same category. The remaining rows show the results when we train and test the models on proteins of different categories.

Table 2 reveals that easy cases perform extremely well when both training and testing are done in the same easy category. The highest purity is 98.3 (train: 1dtdb, test: 1wapa), and there are two models that provide more than 90% of both n and p (XGBoost and BB). Three out of five models output both high n and p (more than 80%): BB, RT, and XGBoost. The last row of Table 2 indicates that the easy cases are good targets even if the training is done on proteins of a different category (medium). We can achieve as high as 100% of n , and purity p is satisfactory (55 and 66.4%). Moreover,

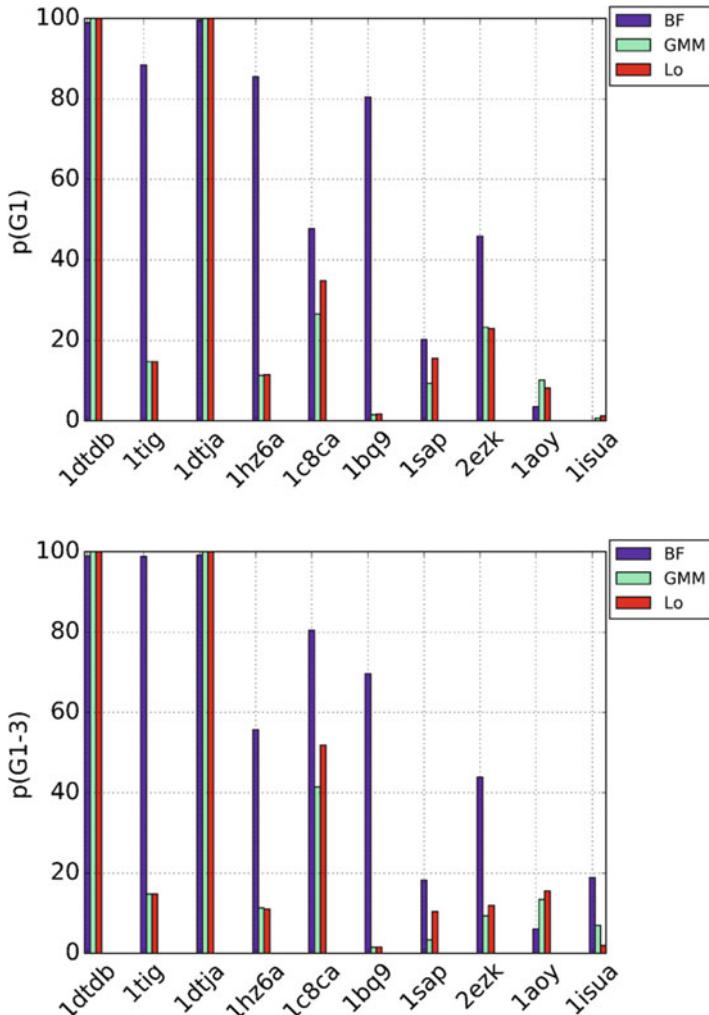


Fig. 4 Comparison of communities and basins in terms of p , selected by size and energy

easy cases are also good for training even if the target is of a different category (medium case, Rows 3 and 4). Satisfactory purity p is obtained by two models, XGBoost (p is 96.4%) and BB (p is 63.6%). When training and testing are done both on the medium category, performance is not as high as in the other scenarios, but it is still good; the maximum purity achieved in this scenario is 64.57 (Row 5). In summary, BB and XGBoost perform the best among all five models in terms of achieving better p and n . Under-sampling the dataset (XGBoost-us and XGBoost-us-avg) helps in obtaining comparatively good purity when the target becomes harder (medium case).

Table 3 shows the classification results in Setting 2. In this setting, we combine all the test cases of the same category into

Table 2

n proportion of true positives, *p* purity, hit number of true pure basins detected, *pr* precision, *r* recall, *f* f-measure, *acc* accuracy

(train size, test size)	Train, test, # pure basins: train, test	XGBoost	XGBoost -us	XGBoost -us-avg	BB	RT
		n%, p%, hits				
		pr, r, f, acc				
Train: easy Test: easy (3441, 5398)	Train: 1dtdb, test: 1wapa, # pure basins: 13, 18	98.3, 94.3, 15	100, 60, 18	99.9, 59.3, 17	99.8, 80.6, 16	98.2, 85.6, 14
		0.54, 0.83, 0.65, 0.99	0.1, 1.0, 0.13, 0.96	0.1, 0.99, 0.15, 0.95	0.3, 0.9, 0.44, 0.99	0.41, 0.78, 0.54, 0.99
Train: easy Test: easy (3441, 2473)	Train: 1dtdb, test: 1dtja, # pure basins: 13, 89	35.2, 94.2, 17	95.1, 72.6, 47	94.3, 73.1, 43	90.96, 91.7, 29	20, 99.6, 1
		0.85, 0.2, 0.31, 0.96	0.3, 0.53, 0.37, 0.94	0.31, 0.5, 0.36, 0.93	0.7, 0.33, 0.44, 0.97	1.0, 0.01, 0.02, 0.96
Train: easy Test: med (3441, 6435)	Train: 1dtdb, test: 1c8ca, # pure basins: 13, 210	31.6, 31.9, 17	86.2, 20.6, 154	81.3, 21.7, 131	56.3, 31.4, 42	41.3, 40, 11
		0.2, 0.08, 0.11, 0.96	0.08, 0.73, 0.14, 0.71	0.08, 0.63, 0.14, 0.74	0.14, 0.2, 0.17, 0.93	0.14, 0.05, 0.08, 0.96
Train: easy Test: med (3441, 8571)	Train: 1dtdb, test: 1fwp, # pure basins: 13, 240	3.5, 96.4, 1	45.7, 37.9, 82	45.9, 38.6, 81	18.5, 63.6, 10	25.7, 39.9, 17
		1.0, 0.004, 0.01, 0.97	0.23, 0.34, 0.28, 0.95	0.24, 0.34, 0.25, 0.94	0.4, 0.04, 0.08, 0.97	0.23, 0.07, 0.11, 0.97
Train: med Test: med (6435, 8571)	Train: 1c8ca, test: 1fwp, # pure basins: 210, 240	9.61, 64.57, 12	20, 60.3, 35	33.9, 52.7, 65	14.1, 69, 19	2.3, 58.8, 13
		0.41, 0.05, 0.09, 0.97	0.37, 0.15, 0.2, 0.97	0.3, 0.27, 0.26, 0.96	0.42, 0.08, 0.13, 0.97	0.34, 0.05, 0.1, 0.97

Train: med	Train: 1c8ca,	13, 20.4, 3	11.7, 12.3, 3	65.4, 28.4 , 3	60.7, 39 , 3	16, 10.7, 2
Test: med (6435, 276)	test: 1hz6a, # pure basins: 210, 5	0.2, 0.6, 0.3, 0.95	0.18, 0.6, 0.27, 0.94	0.11, 0.7, 0.2, 0.88	0.25, 0.6, 0.35, 0.96	0.13, 0.4, 0.2, 0.94
Train: med	Train: 1c8ca,	23.2, 30.4, 17	100, 55 , 18	94.7, 46 , 17	100 , 66.4 , 18	23.1, 37.2, 16
Test: easy (6435, 5398)	1wapa, # pure basins: 210, 18	0.07, 0.94, 0.13, 0.96	0.05, 1.0, 0.1, 0.94	0.04, 0.99, 0.07, 0.91	0.08, 1.0, 0.15, 0.96	0.13, 0.89, 0.23, 0.98

one dataset, and then do a 60–40 split to obtain a training and a testing dataset. Table 3 shows the utility of under-sampling the dataset. The best performance in terms of n and p is obtained by XGBoost, when the dataset is under-sampled (n is 89% and p is 89%). Good results are also achieved by model RT (n is 89% and p is 75.5%). BB also proves effective for the medium and hard cases (achieving the best and second best results in these categories). Overall, model RT and BB win in this setting, mainly due to the fact that models that focus more on the balancing the data before training (BB, RT, and XGBoost-us) can better address class imbalance in data distribution.

Table 4 shows the classification results in Setting 4. In this setting, we combine all the targets of the same category, leaving out only one target of that category for testing and training on the rest. Table 4 emphasizes the observations revealed by Table 3; when the dataset size is considerably big, balancing the dataset before helps classification models achieve better p and n . Specifically, similar to results related above, BB and RT consistently obtain the best and second best results in n and p for all the cases. They provide purity as high as 85.5% (Row 4) and n as high as 99.3% (Row 2). The lowest purity that these two models obtain is 13.3% and p of 22.4% (hard case, last row), which is not a surprise considering the subpar quality of the decoys in the target 1aoz and comparable with the results obtained by unsupervised learning (related above).

Table 5 shows the results of the regression models. In this setting, the goal is to predict the purity of the target test case. Table 5 shows the effectiveness of the boosting technique, revealed by XGBoost. In most of the cases, the lowest RMSE and MAE are obtained by XGBoost. The lowest MAE (0.0092) results on the

Table 3

n proportion of true positives, *p* purity, hit number of true pure basins detected, *pr* precision, *r* recall, *f* f-measure, *acc* accuracy

	Train, test, # pure basins: test	XGBoost	XGBoost -us	XGBoost -us-avg	BB	RT
		n, p, hits	n, p, hits	n, p, hits	n, p, hits	n, p, hits
		pr, r, f, acc	pr, r, f, acc	pr, r, f, acc	pr, r, f, acc	pr, r, f, acc
Easy	Train: 75% test: 17924	97.9, 40.6, 59	89, 89, 79	88.2, 44.5, 70	98.8, 40.2, 73	89, 75.5, 37
		0.11, 0.67, 0.18, 0.91	0.06, 0.9, 0.11, 0.78	0.05, 0.8, 0.1, 0.78	0.1, 0.83, 0.19, 0.89	0.44, 0.42, 0.43, 0.98
Medium	Train: 75% test: 30097	53.9, 11.7, 209	55, 12.8, 230	73.8, 16.7, 301	35.1, 19.7, 137	25, 11.3, 135
		0.11, 0.54, 0.19, 0.82	0.12, 0.59, 0.2, 0.81	0.13, 0.8, 0.22, 0.78	0.17, 0.35, 0.23, 0.91	0.14, 0.35, 0.2, 0.89
Hard	Train: 75% test: 29889	59.3, 11.1, 149	61.7, 10.4, 148	63.5, 10.6, 152	49, 12.4, 138	27.3, 12.8, 91
		0.05, 0.7, 0.09, 0.69	0.05, 0.69, 0.09,	0.05, 0.71, 0.09,	0.06, 0.64, 0.11,	0.07, 0.42, 0.12,
				0.68	0.68	0.77
						0.86

easy cases (train: 1dtdb, test: 1wapa), which reiterate the fact that these targets are easy for prediction due to the quality of Rosetta-generated decoys for them. Although XGBoost provides superior results, SVR proves its utility in medium cases by providing better MAE.

Table 4

n proportion of true positives, *p* purity, hit number of true pure basins detected, *pr* precision, *r* recall, *f* f-measure, *acc* accuracy

Training models and size	Test size and model, # pure basins: train, test	XGBoost	XGBoost -us	XGBoost -us-avg	BB	RT
		n, p, hits	n, p, hits	n, p, hits	n, p, hits	n, p, hits
		pr, r, f, acc	pr, r, f, acc	pr, r, f, acc	pr, r, f, acc	pr, r, f, acc
Train size: 17896, easy: 1dtdb, 1wapa, 1dtja, 1tig	Test size: 6003, easy: 1ail, # pure basins: 299, 46	95, 22.6, 33	95.6, 20.5, 35	96, 13.5, 32	94.5, 26.4, 35	92.8, 27.6, 25
		0.12, 0.72, 0.2, 0.96	0.1, 0.76, 0.17, 0.94	0.03, 0.7, 0.05, 0.79	0.12, 0.76, 0.21, 0.96	0.2, 0.54, 0.3, 0.98
Train size: 20458, easy: 1ail, 1wapa, 1dtja, 1tig	Test size: 3441, easy: 1dtdb, # pure basins: 332, 13	99.8, 47.3, 13	99.4, 68.6, 12	99.6, 40.4, 12	99.5, 63.8, 12	99.3, 73.9, 11
		0.02, 1.0, 0.04, 0.84	0.06, 0.92, 0.11, 0.94	0.01, 0.95, 0.02, 0.62	0.05, 0.92, 0.1, 0.94	0.08, 0.85, 0.15, 0.96
Train size: 39854, medium: 1c8ca, 2ci2, 1bq9, 1fwp, 1sap, 1hhp	Test size: 276: medium: 1hz6a, # pure basins: 1613, 5	57.7, 44.5, 2	59.9, 31.4, 2	83, 15.4, 3	55.5, 85.5, 1	55.5, 72.8, 1
		0.5, 0.4, 0.44, 0.98	0.09, 1.0, 0.14, 0.91	0.04, 0.76, 0.08, 0.69	1.0, 0.2, 0.33, 0.99	0.5, 0.2, 0.3, 0.98
Train size: 34372, medium:	Test size: 5718,	55.4, 28.7, 20	78.6, 12.6, 37	65.9, 24.1, 28	46, 43, 12	51.7, 16.2, 11
1c8ca,2ci2, 1bq9,1fwp, 1sap, 1hz6a	medium: 1hhp, # pure basins: 1551, 67	0.19, 0.29, 0.23, 0.98	0.1, 0.55, 0.16, 0.93	0.09, 0.42, 0.14, 0.82	0.2, 0.18, 0.19, 0.98	0.08, 0.16, 0.11, 0.97
Train size: 35729, hard: 2h5nd, 2ezk, 1cc5, 1isua, 1aly	Test size: 4763, hard: 1aoy, # pure basins: 607, 228	59.5, 9.6, 151	70.6, 10.2, 161	70.2, 9.9, 176	22.4, 13.3, 50	7.7, 11.6, 14
		0.04, 0.66, 0.08, 0.30	0.05, 0.71, 0.09, 0.28	0.05, 0.77, 0.08, 0.21	0.07, 0.22, 0.11, 0.83	0.06, 0.06, 0.06, 0.91

Table 5
RMSE and MAE of regression models

Train and test models (data size)	SVR	XGBoost
Training: Easy cases, test: Easy and medium cases		
Train: easy, 1dtdb, 60% (2064)	RMSE: 0.0757	RMSE: 0.1175
Test: easy, 1dtdb, 40% (1377)	MAE: 0.0352	MAE: 0.0771
Train: easy, 1dtdb (3441) Test: easy, 1wapa (5398)	RMSE: 0.0734 MAE: 0.0187	RMSE: 0.0562 MAE: 0.0092
Train: easy, 1dtdb (3441) Test: easy, 1dtja (3863)	RMSE: 0.2002 MAE: 0.0564	RMSE: 0.1957 MAE: 0.0487
Train: easy, 1dtdb (3441) Test: medium, 1c8ca (6435)	RMSE: 0.2148 MAE: 0.0780	RMSE: 0.2097 MAE: 0.0749
Training: Medium case, test: easy and medium cases		
Train: medium, 1c8ca, 60% (3861)	RMSE: 0.2122	RMSE: 0.1953
Test: medium, 1c8ca, 40% (2574)	MAE: 0.0750	MAE: 0.1078
Train: medium, 1c8ca (6435)	RMSE: 0.2036	RMSE: 0.1716
Test: medium, 1fwp (8571)	MAE: 0.0789	MAE: 0.0750
Train: medium, 1c8ca (6435)	RMSE: 0.1700	RMSE: 0.1525
test: medium, 1hz6a (276)	MAE: 0.0545	MAE: 0.0814
Train: medium, 1c8ca (6435)	RMSE: 0.1073	RMSE: 0.1070
test: medium, 1sap (5096)	MAE: 0.0403	MAE: 0.0915
Train: medium, 1c8ca (6435)	RMSE: 0.1034	RMSE: 0.0911
test: easy, 1dtdb (3441)	MAE: 0.0317	MAE: 0.0626

Altogether, the results reported in Tables 2, 3, and 5 suggest the promise that supervised learning methods hold in basin-based decoy selection. In many cases, the results are comparable with those obtained via unsupervised learning.

4 Notes

The findings we relate in this paper show the utility of leveraging the organization of protein structure spaces or energy landscapes to learn which structural states are relevant for function. In particular, better results are obtained in the strict context of decoy selection when leveraging the organization of the energy landscape rather than ignoring energy and conducting all analysis on the structure space. In addition to evaluating performance in the context of unsupervised learning, this paper evaluates several supervised learning methods under the umbrella of both classification and regression. Though a very challenging setting due to the class imbalance distribution and the varying quality of decoys, the presented results show that in many cases, supervised learning methods achieve the same performance as unsupervised learning ones. It should be noted that the supervised learning methods we adopt here to assess their utility in decoy selection are pretty simple. However, even these simple methods are able to provide us with competitive results, suggesting further research in this direction is promising.

The work opens several lines of enquiry. For instance, the decoys in identified communities or basins can be further assessed by different scoring functions for indicators of nativeness. Community detection itself can be improved and lifted from the structure space to the energy landscape, by integrating energy in the construction of the nngraph. While the presented work focuses on an application in template-free protein structure prediction, the work may be useful in other settings where an organization of uncomplexed or complexed molecular structure data promises to reveal functionally relevant structural states captured in silico.

References

- Boehr DD, Wright PE (2008) How do proteins interact? *Science* 320(5882):1429–1430
- Maximova T, Moffatt R, Ma B, Nussinov R, Shehu A (2016) Principles and overview of sampling methods for modeling macromolecular structure and dynamics. *PLoS Comp Biol* 12(4):e1004619
- Leaver-Fay A et al (2011) ROSETTA3: an object-oriented software suite for the simulation and design of macromolecules. *Methods Enzymol* 487:545–574
- Xu D, Zhang Y (2012) Ab initio protein structure assembly using continuous structure fragments and optimized knowledge-based force field. *Proteins: Struct Funct Bioinf* 80 (7):1715–1735. <https://doi.org/10.1002/prot.24065>
- Olson B, Shehu A (2013) Multi-objective stochastic search for sampling local minima in the

- protein energy surface. In: ACM conference on bioinformatics, computational biology (BCB), Washington, DC, pp 430–439
6. Clausen R, Shehu A (2014) A multiscale hybrid evolutionary algorithm to obtain sample-based representations of multi-basin protein energy landscapes. In: ACM conference on bioinformatics, computational biology (BCB), Newport Beach, CA, pp 269–278
 7. Shehu A, Plaku E (2016) A survey of computational treatments of biomolecules by robotics-inspired methods modeling equilibrium structure and dynamics. *J Artif Intell Res* 59:509–572
 8. Shehu A, Clementi C, Kavraki LE (2007) Sampling conformation space to model equilibrium fluctuations in proteins. *Algorithmica* 48 (4):303–327
 9. Okazaki K, Koga N, Takada S, Onuchic JN, Wolynes PG (2006) Multiple-basin energy landscapes for large-amplitude conformational motions of proteins: structure-based molecular dynamics simulations. *Proc Natl Acad Sci U S A* 103(32):11844–11849
 10. Boehr DD, Nussinov R, Wright PE (2009) The role of dynamic conformational ensembles in biomolecular recognition. *Nat Chem Biol* 5 (11):789–796
 11. Nussinov R, Wolynes PG (2014) A second molecular biology revolution? The energy landscapes of biomolecular function. *Phys Chem Chem Phys* 16(14):6321–6322
 12. Frauenfelder H, Sligar SG, Wolynes PG (1991) The energy landscapes and motion on proteins. *Science* 254(5038):1598–1603
 13. Bryngelson JD, Onuchic JN, Socci ND, Wolynes PG (1995) Funnels, pathways, and the energy landscape of protein folding: a synthesis. *Proteins Struct Funct Genet* 21 (3):167–195
 14. Shehu A (2015) A review of evolutionary algorithms for computing functional conformations of protein molecules. In: Zhang W (ed) Computer-aided drug discovery, Springer methods in pharmacology and toxicology series
 15. Samoilenco S (2008) Fitness landscapes of complex systems: insights and implications on managing a conflict environment of organizations. *Complex Organ* 10(4):38–45
 16. Kryshtafovych A, Fidelis K, Tramontano A (2011) Evaluation of model quality predictions in CASP9. *Proteins* 79(Suppl 10):91–106
 17. Kryshtafovych A, Barbato A, Fidelis K, Monastyrskyy B, Schwede T, Tramontano A (2014) Assessment of the assessment: evaluation of the model quality estimates in CASP10. *Proteins* 82(Suppl 2):112–126
 18. Moult J, Fidelis K, Kryshtafovych A, Schwede T, Tramontano A (2014) Critical assessment of methods of protein structure prediction (CASP)—round X. *Proteins: Struct Funct Bioinf* 82:109–115
 19. Moult J, Fidelis K, Kryshtafovych A, Schwede T, Tramontano A (2018) Critical assessment of methods of protein structure prediction (CASP)—round XII. *Proteins* 86 (Suppl 1):7–15. <https://doi.org/10.1002/prot.25415>
 20. Uziela K, Wallner B (2016) Proq2: estimation of model accuracy implemented in rosetta. *Bioinformatics* 32(9):1411–1413
 21. Liu T, Wang Y, Eickholt J, Wang Z (2016) Benchmarking deep networks for predicting residue-specific quality of individual protein models in casp11. *Sci Rep* 6(19):301
 22. Ginalski K, Elofsson A, Fischer D, Rychlewski L (2003) 3D-Jury: a simple approach to improve protein structure predictions. *Bioinformatics* 19(8):1015–1018
 23. Wallner B, Elofsson A (2006) Identification of correct regions in protein models using structural, alignment, and consensus information. *Protein Sci* 15(4):900–913
 24. Lorenzen S, Zhang Y (2007) Identification of near-native structures by clustering protein docking conformations. *Proteins* 68 (1):187–194
 25. Zhang Y, Skolnick J (2004) Spicker: a clustering approach to identify near-native protein folds. *J Comput Chem* 25(6):865–871
 26. Molloy K, Saleh S, Shehu A (2013) Probabilistic search and energy guidance for biased decoy sampling in ab-initio protein structure prediction. *IEEE/ACM Trans Bioinf Comput Biol* 10(5):1162–1175
 27. Shehu A (2013) Probabilistic search and optimization for protein energy landscapes. In: Aluru S, Singh A (eds) *Handbook of computational molecular biology*, Chapman & Hall/CRC Computer & Information Science Series-Boca Raton
 28. Guan W, Ozakin A, Gray A, et al (2011) Learning protein folding energy functions. In: International conference data mining. IEEE, pp 1062–1067
 29. Jing X, Wang K, Lu R, Dong Q (2016) Sorting protein decoys by machine-learning-to-rank. *Sci Rep* 6(31):571
 30. He Z, Alazmi M, Zhang J, Xu D (2013) Protein structural model selection by combining consensus and single scoring methods. *PLoS One* 8(9):e74006

31. Pawlowski M, Kozlowski L, Kloczkowski A (2016) Mqapsingle: a quasi single-model approach for estimation of the quality of individual protein structure models. *Proteins* 84(8):1021–1028
32. Cao R, Wang Z, Wang Y, Cheng J (2014) Smoq: a tool for predicting the absolute residue-specific quality of a single protein model with support vector machines. *BMC Bioinform* 15(1):120
33. Nguyen SP, Shang Y, Xu D (2014) Dl-pro: a novel deep learning method for protein model quality assessment. In: International conference on neural networks (IJCNN). IEEE, pp 2071–2078
34. Manavalan B, Lee J, Lee J (2014) Random forest-based protein model quality assessment (rfmq) using structural features and potential energy terms. *PLoS One* 9(9):e106542
35. Chatterjee S, Ghosh S, Vishveshwara S (2013) Network properties of decoys and casp predicted models: a comparison with native protein structures. *Mol BioSyst* 9(7):1774–1788
36. Mirzaei S, Sidi T, Keasar C, Crivelli S (2016) Purely structural protein scoring functions using support vector machine and ensemble learning. In: IEEE/ACM transactions on computational biology and bioinformatics
37. Kabsch W (1976) A solution for the best rotation to relate two sets of vectors. *Acta Cryst* A32:922–923
38. Yang Z, Algesheimer R, Tessone CJ (2016) A comparative analysis of community detection algorithms on artificial networks. *Sci Rep* 6(30):750
39. Cazals F, Dreyfus T (2017) The structural bioinformatics library: modeling in biomolecular science and beyond. *Bioinformatics* 33(7):997–1004
40. Zhou ZH (2012) Ensemble methods: foundations and algorithms. CRC Press, Boca Raton
41. Friedman JH (2001) Greedy function approximation: a gradient boosting machine. *Ann Stat* 29(5):1189–1232
42. Chen T, Guestrin C (2016) Xgboost: a scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining. ACM, pp 785–794
43. Tomek I (1976) Two modifications of CNN. *IEEE Trans Syst Man Cybernet* 6:769–772
44. Akhter N, Shehu A (2017) From extraction of local structures of protein energy landscapes to improved decoy selection in template-free protein structure prediction. *Molecules* 23(1):216
45. Berman HM, Henrick K, Nakamura H (2003) Announcing the worldwide Protein Data Bank. *Nat Struct Biol* 10(12):980–980
46. Yang J, Leskovec J (2012) Defining and evaluating network communities based on ground-truth. In: International conference on data mining (ICDM), pp 745–754
47. Bastian M, Heymann S, Jacomy M (2009) Gephi: an open source software for exploring and manipulating networks. In: International AAAI conference on weblogs and social media. AAS, pp 361–362
48. Jacomy M, Venturini T, Heymann S, Bastian M (2014) ForceAtlas2, a continuous graph layout algorithm for handy network visualization designed for the Gephi software. *PLoS One* 9(6):e98679



Chapter 9

Exploring Protein Supersecondary Structure Through Changes in Protein Folding, Stability, and Flexibility

Douglas E. V. Pires, Carlos H. M. Rodrigues, Amanda T. S. Albanaz,
Malancha Karmakar, Yoochan Myung, Joicymara Xavier,
Eleni-Maria Michanetzi, Stephanie Portelli, and David B. Ascher

Abstract

The ability to predict how mutations affect protein structure, folding, and flexibility can elucidate the molecular mechanisms leading to disruption of supersecondary structures, the emergence of phenotypes, as well guiding rational protein engineering. The advent of fast and accurate computational tools has enabled us to comprehensively explore the landscape of mutation effects on protein structures, prioritizing mutations for rational experimental validation.

Here we describe the use of two complementary web-based *in silico* methods, DUET and DynaMut, developed to infer the effects of mutations on folding, stability, and flexibility and how they can be used to explore and interpret these effects on protein supersecondary structures.

Key words Missense mutations, Protein stability and folding, Machine learning, Normal mode analysis, Graph-based signatures, DUET, DynaMut

1 Introduction

Proteins are marginally stable, versatile macromolecules involved in a large variety of biochemical processes which are strictly linked and regulated by their native conformation. Mutations leading to changes in protein folding, stability, and conformation can have large phenotypic consequences, responsible for the development of many genetic disorders [1–14], including cancers, and even responsible for changes in drug susceptibility [15–27]. While these effects are commonly thought about in terms of reduced protein stability, mutations leading to increased stability and rigidification of the molecule can be equally deleterious. Maintaining, or enhancing, protein stability, and the identification of mutations that do not negatively affect protein stability, also remains one of the most difficult and important challenges in protein engineering.

While experimental validation of protein thermodynamic parameters remains a laborious task, the development of novel robust and scalable computational methods (Table 1) has allowed for the evaluation of the complete landscape of structural effects of mutations in a protein system and their effects on protein stability and flexibility within minutes, enabling rapid mutation prioritization.

Using the concept of graph-based signatures, we have developed robust methods for quantitatively analyzing effects of single missense mutations on protein stability, flexibility, and interactions [9, 28–37]. DUET [37] (<http://biosig.unimelb.edu.au/duet>) is a machine learning-based approach that integrates and optimizes two complementary methods in an optimized predictor (mCSM-Stability [36] and SDM [38]) using support vector machines. This method enables the accurate assessment of the effects of mutations on protein folding and stability. DynaMut [28] (<http://biosig.unimelb.edu.au/dynamut>) is a novel method that takes into account molecular motions and, by combining the graph-based signatures with coarse-grained normal mode analysis, generates a consensus prediction of effects of mutations on the protein conformational repertoire. These methods together compose a powerful platform that allows users to navigate the landscape of mutations effects on folding, stability, and flexibility.

2 Materials

DUET and DynaMut are structure-based methods for assessing effects of single-point missense mutations on protein stability/folding and protein flexibility/conformation, respectively. For both methods, users are required to provide:

1. Wild-type protein structure in PDB format: For both methods, a wild-type structure of the protein of interest in the Protein Data Bank [39] format (.pdb) must be provided to perform the predictions. This can be either (a) an experimentally solved structure, with previously solved structures available in the Protein Data Bank, or (b) a model, for instance, obtained via comparative homology modeling (*see Note 1* on how to deal with oligomeric structures). We have previously shown that using homology models built using templates down to 25% sequence identity does not significantly reduce predictive performance of either method (*see Note 2*). Users have the option to either upload the structure file or provide the PDB accession code when they wish to use an experimental structure previously deposited into the PDB (<http://www.rcsb.org> or <http://www.ebi.ac.uk/pdbe/>) (*see Note 3*).
2. Mutation information: The user also needs to supply information on the mutation or mutations they wish to analyze,

Table 1

List of freely available web servers and software for predicting effects of single-point mutations on protein folding, thermostability, and flexibility

	Method	Technique	Data set	Correlation	DOI	Publication year
Folding	mCSM-Stability	Structural signatures	ProTherm—351 mutations	0.73	https://doi.org/10.1093/bioinformatics/btt691	2014
	SDM2	Environment-specific substitution tables	ProTherm—351 mutations	0.61	https://doi.org/10.1093/nar/gkx439	2017
	DUET	Integrated approach	ProTherm—351 mutations	0.71	https://doi.org/10.1093/nar/gku411	2014
	Eris	Physical force field with atomic modeling	ProTherm—351 mutations	0.35	https://doi.org/10.1038/nmeth0607-466	2007
	I-Mutant 2.0	Neighboring residue composition	ProTherm—351 mutations	0.29	https://doi.org/10.1093/nar/gki375	2005
	Auto-Mute	Delaunay tessellation	ProTherm—351 mutations	0.46	https://doi.org/10.1155/2014/278385	2014
	CUPSAT	Atom potentials and torsion angle potentials	ProTherm—351 mutations	0.37	https://doi.org/10.1093/nar/gkl190	2006
	MAESTRO	Statistical scoring functions	ProTherm—351 mutations	0.70	https://doi.org/10.1186/s12859-015-0548-6	2015
	FoldX	Empirical full-atom force field	ProTherm—351 mutations	0.35	https://doi.org/10.1093/nar/gki387	2005
	PoPMuSiC	Statistical potentials and neural networks	ProTherm—351 mutations	0.67	https://doi.org/10.1186/1471-2105-12-151	2011
Thermal stability	NeEMO	Residue interaction networks	ProTherm—351 mutations	0.67	https://doi.org/10.1186/1471-2164-15-S4-S7	2014
	HoTMuSiC	Statistical potentials	ProTherm—1626 mutations	0.59	https://doi.org/10.1038/srep23257	2015
	FireProt	Structural and evolutionary information	ProTherm—1152 mutations	87% precision	https://doi.org/10.1093/nar/gkx285	2017
Flexibility	DynaMut	Structural signatures and NMA	ProTherm (2004)—351 mutations	0.69	https://doi.org/10.1093/nar/gky300	2018

including (1) the chain identifier (one-letter code of the chain, which corresponds to the 22nd column of the coordinate section in the PDB file where the mutation occurs) (*see Note 1*) and (2) the mutation code, which consists of the one-letter amino acid residue code of the wild-type residue, the residue number position as in the PDB file (columns 23–26 of the coordinate section), and the one-letter code of the mutated residue (e.g., R282W denotes a mutation from arginine to tryptophan at residue position 282).

3 Methods

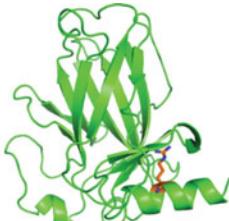
3.1 Predicting and Analyzing Effects of Mutation on Protein Stability and Folding with DUET

1. DUET is freely available as a user-friendly web interface and is compatible with most operating systems and browsers. Open up the prediction server, <http://biosig.unimelb.edu.au/duet/stability>, on a web browser of your preference.
2. Provide the wild-type protein structure of interest by either uploading a PDB file or supplying a valid four-letter PDB accession code (Fig. 1a).
3. DUET offers users the option of two prediction modes, (a) assessing stability effects of a single mutation or (b) systematically evaluating all possible mutations at a given residue position. For a single mutation, users need to provide the mutation information and the mutation chain. For systematic evaluation, the one-letter code of the mutated residue is omitted.

3.2 DUET Prediction Output

1. If a single mutation is provided, after processing, the results page is shown (Fig. 1b), which includes information about the mutation and the predicted effects on stability for DUET and for the individual methods (mCSM-Stability and SDM). An interactive molecular visualization is also shown, allowing users to inspect the wild-type residue environment.
2. For systematic evaluation of a given residue, the predicted effects on protein stability for all 19 possible mutations are shown in tabular format (Fig. 1c).
3. Predicted effects are given as the change in Gibbs Free Energy, $\Delta\Delta G$ (kcal/mol), with negative values denoting destabilizing mutations and positive values, stabilizing ones. While users should interpret the values in the context of the protein system being studied, previous studies have used a rule of thumb that highly destabilizing/stabilizing mutations are those with a predicted $|\Delta\Delta G| > 1.0$ kcal/mol; and moderately destabilizing/stabilizing mutations are those with a predicted $|\Delta\Delta G|$ between 0.5 and 1.0. *See Notes 4 and 5 for further information on how to interpret results.*

A



Run example

Step 1: Please provide a wild-type structure (PDB format)

Description

Upload your own structure:
 No file chosen

OR

Provide a 4-letter PDB code:
 (Example: 2OCJ)

Step 2: Please provide the mutation information

Description

Single mutation

Mutation (Example: I232T)

Mutation chain (Example: A)

Systematic

Residue (Example: I232)

Mutation chain (Example: A)

Submit

Submit

B

mCSM Predicted Stability Change ($\Delta\Delta G$):
-2.365 Kcal/mol (Destabilizing)

SDM Predicted Stability Change ($\Delta\Delta G$):
-3.61 Kcal/mol (Destabilizing)

DUET Predicted Stability Change ($\Delta\Delta G$):
-2.491 Kcal/mol (Destabilizing)

Mutation:
Wild-type: ILE
Position: 232
Mutant-type: THR
Chain: A
Secondary structure: Loop or irregular

4

3

5

View

Rotate
Translate
Zoom
Slab

Reset view

C

Predicted Stability Change ($\Delta\Delta G$):

10 records per page

Search:

Index	Chain	Wild Residue	Residue Position	Mutant Residue	RSA (%)	mCSM predicted $\Delta\Delta G$	SDM predicted $\Delta\Delta G$	DUET predicted $\Delta\Delta G$
1	A	I	232	A	9.2	-2.372	-4.27	-3.071
2	A	I	232	V	9.2	-1.408	-1.91	-1.588
3	A	I	232	L	9.2	-0.959	-0.58	-0.737
4	A	I	232	G	9.2	-2.871	-2.05	-3.22
5	A	I	232	S	9.2	-2.694	-2.55	-2.879
6	A	I	232	W	9.2	-1.759	-1.16	-1.696
7	A	I	232	T	9.2	-2.365	-1.53	-2.343
8	A	I	232	Q	9.2	-1.943	-1.25	-1.832
9	A	I	232	E	9.2	-2.167	-0.84	-1.994
10	A	I	232	C	9.2	-1.509	-1.31	-1.559

Showing 1 to 10 of 19 entries

← Previous 1 2 Next →

6

Run another prediction

Download mutant PDB file

Molecule Visualization

Fig. 1 DUET submission and results web interface. (a) The submission page allows users to either provide its own PDB file or inform an accession code of a protein of interest (7). Users have the option to analyze a

3.3 Predicting and Analyzing Effects of Mutations on Protein Flexibility and Conformation with DynaMut

1. As with DUET, DynaMut predicted changes upon mutation in protein stability are presented as a change in the Gibbs Free Energy of folding and stability ($\Delta\Delta G$ in kcal/mol), calculated as the difference between the wild-type and mutant proteins: $\Delta\Delta G = \Delta G_{wt} - \Delta G_{mt}$. A positive value denotes a stabilizing mutation, while a negative value denotes a destabilizing one. The DynaMut consensus prediction uses both normal mode analysis and graph-based signatures to more accurately identify stabilizing mutations, a limitation of other published approaches (Fig. 2b).
2. DynaMut is also freely available for use freely as a user-friendly web interface. In order to run a prediction, open up the DynaMut prediction page at <http://biosig.unimelb.edu.au/dynamut/prediction> on a web browser of your preference (the web server is compatible with the most common operating systems and browsers).
3. Users have the option to either evaluate a single mutation or provide a text file with a list of mutations to be evaluated in the same format discussed above to run DUET (Fig. 2a). There are no limits on the number of mutations that can be analyzed.
4. For both predictions modes, users are required to provide the wild-type protein structure of interest by either uploading a PDB file or supplying a valid four-letter code PDB accession code of a deposited experimental structure (Fig. 2a).

3.4 DynaMut Prediction Output

1. Prediction results: DynaMut will present the results under three main separate tabulated headings: (1) variation of Gibbs Free Energy predictions, (2) interatomic interactions, and (3) deformation/fluctuation analysis. See Notes 4 and 5 for further information on how to interpret results.
2. DynaMut also graphically displays the resulting change in vibrational energy between the wild-type and mutant structures (Fig. 2b). This highlights regions predicted to be more flexible (red) or less flexible (blue) upon mutation. All calculations and representations can be downloaded through links located at the bottom of the results page.

←

Fig. 1 (continued) specific mutation or perform a systematic analysis of all mutations for a given residue (2). (b) For single-mutation prediction, the mutation identification (3) and the predicted effects on stability are shown (4), as well as an interactive molecular visualization (5). (c) For systematic evaluation of mutation on a given residue, the results are shown in tabular format

A

1

Single Mutation

Provide a wild-type structure*

Submit a molecule in PDB format.

Wild-type (Ex.: 1U46)

No file chosen

OR

PDB Accession
1U46

2

Mutation List ⓘ

Provide a wild-type structure*

Submit a molecule in PDB format.

Wild-type* - PDB format (Ex.: 2XB7)

No file chosen

OR

PDB Accession
2XB7

Mutation details

Mutation*	Chain*
E346K	A

Email ⓘ (optional)
your@email.com

Mutation details

Mutation list file* ⓘ	Chain*
<input type="button" value="Choose File"/> No file chosen	A

Email ⓘ (optional)
your@email.com

B

ΔΔG Predictions Interatomic Interactions Deformation and Fluctuation Analysis

3

Prediction Outcome
ΔΔG: -0.457 kcal/mol (Destabilizing)

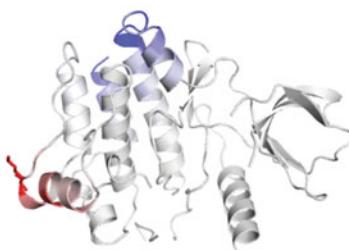
4

NMA Based Predictions
ΔΔG ENCoM: -0.139 kcal/mol (Destabilizing)

Other Structure-Based Predictions

- ΔΔG mCSM: -0.371 kcal/mol (Destabilizing)
- ΔΔG SDM: -0.160 kcal/mol (Destabilizing)
- ΔΔG DUET: -0.203 kcal/mol (Destabilizing)

Δ Vibrational Entropy Energy Between Wild-Type and Mutant
ΔΔS_{vib} ENCoM: 0.174 kcal.mol⁻¹.K⁻¹ (Increase of molecule flexibility)



5

Fig. 2 DynaMut submission and results web interface. (a) The submission page allows for the analysis of a single-point mutation (1) or a list of mutations (2). The main results page (b) depicts the predicted effect of mutation by DynaMut (3) as well as predicted effects by its individual components (4). A depiction of the calculated difference in vibration entropy (5) is also shown

3. When multiple mutations are analyzed, these results are presented in a tabulated format, where users are able to open up and analyze each mutation within the single-mutation analysis result interface.

3.5 Visualizing Effects of Mutations on Protein Structure

1. DynaMut also enables visualization of the effects of a mutation within the wild-type and mutant protein structure (Fig. 3).
2. The interatomic interactions made by the wild-type and mutant residues, calculated using Arpeggio [30] (<http://biosig.unimelb.edu.au/arpegioweb/>), are visually shown. This enables the user to identify how the mutation will affect the local interaction network—important for maintaining protein stability (Fig. 3a).
3. The normal mode analysis predictions are also shown, highlighting changes in vibrational energy between the wild-type and mutant structures (Fig. 3b).
4. All these representations are downloadable as Pymol session files from links at the bottom of the results page.

4 Notes

1. It is important to notice that both methods, DUET and Dyna-Mut, were conceived to analyze monomer structures. In case of analysis of oligomers, users are advised to filter their PDB files prior to submission, filtering chains of interest (for instance, using the PDBest software [40]). The servers will consider all chains submitted; however, a warning message is exhibited. When considering the effects of mutations on oligomeric structures, it is also important to consider the effects of the mutations on the affinity of the monomers to form the oligomer. This can be assessed using mCSM-PPI (http://biosig.unimelb.edu.au/mcsm/protein_protein).
2. The chain ID for the provided PDB file is a mandatory field, and blank characters are not allowed. Some homology modeling tools do not automatically add a chain ID. If this is the case, the user will need to modify the PDB file prior to submission to the servers. There are several tools available to perform this task.¹
3. Another source of error comes from structures with multiple models. It is an important practice to filter NMR structures, selecting a single model.
4. Special cases: Mutations to and from prolines. Prolines are the only amino acid whose amino group is connected to the side chain, which in the context of the peptide bond greatly limits torsional angles. The nature of this residue, therefore, needs to be taken into account while analyzing mutation effects. For instance, (1) mutations to prolines in the middle of alpha-helices can introduce kinks, affecting local structure, and

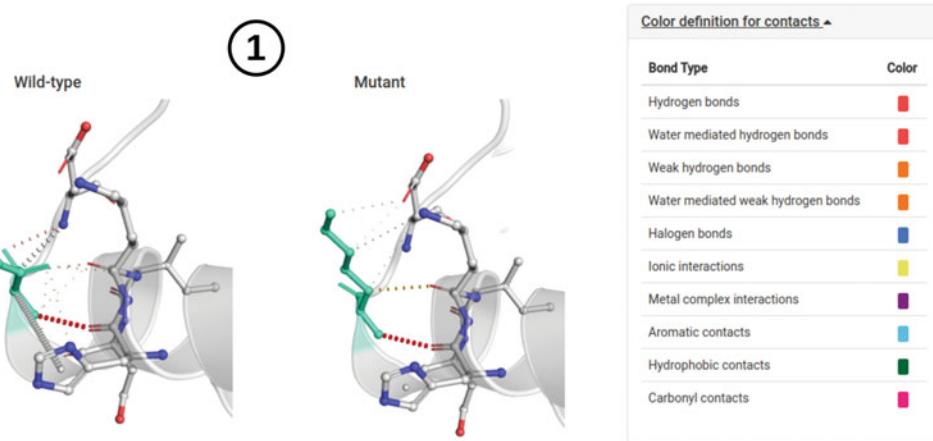
¹ <http://www.canoz.com/sdh/renamelpdbchain.pl>

A

ΔΔG Predictions Interatomic Interactions Deformation and Fluctuation Analysis

2

Prediction of Interatomic Interactions

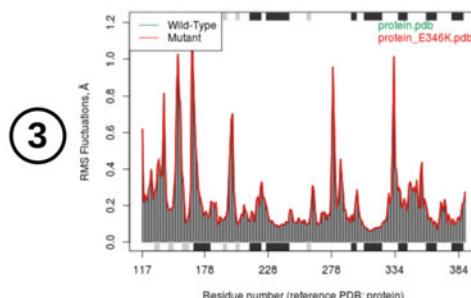


B

ΔΔG Predictions Interatomic Interactions Deformation and Fluctuation Analysis

Ensemble NMA of Wild-type and Mutant

Wild-type and Mutant sequence were extracted from their respective 3D structures and then aligned. The results of normal mode data for each of the sequences are displayed below.



Type of secondary structure on each region of the sequence is added to the top and bottom margins of the plot (helices black and strands gray)

Visual analysis of Atomic Fluctuation

Atomic Fluctuation provides the amplitude of the absolute atomic motion.
Calculations performed over the first 10 non-trivial modes of the molecule.

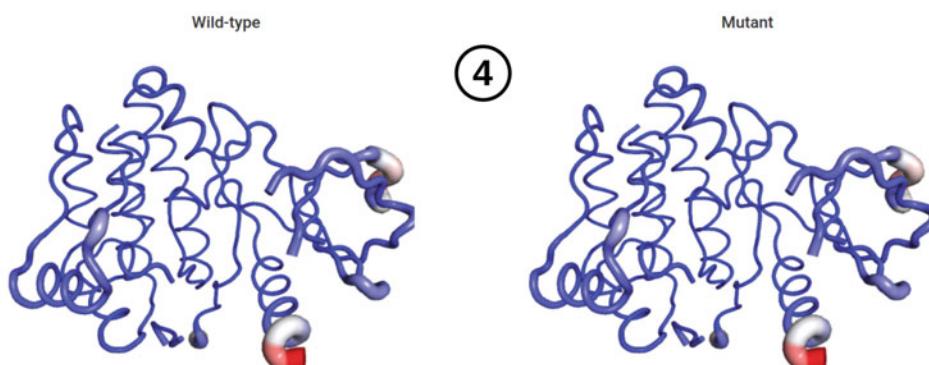


Fig. 3 DynaMut secondary results web interface. (a) A depiction of the calculated interatomic interactions (1) for wild-type and mutant proteins is shown, with interactions identified by color (2). (b) Depicts visualizations of the deformation and fluctuation analysis as fluctuation plot per residue (3) and atomic fluctuation in the context of the structures (4). Figure and individual files (pymol files for molecular visualization) are available for download

(2) since prolines are commonly found in turns and loops, their substitution might interfere with the formation of supersecondary structures such as hairpin loops.

5. Special cases: mutations of positive-phi glycines. Similarly to prolines, positive-phi glycines, while rare in experimental structures, should also be given special consideration due to its torsional angles. Glycines are the only residues capable of adopting positive-phi angles. These glycines are usually conserved across evolution, meaning that mutations of positive-phi glycines tend to be destabilizing.

Acknowledgments

This work was supported by the Australian Government Research Training Program Scholarship [to Y.M., M.K., C.H.M.R. and S.P.]; the Jack Brockhoff Foundation [JBF 4186, 2016 to D.B.A.]; a Newton Fund RCUK-CONFAP Grant awarded by The Medical Research Council (MRC) and Fundação de Amparo à Pesquisa do Estado de Minas Gerais (FAPEMIG) [MR/M026302/1 to D.B.A. and D.E.V.P.]; the National Health and Medical Research Council of Australia [APP1072476 to D.B.A.]; the Victorian Life Sciences Computation Initiative (VLSCI), an initiative of the Victorian Government, Australia, on its Facility hosted at the University of Melbourne [UOM0017]; the Instituto René Rachou (IRR/FIOCRUZ Minas), Brazil, and Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) [to D.E.V.P.]; and the Department of Biochemistry and Molecular Biology, University of Melbourne [to D.B.A.].

References

1. Andrews KA, Ascher DB, Pires DEV, Barnes DR, Vialard L, Casey RT, Bradshaw N, Adlard J, Aylwin S, Brennan P, Brewer C, Cole T, Cook JA, Davidson R, Donaldson A, Fryer A, Greenhalgh L, Hodgson SV, Irving R, Laloo F, McConachie M, McConnell VPM, Morrison PJ, Murday V, Park SM, Simpson HL, Snape K, Stewart S, Tomkins SE, Wallis Y, Izatt L, Goudie D, Lindsay RS, Perry CG, Woodward ER, Antoniou AC, Maher ER (2018) Tumour risks and genotype-phenotype correlations associated with germline variants in succinate dehydrogenase subunit genes SDHB, SDHC and SDHD. *J Med Genet* 55(6):384–394. <https://doi.org/10.1136/jmedgenet-2017-105127>
2. Trezza A, Bernini A, Langella A, Ascher DB, Pires DEV, Sodi A, Passerini I, Pelo E, Rizzo S, Niccolai N, Spiga O (2017) A computational approach from gene to structure analysis of the human ABCA4 transporter involved in genetic retinal diseases. *Invest Ophthalmol Vis Sci* 58 (12):5320–5328. <https://doi.org/10.1167/ iovs.17-22158>
3. Traynelis J, Silk M, Wang Q, Berkovic SF, Liu L, Ascher DB, Balding DJ, Petrovski S (2017) Optimizing genomic medicine in epilepsy through a gene-customized approach to missense variant interpretation. *Genome Res* 27(10):1715–1729. <https://doi.org/10.1101/gr.226589.117>
4. Soardi FC, Machado-Silva A, Linhares ND, Zheng G, Qu Q, Pena HB, Martins TMM, Vieira HGS, Pereira NB, Melo-Minardi RC, Gomes CC, Gomez RS, Gomes DA, Pires DEV, Ascher DB, Yu H, Pena SDJ (2017) Familial STAG2 germline mutation defines a new human cohesinopathy. *NPJ Genom Med*

- 2.7. <https://doi.org/10.1038/s41525-017-0009-4>
5. Ramdzan YM, Trubetskoy MM, Ormsby AR, Newcombe EA, Sui X, Tobin MJ, Bongiovanni MN, Gras SL, Dewson G, Miller JML, Finkbeiner S, Moily NS, Niclis J, Parish CL, Purcell AW, Baker MJ, Wilce JA, Waris S, Stojanovski D, Bocking T, Ang CS, Ascher DB, Reid GE, Hatters DM (2017) Huntington inclusions trigger cellular quiescence, deactivate apoptosis, and lead to delayed necrosis. *Cell Rep* 19(5):919–927. <https://doi.org/10.1016/j.celrep.2017.04.029>
 6. Jubb HC, Pandurangan AP, Turner MA, Ochoa-Montano B, Blundell TL, Ascher DB (2017) Mutations at protein-protein interfaces: small changes over big surfaces have large impacts on human health. *Prog Biophys Mol Biol* 128:3–13. <https://doi.org/10.1016/j.pbiomolbio.2016.10.002>
 7. Chirgadze DY, Ascher DB, Blundell TL, Sibanda BL (2017) DNA-PKcs, allosteric, and DNA double-strand break repair: defining the structure and setting the stage. *Methods Enzymol* 592:145–157. <https://doi.org/10.1016/bs.mie.2017.04.001>
 8. Casey RT, Ascher DB, Rattenberry E, Izatt L, Andrews KA, Simpson HL, Challis B, Park SM, Bulusu VR, Lalloo F, Pires DEV, West H, Clark GR, Smith PS, Whitworth J, Papathomas TG, Taniere P, Savisaar R, Hurst LD, Woodward ER, Maher ER (2017) SDHA related tumorigenesis: a new case series and literature review for variant interpretation and pathogenicity. *Mol Genet Genomic Med* 5(3):237–250. <https://doi.org/10.1002/mgg3.279>
 9. Pires DE, Chen J, Blundell TL, Ascher DB (2016) In silico functional dissection of saturation mutagenesis: Interpreting the relationship between phenotypes and changes in protein stability, interactions and activity. *Sci Rep* 6:19848. <https://doi.org/10.1038/srep19848>
 10. Nemethova M, Radvanszky J, Kadasi L, Ascher DB, Pires DE, Blundell TL, Porfirio B, Mannoni A, Santucci A, Milucci L, Sestini S, Biolcati G, Sorge F, Aurizi C, Aquaron R, Alsbou M, Lourenco CM, Ramadevi K, Ranganath LR, Gallagher JA, van Kan C, Hall AK, Olsson B, Sireau N, Ayooob H, Timmis OG, Sang KH, Genovese F, Imrich R, Rovensky J, Srinivasaraghavan R, Bharadwaj SK, Spiegel R, Zatkova A (2016) Twelve novel HGD gene variants identified in 99 alkaptonuria patients: focus on ‘black bone disease’ in Italy. *Eur J Hum Genet* 24(1):66–72. <https://doi.org/10.1038/ejhg.2015.60>
 11. Usher JL, Ascher DB, Pires DE, Milan AM, Blundell TL, Ranganath LR (2015) Analysis of HGD gene mutations in patients with alkaptonuria from the United Kingdom: identification of novel mutations. *JIMD Rep* 24:3–11. https://doi.org/10.1007/8904_2014_380
 12. Jafri M, Wake NC, Ascher DB, Pires DE, Gentle D, Morris MR, Rattenberry E, Simpson MA, Trembath RC, Weber A, Woodward ER, Donaldson A, Blundell TL, Latif F, Maher ER (2015) Germline mutations in the CDKN2B tumor suppressor gene predispose to renal cell carcinoma. *Cancer Discov* 5(7):723–729. <https://doi.org/10.1158/2159-8290.CD-14-1096>
 13. Hnizda A, Fabry M, Moriyama T, Pachl P, Kugler M, Brinsa V, Ascher DB, Carroll WL, Novak P, Zaliova M, Trka J, Rezacova P, Yang JJ, Veverka V (2018) Relapsed acute lymphoblastic leukemia-specific mutations in NT5C2 cluster into hotspots driving intersubunit stimulation. *Leukemia*. <https://doi.org/10.1038/s41375-018-0073-5>
 14. Sibanda BL, Chirgadze DY, Ascher DB, Blundell TL (2017) DNA-PKcs structure suggests an allosteric mechanism modulating DNA double-strand break repair. *Science* 355 (6324):520–524. <https://doi.org/10.1126/science.aak9654>
 15. Vedithi SC, Malhotra S, Das M, Daniel S, Kishore N, George A, Arumugam S, Rajan L, Ebenezer M, Ascher DB, Arnold E, Blundell TL (2018) Structural implications of mutations conferring rifampin resistance in mycobacterium leprae. *Sci Rep* 8(1):5016. <https://doi.org/10.1038/s41598-018-23423-1>
 16. Karmakar M, Globan M, Fyfe JAM, Stinear TP, Johnson PDR, Holmes NE, Denholm JT, Ascher DB (2018) Analysis of a novel pncA mutation for susceptibility to pyrazinamide therapy. *Am J Respir Crit Care Med*. <https://doi.org/10.1164/rccm.201712-2572LE>
 17. Holt KE, McAdam P, Thai PVK, Thuong NTT, Ha DTMH, Lan NN, Lan NH, Nhu NTQ, Hai HT, Ha VTN, Thwaites G, Edwards DJ, Nath AP, Pham K, Ascher DB, Farrar J, Khor CC, Teo YY, Inouye M, Caws M, Dunstan SJ (2018) Frequent transmission of the Mycobacterium tuberculosis Beijing lineage and positive selection for EsxW Beijing variant in Vietnam. *Nat Genet* 50:849–856
 18. Singh V, Donini S, Pacitto A, Sala C, Hartkoorn RC, Dhar N, Keri G, Ascher DB, Mondesert G, Vocat A, Lupien A, Sommer R, Vermet H, Lagrange S, Buechler J, Warner DF, McKinney JD, Pato J, Cole ST, Blundell TL, Rizzi M, Mizrahi V (2017) The inosine monophosphate dehydrogenase, Guab2, is a

- vulnerable new bactericidal drug target for tuberculosis. *ACS Infect Dis* 3(1):5–17. <https://doi.org/10.1021/acsinfecdis.6b00102>
19. Park Y, Pacitto A, Bayliss T, Cleghorn LA, Wang Z, Hartman T, Arora K, Ioerger TR, Sacchettini J, Rizzi M, Donini S, Blundell TL, Ascher DB, Rhee K, Breda A, Zhou N, Dartois V, Jonnala SR, Via LE, Mizrahi V, Epemolu O, Stojanovski L, Simeons F, Osuna-Cabello M, Ellis L, MacKenzie CJ, Smith AR, Davis SH, Murugesan D, Buchanan KI, Turner PA, Huggett M, Zuccotto F, Rebollo-Lopez MJ, Lafuente-Monasterio MJ, Sanz O, Diaz GS, Lelievre J, Ballell L, Selenski C, Axtman M, Ghidelli-Disse S, Pflaumer H, Bosche M, Drewes G, Freiberg GM, Kurnick MD, Srikanthan M, Kempf DJ, Green SR, Ray PC, Read K, Wyatt P, Barry CE 3rd, Boshoff HI (2017) Essential but not vulnerable: indazole sulfonamides targeting inosine monophosphate dehydrogenase as potential leads against mycobacterium tuberculosis. *ACS Infect Dis* 3(1):18–33. <https://doi.org/10.1021/acsinfecdis.6b00103>
 20. Pandurangan AP, Ascher DB, Thomas SE, Blundell TL (2017) Genomes, structural biology and drug discovery: combating the impacts of mutations in genetic disease and antibiotic resistance. *Biochem Soc Trans* 45(2):303–311. <https://doi.org/10.1042/BST20160422>
 21. Albanaz ATS, Rodrigues CHM, Pires DEV, Ascher DB (2017) Combating mutations in genetic disease and drug resistance: understanding molecular mechanisms to guide drug design. *Expert Opin Drug Discov* 12(6):553–563. <https://doi.org/10.1080/17460441.2017.1322579>
 22. White RR, Ponsford AH, Weekes MP, Rodrigues RB, Ascher DB, Mol M, Selkirk ME, Gygi SP, Sanderson CM, Artavanis-Tsakonas K (2016) Ubiquitin-dependent modification of skeletal muscle by the parasitic nematode, *Trichinella spiralis*. *PLoS Pathog* 12(11):e1005977. <https://doi.org/10.1371/journal.ppat.1005977>
 23. Silvino AC, Costa GL, Araujo FC, Ascher DB, Pires DE, Fontes CJ, Carvalho LH, Brito CF, Sousa TN (2016) Variation in human cytochrome P-450 drug-metabolism genes: a gateway to the understanding of *Plasmodium vivax* relapses. *PLoS One* 11(7):e0160172. <https://doi.org/10.1371/journal.pone.0160172>
 24. Phelan J, Coll F, McNerney R, Ascher DB, Pires DE, Furnham N, Coeck N, Hill-Cawthorne GA, Nair MB, Mallard K, Ramsay A, Campino S, Hibberd ML, Pain A, Rigouts L, Clark TG (2016) Mycobacterium tuberculosis whole genome sequencing and protein structure modelling provides insights into anti-tuberculosis drug resistance. *BMC Med* 14:31. <https://doi.org/10.1186/s12916-016-0575-9>
 25. Kano FS, Souza-Silva FA, Torres LM, Lima BA, Sousa TN, Alves JR, Rocha RS, Fontes CJ, Sanchez BA, Adams JH, Brito CF, Pires DE, Ascher DB, Sell AM, Carvalho LH (2016) The presence, persistence and functional properties of *Plasmodium vivax* duffy binding protein II antibodies are influenced by HLA class II allelic variants. *PLoS Negl Trop Dis* 10(12):e0005177. <https://doi.org/10.1371/journal.pntd.0005177>
 26. Ascher DB, Wielens J, Nero TL, Doughty L, Morton CJ, Parker MW (2014) Potent hepatitis C inhibitors bind directly to NS5A and reduce its affinity for RNA. *Sci Rep* 4:4765. <https://doi.org/10.1038/srep04765>
 27. Hawkey J, Ascher DB, Judd LM, Wick RR, Kostoulias X, Cleland H, Spelman DW, Padiglione A, Peleg AY, Holt KE (2018) Evolution of carbapenem resistance in *Acinetobacter baumannii* during a prolonged infection. *Microb Genom*. <https://doi.org/10.1099/mgen.0.000165>
 28. Rodrigues CHM, Pires DEV, Ascher DB (2018) DynaMut: predicting the impact of mutations on protein conformation, flexibility and stability. *Nucleic Acids Res*. <https://doi.org/10.1093/nar/gky300>
 29. Pires DE, Ascher DB (2017) mCSM-NA: predicting the effects of mutations on protein-nucleic acids interactions. *Nucleic Acids Res* 45:W241–W246. <https://doi.org/10.1093/nar/gkx236>
 30. Jubb HC, Higuero AP, Ochoa-Montano B, Pitt WR, Ascher DB, Blundell TL (2017) Arpeggio: a web server for calculating and visualising interatomic interactions in protein structures. *J Mol Biol* 429(3):365–371. <https://doi.org/10.1016/j.jmb.2016.12.004>
 31. Pires DE, Blundell TL, Ascher DB (2016) mCSM-lig: quantifying the effects of mutations on protein-small molecule affinity in genetic disease and emergence of drug resistance. *Sci Rep* 6:29575. <https://doi.org/10.1038/srep29575>
 32. Pires DE, Ascher DB (2016) CSM-lig: a web server for assessing and comparing protein-small molecule affinities. *Nucleic Acids Res* 44(W1):W557–W561. <https://doi.org/10.1093/nar/gkw390>
 33. Pires DE, Ascher DB (2016) mCSM-AB: a web server for predicting antibody-antigen affinity changes upon mutation with graph-based signatures. *Nucleic Acids Res* 44(W1):

- W469–W473. <https://doi.org/10.1093/nar/gkw458>
34. Pires DE, Blundell TL, Ascher DB (2015) Platinum: a database of experimentally measured effects of mutations on structurally defined protein-ligand complexes. *Nucleic Acids Res* 43(Database issue):D387–D391. <https://doi.org/10.1093/nar/gku966>
35. Pires DE, Blundell TL, Ascher DB (2015) pkCSM: predicting small-molecule pharmacokinetic and toxicity properties using graph-based signatures. *J Med Chem* 58(9):4066–4072. <https://doi.org/10.1021/acs.jmedchem.5b00104>
36. Pires DE, Ascher DB, Blundell TL (2014) mCSM: predicting the effects of mutations in proteins using graph-based signatures. *Bioinformatics* 30(3):335–342. <https://doi.org/10.1093/bioinformatics/btt691>
37. Pires DE, Ascher DB, Blundell TL (2014) DUET: a server for predicting effects of mutations on protein stability using an integrated computational approach. *Nucleic Acids Res* 42. (Web Server issue: W314–W319. <https://doi.org/10.1093/nar/gku411>
38. Pandurangan AP, Ochoa-Montano B, Ascher DB, Blundell TL (2017) SDM: a server for predicting effects of mutations on protein stability. *Nucleic Acids Res* 45:W229–W235. <https://doi.org/10.1093/nar/gkx439>
39. Berman HM, Westbrook J, Feng Z, Gilliland G, Bhat TN, Weissig H, Shindyalov IN, Bourne PE (2000) The protein data bank. *Nucleic Acids Res* 28(1):235–242
40. Goncalves WR, Goncalves-Almeida VM, Arruda AL, Meira W Jr, da Silveira CH, Pires DE, de Melo-Minardi RC (2015) PDBest: a user-friendly platform for manipulating and enhancing protein structures. *Bioinformatics* 31(17):2894–2896. <https://doi.org/10.1093/bioinformatics/btv223>



Chapter 10

Protodomains: Symmetry-Related Supersecondary Structures in Proteins and Self-Complementarity

Philippe Youkharibache

Abstract

We will consider in this chapter supersecondary structures (SSS) as a set of secondary structure elements (SSEs) found in protein domains. Some SSS arrangements/topologies have been consistently observed within known tertiary structural domains. We use them in the context of repeating supersecondary structures that self-assemble in a symmetric arrangement to form a domain. We call them protodomains (or protofolds). Protodomains are some of the most interesting and insightful SSSs. Within a given 3D protein domain/fold, recognizing such sets may give insights into a possible evolutionary process of duplication, fusion, and coevolution of these protodomains, pointing to possible original protogenes. On protein folding itself, pseudosymmetric domains may point to a “directed” assembly of pseudosymmetric protodomains, directed by the only fact that they are tethered together in a protein chain. On function, tertiary functional sites often occur at protodomain interfaces, as they often occur at domain-domain interfaces in quaternary arrangements.

First, we will briefly review some lessons learned from a previously published census of pseudosymmetry in protein domains (Myers-Turnbull, D. et al., J Mol Biol. 426:2255–2268, 2014) to introduce protodomains/protofolds. We will observe that the most abundant and diversified folds, or superfolds, in the currently known protein structure universe are indeed pseudosymmetric. Then, we will learn by example and select a few domain representatives of important pseudosymmetric folds and chief among them the immunoglobulin (Ig) fold and go over a pseudosymmetry supersecondary structure (protodomain) analysis in tertiary and quaternary structures. We will point to currently available software tools to help in identifying pseudosymmetry, delineating protodomains, and see how the study of pseudosymmetry and the underlying supersecondary structures can enrich a structural analysis. This should potentially help in protein engineering, especially in the development of biologics and immunoengineering.

Key words Protein structure, Protodomains, Supersecondary structure, Symmetry, Pseudosymmetry, Immunoengineering, Domains, Fold, Folding, Engineering, Quaternary structure, Immunoglobulins, Sm, Hfq, GPCR, Sweet protein, FN3, Type I cytokine receptor, CHR, IL-2R, IL-21R, GHR, GHbp

Electronic supplementary material: The online version of this chapter (https://doi.org/10.1007/978-1-4939-9161-7_10) contains supplementary material, which is available to authorized users.

1 Introduction

1.1 Structural Protein Domains

Protein domains have been used by nature as building blocks in larger chains and protein complexes. Biologists have used them to build chimeric proteins, following one of nature's paths in fusing domains together assuming a function for each domain. Such is the case of immunotoxins where an antibody-based domain is fused to a bacterial toxin, the first one for binding to a tumor cell surface antigen target and the second one for cell killing [2]. More recently CAR T-cell therapies have made use of CARs (chimeric antigen receptors) that go beyond in the "engineering" of new proteins by fusing domains in a single chain. In the case of CARs, in addition to fused immunoglobulin (Ig) domains (scFv), entirely new domains are composed of subdomains extracted from various T-cell surface proteins (CD28 and/or CD8 and CD3z) in order to retain desired functional properties [3].

Nature has used a protodomain fusion mechanism in a distant evolutionary past, or so it seems, when one observes pseudosymmetric domains. Hence, we can gain insights in domain creation from an analysis of tertiary pseudosymmetry. Many domains have been structurally characterized, so we can not only look at domain creation but domain evolution in terms of the constituting parts. The immunoglobulin fold [4, 5] is at the heart of a very large number of cell surface proteins of the immune systems [6, 7], beyond immunoglobulins themselves. We will review its tertiary symmetry as well as one level of quaternary structure symmetry in the case of CD8, as a revealing example. Also, as we seek to move to lighter therapeutic proteins, from Fabs to Fvs to single Ig domains as antigen binding domains, Ig domain-level pseudosymmetry properties may be able to guide some immunoengineering efforts.

1.2 Domain-Level Pseudosymmetry and Structural Protodomains

1.2.1 Systematic Census of Tertiary Pseudosymmetry

Structural pseudosymmetry in protein domains has been observed very early on, even within the very first protein structures solved, for example, ferredoxin, myohemerythrin, serine and aspartyl proteases, immunoglobulins, the TIM barrels (triose-phosphate isomerase), or the Rossmann fold [8–15]. It is interesting to note that some of these domains that were characterized early turned out to be some of the most diversified and prototypic domains: in the SCOP classification [16, 17], they are noted d.58, a.24, b.47, b.49, b.1, c.1, and c.2, respectively.

Structural pseudosymmetry corroborated observations a decade earlier of possible ancestral gene duplications within today's genes [12–15] and established a basis for interpreting sequence duplication with pseudosymmetry, hence conceptually defining what we now call "protodomains." We recently performed a systematic census of tertiary pseudosymmetry in the currently known universe of protein domains in the PDB database. We found that a

Table 1
Pseudosymmetry for major structural classes and for the most diversified folds

Fold class	# Folds in class	# SFs in class	% SFs with symmetry	Superfolds: most diversified fold in class	# SFs in fold	% SFs with symmetry ^a
A	284	507	19%	a.24	28	57%
B	174	354	25%	b.1	28	39%
C	147	244	17%	c.1	33	36%
D	376	551	14%	d.58	59	58%
F	57	109	24%	f.13 (GPCRs)	1 ^b	N/A
1038	1765	20%				

Fold classes according to SCOP 1.75 (A, all alpha; B, all beta; C, alpha+beta; D, alpha-beta mixed; F, membrane proteins). Total number of folds and superfamilies (SFs) in class, with percentage of SFs deemed symmetrical. “Superfolds”, i.e. folds with the highest number of superfamilies in class, as a measure of their diversification. For each of them the percentage of superfamilies exhibiting pseudosymmetry (these results were obtained computationally using a threshold of 30%, i.e. a minimum of 30% of superfamilies associated with a given fold were found pseudosymmetric (see Ref. 1, Table S2). In that study 1831 superfamilies representing 157,432 domains were used, including Class E, not shown)

^aRepresentatives of superfamilies were used. Pseudosymmetry was detected for a number of them for each fold. With a score of 30% or more the fold is “called” as symmetric. Experience shows that other folds are symmetric but were undetected with the parameters used. An example would be the Hfq/Sm fold and others sharing an SH3 topology (b.34/b.38), which fall under that 30% threshold

^bWe added GPCRs, classified as one fold, one superfamily in SCOP. Technically it could be classified as A: all alfa. It represents a special case of a highly diversified structural domain within a single superfamily with over 800 different GPCRs just in humans and a staggering 2300 hundred in elephants, diversifying ligand binding for a conserved signaling function within cells

significant number of protein domains (folds) exhibit pseudosymmetry. We can decompose such domains into protodomains (prototofolds), i.e., supersecondary structures related by symmetry.

We shall mention here the top five protein fold classes in that study where, on average, 20% of the folds exhibit internal pseudosymmetry (see Table 1 hereafter and Table S2 in [1]). In these classes the most diversified folds, i.e., those with the highest number of functional superfamilies, were all pseudosymmetric: a.24 (four-helix bundle/myohemerythrin), b.1 (immunoglobulin), c.1 (TIM), and d.58 (ferredoxin) in the SCOP classification [16, 17]. In that classification, membrane proteins (Class F) are grouped together yet two-thirds are alpha-helical folds vs. approximately one-third of all beta-sheet folds, with 24% overall exhibiting symmetry. We chose to highlight the 7-transmembrane protein fold (GPCRs) with a different criterion. It is a single fold and family with a conserved signaling function for an astounding ligand diversity. We can call superfolds these highly resilient folds associated with a large number of superfamilies and highly diversified functions.

Pseudo symmetry is a geometrical property. It does however establish a link to folding, evolution, and biological function. The knowledge of protodomains and symmetry operators defines a

pseudosymmetric domain entirely, apart from a variable linker region, most often short, chaining protodomains within a domain. While protein domains are well defined and have been extensively classified through a number of taxonomies (SCOP, CATH, ECOD) [16–19], the underlying protodomains, in the case of pseudosymmetric domains, have not. Hence the first task is to delineate them and analyze them in terms of similarities and differences, through structure-based sequence alignments.

1.3 Symmetry and Self-Association

1.3.1 Quaternary Symmetry and Self-Assembly

Symmetry in quaternary structures has been extensively studied [20–23]. Among the 3D macromolecular structural complexes in the Protein Data Bank (PDB), symmetry is pervasive [20–23]. The PDB (www.rcsb.org) stores all publicly available structures. As of today, it contains 140,000 structures of macromolecular complexes, with 51% of oligomers: 50,600 (38%) of homomers and 18,000 (13%) of heteromers. In terms of quaternary symmetry, ca. 53,000 structures represent symmetric complexes, with close to 42,000 (78%) presenting a cyclic symmetry and 10,000 (19%) presenting a dihedral symmetry. While quaternary cyclic symmetry is observed up to the 39th order (C39), as in the Vault ribonucleoprotein particle (PDBid: 4HL8), the C2 symmetry represents the vast majority of symmetric structures with ca. 32,000 representatives, of which ca. 31,000 are homodimers.

While these numbers correspond to structures obtained to date on all macromolecular complexes, and are not necessarily fully representative of all (fluctuating) protein complexes *in vivo*, they nevertheless indicate a natural principle of self-assembly of macromolecules [24]. It is natural to view quaternary symmetry or pseudosymmetry as a result of oligomerization of homomers or heteromers, demonstrating the propensity of protein domains to self-assemble.

1.3.2 Tertiary PseudoSymmetry and Self-Assembly of Supersecondary Structures

Most known oligomeric protein structures are symmetric or pseudosymmetric and can be classified using closed symmetry groups. The same is true from pseudosymmetric domains, where at least 20% of known protein domains are pseudosymmetric (see Table 1). This reflects a seemingly similar self-association process of protodomains. Of course, protodomains are chained together, and they have little choice but to assemble, yet they favor a pseudosymmetric arrangement, a pseudosymmetric fold. The vast majority of pseudosymmetric tertiary domains exhibits C2 symmetry, as in known quaternary structures. Higher-order symmetries are also observed in tertiary as in quaternary structures. Pseudosymmetry order up to 30 can be found in, for example, Toll-like receptor 8 (PDBid: 4R0A) with 29 repeats and room for an extra one, where each consecutive repeat/protodomain is related by a rotation operation of 12 degrees around a common central axis. Dihedral symmetry is also observed in tertiary as in quaternary structures.

Quaternary symmetry is a geometrical property and results from monomeric proteins self-assembling at the domain level. The same is true from pseudosymmetry domains in terms of the protodomains they are composed of. Analogously, one can also regard pseudosymmetric domains as pseudoquaternary structures and see a continuum in complexity buildup from subdomain to supra-domain organizations, from protodomain to domain assemblies. This parallel also points to a possible ancestral world where protodomains may have oligomerized spontaneously. At the gene level, it is accepted as a duplication-fusion model to lead to pseudosymmetric protein domains [25–27]. A good example of such a possible duplication-fusion event can be seen in comparing semisweet vs. sweet protein domains (Fig. 7). Of course, in today’s genomes and gene organization, it is not straightforward to reconcile protodomains and possible original protogenes. Yet it can be rewarding to analyze pseudosymmetry as a structural property, regardless of genomic organization.

1.3.3 Self-Assembly Is a Universal Molecular Organizational Principle

Self-assembly and the resulting observed symmetry is in fact a property of all biological macromolecules. Symmetry and self-assembly is of course the main characteristic of DNA pairing; in nucleosomes DNA exhibits an exquisite global C₂ symmetry, with the histones assembly exhibiting three levels of C₂ symmetry (Fig. S6). Recent RNA crystal structures also show that several Riboswitches RNAs exhibit symmetry whether at the tertiary or quaternary level [28] (Fig. 9). The active site of the ribosome itself, a remnant of a proto-ribosome in the RNA world, displays pseudosymmetry [29]. Self-assembly, based on non-covalent interactions, can be seen as a principle for complexity buildup of molecular systems of any size. Mimicking biological systems, and beyond molecular chemistry based on the covalent bond, a whole new field of “supramolecular chemistry” has been aiming in the last 20 years at developing highly complex chemical systems from molecular components interacting through non-covalent intermolecular forces [30, 31].

1.4 Analyzing Self-Assembling Supersecondary Structures

A point group symmetry operation between two or more entities establishes a **structural equivalence relation between these entities**. Two residues or sets of residues related by pseudosymmetry in equivalent positions can be analyzed in terms of “internal” sequence conservation (identity, similarity or lack of), structure, and topology. If one assumes a duplication event, then this opens the door to studying the parallel evolution or coevolution of protodomains within a domain and their interfaces. In studying coevolving SSSs and drilling down coevolving SSEs and residues at

equivalent positions, “internal” conservation or nonconservation of residues may be linked to either folding, coevolution of protodomain interfaces, oligomeric interfaces, or function.

Molecular interfaces can vary greatly, but as soon as we look at tertiary or quaternary symmetric arrangements, structurally homologous supersecondary structures emerge. SSSs form interfaces with symmetrically interacting SSEs. Hence protodomains have to be self-complementary where they are in contact. These contacts can vary widely from a few residues to a number of entirely self-complementary/self-interacting SSEs. They are based on nonbonded residue interactions for both alpha and for beta structures. Beta structures have, in addition, a beta strand pairing mechanism through hydrogen bonding at the backbone level, to form beta sheets. We shall see two magnificent examples in the following with the Ig fold (Figs. 1, 2, 3, 4, 5, and 6) and the Sm fold (Figs. 8 and S2). One can use symmetric and pseudosymmetric SSS decomposition at any level of complexity to analyze molecular interfaces and gain knowledge in the determinants of self-assembling systems. Pseudosymmetry and protodomain delineation of protein domains and, beyond, symmetric quaternary organization of biological units lead us to a method to analyze complexity buildup in biological systems through an architectural/organizational principle of protein structure.

2 Materials

For the analysis we need structural **data**, obtained by any structural biology method such as X-ray, NMR, or EM, and **software** tools to analyze them, i.e., dissect them, delineate SSEs and SSSs, and compare them in terms of sequence (1D), topology (2D), and structure (3D).

2.1 Structural Databases

2.1.1 The PDB (Protein Data Bank) and Derived Resources (NCBI Structure)

The main source of protein structure is the PDB, available through worldwide servers in the USA (PDB/RCSB), Europe (PDBe), and Japan (PDBj) [22, 32, 33]. Derived resources such as NCBI Structure (MMDB) integrate structural information with multiple databases on sequence-related information and evolutionary family classifications such as CDD [34] as well as offer structural comparisons (VAST+) across the entire PDB [35].

2.1.2 Structural Taxonomies

The two main structural classifications in use are SCOP [16, 17] and CATH [18]. More recently ECOD has been added [19]. SCOP is based on manual curation, while the others are automated. We use primarily SCOP in this work, yet the lack of automation is an issue in dealing with new structures.

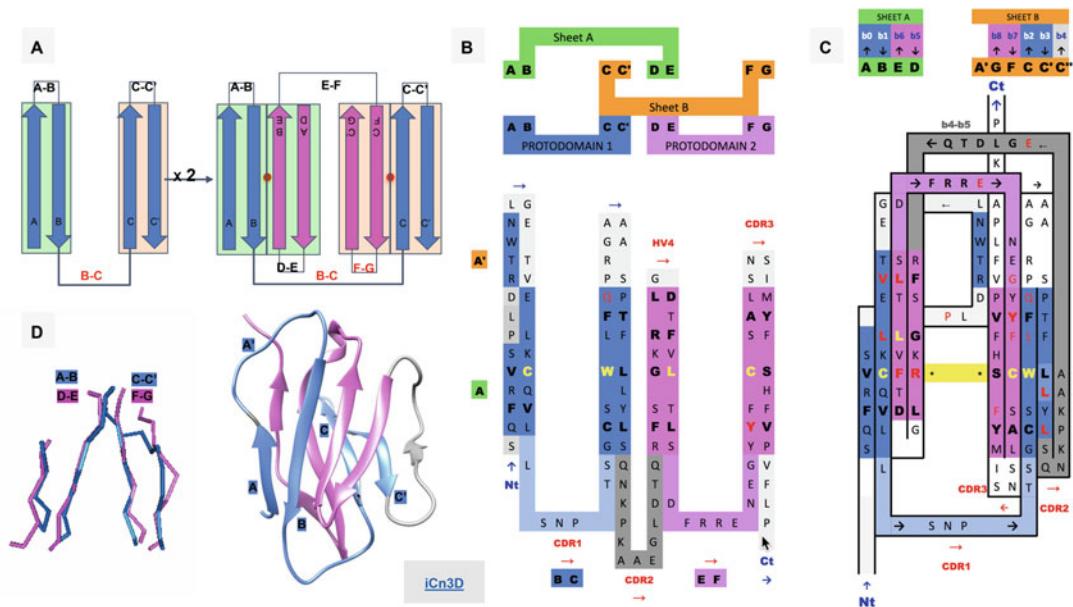


Fig. 1 Ig Greek key protodomain topology, duplication, and symmetric arrangement of protodomains. (a) Idealized Ig protodomain motif topology bb-bb, with 2 beta hairpins connected by a Greek key linker (b, c), duplication, and schematic arrangement: AIB + CIC' = DIE + FIG. Each hairpin theoretically forms to a plane. Each Ig type will present departures from idealized protodomains in their domain context, due to either the protodomain-protodomain linker (not shown here) or some partial structural rearrangement of strand A (see (d) and Fig. 2 for variants in IgV, VNAR, and IgC). Protodomain strands will be displayed blue and magenta for consecutive protodomains 1 and 2, respectively. Planes/Sheets A and B will be consistently shown in green and orange background color. (b) Topology/sequence of consecutive protodomains AIB – CIC' + DIE – FIG. Interestingly, the well-known CDR1 loop in immunoglobulins appears as the Greek key linker between strands B and C, while the CDR2 is formed by linking the two protodomains (as we shall see in Fig. 2 this is where most Ig domains vary depending of the length and shape of this linker, which presents some secondary structure in the case of IgV giving rise to CDR2). (c) 2D Topology/sequence map strand arrangement of protodomains corresponding to the 3D domain C2 symmetry with the formation of symmetry equivalent B<=>E and C<=>F strand-strand protodomain interface, bringing hairpins AIB and DIE in the same sheet (Sheet A or AIB||EID) and correlatively bringing hairpins CIC' and FIG in the same sheet (Sheet B or G|I|I|CIC') facing each other as in a sandwich. A simple 3D rotation through a common axis gives a structural correspondence of the two protodomains AIB – CIC' and DIE – FIG, with a structural alignment (see (d)) varying usually between 1 and 2A in the most distorted cases. The well-known CCW(L) pattern highlighted in yellow is mapped at the protodomain level in symmetrically equivalent positions (see also Fig. 3d). (d) 3D protodomain alignment for a CD8a domain (1CD8) that superimpose with an RMSD of 1.98 (see Fig. 4 for corresponding sequence alignment) showing only structurally aligned residues, with ribbon picture (produced by Chimera [90]) showing strand definitions. Protodomain 1 in blue and protodomain 2 in magenta. Domain visualization with Sheet A in front in the order AIB||EID'. Link to iCn3D <https://d55qc.app.goo.gl/bmCQRj7DWcmqsmna6>

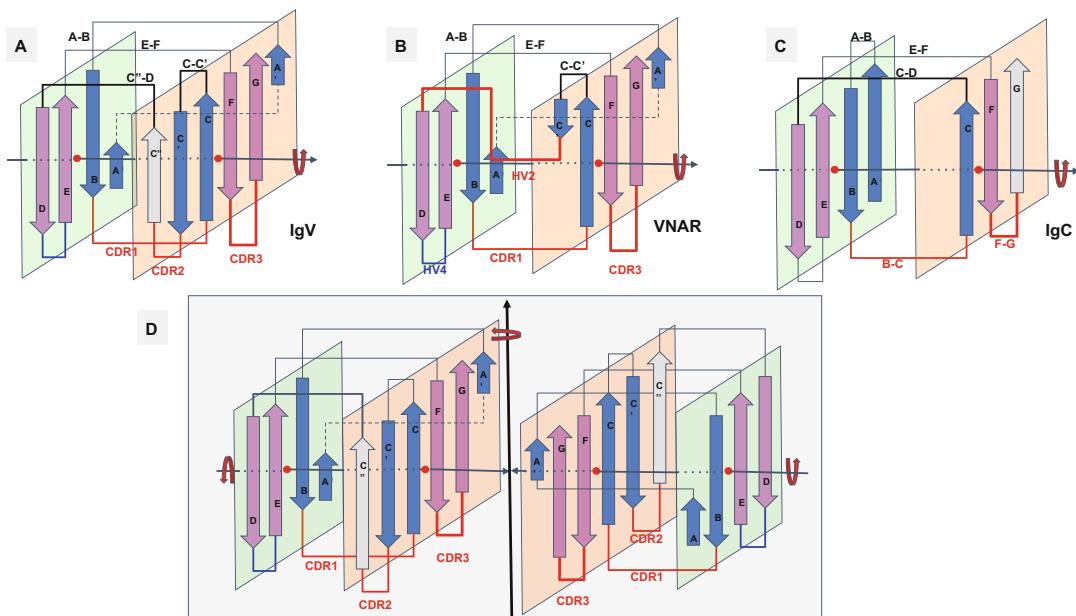


Fig. 2 Ig domain topologies for IgV, Shark VNAR, and IgC. **(a)** In IgV domains, the A strand, with a flexible hinge in the middle, usually a cis-proline or a stretch of glycines, swaps the upper part of the strand from Sheet A to Sheet B in a parallel model. So-called domain swaps, which are most often SSE swaps among symmetric packing pairs of domains, are observed ubiquitously. Here we can refer to it as a protodomain (half-strand) swap by analogy. The linker between protodomains in this example of an IgV type domain forms a C'' strand as an extension of Sheet B and the CDR2 loop between C' and C'', as well as a loop C''-D bridging Sheet B back to Sheet A. **(b)** VNAR shows that same domain-level organization with two protodomains, yet a much smaller inter-proto domain linker, eliminating the linker's supersecondary structure and the CDR2 loop. Instead, a short HV2 linker is observed. In the literature, C' is usually included in the HV2 region, as it is very short. In addition, a hydrophilic set of residues on Sheet B, i.e., strands GIFIICIC', facing out rather than hydrophobic in IgV, do not permit the formation of a symmetric dimer (as in D). This may also be due in part to the absence of an overall supersecondary structure of the linker in IgV (including C'), which may help patching an otherwise possibly semi-open eight-stranded barrel. **(c)** IgC. Here we consider only the IG C1-set, i.e., the antibody constant domain-like to exemplify an Ig constant domain protodomain connectivity. In this case the final domain is formed by a full four-stranded AIBIIED Sheet A, with no half swapping of strand A, vs. a three-stranded GIFIIC Sheet B, no C' strand. Interestingly this enables C-domain-level dimerization through that four-stranded Sheet A as opposed the IgV dimer interface obtained through Sheet Bs, enabling a further helical level symmetric arrangements of chained Ig domains. When looking at an IgC protodomain alignment, only three strands are considered. **(d)** IgV dimer. In CD8aa, two IgV domains pack together symmetrically as homodimers through their Sheet B (GIFIICIC') facing out form an eight-stranded semi-closed central barrel, with external strands C' and G of two domains closing the central (quaternary) barrel symmetrically. In CD8ab, as in IgV light and heavy chain quaternary assembly, they pack pseudosymmetrically as heterodimers (see Figs. 3, 4, and S1). As the heterogeneity of domains increases, and even if a pseudosymmetry is maintained at the sheet level, packing, i.e., quaternary interface, becomes more asymmetric, and central barrels become open with an asymmetrical arrangement between “closing strands” C/G, resulting in at least one side of the central dimer barrel open. This is the case of a PD1-PDL1 pair (see Fig. 4)

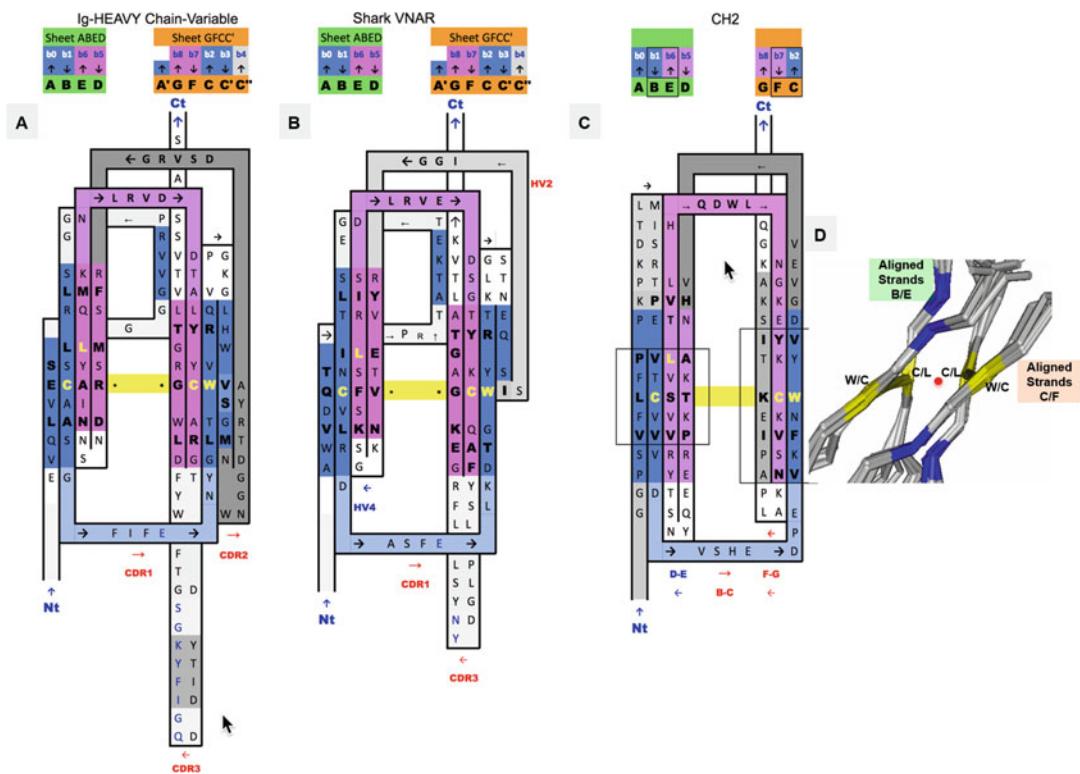


Fig. 3 2D Sequence/topology maps of Ig domain topologies for IgV, Shark VNAR, and IgC. (a–c) Corresponding to schematic topology drawing in Fig. 2 for IgV, VNAR, and IgC, respectively: Topology/sequence map alignments based on 3D structure domain- and protodomain-level alignments of a Human Antibody Fab 5ESV (chain H, IgV domain), Shark VNAR 1VES, and an CH2 domain-isolated 3DJ9 and/or in an Fc chain context 4NOU (chain E). (d) Central strands B/E on Sheet A (AIBIIIEID) and on Sheet B (GIFIIICIC'). Protodomain 1 = AIB – C(C')/Protodomain 2 = DIE – F(G). From 3D structure 2ATP (see Fig. 5) of CD8ab. The exact same pattern is observed here in IgV, VNAR, and IgC. These are four invariant residues (L can vary somewhat and be replaced by another hydrophobic residue). The cystine bridge flanked by a tryptophan is a well-known pattern that in fact exhibits pseudosymmetry with the residues in symmetry equivalent positions: C Cys (Strand B) \leftrightarrow L Leu (Strand E) and W Trp (Strand C) \leftrightarrow C Cys (Strand F). (d) Within a domain C/L on Sheet A central strands B/E and W/C on Sheet B central strands C/F occupy symmetry equivalent positions. The symmetry axis, perpendicular to the beta sheets A and B and the plane of the paper, is represented by a red dot. In symmetric dimers the two C2 domain axes coincide. These schematic maps are idealized showing vertical strands. The two sheets forming the central barrel are actually tilted vs. each other (relative rotation of one domain vs. the other around the common domain-dimer symmetry axis). This is true of any beta strand in any barrel

2.2 Software

2.2.1 Interactive Protodomain Delineation and Symmetry Analysis

The main computational engine needed to detect structural symmetry is structural alignment software. There are numerous tools and servers to perform structural alignments automatically between protein domains. Most are not configured to enable protodomain analysis. While this was the norm in the early days, few programs today allow interactive multiple structure alignment of proteins at any level. **Cn3D** [36, 37] allows the retrieval of structural

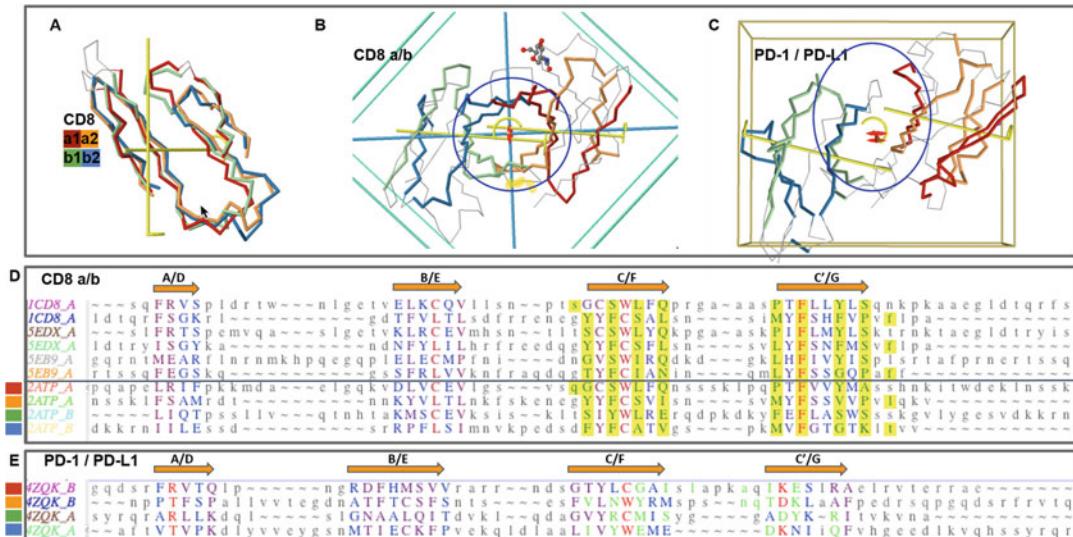


Fig. 4 CD8ab and PD1/PDL1 heterodimers. Protodomains and quaternary symmetric arrangements. **(a)** CD8ab (structure of mouse CD8ab: 2ATP). Four protodomains aligned for CD8a and CD8b colored red/orange and green/blue, respectively. Automatic symmetry detection and protodomain alignment performed with CE-symm and displayed with Jmol. Average RMS on protodomains as computed by CE-symm is 2.71. **(b)** CD8ab dimer with two orthogonal axes of symmetry: Two C2 levels of symmetry detected as overall D2 symmetry, meaning the two axes domain level and dimer level intersect in the center of symmetry, as for a CD8aa homodimer (see schematic representation in Fig. 2d). A small departure from perfect symmetry is observed between the actual domain-level yellow axes of symmetry vs. perfect orthogonality to the dimer axis, perpendicular to the plane of the paper. One can see a pseudosymmetric eight-stranded central barrel formed by the two faces of each monomer, from both sheets G/F/I/C/C' facing each other (the symmetric homodimer CD8aa—structure 1CD8 is presented in Fig. S1 with an iCn3D Link). **(c)** PD-1/PD-L1 receptor ligand interface (structure of human PD1-PDL1: 4ZQK). Here we still have a pseudosymmetry for each domain, and for the heterodimer, the two external faces of the respective Sheet B of PD1 and PDL1 are shifted laterally relative to each other, to form the interface. We still have two C2 levels of symmetry but the domain-level axes do not cross with the dimer axis on the center of symmetry. There is still a C2 domain level of symmetry for each domain, and a dimer center and C2 axis of (pseudo) symmetry, but not a D2 symmetry. The average on automatic detection RMS is 3.38A. **(d)** Optimized structural alignment of protodomains of CD8a in the homodimer CD8aa (structures of CD8aa: Human 1CD8; Swine 5EDX; Chicken 5EB9) and the heterodimer CD8ab (structure of mouse CD8ab: 2ATP chains A and B, respectively). The RMSD for the optimized multiple domains/protodomains alignment for each first and second protodomain vs. the first Human CD8a protodomain are **1.61, 0.436, 1.71, 0.852, 1.57, 0.522, 1.95, 0.895, and 1.54 A**, respectively. The computer-generated alignment is higher by 1–2A (this is usually the case). In this case it does a good job to match key symmetry equivalent residues, especially C/L and W/C. However accurate delineation and multiple structure alignment is only possible through interactive software Cn3D currently. Noticeable is the absolutely conserved F residue in strands C' and G. Interface residues are contributed pseudosymmetrically as can be seen in the alignment for residues colored in green and highlighted in yellow, except for F colored red. **(e)** Optimized protodomain alignment of PD-1 and PD-L1 (structure 4ZQK chains B and A, respectively). In this case the automatic alignment is not as good as for CD8ab but is good enough to detect two levels of symmetry. The structural alignment optimized interactively gives a very good RMSD for the four protodomains with **1.73, 1.53, and 1.84 A**, respectively, for the second PD-1 and the first and second PDL1 protodomains relative to the first PD-1 protodomain. Noticeable is a C/M match vs. a C/L match between protodomains of PD-1 vs. PD-L1. On the PD-1/PD-L1 interface, it is clearly not as symmetric as for CD8 as the barrel opens on one side vs. the other, with the relative shift of the domains external faces of Sheets B (G/F/I/C/C') observed (see **c** vs. **b**)

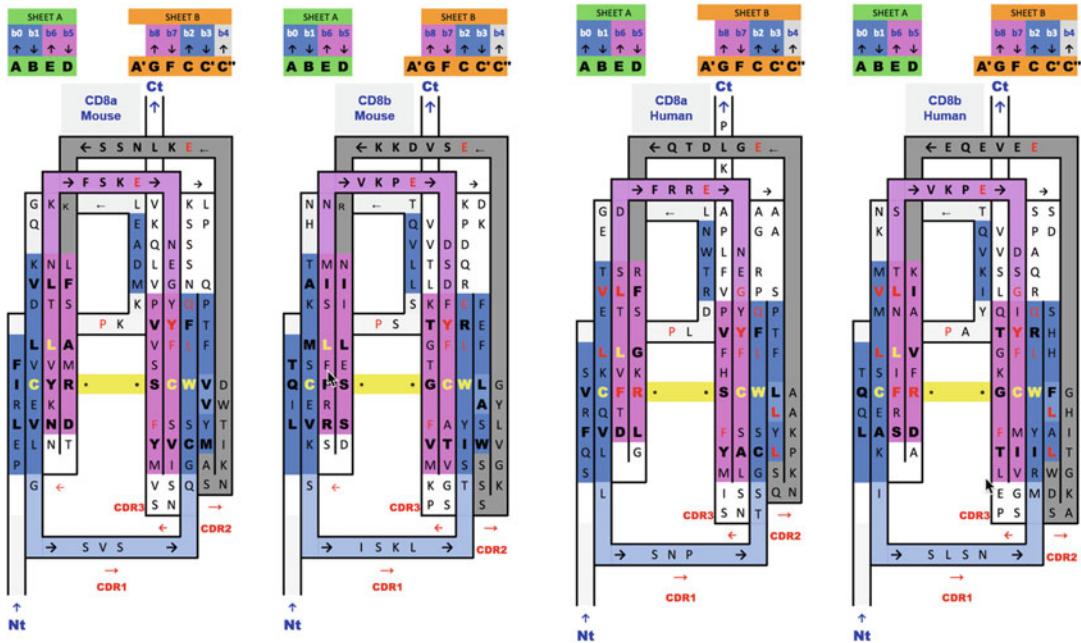


Fig. 5 2D Sequence/topology maps and alignments of Ig domain heterodimers of human and mouse CD8ab. Sequence/topology map alignments based on 3D structure domain- and protodomain-level alignments of CD8a and CD8b in a mouse structure of CD8ab (2ATP) and a human structure of CD8a in a CD8aa homodimer context (1CD8) with a human sequence mapped onto the mouse structure. Corresponding protodomain structure-based sequence alignments are available in Fig. 4. Topology/sequence maps corresponding to schematic topology drawing in Fig. 2d

alignment from NCBI's VAST+ alignment databases [35]. It allows interactive multiple structure alignment of domains, and, very importantly for our objective, one can superimpose a domain onto itself and hence delineate protodomains accurately.

2.2.2 Automatic Pseudosymmetry Detection Protodomain Delineation

Two recent programs perform symmetry analysis and enable domain-level pseudosymmetry detection: **SYMD** [38, 39] and **CE-symm** [1]. These programs will do a good job in most cases, yet they do not give exactly the same protodomain delineation. For a very accurate protodomain delineation, an interactive step using Cn3D may be the best final option, especially in complex cases (*see* later in Subheadings 3 and 4).

2.2.3 Structural Visualization and Analysis

iCn3D [40] is the new JavaScript viewer from NCBI available as open source (<https://github.com/ncbi/icn3d>). iCn3D (I-see-in-3D) allows interactive visualization but also structural analysis and comparisons of biological macromolecular assemblies and molecular interactions in a web browser using three levels of complexity 1D (sequence), 2D (topology/cartoon), and 3D (structure). Very importantly it allows scientists to exchange annotated

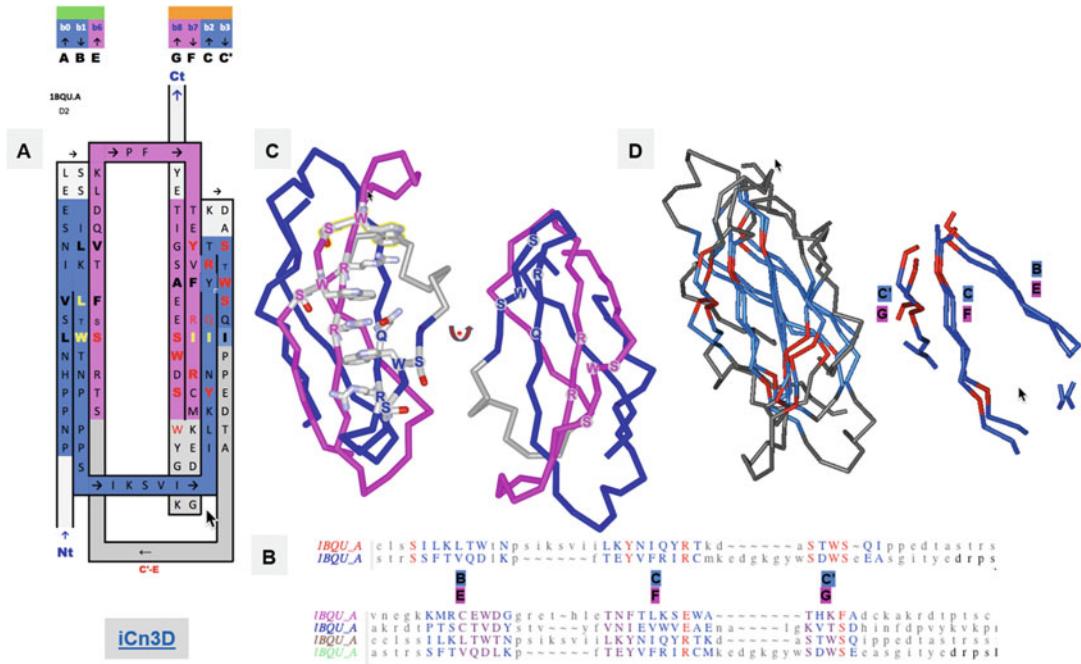


Fig. 6 FN3 Ig domains. **(a)** Another Ig-fold variant, the FN3 superfamily, with the example of the cytokine-binding homology region (CHR) of the cell surface receptor gp130, the second FN3 domain proximal to the membrane surface. The inter-protodomain linker now connects C'-E through a Greek key loop bridging the two sheets, composed of AIBII(E) and GIFIIC'(C'). In this case, what would otherwise be a D strand in linking back to the C' strand (Fig. 2), removing one strand from the other Sheet A (AIBII(E)) rather than Sheet B (GIFIIC'(C')) as in IgC (see Figs. 2 and 3). The sequence patterns SxWS in strands C' and G and R/QxR in strands C and F match symmetrically. **(b)** Structure-based protodomain sequence alignment for domain 2, followed by domain 1 and 2 together, respectively, where one can observe each domain idiosyncratic protodomain “internal conservation” sequence patterns (see text for details). In domain 2 residues S, Y, and R, SxWS are matched, while in domain 1, the pattern is totally different with residues C, D, N, and E. Only one residue S is common to three out of four protodomains, while a R vs. E in symmetrically equivalent strands C and F is observed consistently, a residue which is part of that ClIF zipper (see text and Fig. S2). RMSD is 1.8A between domain 2 protodomains and 2.89A for domain 1, 2.2, and 2.5, respectively, vs. domain 2 protodomains (multiple alignment). **(c)** The symmetric sequence patterns matched in structure forming a cation-pi ladder W*R*W*R*W from both (W)SxWS, the so called WS motif (see text). **(d)** Structure alignment of the two protodomains matching Strands B-CIC' and E-FIG that combine as (A)BII(E) (Sheet A) and GIFIIC'(C') on Sheet B, corresponding to the pairwise protodomain alignment of domain 2 (see sequence alignment in B) <https://d55qc.app.goo.gl/DcmplJy2CVmxtHKN2>

visualizations, such as in figures hereafter (see web links on Figs. 1, 6, 8, 9, S1, S2, and S5).

Jmol [41], written in Java, is used for quaternary symmetry visualization on the RCSB web site [22] and is also used with CE-symm to visualize pseudosymmetries and multilevel symmetries combined, i.e., quaternary and tertiary. However, we should expect the use of JavaScript viewers. **NGL** [42] or iCn3D down the road for 3D visualization.

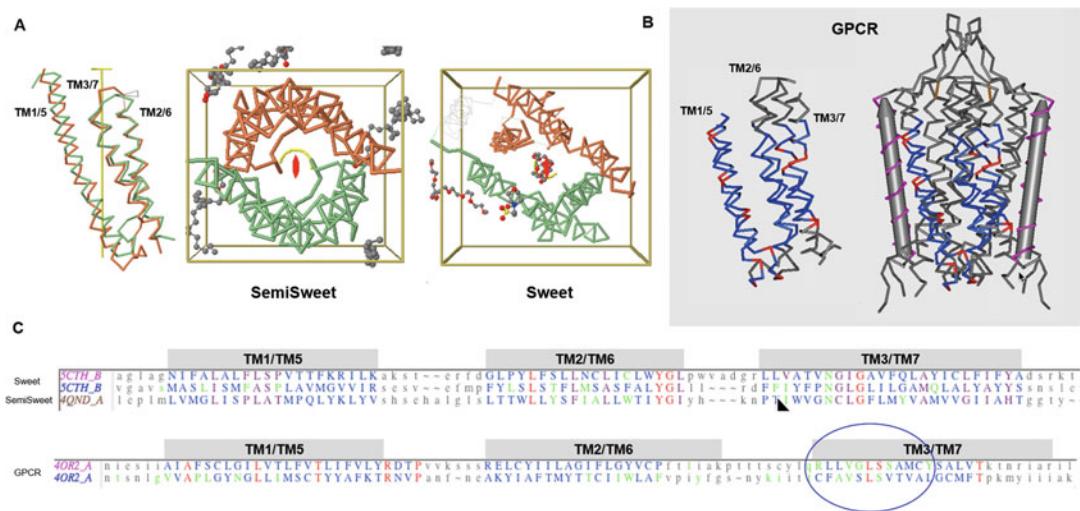


Fig. 7 7-Transmembrane helical (7-TMH) proteins. Sweets and GPCRs. **(a)** Sweet protodomains (3-TMH) aligned, bacterial SemiSweet (3TMH) dimer, and 7-TMH Sweet Protein. The linker between the two Sweet protodomains forms an additional transmembrane helix (TM4). While formed with three consecutive helices in sequence, a protodomain exhibits a 1–3–2 structural arrangement in 3D that is duplicated to form a symmetric pseudosymmetric domain equivalent to a Bacterial SemiSweet symmetric dimer (less TM4). The two protodomains match each other with a RMSD of 1.36 (Sweet protein structure 5CTH) after optimization (automatic detection alignment was 2.91A). A bacterial SemiSweet 3-TMH “domain” aligns with Sweet protodomains with an RMSD of 1.98A (SemiSweet structures 3QND/3QNC). The 7-TMH and 3-TMH dimer align very well not only at the protodomain level but at the dimer vs. pseudo-dimer level. Here displayed with the symmetry axis perpendicular to the plane. The ligand lies on the axis of symmetry. **(b)** 7-TMH Class C GPCR (structure 4OR2—metabotropic glutamate receptors (mGlus) bound to an allosteric modulator) protodomain optimum alignment with an RMSD of 3.32A through interactive alignment software Cn3D. GPCRs can also be considered with a two-protodomain arrangements. The two protodomains exhibit a distinct 1–2–3 organization in 3D. Here we display the alignment of the whole 7-TMH protein onto itself; the symmetry match can be observed with a solid gray cylinder for the TM4 “linker.” Unlike Sweet, symmetry detection programs do not detect pseudosymmetry systematically but can, in a few cases, using stringent criteria. Interactive alignment of 3-TMH protodomains was used as the method of choice in this case. In all known Class A structures, we have examined, but also in the two Class C structures currently available, pseudosymmetry highlights symmetry equivalent residues in TM1/5, TM2/6, and TM3/7, in a systematic way for some key residues. The structurally aligned TM3/7 helices often exhibit a pseudosymmetric sequence motif (see C and text), framing ligand-binding residues pseudosymmetrically, with ligands lying for a significant part on the axis of symmetry. **(c)** Associated sequence alignments with mapped ligand-binding residues (or rather residues within a 4A radius from ligand) for Sweet/SemiSweet (sugar) and for the GPCR vs. its ligand. It is important to note that for any pseudosymmetric domain, a protodomain defines a domain entirely (see text). A protodomain is usually idiosyncratic. Here the Sweet protodomain is very different in topology 1–3–2 vs. GPCRs with 1–2–3 topology, yet each defines its domain through that same duplication and pseudosymmetric arrangement

2.3 2D Representations: Topology/ Sequence Maps

There are a few programs that may be useful to represent graphically domain-level topologies (2D), in particular Pro-origami [43]. Such representations would benefit from using and depicting internal pseudosymmetry to identify the repeating and symmetrically organized supersecondary structures. It is not an easy task to

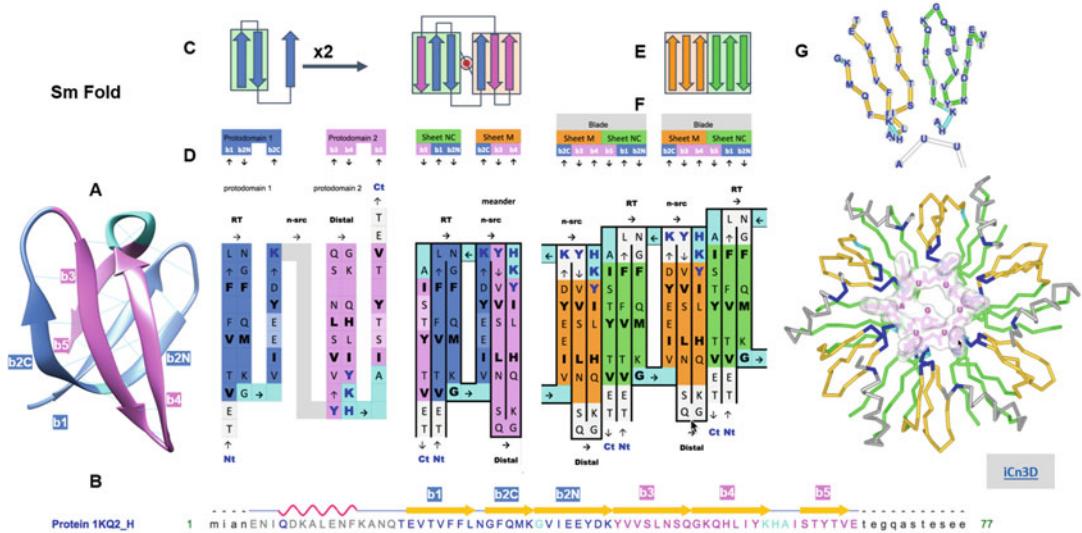


Fig. 8 Complexity buildup through hierarchical symmetric arrangements of protodomains, domains, and oligomeric assemblies of the Sm fold (Hfq). **(a)** 3D structure of the bacterial Sm barrel (Hfq) (structure 1KQ2, N-terminal helix omitted for clarity). It is a small beta barrel with an SH3-like topology, usually considered as a five-stranded beta barrel (strands b1–b5). It is better represented as a six-membered barrel sandwich (splitting the long and sharply bent b2 strand in b2N and b2C at the Gly position, since b2N and b2C participate in two orthogonal sheets denoted either A and B or NC or M, as b1 and b5 N and C terminus come together in an antiparallel mode in the first sheet, vs. M a meander formed of b2C-b3-b4). Even a small barrel composed of 50 residues can exhibit C2 symmetry, bringing down to 20–25 residues the protodomain size with a bb-b topology formed by a hairpin b1lb2N-b2C. An SH3-topology, for SH3-like domains as for the Sm fold, is equivalent to short Greek key, with a simple Glycine in protodomain 1 and a 3–10 helix in protodomain 2 linking the two (orthogonal) sheets of the barrel. **(b)** Sequence and strand definition of an Hfq domain (1KQ2) with protodomain 1 in blue (res T20-K41) and protodomain 2 in magenta (res. Y42-E66). **(c)** 2D Schematic topology representation of protodomain and protodomain duplication with symmetric arrangement. **(d)** Sequence of two successive protodomains' delineation and corresponding 2D topology/sequence map. Protodomain 1 in blue color, 2 in magenta. Linker residues G in protodomain 1 and YKHA (3–10 helix) in protodomain 2 are highlighted in cyan. Protodomains b1lb2N-b3 and b4lb5-b6 come together symmetrically with a b2Cllb3 b5llb1 to form three-stranded Sheets A (green) and B (orange). Hydrophobic residues forming the core of the barrel are in bold black. RNA-binding residues are highlighted in dark-blue bold characters. **(e)** Schematic representation of the formation of a six-stranded blade from Sheet M (orange) and Sheet NC (green) of consecutive monomers. **(f)** “Quaternary” topology/sequence map of a dimer, with a b5lb4 quaternary interface. **(g)** A six-stranded blade representation in 3D labeled by sequence and the Hfq hexamer with RNA nucleotides binding at the interface between domains (RNA-binding residues highlighted in dark blue, as in (d)). All six strands can be considered calibrated with 5–6 residues (considering bulges, in the case of Sm in b2C and b5 symmetrically), so they form two calibrated three-stranded beta sheets that dock to form six-stranded blade, resulting in a six-bladed Hfq ring structure of C6 symmetry. A beautiful example of complexity buildup. Link to iCn3D: <https://d55qc.app.goo.gl/pgk8GcZZNSs9KSMU6>

represent 3D symmetry or pseudosymmetry in a 2D depiction for any pseudosymmetric domain with very diverse topologies. One solution is to use 2D templates, and this is what we will do, at least for the chosen beta structures used in this chapter (*see* Figs. 1, 3, 5, 6, and 8).

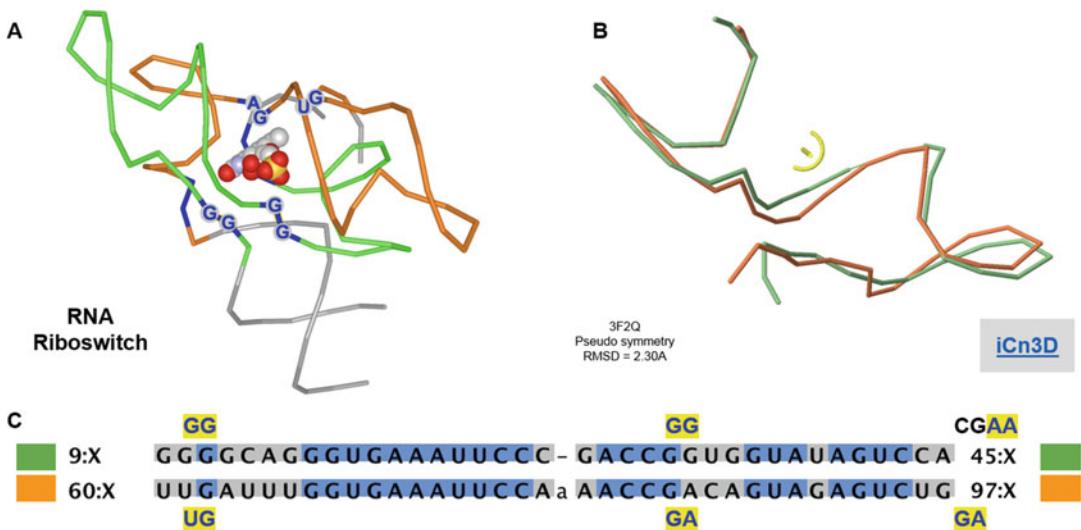


Fig. 9 RNA protodomains in riboswitches and ligand binding. (a) Symmetric arrangements of RNA protodomains. Ligand-binding residues are highlighted dark blue. The ligand is on the axis of symmetry perpendicular to the paper plane. (b) Protodomains are structurally aligned with an RMS of 2.30 Å (at the symmetry detection step, no optimization performed). (c) Structure-based protodomain sequence alignment. Ligand-binding residues are in dark blue highlighted in yellow. They match exactly in sequence position in both protodomains (GG/UG–GG/GG, and a small offset on the third binding dinucleotide AA/GA just outside of the delineated protodomains). Link to iCn3D: <https://d55qc.app.goo.gl/gJee10ict11uT0Y22>

2D topological representations of protodomains and domains may also allow visualization of quaternary arrangements (*see* Figs. 8 and S4 for an example). Another interest of using such representations is the possibility of threading the sequence onto SSS depictions. For beta sheets one can represent lateral residue contacts due to H-bonding. Some further 3D structural information can also be mapped, by highlighting some key tertiary, quaternary, or ligand contacts, as well as any sequence conservation or mutations (*see*, e.g., Fig. S5). They also help visualize clustered sequence-topology patterns and allow a straightforward parallel topology/sequence alignment, where it may be easier to see patterns than in 3D, especially for non-experts. These representations could be considered to some extent 2½D. Future developments should aim at integrating such representations into existing visualization software to represent and use simultaneously 1D sequence, 2D topologies, and 3D structures [44] at any level of detail (residue, SSE, SSS/protodomain, domain, chain, multidomain assemblies). We use topological representations of SSSs to describe supersecondary structures and their pseudosymmetric arrangements. (Note that as there are four ways to represent symmetrically organized sheets on a flat surface in terms of their order, we have chosen one in Ig domains: ABED for Sheet A and GFCC' for Sheet B; *see* Figs. 1, 2, and 3.)

3 Methods

3.1 Learning by Example: Selecting Structures

At this stage a structural analysis through pseudosymmetry is still more an art than a science, but with practice, we learn. Also, analytic software tools are still in their infancy and not integrated. Each step requires a tool and some tools are not automated. So, we will approach learning through practical examples.

While pseudosymmetry is found in all classes of protein structures (*see* Table 1), beta structures offer more examples than any other class. Beta protodomains are easier to delineate accurately than alpha or alpha-beta structures (*see Notes*). They offer splendid examples of complexity buildup through symmetric arrangements of supersecondary structures at both the tertiary and quaternary level. We will use examples of well-known protein domains with underrecognized (Ig) or unrecognized (Sm) tertiary pseudosymmetry despite their importance.

3.1.1 The Immunoglobulin Fold: Tertiary and Quaternary Structure Analysis

We will analyze the immunoglobulin fold (Ig fold). Although the quaternary symmetry of immunoglobulins is very well known [5], the pseudosymmetry of the Ig domain itself, which had been noticed early on even at the sequence level [13], has not been systematically analyzed or used in protein engineering to the extent possible. Also, despite the various immunoglobulin types with a different number of strands, the variable and constant domains, all exhibit pseudosymmetry. A protodomain decomposition can highlight pseudosymmetry, as well as the various loops in that context, especially Complementarity Determining Regions (CDRs) in the case of immunoglobulins. The E form shows more irregularities in matching protodomains however. We will not discuss all types, as it is beyond the scope of this chapter.

The immunoglobulin fold (SCOP b.1), beyond the immunoglobulin family itself (b.1.1), is ubiquitous. It is the most functionally diverse beta sandwich barrel fold with 28 distinct superfamilies (in SCOP 1.75). A newer taxonomies such as ECOD regroups even more superfamilies such as P53, even when classified as different folds in SCOP (b.2). From a pseudosymmetry standpoint, P53 offers a parallel to immunoglobulins, yet it is a more complex domain (Fig. S5). In the immune system, a majority of cell surface proteins are composed of Ig domains, and many such domains are involved in checkpoints: PD-1–PD-L1, for example, are not only each composed of Ig domains, they interact together as receptor ligand (*see* Fig. 4). The study of Ig interfaces in terms of supersecondary structures can offer valuable structural insights in the design of checkpoint blockade therapies.

Given the enormous interest in using immunoglobulins as Fabs, Fvs, or single domains to target antigens as in CARs, checkpoint blockade inhibitors, or multispecific antibodies, we'll focus

mainly on the Ig fold, as an example of both protodomain and domain-level interactions. As mentioned above, the Ig fold is shared by 28 superfamilies, of which immunoglobulins are one. The method of deconstruction and analysis of protein domains in supersecondary structures to analyze them and the families within should be useful for numerous folds and superfamily variants. We will explore, for example, the FN3 domain (SCOP b.1.2), an interesting variant of the Ig fold (Figs. 6 and S2).

3.1.2 The Sm Fold and Hierarchical Complexity Buildup Through Symmetric Arrangements

Complexity can build up through oligomerization. In beta structures not only nonbonded interactions but also beta sheet formation in either parallel or antiparallel represents a very clever self-assembly mechanism. Sm-like oligomers assemble through that mechanism, mostly as homo-hexamers in bacteria, homo-heptamers in archaea, and hetero-heptamers in eukaryotes. There are additional variants of 3-, 5-, and 8-mers [45] (the 3-mer formation is somewhat different breed, yet it still exemplifies a symmetric assembly of beta sheets). We shall briefly describe the hierarchical assembly of bacterial Sm (Hfq) hexamers (SCOP b.38; Figs. 8, S3, and S4). Many doughnut-like oligomers possess multilevel symmetries in terms of protodomains, domains, and larger quaternary structure. This is a prototypic example [46].

3.1.3 Helical Protodomains: Sweets and GPCRs

Many membrane proteins exhibit pseudosymmetry (Table 1), and two thirds of known membrane protein structures are helical. We shall briefly look at a couple of 7-transmembrane helical proteins and compare eukaryotic Sweet vs. bacterial Semisweet proteins in a pseudosymmetric tertiary arrangement vs. a quaternary dimeric arrangement, respectively. We will make a parallel with another very important structural family: GPCRs (SCOP f.13) (Fig. 7).

3.1.4 RNA Protodomains: Riboswitches

As mentioned before, proteins are not the only biological macromolecules exhibiting symmetry at either the quaternary or tertiary level. RNA riboswitches provide magnificent examples [28], and we shall look at one example of RNA protodomains [47] (see Fig. 9).

3.2 Pseudosymmetry Protodomain Analysis (PSA) Method

The method is pretty straightforward. It involves two initial steps, **symmetry detection** and **protodomain delineation**, followed by as many **analysis** steps as one wishes to perform.

Symmetry detection gives the symmetry point group and a first delineation of protodomains. Both tertiary and quaternary structural symmetries can be determined at the same time.

A second step is usually required to optimize protodomain boundaries and structural alignment for an accurate protodomain delineation. A third obvious step is to analyze 1D sequence patterns resulting from the 3D structural alignment of protodomains. From

there one can branch into a deeper structural analysis and understand self-complementarity of secondary structure elements. This is where a 2D sequence-topology analysis, as well as a 3D structural analysis helps, bringing together sequence conservation, symmetry, folding, and possibly function (for example, in cases where ligand binding may involve residues in symmetry-related positions in protodomains). Beyond, this may open perspectives for deeper evolutionary analyses. We will analyze a set of examples and summarize results in Figs. 1, 2, 3, 4, 5, 6, 7, 8, and 9. They should be self-explanatory, as we will present geometrical properties in simplified schematic representations and alignments. We will make use of 2D topology/sequence maps for the beta structures analyzed (Figs. 1, 2, 3, 4, 5, and 6).

3.2.1 Symmetry Detection

In many cases, recently developed computer programs allow the detection of internal pseudosymmetry in tertiary structure [1, 38]. We use the program CE-symm to that effect. A newer version of the software allows quaternary symmetry analysis of multidomain complexes at the same time (<https://github.com/rcsb/symmetry>), and we will see an example in Fig. 4. There are cases however where one has to revert to interactive alignment software to align a domain onto itself. We will see such a case with GPCRs (Fig. 7) and the Sm fold (Fig. 8). In all cases we optimize protodomain delineation through interactive alignments for accuracy.

3.2.2 Protodomain Delineation: Optimization Through Structural Alignment

Protdomain alignment may highlight key residues that may be internally conserved for a structural reason (folding/assembly) or for a functional reason. In most cases, the degree of overall internal conservation is low. This is a hallmark of many pseudosymmetric domains, unlike most domain/family level sequence-structure conservation, except for some clear duplication cases where protodomain homology between protodomains may be as high as 40% [48]. In such cases duplicated “protodomains” tend to have a larger size and can be considered fused domains. The low level of “internal conservation” observed however is most likely due to a long protodomain-protdomain coevolution within each and every protein domain but also at quaternary interfaces. In order to call internally conserved residues, an accurate structural alignment of the protodomain is required.

Using pseudosymmetry provides a framework for structural analysis. It allows a deconstruction of protein domains in well-defined parts that may also lead into evolutionary and/or functional analysis. The reconstruction of a domain from parts leads into a coevolutionary analysis of the parts and their interfaces and in the understanding of molecular self-assembly. This opens perspectives in developing the analysis method further in that direction. We will focus essentially on structural and topological analysis in this

chapter: delineating supersecondary structures (SSS) related by symmetry and highlighting self-complementarity of interfacing SSEs, as is the case in symmetry equivalent strands (B|E) and (C|F) in immunoglobulins (Figs. 1, 2, and 3), for example.

3.2.3 Sequence Analysis: Based on Protodomain Structure Alignment

The structural alignment of protodomains naturally highlights “internally conserved” residues, i.e., identical residues at symmetrically equivalent positions. These can point to a possible original protogene duplication if the degree of conservation/sequence homology is high enough to call such a duplication. A point group symmetry operation between two or more entities establishes a structural equivalence relation between these entities. Two residues or sets of residues related by pseudosymmetry in equivalent positions can be analyzed in terms of conservation (identity, similarity or lack of) as for domains. However, “internal conservation” may or may not be as significant as in domain comparisons, as each and every domain had its own evolutionary history and each and every protodomain within had its own coevolutionary history with symmetrically related protodomains. Sometimes it may be best to talk about coincidence rather than conservation, unless a pattern appears on more than two protodomains, and/or that same pattern is observed across multiple homologous domains for their respective protodomain alignments. Nevertheless, analyzing a pattern is usually fruitful, even if the pattern belongs to that domain alone. In which case it may be functional (*see* Fig. 6 for the example of a FN3 domain).

Also, one may have different residues in symmetry-related positions that are conserved as residue pairs across domains of a given family or superfamily. This is the case of the symmetrically related W/C and C/L residue pairs in immunoglobulin domains (*see* Figs. 2 and 3). In such case we may have a coevolved pair conservation. This pattern is easy to see. There are coevolved pairs in many pseudosymmetric domains that would benefit from AI software to identify such patterns, as we are not used or trained to capture such patterns, but this is something a symmetric decomposition of domains in protodomains may help us identify. An interesting example can be found in decomposing an FN3 domain (SCOP b.1.2) for type I cytokine receptors (gp130/IL-6R, IL-2R, IL-21R, GHR, etc.) (*see* Figs. 6 and S2).

There are also cases where one has to shift the sequence by 1–2 residues in beta or 3–5 in alpha helical protodomains to identify matching sequence patterns, as sequence may shift on structure during domain evolution. This adds a level of difficulty in the identification of matching residues and patterns.

3.2.4 2D Topology/ Sequence Analysis

Having delineated and aligned protodomains structurally, and eventually observing some sequence patterns of internally conserved residues between them one, one can analyze the sequence relations in 2D and 3D and protodomain-protoprotein interfaces specifically.

Internal residue conservation is usually highly idiosyncratic, and for two families sharing a fold or even for two different domains within a family, these may not be the same (Figs. 6 and S2). This points to a fact that each domain has its own internal protodomain coevolution history, and internally conserved residues may not be analyzed along the same line as in conservation studies across family or superfamily members. We need to depart from the usual sequence conservation patterns to some extent. It is both an evolution and a coevolution analysis. When looked in the 3D context, or simply in the 2D topological context, sequence conservation patterns and their structural or functional meaning may start to emerge.

Using 2D Topology/ Sequence Maps

2D topology/sequence maps are very useful in analyzing sequence patterns in a topological context. In Fig. 1, we represent a topological representation of an Ig protodomain, itself a set of two beta hairpins ($A|B$)-(C|C'), connected by a Greek key linker [B-C], which duplicates as ($D|E$)-(F|G) connected by a Greek key linker [E-F], using the usual Ig nomenclature of strand names. The two protodomains assemble symmetrically.

Figure 2 shows variants of the immunoglobulin domain, the variable domain IgV, the constant domain IgC, and also the shark variable domain VNAR, for comparison. Following a general beta sheet H-bonding (lateral) association pattern, in antiparallel, each of the hairpins associates with its duplicate in a symmetric fashion with the symmetry-related strands ($B|E$) and ($C|F$) coming together. The IgV or other Ig domain variants have been described in great detail, yet a protodomain decomposition allows us to see Ig domain variants with a fresh look, bringing them within the same framework, varying essentially the inter-protoprotein linker.

- In **IgV** the linker has some hypervariable sequence with some secondary structure: it forms a CDR2 loop from the C' strand to an additional C'' strand and a [C''-D] loop to bridge back with the Sheet A ($A|B||E|D$).
- In shark immunoglobulins (IgNAR) [49–51], the variable domain **VNAR** has a shorter hypervariable region linker [C'-D], named HV2 between a smaller C' strand bridging Sheet B ($G|F||C|C'$) back to Sheet A ($A|B||E|D$), no CDR2, no C''.
- In **IgC**, what would otherwise be the C' strand serves directly as a connector between the two sheets, as if the duplication did not

evolve any linker. Figure 3 compares topology/sequence maps for IgV, VNAR, and IgC.

- **FN3** is another variant of the Ig fold (SCOP b.1.2). In a similar way to the IgC domain, a strand, this time D on Sheet A as opposed to C' in Sheet B, is removed to serve as a linker. The linker now connects C'-E through a Greek key loop bridging the two sheets. Sheet A is composed of (A|B||E) and Sheet B is composed of (G|F||C|C'). We use the cytokine-binding homology region (CHR) of the cell surface receptor gp130 (interleukin 6 receptor) as an example (Fig. 6) and the homologous growth hormone receptor (GHR) (Fig. S2).

Sheets A and B are consistently shown in figures with a green and orange background color, respectively, while we use blue and magenta for consecutive protodomains, respectively. In protodomain analysis the emphasis is put on sheets facing in, i.e., facing each other to form a domain core, i.e., a protodomain interface; hence we represent the **in-facing (domain core) residues in bold**. Naturally those in between, not in bold, are facing out (on the strand edging the barrel, especially C'' in/out may not be relevant). Residues facing out in Sheet B (G|F||C|C') form the quaternary interface between IgV domains (Figs. 2, 3, and 4). That interface is in itself a central barrel in homodimers, as well as in heterodimers such as CD8ab (Fig. 4). In IgC quaternary interfaces (not shown) are formed by the external faces of Sheets A (A|B||E|D) (Figs. 2 and 3).

Beyond 1D sequence patterns, in aligning topology/sequence maps, we can find **2D sequence patterns**, as residues cluster in beta sheets. The well-known CCW conserved pattern (a disulfide bridge flanked by a Trp) is easy to spot, to which one could add a hydrophobic fourth residue, in most cases L, as highlighted through the symmetry operation (*see* Fig. 3d). In the case of Immunoglobulin domains another interesting pattern in Sheet G|F||C(|C') is, a transversal 2D “T-Y-R motif” Threonine, tYrosine, aRginine. In CD8a, only the Tyrosine (two residues upstream from the Cys residue in strand F, is absolutely conserved, in CD8b however the full motif is there (*see* Figs. 1, 3 and 5). These 2D depictions give us a general topology/sequence map of the domains. A number for each residue, such as Kabat or IMGT reference numbering, can be used to pinpoint side residue lateral contacts in beta sheets [52–54].

3.2.5 2D/3D Protodomain Interface Analysis

Once protodomains are delineated and their topologies mapped, one should look at how symmetry-related SSEs pack together to form an interface. Are symmetrically equivalent SSEs interacting directly? How, in particular, are the protodomain interfaces formed in terms of their individual SSEs? This will bring insightful information on self-assembly and domain formation itself. In terms of interactions between SSEs, are they backbone vs. side-chain

packing level? The former of course is for beta or mixed alpha-beta structures. A backbone level assembly of SSSEs through hydrogen bonding is a hallmark of beta strands interactions in forming hairpins and sheets, while nonbonded (side chains) interactions is a general mechanism common to both beta strands and alpha helical SSSEs. We will see both types of examples in important beta folds: Ig (Figs. 1, 2, 3, 4, 5, and 6), Sm (Fig. 7), and alpha folds GPCRs (Fig. 8). Beta folds provide both packing mechanisms; this may be one reason why beta structure architectures exhibit a higher level of pseudosymmetry overall than other classes (Table 1). Inter-protodomain interfaces together with inter-domain quaternary interfaces may allow an interesting decoding of biological units' complexity buildup (*see* examples in Figs. 4 and 8).

3.2.6 3D Structure-Function Analysis

Local Sequence Pattern Matching from Structural Protodomain Alignment

Sequence patterns obtained from protodomain alignments are idiosyncratic. One should look where “internally conserved” residues are located in 3D, in a domain (facing in) or at a quaternary interface (facing out). A structural alignment between protodomains creates a correspondence, a mapping, and an equivalence in 3D positions. It may have some significance from a folding, assembly, or functional point of view. It may also be a coincidence in a unique evolutionary process of a domain.

Let's take the example of the gp130 cytokine-binding homology region (CHR), a type I cytokine receptor, composed of two FN3 domains (structure PDBid: 1BQU). The 3D structural alignment of protodomains (Fig. 6) shows an internal conservation in the second FN3 domain, proximal to the membrane, where C|C' and F|G hairpins are self-complementing each other symmetrically. A “conserved” pattern emerges. The structure exhibits not one but two (W)SxSW patterns in the external strands C' and G, placed symmetrically. We observe that these aromatic residues are part of an extended cation- π ladder W-R-W-R-W-R, where the arginine residues R are positioned symmetrically in the central strands C and F. The structure-based sequence alignment (Figs. 6 and S2) shows a longer conservation pattern with QyR in strand C<=>RxR in strand F at the domain family level as well as at the protodomain level for each domain. These symmetric patterns across the entire Sheet B (GFCC') can be seen in the 2D topology/sequence maps (Figs. 6 and S2). One can now analyze the structural self-complementarity that may be linked to folding and function, most likely a combination of both. The sequence alignment of strands C and F highlights what we could call an antiparallel hydrophilic beta zipper, with hydrophilic/charged residues matching (in orange in figures).

The WSxWS motif, also called the WS motif, is well-known and has been actively investigated, but it is still enigmatic. In the case of the IL-21 receptor, it has been linked to sugar binding

[55, 56]. It is conserved in the family in protodomain 2. The symmetry-related pattern **SxWS** in protodomain 1 however is not as conserved. **W** is conserved in IL-2R, but not in IL-21R. The central zipper however, with the **QyR-RxR** pattern, seems more conserved. If we now look at a more distantly related protein, the growth hormone receptor (GHR), the central strands C|F form an extended hydrophilic zipper conserving the pattern as **QyK-RxK** (Fig. S2). The WS motif is now replaced by **YGEFS** [56, 57], while in the symmetrically related motif **WK-MM** in protodomain 2 conserves **W**. The aromatic residues **Y,F** and **W** from strands G and C, respectively, intercalate to form an extended cation-pi ladder with the central **R** and **K** residues. In GHR, **Y** and **F** are structurally equivalent to the two **W** in protodomain 2 of CHR (*see* Fig. S2).

We have here an example where protodomain sequence conservation patterns vs. family/superfamily patterns can help identify some structural and functional residues, and coevolutive pairs, without overinterpreting what internal conservation patterns may mean.

Ligand Binding

Many membrane proteins exhibit pseudosymmetry (Table 1). This has been reviewed extensively [58, 59]. Two thirds of known membrane protein structures are helical. 7-transmembrane helical proteins give examples of alpha folds exhibiting pseudosymmetry. In the case of 7-TMH eukaryotic Sweet proteins, we can compare protodomains to 3-TMH bacterial SemiSweet monomers directly, and the 7-TMH whole domain to the SemiSweet (2x3TMH) dimer, in structure and in ligand-binding function (Fig. 7). This pseudosymmetric tertiary arrangement matching a quaternary dimeric arrangement points to a possible duplication-fusion at the origin of the pseudosymmetric 7-TMH [60, 61].

We will make a parallel with another very important structural family: GPCRs (SCOP f.13). Although Rhodopsin pseudosymmetry has been detected anecdotally in the literature [62], no current symmetry detection program [1, 38] can detect GPCR pseudosymmetry systematically. While we now have over 100 GPCR structures in the PDB database, pseudosymmetry is detected for a handful of GPCRs using symmetry detection software that often requires stringent matching criteria for protodomain delineation (*see Notes*). Rhodopsin, a GPCR Class A, or Metabotropic Glutamate Receptor 1, a GPCR Class C for which structures are available (PDBids: 1F88; 4OR2, respectively) [63–66], are cases where one can detect a pseudosymmetry with a stringent criterion. Otherwise, only careful manual structural alignments can lead to a solid pseudosymmetry analysis. A protodomain alignment of a Class C GPCR is presented in Fig. 7.

In Sweet, one can observe the ligand lying on the axis of symmetry. One can observe some residues in symmetrically

equivalent positions binding the ligand (Fig. 7) in TM3/TM7, N in the internally conserved NG(L/I)G pattern, and the structurally aligned TM3 of the semisweet protein, with a matching pattern **NCLG**. In GPCR Class A we find multiple instances of such patterns in TM3/TM7. The pattern is always idiosyncratic, and this is consistent with ligand-binding specificity. (Note that the topology of Sweet and GPCR protodomains is different. TM3 is in a different position, with a 132 topology in Sweet vs. 123 in GPCRs. See Fig. 7) The ligand has usually a fragment lying on the axis of symmetry with TM3/TM7 anchor residues offering a pseudosymmetric structural arrangement, but also an intriguing sequence pattern, always different across different GPCRs. In the chosen example, in Class C, while there is an offset between the ligand and the axis of symmetry, the binding region of TM3/7 offers a sequence pattern **VxLS** with surrounding residues providing binding. The same is true in Retinal, for example, with a matching **FFA(K/T)** pattern between TM3 and TM7 preceding the anchor Lysine residue to Retinal. We have performed systematic multiple alignments on known GPCR structures, and one can find recurrent patterns of matching residue pairs and ligand-binding positions around in structurally matching TM3/TM7 in each structure independently [67].

3.2.7 Multilevel Symmetries: Complexity Buildup and “Quaternary Topologies”

Inter-protodomain interfaces together with inter-domain quaternary interfaces may allow an interesting decoding of functional biological units’ complexity buildup. In Fig. 4, we can see how domain-level symmetry and dimer symmetry can produce a pseudo-D2 overall symmetry, reflected in the eight-stranded central beta barrel composed of 4 hairpins ($2^* \text{G|F||C|C}'$) (see Figs. 2 and 4).

Another example of higher symmetry buildup with a domain level and a quaternary structure symmetry can be found in the Sm fold. In Fig. 8 we represent Hfq, the bacterial Sm hexamer. The Sm barrel is a small beta barrel (SBB) of SH3-like topology, usually considered as a five-stranded beta barrel, but it is better represented as a six-membered barrel sandwich with a highly bent central strand (that we split in two) that bridges two orthogonal sheets of the barrel. Even a small barrel composed of ca. 50 residues can exhibit C2 symmetry, bringing down to 20–25 residues the protodomain size with a b|b-b topology formed by a hairpin and a third strand connected with either a simple glycine or a 3–10 helix (the protodomain alignment is available in Fig. S3). Both result in bridging the two beta sheets of the small barrel sandwich, in a similar way that Greek key loops bridge the sheets of an Ig barrel.

A three-stranded Sheet B of a monomer can then dock laterally with a three-stranded Sheet A of another monomer to form a six-stranded antiparallel sheet (blade) and so on in building a six-bladed doughnut-shaped ring hexamer. Sm-like oligomers

assemble through their b4–b5 strands as homo-hexamers in bacteria, homo-heptamers in archaea, and hetero-heptamers in eukaryotes. We can use 2D topology/sequence maps to represent a domain but also the formation of a larger hexamer (Figs. 8 and S4).

3.2.8 RNA Protodomains

Finally, proteins are not the only biological macromolecules to exhibit pseudosymmetry or as a matter of fact secondary structure. We mentioned the proto-ribosome but also riboswitches that offer splendid examples of pseudosymmetry. This has been reviewed elsewhere [28]. In Fig. 9 we show a riboswitch “protodomain” decomposition [47], which also presents a symmetric ligand-binding pattern.

The symmetry detection was performed directly with CE-symm [1] (<https://github.com/rcsb/symmetry>), as it can operate on proteins, RNA, and DNA.

4 Notes

4.1 Pseudosymmetry Is Found by a Computer Program, Yet Protodomains Exact Delineation and Alignment Need Some Refinement

The two programs that detect pseudosymmetries [1, 38, 39] offer examples where one can find a symmetry with one and not the other. Hence users tend to use both to identify symmetry and protodomains. The latter will vary slightly when symmetry is found by both. This is common. Computer programs use different algorithms, and scientists tend to use a consensus approach by using more than one program for asserting a result or making sure nothing may be missed [68, 69]. However, although programs have become better to delineate protodomains, there are (many) cases when pseudosymmetry is found but needs refinement in structure alignment, and depending on the goal of such alignment, one may also introduce some shifts in sequence (by 1–2 amino acids eventually in a SSE). Programs may be used with various parameters, but by experience, if one is trying to get at the most accurate structural alignment, manual alignment has no substitute. It is a well-known phenomenon of pattern recognition from a human eye. Even if a structural alignment gets a very good match, if one is interested in sequence conservation among protodomains, this can be optimized starting from a structure-based protodomain delineation, and sometimes a shift by one or two residues could reveal an internally conserved sequence pattern in helical systems (in helical systems, a structural alignment can be shifted up to 4–5 residues without significant change in RMSD. This corresponds to a helix turn shift along a helix axis. Patterns such as in Fig. 7 for GPCRs are an example). Whether that pattern is meaningful or not will necessitate deeper analysis.

4.2 Pseudosymmetry Is Not Identified by a Computer Program

Naturally, even if 20% of domains may possess pseudosymmetry, and even if this number may be conservative, as it was determined using one representative per superfamily, the majority of individual protein domains do not exhibit pseudosymmetry. Hence if one does not identify symmetry, there are chances it is a correct assessment by the program. It can however be frustrating to miss pseudosymmetry if it is not detected. The problem with non-detection of symmetry may be linked to different factors, when symmetry should be identified: programs use numerical cutoffs on all sorts for internal parameters. On the other hand, and it is often the case, structure quality varies, as well offer some conformational variability. If symmetry is not detected on one structure, it may be detected on a homologous one for a given set of default parameters. So, one may use alternative structures. Also, some parameters can be adjusted from default values, and sometimes this can be an iterative process, depending on the candidate structure analyzed. A very useful parameter to adjust in CE-symm, for example, is the maximum RMSD between symmetry-related protodomains (-maxrmsd), which can be made stringent (ca. 2A RMS for beta folds and 3–3.5A in alpha, but these can be varied by increments to seek a significant alignment). This will lead to protodomain delineation that may be shorter in aligned length but with more accurate structural alignments. In some cases, this can give as good results as manual alignments.

4.3 Checking a Pseudosymmetry Interactive Alignment

If residues are aligned, their symmetry-transformed images must be aligned. Even if dual alignments of protodomain 1 on 2 (i and j in general) and vice versa simultaneously cannot be performed by a program such as Cn3D during interactive alignment, one can check this reciprocal match visually during an interactive alignment with a simple equivalence rule (see Fig. S3 in the example of the Sm fold):

$$\text{If Res. } i \text{ (domain)} \Leftrightarrow \text{Res. } j \text{ (domain copy)} \text{ then Res. } j \text{ (domain)} \Leftrightarrow \text{Res. } i \text{ (domain copy)}$$

4.4 Helical Protein Structural Alignment and 7TM GPCR Fold (f.13)

Membrane proteins are classified as a separate class within SCOP (Class F) regardless of their secondary structure makeup. They show a higher pseudosymmetry rate than other structural classes of globular proteins, apart from all beta structures (Class B). This has been widely reviewed [58]. Although Rhodopsin pseudosymmetry has been detected anecdotally in the literature [62], no current symmetry detection program [1, 38] can detect GPCR pseudosymmetry systematically. While we now have over 100 GPCR structures in the PDB database, pseudosymmetry is detected for a handful of GPCRs by the CE-symm software that often require stringent matching criteria for protodomains (see above). Rhodopsin, a Class A GPCR, or the Metabotropic Glutamate Receptor 1, a Class C GPCRs, for which structures are

available, are cases where one can find a pseudosymmetry with a stringent criterion. We use interactive structural alignment (Cn3D) to lead to accurate alignments (Fig. 7). It is frequent in helical protein structural alignments to match structures within a 3.5–4 Å RMSD, even when sequence matching is indicative of homology. Most of our reliable protodomain alignments for helical structures will lie between 2.0 and 4.0 Å RMS, with the majority in the 2.5–3.5 Å RMSD between protodomains made of helical SSSs. This would also be the case in alpha-beta structures, while pure beta will tend to show lower RMS deviations between well-delineated protodomains, i.e., in the 1–3 Å range. This is mostly due to the relative translational movements of helices along their helix axes for corresponding helical SSEs, while beta structure matches do not have this degree of freedom. In helical systems nonbonded interactions are responsible for SSSs formation, while in beta structures, strands and inter-strands' H-bond networks forming sheets lead to a higher structural conservation of SSSs.

4.5 Pseudosymmetry Is Detected But May or May Not Be Relevant

It also happens, when we do not expect pseudosymmetry, that the program identifies one or more symmetry operations we did not expect. This happens in detecting multiple levels of symmetry; sometimes one finds more symmetries than expected. This can be quite useful if it points to a local symmetry. It may also be irrelevant depending on the objective of symmetry detection. These higher than expected symmetries are interesting in a sense that they are purely geometric and can help understand an overall architecture better. They may also be useful for protein engineering, as opposed to, for example, identifying evolutionarily related duplication-fusion events. There are also local symmetries that may be of interest if they can be related to a function. The concept of local symmetry is ubiquitous in chemistry [70], even on small substructural groups such as -CH₂ or -CH₃ motifs (with C₂v and C₃v local symmetries). Without going down to that level for proteins, one can find pseudosymmetric arrangements at the supersecondary structure level as in, for example, the interaction between 2 hairpins in a larger, overall asymmetric protein; it may be interesting then to look at it from a point of view of local symmetry and departure from symmetry.

4.6 “Translational Symmetry”: Structural Repeats Related by Arbitrary Rotation-Translation

In looking for symmetry-related protodomains, one may miss structural repeats whose arrangement is related by an arbitrary rotation-translation. Tandem repeats are one example. It occurs extremely frequently, for example, in DNA-binding proteins. While this is a clear pattern for tandem domain duplication, it may be hidden if structural elements are small SSSs that repeat without symmetry. This is also called a “translational” symmetry.

4.7 SSE Swapping

Numerous structures with cyclic symmetry of order n , where $n > 2$, show a tendency to form “closed” structures, where the n th repeat interfaces with the first one in the same way as any other pair of consecutive repeats, but without a linker however. In these cases, one often observes one or more SSEs swapping between the terminal repeats/protodomains. It is quite common in beta structures to observe strand swapping. This is the case, for example, in propellers. This is also observed in alpha structures. For a review on domain swapping, see [71].

4.8 The Question Is: Are Well-Defined

Protodomains Shared Among Different Folds?

Structurally conserved fragments across domains have been widely observed, some forming well-defined SSSs, and they have been puzzling to a number of scientists. Some authors have been trying to even identify a set of such fragments as forming the base for a structural “vocabulary” of ancient peptides at the origin of the formation of current domains [72]. They believed that “assembly from non-identical fragments may have been one of the primary forces in the evolution of domains” but, to their surprise, “did not find even one domain that contained two or more different fragments from their set of fragments.” They found “instead that fragments either form folds by repetition or in single copy, decorated by heterologous structural elements, “finding the reasons for the lack of fragment combinations unclear.”

In fact, this is consistent with our findings on protodomains. Repetitive SSSs are highly idiosyncratic in forming domains when combining symmetrically to form domains. In other words, protodomain SSSs represent a signature of a pseudosymmetric domain/fold and may not be found in any other domain/fold. This is an interesting observation, as it points to self-association as a driving force in the formation of pseudosymmetric domains. In other words, self-association seems to be a cause of the observed symmetric organization of pseudosymmetric domains. We did not yet assess this observation exhaustively on the whole protein universe, nor can we be sure the lack of “hits” across domains in the PDB is not due to technical limitations of our current tools. We plan to perform such systematic studies in the future. In most cases where we searched for a protodomain across domains in the PDB, we did not find any other domain (“hit”) other than the pseudosymmetric domains formed out of the protodomain or homologs, except a handful of cases. More cases naturally must exist, yet the proportion vs. pseudosymmetric protodomain assemblies should be small but certainly highly instructive on evolution [73].

4.9 Tools of the Trade

All along we have been using structure alignment to delineate protodomains. It gives us a tool to compare sequences that may have an interest by themselves in terms of evolution of domains in the classical sense. Yet much more important are the protodomain-protodomain interactions, the interfaces they form, and from

where they coevolved to form a specific domain with a specific function (or more) using a universal self-associating principle of supersecondary structures. The current method is in its infancy both in terms of tools and applications. Current alignment tools using structure/sequence are limited in looking at one dimension of the problem: structure/sequence similarity. We need a tool to study the self-complementarity of these SSSs and their constituting SSEs in parallel or in antiparallel, for strands and helices at a minimum. Hence, we need to develop complementary alignment tools, where we can match sequence and assembly of these sequences as we match complementary structural elements. This, naturally, can and should extend to any quaternary arrangement. Pseudosymmetric domain arrangements are simply, in that regard, pseudoquaternary structures.

Pseudosymmetry gives correspondences (pseudo-equivalence) at all levels: between SSSs, between SSEs, and down to the residue level, as we have seen in examples. It should find applications in the study of protein folding and structure-function relationships and certainly in the study of coevolution of protein domains and their quaternary arrangements at various levels of complexity. It is also reasonable to think that applications may be found at the local symmetry level. In fact, we already use local symmetry in the very definition of secondary structures, alpha helices and beta strands, which are periodic in nature. Symmetry at the supersecondary structure level is a natural step up in complexity that still reveals periodicity. After all, symmetry is an overarching principle in all Sciences at all scales [74]. It should in fact be seen as surprising that we do not make a wider use of symmetry in proteins.

4.10 Structures Used in This Work

- CD8: 1CD8 [75], 2ATP [76], 5EDX/5 EB9 [77].
- PD-1/PD-L1: 4ZQK [78].
- IgV: 5ESV [79].
- VNAR: 1VES [50].
- IgC: 3DJ9 [80], 4N0U [81].
- FN3: 1BQU [82], 3HHR [83], 2ERJ [84], 3TGX [56].
- Sweet/SemiSweet: 5CTH [85], 4QNC/4QND [86].
- GPCR: 4OR2 [65], 1F88 [63].
- Sm/Hfq: 1KQ2 [87].
- Riboswitch: 3F2Q [47].
- P53: 1TUP [88].
- Nucleosome: 3C1B [89].

Acknowledgments

I would like to thank Jiyao Wang at NCBI who developed most of iCn3D software and has been working very hard to release the new version of the software to allow some key visualization on time for this paper and all members of the NCBI Structure group headed by Steve Bryant who participated; Peter Rose at SDSC who guided me through RCSB's symmetry categorizations in quaternary structure and who developed the symmetry visualization in Jmol, used at RCSB and within CE-symm; and Spencer Bliven and Aleix Latifa who developed CE-symm further to allow multilevel symmetry determination, both at the quaternary and tertiary levels simultaneously. A special thought for Guido Capitani who supported that last effort and who passed away last year, far too young, before we had time to join forces on tertiary/quaternary structural analysis. I miss him both at a personal level and scientifically. Thank you to Stella Veretnik for discussions over the years on small beta barrels. Thank you to Phil Bourne who gave me the opportunity to resume work on pseudosymmetry at the NIH while initiated long ago at Columbia University with Cy Levinthal, Barry Honig, and Wayne Hendrickson. Thank you to Tom Misteli at the National Cancer Institute for his support, giving me the opportunity to pursue applications of these concepts in the aim of developing rational design methods for immunotherapy. Finally, I would like to thank Mitchell Ho at the NCI for introducing me to Shark Immunoglobulins.

This research was supported in part by the Intramural Research Program of the National Cancer Institute and the National Library of Medicine, NIH.

References

1. Myers-Turnbull D et al (2014) Systematic detection of internal symmetry in proteins using CE-Symm. *J Mol Biol* 426:2255–2268
2. Alewine C, Hassan R, Pastan I (2015) Advances in anticancer immunotoxin therapy. *Oncologist* 20:176–185
3. Kochenderfer JN, Rosenberg SA (2013) Treating B-cell cancer with T cells expressing anti-CD19 chimeric antigen receptors. *Nat Rev Clin Oncol* 10:267–276
4. Chothia C, Lesk AM (1987) Canonical structures for the hypervariable regions of immunoglobulins. *J Mol Biol* 196:901–917
5. Chothia C, Novotný J, Brucolieri R, Karplus M (1985) Domain association in immunoglobulin molecules. The packing of variable domains. *J Mol Biol* 186:651–663
6. Díaz-Ramos MC, Engel P, Bastos R (2011) Towards a comprehensive human cell-surface immunome database. *Immunol Lett* 134:183–187
7. Nacim F, Nagesh Rao P, Song SX, Grody WW (2013) Atlas of hematopathology. Academic, New York, pp 25–46. <https://doi.org/10.1016/B978-0-12-385183-3.00002-4>
8. McLachlan AD (1972) Gene duplication in carp muscle calcium binding protein. *Nat New Biol* 240:83–85
9. Blundell TL, Sewell BT, McLachlan AD (1979) Four-fold structural repeat in the acid proteases. *Biochim Biophys Acta* 580:24–31
10. McLachlan AD (1987) Gene duplication and the origin of repetitive protein structures. *Cold Spring Harb Symp Quant Biol* 52:411–420

11. Hendrickson WA, Ward KB (1977) Pseudosymmetry in the structure of myohemerythrin. *J Biol Chem* 252:3012–3018
12. Eck RV, Dayhoff MO (1966) Evolution of the structure of ferredoxin based on living relics of primitive amino acid sequences. *Science* 152:363–366
13. Urbain J (1969) Evolution of immunoglobulins and ferredoxins and the occurrence of pseudosymmetrical sequences. *Biochem Genet* 3:249–269
14. Barker WC, Ketcham LK, Dayhoff MO (1978) A comprehensive examination of protein sequences for evidence of internal gene duplication. *J Mol Evol* 10:265–281
15. Delhaise P, Wuilmart C, Urbain J (1980) Relationships between alpha and beta secondary structures and amino-acid pseudosymmetrical arrangements. *Eur J Biochem* 105:553–564
16. Lo Conte L et al (2000) SCOP: a structural classification of proteins database. *Nucleic Acids Res* 28:257–259
17. Chandonia J-M, Fox NK, Brenner SE (2017) SCOPe: manual curation and artifact removal in the structural classification of proteins—extended database. *J Mol Biol* 429:348–355
18. Sillitoe I, Dawson N, Thornton J, Orengo C (2015) The history of the CATH structural classification of protein domains. *Biochimie* 119:209–217
19. Cheng H et al (2014) ECOD: an evolutionary classification of protein domains. *PLoS Comput Biol* 10:e1003926
20. Goodsell DS, Olson AJ (2000) Structural symmetry and protein function. *Annu Rev Biophys* 29:105–153
21. Levy ED, Pereira-Leal JB, Chothia C, Teichmann SA (2006) 3D complex: a structural classification of protein complexes. *PLoS Comput Biol* 2:e155
22. Rose PW et al (2015) The RCSB Protein Data Bank: views of structural biology for basic and applied research and education. *Nucleic Acids Res* 43:D345–D356
23. Young JY et al (2018) Worldwide Protein Data Bank biocuration supporting open access to high-quality 3D structural biology data. *Database* 2018
24. Levy ED, Boeri Erba E, Robinson CV, Teichmann SA (2008) Assembly reflects evolution of protein complexes. *Nature* 453:1262–1265
25. Blaber M, Lee J, Longo L (2012) Emergence of symmetric protein architecture from a simple peptide motif: evolutionary models. *Cell Mol Life Sci* 69:3999–4006
26. Andrade MA, Perez-Iratxeta C, Ponting CP (2001) Protein repeats: structures, functions, and evolution. *J Struct Biol* 134:117–131
27. Abraham A-L, Pothier J, Rocha EPC (2009) Alternative to homo-oligomerisation: the creation of local symmetry in proteins by internal amplification. *J Mol Biol* 394:522–534
28. Jones CP, Ferré-D’Amaré AR (2015) RNA quaternary structure and global symmetry. *Trends Biochem Sci* 40:211–220
29. Bashan A et al (2003) Structural basis of the ribosomal machinery for peptide bond formation, translocation, and nascent chain progression. *Mol Cell* 11:91–102
30. Lehn J-M (2002) Toward self-organization and complex matter. *Science* 295:2400–2403
31. Lehn J-M (2013) Perspectives in chemistry—steps towards complex matter. *Angew Chem Int Ed Engl* 52:2836–2850
32. Gutmanas A et al (2014) PDBe: Protein Data Bank in Europe. *Nucleic Acids Res* 42: D285–D291
33. Kinjo AR et al (2017) Protein Data Bank Japan (PDBj): updated user interfaces, resource description framework, analysis tools for large structures. *Nucleic Acids Res* 45:D282–D288
34. Marchler-Bauer A et al (2015) CDD: NCBI’s conserved domain database. *Nucleic Acids Res* 43:D222–D226
35. Madej T et al (2014) MMDB and VAST+: tracking structural similarities between macromolecular complexes. *Nucleic Acids Res* 42: D297–D303
36. Wang Y, Geer LY, Chappay C, Kans JA, Bryant SH (2000) Cn3D: sequence and structure views for Entrez. *Trends Biochem Sci* 25:300–302
37. Madej T et al (2012) MMDB: 3D structures and macromolecular interactions. *Nucleic Acids Res* 40:D461–D464
38. Kim C, Basner J, Lee B (2010) Detecting internally symmetric protein structures. *BMC Bioinformatics* 11:303
39. Tai C-H, Paul R, Dukka KC, Shilling JD, Lee B (2014) SymD webserver: a platform for detecting internally symmetric protein structures. *Nucleic Acids Res* 42:W296–W300
40. Wang J, Youkhariabache P, Zhang D, Lanczycki CJ, Geer RC, Madej T, Phan L et al (2018) iCn3D, a web-based 3D viewer for the visualization of biomolecular structure and sequence annotation. *bioRxiv*. <https://doi.org/10.1101/501692>
41. Jmol: an open-source browser-based HTML5 viewer and stand-alone Java viewer for chemical

- structures in 3D. <http://jmol.sourceforge.net/>
42. Rose AS, Hildebrand PW (2015) NGL Viewer: a web application for molecular visualization. *Nucleic Acids Res* 43:W576–W579
 43. Stivala A, Wybrow M, Wirth A, Whisstock JC, Stuckey PJ (2011) Automatic generation of protein structure cartoons with Pro-origami. *Bioinformatics* 27:3315–3316
 44. Youkharibache P (2017) Twelve elements of visualization and analysis for tertiary and quaternary structure of biological molecules. *bioRxiv* 153528. [10.1101/153528](https://doi.org/10.1101/153528)
 45. Mura C, Randolph PS, Patterson J, Cozen AE (2013) Archaeal and eukaryotic homologs of Hfq: A structural and evolutionary perspective on Sm function. *RNA Biol* 10:636–651
 46. Youkharibache P et al (2019) The small β -barrel domain: a survey-based structural analysis. *Structure* 27 (1): 6–26. <https://doi.org/10.1016/j.str.2018.09.012>
 47. Serganov A, Huang L, Patel DJ (2009) Coenzyme recognition and gene regulation by a flavin mononucleotide riboswitch. *Nature* 458:233–237
 48. Patikoglou GA et al (1999) TATA element recognition by the TATA box-binding protein has been conserved throughout evolution. *Genes Dev* 13:3217–3230
 49. Stanfield RL, Dooley H, Flajnik MF, Wilson IA (2004) Crystal structure of a shark single-domain antibody V region in complex with lysozyme. *Science* 305:1770–1773
 50. Streltsov VA et al (2004) Structural evidence for evolution of shark Ig new antigen receptor variable domain antibodies from a cell-surface receptor. *Proc Natl Acad Sci U S A* 101:12444–12449
 51. Feige MJ et al (2014) The structural analysis of shark IgNAR antibodies reveals evolutionary principles of immunoglobulins. *Proc Natl Acad Sci U S A* 111:8155–8160
 52. Kabat EA, Wu TT, Reid-Miller M, Perry HM, Gottesman KS (1987) Sequences of proteins of immunological interest, 4th ed. National Institutes of Health, Bethesda
 53. Lefranc M-P et al (2003) IMGT unique numbering for immunoglobulin and T cell receptor variable domains and Ig superfamily V-like domains. *Dev Comp Immunol* 27:55–77
 54. Zhang Y-F, Ho M (2017) Humanization of rabbit monoclonal antibodies via grafting combined Kabat/IMGT/Paratome complementarity-determining regions: Rationale and examples. *MAbs* 9:419–429
 55. Siupka P, Hamming OT, Kang L, Gad HH, Hartmann R (2015) A conserved sugar bridge connected to the WSXWS motif has an important role for transport of IL-21R to the plasma membrane. *Genes Immun* 16:405–413
 56. Hamming OJ et al (2012) Crystal structure of interleukin-21 receptor (IL-21R) bound to IL-21 reveals that sugar chain interacting with WSXWS motif is integral part of IL-21R. *J Biol Chem* 287:9454–9460
 57. Baumgartner JW, Wells CA, Chen CM, Waters MJ (1994) The role of the WSXWS equivalent motif in growth hormone receptor function. *J Biol Chem* 269:29094–29101
 58. Forrest L, Structural R (2015) Symmetry in membrane proteins. *Annu Rev Biophys* 44:311–337
 59. Forrest LR (2013) Structural biology. (Pseudo-)symmetrical transport. *Science* 339:399–401
 60. Feng L, Frommer WB (2015) Structure and function of SemiSWEET and SWEET sugar transporters. *Trends Biochem Sci* 40:480–486
 61. Hu Y-B et al (2016) Phylogenetic evidence for a fusion of archaeal and bacterial SemiSWEETs to form eukaryotic SWEETs and identification of SWEET hexose transporters in the amphibian chytrid pathogen Batrachochytrium dendrobatidis. *FASEB J* 30:3644–3654
 62. Choi S, Jeon J, Yang J-S, Kim S (2008) Common occurrence of internal repeat symmetry in membrane proteins. *Proteins* 71:68–80
 63. Palczewski K et al (2000) Crystal structure of rhodopsin: A G protein-coupled receptor. *Science* 289:739–745
 64. Li J, Edwards PC, Burghammer M, Villa C, Schertler GF (2004) Structure of bovine rhodopsin in a trigonal crystal form. *J Mol Biol* 343:1409–1438
 65. Wu H et al (2014) Structure of a class C GPCR metabotropic glutamate receptor 1 bound to an allosteric modulator. *Science* 344:58–64
 66. Christopher JA et al (2015) Fragment and structure-based drug discovery for a class C GPCR: discovery of the mGlu5 negative allosteric modulator HTL14242 (3-Chloro-5-[6-(5-fluoropyridin-2-yl)pyrimidin-4-yl]benzonitrile). *J Med Chem* 58:6653–6664
 67. Youkharibache P, Tran A, Abrol R (2018) 7-Transmembrane Helical (7TMH) Proteins: Pseudo-Symmetry and Conformational Plasticity. *bioRxiv*. <https://doi.org/10.1101/465302>
 68. Stamm M, Forrest LR (2015) Structure alignment of membrane proteins: comparison of available tools and a consensus strategy. *Proteins* 83(9):1720–1732

69. Korkmaz S et al (2017) Quaternary structure evaluation tool for protein assemblies. *bioRxiv* 224196. <https://doi.org/10.1101/224196>
70. Kettle SFA (2007) Symmetry and structure: readable group theory for chemists. Wiley. <https://market.android.com/details?id=book-KoywQgAACAAJ>
71. Liu Y, Eisenberg D (2002) 3D domain swapping: as domains continue to swap. *Protein Sci* 11:1285–1299
72. Alva V, Söding J, Lupas AN (2015) A vocabulary of ancient peptides at the origin of folded proteins. *elife* 4:e09410
73. Petrey D, Fischer M, Honig B (2009) Structural relationships among proteins with different global topologies and their implications for function annotation strategies. *Proc Natl Acad Sci U S A* 106:17377–17382
74. Kellman ME (1996) Symmetry in chemistry from the hydrogen atom to proteins. *Proc Natl Acad Sci U S A* 93:14287–14294
75. Leahy DJ, Axel R, Hendrickson WA (1992) Crystal structure of a soluble form of the human T cell coreceptor CD8 at 2.6 Å resolution. *Cell* 68:1145–1162
76. Chang H-C et al (2005) Structural and mutational analyses of a CD8alphabeta heterodimer and comparison with the CD8alphaalpha homodimer. *Immunity* 23:661–671
77. Liu Y, Li X, Qi J, Zhang N, Xia C (2016) The structural basis of chicken, swine and bovine CD8 $\alpha\alpha$ dimers provides insight into the co-evolution with MHC I in endotherm species. *Sci Rep* 6:24788
78. Zak KM et al (2015) Structure of the complex of human programmed death 1, PD-1, and its ligand PD-L1. *Structure* 23:2341–2348
79. Gorman J et al (2016) Structures of HIV-1 Env V1V2 with broadly neutralizing antibodies reveal commonalities that enable vaccine design. *Nat Struct Mol Biol* 23:81–90
80. Prabakaran P et al (2008) Structure of an isolated unglycosylated antibody C(H)2 domain. *Acta Crystallogr D Biol Crystallogr* 64:1062–1067
81. Oganessian V et al (2014) Structural insights into neonatal Fc receptor-based recycling mechanisms. *J Biol Chem* 289:7812–7824
82. Bravo J, Staunton D, Heath JK, Jones EY (1998) Crystal structure of a cytokine-binding region of gp130. *EMBO J* 17:1665–1674
83. de Vos AM, Ultsch M, Kossiakoff AA (1992) Human growth hormone and extracellular domain of its receptor: crystal structure of the complex. *Science* 255:306–312
84. Stauber DJ, Debler EW, Horton PA, Smith KA, Wilson IA (2006) Crystal structure of the IL-2 signaling complex: paradigm for a heterotrimeric cytokine receptor. *Proc Natl Acad Sci U S A* 103:2788–2793
85. Tao Y et al (2015) Structure of a eukaryotic SWEET transporter in a homotrimeric complex. *Nature* 527:259–263
86. Xu Y et al (2014) Structures of bacterial homologues of SWEET transporters in two distinct conformations. *Nature* 515:448–452
87. Vrentas C et al (2015) Hfq in *Bacillus anthracis*: role of protein sequence variation in the structure and function of proteins in the Hfq family. *Protein Sci* 24:1808–1819
88. Cho Y, Gorina S, Jeffrey PD, Pavletich NP (1994) Crystal structure of a p53 tumor suppressor-DNA complex: understanding tumorigenic mutations. *Science* 265:346–355
89. Lu X et al (2008) The effect of H3K79 dimethylation and H4K20 trimethylation on nucleosome and chromatin structure. *Nat Struct Mol Biol* 15:1122–1124
90. Pettersen EF et al (2004) UCSF Chimera—a visualization system for exploratory research and analysis. *J Comput Chem* 25:1605–1612



Chapter 11

$\beta\alpha\beta$ Super-Secondary Motifs: Sequence, Structural Overview, and Pursuit of Potential Autonomously Folding $\beta\alpha\beta$ Sequences from $(\beta/\alpha)_8$ /TIM Barrels

Rajasekhar Varma Kadamuri, Shivkumar Sharma Irukuvajjula, and Ramakrishna Vadrevu

Abstract

$\beta\alpha\beta$ super-secondary structures constitute the basic building blocks of $(\beta/\alpha)_8$ class of proteins. Despite the success in designing super-secondary structures, till date, there is not a single example of a natural $\beta\alpha\beta$ sequence known to fold in isolation. In this chapter, to address the finding the “needles” in the haystack scenario, we have combined the sequence preferences and structural features of independent $\beta\alpha\beta$ motifs, dictated by natural selection, with rationally derived parameters from a designed $\beta\alpha\beta$ motif adopting stable fold in solution. Guided by this approach, a set of potential $\beta\alpha\beta$ sequences from $(\beta/\alpha)_8$ /TIM barrels are proposed as likely candidates for autonomously folding based on the assessment of their foldability.

Key words $\beta\alpha\beta$ motifs, Super-secondary structures, Main-chain to side-chain hydrogen bonding, $(\beta/\alpha)_8$ barrels, TIM barrel

1 Introduction

The acquired complex three-dimensional conformations of proteins is a culmination of simple structural fragments like $\alpha\text{-}\alpha$, $\beta\text{-}\beta$, $\alpha\text{-}\beta$, and $\beta\text{-}\alpha$ units [1]. The precise sequence and formation of local secondary/super-secondary structures such as $\alpha\text{-}\alpha$ hairpins, β hairpins, and $\beta\alpha\beta$ units and their optimal packing between them lead to thermodynamically stable tertiary structures. Repetition of these super-secondary motifs leads to the formation of the four main protein structural classes: α -helical, β sheet, β/α , and $\alpha + \beta$. In fact, despite the enormous protein sequence space, the observation of merely four major structural classes of proteins, perhaps, reflects the constraint imposed by the limited set of super-secondary motifs [2]. Thus, comprising more than 70%, in protein folds, the super-secondary structures can serve as the building blocks in a modular assembly fashion [3–5] leading to the evolution of structural and

functional domains. Further, their recurrence is hypothesized to accelerate evolution of folded structure via gene duplications, mutations, and shuffling [6, 7]. Another interesting conjecture is that they could probably be the remnants from an ancient peptide world [8]. Also, from studies involving the comparison of domains from different proteins, it became obvious that the complex protein structures have evolved from simple independently folding super-secondary fragments [9, 10]. In a study conducted by Riechmann and Winter, it was observed that new stable proteins can be designed by combinatorial assembly of fragments [11, 12]. In fact, independently folding units (foldons, schemas, and closed loops) proposed by various studies matched broadly to the existing super-secondary structures motifs in protein structures [13, 14]. In this direction, the past decade has witnessed efforts from various research groups put into the design, prediction, and analysis of sequences that can adopt structures and acquire native-like stable conformations. The observation of secondary and super-secondary structures in isolated peptide fragments forms proteins, and in de novo designed sequences underscore the role of independently folding structural units and their role in assembly of higher-order structures [15–27].

Given the knowledge of first principles of folding, stability, and sequence to structure relationship obtained from design and prediction of autonomously folding units further encouraged researchers to design/build completely new proteins, but designing of larger proteins has been very challenging. However, indeed, the knowledge gleaned from the folded tertiary structures helped decipher the role of local interactions in folding and stability of secondary and super-secondary structural units [28, 29]. Plethora of studies of peptide fragments that were designed to fold into stable independent motifs like helix-loop-helix, beta-beta-alpha, beta-sheets, etc. were aggressively pursued leading to the accumulation of vast information on sequence and structural features contributing to the folding and stability of protein building block motifs [15–26, 30, 31]. Although experimental and theoretical studies unambiguously indicated clearly that the $\beta\alpha\beta$ structural unit acts as a minimal unit of folding and stability of $(\beta/\alpha)_8$ /TIM barrel proteins [32–35], it is striking to note that, so far, there is not a single instance of a $\beta\alpha\beta$ sequence from natural $(\beta/\alpha)_8$ proteins that adopts a stable native fold in isolation. This prompted us to ask ourselves, if autonomously folding $\beta\alpha\beta$ motifs be found from natural sequences and if so can they be identified. In this chapter, we have attempted to address the finding the needles in the haystack scenario; firstly, we obtained an overview of the $\beta\alpha\beta$ super-secondary structures from TIM barrels. Sequence and structural analysis of natural $\beta\alpha\beta$ sequences from TIM barrels will shed light on the optimized parameters due to evolutionary pressure for conserved (1) position-specific preferences of amino acids and

(2) favorable enthalpic contributions. Knowledge, thus obtained combined with the rational aspects of folding and stability derived from a designed $\beta\alpha\beta$ motif capable autonomous folding in aqueous solution, was then applied to explore and propose native $\beta\alpha\beta$ sequences from the barrel folds that can potentially fold in isolation.

2 TIM Barrel Fold: $\beta\alpha\beta$ Super-Secondary Structures

TIM barrel fold belonging to the α/β class of proteins is a very frequent class of enzymes observed in more than 10% of the known protein structures. They are known to occur in five of the six enzyme classes [32]. The fold is made up of a regular repeating $\beta\alpha\beta$ motif leading to the arrangement β strands and α helices in an alternating repetitive pattern. The repetitive arrangement of $\beta\alpha\beta$ motif in TIM barrel fold results in the formation of a stable inner core made up of β strands shielded by amphipathic α helices. The loops are categorized as α/β and β/α loops connecting α helices to the β strands and β strands to the α helices, respectively. Given their catalytic versatility, TIM barrel proteins have been the target of interest for protein manipulations and designing studies [36–41].

2.1 Sequence and Structural Features

The typical size of the independent $\beta\alpha\beta$ units, consisting of two parallel strands and an intervening α helix connected by the two kinds of loops, peaks between 30 and 45 residues in length (Fig. 1) and consistent with the size of sequences observed in the formation of autonomous folding (foldons, schemas, closed loops) [13, 42, 43, 14, 28]. Not surprisingly, the ideal lengths of the individual structural components of the $\beta\alpha\beta$ motif add up to the idealized size of the $\beta\alpha\beta$ motifs. The typical length of α helices hovers between 9 and 14, and the length of the 2 parallel β strands, adjacent to α helices, peaks between 4 and 5 residues. From our previous studies, it was observed that $\alpha\beta$ and $\beta\alpha$ loops connecting the helices to strands and strands to helices, respectively, show distinct preferences [44] and have a significant role both in foldability and stability [36, 37]. $\alpha\beta$ loop connections are dominated by 4–7 residues in length, while $\beta\alpha$ loops are longer than 7 residues [44]. Shorter loops, particularly, the $\alpha\beta$ type, could be more effective in promoting hydrophobic clustering of nonpolar amino acids between the N termini of the strands and middle portion of the succeeding α helices (Fig. 2) [44]. For individual lengths of β strands and α helices, positional preferences of amino acids provide further support to this notion. It is apparent from (Fig. 3) that the N termini of β strands are dominated by nonpolar aliphatic amino acids which can effectively interact with the nonpolar amino acids, centered toward the middle portion of the amphipathic α helices.

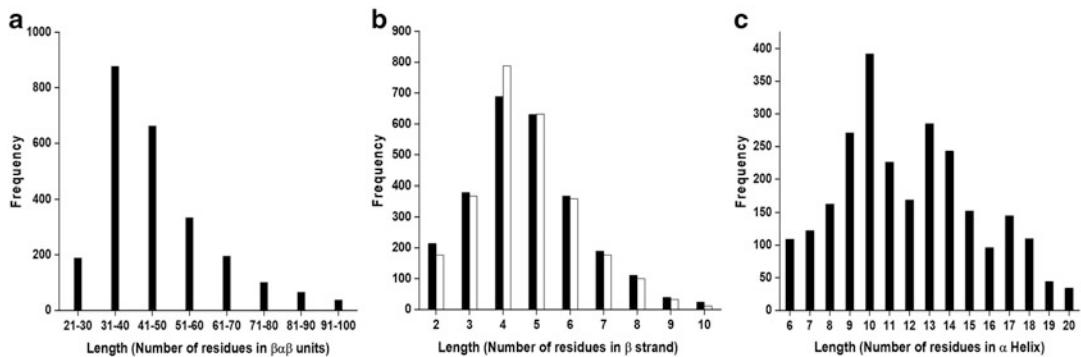


Fig. 1 Length distribution (number of amino acids) of (a) $\beta\alpha\beta$ motifs, (b) β strands, (c) α helices lengths from the dataset of TIM barrel proteins. In (b), open bars represent the number of residues in the strand preceding the central α helix of the $\beta\alpha\beta$ unit. Closed bars represent the number of residues in the succeeding strand

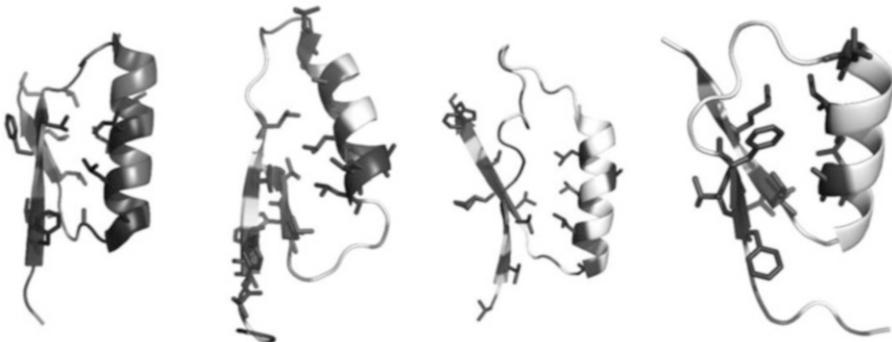
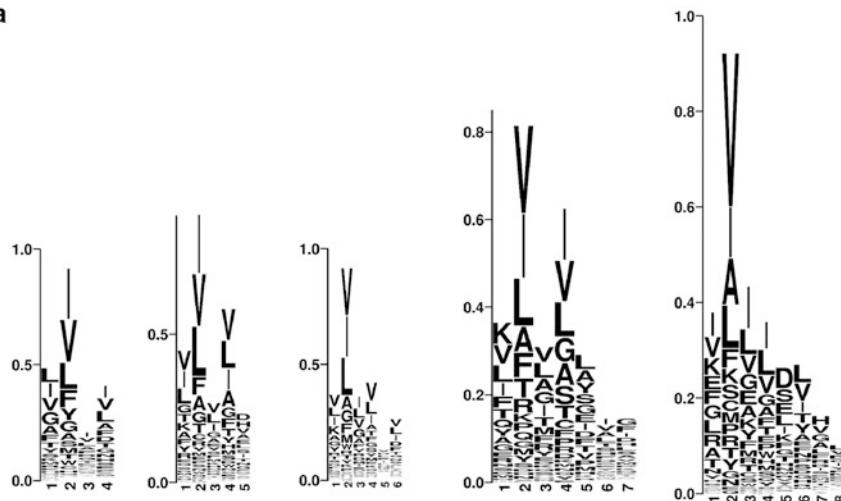


Fig. 2 Representative $\beta\alpha\beta$ units from TIM barrel proteins depicting the interactions between nonpolar residues in the helices and strands. Nonpolar isoleucine, valine, leucine, methionine, phenylalanine, tryptophan, and proline were considered. The nonpolar interactions between the helix and the strands are dominant. The $\beta\alpha\beta$ units are taken from (left to right) from 1V7Y, 1VD6, 2C3Z, and 2XFR, respectively. [60]

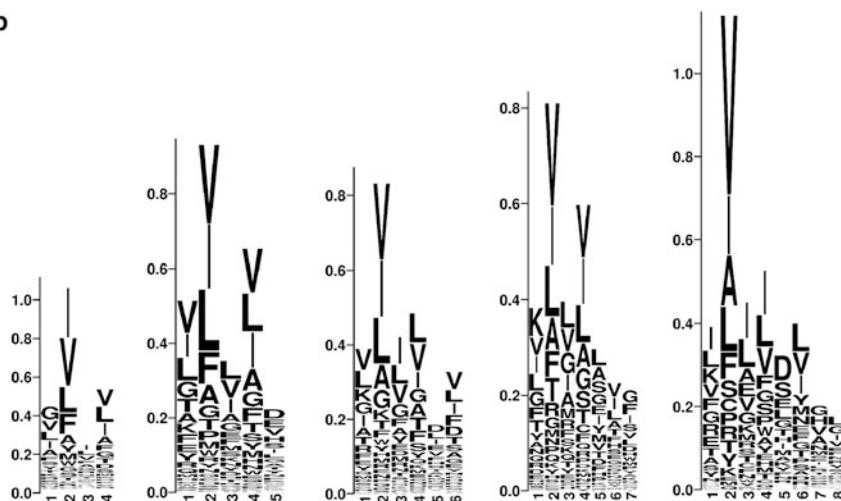
2.2 Long-Range Stabilizing Interactions

Unlike in the full-length proteins, the challenging scenario in the folding and stabilization of independently folding motifs is the lack of adequate enthalpic interactions to compensate for the entropic penalty arising from ordering the chain. Increased network of non-covalent interactions from the inter-residue contacts in smaller fragments of proteins can potentially overcome the unfavorable entropic cost of chain ordering. Our analysis of the TIM barrel structures revealed that the side chains of arginine and lysine participate in an interesting long-range interaction. The polar side chains of arginine and lysine residues, resident in the $\alpha\beta$ loops, are involved in hydrogen-bonding interactions with the main-chain carbonyl oxygen atom of residues in the preceding β strand or the succeeding α helix (Fig. 4). These interactions are reminiscent of the nonlocal side-chain to main-chain hydrogen-bonding interactions, clamping $\beta\alpha$ hairpins and bracketing specific $\beta\alpha\beta$ modules in TIM barrels [33, 45]. It was experimentally demonstrated that

a



b



c

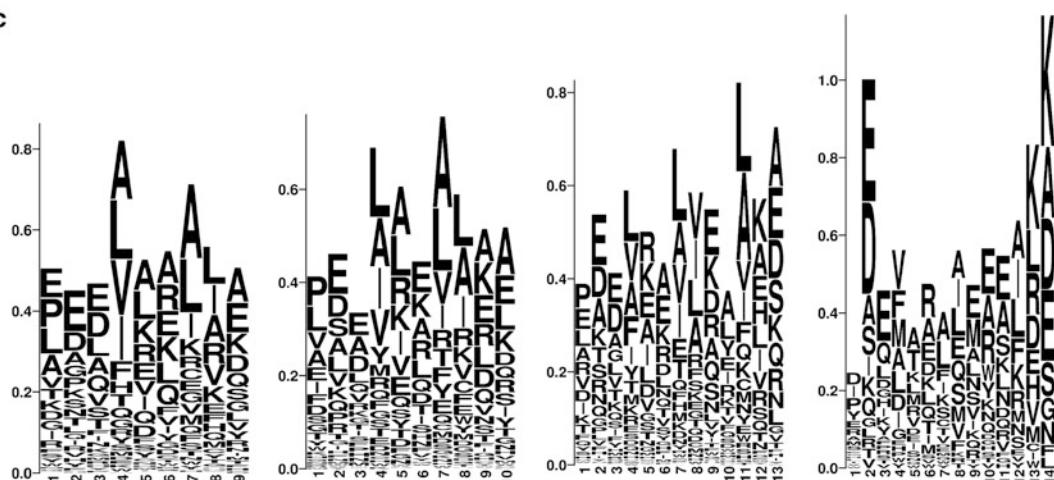


Fig. 3 Sequence profiles of the β strands segregated size wise from the $\beta\alpha\beta$ units. The two β strands from the $\beta\alpha\beta$ units were separately segregated and the pooled according to the size before generating the profiles.

mutations leading to the disruption of main-chain to side-chain $\beta\alpha$ clamp interactions lead to dramatic loss in the stability of TIM barrel proteins [33, 45].

From our overview of the analyses of the independent $\beta\alpha\beta$ units, couple of trends are apparent. First is reduced conformational entropy in the form of short loops, particularly, the $\alpha\beta$ loops which could be a contributing factor for stability. In fact, nature adapted this strategy for enhancing stability in the case of hyperthermophilic proteins, wherein the loop size is considerably smaller than the mesophilic counterparts [46]. Second, the enthalpic contributions arising from (1) clustering of nonpolar amino acids between the β stands and α helices and (2) long-range side-chain to main-chain hydrogen-bonding interactions within the individual $\beta\alpha\beta$ units. In combination with other non-covalent interactions inherent to the $\beta\alpha\beta$ units, these interactions can be an additional contribution to the stability. It is well known that higher density of non-covalent interactions contributes significantly to both folding and stability [46].

2.3 In Pursuit of Autonomously Folding $\beta\alpha\beta$ Units

In a very recent study, it was observed that a de novo designed peptide sequence of 38 amino acids long adopted a well-folded and stable $\beta\alpha\beta$ conformation in aqueous solution [22]. In their design strategy, they have included the basic rules in the design of α helices

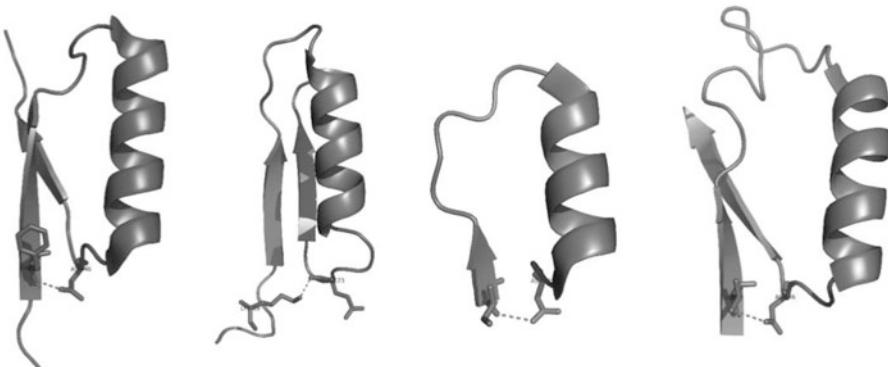


Fig. 4 Long-range side-chain to main-chain hydrogen-bonding interactions ($\beta\alpha$ clamp interactions). Interactions are shown in selected $\beta\alpha\beta$ motifs from TIM barrel proteins. The side chains of the residues involved in the hydrogen bond interactions (dotted lines) with the backbone carbonyl atoms are shown as sticks. The structures were generated using Pymol v 1.8. [61]. The $\beta\alpha\beta$ units are taken (left to right) from 1V7Y, 1VD6, 2XFR, and 3ELF, respectively

Fig. 3 (continued) Positional preferences for the residues in (a) β strand 1 (strand preceding the central α helix), (b) β strand 2 (strand succeeding the central α helix) and (c) helices from the $\beta\alpha\beta$ units. Stack represents conservation at the respective position, and the height indicates the relative frequency of occurrence of the particular amino acid. Amino acids are represented as single letter abbreviations. Figure is generated using Weblogo [61]

and β sheets from observations obtained from the survey of protein structures and $\beta\alpha\beta$ units. Secondary structure preferences of amino acids, short-range interactions between side chains of amino acids, capping preferences, etc. were included in the design of a $\beta\alpha\beta$ motif. Sequence optimization from the initial sequence resulted in a successful design of the following sequence, GSGQVRTIWVGGTP EELKKLKEEAKKANIRVTFWGD, which adopted a well-folded and stable $\beta\alpha\beta$ structure in isolation [22]. From this solitary example of an autonomously folding $\beta\alpha\beta$, we set out to dissect the attributing features within it as benchmark parameters for autonomous folding of naturally occurring $\beta\alpha\beta$ units from TIM barrels.

As a first step, we assessed the propensity of the sequence that adopts α -helix conformation in the $\beta\alpha\beta$ -folded state, using “AGADIR” [47–50], an algorithm that estimates the propensity of a sequence to adopt α -helix conformation based on the experimentally observed percent helicity of a large number of peptide sequences that actually adopted helices in solution [47–50]. AGADIR estimates 39% helicity (at pH 7.0; 25 °C) for the part of the sequence that adopts helix conformation in the designed $\beta\alpha\beta$ motif. Thus, a comparable helix propensity is considered as one of the requirements for the $\beta\alpha\beta$ folding. Second, the loops connecting the two strands to the central helix are short. Third, in the rational design of the above $\beta\alpha\beta$ sequence, the authors placed tryptophan residues (W) at the C-termini of the two β strands [22] to emulate the effective role of tryptophan zippers in stabilizing β hairpins [19, 51]. As a substitute to this stabilizing interaction, the long-range $\beta\alpha$ clamp interactions were considered as an equivalent and one of the essential parameters for $\beta\alpha\beta$ stabilization. The short-listed $\beta\alpha\beta$ candidate sequences deduced from the benchmarked criteria were further assessed for foldability/folding propensity using predictive tools.

2.4 Foldability Assessment of the Potential $\beta\alpha\beta$ Candidates

Sequence-based structure prediction methods were applied to assess the foldability of the potential $\beta\alpha\beta$ candidates short-listed based on the considerations discussed in the earlier sections. The short-listed $\beta\alpha\beta$ sequences along with the autonomously folding designed sequence serving as a positive control for folding ability were subjected to structure predictions. QUARK [52] and PEPFOLD3 [53–55] tools were employed to predict the likeliness of the peptide sequences to fold into independent $\beta\alpha\beta$ units. QUARK is an algorithm [52] for the ab initio protein folding/protein structure prediction. The method predicts the foldable 3D models for the provided protein sequence of less than 200 residues long using replica-exchange Monte Carlo simulations (REMC), simulations guided by an optimized atomic-level knowledge-based force field. QUARK provides ten best likely structure models which are selected based on clustering using revised SPICKER program [56] and template modeling score (TM-score) of models generated from

the REMC simulations [57]. PEPFOLD3 [53–55] is a de novo approach for predicting peptide structure for amino acid sequences of length 5–50 residues long; PEPFOLD3 works based on Hidden Markov Model sub-optimal conformation sampling approach [58], describing the polypeptide chain conformation using a series of local overlapping canonical conformations of 4 residue amino acids fragments. PEPFOLD3 provides five best structure models for the given amino acid sequence based on sOPEP energy (optimized potential for efficient structure prediction) [59].

3 Methods

3.1 Identification and Segregation of $\beta\alpha\beta$ Units from TIM Barrels

A total of 420 nonredundant TIM barrel proteins were mined from Protein Date Bank with <30% sequence similarity and structural resolution of ≤ 3.0 Å. About 2500 $\beta\alpha\beta$ units were extracted from 420 nonredundant TIM barrel proteins using in-house developed python scripts which identified and extracted the $\beta\alpha\beta$ units based on the secondary structure information calculated by DSSP program. Taking into consideration the requirement of short loops, a loop size <14 residues was implemented as the first filter, which resulted in reducing the number of $\beta\alpha\beta$ units to 1608. Following, this, the helix sequence regions of the 1608 $\beta\alpha\beta$ units and the designed $\beta\alpha\beta$ sequence were estimated for their helix forming tendency using AGADIR server at temperature 25 °C and pH 7. The N- and C-termini of the sequences were acetylated and amidated. The estimates from AGADIR ranged from 0 to 53% helix forming tendency for the helix sequences. The helical region of the designed $\beta\alpha\beta$ sequence with 38% estimated helicity when treated as the benchmark score yielded 15 sequences with $\geq 30\%$ estimated helicity. This set of 15 sequences were further manually checked for the presence of long-range side-chain to main-chain interactions and the 10 $\beta\alpha\beta$ sequences found to be possessing the long-range main-chain to side-chain clamps were finally short-listed (Table 1). The entire process of short-listing of potential self-folding $\beta\alpha\beta$ units is summarized and shown in Fig. 5.

4 Structure Prediction of the $\beta\alpha\beta$ Units

The $\beta\alpha\beta$ sequences that were short-listed from the dataset of TIM barrels were assessed for foldability, independently by two different structure prediction methodologies, QUARK and PEPFOLD3. The final 10 $\beta\alpha\beta$ candidate sequences identified, along with the designed sequence, 14 (2KI0), serving as positive control were submitted for structure prediction on QUARK server. The summary of conformations predicted for the sequences is shown in Table 2. $\beta\alpha\beta$ sequences, 3 (3HJE), 6 (2OZT), 8 (3ELF), and

Table 1

Summary of the short-listed potential $\beta\alpha\beta$ candidates arrived from the methodology followed in Figure 5

Sequence number	Sequence source (PDB ID)	Sequence	Helix propensity (AGADIR prediction, %)	Presence of long-range interactions ^a
1	1VD6	PQAVFNVELKSFPG <u>LGE</u> EAA RRLAALL RGREG <i>VWWSSFDP</i>	53	Yes
2	3BOF	GRSLFNSAK VDEEELEM <u>KINLLKKYGGTLI</u> VLLMG	38	Yes
3	3HJE	EVD G YR <u>D</u> HIDGLF <u>KPEEYL</u> RRLK NKIGNKHIFVEKI	38	Yes
4	2XFR	NYV Q V <u>T</u> MLPLDAVSVNN RFEKG <u>DELRAQLRKLV</u> EAG VD G VM <u>V</u> DVWGL	38	Yes
5	3KZP	YIEIKF SLIHFKNIP <u>LEDLLLFIKA</u> <u>WANFAQKNKLDFV</u> VEGIETK	37	No
6	2HZG	HGKR <u>P</u> T <u>YASLLFGDTPQ</u> ETLERA <u>RAARRDGFAAVKF</u> GWGP	37	No
7	1IQ8	GFE I IITNSYIIYK <u>DEEL</u> RRKALELG IHRM <u>LDYNGIIE</u> VDSGS	36	No
8	2QV5	FAD V LLDGEV <u>TEASIL</u> RKLDDLERIARRNG QAI <u>GVA</u> SAFD	35	Yes
9	2OZT	GQTTFK <u>WKVG</u> VMSP <u>EEE</u> QAILK ALA <u>ALPPGAKL</u> RLDANGSWD	34	Yes
10	1VEM	GIVLN <u>GENALS</u> IGNE <u>EEYK</u> RVAEMAF NYNFAG <u>FTLLRY</u>	34	Yes
11	3ELF	AKPF <u>D</u> FVFHGGSGSL <u>KSEIEEAL</u> RYG VVK <u>MNV</u> DTD	33	Yes
12	4GVF	HPLV <u>GG</u> L <u>LFTR</u> NYHD <u>PEQL</u> RELVRQ IR <u>AASRNH</u> L <u>VV</u> AVD QEGGRV	33	Yes
13	1SFS	YPKFW <u>GR</u> T lsevpnv <u>SEGL</u> TRDEIVR IRNYGV <u>KVL</u> LPI YNAAF	31	Yes
14	2KI0 ^b	GSGQV <u>RTI</u> WVG <u>G</u> TPEELKK LKE <u>AKKAN</u> IR <u>VT</u> F WGD	38	No

Sequences 1–13 are identified from TIM barrel proteins. Sequence 14 (2KI0) is the designed sequence. The strand regions of the $\beta\alpha\beta$ units are shown in bold and italics, while the α helices are shown as bold and underlined. The description on the long-range side-chain to main-chain hydrogen bond ($\beta\alpha$ clamp) interactions is given in the text

^aLong-range side-chain-main-chain hydrogen bond ($\beta\alpha$ clamp) interactions (see text/Fig. 6)

^bDesigned $\beta\alpha\beta$ sequence (control sequence)

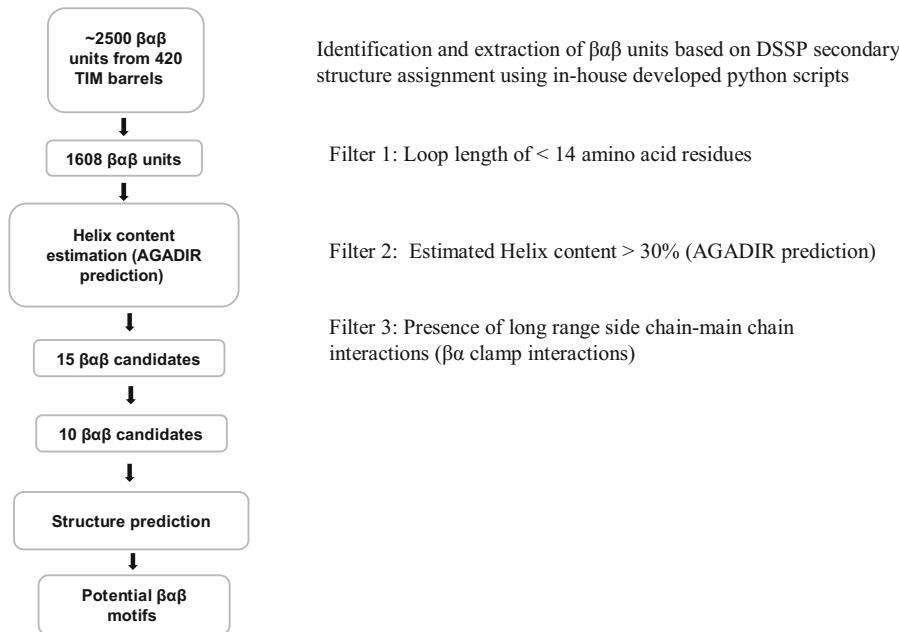


Fig. 5 Schematic representation of the methodology adopted for short-listing potential $\beta\alpha\beta$ sequences from TIM barrel proteins. Helix content estimates were obtained by submitting the appropriate sequences to the server hosting AGADIR

9 (4GVF), display good probability to adopt $\beta\alpha\beta$ conformations (Table 2 and Fig. 6a). It should be noted that for the negative control sequences, 11 (3E96), 12 (3LYF), and 13 (3M0Z), which show a poor estimated helix propensity (0, 5, and 10%), show poor chances of attaining $\beta\alpha\beta$ fold. The observation that the designed $\beta\alpha\beta$ sequence, 14, (2KI0) shows excellent ability to attain $\beta\alpha\beta$ conformations is noteworthy. Similar results were predicted from PEPFOLD3. The sequences, particularly 5 (2OZT) and 8 (3ELF) along with the designed $\beta\alpha\beta$ sequence 15 (2KI0), show a strong folding tendency (Table 2 and Fig. 6b). In summary, at least a few short-listed $\beta\alpha\beta$ sequences are predicted to possess a high propensity to attain $\beta\alpha\beta$ conformation.

5 Conclusion and Outlook

The overview of sequence and structural features of the individual $\beta\alpha\beta$ units from TIM barrels captured the plausible role of short loops and some specific non-covalent interactions such as long-range hydrogen-bonding interactions and hydrophobic clustering of nonpolar residues between the strand and helix. The sequence and structural features gleaned from the individual $\beta\alpha\beta$ units occurring natural barrels are more likely to reflect the evolutionary optimization for folding and stability. Combining this with

Table 2**Summary of the structure prediction results for the short-listed βαβ candidates**

Sequence number	Sequence source (PDB ID)	βαβ amino acid sequence	Predicted structures		
			Helix propensity (AGADIR prediction, %)	QUARK (out of 10)	PEPFOLD3 (out of 5)
1	1VD6	PQAVFNVELKSFPGLGEEAA RRLAALLRGREGVVVSSFDP	53	0	0
2	3BOF	GRSLFNSAK VDEEELEMKINLLKKYGG TLIVLLMG	38	4	0
3	3HJE	EVDGYRIDHIDGLFKPEEYL RRLKNKIGNKHIFVEKI	38	5	2
4	2XFR	NYVQVYVMLPLDAVSVNN RFEKGDELRAQLRKLVAG VDGVMVDVWGL	38	0	1
5	2QV5	FADVLLDGEVTEASIL RKLDLDERIARRNGQAIGVA SAFD	35	1	0
6	2OZT	GQTTFWKVGVMSPEEE QAILKALLAALPPGAKL RLDANGSWD	34	10	5
7	1VEM	GIVLNGENALSIGNEEYK RVAEMAFNYNFAGFTLLRY	34	2	0
8	3ELF	AKPFDFVFHGGSGSLK SEIEEALRYGVVKMNVDTD	33	7	4
9	1SFS	YPKFWGRYLSEVPNVSEGL TRDEIVRIRNYGVKVLLPI YNAAF	31	0	0
10	3E96	HQIAWICGTAEKWAPFF WHAGAKGFTSLV	0	0	0
11	4GVF	FDGVIFSDDLSMEGAAIMG SYAERAQASLDAGCDMIL VCNN	2	0	0
12	3LYF	GADVGLLEGFRSKEQAAAA VAALAPWPLLNSVENG	5	0	0
13	3M0Z	GGSSIKYFPGLKHRAEFEA VAKACAAHDFWLEPTGG	10	0	0
14	2KI0 ^b	GSGQVRTIWVGG TPEELKKLKEEAKKANIRVTW WGD	38	10	3

The number of predicted βαβ conformation for the given sequences from QUARK (ab initio) and PEPFOLD3 (Hidden Markov) approaches are indicated. The number denotes out of 10 predicted structures from QUARK, and out of 5 predicted structures for PEPFOLD3. The sequences, 11–13 were included as negative controls (sequences with very poor helix forming tendency). ^b The designed βαβ sequence, Sequence 14 served as a control sequence.

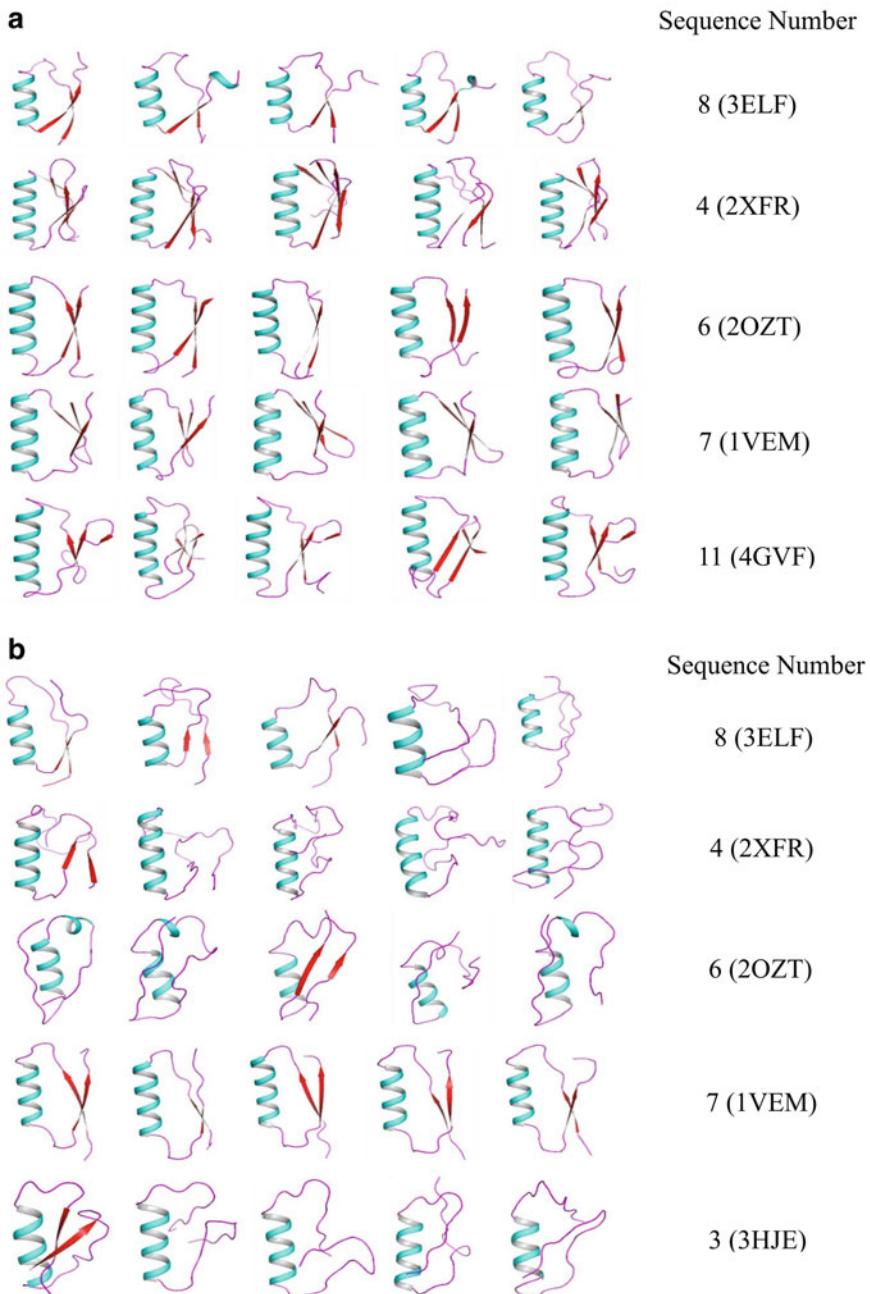


Fig. 6 Representative predicted conformations for the sequences from (a) QUARK and (b) PEPFOLD3. β strands are indicated in red and α helices in cyan

benchmarked parameters for a designed $\beta\alpha\beta$ sequence capable of folding in isolation, a search for naturally occurring $\beta\alpha\beta$ sequences, optimized for autonomous folding, leads to few potential candidates. The propensity for self-folding when further explored by

assessing their structure forming probability indicated that a set of $\beta\alpha\beta$ sequences possess the potential for self-folding. Experimental verification for self-folding will provide insights for further optimizing of their sequences for folding and stability. Conservative changes with no concomitant effect on folding and stability can lead to the repository of $\beta\alpha\beta$ sequences that fold independently. Thus, the proposed approach can pave way for the development of algorithms that can predict not only independently folding $\beta\alpha\beta$ motifs but also creation of novel TIM barrels.

Acknowledgments

The authors thank University Grants Commission, India, for the initial part of this research. RVK is grateful to UGC, India, and Birla Institute of Technology and Science Pilani, Hyderabad Campus, for financial support in the form of research fellowship.

References

1. Grishin NV (2001) Fold change in evolution of protein structures. *J Struct Biol* 134 (2–3):167–185. <https://doi.org/10.1006/jsb.2001.4335>
2. Soding J, Lupas AN (2003) More than the sum of their parts: on the evolution of proteins from peptides. *Bio Essays* 25(9):837–846. <https://doi.org/10.1002/bies.10321>
3. Salem GM, Hutchinson EG, Orengo CA, Thornton JM (1999) Correlation of observed fold frequency with the occurrence of local structural motifs. *J Mol Biol* 287(5):969–981. <https://doi.org/10.1006/jmbi.1999.2642>
4. Cui Y, Wong WH, Bornberg-Bauer E, Chan HS (2002) Recombinatoric exploration of novel folded structures: a heteropolymer-based model of protein evolutionary landscapes. *Proc Natl Acad Sci U S A* 99 (2):809–814. <https://doi.org/10.1073/pnas.022240299>
5. Bogard LD, Deem MW (1999) A hierarchical approach to protein molecular evolution. *Proc Natl Acad Sci U S A* 96(6):2591–2595
6. Marcotte EM, Pellegrini M, Yeates TO, Eisenberg D (1999) A census of protein repeats. *J Mol Biol* 293(1):151–160. <https://doi.org/10.1006/jmbi.1999.3136>
7. Doolittle RF (1995) The multiplicity of domains in proteins. *Annu Rev Biochem* 64:287–314. <https://doi.org/10.1146/annurev.bi.64.070195.001443>
8. 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 (2–3):191–203. <https://doi.org/10.1006/jsb.2001.4393>
9. Orengo CA, Thornton JM (2005) Protein families and their evolution—a structural perspective. *Annu Rev Biochem* 74:867–900. <https://doi.org/10.1146/annurev.biochem.74.082803.133029>
10. Soding J, Biegert A, Lupas AN (2005) The HHpred interactive server for protein homology detection and structure prediction. *Nucleic Acids Res* 33(Web Server):W244–W248. <https://doi.org/10.1093/nar/gki408>
11. Riechmann L, Winter G (2000) Novel folded protein domains generated by combinatorial shuffling of polypeptide segments. *Proc Natl Acad Sci U S A* 97(18):10068–10073. <https://doi.org/10.1073/pnas.170145497>
12. Riechmann L, Winter G (2006) Early protein evolution: building domains from ligand-binding polypeptide segments. *J Mol Biol* 363 (2):460–468. <https://doi.org/10.1016/j.jmb.2006.08.031>
13. Panchenko AR, Luthey-Schulten Z, Cole R, Wolynes PG (1997) The foldon universe: a survey of structural similarity and self-recognition of independently folding units. *J Mol Biol* 272(1):95–105. <https://doi.org/10.1006/jmbi.1997.1205>
14. Berezovsky IN, Guarnera E, Zheng Z (2017) Basic units of protein structure, folding, and

- function. *Prog Biophys Mol Biol* 128:85–99. <https://doi.org/10.1016/j.pbiomolbio.2016.09.009>
15. Zeng J, Jiang F, Wu YD (2016) Folding simulations of an alpha-helical hairpin motif alphatalpha with residue-specific force fields. *J Phys Chem B* 120(1):33–41. <https://doi.org/10.1021/acs.jpcb.5b09027>
 16. 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(9):584–590
 17. Searle MS, Williams DH, Packman LC (1995) A short linear peptide derived from the N-terminal sequence of ubiquitin folds into a water-stable non-native beta-hairpin. *Nat Struct Biol* 2(11):999–1006
 18. Stanger HE, Syud FA, Espinosa JF, Giriat I, Muir T, Gellman SH (2001) Length-dependent stability and strand length limits in antiparallel beta-sheet secondary structure. *Proc Natl Acad Sci U S A* 98(21):12015–12020. <https://doi.org/10.1073/pnas.211536998>
 19. Cochran AG, Skelton NJ, Starovasnik MA (2001) Tryptophan zippers: stable, monomeric beta-hairpins. *Proc Natl Acad Sci U S A* 98(10):5578–5583. <https://doi.org/10.1073/pnas.091100898>
 20. Sadqi M, de Alba E, Perez-Jimenez R, Sanchez-Ruiz JM, Munoz V (2009) A designed protein as experimental model of primordial folding. *Proc Natl Acad Sci U S A* 106(11):4127–4132. <https://doi.org/10.1073/pnas.0812108106>
 21. Religa TL, Johnson CM, Vu DM, Brewer SH, Dyer RB, Fersht AR (2007) The helix-turn-helix motif as an ultrafast independently folding domain: the pathway of folding of Engrailed homeodomain. *Proc Natl Acad Sci U S A* 104(22):9272–9277. <https://doi.org/10.1073/pnas.0703434104>
 22. Liang H, Chen H, Fan K, Wei P, Guo X, Jin C, Zeng C, Tang C, Lai L (2009) De novo design of a beta alpha beta motif. *Angew Chem* 48(18):3301–3303. <https://doi.org/10.1002/anie.200805476>
 23. Marqusee S, Robbins VH, Baldwin RL (1989) Unusually stable helix formation in short alanine-based peptides. *Proc Natl Acad Sci U S A* 86(14):5286–5290
 24. Ihlalainen JA, Paoli B, Muff S, Backus EH, Bredenbeck J, Woolley GA, Cafisch A, Hamm P (2008) Alpha-Helix folding in the presence of structural constraints. *Proc Natl Acad Sci U S A* 105(28):9588–9593. <https://doi.org/10.1073/pnas.0712099105>
 25. Petukhov M, Tatsu Y, Tamaki K, Murase S, Uekawa H, Yoshikawa S, Serrano L, Yumoto N (2009) Design of stable alpha-helices using global sequence optimization. *J Pept Sci* 15(5):359–365. <https://doi.org/10.1002/psc.1122>
 26. Yakimov A, Rychkov G, Petukhov M (2014) De novo design of stable alpha-helices. *Methods Mol Biol* 1216:1–14. https://doi.org/10.1007/978-1-4939-1486-9_1
 27. Ramakrishna V, Sasidhar YU (1997) A pentapeptide model for an early folding step in the refolding of staphylococcal nuclease: the role of its turn propensity. *Biopolymers* 41(2):181–191. [https://doi.org/10.1002/\(SICI\)1097-0282\(199702\)41:2<181::AID-BIP5>3.0.CO;2-P](https://doi.org/10.1002/(SICI)1097-0282(199702)41:2<181::AID-BIP5>3.0.CO;2-P)
 28. Baker EG, Bartlett GJ, Porter Goff KL, Woolfson DN (2017) Miniprotein design: past, present, and prospects. *Acc Chem Res* 50(9):2085–2092. <https://doi.org/10.1021/acs.accounts.7b00186>
 29. Kister AE, Potapov V (2013) Amino acid distribution rules predict protein fold. *Biochem Soc Trans* 41(2):616–619. <https://doi.org/10.1042/BST20120308>
 30. Struthers MD, Cheng RP, Imperiali B (1996) Design of a monomeric 23-residue polypeptide with defined tertiary structure. *Science* 271(5247):342–345
 31. Dahiyat BI, Mayo SL (1997) De novo protein design: fully automated sequence selection. *Science* 278(5335):82–87
 32. Nagano N, Orengo CA, Thornton JM (2002) One fold with many functions: the evolutionary relationships between TIM barrel families based on their sequences, structures and functions. *J Mol Biol* 321(5):741–765
 33. Yang X, Kathuria SV, Vadrevu R, Matthews CR (2009) Betaalpha-hairpin clamps brace betalpha-phabeta modules and can make substantive contributions to the stability of TIM barrel proteins. *PLoS One* 4(9):e7179. <https://doi.org/10.1371/journal.pone.0007179>
 34. Zitzewitz JA, Gualfetti PJ, Perkins IA, Wasta SA, Matthews CR (1999) Identifying the structural boundaries of independent folding domains in the alpha subunit of tryptophan synthase, a beta/alpha barrel protein. *Protein Sci* 8(6):1200–1209. <https://doi.org/10.1101/ps.8.6.1200>
 35. Frenkel ZM, Trifonov EN (2005) Closed loops of TIM barrel protein fold. *J Biomol Struct Dyn* 22(6):643–656. <https://doi.org/10.1080/07391102.2005.10507032>
 36. Huang PS, Feldmeier K, Parmeggiani F, Fernandez Velasco DA, Hocker B, Baker D (2016)

- De novo design of a four-fold symmetric TIM-barrel protein with atomic-level accuracy. *Nat Chem Biol* 12(1):29–34. <https://doi.org/10.1038/nchembio.1966>
37. Koga N, Tatsumi-Koga R, Liu G, Xiao R, Acton TB, Montelione GT, Baker D (2012) Principles for designing ideal protein structures. *Nature* 491(7423):222–227. <https://doi.org/10.1038/nature11600>
38. Ochoa-Leyva A, Montero-Moran G, Saab-Rincon G, Brieba LG, Soberon X (2013) Alternative splice variants in TIM barrel proteins from human genome correlate with the structural and evolutionary modularity of this versatile protein fold. *PLoS One* 8(8):e70582. <https://doi.org/10.1371/journal.pone.0070582>
39. Ochoa-Leyva A, Barona-Gomez F, Saab-Rincon G, Verdel-Aranda K, Sanchez F, Soberon X (2011) Exploring the structure-function loop adaptability of a (beta/alpha) (8)-barrel enzyme through loop swapping and hinge variability. *J Mol Biol* 411(1):143–157. <https://doi.org/10.1016/j.jmb.2011.05.027>
40. Ochoa-Leyva A, Soberon X, Sanchez F, Arguello M, Montero-Moran G, Saab-Rincon G (2009) Protein design through systematic catalytic loop exchange in the (beta/alpha) eight fold. *J Mol Biol* 387(4):949–964. <https://doi.org/10.1016/j.jmb.2009.02.022>
41. Nagarajan D, Deka G, Rao M (2015) Design of symmetric TIM barrel proteins from first principles. *BMC Biochem* 16:18. <https://doi.org/10.1186/s12858-015-0047-4>
42. Berezovsky IN, Grosberg AY, Trifonov EN (2000) Closed loops of nearly standard size: common basic element of protein structure. *FEBS Lett* 466(2–3):283–286
43. Chintapalli SV, Yew BK, Illingworth CJ, Upton GJ, Reeves PJ, Parkes KE, Snell CR, Reynolds CA (2010) Closed loop folding units from structural alignments: experimental foldons revisited. *J Comput Chem* 31(15):2689–2701. <https://doi.org/10.1002/jcc.21562>
44. Kadumuri RV, Vadrevu R (2017) Diversity in alphabeta and betaalpha loop connections in TIM barrel proteins: implications for stability and design of the fold. *Interdiscip Sci Comput Life Sci.* <https://doi.org/10.1007/s12539-017-0250-7>
45. Yang X, Vadrevu R, Wu Y, Matthews CR (2007) Long-range side-chain-main-chain interactions play crucial roles in stabilizing the (betaalpha)8 barrel motif of the alpha subunit of tryptophan synthase. *Protein Sci* 16(7):1398–1409. <https://doi.org/10.1110/ps.062704507>
46. Balasco N, Esposito L, De Simone A, Vitaliano L (2013) Role of loops connecting secondary structure elements in the stabilization of proteins isolated from thermophilic organisms. *Protein Sci* 22(7):1016–1023. <https://doi.org/10.1002/pro.2279>
47. Munoz V, Serrano L (1995) Elucidating the folding problem of helical peptides using empirical parameters. III. Temperature and pH dependence. *J Mol Biol* 245(3):297–308. <https://doi.org/10.1006/jmbi.1994.0024>
48. Munoz V, Serrano L (1995) Elucidating the folding problem of helical peptides using empirical parameters. II. Helix macrodipole effects and rational modification of the helical content of natural peptides. *J Mol Biol* 245(3):275–296
49. Munoz V, Serrano L (1997) Development of the multiple sequence approximation within the AGADIR model of alpha-helix formation: comparison with Zimm-Bragg and Lifson-Roig formalisms. *Biopolymers* 41(5):495–509. [https://doi.org/10.1002/\(SICI\)1097-0282\(19970415\)41:5<495::AID-BIP2>3.0.CO;2-H](https://doi.org/10.1002/(SICI)1097-0282(19970415)41:5<495::AID-BIP2>3.0.CO;2-H)
50. Lacroix E, Viguera AR, Serrano L (1998) Elucidating the folding problem of alpha-helices: local motifs, long-range electrostatics, ionic-strength dependence and prediction of NMR parameters. *J Mol Biol* 284(1):173–191. <https://doi.org/10.1006/jmbi.1998.2145>
51. Streicher WW, Makhatadze GI (2006) Calorimetric evidence for a two-state unfolding of the beta-hairpin peptide trpzip4. *J Am Chem Soc* 128(1):30–31. <https://doi.org/10.1021/ja056392x>
52. Xu D, Zhang Y (2012) Ab initio protein structure assembly using continuous structure fragments and optimized knowledge-based force field. *Proteins* 80(7):1715–1735. <https://doi.org/10.1002/prot.24065>
53. Lamiable A, Thevenet P, Rey J, Vavrusa M, Derreumaux P, Tuffery P (2016) PEP-FOLD3: faster de novo structure prediction for linear peptides in solution and in complex. *Nucleic Acids Res* 44(W1):W449–W454. <https://doi.org/10.1093/nar/gkw329>
54. Shen Y, Maupetit J, Derreumaux P, Tuffery P (2014) Improved PEP-FOLD approach for peptide and miniprotein structure prediction. *J Chem Theory Comput* 10(10):4745–4758. <https://doi.org/10.1021/ct500592m>
55. Thevenet P, Shen Y, Maupetit J, Guyon F, Derreumaux P, Tuffery P (2012) PEP-FOLD: an updated de novo structure prediction server for both linear and disulfide bonded cyclic peptides. *Nucleic Acids Res* 40. (Web Server issue:

- W288–W293. <https://doi.org/10.1093/nar/gks419>
- 56. Zhang Y, Skolnick J (2004) SPICKER: a clustering approach to identify near-native protein folds. *J Comput Chem* 25(6):865–871. <https://doi.org/10.1002/jcc.20011>
 - 57. Zhang Y, Skolnick J (2004) Scoring function for automated assessment of protein structure template quality. *Proteins* 57(4):702–710. <https://doi.org/10.1002/prot.20264>
 - 58. Altschul SF, Madden TL, Schaffer AA, Zhang J, Zhang Z, Miller W, Lipman DJ (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res* 25(17):3389–3402
 - 59. Sippl MJ (1990) Calculation of conformational ensembles from potentials of mean force. An approach to the knowledge-based prediction of local structures in globular proteins. *J Mol Biol* 213(4):859–883
 - 60. Schrodinger L (2015) The PyMOL Molecular Graphics System, Version 1.8
 - 61. Crooks GE, Hon G, Chandonia JM, Brenner SE (2004) WebLogo: a sequence logo generator. *Genome Res* 14(6):1188–1190. <https://doi.org/10.1101/gr.849004>



Chapter 12

Formation of Cross-Beta Supersecondary Structure by Soft-Amyloid Cores: Strategies for Their Prediction and Characterization

M. Rosario Fernández, Irantzu Pallarès, Valentín Iglesias, Jaime Santos, and Salvador Ventura

Abstract

Proteins with prion-like behavior are attracting an increasing interest, since accumulating evidences indicate that they play relevant roles both in health and disease. The self-assembly of these proteins into insoluble aggregates is associated with severe neuropathological processes such as amyotrophic lateral sclerosis (ALS). However, in normal conditions, they are known to accomplish a wide range of functional roles. The conformational duality of prion-like proteins is often encoded in specific protein regions, named prion-like domains (PrLDs). PrLDs are usually long and disordered regions of low complexity. We have shown that PrLDs might contain soft-amino cores that contribute significantly to trigger their aggregation, as well as to support their propagation. Further exploration of the role of these sequences in the conformational conversion of prion-like proteins might provide novel insights into the mechanism of action and regulation of these polypeptides, enabling the future development of therapeutic strategies. Here, we describe a set of methodologies aimed to identify and characterize these short amyloid stretches in a protein or proteome of interest, ranging from *in silico* detection to *in vitro* and *in vivo* evaluation and validation.

Key words Protein aggregation, Amyloid, Soft-amino core, Fibril, Cross-beta-sheet, Bioinformatics, Prion-like

1 Introduction

It is well established that protein aggregation into amyloid-like structures is associated with a broad range of human pathologies such as Alzheimer's disease, Parkinson disease, or type II diabetes. Moreover, the number of proteins showing amyloid behavior is increasing year after year [1]. Amyloids are insoluble fibrillar assemblies characterized by a distinctive cross- β diffraction pattern when analyzed by X-ray diffraction and affinity for dyes like Congo red [2]. Gaining insight into the basis of amyloid formation has become of high interest since a better knowledge may provide new therapeutic avenues for amyloid diseases. It is widely accepted that the

assembly of amyloid proteins into the cross- β structure appears to obey the so-called amyloid-short-stretch hypothesis, in which intermolecular contacts between residues in short regions of the protein are responsible for nucleating the protein aggregation [3]. Prions constitute a subclass of amyloids which are able to switch between the soluble and the aggregated state and bear an infective ability. Although prions have been traditionally associated to neuropathology in mammals, yeast prions confer selective advantages under certain circumstances [4–7]. The number of putative prion-like proteins vary widely between species, from above 20% from *Dictyostelium discoideum* to less than 1% in the case of virus [8]. However, the fact that they have been predicted to exist in all kingdoms of life suggests that prion-like conformational conversion might be implied in evolutionary conserved functions. Recently we have suggested that the aggregation of yeast prions, and prion-like proteins in general, is mechanistically similar to that of classical amyloidogenic proteins, with soft-amyloid cores embedded in long and disordered prion domains (PrDs) acting as nucleation elements for the self-assembly reaction. The main differences between the soft-amyloid cores in PrDs and the classical amyloid cores of pathogenic proteins are that the former exhibits a biased amino acid composition, in which Q/N residues are enriched and are significantly longer, in such a way that the aggregation potential is weaker and less concentrated, allowing the corresponding proteins to remain soluble, unless aggregation is triggered by extrinsic factors [9]. This hypothesis has been recently experimentally confirmed for well-characterized yeast prions [10, 11], for the Rho terminator factor from the pathogenic bacteria *Clostridium botulinum* [12, 13], as well as for several human prion-like proteins [9]. In this chapter, we describe the methodology pipeline used for the prediction and analysis of putative soft-amyloid cores in PrDs and prion-like domains (PrLDs). This includes the identification of putative PrLDs and their associated soft-amyloid cores by computational methods followed by in vitro and in vivo approaches aimed to their structural characterization and functional validation (Fig. 1).

2 Materials

2.1 Computational Identification of PrLD and Soft-Amyloid Cores in Proteomes

For PrLD and soft-amyloid cores identification, PAPA is written in Python and uses Python 2.7 as the interpreter. PLAAC runs on JAVA. pWALTZ executable requires Linux.

2.2 Detection of Aggregates

1. Low-binding syringe filters with 0.45 μm (Millipore, Merck KGaA, Darmstadt, Germany).

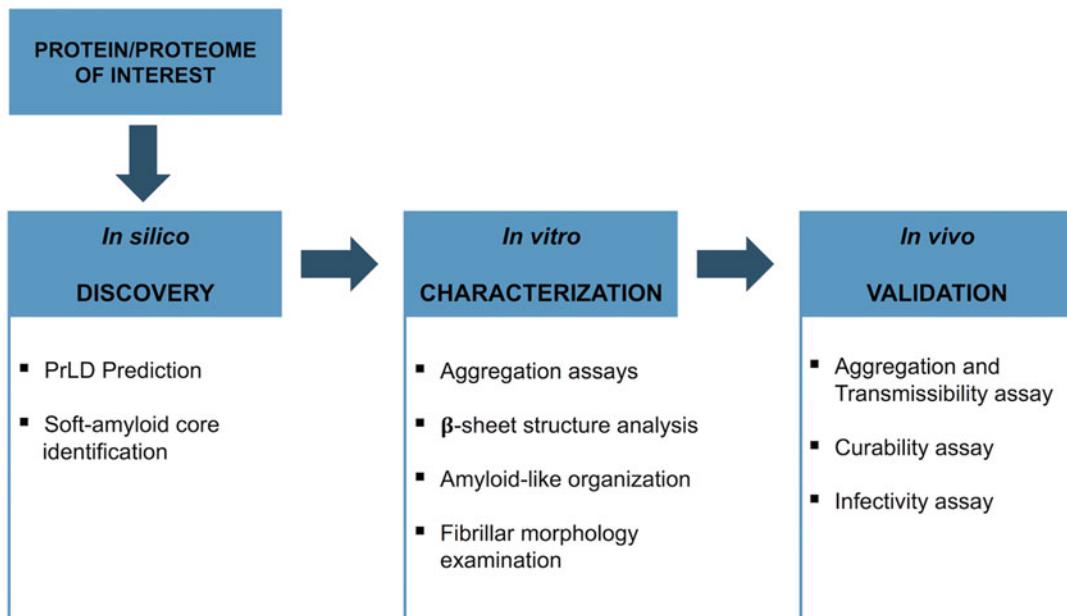


Fig. 1 Methodology overview: stages of the soft-amyloid core prediction and characterization pipeline

2. Cary 100 or 400 UV/Vis spectrophotometer (Varian, Palo Alto, CA, USA).
 3. Cary Eclipse spectrofluorometer (Varian, Palo Alto, CA, USA).
 4. GraphPad Prism 5, 6, or 7 (GraphPad Software Inc., La Jolla, CA, USA).
- 2.3 Bis-ANS Relative Fluorescence Determination**
1. Bis-ANS (Sigma Aldrich, Merck KGaA, Darmstadt, Germany). The required amount of Bis-ANS is diluted in Milli-Q water to obtain a stock solution of 200 μ M.
 2. Cary Eclipse spectrofluorometer (Varian, Palo Alto, CA, USA).
- 2.4 Amyloid Dyes**
1. Congo red (CR) (Sigma Aldrich, Merck KGaA, Darmstadt, Germany). The required amount of CR is diluted in Milli-Q water to obtain a stock solution of 200 μ M. CR stock solution is filtered through nucleopore polycarbonate membranes 0.4 μ M pore size for removing dye aggregates.
 2. Thioflavin-T (Sigma Aldrich, Merck KGaA, Darmstadt, Germany). The required amount of Th-T is diluted in Milli-Q water to obtain a stock solution of 250 μ M.
 3. Centrifuge Eppendorf 5418 (Eppendorf, Hamburg, Germany).
 4. Cary 100 UV/Vis spectrophotometer (Varian, Palo Alto, CA, USA).
 5. Cary Eclipse spectrofluorometer (Varian, Palo Alto, CA, USA).

6. Optic microscope with cross-polarized light (Leica Microsystems, Wetzlar, Germany).
7. Leica fluorescence DMRB microscope with a narrow green band filter (Leica Microsystems, Wetzlar, Germany).

2.5 Secondary Structure Analysis

1. Jasco-810 spectropolarimeter (Jasco International Co. Ltd., Tokyo, Japan).
2. Bruker Tensor 27 FT-IR Spectrometer (Bruker Optics Inc., Karlsruhe, Germany) with a Golden Gate MKII ATR accessory.
3. OPUS MIR Tensor 27 software (OPUS Optics User Software, Bruker Optics Inc., Karlsruhe, Germany).
4. The PeakFit package (Systat Software, San Jose, CA, USA).

2.6 Aggregation Kinetics

1. Congo red (CR) (Sigma Aldrich, Merck KGaA, Darmstadt, Germany), prepared as described in Subheading [2.4](#).
2. Thioflavin-T (Sigma Aldrich, Merck KGaA, Darmstadt, Germany), prepared as described in Subheading [2.4](#).
3. Cary Eclipse spectrofluorometer (Varian, Palo Alto, CA, USA).
4. Jasco-810 spectropolarimeter (Jasco International Co. Ltd., Tokyo, Japan).
5. Cary 100 or 400 UV/Vis spectrophotometer (Varian, Palo Alto, CA, USA).
6. GraphPad Prism 5, 6, or 7 (GraphPad Software Inc., La Jolla, CA, USA).

2.7 TEM

1. Uranyl acetate (Sigma Aldrich, Merck KGaA, Darmstadt, Germany).
2. JEOL JEM-1400 electron microscope (JEOL Peabody, MA, USA), operated at 80 kV accelerating voltage.

2.8 Functional Substitution of *Sup35p* PrD by Soft-Amyloid Cores

All mediums and buffers are sterilized by autoclaving unless indicated.

1. Yeast nitrogen base, bacto-peptone, bacto-agar, and bacto-yeast extract are from Becton, Dickinson. The rest of reagents are from Sigma-Aldrich unless specifically indicated.
2. A *SUP35* expression vector with *URA3* marker gene. The vector pUKC1620 from Tuite laboratory is one possibility [[14](#)]. Replace the N-terminal of *SUP35* domain with the soft-amyloid core.
3. A yeast expression vector with *TRP1* selection marker and strong promoter like by example p424 GPD (ATCC® 87357™). The Core-GFP can be introduced by the classic cloning methods.

4. *Saccharomyces cerevisiae* strain LJ14 derived of the strain 74D-694 (*MATa ade1-14UGA trp1-289 his3Δ-200 ura3-52 leu2-3,112 sup35::loxP* [pYK810]), or similar [14].
5. YNB: 0.21% yeast nitrogen base (without amino acids and ammonium sulfate), 0.63% ammonium sulfate.
6. Agar: 3.2% bacto-agar in water.
7. 10× glucose: 20% glucose.
8. 10× drop-out mix: 0.04% adenine hemisulfate, 0.04% arginine HCl, 0.04% histidine HCl, 0.04% isoleucine, 0.04% leucine, 0.04% lysine HCl, 0.04% methionine, 0.04% tyrosine, 0.06% phenylalanine, 0.06% tryptophan, 0.2% serine, 0.2% threonine, 0.025% uracil, 0.2% valine, 0.1% aspartic, and 0.1% glutamic acid. For each specific drop-out mix, the indicated amino acid or nucleotide is excluded. By example, in drop-out -His histidine is excluded.
9. SD-X (X, amino acid or nucleotide) plates: mix 320 mL agar (around 80 °C) with 80 mL YNB, 50 mL of the corresponding 10× drop-out mix (by example drop-out -His for SD-His), and 50 mL glucose. Pour into Petri dishes and let cool down.
10. 1/4 YPD plates: 0.25% bacto-yeast extract, 2% bacto-peptone, 2% bacto-agar, and 2% glucose. It is recommended to autoclave glucose in water separately and mix with the rest of the mixture just before pouring into the Petri dishes.
11. SD-His 5-FOA: prepare SD-His as indicated previously, and add 0.1% 5-fluoroorotic acid (filter sterilized by using 0.2 µm filter) before pouring into the plates.
12. PBS: 10 mM Na₂HPO₄, 1.8 mM KH₂PO₄, pH 7.4, 137 mM NaCl, 2.7 mM KCl.
13. InnovaR 43 Incubator shaker (Eppendorf, Inc., USA).
14. Oven at 30 °C (by example Memmert IPP30, GmbH, Germany).
15. Confocal microscope Leica TCS SP5 or Fluorescent microscope Leica DMRB (Leica Microsystems AG, Germany).
16. Microscope slides and coverslips.

2.9 [PSI^t] Curing

1. YPD plates: 1% bacto-yeast extract, 2% bacto-peptone, 2% bacto-agar, and 2% glucose. Autoclave glucose separately, and add to the mixture before pouring into the plates.
2. 5 mM GdnHCl YPD plates: prepare YPD as previously described, let it cool down to 60–70 °C, and add 5 mM guanidine hydrochloride (GdnHCl) filter sterilized with a 0.2 µm filter before pouring into Petri dishes.
3. Oven (by example Memmert IPP30, GmbH, Germany).

2.10 In Vivo Seeding with Aggregated Protein or Peptides

1. YPD prepared as described in Subheading 2.9.
2. 1 M sorbitol: prepare 1 M sorbitol in Milli-Q water and autoclave.
3. SCEM: 1 M sorbitol, 0.1 M sodium citrate (pH 5.8), 10 mM EDTA, and 10 mM DTT (DTT should be added each time from a 1 M stock solution stored at -20 °C).
4. 5% SDS: dissolve 0.5 g of SDS in 10 mL of water.
5. Lyticase: from *Arthrobacter luteus* (Sigma-Aldrich), resuspend the lyophilized powder in 20 mM phosphate, pH 7.5, and 50% glycerol at 10,000 U/mL.
6. Salmon sperm DNA: dissolve 400 mg of salmon sperm DNA (Sigma) in water to 1 mg/mL. Incubate at 65 °C overnight and sonicate for 1 min. Repeat the sonication 3-5 times until the solution is less viscous. Incubate again at 95 °C for 10 min and repeat sonication step. Let it cool down and dispense into 1 mL aliquots. Keep at -20 °C.
7. Top-agar: 1 M sorbitol, 2.5% bacto-agar.
8. 10× YNB: 2.1% yeast nitrogen base (without amino acids and ammonium sulfate), 6.3% ammonium sulfate.
9. 10× drop-out -Ura -His: prepare dropout as indicated in Subheading 2.8 but excluding uracil and histidine.
10. Glucose: prepare as indicated in Subheading 2.8.
11. STC: 1 M sorbitol, 30 mM CaCl₂, 10 mM Tris-HCl pH 7.5.
12. 20% PEG solution: 20% [w/v] PEG 8000, 30 mM CaCl₂, 10 mM Tris pH 7.5.
13. SOS: 1 M sorbitol, 7 mM CaCl₂, 0.25% yeast extract, 0.5% bacto-peptone.
14. SD-Ura-His: prepare as indicated in Subheading 2.8.
15. InnovaR 43 Incubator Shaker (Eppendorf, Inc., USA).
16. Allegra 6R centrifuge (Beckman Coulter-Life Sciences, USA).
17. VWR® ultrasonic cleaner (VWR®) and B Braun Labsonic®U (Braun Biotech International, GmbH, Germany).
18. Oven at 30 °C (by example Memmert IPP30, GmbH, Germany).

3 Methods

3.1 Computational Identification of PrLD and Soft-Amyloid Cores in Proteins and Proteomes

Yeast prions research uncovered that these proteins share common features, including the presence of disordered low complexity domains, enriched in Q/N residues, and containing cryptic soft-amyloid cores. These computationally addressable signatures ignited the development of different tools to discover proteins

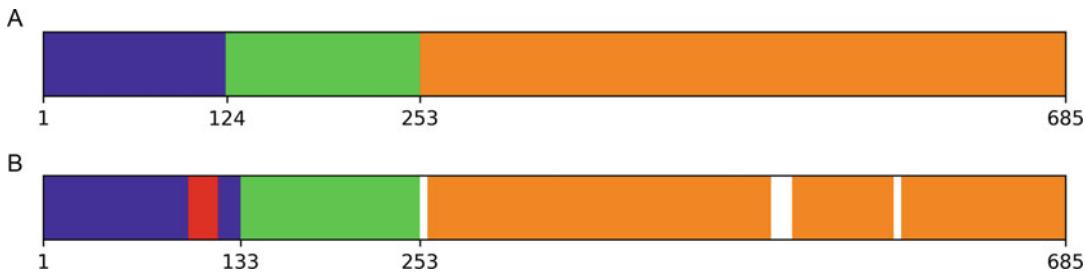


Fig. 2 Sup35p prion domain layout as observed or predicted. **(a)** The Sup35p protein division on the basis of its Met residues; the PrD N-domain from Met1 in blue, the highly charged M-domain from Met124 in green, and the functional translation termination C-domain starting with Met254 in orange [58]. **(b)** Domain distribution as predicted by state-of-the-art algorithms; PLAAC PrD in blue, pWALTZ soft-amyloid core in red, disorder prediction by IUPRED in green [59], and functional Pfam domains in orange [60]

with similar behavior in other organisms (Fig. 2). Large-scale screenings identified prion-like proteins in different organisms across all taxonomic divisions [12, 13, 15, 16], despite the load of prion-like proteins differed significantly between species [8, 17]. Below we briefly describe the different algorithms researchers might want to use to identify a PrLD and the corresponding soft-amyloid core in a given protein, in case they do not know its identity in advance.

3.1.1 Prion Aggregation Prediction Algorithm (PAPA)

PAPA is a composition-based algorithm based on the experimental characterization of sequential variants of Sup35p yeast prion [18]. It applies as a first step FoldIndex [19] as a disorder predictor. The composition of stretches predicted to be disordered will be compared to residue frequencies in prion-forming mutants of Sup35p using a 41 amino acid residues sliding window. The best scoring stretch is presented. The user should input a FASTA file with the desired sequences to be evaluated (*see Notes 1 and 2*). The algorithm will return the sequence name, the best score per sequence, and its position. Those sequences scoring 0.05 or above are predicted to have prion potential, while the given PrLD position corresponds to the center of the first positive window.

3.1.2 Prion-Like Amino Acid Composition (PLAAC)

PLAAC is another algorithm which detects prion domains based on compositional similarities to yeast PrD [20]. Lindquist's lab implemented a hidden Markov model (HMM) based on three known yeast prions [21]. This approach was largely improved by extending the HMM to 28 yeast proteins that experimentally displayed switching behavior or amyloid formation. The program uses the HMM-derived residue log-likelihood for every position and calculates the probability of each amino acid to be part of a PrLD, having into account the amino acid background frequencies in a given proteome. The sum of the values is computed, and any protein with a COREscore >0 is considered a putative PrLD (*see Notes*

2 and **3**). PLAAC will retrieve an output file with several values. For selecting the PrLD, the provided “COREaa” and “PRDaa” outputs differ slightly, as COREaa will select residues with the maximum score for a user provided length, while PRDaa will contain COREaa and the adjacent regions still displaying significant PrLD probability. Therefore, we will select PRDaa as the PrLD.

3.1.3 pWALTZ

pWALTZ is an update of the experimentally based amyloid predictor WALTZ, for the detection of soft-amino acid cores [22, 23]. It applies a 21-residue window, based on the minimum length of prion HET-s of *Podospora anserina* to form a transmissible β-fold. The algorithm will scan for soft-amino acid cores along the provided FASTA sequences and return the most probable candidate and its score (*see Notes 2, 4, and 5*). The highest performance for yeast PrD was achieved with a cutoff of 73.55. This is the algorithm’s default cutoff, although this threshold can be adapted for different species or purposes.

3.1.4 PrionW

PrionW searches for both prion-like domain and soft-amino acid core within them. It applies a disorder restriction with FoldIndex [19] and defines the prion-like domain searching for a Q/N-enrichment, similar to the one observed in yeast PrD [24]. Finally, it applies the pWALTZ algorithm on the Q/N-rich PrLD and for those sequences with positive results returns the selected PrLD and soft-amino acid core, along with the score (*see Notes 2, 5, and 6*). The algorithm will retrieve a downloadable output file with results for all positive sequences. The default cutoff parameters (73.55 pWALTZ score and 20% Q/N richness) allowed the best discrimination in yeast prions [24]. However, these parameters should be fine-tuned according to each particular proteome.

3.2 Soft-Amyloid Cores In Vitro Characterization

The predicted soft-amino acid cores are synthesized to explore their ability to drive the amyloid formation in vitro. These peptides bear the intrinsic ability to form intermolecular β-sheet assemblies and build amyloid fibrils. Here, we describe a set of standard techniques for amyloid characterization, which do not need especially expensive equipment or high-level training. This set of tools is suitable for routine monitoring of the aggregation state and verification of the fulfillment of the classical amyloid features.

3.2.1 Detection of Aggregates

Aggregates diffract the light proportionally to the amount of aggregated material increasing the sample turbidity; therefore, the presence of aggregates can be easily detected by measuring this parameter (*see Note 7*). Turbidimetry and light-scattering techniques are based on this principle and are widely used as a first step to characterize the amyloid formation [25].

The absorbance at wavelengths where the protein does not absorb (400 nm, for instance) is used to monitor the turbidity.

An increase in the absorbance correlates with the light diffracted by the aggregated forms [26, 27].

1. Prepare samples at the desired concentration (usually between 5 and 100 μM). Aggregation buffers can be used to accelerate the reaction. The sample can be incubated at different temperatures, with or without stirring (*see Note 8*).
2. Measure the absorbance at the selected wavelength. It is important to remain in the linear range of the spectrophotometer.
3. The increase in absorbance compared with that of the soluble protein indicates the presence of aggregates.

Synchronous light scattering is a technique based on the same principle but employing a spectrofluorometer, leading to a higher sensitivity. The excitation light diffracted by the sample is collected by a detector at the same wavelength. The intensity of the diffracted light is a measurement of the sample scattering which increases with aggregation.

1. Samples are prepared in the same way as previously described (*see step 1* in turbidity monitoring).
2. Excitation is fixed on a wavelength where the sample does not absorb or emits light (PEG 360 nm) and the emission in a range at least 10 nm higher and lower (350–370 nm). Acquisition is usually performed using slit widths of 5 nm, 0.5 nm interval, and 1000 nm/min scan rate at low sensibility. These parameters can be modified upon necessity.
3. The increase in the fluorescence signals in comparison with the soluble protein correlates with the light diffracted by the aggregates.

3.2.2 Bis-ANS Relative Fluorescence Determination

1-Anilinonaphthalene-8-sulfonate (ANS) and its dimeric analog 4,4'-bis-1-anilinonaphthalene-8-sulfonate (Bis-ANS) (Fig. 3a) are two of the most widely used dyes for partly folded intermediates characterization in protein-folding pathways [28]. Although both amphiphilic dyes present similar intrinsic fluorescence properties, in general Bis-ANS exhibits an increased binding affinity to the protein. This promotes a greater fluorescence enhancement compared to ANS when the protein undergoes structural changes, providing an environment shielded from water [28, 29].

In these conditions, the Bis-ANS fluorescence emission maximum presents a characteristic blueshift as well as an increase in the quantum yield. Thus, these spectral changes can provide valuable information about protein structural changes and more importantly evidence the exposure of hydrophobic core regions [12, 30].

1. The relative fluorescence spectra of Bis-ANS can be obtained using a spectrofluorometer measuring with an excitation

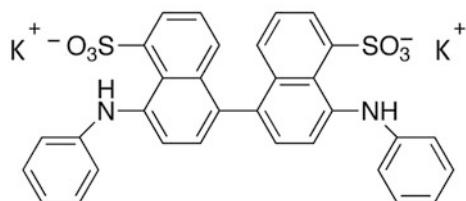
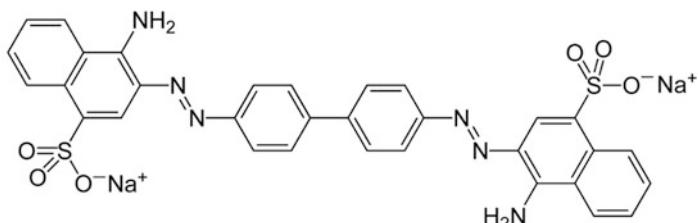
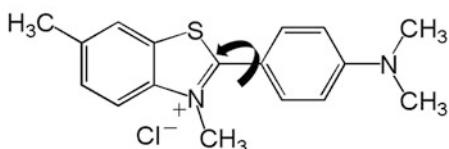
A**B****C**

Fig. 3 Structures of commonly used dyes for amyloid characterization: (a) Bis-ANS, (b) Congo red, and (c) Th-T; the rotation bond is indicated

wavelength of 370 nm and an emission range from 400 to 700 nm at the required temperature.

2. Samples are incubated at room temperature for 5–10 min prior to fluorescence measurements in the presence of 10 μM final dye concentration.
3. To visualize the characteristic blueshift in the incubated samples, the comparison of the fluorescence spectra of bound and free forms of Bis-ANS is recommended.

3.2.3 Amyloid Dyes

To confirm that the high-ordered beta-sheet assemblies are organized into amyloid-like supramolecular, the binding of specific and well-established dyes such Congo red (CR) and Thioflavin-T (Th-T) is convenient.

Since the 1920s when Benhold [31] and Divry established CR as a histological indicator of amyloid fibrils with relatively high specificity, it has been widely used for evidencing the presence of amyloid structures. CR, which is the sodium salt of the benzidine-diazo-bis-1-naphthylamine-4-sulfonic acid, undergoes a

characteristic spectral shift and birefringence upon binding to cross- β -sheet supersecondary structures (Fig. 3b). The spatial ordering that CR molecules adopt in aqueous solutions differs from the one that dye adopts in the presence of β -enriched amyloid fibril where it becomes torsionally restricted. This induces the characteristic shift in its absorption maximum from orange-red (490 nm) to pink (540 nm) [9, 12].

1. CR spectra can be obtained using a UV/Vis spectrophotometer measuring in wavelength-scanning mode in the 300–700 nm range using a matched pair of quartz cuvettes of 1 cm optical length, with instrument blanked on the appropriate buffer and thermostated at the required temperature.
2. The samples are sonicated for 10 min in an ultrasonic bath before dye addition. Note that this step is optional but recommended when the concentration of amyloid material is high.
3. Samples are incubated at room temperature for 10–15 min prior to spectral measurements in the presence of 5 μM final dye concentration (*see Notes 9 and 10*).
4. To observe the characteristic amyloid band at \approx 541 nm, the differential CR is plotted where the spectrum of the free CR must be subtracted from that of bound CR.

CR-binding mechanism has been explained through Scatchard analysis of equilibrium by assuming an independent binding mode relying upon the ability to discriminate between free and bound CR [32].

The present report describes a method of discriminating between bound and free dye by centrifugation and quantification of the non-precipitated dye [33]. Therefore, in this method only the knowledge of the molar absorptivity of the dye is required in comparison with the method developed by Klunk and collaborators [34], in which the knowledge of both the molar absorptivities of free and bound forms of dyes and peptides is required for the quantification.

1. CR spectra can be obtained as using a UV/Vis spectrophotometer measuring in wavelength-scanning mode in the 300–700 nm range and the instrument blanked on the appropriate buffer and thermostated at the required temperature.
2. Samples are incubated at room temperature for 5–10 min.
3. After incubation, the samples are centrifuged at $15,000 \times g$ for 30 min at room temperature.
4. The spectra of the supernatant are measured in the 300–700 nm range.
5. The amount of free CR in the soluble fraction is determined from the molar absorptivity of the free CR. Then, the amount

of dye bound to amyloid structures is calculated by subtracting the obtained concentration of the free CR from the total CR concentration in the assay.

The specific orientation that adopts CR when bound to β -sheet-rich amyloid fibrils renders a characteristic apple-green birefringence when CR-stained samples are microscopically examined through a cross-polarized light, thus providing a second sensitive diagnostic tool for demonstrating the presence of amyloid nature material [35, 36].

1. Samples are incubated at room temperature for 1 h in the presence of 50 μM CR.
2. When the concentration of amyloid material in the sample is low, we recommend centrifuging the above incubated sample at $14,000 \times g$ for 5 min at room temperature (*see Note 11*).
3. The precipitated sample is placed on a microscope slide and sealed.
4. The characteristic apple-green birefringence of CR-stained samples can be examined under cross-polarized light using an optic microscope.

Th-T has become an essential tool for imaging and monitoring the formation and growth of amyloid fibrils since its first description in 1959 [37–39]. Th-T is a molecule that includes a pair of benzothiazole and benzaminic rings freely rotating around a shared C–C bound (Fig. 3c). In aqueous solutions, Th-T rings rotate about one another promoting the stabilization of the relaxed form of the dye and therefore determining a low-quantum yield of fluorescence of Th-T under these conditions. When bound to amyloid-like structures, the rigidity of the environment prevents the rotation of Th-T rings inducing a high-quantum yield of fluorescence emission around 480 nm when excited at 440 nm (*see Note 12*) [9, 12, 40].

1. The relative fluorescence spectra of Th-T can be obtained using a spectrofluorometer measuring with an excitation wavelength of 440 nm and an emission range from 460 to 600 nm at the required temperature.
2. The samples are sonicated for 10 min in an ultrasonic bath before dye addition. Note that this step is optional but recommended when the concentration of amyloid material is high.
3. 25 μM of Th-T is added to each sample.
4. Samples are incubated at room temperature for 5 min prior to fluorescence measurements.

5. For monitoring the changes in Th-T relative fluorescence intensity over the time, maximal Th-T emission fluorescence at around 480 nm is followed when excited at 440 nm.

Th-T fluorescence microscopy assay turns out to be a simply and rapid alternative to examine the presence of amyloid-like material [9, 12]. Note that, whereas Th-T is the preferred dye for in vitro studies, Th-S is recommended when intracellular amyloid-like detection is needed. Th-S presents the unique capacity to penetrate biological membranes and to stain amyloid-like structures inside living cells [41, 42].

1. Samples are incubated at room temperature for 1 h in the presence of 125 μM Th-T.
2. The incubated samples are centrifuged at $14,000 \times g$ for 10 min at room temperature.
3. The samples are washed to remove the excess of Th-T by centrifugation and resuspension of the aggregated material in the appropriate buffer for two times.
4. The precipitated fraction is suspended in a final volume of 10 μL , placed on a microscope slide and sealed (*see Note 13*).
5. Images of the amyloid material bound to Th-T can be acquired at 40-fold magnification under UV light using a fluorescence microscope.

3.2.4 Secondary Structure Analysis

As a next step, the secondary structure is analyzed to provide further details about the aggregate features. In amyloid fibrils, peptides or proteins are stacked in antiparallel β -sheets perpendicular to the fibril axis, forming a structure known as cross- β [1]. A characteristic shift to β -conformations can be observed upon amyloid fibrillation, and therefore, the analysis of the β -sheet content is commonly used for amyloid identification. Circular dichroism (CD) and Fourier transform infrared (FT-IR) are widely used techniques for secondary structure analysis that provide insightful information about the aggregation state of a sample.

CD spectroscopy in the far-UV yields information of the relative secondary structure content of the sample. Since amyloid aggregation triggers a shift from the native spectra to a β -sheet-enriched one, characterized by a minimum at 217 nm, CD can determine the formation of the cross- β structure and the aggregation state of a sample [1, 43, 44] (*see Note 14*).

1. Sample is placed on a quartz cell (0.1 or 1 cm path length) at concentrations from 5 to 20 μM .
2. CD spectra are recorded in a wavelength from 190 to 250 nm at a scan rate of 100 nm/min and a resolution of 1 cm^{-1} at room temperature. Between 5 and 10 acquisitions are advised (*see Note 15*).

3. If it is necessary, CD spectra can be deconvoluted using different software such as K2D2 Suite (<http://k2d2.ogic.ca/>) [45] or DichroWeb (<http://dichroweb.cryst.bbk.ac.uk/>) [46].

FT-IR techniques provide detailed information about the secondary structure, allowing the differentiation of parallel, antiparallel, and intermolecular β -sheets. As the cross- β structure is assembled by antiparallel and intermolecular β -sheets, FT-IR spectroscopy complements the CD analysis by assessing the specific properties of the previously detected β -enrichment. Hence, amyloids show a characteristic band at $1620\text{--}1630\text{ cm}^{-1}$ corresponding to intermolecular β -sheets, plus an enrichment in antiparallel β -sheets at 1692 cm^{-1} . Solution absorption FT-IR is commonly used for protein characterization, but, in the case of amyloids, their tendency to precipitate compromises the analysis. This amyloid fibril insolubility can be overcomed by analyzing them in solid state, as a dried thin layer, using attenuated total reflection (ATR) FT-IR. This approach has become very popular for the analysis of amyloid fibrils as it does not require previous sample manipulation, it is cheaper, and it uses less amount of sample, around 100 ng [12, 47, 48].

1. IR spectra can be measured without previous manipulation, but, in case of buffer interference, it can be avoided by centrifugation and resuspension with Milli-Q water (two or three repetitions).
2. 5–10 μL of the sample are placed in a FT-IR spectrometer with a Golden Gate MKII ATR accessory and dried under N_2 . It is recommended to perform several drying steps (add smaller volumes ($\approx 2\text{ }\mu\text{L}$), dry, and add again), to avoid spilling sample out of the sample slot.
3. Acquire the spectra from 1700 to 1600 cm^{-1} of wave number at a resolution of 1 cm^{-1} . Accumulate 20 independent scans. It is recommended that the intensity of the peaks reaches 0.2.
4. The spectra can be fitted to overlapping Gaussian curves to obtain the contribution of each type of secondary structure. The amplitude, center, and bandwidth at half of the maximum amplitude and area of each Gaussian function are calculated by the use of a nonlinear peak-fitting program such as PeakFit (see Subheading 2.5). The second derivate is key to determine the center of each Gaussian curve, that is, the wave number at which the different spectrum components are located.

3.2.5 Aggregation Kinetics

Amyloid aggregation is a time-dependent reaction whose kinetics are characterized by three phases: a nucleation or lag phase, an elongation or exponential phase, and a plateau phase (Fig. 4). The transition to aggregated states can be monitored in a time-course experiment using several strategies such as the previously

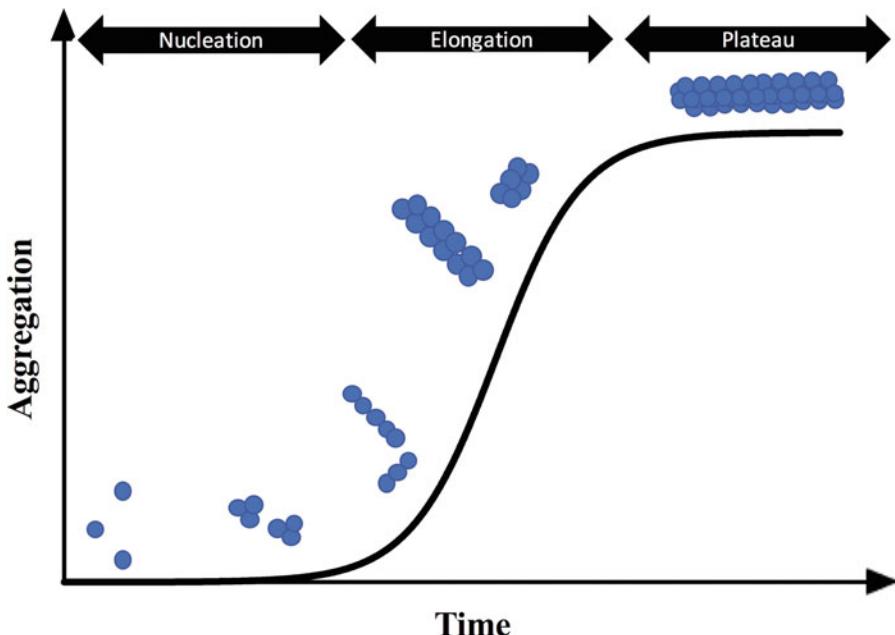


Fig. 4 Standard aggregation kinetics of amyloid-forming proteins: Ideal representation of the three key stages in amyloid aggregation. Schematic illustration of protein associations in each stage was added for a better understanding

commented Th-T binding, CD spectra shift to β -structure, light scattering, or turbidimetry [49, 50] (see Note 16). These experiments yield detailed information about the aggregation mechanism, and the results can be fitted to different equations to describe the kinetics [51, 52]. Further information and different equations are review by Morris et al. [53]. For instance, as the aggregation process can be described as an autocatalytic reaction, the following equation can be used to describe the reaction:

$$f = \frac{\rho(\exp[(1 + \rho)kt] - 1)}{(1 + \rho^* \exp[(1 + \rho)kt])} \quad (1)$$

$$k = k_c \alpha$$

where “ α ” is the protein concentration and “ ρ ” represents a dimensionless constant to describe the ratio of k_n to k . The values of “ ρ ” and “ k ” can be extrapolated by nonlinear regression of “ f ” against “ t ”.

1. Samples are prepared as described in step 1 from turbidity monitoring (Subheading 3.2.1).
2. Measure the aggregation using one of the previously commented strategies (turbidimetry, light scattering, and Th-T-relative fluorescence) at the desired time interval until a plateau is reached.

3. Normalize the curve.
4. For accurate data analysis, use the process described in previous review. In brief, analyze the data by fitting to Eq. 1 using a nonlinear regression program and extract the apparent constants from the regressions (*see Note 17*).

Seeding is an intrinsic property of amyloids by which pre-formed aggregates are able to nucleate the aggregation, promoting an acceleration of their aggregation kinetics. The seeding effect relies mainly on a shortening of the nucleation phase, producing a fast elongation phase. As this property is described as a key feature of amyloids, a positive seeding assay indicates that the generated putative amyloids fulfill this requirement.

1. The sample is prepared as described in **step 1** from turbidity monitoring (Subheading 3.2.1) but adding between 1% and 10% of a pre-aggregated sample.
2. Kinetics can be followed and analyzed as described above.
3. A positive seeding produces an effect on the “ kn ,” reducing the nucleation phase

3.2.6 Transmission Electronic Microscopy

Negative-stain transmission electron microscopy (TEM) has become the essential method to detect the presence of amyloid fibrils at high resolution as well as for assessing their morphology.

1. Place 5 μL of protein sample at a concentration of ~10 μM on the carbon-coated grid. Note that when the concentration of amyloid material is high, it is recommended to sonicate the samples for 10 min in an ultrasonic bath to avoid damaging the grids (*see Note 18*).
2. Samples are incubated at room temperature for 5 min.
3. The excess of liquid can be absorbed placing carefully ashless filter paper at the edge of the grid.
4. Immediately, the samples are stained with a 2% solution (w/v) of uranyl acetate and incubated for another 2 min.
5. Again, the excess of liquid is absorbed using ashless filter paper, and the grids are left to air-dry (*see Note 19*).
6. Amyloid fibrils are usually unbranched with ~5–15 nm in width. The fibrils can be easily detected at low magnification (10,000 \times).

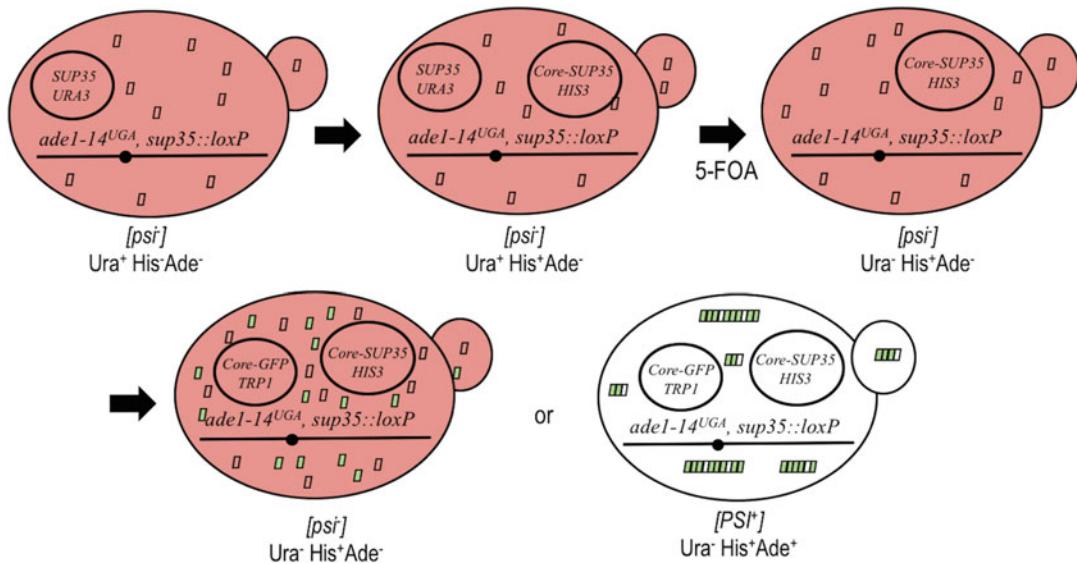


Fig. 5 Functional substitution of Sup35p by Core-Sup35p chimeras. Yeast cells with a premature stop codon in *ade1-14* gene and having soluble Sup35p terminator factor (empty rectangles) expressed from the *URA3-CEN*-expression plasmid show red phenotype and are dependent on adenine supplementation. These cells are then transformed with a *HIS3-CEN*-plasmid expressing the chimera *Core-SUP35*, and the *URA3*-vector is counter-selected with FOA. The unique source Sup35p is now the chimera. Next, transformation with a new plasmid capable to overexpress the fusion protein Core-GFP induces the aggregation of Core-Sup35p. Cells with aggregated Core-Sup35p are white and are not dependent on adenine supplementation. Moreover, they show punctate fluorescence foci (green rectangles)

3.3 In Vivo Validation of Soft-Amyloid Cores

3.3.1 Functional Substitution of Sup35p PrD by Soft-Amyloid Cores

In vivo validation can be carried out by substituting the PrD from the well-characterized yeast translational termination factor Sup35p with the soft-amyloid core sequence. Sup35p corresponds to the translation termination factor eRF3 and additionally accounts for the *[PSI⁺]* prionoid phenotype. When Sup35p aggregates in *[PSI⁺]* cells, the lack of soluble functional protein leads to a decrease in translation termination efficiency, which can be detected by a nonsense suppressor assay [54]. In this assay, a premature stop codon is present in *ade1-14* gene which inactivates it. Therefore, when Sup35p is soluble, these *[psi⁻]* cells are auxotrophic for adenine and show accumulation of the red metabolite from adenine biosynthesis 5'-P-ribosyl-5-aminoimidazole. In the opposite side, when Sup35p aggregates, the premature stop codon can be read-through, and *[PSI⁺]* cells can grow without adenine supplementation and become white or pink. Sup35p has three domains: the N-terminal bearing the prion-like domain, the middle domain (M) important for the mitotic stability of the prion, and the C-terminal domain responsible of the translation termination activity (Fig. 2). The modularity of Sup35p allows the substitution of its N-terminal domain by other PrLD, or in our case the soft-amyloid cores, and therefore to assay its ability to induce the *[PSI⁺]*

conversion in vivo. Parham and collaborators developed a plasmid-based assay [55]. This assay uses a yeast strain in which the *SUP35* chromosomal copy was disrupted by *loxP*. However, *SUP35* activity is not dispensable for the cell; therefore the viability is supported by expressing the wild-type *SUP35* gen from a *URA3* centromeric (*CEN*) plasmid (pYK810). Next, an expression plasmid based in *HIS3* selection marker, which expresses the new chimeras *Core-SUP35*, is introduced into the yeast (Fig. 5). Cells are then forced to lose the pYK810 plasmid by growing them in 5-fluoroorotic acid (5-FOA)-containing medium. This means that yeast becomes then auxotrophic for uracil (Ura⁻ phenotype). Yeast colonies are then tested for the *ade1-14* nonsense suppressor assay, where [*PSI*⁺] colonies show white or pink color, and Ade⁺ phenotype, while [*psi*⁻] colonies are red and Ade⁻. Since [*PSI*⁺] phenotype has low frequency, [*psi*⁻] cells are next transformed with and overexpression vector able to express the *Core-GFP* (or other fluorescent protein) chimera. This will induce aggregation and increase the [*PSI*⁺] phenotype (Fig. 5).

1. Introduce the selected *SUP35* chimeras (where the N-terminal domain is substituted by the soft-amyloid core) into a yeast expression vector (*CEN* vector) bearing a *HIS3* selection marker by the standard cloning methods (see Note 20).
2. Transform with the standard PEG/LiAc method [56] the yeast strain LJ14 derived from the strain 74D-694 (*MATa ade1-14UGA trp1-289 his3Δ-200 ura3-52 leu2-3,112 sup35::loxP* [pYK810]) [14] with the vectors containing the *Core-SUP35* chimeras, and select them on histidine lacking synthetic plates (SD-His plates). Incubate the plates for 3–5 days at 30 °C.
3. Inoculate individual colonies into SD-His 5-FOA to counter-selecting cells that have lost the pYK810 plasmid. Grow cells at 30 °C. Since 5-FOA results toxic to yeast having *URA3* gene active, cells that grow faster are those which have lost the pYK810 plasmid.
4. Replica plate those colonies which have grown on SD-Ura on SD-Ade and 1/4 YPD. Incubate plates at 30 °C for 3 days (colonies that do not grow on SD-Ura are those that have lost pYK810 plasmid). Transfer 1/4 YPD plates to 4 °C for 1 day, and inspect for positive cells showing the white/pink phenotype [*PSI*⁺] and the ability to grow without adenine. The probability of prion conversion is quite low; therefore probably it will be necessary induce its conversion as detailed in the next steps.
5. To increase the efficiency in prion conversion, cells with Ade⁻ phenotype and red color [*psi*⁻] are transformed with second plasmid able to overexpress the core fused to a fluorescent protein (by example *Core-GFP*) in a 2 μm yeast expression

vector bearing a different marker (for instance *TRP1*). Select cells by in the appropriate selective medium SD-Trp (in our example). Grow cells at 30 °C for 3 days (*see Note 21*).

6. Replica plate in SD-Ade to select [*PSI*⁺] cells. Incubate the plate at 30 °C.
7. Pick single colonies able to grow without adenine into 96-well microtiter plates containing 50 µL sterile PBS by well, and mix by pipetting up and down.
8. Inoculate these colonies from the microtiter plates simultaneously on 1/4 YPD and SD-Trp, (and optionally SD-Ade), and incubate them at 30 °C. After 3 days, transfer 1/4 YPD to 4 °C for 1 day more before inspection and photo documentation of the red [*psi*⁻] or white/pink phenotype [*PSI*⁺].
9. Positive cells should also show fluorescent foci when inspected by fluorescent microscopy. Take a fraction of positive colonies into Eppendorf tubes containing 100 µL PBS and vortex. Take 2 µL of the cell suspensions and place it in a microscope slide. Place a coverslip, and examine them under a fluorescent microscope at 1000× final magnification.

Positive [*PSI*⁺] colonies are those showing punctate fluorescence foci and white/pink phenotype when growing in 1/4 YPD plus Ade⁺ phenotype (*see Note 22*).

3.3.2 [*PSI*⁺] Curing

The propagation of [*PSI*⁺] prion seeds can be blocked by low concentrations of guanidine hydrochloride (GdnHCl). Reversion of [*PSI*⁺] phenotype to [*psi*⁻] is considered a classical sign of prionoid behavior. This allows discarding that Ade⁺ phenotype is not just due to a stop codon reversion in the yeast strain.

1. Positive colonies [*PSI*⁺] are simultaneously inoculated on YPD or YPD plus 5 mM GdnHCl plates to block prion propagation, and let them grow at 30 °C.
2. Colonies are inoculated side by side onto 1/4 YPD and, after growing at 30 °C, inspected for the change in color from white/pink to red depending if they come from YPD or YPD plus 5 mM GdnHCl. The change in color to red after GdnHCl treatment indicates [*PSI*⁺] curing (*see Note 23*).

3.3.3 In Vivo Seeding with Aggregated Protein or Peptides

A strategy to probe the infectivity of the prionoid protein (or a peptide), previously aggregated in vitro or in vivo, is to test its ability to seed aggregation into yeast by the method of spheroplasts' transformation previously reported by Tanaka and collaborators [57]. Spheroplasts are prepared from red colony [*psi*⁻] cells expressing the *PrLD-SUP35* that have lost the [*PSI*⁺] phenotype and transformed with the aggregated protein or peptide previously sonicated to increase the number of seeds.

1. Inoculate 20 mL of YPD with one red colony [ψ^-] expressing the *SUP35* chimera, and let it grow at 30 °C, 250 rpm overnight.
2. Inoculate 6 mL from the previous culture into 60 mL of YPD, and grow until $OD_{600} \approx 0.8$ in the same conditions.
3. Pellet cells by centrifugation at $3600 \times g$ (Allegra 6R centrifuge, Beckman Coulter).
4. Wash cells with 20 mL sterile water and pellet cells at $3600 \times g$.
5. Wash cells with 20 mL of 1 M sorbitol and pellet cells at $3600 \times g$.
6. Resuspend cells in 20 mL of SCEM buffer.
7. Keep a sample of 400 µL into an Eppendorf tube which will be used later for measuring the spheroplast efficiency.
8. Add 40 µL of lyticase (10,000 U/mL) to the cell suspension from the **step 6**, and incubate at 30 °C with occasional inversion.
9. After 40 min, measure spheroplasts' efficiency by mixing 400 µL of 1 M sorbitol plus 400 µL of spheroplasts' suspension and 290 µL 5% SDS. Do the same procedure with the pre-lyticase sample taken at **step 7**. Measure OD_{800} of both pre- and post-lyticase mixtures. The decrease in OD_{800} should reach around 80–90%; if not, keep the lyticase incubation for longer time, and repeat this step. While the lyticase incubation is working, perform the **steps 10–13**.
10. Denature salmon sperm DNA (1 mg/mL) at 95 °C for 10 min, and keep in ice (*see Note 24*).
11. Melt the top-agar at 80 °C. When melted, prepare for each transformation 10 mL supplemented top-agar by mixing 7 mL top-agar plus 1 mL of 10× YNB, 1 mL 10× drop-out –Ura –His, and 1 mL glucose 20%. Incubate the mixture at 50 °C until it is needed.
12. Incubate the selective plates SD-Ura –His at 37 °C until they are needed.
13. Sonicate the fibers in a water bath for 10 min twice.
14. Spin the spheroplasts at $900 \times g$ at room temperature for 5 min.
15. Resuspend the cell pellet gently in 1 M sorbitol by using a 1 mL pipette, and spin at $900 \times g$ for 5 min.
16. Resuspend the pellet in 10 mL STC buffer and spin at $900 \times g$ for 5 min.
17. Resuspend cells in 400 µL STC.

18. In Eppendorf tubes mix 100 μL of spheroplast cells, with 4 μL of a yeast vector containing *URA3* selection marker (like pRS416), 13 μL of ssDNA, and 13 μL protein extract from [*PSI⁺*] cells (or the aggregated and sonicated peptide). Prepare a negative control without the protein extract (or the peptide). Incubate the mixture for 10 min (see Notes 25 and 26).
19. Add 1170 μL of filter sterilized 20% PEG solution. Mix and incubate for 15–30 min.
20. Centrifuge at 5700 $\times g$ for 1–2 min.
21. Resuspend cells pellet into SOS medium, and incubate at 28–30 °C between 30 and 45 min.
22. Mix cells with 9 mL of supplemented top-agar, and pour it into the selective plates which have been previously tempered at 37 °C.
23. Incubate plates at 30 °C for 3–5 days, and inspect for [*PSI⁺*] phenotype as described previously.

4 Notes

1. The PAPA web server and Python script can be accessed at <http://combi.cs.colostate.edu/supplements/papa/>. Even if there is a web server, it cannot be used for detecting prion-like domains at the proteome level through it, as only one sequence is allowed at a time.
2. The user should note that all mentioned programs retrieve the best candidate under their scoring systems. This does not imply that some sequences may hold multiple PrLDs or soft-amino acid candidates.
3. Both the PLAAC Java script and the web server (accessible at <http://plaac.wi.mit.edu/>) allow individual or multiple sequence analysis. However, although the PLAAC script is very intuitive for programmers, the web server performs the same tasks with a very user-friendly website for non-trained users.
4. pWALTZ can be obtained at <http://bioinf.uab.es/pWALTZ/>. The executable file requires Linux. After download, file permissions should be changed to execute the file. This can be easily done at the terminal, in the file directory by typing “sudo chmod +x pwaltz2.7.” Afterward, a typical usage would be “./pwaltz2.7 fasta_file x,” where “fasta_file” would contain the PrLD FASTA sequences and optional cutoff “x” can be applied. If it is not included, the default cutoff of 73.55 will be applied.

5. pWALTZ (and PrionW) will not accept nonstandard amino acid residues (B, U, X, Z).
6. The PrionW webserver is accessible at <http://bioinf.uab.cat/prionw/>. It accepts up to 10,000 sequences per run. For more extensive proteomes, users should divide their input sequences into two or more files and submit them separately.
7. Be aware that differences in the aggregate morphology affect the diffraction; therefore, comparisons should be done carefully.
8. If the measurement is not done with stirring, the sample may precipitate in the bottom of the cuvette; the measurement should be done quickly to avoid it.
9. CR cannot be used for amyloid structures detection at acidic pH (<pH 3), since no spectral change is observed in these conditions.
10. The optimal CR concentrations for spectrophotometric analysis are in the range from 5 to 20 μM .
11. At this point, an optional wash to remove the excess of CR before the sample analysis can be performed by centrifugation and resuspension of the aggregated material in Milli-Q water for three times.
12. Th-T excitation and emission wavelengths can be slightly different depending on each specific amyloid aggregate morphology.
13. Wet mounts cannot be stored over long times as water evaporates; therefore we recommend examining the slides immediately.
14. If the native conformation is β -enriched, this shift cannot be observed.
15. Buffers with chiral molecules of a high-ion strength interfere with the signal and compromise the quality of the measure; they should be avoided or diluted with Milli-Q water in the sample preparation.
16. Transition to β structures can be measured by CD at a fixed wavelength of 217 nm.
17. For high-throughput screenings, the aggregation kinetics can be followed performing the aggregation in a 96-well plate, using a Victor 3.0 Multilabel Reader, PerkinElmer (Waltham, MA, USA).
18. Excessive salt in the sample buffer should be avoided, since salt crystals can interfere with TEM imaging. Centrifuging the sample buffer can be easily exchanged by Milli-Q water.
19. The grids can be immediately observed or can be stored for months and even years if properly stored in dust- and moisture-free EM grid boxes prior to examination.

20. In fact, this procedure can be adapted for any other selection marker (except for *URA3*). For instance, the pUKC1620 vector from Tuite laboratory can be used for this purpose by replacement of the *BamHI* and *EcoRV* N-terminal *SUP35* fragment with the soft-amino acid core [14].
21. It is recommended to confirm endogenous expression of fusions Core-Sup35p and Core-GFP fusions by Western blot analysis with an antibody against Sup35p and GFP (GeneTex or [Antibodies-online.com](#) Sup35, anti-GFP antibody (JL-8 antibody; Clontech for anti YFP), Mountain View, CA, Clontech).
22. This procedure can be adapted for alternatively expressing the PrLD-*SUP35* chimera and induce it to aggregate by overexpression of the soft-amino acid core (Core-GFP).
23. Alternatively, curability of [*PSI*⁺] can be achieved by overexpression of Hsp104 from a 2-μm plasmid with a strong promoter. Nevertheless, Hsp104 overexpression or inhibition with GdnHCl has proven to cure [*PSI*⁺] prion cells, but this is not true for all kind of yeast prions.
24. Sonicated calf thymus DNA can be used instead.
25. The *URA3*-vector is transformed together with the protein in order to select co-transformants by growing the cells in selective medium SD-Ura.
26. Peptide transformation is considerably less efficient than transformation with protein extracts derived from [*PSI*⁺] cells expressing the chimeras.

Acknowledgments

This work was funded by the Spanish Ministry of Economy and Competitiveness BIO2016-783-78310-R to S.V. and by ICREA, ICREA-Academia 2015 to S.V.

References

1. Chiti F, Dobson CM (2017) Protein misfolding, amyloid formation, and human disease: a summary of progress over the last decade. *Annu Rev Biochem* 86:27–68
2. Sipe JD, Benson MD, Buxbaum JN et al (2016) Amyloid fibril proteins and amyloidosis: chemical identification and clinical classification International Society of Amyloidosis 2016 Nomenclature Guidelines. *Amyloid* 23:209–213
3. Ventura S, Zurdo J, Narayanan S et al (2004) Short amino acid stretches can mediate amyloid formation in globular proteins: the Src homology 3 (SH3) case. *Proc Natl Acad Sci U S A* 101:7258–7263
4. Sikorska B, Liberski PP (2012) Human prion diseases: from Kuru to variant Creutzfeldt-Jakob disease. *Subcell Biochem* 65:457–496
5. Halfmann R, Jarosz DF, Jones SK et al (2012) Prions are a common mechanism for phenotypic inheritance in wild yeasts. *Nature* 482:363–368
6. True HL, Lindquist SL (2000) A yeast prion provides a mechanism for genetic variation and phenotypic diversity. *Nature* 407:477–483

7. Si K (2015) Prions: what are they good for? *Annu Rev Cell Dev Biol* 31:149–169
8. Espinosa Angarica V, Ventura S, Sancho J (2013) Discovering putative prion sequences in complete proteomes using probabilistic representations of Q/N-rich domains. *BMC Genomics* 14:316
9. Batlle C, de Groot NS, Iglesias V et al (2017) Characterization of soft amyloid cores in human prion-like proteins. *Sci Rep* 7:12134
10. Sant'Anna R, Fernandez MR, Batlle C et al (2016) Characterization of amyloid cores in prion domains. *Sci Rep* 6:34274
11. Valtierra S, Du Z, Li L (2017) Analysis of small critical regions of SwI1 conferring prion formation, maintenance, and transmission. *Mol Cell Biol* 37:e00206-17
12. Pallares I, Iglesias V, Ventura S (2015) The Rho termination factor of Clostridium botulinum contains a prion-like domain with a highly amyloidogenic core. *Front Microbiol* 6:1516
13. Yuan AH, Hochschild A (2017) A bacterial global regulator forms a prion. *Science* 355:198–201
14. Marchante R, Rowe M, Zenthon J et al (2013) Structural definition is important for the propagation of the yeast [PSI+] prion. *Mol Cell* 50:675–685
15. Malinovska L, Palm S, Gibson K et al (2015) Dictyostelium discoideum has a highly Q/N-rich proteome and shows an unusual resilience to protein aggregation. *Proc Natl Acad Sci U S A* 112:E2620–E2629
16. Chakrabortee S, Kayatekin C, Newby GA et al (2016) Luminidependens (LD) is an Arabidopsis protein with prion behavior. *Proc Natl Acad Sci U S A* 113:6065–6070
17. Michelitsch MD, Weissman JS (2000) A census of glutamine/asparagine-rich regions: implications for their conserved function and the prediction of novel prions. *Proc Natl Acad Sci U S A* 97:11910–11915
18. Toombs JA, Petri M, Paul KR et al (2012) De novo design of synthetic prion domains. *Proc Natl Acad Sci U S A* 109:6519–6524
19. Prilusky J, Felder CE, Zeev-Ben-Mordehai T et al (2005) FoldIndex: a simple tool to predict whether a given protein sequence is intrinsically unfolded. *Bioinformatics* 21:3435–3438
20. Lancaster AK, Nutter-Upham A, Lindquist S et al (2014) PLAAC: a web and command-line application to identify proteins with Prion-Like Amino Acid Composition. *Bioinformatics (Oxford, England)* 30:2–3
21. Alberti S, Halfmann R, King O et al (2009) A systematic survey identifies prions and illuminates sequence features of prionogenic proteins. *Cell* 137:146–158
22. Sabate R, Rousseau F, Schymkowitz J et al (2015) What makes a protein sequence a prion? *PLoS Comput Biol* 11:e1004013
23. Maurer-Stroh S, Debulpae M, Kuemmerer N et al (2010) Exploring the sequence determinants of amyloid structure using position-specific scoring matrices. *Nat Methods* 7:237–242
24. Zambrano R, Conchillo-Sole O, Iglesias V et al (2015) PrionW: a server to identify proteins containing glutamine/asparagine rich prion-like domains and their amyloid cores. *Nucleic Acids Res* 43:1–7
25. Zhao R, So M, Maat H et al (2016) Measurement of amyloid formation by turbidity assay—seeing through the cloud. *Biophys Rev* 8:445–471
26. Sant'Anna R, Gallego P, Robinson LZ et al (2016) Repositioning tolcapone as a potent inhibitor of transthyretin amyloidogenesis and associated cellular toxicity. *Nat Commun* 7:10787
27. Hammarstrom P, Jiang X, Hurshman AR et al (2002) Sequence-dependent denaturation energetics: a major determinant in amyloid disease diversity. *Proc Natl Acad Sci U S A* 99 (Suppl 4):16427–16432
28. Rosen CG, Weber G (1969) Dimer formation from 1-amino-8-naphthalenesulfonate catalyzed by bovine serum albumin. A new fluorescent molecule with exceptional binding properties. *Biochemistry* 8:3915–3920
29. Stryer L (1965) The interaction of a naphthalene dye with apomyoglobin and apohemoglobin. A fluorescent probe of non-polar binding sites. *J Mol Biol* 13:482–495
30. Hawe A, Sutter M, Jiskoot W (2008) Extrinsic fluorescent dyes as tools for protein characterization. *Pharm Res* 25:1487–1499
31. Steensma DP (2001) “Congo” red: out of Africa? *Arch Pathol Lab Med* 125:250–252
32. 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* 37:1273–1281
33. Sabate R, Estelrich J (2003) Pinacyanol as effective probe of fibrillar beta-amyloid peptide: comparative study with Congo Red. *Bio-polymers* 72:455–463
34. Klunk WE, Jacob RF, Mason RP (1999) Quantifying amyloid beta-peptide (Abeta) aggregation using the Congo red-Abeta (CR-abeta) spectrophotometric assay. *Anal Biochem* 266:66–76

35. Sabate R, Espargaro A, Saupe SJ et al (2009) Characterization of the amyloid bacterial inclusion bodies of the HET-s fungal prion. *Microb Cell Factories* 8:56
36. de Groot NS, Parella T, Aviles FX et al (2007) Ile-phe dipeptide self-assembly: clues to amyloid formation. *Biophys J* 92:1732–1741
37. Vassar PS, Culling CF (1959) Fluorescent stains, with special reference to amyloid and connective tissues. *Arch Pathol* 68:487–498
38. Hobbs JR, Morgan AD (1963) Fluorescence microscopy with Thioflavine-T in the diagnosis of amyloid. *J Pathol Bacteriol* 86:437–442
39. LeVine H 3rd (1993) Thioflavine T interaction with synthetic Alzheimer's disease beta-amyloid peptides: detection of amyloid aggregation in solution. *Protein Sci* 2:404–410
40. Sant'Anna R, Fernández MR, Batlle C et al (2016) Characterization of amyloid cores in prion domains. *Sci Rep* 6:34274
41. Urbanc B, Cruz L, Le R et al (2002) Neurotoxic effects of thioflavin S-positive amyloid deposits in transgenic mice and Alzheimer's disease. *Proc Natl Acad Sci U S A* 99:13990–13995
42. Espargaro A, Sabate R, Ventura S (2012) Thioflavin-S staining coupled to flow cytometry. A screening tool to detect in vivo protein aggregation. *Mol BioSyst* 8:2839–2844
43. Marinelli P, Pallares I, Navarro S et al (2016) Dissecting the contribution of *Staphylococcus aureus* alpha-phenol-soluble modulins to biofilm amyloid structure. *Sci Rep* 6:34552
44. 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* 12:407–423
45. Perez-Iratxeta C, Andrade-Navarro MA (2008) K2D2: estimation of protein secondary structure from circular dichroism spectra. *BMC Struct Biol* 8:25
46. Whitmore L, Wallace BA (2008) Protein secondary structure analyses from circular dichroism spectroscopy: methods and reference databases. *Biopolymers* 89:392–400
47. Sarroukh R, Goormaghtigh E, Ruysschaert JM et al (2013) ATR-FTIR: a "rejuvenated" tool to investigate amyloid proteins. *Biochim Biophys Acta* 1828:2328–2338
48. 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* 404:337–352
49. Collins SR, Douglass A, Vale RD et al (2004) Mechanism of prion propagation: amyloid growth occurs by monomer addition. *PLoS Biol* 2:e321
50. Pujols J, Pena-Diaz S, Conde-Gimenez M et al (2017) High-throughput screening methodology to identify alpha-synuclein aggregation inhibitors. *Int J Mol Sci* 18:E478
51. Jarrett JT, Lansbury PT Jr (1993) Seeding "one-dimensional crystallization" of amyloid: a pathogenic mechanism in Alzheimer's disease and scrapie? *Cell* 73:1055–1058
52. Sabate R, Gallardo M, Estelrich J (2003) An autocatalytic reaction as a model for the kinetics of the aggregation of beta-amyloid. *Biopolymers* 71:190–195
53. Morris AM, Watzky MA, Finke RG (2009) Protein aggregation kinetics, mechanism, and curve-fitting: a review of the literature. *Biochim Biophys Acta* 1794:375–397
54. Cox BS (1965) Ψ, A cytoplasmic suppressor of super-suppressor in yeast. *Heredity* 20:505
55. Parham SN, Resende CG, Tuite MF (2001) Oligopeptide repeats in the yeast protein Sup35p stabilize intermolecular prion interactions. *EMBO J* 20:2111–2119
56. Gietz RD, Schiestl RH (2007) High-efficiency yeast transformation using the LiAc/SS carrier DNA/PEG method. *Nat Protoc* 2:31–34
57. Tanaka M, Collins SR, Toyama BH et al (2006) The physical basis of how prion conformations determine strain phenotypes. *Nature* 442:585–589
58. Ter-Avanesyan MD, Kushnirov VV, Dagkesamanskaya AR et al (1993) Deletion analysis of the SUP35 gene of the yeast *Saccharomyces cerevisiae* reveals two non-overlapping functional regions in the encoded protein. *Mol Microbiol* 7:683–692
59. Dosztanyi Z, Csizmok V, Tompa P et al (2005) IUPred: web server for the prediction of intrinsically unstructured regions of proteins based on estimated energy content. *Bioinformatics* 21:3433–3434
60. Finn RD, Coggill P, Eberhardt RY et al (2016) The Pfam protein families database: towards a more sustainable future. *Nucleic Acids Res* 44: D279–D285



Chapter 13

Stable Substructures in Proteins and How to Find Them Using Single-Molecule Force Spectroscopy

Katarzyna Tych and Gabriel Žoldák

Abstract

Three-dimensional structures of proteins are a source of fascination for scientists, due to the beauty of their sequence-encoded architectures and their highly diverse range of functions. These functions include acting as powerful catalysts, signal receptors, and versatile molecular motors as well as being building blocks for macroscopic structures, thus defining the shape of multicellular organisms. How protein structure is organized and assembled at the sub-nanometer scale is of great current interest. Specifically, the discovery of stable substructures and supersecondary structures has inspired research into their potential use in rationally engineered proteins with tailor-made properties. Here, we show how the search for stable substructures in large proteins can benefit from recent advances in single-molecule force spectroscopy using highly sensitive dual-beam optical tweezers. Our chapter provides a step-by-step description of the experimental workflow for (1) preparing proteins for mechanical interrogation, (2) interpreting the data, and (3) avoiding the most commonly occurring mistakes.

Key words Protein elasticity, Mini-proteins, Engineering, Nanomechanics, Laser traps

1 Introduction

Engineering proteins with tailor-made mechanical and enzymatic properties is an exciting goal with a number of applications in fields such as biomedicine, ecology, and smart nanomaterials. We are currently far from achieving this ambitious goal; even for the design of simple proteins, the number of potential combinations is astonishingly high and hence not possible to explore by a brute force systematic approach. Therefore, the most cost- and the time-effective way is to design functional proteins based on specific design principles [1]. To gain valuable information about structure-sequence relationships, it is possible to apply principles used for protein structure prediction. For example, supersecondary structures, or motifs, have been classified and are commonly used for the prediction of structures [2, 3]. Examples of such supersecondary structures, where secondary structure elements combine

to form a larger-scale structure, include beta-barrels and coiled-coils. Whether supersecondary structures retain structural stability in isolation is unclear. Often, because of their small size, stable supersecondary substructures are difficult to identify using standard approaches; recently, stable substructures have been identified within proteins by mechanical interrogation using single-molecule optical tweezers [4–12].

Optical tweezers have their origin in physics, specifically in the work of Dr. Ashkin during his work at Bell Laboratories in the 1980s [13, 14]. The discovery of light trapping and therefore optical tweezers sparked research in diverse areas: from single molecules to cells [13, 15]. Examples of studied systems include molecular motors [16] and polymerases [17], the protease machinery [18, 19], and the folding energy landscapes of proteins and nucleic acids [12, 20]. However, there are still significant hurdles, which limit the widespread application of optical tweezers, such as the availability of instruments capable of performing such experiments to a high level of accuracy, and the complexity of the experimental protocols.

In recent years, several commercial optical tweezer instruments have been released; thus current challenges relate to the development of reliable molecular assays and methods. The protocols and practical advice described in this chapter are provided with the aim of addressing this. Single-molecule force spectroscopy protein assays are known for their low efficiency, intrinsic complexity, and necessity of repeatedly optimizing assay conditions. In this chapter, we explain in detail the individual steps necessary for carrying out single-molecule force spectroscopy measurements using optical tweezers and various checks to perform along the way. As many biological components are used for optical tweezer experiments, the evaluation of their performance, purity, and quality before starting single-molecule experiments is crucial. Despite implementing such quality control checks prior to starting single-molecule force experiments, we often found that “the proof is in the pudding,” and while some controls were essential for the performance of optical tweezer experiments, they were not entirely predictive of success, and several optimization rounds were necessary. It should be emphasized that each research group uses their own experimental protocols and molecular constructs; here we focus solely on our own procedures.

Subheading 3 is divided into five parts: (1) preparing protein-DNA conjugates, (2) preparing functionalized beads, (3) assembling a sample for force experiments, (4) performing single-molecule force experiments, and (5) identifying stable substructures.

2 Materials

All biochemical components have to be of analytical grade. Please use ultrapure water. Buffers and solutions should be degassed and filtered through 0.22 µm filter. Measuring and adjusting pH should be conducted using a calibrated pH meter.

2.1 Chemicals

1. TCEP: 200 mM TCEP stock solution pH adjusted.
2. PBS: phosphate buffer saline pH 6.7.
3. Sodium carbonate: 100 mM sodium carbonate pH 9.6 comprising 65% 420 mg/50 mL NaHCO₃ (84.01 g/mol) and 35% 530 mg/50 mL Na₂CO₃ (105.99 g/mol), frozen in 10 mL aliquots (-20 °C).
4. Sodium phosphate: 10 mM sodium phosphate pH 7.4 comprising 80% 78 mg/50 mL NaH₂PO₄·2H₂O (156.01 g/mol) and 20% 179 mg/50 mL Na₂HPO₄·12H₂O (358.15 g/mol), frozen in 10 mL aliquots (-20 °C).
5. I7.4: 25 mM imidazole (170.2 mg in 100 mL using a molecular weight of 60.08 g/mol). pH adjusted using 1 M HCl to pH 7.4. Keep 50 mL for I6.5, below, before the addition of 0.05% Tween-20 (250 µL of 10% pre-dilution per 50 mL).
6. I6.5: 25 mM imidazole pH 6.5 with 0.05% Tween-20. Adjust 50 mL of I7.4 to pH 6.5 and add 0.05% Tween-20 as for I7.4.
7. I6.5TT: 25 mM imidazole, pH 6.5, 0.1% Tween-20. Use 10 mL of I6.5 and add Tween-20 to a final concentration of 0.1%.
8. 5% Sulfo-NHS: 5 mg/100 µL NHS (*N*-hydroxysulfosuccinimide, ThermoFisher, #24510). Solubilize in H₂O immediately before use.
9. 5% EDC: 5 mg/100 µL EDAC (*N*-ethyl-*N'*-(3-dimethylaminopropyl)-carbodiimide-hydrochloride, 191.70 g/mol, ThermoFisher, #22980)). Solubilize in I6.5 immediately before use.
10. 10× Tris-HCl: 0.8 M Tris-HCl pH 7.4: dissolve 4.85 g/50 mL Tris-HCl (121.14 g/mol), adjust pH 7.4 with HCl, and freeze in aliquots (-20 °C).
11. Sodium phosphate with PEG: 10 mM sodium phosphate, pH 7.4, 0.05% Tween-20, 10 mg/mL PEG-1000. Heat the PEG (~70 °C) to make it fluid, and then make a pre-dilution in sodium phosphate. Combine 7.5 µL 10% Tween-20 with 15 mg PEG-1000. Adjust volume to 1.5 mL with 10 mM sodium phosphate, pH 7.4.
12. Sodium phosphate with BSA: 10 mM sodium phosphate pH 7.4, 0.05% Tween-20, 10 mg/mL BSA: Combine

- 1.5 mL sodium phosphate, 7.5 µL 10% Tween-20 and 15 mg BSA (Sigma, #A0281).
13. Storage buffer for 5 mg anti-digoxigenin beads: sodium phosphate with PEG with 5% glycerol and 0.1% sodium azide. Combine 10 µL glycerol, 188 µL sodium phosphate with PEG and 2 µL 10% NaN₃.
 14. Storage buffer for 5 mg streptavidin beads: sodium phosphate with BSA with 5% glycerol and 0.1% sodium azide. Combine 10 µL glycerol, 188 µL sodium phosphate with BSA and 2 µL 10% NaN₃.

2.2 Biochemicals

1. Oligonucleotides with three biotin groups.
2. Oligonucleotides with three digoxigenin groups.
3. Polyclonal Fab fragments antibody against digoxigenin.
4. Polyclonal anti-digoxigenin-rhodamine Fab fragments.
5. Streptavidin-functionalized beads.
6. Oligonucleotides with a maleimide group.
7. λ-DNA.
8. Components for PCR reaction.
9. Oxygen scavenger system. The oxygen scavenger system consists of three components: glucose oxidase, catalase, and glucose. Glucose is dissolved at a final concentration of 33%; glucose oxidase is dissolved to 26 U/mL. Catalase is dissolved to 1700 U/mL. Enzyme activities are measured using standard commercially available colorimetric assay. All three components are aliquoted (20 µL) and stored at -20 °C.

3 Methods

Here, all necessary procedures are described in a step-by-step fashion. Work is performed at room temperature if not stated otherwise.

3.1 Preparing Proteins and Protein-DNA Conjugates

Protein preparation steps are easy to establish and implement in a biochemical laboratory with standard biochemical equipment. Several additional points must be considered prior to commencing the experiments and purification steps. Figure 1 summarizes the steps in the experimental pipeline.

1. Protein design for optical trapping. Introduce cysteine residues in a protein for DNA-maleimide conjugation at carefully chosen positions (Fig. 1). If the protein is to be investigated for the first time using optical tweezers, the default positions for the cysteines would be at the N- and C-termini, flanked by

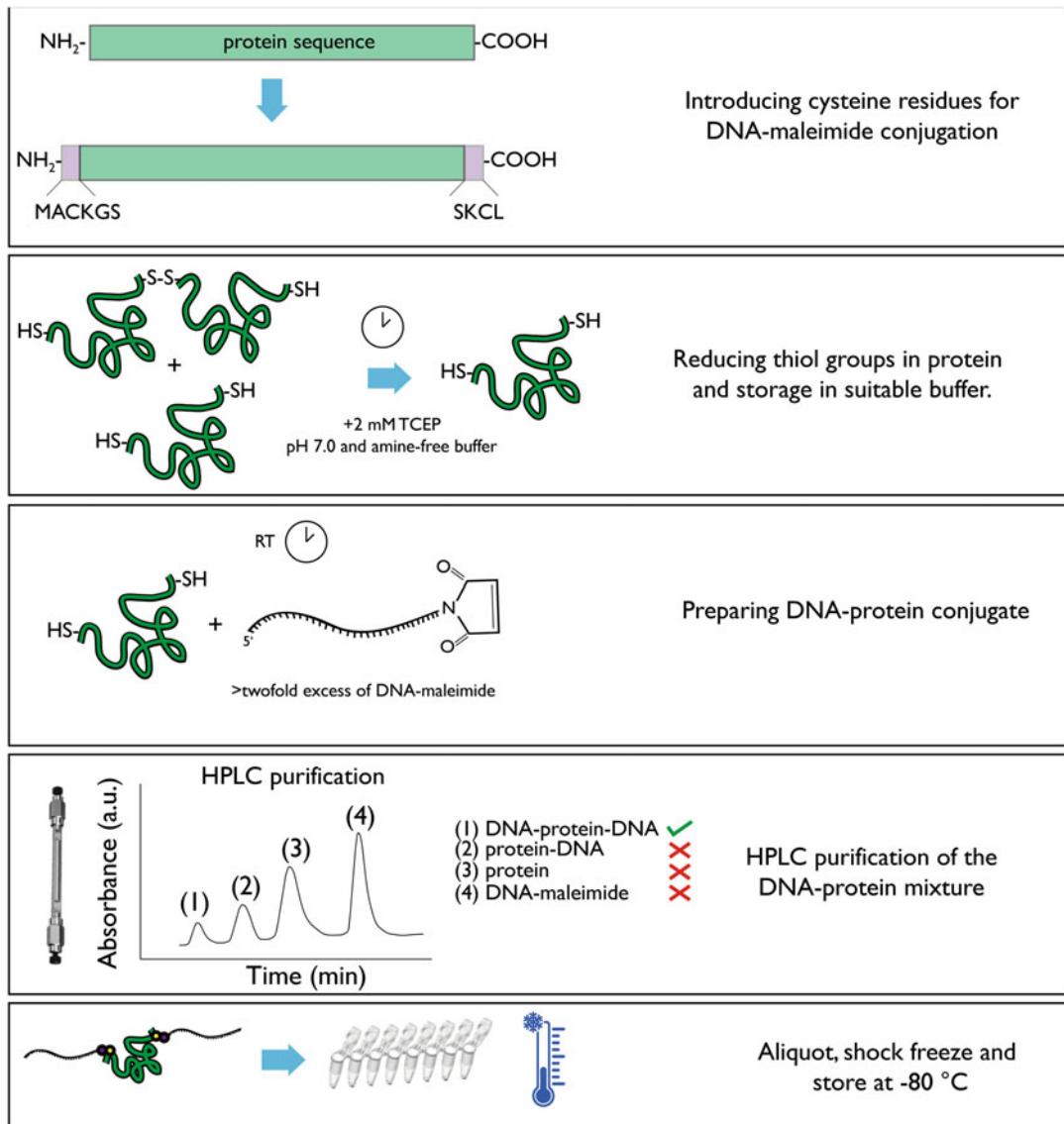


Fig. 1 Preparing proteins for single-molecule optical trapping experiments. The experimental workflow, starting from modifications to the amino acid sequence to the purification of the final protein-DNA constructs, is shown. In the first step, cysteine residues and, optionally, flanking amino acids are introduced into the protein sequence. The protein is then expressed and purified. In order to prepare the protein for attachment to DNA oligonucleotides, it must first be placed under reducing conditions (in this case in 2 mM TCEP), to reduce the thiol groups and break apart any disulfide bridges which may have formed between proteins. The TCEP is then removed, and the protein is incubated with a twofold excess of DNA-maleimide oligonucleotides. Finally, after a suitable incubation time, the protein construct consisting of two DNA-maleimide oligonucleotides and a single protein molecule can be obtained by size-exclusion chromatography. The resulting constructs can then be shock-frozen and stored at -80 °C until required for experiments

additional amino acids. For the N-terminus, a MACKSS sequence can be added to the wild-type protein, where the letter M is the first methionine. Here, the flanking residues before the start of the native protein sequence minimize adverse interactions with the folded protein. For example, the DNA used to attach the protein to the beads is a highly negatively charged molecule, which might interfere and destabilize the electrostatic potential at the protein surface. The positive charge of the lysine residue after the reactive cysteine residue can support conjugation by favorable electrostatic interaction with a negatively charged DNA-maleimide [21]. Note that due to the addition of flanking residues at the N-terminus, the sequence of the protein will be shifted by several residues. Therefore, using the numbering according to the wild-type protein is recommended. Please *see Note 1* for other general requirements of the protein and possible solutions.

2. Protein purification for conjugation reaction (His tag). For convenient purification and analysis, our protein constructs usually contain a hexahistidine tag at the C-terminus. This is also important for subsequent processing steps. If the hexahistidine tag interferes with downstream applications, proteins can be expressed with cleavable strep-tag or SUMO-tag. Reverse affinity columns can then be used to remove uncleaved protein, the protease, and the tag; the tag-free pure protein can then be found in the flow-through fraction. After regular expression and purification steps, the protein is finally purified using gel filtration and dialyzed against the desired buffer. Ideally, the buffer is free of amine groups. A tenfold excess concentration of TCEP should be added to reduce all thiol groups entirely. The reduced protein is then shock-frozen in liquid nitrogen and stored at –80 °C.
3. Preparing for the conjugation reaction. In the first step, short azide- or maleimide-functionalized single-stranded DNA oligonucleotides are attached to the protein. Oligonucleotides with maleimide groups can be synthesized by several companies. However, differences in the batch quality and significant batch-to-batch variations were observed which led us to the development of several control checks. Specifically, we used Ellman's test [22] to monitor the decrease in the concentration of thiol groups after their reaction with maleimide, size-exclusion chromatography to separate monomeric maleimide oligonucleotides from oligomerized ones, and hybridization assays. Please *see Note 2* for more details on the control methods.
4. Conjugation reaction. Before starting the conjugation reaction, the exact concentration of the individual reactive components is important to set the correct ratio between azide/

maleimide oligonucleotide and protein (2:1). To this end, protein and DNA concentrations are measured using a low-volume spectrophotometer (e.g., Nanodrop). In our reaction mixture, we usually mix 20–100 μ L of 50 μ M-reduced protein with oligonucleotides at twofold excess. For more details, *see Note 2*. The protein and oligonucleotides are incubated for 2–3 h at room temperature or left overnight at 4 °C or on ice.

5. Purification DNA-protein conjugates. After incubation, the sample contains several intermediate products, and therefore an additional purification step is necessary. In our lab, we have established HPLC procedures using high-pressure analytical columns (e.g., Yarra, SEC-3000, 3 μ m) for fast purification (ca. 15 min). While standard FPLC takes much longer, analytical columns provide excellent resolution and an adequate yield for single-molecule force spectroscopy experiments. After loading the conjugation mixture, the elution profile consists of several peaks (Fig. 1). To pick the correct fraction (DNA-protein-DNA conjugate), control HPLC experiments using just the DNA oligonucleotides as well as just the protein are highly recommended. At this stage, the elution profiles should consist of a single peak. The conjugation mixture should yield the combination of these peaks (corresponding to DNA only and protein only) and two new peaks (the DNA-protein conjugate with one oligonucleotide and with two oligonucleotides). The peak containing two DNA oligonucleotides and the protein is collected and used for single-molecule experiments. Some additional peaks can be seen as well (*see Note 3*). The correct fraction is then aliquoted and shock-frozen at –80 °C (Fig. 1).
6. Preparing DNA handles. The “DNA handles” contain specific binding epitope groups (e.g., biotin and digoxigenin) and have a single-stranded overhang for attachment to the short DNA oligonucleotides with which the protein has been functionalized. To produce these handles, the DNA sequence is amplified through PCR using a suitable polymerase which is free of exonuclease activity. For our applications, minor changes errors in the DNA sequence are unimportant, while maintaining an intact single-stranded overhang is of utmost importance. We found that the single-stranded overhang degraded during the PCR reaction when a DNA polymerase with exonuclease activity was used. PCR products are purified by commercially available kits. The homogeneity and purity of the DNA handles should be checked by agarose gel electrophoresis.

3.2 Preparing Functionalized Beads

Functionalized beads are commercially available. However, we experienced substantial batch-to-batch variations in the radius and functionalization levels of commercial beads, and therefore we

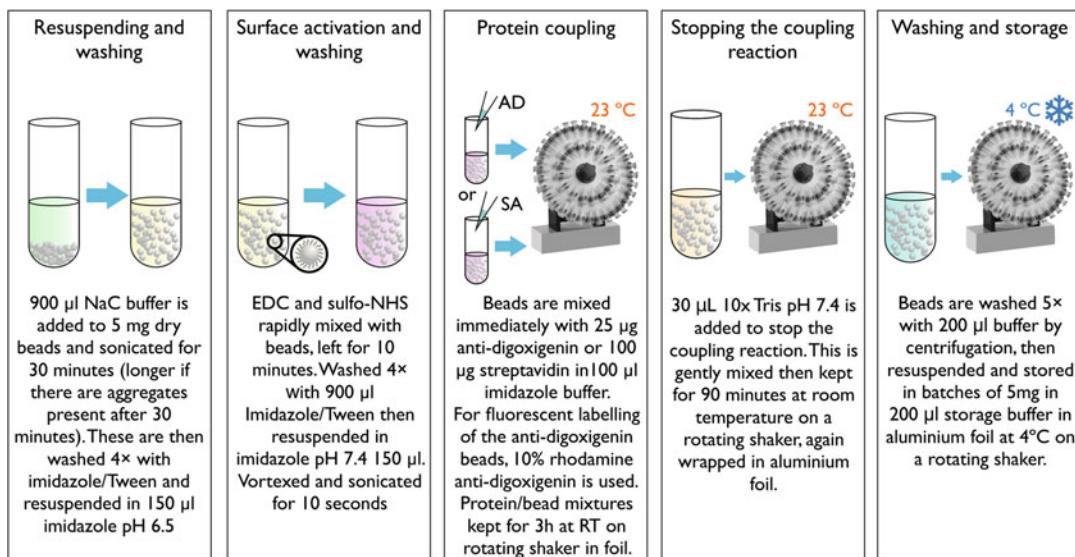


Fig. 2 Schematic illustration of the steps involved in preparation and storage of functionalized beads for optical trapping experiments. For definitions of buffer abbreviations, see Subheading 2

recommend preparing functionalized beads in-house. Beads can also be fluorescently labeled, e.g., using anti-digoxigenin-rhodamine fab fragments. This is useful in the case where optical tweezers are equipped with epi-fluorescence imaging for distinguishing between fluorescent and nonfluorescent beads. Beads are stored in a final buffer (*see* below) at 4 °C in a cold room on a continually spinning rotor. Figure 2 summarizes the critical steps in the experimental pipeline.

1. Resuspending and washing (Fig. 2): mix 5 mg dry beads with 900 µL sodium carbonate and put for ~30 min in a bath sonicator. Centrifuge for 1 min at 376 × g in a 2 mL eppendorf tube, and then resuspend by vortexing. Wash 4x with 900 µL I6.5. Resuspend pellet in 150 µL I6.5 by vortexing. After sonification, check for aggregation by visual inspection. In case of visible aggregates, sonicate for another 10 min.
2. Surface activation and washing: these steps must be performed quickly. (a) Dissolve NHS in water, and adjust pH to ~6.2; (b) dissolve EDC in I6.5 immediately, and make premix containing 1.6% EDC and 0.8% NHS in I6.5 (48 µL 5% EDC + 24 µL 5% NHS + 78 µL buffer per 5 mg beads); (c) add the premix to the beads and incubate for 10 min at room temperature on a rotating shaker¹; (d) add I6.5TT to a final volume of 900 µL, wash 4x with 900 µL I6.5TT (resuspend by vortexing), and resuspend the beads in 150 µL I7.4,

¹ During step c, prepare protein coupling premix used in the next step.

again by vortexing; (e) sonicate for 10 s and continue immediately with the next step.

3. Prepare protein coupling premix (per 5 mg beads): (a) mix 90 µg anti-digoxigenin (and 16 µg anti-digoxigenin-rhodamine if fluorescent labeling is desired) in 100 µL I7.4 for anti-digoxigenin-labeled beads or 100 µg streptavidin in 100 µL I7.4 for streptavidin-labeled beads; (b) add the bead suspension to the protein premix and mix gently for 3 h at room temperature on a rotating shaker; aluminum foil should be used to cover the sample—this is particularly important if fluorescent labeling is used.
4. Stopping the coupling reaction: add 30 µL 10× Tris-HCl pH 7.4 and mix gently for 90 min at room temperature on rotating shaker; if fluorescently functionalized, the sample should be covered with aluminum foil.
5. Washing and storage: wash 5× with 200 µL buffer—sodium phosphate with PEG for anti-digoxigenin and sodium phosphate with BSA for streptavidin. Centrifuge at $376 \times g$ for 40 s, but do not vortex. Next, resuspend beads in 200 µL storage buffer and store the sample covered with aluminum foil at 4 °C on a rotating shaker.

3.3 Assembling a Sample for Force Spectroscopy

Once the individual components are purified and ready, it is important to find the optimal ratios between them (e.g., between the DNA-functionalized protein and the DNA handles), to improve the success rate of experiments. The individual steps are shown schematically in Fig. 3.

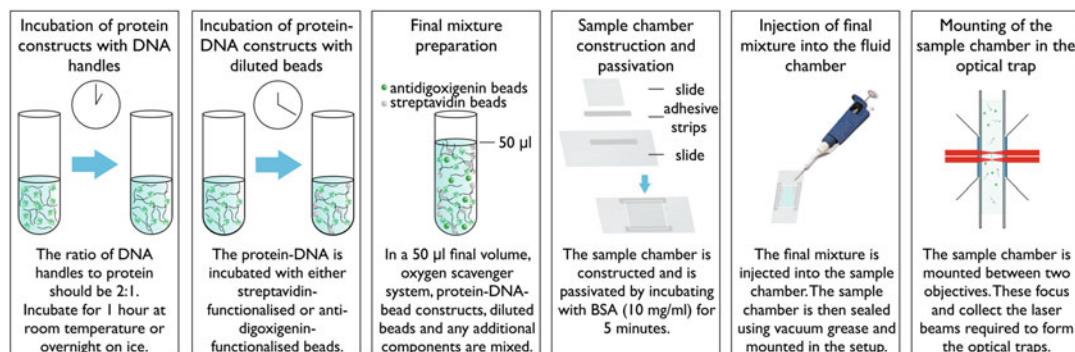


Fig. 3 Schematic illustration of the steps required for coupling the protein-DNA-maleimide-oligonucleotide constructs with DNA handles and beads and injection into a sample chamber in preparation for optical trapping experiments. The protein constructs are first combined with the DNA handles, before being incubated with diluted functionalized beads. The other bead type is then added, alongside the oxygen scavenger system and any other required components, and this mixture is injected into a fluid chamber. This is then mounted into the optical trapping experimental setup

1. Mix DNA-oligonucleotide-functionalized protein with DNA handles (solution “POH”—protein with oligonucleotides and handles) at a 1:2 ratio and incubate at least for an hour. It is possible to incubate the mixture overnight at 4 °C or on ice. In the case of low-temperature functionalization, low concentrations of protein and DNA should be used. For a more concentrated sample (ca. single digit μM range), dilute samples severalfold; a high concentration of assembled constructs results in poor reproducibility of traces and non-single-molecule conditions.
2. Our stock preparations usually contain approximately 3×10^7 beads/ μL . For the sample preparation, our stock solutions are diluted to ca. 10^5 beads/mL. Diluted stock solutions of beads are abbreviated as AD and SA.
3. Mix diluted anti-digoxigenin (AD) or streptavidin beads (SA) with POH; the resulting solutions (POH-AD or POH-SA) are incubated for 15 min.
4. Prepare the final solution, F, by combining the three components of the oxygen scavenger system and, for example, 1 μL POH-SA and 1 μL AD.
5. Load the sample into a preprepared chamber consisting of a coverslip and a microscope slide, seal with vacuum grease, and secure in place between the two objectives of the optical trapping instrument. The sample is now ready for single-molecule force experiments.

3.4 Performing Single-Molecule Force Spectroscopy Experiments

After mounting the sample chamber containing the protein sample over the first objective with a droplet of microscope oil, the second objective is placed in position touching a second oil droplet located on the back side of the glass slide. This setup geometry enables laser beams to pass through the sample chamber with their foci inside of the chamber. There are several adjustments and checks that need to be performed before starting measurements. Figure 4 summarizes the most significant steps. Possible pitfalls are described in Note 4.

Brownian motion of a micron-sized bead trapped in the laser focus can be precisely monitored and tracked by position-sensitive detectors. The optical trap potential can be reasonably well approximated by a harmonic potential which describes the trapping potential with trap stiffness k , typically with a value of 0.3 pN/nm, although this can easily be adjusted. Hence, if the average bead position is displaced by an increment Δx , the force which displaces the bead equals $k\Delta x$. This means that a displacement of the bead center from the trap center by 10 nm must result from an acting force of 3 pN. The calibration of the trap stiffness and of the position-sensitive detectors is described in Note 5.

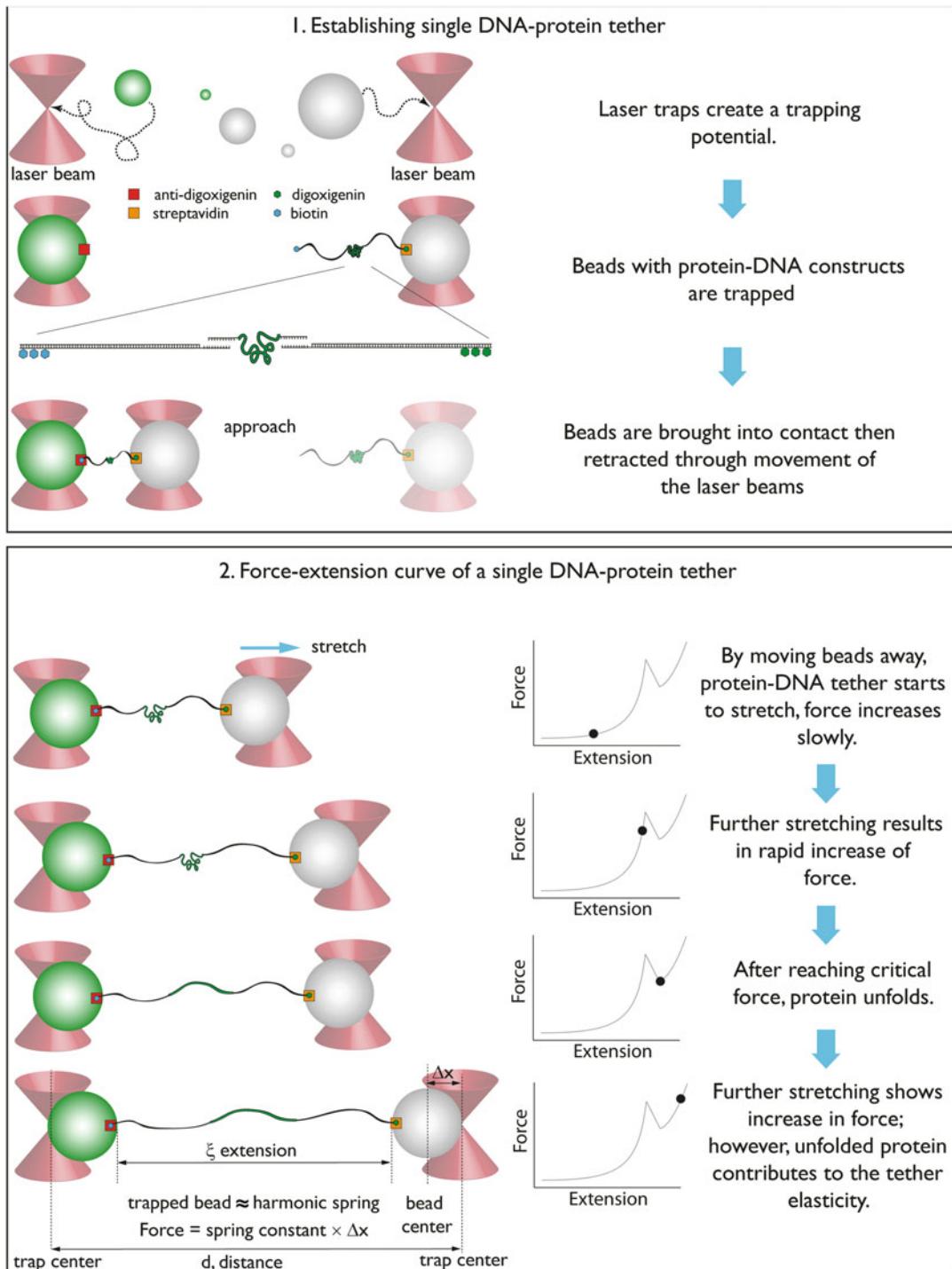


Fig. 4 Optical trapping of functionalized microspheres coupled to protein-DNA constructs and subsequent probing and force-extension experiments using single protein molecule. Force-extension plots (right panel) show the progress of the force-extension curve, while the beads are pulled apart

1. Because bead hydrodynamics change significantly in the proximity of a solid surface such as the glass slide from which the sample chamber is constructed, it is important that the experiments are conducted at a significant distance (tens of micrometers) away from the solid surface. Focused laser beams create a 3D trapping potential for particles, and one can use a trapped bead to find the exact position of the laser focus within the sample. It is possible to find the correct distance from the glass surface by manually adjusting the position of a trapped bead in the z -direction (i.e., using a screw micrometer to move the bead along the axis of the laser beam). At the moment when the trapped bead comes into contact with the surface, it is pushed out of the trapping center. This can be seen visually or measured with the signal from the detectors. After reaching the surface, the sample is shifted back by exactly 20 μm using a calibrated piezo-stage on which the glass slide is mounted. Because of possible drift in the optics and imperfections in the fluid cell geometry, it is recommended that the position of the laser foci relative to the glass surface is checked during experiments.
2. Having ensured that laser foci are located 20 μm away from the glass surface, measurements can begin. One bead of each type, in our case anti-digoxigenin and streptavidin, must be optically trapped in order to subsequently form the protein-DNA-bead dumbbell. It is during this step of finding complementary beads that efficiency is greatly improved when one of the bead types is fluorescently labeled. After catching a pair of complementary beads, one of the laser beams is brought toward the other, controlled by the steerable mirror, thus bringing the two beads together. The steerable mirror is adjustable in x - and y -directions and can be adjusted to control the distance and time for which the beads are brought into contact. Once beads are brought into contact, the protein-DNA-protein construct on the surface of one of the beads can reach the functional sites on the surface of the other and form the so-called dumbbell. This is illustrated in Fig. 4, where an anti-digoxigenin bead, functionalized with DNA-protein-DNA constructs carrying biotin groups, can bind to the streptavidin bead.
3. The physical connection between beads is realized through the DNA-protein-DNA tether; moving one bead in the x -direction results in force equilibration between both beads. In the experiment, the long DNA handles, in addition to providing the functionalization of the protein construct with digoxigenin or biotin, also act as spacers, enabling the protein's mechanical properties to be measured away from the bead surfaces and therefore the laser foci. The mechanical properties of the DNA

handles are well understood and can be described as a non-Hookean spring—at low extensions, force increases slowly, and at extensions close to the DNA contour length, the force increases rapidly. Specifically, DNA mechanics in the force range of interest are well described by the extensible wormlike chain model (*eWLC*) [23]:

$$F_{\text{eWLC}}(\xi) = \frac{k_B T}{p_{\text{DNA}}} \left(\frac{1}{4 \left(1 - \frac{\xi}{L_{\text{DNA}}} \right)^2} - \frac{1}{4} + \frac{\xi}{L_{\text{DNA}}} - \frac{F_{\text{eWLC}}}{K} \right),$$

where $F_{\text{eWLC}}(\xi)$ is the force as a function of DNA extension (ξ), k_B is the Boltzmann constant $1.38064852 \times 10^{-23} \text{ m}^2 \text{ kg/s}^2/\text{K}$, T is temperature (in Kelvin), p_{DNA} is the persistence length of DNA (ca. 25–50 nm), L_{DNA} is the contour length of DNA, and K is the stretch modulus (ca. 500 pN). DNA can be stretched to 65 pN, where a so-called BS transition is observed with a typical zigzag pattern. At this point the DNA has transitioned into an overstretched state, and hence the *eWLC* is no longer valid.

4. The retraction of one bead away from the other can be done following a number of different experimental protocols. The simplest of these are the constant velocity cycles. Here, one bead is retracted from the other at a constant velocity, usually in the range between 1 and 1000 nm/s (bead-to-bead distance increase per second). This type of experimental protocol usually performed in repeated cycles: pulling the beads to a certain distance then retracting, waiting at low force for refolding then again pulling, etc. Each experimental trace from a retraction and an approach can then be analyzed individually, giving information about the unfolding and subsequent refolding of the protein. In another commonly used experimental protocol, a fixed trap-to-trap distance is used, enabling equilibrium measurements to be performed. In this protocol, reversible changes in protein conformation can be observed over time, e.g., protein folding and unfolding. In a typical constant velocity force-extension trace, at a certain extension, a characteristic sudden drop in force is observed, which indicates either protein unfolding or dissociation. Such an event can be further analyzed, giving the rupture force and the corresponding contour length increase. After the protein unfolds, the unfolded polypeptide chain contributes to the mechanics of the tether. Compared to DNA and folded structured proteins, unfolded polypeptide chains are more compliant with a significantly lower persistence length which can be described with the wormlike chain (WLC) model [24]:

$$F_{WLC}(\xi) = \frac{k_B T}{p_{\text{protein}}} \left(\frac{1}{4 \left(1 - \frac{\xi}{L_{\text{protein}}} \right)^2} - \frac{1}{4} + \frac{\xi}{L_{\text{protein}}} \right),$$

where p_{DNA} is the persistence length of the protein (0.5–0.7 nm) and L_{protein} is the contour length of the unfolded protein. Importantly, a contour length of unfolded chain can be used for counting the number of amino acids involved in the unfolding transition by applying a factor of 0.35 nm per amino acids. The measured total extension is a simple combination of the *WLC* and *eWLC* fits. An example of a force-extension curve is shown in Fig. 4.

- Importantly, because the force-extension mechanical properties of the DNA handles are known, they can serve as control, and hence any events showing shorter DNA handle length or showing significant deviation from the expected WLC behavior can be discarded—they are likely the result of multiple tethering connections between beads and/or are due to nonspecific adhesion between proteins and beads.

3.5 Identifying Stable Substructures Using Force Spectroscopy

Single-molecule experiments provide information about the mechanical properties of the protein under investigation including contour length increases and unfolding forces. They also yield information about any stable substructures along the unfolding or refolding pathways of a protein. By measuring the contour length increases in the unfolding or refolding mechanical signature of the protein, the size of any stable substructures can be found. The three most commonly used methods for localizing stable substructures are depicted in Fig. 5. A further alternative method is suggested in Note 6. The protocol for identifying stable substructures using single-molecule force spectroscopy is given below:

- First, the protein is prepared for optical trapping experiments as described previously. In the example given in Fig. 5, the contour length increase when the native, fully folded protein unfolds (ΔL_c) is measured in the first force-extension curve. In this example, the substructure (purple, Fig. 5) unfolds as part of the main unfolding event (this is not always the case).
- After being fully stretched at high forces, the protein is then allowed to refold by bringing the beads closer together and waiting for a specified period of time at low (zero) force. At such low forces, substructures with a significant free energy gain can refold. Hence, during the second retraction cycle, the stretching of the protein shows an unfolding event which is much shorter compared to full-length protein unfolding. We term this shorter unfolding event $\Delta L_{c,\text{substructure}}$, as it corresponds to the unfolding of a mechanically stable substructure

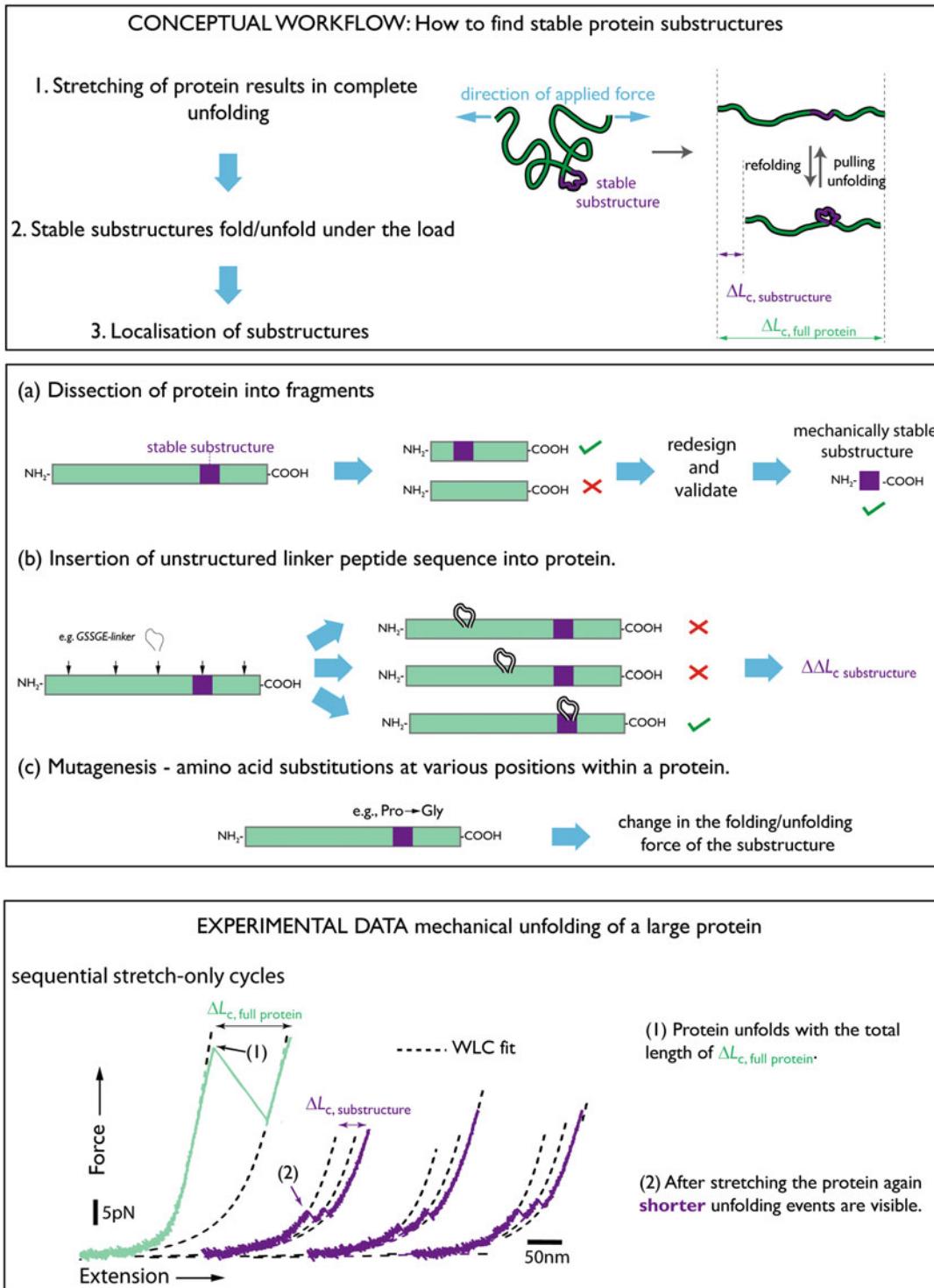


Fig. 5 The conceptual workflow needed for molecular identification of stable substructures, as well as an experimental example of a force-extension curve for full unfolding of a single protein. After unfolding, one can find much shorter unfolding events having lower $\Delta L_{c, \text{substructure}}$ and lower unfolding forces. These events are attributed to mechanically stable substructures with significant free energies

(purple, Fig. 5). Hence, for example, if one finds $\Delta L_{c,\text{substructure}}$ of 35 nm, this means that the substructure has a length of around 100 amino acids. To obtain a more precise estimate, a correction for the initial N- to C-terminus distance has to be subtracted (usually ca. 1–3 nm, this can be measured if a crystal structure of the protein is available). If this distance is unknown, one can still obtain a reasonable estimate of the size of the substructure.

3. While the size of the substructure is easy to estimate from the experiments, for the exact localization of the substructure and hence its molecular identification, additional experiments and new protein constructs are necessary. In our most frequently used method, the protein is dissected into fragments, and these fragments are prepared for single-molecule force experiments as described previously. For example, the protein can be dissected into two halves, A and B. Both new constructs, A and B, are measured in the optical trap, and the ΔL_c values recorded. The fragment containing the substructure will unfold with an ΔL_c value which is the same as that of the substructure (i.e., equals $\Delta L_{c,\text{substructure}}$). Furthermore, assuming that intramolecular interactions play no role in stabilizing the substructure, an identical unfolding force should also be observed. This truncation procedure can be repeated until the stable substructure has been localized.

4 Notes

1. Many proteins contain cysteine residues which can react with the DNA-maleimide oligonucleotides and hence cannot be used for the aforementioned conjugation protocol. However, if the thiol groups of these cysteine residues are buried in the structure or are nonreactive, the protocol can be used. To verify if there are reactive (accessible) thiol groups in native wild-type protein, one can use the Ellman's test [22]. A negative result means that cysteine residues in a native protein are not reactive and hence thiol groups can be introduced and used for DNA-maleimide conjugation. In the case where the protein already contains reactive cysteine residues, other coupling techniques have to be used. For example, halo-tag- [25] and click-chemistry-based conjugation strategies [26] can be employed.
2. In the case where conjugation needs to be optimized, one should start by estimating the exact reactant concentrations: numbers of reactive thiol groups in the protein and numbers of reactive DNA-maleimide oligonucleotides per microliter. The reactivity of the introduced thiol groups in the protein can be estimated directly using Ellman's test. For the DNA-maleimide

oligonucleotides, we recommend the determination of the reactivity and hence the concentration of reactive DNA-maleimide oligonucleotides for every batch purchased, based on our own observations of batch-to-batch and manufacturer-to-manufacturer differences and inconsistencies. The reactivity can be estimated using the reverse of Ellman's test. Here, a two- to threefold molar excess of a thiol-containing substance, such as mercaptoethanol or *N*-acetylcysteine amide, is mixed with the DNA-maleimide oligonucleotide. Once the reaction is completed, the concentration of residual thiol groups is estimated by Ellman's test. From this information, the relative fraction of reactive DNA-maleimide oligonucleotides is determined, which in our case is often between 60% and 80%.

3. HPLC profiles can sometimes be very complicated, but running control HPLC runs with protein- and DNA-maleimide-only samples helps in their interpretation. Ideally, these control HPLC runs should consist of a single uniform peak at a characteristic elution time. However, both protein-only and DNA-only HPLC profiles often contain additional minor peaks, corresponding to dimers and higher-order oligomers. Such profiles were found frequently for commercial batches of DNA-maleimide. In some cases, the long incubation time for the conjugation resulted in partial aggregation of a protein which appears as additional peaks in the HPLC profile. In such cases, buffer and salt concentrations should be optimized to minimize aggregation.
4. There are several potential pitfalls relating to the manipulation and installation of the sample chamber in the optical trapping setup. Here we summarize some recommended simple checks: (a) use glass slides with the correct thickness for which the setup was optimized, (b) match the immersion oil (water- or oil-based) to the objective type (information given by manufacturer), (c) avoid air bubbles in the immersion oil, and degas it if necessary, (d) do not use an excessively large oil droplet, (e) during the experiment, record power spectra regularly and check, and last but not least (f) use a log book to monitor and control the performance of the experimental setup over time.
5. The trap stiffnesses (pN/nm) and the sensitivities of the photo-detectors (nm/V) are calibrated following the method described by Tolić-Nørrelykke et al. [27]. Here, the fluid around two trapped beads is moved at a frequency of 20 Hz using a piezo-actuated table to which the fluid cell is mounted. The power spectral density of the position signal from the beads is measured and the corner frequency fitted.

6. If the protein dissection method described above does not work, an alternative method can be used where a ca. 20 amino acid unstructured linker is inserted into different parts of the protein. When such a construct is measured, a ca. 7 nm increase in $\Delta L_{c,\text{substructure}}$ indicates that the unstructured linker has been inserted within the stable substructure. It is common practice to prepare 5–6 different insertion variants, where the unstructured linker is inserted into different parts of the studied protein structure. Insertion points have to be chosen with care to avoid disruption of the overall protein structure. A linker sequence which has been successfully employed in our lab is the GSSGE repeat. This sequence was chosen based on its successful application for connecting six ClpX subunits in a single polypeptide chain (*see* the seminal work of Martin et al. [28]). The linker sequence has its origin in a natural sequence of gene-3-protein of the fd phage. Another alternative method is based on the introduction of single point mutations at different sites in the protein. Here, through changes in the force-extension signature for $\Delta L_{c,\text{substructure}}$ as well as in the rupture forces, one can conclude that amino acid substitution due to mutagenesis has affected the substructure and therefore pinpoint its location. Finally, one can use the fact that when two cysteine residues are introduced in spatial proximity in a protein structure, they form a disulfide bridge. Generally, a disulfide bridge does not break at the forces applied in optical trapping experiments; therefore part of the structure is prevented from unfolding by a disulfide bridge. This results in a shorter measured total contour length, ΔL_c . If the shorter contour length is found instead of the previously measured $\Delta L_{c,\text{substructure}}$, this indicates that the disulfide bridge is located within the stable substructure.

Acknowledgments

The authors have both worked for several years in the single-molecule laboratory of Prof. M. Rief at the Technical University of Munich (TUM), Germany. The various controls and methods that we describe have been developed and optimized by many talented Ph.D. students and Postdocs; in particular, we would like to thank Daniela Bauer, Anja Gieseke, Dr. Ulrike Majdic, Dr. Marco Grison, Dr. Alexander Mehlich, Dr. Markus Jahn, Dr. Johannes Stigler, and Dr. Ziad Ganim for their valuable contributions to the assay development and for many discussions. We thank Dr. Benjamin Pelz for constructing the extraordinary “Duck Trap” optical tweezers setup (no ducks were harmed in the making

of this instrument). G.Z. is supported by the CVTI reintegration grant. K.T. is supported by SFB863 from the DFG and a HFSP Cross-Disciplinary Postdoctoral Research Fellowship (LT000150/2015-C).

References

1. Rocklin GJ, Chidyausiku TM, Goreshnik I, Ford A, Houlston S, Lemak A, Carter L, Ravichandran R, Mulligan VK, Chevalier A, Arrowsmith CH, Baker D (2017) Global analysis of protein folding using massively parallel design, synthesis, and testing. *Science* 357 (6347):168–175. <https://doi.org/10.1126/science.aan0693>
2. Kister AE, Gelfand I (2009) Finding of residues crucial for supersecondary structure formation. *Proc Natl Acad Sci U S A* 106 (45):18996–19000. <https://doi.org/10.1073/pnas.0909714106>
3. Chiang YS, Gelfand TI, Kister AE, Gelfand IM (2007) New classification of supersecondary structures of sandwich-like proteins uncovers strict patterns of strand assemblage. *Proteins* 68(4):915–921. <https://doi.org/10.1002/prot.21473>
4. Jahn M, Tych K, Girstmair H, Steinmassl M, Hugel T, Buchner J, Rief M (2018) Folding and domain interactions of three orthologs of Hsp90 studied by single-molecule force spectroscopy. *Structure* 26(1):96–105.e104. <https://doi.org/10.1016/j.str.2017.11.023>
5. Mandal SS, Merz DR, Buchsteiner M, Dima RI, Rief M, Zoldak G (2017) Nanomechanics of the substrate binding domain of Hsp70 determine its allosteric ATP-induced conformational change. *Proc Natl Acad Sci U S A* 114(23):6040–6045. <https://doi.org/10.1073/pnas.1619843114>
6. Pelz B, Zoldak G, Zeller F, Zacharias M, Rief M (2016) Subnanometre enzyme mechanics probed by single-molecule force spectroscopy. *Nat Commun* 7:10848. <https://doi.org/10.1038/Ncomms10848>
7. Jahn M, Buchner J, Hugel T, Rief M (2016) Folding and assembly of the large molecular machine Hsp90 studied in single-molecule experiments. *Proc Natl Acad Sci U S A* 113 (5):1232–1237. <https://doi.org/10.1073/pnas.1518827113>
8. Bauer D, Merz DR, Pelz B, Theisen KE, Yacyshyn G, Mokranjac D, Dima RI, Rief M, Zoldak G (2015) Nucleotides regulate the mechanical hierarchy between subdomains of the nucleotide binding domain of the Hsp70 chaperone DnaK. *Proc Natl Acad Sci U S A* 112(33):10389–10394. <https://doi.org/10.1073/pnas.1504625112>
9. Hocking HG, Hase F, Madl T, Zacharias M, Rief M, Zoldak G (2015) A compact native 24-residue supersecondary structure derived from the villin headpiece subdomain. *Biophys J* 108(3):678–686. <https://doi.org/10.1016/j.bpj.2014.11.3482>
10. Jahn M, Rehn A, Pelz B, Hellenkamp B, Richter K, Rief M, Buchner J, Hugel T (2014) The charged linker of the molecular chaperone Hsp90 modulates domain contacts and biological function. *Proc Natl Acad Sci U S A* 111(50):17881–17886. <https://doi.org/10.1073/pnas.1414073111>
11. Zoldak G, Stigler J, Pelz B, Li H, Rief M (2013) Ultrafast folding kinetics and cooperativity of villin headpiece in single-molecule force spectroscopy. *Proc Natl Acad Sci U S A* 110(45):18156–18161. <https://doi.org/10.1073/pnas.1311495110>
12. Stigler J, Ziegler F, Gieseke A, Gebhardt JC, Rief M (2011) The complex folding network of single calmodulin molecules. *Science* 334 (6055):512–516. <https://doi.org/10.1126/science.1207598>
13. Ashkin A, Dziedzic JM (1987) Optical trapping and manipulation of viruses and bacteria. *Science* 235(4795):1517–1520
14. Ashkin A (1980) Applications of laser radiation pressure. *Science* 210(4474):1081–1088. <https://doi.org/10.1126/science.210.4474.1081>
15. Kellermayer MS, Smith SB, Granzier HL, Bustamante C (1997) Folding-unfolding transitions in single titin molecules characterized with laser tweezers. *Science* 276 (5315):1112–1116
16. Svoboda K, Schmidt CF, Schnapp BJ, Block SM (1993) Direct observation of kinesin stepping by optical trapping interferometry. *Nature* 365(6448):721–727. <https://doi.org/10.1038/365721a0>
17. Abbondanzieri EA, Greenleaf WJ, Shaevitz JW, Landick R, Block SM (2005) Direct observation of base-pair stepping by RNA polymerase. *Nature* 438(7067):460–465. <https://doi.org/10.1038/nature04268>

18. Aubin-Tam ME, Olivares AO, Sauer RT, Baker TA, Lang MJ (2011) Single-molecule protein unfolding and translocation by an ATP-fueled proteolytic machine. *Cell* 145(2):257–267. <https://doi.org/10.1016/j.cell.2011.03.036>
19. Maillard RA, Chistol G, Sen M, Righini M, Tan J, Kaiser CM, Hodges C, Martin A, Bustamante C (2011) ClpX(P) generates mechanical force to unfold and translocate its protein substrates. *Cell* 145(3):459–469. <https://doi.org/10.1016/j.cell.2011.04.010>
20. Woodside MT, Anthony PC, Behnke-Parks WM, Larizadeh K, Herschlag D, Block SM (2006) Direct measurement of the full, sequence-dependent folding landscape of a nucleic acid. *Science* 314(5801):1001–1004. <https://doi.org/10.1126/science.1133601>
21. Olins DE, Olins AL, Von Hippel PH (1967) Model nucleoprotein complexes: studies on the interaction of cationic homopolypeptides with DNA. *J Mol Biol* 24(2):157–176
22. Riener CK, Kada G, Gruber HJ (2002) Quick measurement of protein sulphydryls with Ellman's reagent and with 4,4'-dithiodipyridine. *Anal Bioanal Chem* 373(4–5):266–276. <https://doi.org/10.1007/s00216-002-1347-2>
23. Wang MD, Yin H, Landick R, Gelles J, Block SM (1997) Stretching DNA with optical tweezers. *Biophys J* 72(3):1335–1346. [https://doi.org/10.1016/S0006-3495\(97\)78780-0](https://doi.org/10.1016/S0006-3495(97)78780-0)
24. Bustamante C, Marko JF, Siggia ED, Smith S (1994) Entropic elasticity of lambda-phage DNA. *Science* 265(5178):1599–1600
25. England CG, Luo H, Cai W (2015) HaloTag technology: a versatile platform for biomedical applications. *Bioconjug Chem* 26(6):975–986. <https://doi.org/10.1021/acs.bioconjchem.5b00191>
26. Kolb HC, Finn MG, Sharpless KB (2001) Click chemistry: diverse chemical function from a few good reactions. *Angew Chem* 40(11):2004–2021
27. Tolić-Nørrelykke SF, Schaffer E, Howard J, Pavone FS, Julicher F, Flyvbjerg H (2006) Calibration of optical tweezers with positional detection in the back focal plane. *Rev Sci Instrum* 77(10):103101. <https://doi.org/10.1063/1.2356852>
28. Martin A, Baker TA, Sauer RT (2005) Rebuilt AAA + motors reveal operating principles for ATP-fuelled machines. *Nature* 437(7062):1115–1120. <https://doi.org/10.1038/nature04031>



Chapter 14

Supersecondary Structures and Fragment Libraries

Raphael Trevizani and Fábio Lima Custódio

Abstract

The use of smotifs and fragment libraries has proven useful to both simplify and increase the quality of protein models. Here, we present Profrager, a tool that automatically generates putative structural fragments to reproduce local motifs of proteins given a target sequence. Profrager is highly customizable, allowing the user to select the number of fragments per library, the ranking method is able to generate fragments of all sizes, and it was recently modified to include the possibility of output exclusively smotifs.

Key words Smotifs, Supersecondary structures, Fragment library, Protein motifs, Protein structure prediction

1 Protein Families and Domains

“The arrangement seems to be almost totally lacking in the kind of regularities which one instinctively anticipates.” This is the remark that Kendrew et al. used to describe the structure of myoglobin in their seminal article for the field of structural biology [1]. In spite of the paradoxical absence of any regularities in a seemingly complex structure, the authors auspiciously foresaw that some underlying “principles of construction,” albeit unknown at the time, would become apparent in the future. Indeed, the discovery that proteins are made of recurring parts that range from local motifs smaller than secondary structures [2, 3] to a network of long-range interactions that make up an entire domain [4, 5] is one of the cornerstones of structural biology.

Finding the native structure of a protein from the amino acid sequence alone is a formidable computational challenge associated with hundreds of degrees of freedom and complex energy functions that attempt to model the intricate interplay of forces involved in the protein structure [6].

The systematic classification of known proteins indicates that the distribution of known folds is highly skewed, seeing that most fit into a very limited number of families. As of the time of writing,

the CATH database classifies 95 million protein domains into 6119 superfamilies, and the SCOPe database classifies 77,439 PDB entries into 1221 folds, 2008 superfamilies, and 4851 families. A third of the proteins of a standard genome share one of the ten most common superfolds [7], and it is probable that the most prevailing folds have already been identified [8, 9], as it has been reported that approximately 45% of the fold space is already covered for globular proteins [10]. Additionally, the sequence is estimated to be three to ten times less conserved than the structure [11, 12] resulting, thus, in a limited number of different conformations coded by different sequences. These redundancies allow several thousands of models to be made from just a few structures and are the basis for template-based methods, such as comparative modeling [13, 14] and threading [15–17].

Template-based methods are the most ubiquitous and robust source for protein models [13, 14] and are routinely used in combination with experimental techniques for practical applications [18–20]. However, as template-based methods work best in cases with available known structures that share a certain degree of sequence similarity, their efficiency diminishes greatly as the demand for models of smaller protein families and singletons grows [21].

A template-free approach can be used as an alternative if no suitable template is found for the target, by either treating atoms as interacting particles and solving Newtonian equations via molecular dynamics simulation [22] or by using a meta heuristic algorithm such as a Monte Carlo [23] to optimize an energy function mapped by a physics-based classical force field (e.g., CHARMM, AMBER, etc.) or some potential function comprised of statistical and ad hoc terms. However, despite current progress [21, 24], even most state-of-the-art template-free methods still lag behind comparative modeling, chiefly due to the prohibitive cost intrinsic of the problem [6] and imprecisions in theoretical energy functions and models [25]. The shortcomings of template-free methods can be dealt with by taking advantage of the existing modularity in repeating sequence-structure motifs, such as smotifs and protein fragment libraries.

2 Fragment Libraries

The description of secondary structures [26, 27] and regular turns [28, 29] helped to reduce the protein space to a set of smaller, more tractable parts. Next, Jones and Thirup [30] discovered that a seemingly unique motif was, in fact, also found in three other proteins, regardless of any apparent evolutionary correlation. Subsequent efforts [2, 31] lead to the identification and classification of motifs coded by sequences of amino acids that share

common geometrical properties, known as fragment libraries, which were used to improve the comprehension of structural patterns and became fundamental to improve the quality of predicted structures by most CASP top performers, such as TASSER [32–34], ROSETTA [35], and QUARK [36].

A fragment library is traditionally defined as a collection of contiguous, local, short, extensively repeating structural patterns to bias local sequences toward motifs which are likely more consistent with the target. By extent, a fragment is a putative structure associated with a local minimum of the potential energy function. In protein structure prediction problems, for any given target sequence, the most suitable fragments are selected based on some criterion (e.g., sequence similarity), and a mix-and-match approach is used to assemble the final model, which enables the protein to jump between local minima, while it effectively explores the function hypersurface [35, 37].

The optimal fragment library contains as many structurally diverse fragments as possible while avoiding unnecessary redundancies and includes fragments that bear a strong local correlation with the target sequence regardless of the overall sequence similarity. Usually, a sliding window that scores the query sequence against a substitution matrix is used to select the fragments that will compose the library. It is also common to filter by secondary structure prediction [34, 35].

There are several determinants for the success of fragment libraries in protein modeling. Firstly, fragments are modular, as they are stable across different proteins, and the same motif can be found in different protein families, which means the desired conformation for a local sequence does not have to be excised from a restricted set of similar source proteins [3]. Furthermore, through the use of fragments in different patterns, it has been demonstrated it is feasible to explore the conformational universe beyond known structures, thus enabling new folds to be assembled [38].

Secondly, fragments are existing parts of known structures, which means they are stable in at least one chemical context. This frees the algorithm from the responsibility of accurately modeling the local structure and replaces it with the more simple challenge of finding the fragment that best satisfies the local query.

Thirdly, similarly to the philosophy of template-based methods, fragment libraries rely on the fact that the sequence is less conserved than the structure [12], resulting in a relatively small number of different conformations, thus enabling the search algorithm to perform more efficiently.

Finally, fragments are associated with local sequences that belong to different proteins. As they are part of a protein rather than an isolated molecule, they are subject to different forces when the chemical environment is changed. Therefore, despite the well-

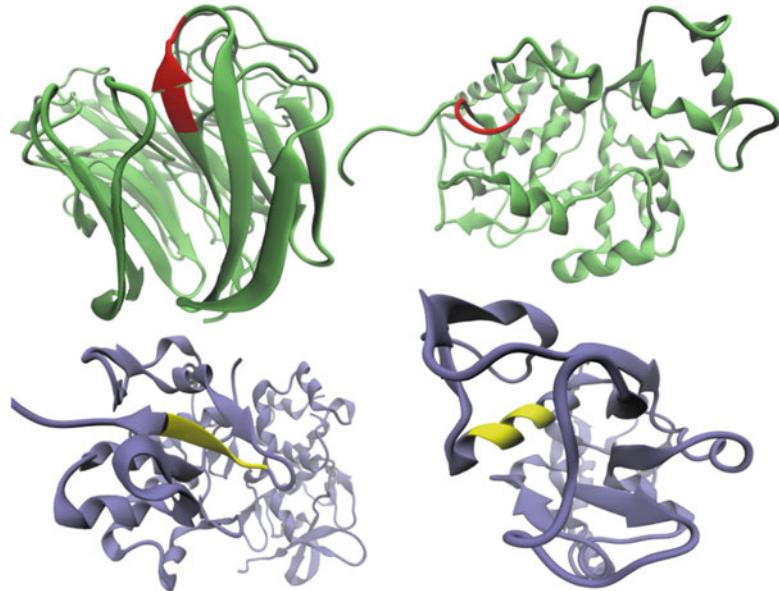


Fig. 1 Example of two identical 5-residue sequences in two different proteins that result in different structural motifs. Top: PDB codes 1F8E and 1BGP, highlighted sequence, DGSAT. Bottom: PDB codes 1LM5 and 1XS5, highlighted sequence, LRLLD

established norm that similar local conformations emerge from similar short sequences, there are notable exceptions wherein completely dissimilar structures come from the exact same sequence ([39], Fig. 1). The variety of structural states for similar sequences and the fact that the same structure can be found across protein families [40, 41] are the key points to the practical use of fragment libraries.

3 Supersecondary Structural Motif (Smotif)

A supersecondary structural motif (smotif) is composed of two flanking, regular secondary structures and their adjoining loop [42]. Thus, there are four possible smotifs for helices (H) and strands (E): E–E, E–H, H–E, H–H. Smotifs share a core characteristic with fragments in the sense that they are recurrent, modular motifs that repeat across different protein families, stabilized by local interactions. However, as smotifs are usually longer than the traditional fragment, they have been hypothesized to provide better local templates, especially for larger proteins [42]. As smotifs are also less numerous than a collection of smaller fragments, the search algorithm can perform more efficiently than traditional fragment assembly [43]. Conversely, the use of structure-derived motifs to model proteins requires them to be well represented in

the PDB, and the probability of finding the desired conformation decreases exponentially with size.

By defining the relative orientation of loops and their flanking secondary structures using four structural descriptors, it is possible to fit all known smotifs into 324 different clusters [44]. The authors then show it is possible to rebuild all known proteins from a finite library of smotifs, which indicates the coverage of smotifs in the PDB is close to complete. Accordingly, no new smotif has been deposited in the PDB since 2000 [10], and loop representations up to ten residues have been reported to have reached saturation in the PDB since as early as 2001 [45]. The same holds true for 12-mers since 2006 despite the exponential growth in sequence databases. Taken together, these works illustrate the completeness of smotifs in the PDB and open the way for several new practical applications in protein modeling studies. For example, a set of principles to combine smotifs in different patterns to explore the conformational space of tertiary structures has been proposed [46], which allowed the design of several structures with novel folds.

While searching for the optimal fragment length in fragment libraries, Handl and collaborators proposed fragments of smaller sizes should be derived from smotifs as they are sufficiently well represented in structural databases [47]. The work also supports the importance of smaller fragments in refining the structure, in particular, β -sheets, and recommends further investigation on which attributes result in a better fragment library. Another database that combines smotifs and fragments is Brix, which contains fragments of 4–14 residues long [48]. The authors showed that loop regions could be better reconstructed using smaller fragments, although regular secondary structures were best approximated by larger fragments. Later, it was reported that the variability of loops connecting helices is greater than those connecting β -strands [49]. Altogether, it is possible to conclude smotifs and fragments should be used concurrently as there is a befitting synergy between them. Smotifs are typically longer and simplify the search space by allowing more empirically determined information to be imported into the problem, whereas fragments are shorter and account for the multiplicity of conformational patterns coded by smaller sequences; as such, fragments are adequate to structural refinements.

4 Profrager: Protein Fragment Generator

Fragment and smotif libraries represent a selection of putative conformations that can be associated with a local sequence of the target protein and are extracted from a database of experimentally determined structures.

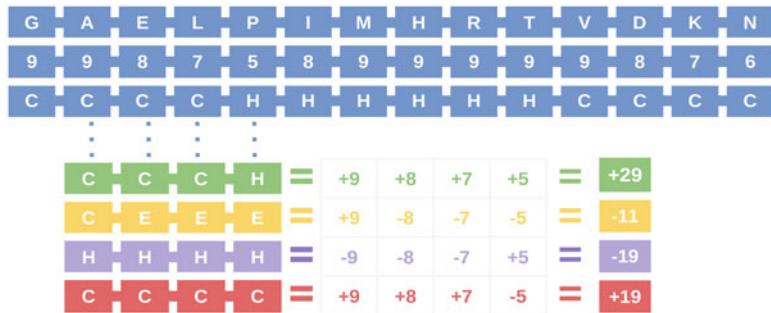


Fig. 2 Process for ranking a fragment using sequence similarity using the BLOSUM62 matrix. The value given for each matching pair is summed, and the result is the total score of the fragment. For example, taking the green fragment, the value of the A–G entry is 0, E–A is –1, L–E is –3, and P–L is –3. Combined, these scores result in –7, which is the sequence similarity score of this fragment

As programs that generate structural libraries should offer various options to guide the choice of fragments and smotifs, Profrager was developed in order to provide the user with fine control over the diversity/specificity desired. Profrager is a flexible program for creating customized structural libraries that allows control over many criteria used to select the fragments, such as the amino acid substitution scoring matrix and the database of experimental structures from which fragments are extracted ([50], <http://www.lncc.br/sinapad/Profrager/>).

At its first step, Profrager uses one of the three subsets of the PDB as sources for the fragments. The two first databases derive from PISCES and contain structures elucidated by X-ray crystallography (*R*-factor below 0.3): the first database contains 207 entries, with at most 10% identity between the sequences and resolution better than 1.0 Å, while the second is composed of 29,328 entries, with no more than 50% sequence identity resolution up to 2.5 Å. The third database is the Rosetta Vall database, with 16,800 sequence entries. These can be effortlessly updated, and customized, by the user from a list of PDB entries, e.g., a PISCES [51] generated culled list.

For a target sequence, the user inputs the desired size, and the program scans the chosen database to select a list of overlapping candidate fragments, starting from the first residue. Profrager can create libraries with fragments and smotifs of any length with no limit of structures for each position. To classify each fragment/smotif, they are assessed against the sequence similarity to the corresponding segment of the target sequence (Fig. 2). Sequence similarity score is computed using an amino acid substitution scoring matrix, and there are four available for the user to choose from: BLOSUM62 (default), BLOSUM45, PAM30, or PAM80.



Fig. 3 Process for ranking a fragment using secondary structure prediction. The total score is the sum of the confidence of PSIPRED (second blue row) for the predicted secondary structures (third blue row). Taking the yellow fragment as an example, the first residue is predicted as a coil (C) with a confidence of 9. As the prediction of the target matches the secondary structure of the fragment, the value used is +9. The secondary structure of the second residue is a strand (E), and the predicted is C, with a confidence of 8. As they do not match, the value used is -8. The same is used for the third (predicted C, fragment E, score of -7) and fourth (predicted H, fragment E, score of -5) residue. The PSIPRED score of the fragment is $+9 - 8 - 7 - 5 = -11$

Sequence similarity takes into account the probability of an amino acid being replaced by another in the protein sequence. Profrager offers the user the possibility to further filter the fragments by their secondary structure. This is performed with the aid of a secondary structure predictor (default: PSIPRED [52], but any other predictor may be used) applied to the target sequence. For each fragment, the secondary structure similarity score is given by the sum of PSIPRED confidence if the prediction matches the secondary structure assigned by STRIDE [53] for protein fragment in the database. Otherwise, the confidence is subtracted from the score (Fig. 3).

Both scores (sequence similarity and secondary structure score) are combined in the final score for ranking the fragment. The user is allowed to choose whether the scores will be weighted and summed (default weights: 1.0 for both scores) or if the fragments will be selected by way of a multi-objective Pareto efficiency strategy [54]. Pareto efficiency employs the concept of dominance where the best fragments in one of the scores are classified as non-dominated and make up the Pareto front. Successive fronts are used to build the fragment libraries until the desired number of fragments for each position is fulfilled. Other options available during library creation are (1) minimum score for a fragment to be included in the library, (2) exclusion of homologous proteins from the database, or (3) exclusion of nonhomologous proteins from the database. Homology is broadly detected as entries containing PSI-BLAST *e*-value <0.05 [55].

The Profrager output contains, for each fragment of each position ranked from best to worst, the following: (1) PDB code and chain of the source protein from which the fragment came; (2) one-letter code of the residue; (3) secondary structure; (4) position in the target sequence; (5) position in the source protein; (6) backbone dihedral angles φ , ψ , and ω ; (7) main chain bond angles defined by N–C α –C, C α –C–N, and C–N–C α ; (8) sequence similarity score; (9) secondary structure score; and (10) final score. Profrager by default also creates an output file fully compatible with the Rosetta software suite and a plot representing the secondary structure distribution throughout the library.

It has been shown that the use of backbone angles from structures with idealized or fixed bond geometries results in less accurate models when compared to fragments with bond angles directly derived from experimental structures, especially for larger proteins [56]. Thence, the output files include the backbone bond angles extracted directly from experimental structures along with the backbone torsion angles (**item 7** in the paragraph above). It is noteworthy that this issue is solved differently by Rosetta, which freezes the bonds under idealized geometries and optimizes the backbone dihedral angles to recreate structures as close as possible to the originals but with idealized geometries [35]. The user is given the choice to use Rosetta's geometry database to generate the libraries, in which case they will contain recalculated dihedral angles and fixed idealized main chain bond angles.

Currently, Profrager has two possible modes of selecting fragment sizes: manual selection, in which the default sizes are 3-, 6-, and 9-mers, and automatic selection based on secondary structure predictions. The automatic selection chooses fragment sizes based on the longest predicted secondary structure segment, i.e., if there is a predicted 30-residue-long extended conformation (β -strand), Profrager will select fragment sizes from 3- to 30-mers. Currently, an experimental third option is available, where fragment sizes are selected with basis on the detected smotifs on the secondary structure prediction, that is, Profrager scans for segments with three consecutive secondary structure types where the middle is a coil. Each detected smotif contributes to a fragment size.

5 Ideal Features of Structural Patterns for Practical Use

While trying to determine the optimal characteristics of a fragment library [57], we found that smaller fragments have the potential to generate more accurate results when models are created by libraries composed of single-sized fragments. In our data, models rebuilt exclusively from 12-residue-long fragments (12-mers) or larger did not have the same quality as those built from smaller fragments, possibly because they might be insufficiently well represented in

databases since the structural diversity grows with the number of residues [48]. When rebuilding the native structure, another identified problem is the insertion of a longer fragment on later stages of the simulation which may bring changes that clash with the rest of the structure.

However, for all fragment sizes, single-size libraries were incapable of generating proper protein models for larger proteins, as the quality of models substantially decreases with the increase of the target size. Such loss in quality is less pronounced when mixing libraries of different sizes because shorter and longer fragments are able to compensate the shortcomings of each other. Despite the disadvantages of longer fragments when compared to smaller sizes, our findings suggest they have desirable, seemingly indispensable properties for better model accuracy when fragments of multiple sizes are used to rebuild the native structure, which is in accordance with previous findings [47].

The best models are obtained by the algorithm when various sizes of fragments are assembled in a particular pattern: in general, the larger the fragment size, the earlier in the simulation the fragment should be used because it brings more empirical information into the model, thus enabling the optimization algorithm to bias the structure toward the native fold. As the simulation progresses, smaller fragments become more important because more subtle changes are required, while, contrastingly, longer fragments matter less because they might cause structural changes that are too extreme. In its final stages, the algorithm inserted almost exclusively the fragments of smallest sizes, indicating they are ideal for local structural refinement (Fig. 4).

A series of different works demonstrate the use of fragments of different sizes generates closer to native models [36, 58], while Rosetta, for example, is a well-established method for protein structure prediction that only uses 9-mers and 3-mers. On the other hand, too many different fragment sizes will result in large libraries which inherently make the search processes inefficient. With this, to improve the efficiency of the search algorithm, we asked the question whether it is possible to remove some sizes without negatively impacting model quality. We found no significant differences between the qualities of the models generated by 4-mers to 8-mers, which implies the existence of a structural redundancy in fragments of close sizes, and therefore, fragments of one size (e.g., 6-mers) can be used to represent an entire set of fragments of similar sizes (e.g., 4- to 8-mers). By contrast, models built by 3-mers and 9-mers were found to be significantly different from each other, and using a mixed library exclusively composed of 3-, 6-, and 9-mers proved adequate to reproduce the native structure with the same quality as that of mixed libraries ranging from 3-mers to 20-mers. The fact that only a few fragment sizes are enough to model the native structure suggests all other sizes seem to add too

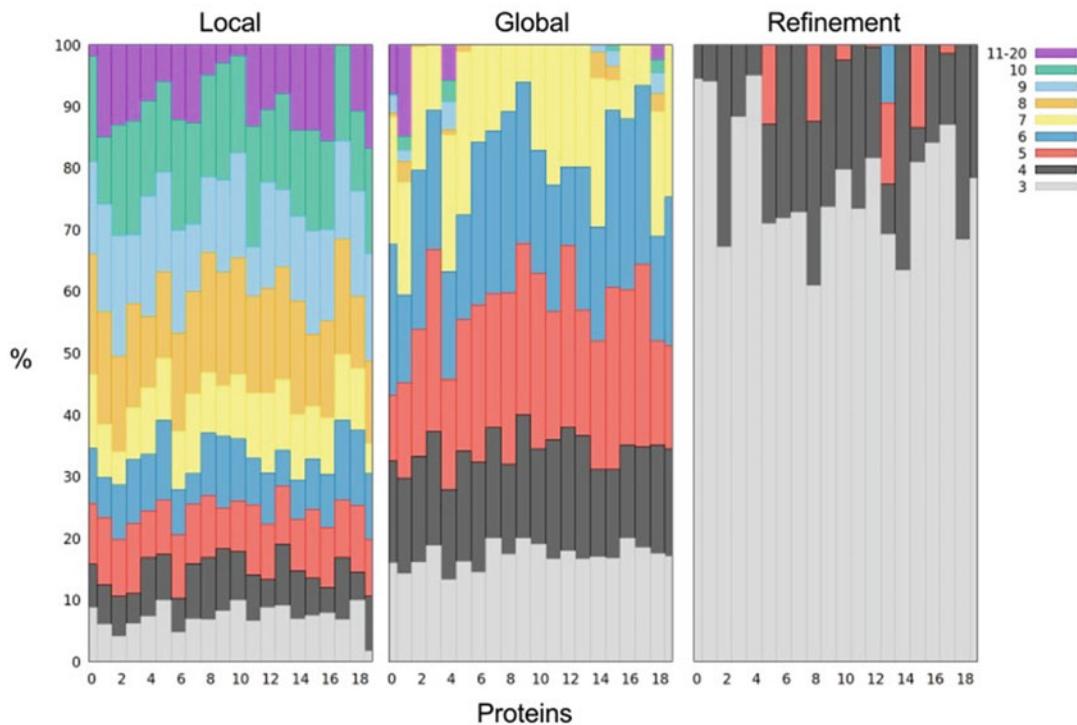


Fig. 4 Frequency of fragment insertion during the assembly of an optimal structure via a greedy algorithm. The bars show the relative amount of use of each fragment size in different stages of the algorithm for the best models obtained. The “Local” and “Refinement” phases attempt to find the fragment that best fits the structure locally, and the “Global” phase inserts the fragment that is optimal for the entire protein [50]. Fragments of all sizes are used during the Local stage, as they all bring empirical information into the model. As the simulation advances, the smaller fragments gradually replace the larger, and fragments with more than ten residues become almost completely ineffective from the Global stage onward. Finally, during the Refinement phase, only the smallest fragments (less than six residues long) are used since they induce smaller perturbations on the structure. X-axis numbers represent the proteins from the test set used by Trevizani et al. [50]

many redundancies and little valuable information, which unnecessarily increases the search effort. It is important to emphasize that the sizes are not necessarily restricted to the specific combination of 3-, 6-, and 9-mers, and, in fact, with the increase of structural data deposited in the PDB, the current size barrier is expected to be surpassed.

References

- Kendrew JC, Bodo G, Dintzis HM, Parrish RG, Wyckoff H, Phillips DC (1958) A three-dimensional model of the myoglobin molecule obtained by X-ray analysis. *Nature* 181 (4610):662–666
- Unger R, Harel D, Wherland S, Sussman JL (1989) A 3d building blocks approach to analyzing and predicting structure of proteins. *Proteins* 5(4):355–373
- Kolodny R, Koehl P, Guibas L, Levitt M (2002) Small libraries of protein fragments

- model native protein structures accurately. *J Mol Biol* 323(2):297–307
4. Nepomnyachiy S, Ben-Tal N, Kolodny R (2017) Complex evolutionary footprints revealed in an analysis of reused protein segments of diverse lengths. *Proc Natl Acad U S A* 114(44):11703–11708
 5. Xie ZR, Chen J, Zhao Y, Wu Y (2015) Decomposing the space of protein quaternary structures with the interface fragment pair library. *BMC Bioinformatics* 16:14
 6. Lee J, Freddolino PL, Zhang Y (2017) Ab initio protein structure prediction. In: Rigden DJ (ed) From protein structure to function with bioinformatics. Springer, Dordrecht, pp 3–35
 7. Cuff AL, Sillitoe I, Lewis T, Clegg AB, Rentzsch R, Furnham N, PellegriniCalace M, Jones D, Thornton J, Orengo CA (2011) Extending cath: increasing coverage of the protein structure universe and linking structure with function. *Nucleic Acids Res* 39: D420–D426
 8. Grant A, Lee D, Orengo C (2004) Progress towards mapping the universe of protein folds. *Genome Biol* 5:107
 9. Andreeva A, Howorth D, Chandonia JM, Brenner SE, Hubbard TJP, Chothia C, Murzin AG (2008) Data growth and its impact on the scop database: new developments. *Nucleic Acids Res* 36:D419–D425
 10. Khafizov K, Madrid-Aliste C, Almo SC, Fiser A (2014) Trends in structural coverage of the protein universe and the impact of the protein structure initiative. *Proc Natl Acad Sci U S A* 111:3733–3738
 11. Chothia C, Lesk AM (1986) The relation between the divergence of sequence and structure in proteins. *EMBO J* 5:823–826
 12. Illergård K, Ardell DH, Elofsson A (2009) Structure is three to ten times more conserved than sequence—a study of structural response in protein cores. *Proteins* 77:499–508
 13. Pieper U, Eswar N, Braberg H, Madhusudhan MS, Davis FP, Stuart AC, Mirkovic N, Rossi A, Marti-Renom MA, Fiser A, Webb B, Greenblatt D, Huang CC, Ferrin TE, Sali A (2004) Modbase, a database of annotated comparative protein structure models, and associated resources. *Nucleic Acids Res* 32: D217–D222
 14. Bienert S, Waterhouse A, de Beer TAP, Tauriello G, Studer G, Bordoli L, Schwede T (2017) The swiss-model repository-new features and functionality. *Nucleic Acids Res* 45: D313–D319
 15. Bowie JU, Lüthy R, Eisenberg D (1991) A method to identify protein sequences that fold into a known three-dimensional structure. *Science* 253(5016):164–170
 16. Buchan DWA, Jones DT (2017) Eigentreader: analogous protein fold recognition by efficient contact map threading. *Bioinformatics* (Oxford, England) 33:2684–2690
 17. Maldonado-Nava FG, Frausto-Solís J, Sánchez-Hernández JP, González Barbosa JJ, Liñán-García E (2018) Comparative study of computational strategies for protein structure prediction. In: Castillo O, Melin P, Kacprzyk J (eds) Fuzzy logic augmentation of neural and optimization algorithms: theoretical aspects and real applications, Studies in computational intelligence, vol 749. Springer, Cham
 18. Cavasotto CN, Phatak SS (2009) Homology modeling in drug discovery: current trends and applications. *Drug Discov Today* 14:676–683
 19. Schmidt T, Bergner A, Schwede T (2014) Modelling three-dimensional protein structures for applications in drug design. *Drug Discov Today* 19:890–897
 20. França TCC (2015) Homology modeling: an important tool for the drug discovery. *J Biomol Struct Dyn* 33:1780–1793
 21. Moult J, Fidelis K, Kryshtafovych A, Schwede T, Tramontano A (2018) Critical assessment of methods of protein structure prediction (CASP)-round XII. *Proteins* 86:7–15
 22. Shaw DE, Grossman J, Bank JA, Batson B, Butts JA, Chao JC, Deneroff MM, Dror RO, Even A, Fenton CH et al (2014) Anton 2: raising the bar for performance and programmability in a special-purpose molecular dynamics supercomputer. In: Proceedings of the international conference for high performance computing, networking, storage and analysis. IEEE Press, Piscataway, NJ, pp 41–53
 23. Bradley P, Misura KM, Baker D (2005) Toward high-resolution de novo structure prediction for small proteins. *Science* 309 (5742):1868–1871
 24. Moult J, Fidelis K, Kryshtafovych A, Schwede T, Tramontano A (2016) Critical assessment of methods of protein structure prediction: progress and new directions in round XI. *Proteins* 84:4–14
 25. Piana S, Klepeis JL, Shaw DE (2014) Assessing the accuracy of physical models used in protein-folding simulations: quantitative evidence from long molecular dynamics simulations. *Curr Opin Struct Biol* 24:98–105
 26. Pauling L, Corey RB (1951) The pleated sheet, a new layer configuration of polypeptide

- chains. Proc Natl Acad Sci U S A 37(5):251–256
27. Pauling L, Corey RB, Branson HR (1951) The structure of proteins; two hydrogen-bonded helical configurations of the polypeptide chain. Proc Natl Acad Sci U S A 37(4):205–211
 28. Venkatachalam CM (1968) Stereochemical criteria for polypeptides and proteins. v. conformation of a system of three linked peptide units. Biopolymers 6(10):1425–1436
 29. Richardson JS (1981) The anatomy and taxonomy of protein structure. Adv Protein Chem 34:167–339
 30. Jones TA, Thirup S (1986) Using known substructures in protein model building and crystallography. EMBO J 5(4):819–822
 31. Han KF, Baker D (1995) Recurring local sequence motifs in proteins. J Mol Biol 251(1):176–187
 32. Wu S, Skolnick J, Zhang Y (2007) Ab initio modeling of small proteins by iterative tasser simulations. BMC Biol 5:17
 33. Roy A, Kucukural A, Zhang Y (2010) I-tasser: a unified platform for automated protein structure and function prediction. Nat Protoc 5(4):725–738
 34. Zhang Y (2008) I-tasser server for protein 3d structure prediction. BMC Bioinformatics 9:40
 35. Rohl CA, Strauss CEM, Misura KMS, Baker D (2004) Protein structure prediction using rosetta. Methods Enzymol 383:66–93
 36. Xu D, Zhang Y (2012) Ab initio protein structure assembly using continuous structure fragments and optimized knowledge-based force field. Proteins 80(7):1715–1735
 37. Levitt M (1992) Accurate modeling of protein conformation by automatic segment matching. J Mol Biol 226(2):507–533
 38. Brunette TJ, Parmeggiani F, Huang PS, Bhabha G, Ekert DC, Tsutakawa SE, Hura GL, Tainer JA, Baker D (2015) Exploring the repeat protein universe through computational protein design. Nature 528:580–584
 39. Li W, Kinch LN, Karplus PA, Grishin NV (2015) Chseq: a database of chameleon sequences. Protein Sci 24:1075–1086
 40. Bonneau R, Baker D (2001) Ab initio protein structure prediction: progress and prospects. Annu Rev Biophys Biomol Struct 30:173–189
 41. Verschueren E, Vanhee P, van der Sloot AM, Serrano L, Rousseau F, Schymkowitz J (2011) Protein design with fragment databases. Curr Opin Struct Biol 21(4):452–459
 42. Pilla KB, Otting G, Huber T (2017) Protein structure determination by assembling supersecondary structure motifs using pseudocontact shifts. Structure (London, England) 1993(25):559–568
 43. Vallat B, Madrid-Aliste C, Fiser A (2015) Modularity of protein folds as a tool for template-free modeling of structures. PLoS Comput Biol 11:e1004419
 44. Fernandez-Fuentes N, Dybas JM, Fiser A (2010) Structural characteristics of novel protein folds. PLoS Comput Biol 6:e1000750
 45. Fernandez-Fuentes N, Fiser A (2006) Saturating representation of loop conformational fragments in structure databanks. BMC Struct Biol 6:15
 46. Koga N, Tatsumi-Koga R, Liu G, Xiao R, Acton TB, Montelione GT, Baker D (2012) Principles for designing ideal protein structures. Nature 491:222–227
 47. Handl J, Knowles J, Vernon R, Baker D, Lovell SC (2012) The dual role of fragments in fragment-assembly methods for de novo protein structure prediction. Proteins 80(2):490–504
 48. Baeten L, Reumers J, Tur V, Stricher F, Lenaerts T, Serrano L, Rousseau F, Schymkowitz J (2008) Reconstruction of protein backbones from the brix collection of canonical protein fragments. PLoS Comput Biol 4(5):e1000083
 49. Vanhee P, Verschueren E, Baeten L, Stricher F, Serrano L, Rousseau F, Schymkowitz J (2011) Brix: a database of protein building blocks for structural analysis, modeling and design. Nucleic Acids Res 39(Database issue):D435–D442
 50. Santos KB, Trevizani R, Custodio FL, Dardenne LE (2015) Profrager web server: fragment libraries generation for protein structure prediction. In: Proceedings of the international conference on Bioinformatics & Computational Biology (BIOCOMP). The Steering Committee of the World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), p 38
 51. Wang G, Dunbrack RL (2003) Pisces: a protein sequence culling server. Bioinformatics 19(12):1589–1591
 52. McGuffin LJ, Bryson K, Jones DT (2000) The psipred protein structure prediction server. Bioinformatics 16(4):404–405
 53. Frishman D, Argos P (1995) Knowledge-based protein secondary structure assignment. Proteins 23(4):566–579
 54. Charnes A, Cooper WW, Golany B, Seiford L, Stutz J (1985) Foundations of data envelopment analysis for pareto-koopmans efficient

- empirical production functions. *J Econ* 30 (1–2):91–107
55. Altschul SF, Madden TL, Schäffer AA, Zhang J, Zhang Z, Miller W, Lipman DJ (1997) Gapped blast and psi-blast: a new generation of protein database search programs. *Nucleic Acids Res* 25(17):3389–3402
56. Holmes JB, Tsai J (2004) Some fundamental aspects of building protein structures from fragment libraries. *Protein Sci* 13 (6):1636–1650
57. Trevizani R, Custódio FL, dos Santos KB, Dardenne LE (2017) Critical features of fragment libraries for protein structure prediction. *PLoS One* 12(1):e0170131
58. Kalev I, Habeck M (2011) Hhfrag: Hmm-based fragment detection using hhpred. *Bioinformatics* 27(22):3110–3116



Chapter 15

Molecular Dynamics Simulations of Conformational Conversions in Transformer Proteins

**Bernard S. Gerstman, Prem P. Chapagain, Jeevan GC,
and Timothy Steckmann**

Abstract

A relatively recently discovered class of proteins known as transformer proteins undergo large-scale conformational conversions that change their supersecondary structure. These structural transformations lead to different configurations that perform different functions. We describe computational methods using molecular dynamics simulations that allow the determination of the specific amino acids that facilitate the conformational transformations. These investigations provide guidance on the location and type of amino acid mutations that can either enhance or inhibit the structural transitions that allow transformer proteins to perform multiple functions.

Key words Transformer proteins, Molecular dynamics, Amyloid, Ebola, VP40

1 Introduction

Relatively recently, a new class of proteins has been labeled [1, 2] as “transformer proteins.” The common feature of transformer proteins is that they undergo large-scale changes in their secondary, supersecondary, and tertiary structure that lead to different, stable conformations. Further, each of these conformations performs different functions. We have used molecular dynamics computational simulations to investigate the conformational conversions in several different transformer proteins and have found common characteristics of the proteins that facilitate the supersecondary structural conversions.

Secondary and supersecondary structural conversions involving transitions from α -helical to β -sheet structure occur in a variety of proteins. This type of structural transformation is an important step leading to the ability of amyloid proteins to form toxic aggregates that are associated with Alzheimer’s, Parkinson, Huntington, and other neurodegenerative and prion diseases [3–12]. The relatively

small cc-beta peptides [13, 14] were engineered to display similar structural conversions at elevated temperatures. The alpha to beta conformational transformation may also have applications in developing interesting biomaterials [15].

There are other transformer proteins whose structural transition involves alpha to beta transformations [16–20]. The protein RfaH is an example [21, 22]. When the RfaH C-terminal domain (CTD) is in contact with the RfaH N-terminal domain, the CTD is in an α -helix conformation [23], and the RfaH acts as a transcription anti-terminator. When the CTD is not in contact with the NTD, the CTD transforms to a β conformation, and RfaH then facilitates translation.

We describe the computational materials and methods that are used to perform molecular dynamics simulations that allow the determination of the specific amino acids that facilitate the conformational transformations. These investigations provide guidance on the location and type of amino acid mutations that can either enhance or inhibit the structural transitions that allow transformer proteins to perform multiple functions.

2 Materials

Molecular dynamics (MD) computational investigations involve three stages:

1. The initial positions of the atoms in the protein and the bonding pattern among the atoms.
2. A molecular dynamics software package that incrementally updates (usually femtosecond timesteps) the position of the atoms. The MD software incorporates a “force field” algorithm that calculates the forces between nearby atoms in order to determine the acceleration experienced by each atom.
3. An analysis software package to analyze the results of the MD computational simulation. After each timestep, the MD software of Stage 2 updates the x , y , z position coordinate of each atom. Software is needed to visualize each update as a frame in a movie, as well as to calculate structural, thermodynamic, and statistical mechanics parameters such as bonded and non-bonded energies, distances between atoms, fluctuations in the positions of atoms, and other important information that provides insight into the dynamics of the system.

We now describe the steps that are necessary to perform each of these stages.

2.1 Initial Configuration of the Protein

The initial configuration (x , y , z coordinates) of the atoms in a protein can be downloaded from the Protein Data Bank (PDB) (www.rcsb.org). Currently, 140,000 structures are available. The term “structures” is especially important for transformer proteins for which the same amino acid sequence can have very different conformations. The steps for downloading the PDB coordinates for the atoms in a protein are the following:

1. Access the PDB website at www.rcsb.org.
2. In the search box, enter identifying information for the molecule of interest. This identifying information can be the name of the protein or the PDB ID code, which is usually given in a scientific publication investigating the molecule. For example, RfaH with the CTD in the alpha configuration has PDB-ID: 2OUG, and when the CTD is in the beta configuration, its configuration is listed with PDB-ID: 2LCL. The amyloid-like $\text{cc}\beta$ peptide that we discuss has PDB-ID: 1S9Z.
3. Download the coordinates into a file that can be used by the MD software.
4. The PDB file may have residues whose identity is known but their coordinates are not available. The software package Modeller [24, 25] should be used to add these missing residues.

2.2 Molecular Dynamics Pre-simulation Preparation

Once the coordinates for the atoms in the protein are downloaded, several steps must be carried out before a proper MD “production” run can be performed:

1. A box is created that the protein is placed in. The box should be large enough so that the protein is at least 10 Å away from the edge.
2. Solvation. Usually, only “crystallographic” water molecules that are located inside the protein are included in the PDB file. The MD simulation software will add water molecules around the protein. There are different ways for the MD software to perform the solvation; a common approach is to use the TIP3P (transferable intermolecular potential with 3 points) water model.
3. Neutralization. Occasionally, the initial configuration of the protein will have a net charge, which can lead to unphysical results. The VMD software package [26] can be used to add counter ions to make the system neutral.
4. Choice of Force Field. Equations (force fields) are used to calculate the interaction energies between atoms. Different force fields are available. The CHARMM36 force field [27] is especially popular for protein MD simulations.
5. Energy Minimization. The positions of the atoms given in the PDB are approximate positions. Therefore, some atoms may be

unrealistically close together or produce unrealistic bond angles. Using these unrealistic coordinates may cause the force field to calculate enormous forces that cause unphysical changes in the positions of atoms.

An energy minimization process allows atoms to make small movements from their initial positions to decrease their repulsions. A standard technique to perform energy minimization is to use the conjugate gradient and line search algorithm for 5 ps.

6. Heating. Most MD simulations are performed at room temperature, and the atoms are given random thermal velocities. In order to avoid large, unphysical motions of atoms, the system is slowly heated to room temperature. A standard approach for heating is to use a linear gradient of 20 K/ps from 20 to 300 K.
7. Equilibration. Even after energy minimization and gradual heating, the system may not be in equilibrium. For example, there may be hotspots and coldspots that are not found in an actual system. Equilibration allows time for the various parts of the system to exchange energy and equilibrate. An example of an equilibration process is that after the system is heated to 300 K, the system is equilibrated with a 2 ns NPT (constant number, constant pressure, constant temperature) run using a 1 fs integration timestep, followed by a 5 ns NVT (constant number, constant volume, constant temperature) run also using a 1 fs integration timestep.

3 Methods

Once the MD simulation is properly prepared, the “production” run can be performed in order to obtain information to determine which amino acids play important roles in the behavior of the protein. There are various methods that can be used to perform the production run.

3.1 Standard Molecular Dynamics Simulations

In the MD production run, various choices must be made to perform the simulation (*see Note 1*). A timestep of 2 fs is often chosen. If atoms are combined into a cluster (coarse grain) to allow the computations to run faster, a timestep of 15 fs can be used. A decision must be made on whether to represent water molecules explicitly or implicitly. Explicit water molecules are introduced through solvation (described above) and provide a detailed representation of the effects of water. However, the explicit water molecules require computational resources. To allow the simulation to run faster, implicit water can be used. If implicit water is chosen, the volume around the protein is given a uniform dielectric constant of water (~80). In addition, the viscosity of water is represented

through a Langevin dynamics damping constant, which is often chosen to be 1 ps^{-1} . The large number of hydrogen atoms in proteins requires substantial computational resources that can be reduced by the use of the RATTLE algorithm to constrain protein bonds involving hydrogen. The SETTLE algorithm is often employed to reduce computational time by constraining rigid water molecules. Multiple timestepping algorithms can also be employed to reduce computational time. The (1–2–4) algorithm calculates interaction through covalent bonds at every timestep, short range nonbonded interactions are calculated every other timestep, and long range electrostatic forces at every fourth step.

3.2 Replica Exchange Molecular Dynamics Simulations

Large-scale structural conversion in proteins of supersecondary structure may require milliseconds or longer to occur. This requires a large amount of computational resources. The structural changes may occur more quickly if the MD simulation is run at a high temperature, but the high temperature will likely cause the system to undergo many large-scale structural changes. Thus, a high-temperature MD simulation may not distinguish between stable structures and unstable structures (*see Note 2*).

MD computational simulations that allow the system to undergo large-scale structural changes in a reasonable computational time and then settle into stable configurations can be made feasible by the use of replica exchange molecular dynamics (REMD) [2, 28, 29]. In REMD simulations, multiple copies of the same initial configuration are created. Independent MD simulations are performed on the replicas. Periodically, the energies of the replicas are compared, and some replicas will have the proper conditions to exchange temperatures. This swapping of temperatures occurs throughout the REMD simulations and provides the possibility for all replicas to change their temperature multiple times. The changing of temperatures allows large-scale structural changes to occur in less simulation time because during stages when a replica is at high temperatures, it is able to escape from configurational kinetic traps and sample large regions of structural configuration space. However, this creates a problem in determining the relative stability of different configurations because at high temperature the system is unlikely to settle into any specific, low-energy configurations. The temperature swapping algorithm facilitates the swapping of a low-energy, stable configuration to switch to a lower temperature and thus increases its probability to remain in a low-energy configuration. It must be emphasized that, though the structural conversion pathways that are observed in REMD computational simulations may be physically realistic, the timescales are not. Usually, the time for the REMD structural transitions to occur is much shorter than realistic times.

To perform REMD, the following steps are employed:

1. Multiple replicas of the initial configuration of the system are created. The choice of the number of replicas can vary but is often in the range of 10–20.
2. Each replica is assigned a different temperature. The temperature range should span a region that includes a temperature that is low enough so that the protein is stable, to a temperature that is high enough so that the protein will quickly undergo forward and backward structural conversions. The spacing between temperatures is usually not constant. Instead the spacing between temperatures usually is chosen to increase exponentially as the temperature increases from one replica to the next.
3. Each replica is independently but simultaneously simulated by molecular dynamics for a short number of timesteps.
4. After a fixed number of MD timesteps (often chosen to be 500), a Metropolis-style algorithm is used to determine if replicas at adjacent temperatures should swap temperatures. The exchange probability is given by $P_{ij} = \min \{1, \exp(-\Delta)\}$ with $\Delta = (E_j - E_i) * [1/(kT_i) - 1/(kT_j)]$ and E_j the potential energy of replica j .
5. Based upon the results of the Metropolis test, some replicas will swap temperatures. Then, the same number of molecular dynamics steps is performed on all replicas, followed by another test for swapping adjacent temperatures.
6. In order for REMD to be effective in searching the configurational space of the protein, the acceptance rate for temperature exchange should be in the range of 20–30%. If the acceptance rate is below this range, additional replicas should be added so that the spacing between temperatures is smaller. If the acceptance rate is above this range, the number of replicas can be reduced so that the temperature spacing is larger.

3.3 Steered Molecular Dynamics Simulations

Steered molecular dynamics (SMD) simulations are a useful tool for determining the specific interactions and residues that are crucial in controlling large-scale structural changes in a protein (*see Note 3*). In SMD, one end of the protein is kept fixed and another part of the protein is pulled. Though this pulling does not occur in the actual protein, the pulling of the protein in the SMD simulations reveals which parts of the supersecondary structure change easily and which parts act as barriers to conformational rearrangement. Using the Jarzynski equality, free-energy differences between different molecular conformations can be calculated from the average work done due to pulling [22, 30]. In order to get a representative average, multiple pulling simulations must be performed. More

simulations will give a better estimate but also require more computational time. The steps for implementing SMD are the following:

1. Instead of a cubic box as used in normal MD simulations, the box must now be rectangular with the long axis of the box oriented along the pulling direction. The length of the long axis of the box is chosen so that there will be 10 Å of buffer between the stretched protein and the walls of the box.
2. The artificial pulling force is applied to a “dummy” atom that is attached via a virtual spring to an atom in the protein, while the other end of the protein is fixed in place. As an example, the dummy atom can be attached to the C_α atom of the C-terminal of the protein, while the C_α of the N-terminal is held fixed. Alternative atoms can be chosen if specific regions of the protein are of special interest.
3. A value must be chosen for the spring constant for the virtual spring. If a small value is chosen, the virtual spring will easily stretch with little stress applied to the protein. If the virtual spring is assigned a large value for the spring constant, the virtual spring will be rigid, and the pulling force will be transferred directly to the protein. In this situation, equilibration rearrangements of the protein that can occur naturally in response to stresses may not have sufficient time to occur, and the resulting structural rearrangements may not reflect the true structural dynamics that occur in the protein. A recommended, midrange spring constant is $k = 3 \text{ kcal/mol}/\text{\AA}^2$.
4. In addition to the strength of the spring constant, another characteristic of the pulling force that must be chosen is whether the pulling is done with constant force or with constant speed. The choice is often dependent on the type of experiments (e.g., atomic force microscopy) that have been performed on the molecule.
5. During SMD, the response of the protein to the pulling force will be more realistic if the protein is allowed to periodically relax by making internal conformational changes. This relaxation can be accomplished by alternating periods of pulling and with periods of no-pulling but with the end atoms remaining fixed. For example, 10 ns of SMD pulling can alternate with 10 ns of relaxation.

3.4 Targeted Molecular Dynamics Simulations

Targeted molecular dynamics (TMD) simulations are used to guide a structural transition to a known, target structure. This allows a structural transition to occur in a computationally feasible time. If the guiding force is relatively low ($\sim k_{\text{atom}} = k/N = 0.4 \text{ kcal/mol}/\text{\AA}^2$, where k is defined below), the conformational changes in the structural transition are reasonably realistic though occurring on a

much shorter timescale. The guiding force is implemented in the simulations by applying an artificial potential energy function:

$$U = \frac{k}{2N} [\text{rmsd}(t) - \text{RMSD}(t)]^2 \quad (1)$$

The value of $\text{rmsd}(t)$ is computed periodically by comparing the simulated structure's atomic coordinates to the final target structure by calculating the root-mean-square deviation (rmsd) of the heavy (non-hydrogen) atoms. The value of $\text{RMSD}(t)$ is the tolerance, i.e., if a rough approximation of the target structure is sufficient, then RMSD is set to a large value. If it is desired to get a simulated structure that is exactly like the target, then RMSD is set to zero. In practice, $\text{RMSD}(t=0)$ is set to a large value and is set up to decrease linearly with time to a small final value.

4 Results

4.1 REMD Investigations of Amyloid-Like Structural Conversion

We investigated [31] the alpha to beta structural transition using the amyloid-like synthetic $\text{cc}\beta$ peptide. To speed up the structural transition, we used REMD computational simulation. Our used 20 replicas and the simulations lasted for 300 ns. The range of temperatures for the 20 replicas was 380–600 K, and the temperatures were spaced exponentially. We performed the test for exchanging temperatures among the replicas at intervals of 500 MD steps. With the range and spacing of temperatures that we used, swapping of temperatures between replicas occurred with a 24% success rate.

At the higher temperatures, α -helices that were originally placed in trimers unfolded quickly and aggregated in β -supersecondary structures. Because of the replica exchange, structural conversions that might require microseconds or milliseconds at room temperatures occurred much faster, and all 20 replicas had reached stable structures after 300 ns. This is shown in Fig. 1 which shows that the root-mean-square deviations in the positions of the atoms in all replicas have reached asymptotic values by 300 ns.

The use of REMD facilitated the alpha to beta structural conversion to occur fast enough to observe. A structural conversion is shown in Fig. 2 which displays frames from a REMD replica that converted to an amyloid-like beta-sheet supersecondary structure with two beta sheets. The two beta sheets display a twist that may allow them to form amyloid-like fibrils that are implicated in neurodegenerative diseases.

Though the timescales in our REMD simulations are not realistic, the stages in the structural conversion may be. Thus, we monitored important molecular aspects during the structural

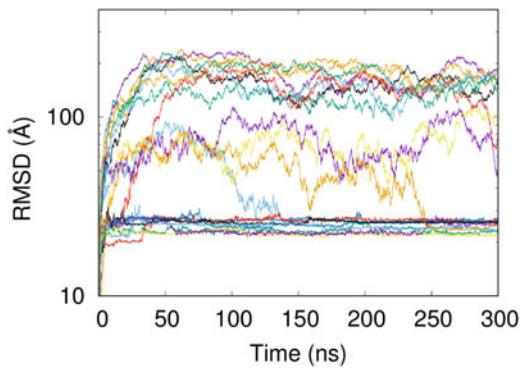


Fig. 1 (Reproduced from ref. 31 with the permission of the American Institute of Physics Publishing) The root-mean-square deviations (RMSD) of all 20 replicas compared to the initial structure. REMD speeds up the structural conversion so that all replicas have settled into their final configuration after 250 ns

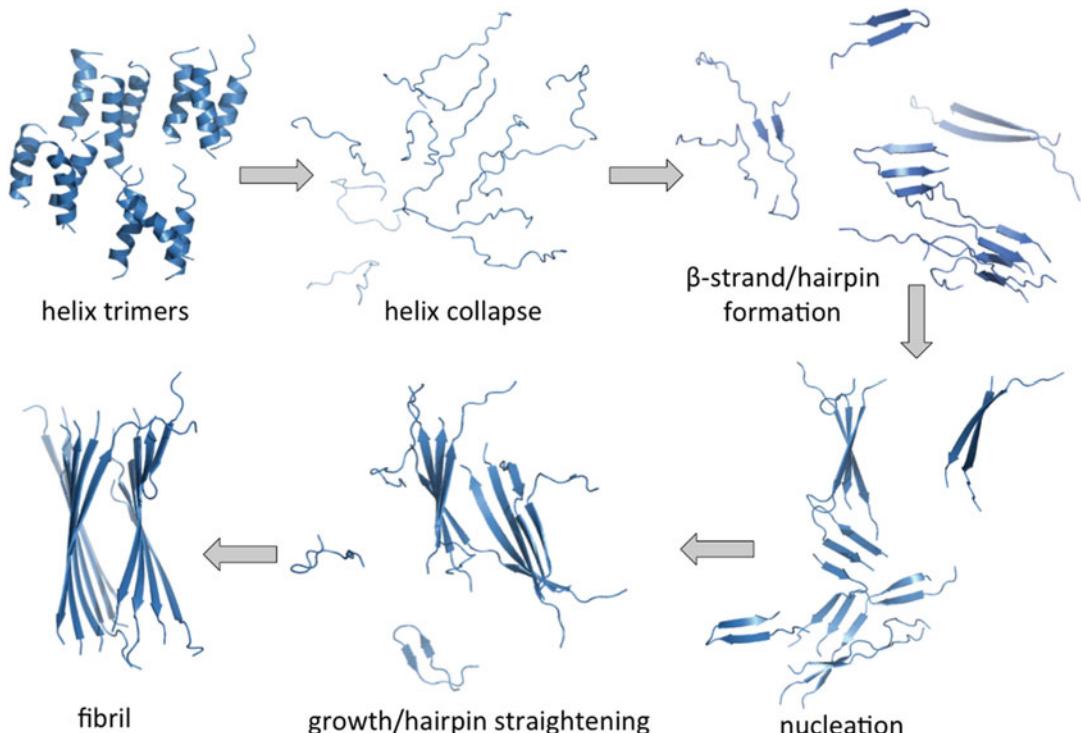


Fig. 2 (Reproduced from ref. 31 with the permission of the American Institute of Physics Publishing) Frames from the MD simulations showing steps in the helix structural transition to β -strands and the formation of β -sheets

conversion. An especially significant parameter is the number and type of hydrogen bonds. The initial, all-alpha configuration contains a large number of intrachain hydrogen bonds, whereas the all-beta configuration contains a large number of interchain

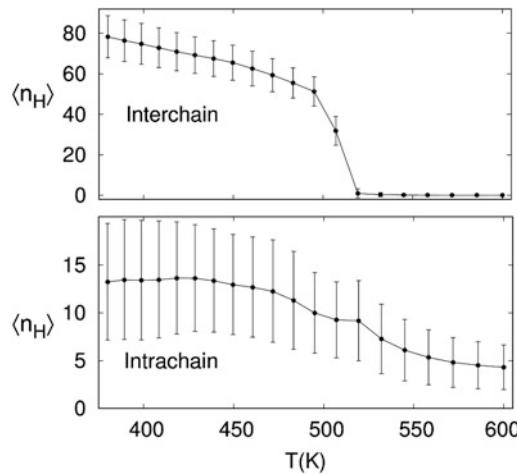


Fig. 3 (Reproduced from ref. 31 with the permission of the American Institute of Physics Publishing) Average number of hydrogen bonds as a function of temperature. Top: interchain H-bonds, bottom: intrachain H-bonds

hydrogen bonds. Figure 3 shows how the number of each type of hydrogen bonds varies with temperature. We find that there is a small temperature window between 450 and 500 K that is especially significant. In this temperature window, starting at 450 K, there is a drop in the number of intrachain bonds as found in alpha structure. In contrast, the number of interchain hydrogen bonds does not start to drop until 500 K. Thus, between 450 and 500 K, the alpha structure is unstable, but the beta structure is stable. Above 500 K, there is a sharp drop in the number of interchain hydrogen bonds and therefore the beta-sheet secondary structure falls apart. Further information from the REMD relevant to the alpha to beta structural transition, along with insight into cooperativity in the process, can be found in ref. 31.

4.2 RfaH Structural Conversion

Another protein that undergoes large-scale supersecondary structural transitions is the transformer protein RfaH. To investigate the molecular details of the RfaH CTD's structural conversion [22] from all-alpha to all-beta, we used both TMD and SMD computational simulations [32, 33]. The targeted molecular dynamics and steered molecular dynamics simulations were especially helpful in determining the amino acid residues at the interface of the CTD and NTD that inhibit the structural conversion. Breaking of these important interfacial bonds allows the CTD structural conversion to occur relatively quickly.

For the TMD simulations, after downloading the RfaH protein molecule, the system was solvated and electrically neutralized using VMD. Long-range, nonbonded interactions were treated using the

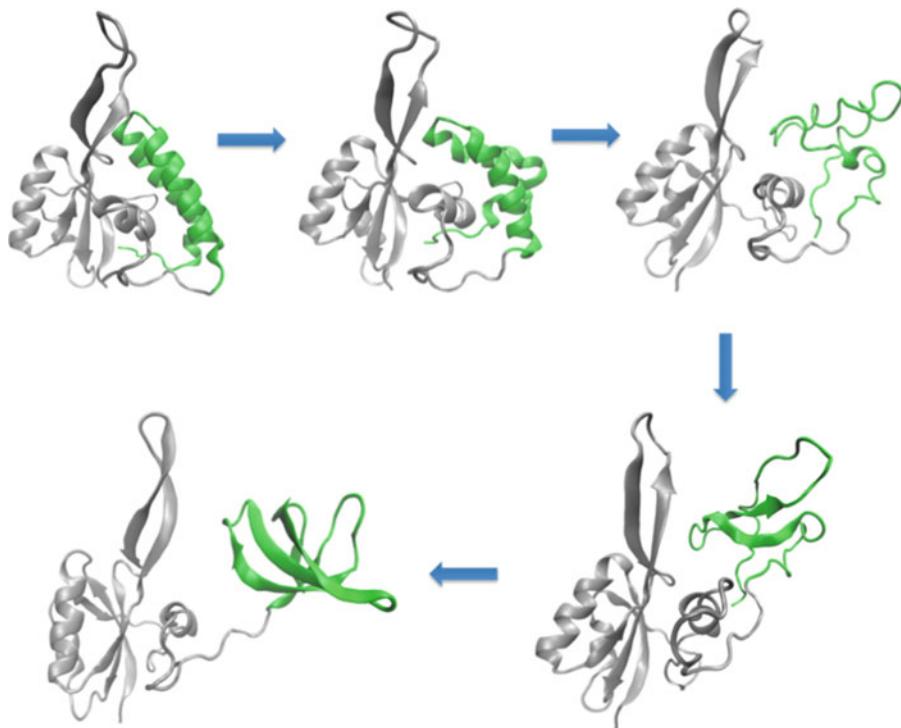


Fig. 4 (Reproduced from ref. 22 with permission from the American Chemical Society) Frames from a targeted molecular dynamics simulation showing the CTD (green) transforming from all-alpha to all-beta structure after the CTD separates from the NTD (gray)

Ewald method [34] with a 12 Å nonbonded cutoff. The NAMD software package using the CHARMM36 force field was used to run the computational simulations. The initial configuration had the CTD in the all-alpha configuration. A typical α -helix to β -barrel structural transformation process is shown in Fig. 4 which displays frames of the structural transition of the CTD from all-alpha to all-beta. As explained above, the target $\text{RMSD}(t)$ in Eq. 1 is decreased throughout the simulation toward a final, small value. We found that during our TMD simulation, the alpha-helical structure disappears when the $\text{RMSD}(t)$ target has been reduced by approximately half of its initial value. The final, all-beta structure had an rmsd of 0.94 Å rmsd compared to the experimental structure [21].

An important finding [22] was that the CTD does not begin to fold into the all-beta target structure until all of the interdomain interactions with the NTD are broken, except for the peptide bonds along the linker segment. The salt-bridge between E48 and R138 was found to be especially important. In Fig. 5a, we plot the native contacts Q as a function of time. Fig. 5a includes plots for CTD intradomain contacts for both alpha-CTD and beta-CTD, as well as a plot of the NTD-CTD interdomain contacts. There is a sharp

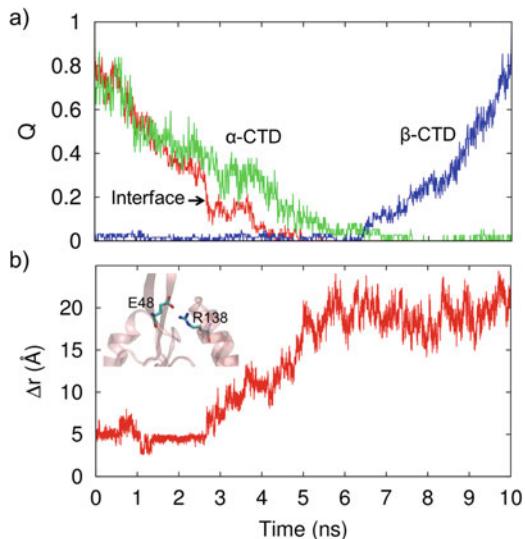


Fig. 5 (Reproduced from ref. 22 with permission from the American Chemical Society) **(a)** Time dependence of native contacts Q during the TMD simulation: NTD-CTD interface (red), intradomain for alpha-CTD (green), and beta-CTD (blue). **(b)** Interdomain salt-bridge separation distance (Δr) between the E48 oxygen and the R138 nitrogen

drop in the interdomain Q at 3 ns and again at 4 ns when the E48-R138 salt-bridge breaks. The importance of the breaking of the salt-bridge is displayed in Fig. 5b which shows that one of the last interdomain bonds that is broken before the structural conversion is the E48-R138 salt-bridge.

This all-alpha to all-beta conversion of the RfaH CTD was explored further using steered molecular dynamics simulations. The SMD atom was the C_α atom at the C-terminal (residue L162). The dummy atom was attached to this atom with a virtual spring of $k = 3$ kcal/mol/Å². The N-terminal's (residue M1) C_α was kept fixed. The dummy atom was pulled at a constant speed of 2 Å/ns. The magnitude of the steering force necessary to maintain the constant speed was recorded at 1 ps intervals. The SMD was performed by alternating pulling and relaxation at 10 ns intervals. The extension of the protein was maintained during each 10 ns relaxation interval by fixing the first C_α in the NTD and the final C_α in the CTD but allowing free motion of all other atoms. The Jarzynski equality was used to calculate the potential of mean force (PMF) by averaging the work of nine separate SMD trajectories. Figure 6 shows the resulting free-energy profile as a function of the distance between the C_α of the first and last amino acids (M1-L162). Amino acids 158–162 in the CTD tail segment in the initial state (A) undergo a rearrangement into an especially stable alpha structure (B). Next, CTD tail segment interactions with the

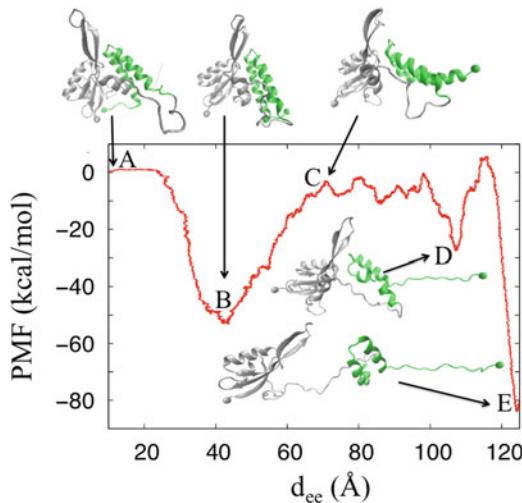


Fig. 6 (Reproduced from ref. 22 with permission from the American Chemical Society) The potential of mean force as a function of the distance between the first amino acid (M1) and last amino acid L162 (C–C) for the RfaH protein

NTD break as the system transitions from B to structure C. Then, the CTD with its alpha-helical structure mostly intact separates from the NTD but maintains the strong E48-R138 interdomain salt-bridge (state D). Breaking of this strong salt-bridge presents a large barrier between D and E. Only after this strong salt-bridge is broken can the CTD and NTD fully separate into state E.

5 Notes

1. Molecular dynamics computational simulations can provide wonderful insight into the molecular and atomic level details of the structural dynamics and functioning of biomolecules. The continuous increases in computer power make MD simulations increasingly more useful [26, 27]. However, there are a large number of parameters that must be carefully regulated in order for MD simulations to be physically realistic.
2. Many molecular events occur on timescales that are too long for conventional MD simulations to investigate. Fortunately, there are specialized MD techniques that will speed up structural transitions such as replica exchange [2, 28, 29], steered, and targeted, to name a few. The specialized MD techniques may allow large-scale structural transitions to occur on timescales that are short enough to observe in the computational simulations. It must be remembered that the timescales using these specialized MD techniques for the structural transitions are not physically realistic. However, the stages in the structural

conversion may be realistic if the simulations are run properly. An important check is to rerun the simulations multiple times. If the structural conversions occur differently, then the results require careful interpretation before any claims are made.

3. Properly performed MD simulations can be used to obtain information that is not immediately apparent. For example, multiple steered molecular dynamics simulations can be used with the Jarzynski equality [22, 30] to produce a potential energy diagram of various structural configurations.

References

1. Knauer SH, Artsimovitch I, Rösch P (2012) Transformer proteins. *Cell Cycle* 11:4289–4290
2. GC JB, Bhandari YR, Gerstman BS, Chapagain PP (2014) Molecular dynamics investigations of the α -helix to β -barrel conformational transformation in the RfaH transcription factor. *J Phys Chem B* 118:5101–5108
3. Zhou M, Ottenberg G, Sferrazza GF, Lasmezas CI (2012) Highly neurotoxic monomeric alpha-helical prion protein. *Proc Natl Acad Sci U S A* 109:3113–3118
4. Eisenberg D, Jucker M (2012) The amyloid state of proteins in human diseases. *Cell* 148:1188–1203
5. Straub JE, Thirumalai D (2011) Toward a molecular theory of early and late events in monomer to amyloid fibril formation. *Annu Rev Phys Chem* 62:437–463
6. Brundin P, Melki R, Kopito R (2010) Prion-like transmission of protein aggregates in neurodegenerative diseases. *Nat Rev Mol Cell Biol* 11:301–307
7. DeMarco ML, Daggett V (2004) From conversion to aggregation: protofibril formation of the prion protein. *Proc Natl Acad Sci U S A* 101:2293–2298
8. Diaz-Espinoza R, Soto C (2012) High-resolution structure of infectious prion protein: the final frontier. *Nat Struct Mol Biol* 19:370–377
9. Huang L, Jin R, Li J, Luo K, Huang T, Wu D, Wang W, Chen R, Xiao G (2010) Macromolecular crowding converts the human recombinant PrPC to the soluble neurotoxic β -oligomers. *FASEB J* 24:3536–3543
10. Sang JC, Lee CY, Luh FY, Huang YW, Chiang YW, Chen RP (2012) Slow spontaneous alpha-to-beta structural conversion in a non-denaturing neutral condition reveals the intrinsically disordered property of the disulfide-reduced recombinant mouse prion protein. *Prion* 6:489–497
11. Khandogin J, Brooks CL 3rd (2007) Linking folding with aggregation in Alzheimer's beta-amyloid peptides. *Proc Natl Acad Sci U S A* 104:16880–16885
12. Steckmann T, Awan Z, Gerstman BS, Chapagain PP (2012) Kinetics of peptide secondary structure conversion during amyloid beta-protein fibrillogenesis. *J Theor Biol* 301:95–102
13. Kammerer RA, Kostrewa D, Zurdo J, Detken A, Garcia-Echeverria C, Green JD, Muller SA, Meier BH, Winkler FK, Dobson CM et al (2004) Exploring amyloid formation by a de novo design. *Proc Natl Acad Sci U S A* 101:4435–4440
14. Steinmetz MO, Gattin Z, Verel R, Ciani B, Stromer T, Green JM, Tittmann P, Schulze-Briese C, Gross H, van Gunsteren WF et al (2008) Atomic models of de novo designed cc beta-Met amyloid-like fibrils. *J Mol Biol* 376:898–912
15. Woolfson DN, Ryadnov MG (2006) Peptide-based fibrous biomaterials: some things old, new and borrowed. *Curr Opin Chem Biol* 10:559–567
16. Ding F, Borreguero JM, Buldyrey SV, Stanley HE, Dokholyan NV (2003) mechanism for the alpha-helix to beta-hairpin transition. *Proteins* 53:220–228
17. Hansen MB, Ruizendaal L, Löwik DWPM, van Hest JCM (2009) Switchable peptides. *Drug Discov Today Technol* 6:e33–e39
18. Qin Z, Buehler MJ (2010) Molecular dynamics simulation of the α -helix to β -sheet transition in coiled protein filaments: evidence for a critical filament length scale. *Phys Rev Lett* 104:198304
19. Wang X, Bergenfeld I, Arora PS, Canary JW (2012) Reversible redox reconfiguration of

- secondary structures in a designed peptide. *Angew Chem Int Ed Eng* 51:2099–13101
20. Yoon S, Welsh WJ (2005) Rapid assessment of contact-dependent secondary structure propensity: relevance to amyloidogenic sequences. *Proteins* 60:110–117
21. Burmann BM, Knauer SH, Sevostyanova A, Schweimer K, Mooney RA, Landick R, Artsimovitch I, Rosch P (2012) An alpha helix to beta-barrel domain switch transforms the transcription factor RfaH into a translation factor. *Cell* 150:291–303
22. GC JB, Gerstman BS, Chapagain PP (2015) The role of the interdomain interactions on RfaH dynamics and conformational transformation. *J Phys Chem B* 119(40):12750–12759
23. Svetlov V, Nudler E (2012) Unfolding the bridge between transcription and translation. *Cell* 150:243–245
24. Eswar N, Webb B, Marti-Renom MA, Madhusudhan MS, Eramian D, Shen MY, Pieper U, Sali A (2006) Comparative protein structure modeling using Modeller. *Curr Protoc Bioinformatics Chapter 5:Unit 5.6*
25. Sanchez R, Sali A (2000) Comparative protein structure modeling. Introduction and practical examples with modeller. *Methods Mol Biol* 143:97–129
26. Humphrey W, Dalke A, Schulten K (1996) VMD: visual molecular dynamics. *J Mol Graph* 14:33–38
27. Vanommeslaeghe K, Hatcher E, Acharya C, Kundu S, Zhong S, Shim J, Darian E, Guvench O, Lopes P, Vorobyov I, Mackerell AD Jr (2010) CHARMM general force field: a force field for drug-like molecules compatible with the CHARMM all-atom additive biological force fields. *J Comput Chem* 31:671–690
28. Rao F, Cafisch A (2003) Replica exchange molecular dynamics simulations of reversible folding. *J Chem Phys* 119(7):4035–4042
29. Zhang W, Wu C, Duan Y (2005) Convergence of replica exchange molecular dynamics. *J Chem Phys* 123(15):154105
30. Martin HS, Jha S, Coveney PV (2014) Comparative analysis of nucleotide translocation through protein nanopores using steered molecular dynamics and an adaptive biasing force. *J Comput Chem* 35:692–702
31. Steckmann T, Bhandari YR, Chapagain PP, Gerstman BS (2017) Cooperative structural transitions in amyloid-like aggregation. *J Chem Phys* 146:135103
32. Isralewitz B, Gao M, Schulten K (2001) Steered molecular dynamics and mechanical functions of proteins. *Curr Opin Struct Biol* 11:224–230
33. Phillips JC, Braun R, Wang W, Gumbart J, Tajkhorshid E, Villa E, Chipot C, Skeel RD, Kale L, Schulten K (2005) Scalable molecular dynamics with NAMD. *J Comput Chem* 26:1781–1802
34. Essmann U, Perera L, Berkowitz ML, Darden T, Lee H, Pedersen LG (1995) A smooth particle mesh Ewald method. *J Chem Phys* 103:8577–8593



Chapter 16

Sequence Pattern for Supersecondary Structure of Sandwich-Like Proteins

Alexander E. Kister

Abstract

The goal is to define sequence characteristics of beta-sandwich proteins that are unique for the beta-sandwich supersecondary structure (SSS). Finding of the conserved residues that are critical for protein structure can often be accomplished with homology methods, but these methods are not always adequate as residues with similar structural role do not always occupy the same position as determined by sequence alignment. In this paper, we show how to identify residues that play the same structural role in the different proteins of the same SSS, even when these residue positions cannot be aligned with sequence alignment methods. The SSS characteristics are (a) a set of positions in each strand that are involved in the formation of a hydrophobic core, residue content, and correlations of residues at these key positions, (b) maximum allowable number of “low-frequency residues” for each strand, (c) minimum allowed number of “high-frequency” residues for each loop, and (d) minimum and maximum lengths of each loop. These sequence characteristics are referred to as “sequence pattern” for their respective SSS. The high specificity and sensitivity for a particular SSS are confirmed by applying this pattern to all protein structures in the SCOP data bank. We present here the pattern for one of the most common SSS of beta-sandwich proteins.

Key words Supersecondary structure, Secondary structure, Supersecondary structure prediction, Sequence analysis, Immunoglobulin fold, Sequence-structure relationship, Sequence alignment, Structure comparison

1 Introduction

1.1 Strengths and Weaknesses of the Existing Approaches to Predicting Structure from Sequence

Understanding the relationship between amino acid sequences and protein structure is critical for predicting the 3D structure and for protein design. In theory, a very detailed calculation of thousands of strong and weak interactions between amino acids answers the question of how a sequence of amino acids folds into a stable three-dimensional structure [1–3]. The advantage of this ab initio approach is that it only takes into account the physicochemical properties of residues in a sequence and does not require prior knowledge of known structures. The disadvantage is the necessity of carrying out energy calculations to a high degree of accuracy as folded conformation is generally 5–10 kcal/mol more stable than

the unfolded chain [4]. Currently, the application of ab initio structure modeling is limited, because of the computational difficulties, especially for proteins that are more than 60–70 residues long [5–7].

A more practical and widely used approach to this problem is homology-based sequence alignment. The challenge of homology-based methods is to align identical or similar residues in respective columns [8, 9]. Sequence alignments are generally successful for predicting the 3D structure if at least one-third of residues in the query sequence have identical and similar residues with proteins with known structures [10]. This observation implies that only a minority of residues are essential for protein folding. But it is unlikely that the 70% of nonmatching residues are inconsequential for structural stability and may be replaced by other residues at random. It is assumed that existing proteins underwent extensive natural selection and that the “nonessential” residues were selected because they have a supportive role to play in the formation of the specific 3D fold. If this is the case, then variability among supportive residues would also be limited, albeit not as much as the variability among the essential residues, and it should be possible to identify rules that restrict repertoire of these supportive, non-conserved residues. The discovery of such rules for the supportive residues requires an approach different from homology alignment. Another problem with sequence alignment approach is that it may lead to erroneous conclusions about roles of residues in structures, especially if secondary structural data are excluded from the analysis. For example, residues in the same (but erroneous aligned) column may, in fact, belong to different strands or loops.

An additional argument for developing alternatives to homology-based approach is that there exist a number of exceptions to the “one-third sequence similarity” criteria. Proteins with high sequence identity may have very different folds. One example is of two proteins with 40% sequence identity, one of which has a mixed $\alpha + \beta$ structure and the other—a completely different all α -helical structure [11]. An even more surprising example is of two small proteins that have 95% sequence identity (53/56 residues in common), yet their three nonidentical residues are responsible for a dramatic shift from 3α fold to α/β fold [12]. Conversely, proteins with similar structures may have highly dissimilar sequences. For example, anaerobic cobaltochelatase and ferrochelatase have similar structures, but only 11% sequence identity [13]. These examples make it clear that methods based on sequence homology are not always adequate for investigating sequence-structure relationship.

1.2 Developing a Non-homology-Based Approach to Predicting Structure from Sequence: Sequence Patterns

To compensate for some of the drawbacks of homology methods, we develop an alternative approach that aims to identify the sequence and structural characteristics that are sensitive and specific to protein domains with the same SSS. This set of characteristics is referred to as “sequence pattern” of the respective SSS. By definition, “identical SSS” of domains have (a) the same secondary structure—identical order of the secondary structural elements (beta strands and helices) in sequence—and (b) the same mutual arrangement of the secondary structural elements in 3D. SSS rather than 3D atomic structure is chosen as the criterion of structural similarity because the widely used structural alignment methods often yield divergent results even for relatively similar proteins [14–16]. The criteria of SSS identity are less ambiguous than of atomic 3D structure similarity. SSS could be regarded as a “protein skeleton” as opposed to “full-bodied” atomic three-dimensional protein structure. It allows us to focus on the structural features specific to a particular fold and exclude from consideration residues that are mainly responsible for functional properties.

Determining the sequence pattern (SP) of SSS first takes into account the most important requirement of protein structure stability—the presence of a hydrophobic core in the interior of a structure [17–20]. Since the core is formed by interactions between the (mostly) hydrophobic residues within the strands and helices, the positions of these residues need to be relatively fixed in secondary structure elements. The residue content of these positions and distances between these positions in sequences are the most important distinguishing characteristics of SP for the respective SSS. To determine the residue content at these positions requires an approach to alignment different from the one based on the similarity of residues.

The method of alignment of residues in strands proposed here is based on the analysis of hydrogen bond contacts between the strands in the beta sheets. These contacts may be represented as a network of H-bonds. It is assumed that beta proteins with the same SSS have a similar network of hydrogen bonds between strands in every beta sheet (Fig. 1). An alignment of residues in strands may be conceptualized as a superposition of H-bond networks of beta sheets. The residues which are superimposed on each other are assigned the same position in the strand. The contents of residues at respective positions that make up the hydrophobic core—i.e., whose side chains are directed to an interior of the protein domain—can then be analyzed.

In addition to specifying the location of the residues that make up the hydrophobic core, the SP includes information about the “unfavorable residues” for strands and loops. These unfavorable residues are determined from the statistical analysis of residue frequencies in the strands and loops of protein structures. SP characteristics specify the maximum number of unfavorable residues

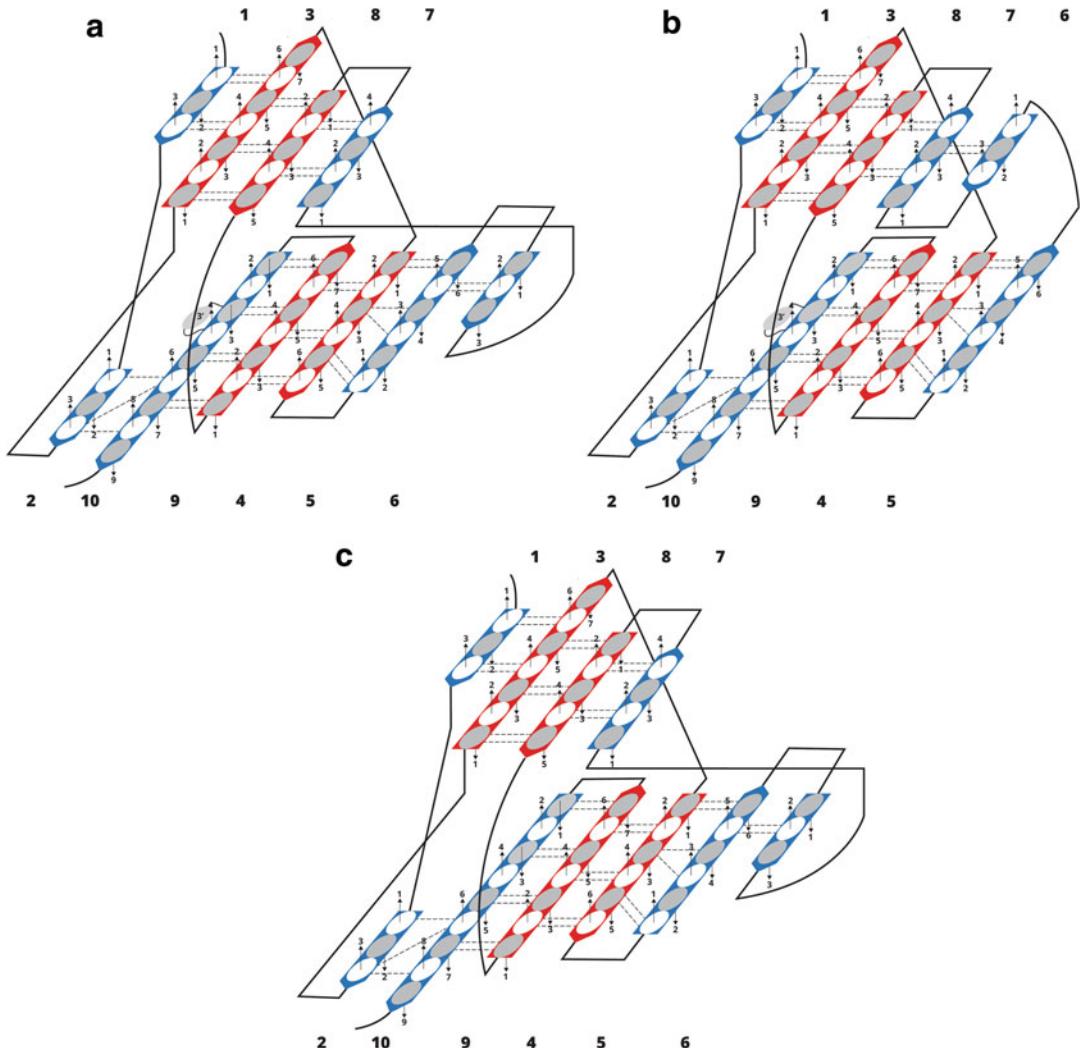


Fig. 1 Supersecondary structures of sandwich-like proteins with a specific “interlock” arrangement of strands 3, 4, 8, and 9. Beta strands are represented by arrows and protein loops are shown as lines. The interlock strands are shown in red (see Note 1). In this model, it is assumed that side chains of residues at “gray” positions 1, 3, and 5 in strands are directed inside (“hydrophobic positions”) (see Note 2), whereas residues at “white” positions 2, 4, and 6 are on the surface (“hydrophilic positions”). Variants (a, b) differ in the location of the edge strand #6 in different sheets; variant (c) strand #10 has no a bulge (see Subheading 3.2)

that can be present in particular strands and loops of SSS. SP also specifies allowable loop lengths between strands. Herein we present SP for a group of immunoglobulin proteins with sandwich-like beta sheets shown in Fig. 1.

2 Methods

2.1 Overview of an Algorithm for Determining Sequence Pattern

The process of defining SP for any SSS involves the following steps:

1. Identifying all known proteins with the given SSS.
2. Selecting a small subset of proteins with the same SSS which share the least sequence similarity. They will be referred to as “target proteins.” The number of target proteins depends on the degree of sequence variability among proteins with the given SSS.
3. The sequence and structural analyses are performed for the target proteins to determine a preliminary set of SP characteristics.
4. All proteins with respective SSS are tested to determine if they comply with SP requirements. If any protein does not satisfy the SP conditions, then this protein is added to the group of the target proteins, and **steps 2 and 3** are repeated. The iterative steps continue until such an SP is found which is present in all proteins with the respective SSS.

After SP is defined, its sensitivity and specificity are determined by testing it on all known protein structures from different structural classes. If at least one false-positive or false-negative protein is detected, the characteristics of the SP are modified, or additional characteristics are developed, in order to improve the specificity and sensitivity of the sequence template.

2.2 Selecting Proteins with Identical SSS

The selection is carried out in four stages:

1. Protein structures with the same SSS are identified based on the structural classification of PDB, SCOP, and SCOPe databases [21–23].
2. Proteins with identical SSS are selected in accordance with PDBsum topology diagrams [24], which show how beta strands form beta sheets.
3. “Target proteins” are then selected. Target proteins are representative structures from different superfamilies and families with the same SSS. Target proteins are chosen for their sequence dissimilarity in order to optimize chances of determining SP that applies all proteins with the same SSS.
4. SSS for the target proteins is verified using SPACE server, which calculates residue-residue contact map [25].

2.3 Determining SP of Beta Proteins with Identical SSS

2.3.1 Alignment of Positions in Strands of Target Proteins

Our goal is to align positions in strands in such a way that residues in the same position have the same structural role in their respective proteins. The alignment is based mainly on the analysis of two structural characteristics of residues: (1) the residue-residue H-bond contacts between neighboring strands in a beta sheet (these interactions are critical for secondary structure formation) and (2) the contacts with residues from other beta sheets. Side chains of the residues at inward-oriented positions participate in a hydrophobic core formation; these interactions are critical for SSS formation.

H-bond contacts in a beta sheet may be represented as a network of H-bonds (Fig. 1). These H-bond networks are superimposed on each other in order to make the structural alignment of target proteins. A criterion of correct structural alignment is that it yields the same H-bond residue-residue contacts in the beta sheet and the same inward-oriented contacts for residues at aligned positions.

The network of H-bonds is considered in toto. Therefore, the superposition of one strand determines the superposition of other strands in the beta sheet. It follows that the variations in the superposition of the H-bond networks are determined solely by shifting one strand in a beta sheet. Thus, in order to find the best alignment of H-bond networks for two structures, it is necessary to verify only 2–3 variants of superposition. As a result of multi-alignment of the H-bond networks, same positions in strands are occupied by residues with essentially same structural characteristics in all target proteins but not always similar residues.

In some cases, the superposition of H-bond networks from different structures does not result in alignment of all positions in the strands. The differences among networks are due mainly to differences in lengths of strands, which affect the number of inter-strand H-bonds.

2.3.2 Definition of “Residue Groups”

To determine residue content of structurally aligned positions, amino acids are divided into several groups:

1. The group of hydrophobic residues: Ile, Val, Leu, Phe, Met, and Trp (“the h group”). Numerous studies have shown that these residues have a propensity for beta-strand formation [26–28]. If the position in the target proteins is occupied by one or only a few residues from the h group, it is assumed that all residues of this group can occupy this position in the proteins with the same SSS.
2. Three other residues—Ala, Tyr, and Cys—may be added to the h group because they often substitute for hydrophobic residues in strands. The weakly hydrophobic residue Ala is considered as both “interior-seeking” and “water-seeking” residue according to many hydrophobicity scales [29]. Polar residue Tyr and

hydrophilic residue Cys may play a role of a hydrophobic residue in specific protein microenvironments [30]. If the residue Ala is found alongside with h group residues at a given position, then this position is coded in SP as “h+A group” position. Similarly, the h+AYC group in SP describes the position occupied by any hydrophobic or partially hydrophobic residues.

3. Residues Ser and Thr were assigned to the “ST group.”
4. The residues Pro, Gly, His, Asn, Glu, and Asp are grouped together based on the observation that they are rarely found within strands (“not strand favorable”—NotFav group).
5. The residues Pro, Gly, Asn, Asp, Glu, Ser, Thr, and Ala are “loop favorable” residues—“loop group.” The characteristics are based on the statistical analysis of the frequency of residues occurring in coil conformations in numerous proteins of different classes. We introduce the semiquantitative concept of “loop score” based on residue frequency in loops. Residues Pro and Gly are the most common residues in loops and are assigned a loop score of “2”; the other residues in the loop group are assigned a loop score of “1.” Loop scores are used to formulate rules that relate to residue content in loops in proteins with the given SSS.

2.3.3 Sequence-Structural Characteristics in SP

For purposes of defining SP, each strand is characterized by the following parameters:

1. The minimum number of inward-oriented positions in a strand occupied by hydrophobic residues from “h” and “h+AYC” groups.
2. The maximum and the minimum number of hydrophobic residues from the “h” and “h+AYC” groups within a strand.
3. The maximum number of allowed not-strand-favorable residues.

In contrast to regular, spiral-shape geometry of strands, the main-chain conformation of loops is highly variable, and lengths of loops between the same two strands differ considerably across structures. Many residues in loops have functional properties involving an interaction with other molecules. Therefore, the sequence and structural alignment method applicable to strands does not apply to loops. Rather than determining residue content in the various positions of loops, we specify three “global” characteristics for every loop based on the analysis of the target proteins:

1. The minimum and the maximum number of residues in a loop.
2. The minimum number of “loop favorable” residues from the “loop group.” The number varies with the length of the loop.

3. The maximum allowed number of the hydrophobic residues from the “h” group. The number varies with the length of the loop.

All these sequence-structural characteristics of strands and loops, which are determined from the analysis of the target proteins, comprise the preliminary SP.

2.3.4 Testing for Sensitivity and Specificity of SP

After the SP has been determined using a subset of target proteins, the next step is to test all other known protein structures with the same SSS to determine the sensitivity of SP for the respective SSS. The procedure starts with a search for the location of strand #1 in the query sequence. The first position of strand #1 is initially selected such that the N-terminus fragment, which precedes the strand, has a minimum length defined in SP. Then all sequence-structural characteristics that are imposed on strand #1 in SP are tested one by one for the sequence fragment that is presumed to correspond to strand #1 in the query sequence. If any condition for presumed strand #1 is violated, then the starting position of the strand is moved one over, and the above testing procedure starts again. The procedure is repeated iteratively until the maximal length of the N-terminus fragment will be reached.

If all conditions for strand #1 are satisfied, then the next step is to test the conditions from the SP for the N-terminus fragment that precedes the strand #1 in the sequence. If the maximal length of the N-terminus fragment is reached and every condition for the strand or for the preceding fragment is not satisfied, then SP for the given SSS needs to be modified. This query protein is then added to the list of the target proteins, and a modified SP is tested for the expanded target protein set. Then the modified SP is tested on all known protein structures with the given SSS. The procedure is repeated until all proteins with this SSS satisfy SP conditions for the first strand and the preceding N-terminus fragment.

Once all strand #1 and N-terminus fragment conditions are satisfied, the iterative procedure is repeated for strand #2 and for the loop between strand #1 and #2. If SP conditions for strand #2 and the intervening loop are satisfied, the algorithm moves on to strand #3 and the intervening loop between strands #2 and #3 and so on until the last strand and loop will be reached. If the SP satisfies all strand-loop pairs for all proteins with the given SSS, the test of sensitivity is considered to be successfully completed.

The next step is to test all other known protein structures with different SSS in order to determine the specificity of SP for SSS as shown in Fig. 1. The procedure starts in the same way as for the test of sensitivity with a search for a possible location strand #1 in a query sequence. If at any stage of the analysis, the SP criteria for the given strand or loop are not met, then the query protein is considered to be a true negative. If, on the other hand, the query sequence

satisfies all the SP criteria, then the query protein is considered to be a false positive. To improve the specificity, SP may then be modified and is tested again, first for sensitivity and then for specificity, as outlined above.

3 Results

3.1 Selecting Target Proteins

We selected 1315 domains of proteins with 10 strands in 2 sandwich-like sheets (Fig. 1) from the proteins of “immunoglobulin-like beta-sandwich” fold and family of V-set domains, according to SCOP and SCOPe classifications (b.1.1.1, b.1.1.0) [22, 23]. Sequences were taken from PDB records [21]. Secondary structure data and SSS topology data from PDBSum database were used to deduce supersecondary structure [24]. The SSS topology was checked and corrected using the SPACE server [25].

3.2 SSS Sequence Pattern

1. *N-tail fragment before strand #1:* length 1–7 residues. Highly variable contents of residues and lengths (in protein structures 1insn L: DIV; 1xed A: KSP).

Not allowed: more than two hydrophobic residues from the “h” group.
2. *Strand #1:* minimum length—three residues (2atp B: IQT).

Required: at least one residue from the “h+A” group.
Not allowed: both Pro and Gly residues.
3. *The loop between strands #1 and #2:* length 3–7 residues.

Required: loop score >1, if the length of the loop is between three and four residues; loop score >3, if the length of the loop is between five and seven residues.
Not allowed: any residues from the “h” group, if the length of the loop is between three and five residues; more than two residues from the “h” group if the length of the loop is between six and seven residues.
4. *Strand #2:* minimum length—three residues (1rhh B: EVK). Two consecutive strands #1 and #2 must have the same orientation in the different sheets (Fig. 1).

Required: one or more residues from the “h+A” group at position #1 or 2; if Ala is a sole hydrophobic residue in the strand, then residue from the ST group is presented as well (2rhe A: SAS).
Not allowed: more than one residue from the h group; Pro or Gly residues are not allowed at position #2; more than one residue from NotFav group.
5. *The loop between strands #2 and #3:* length 3–7 residues.

Required: loop score >2, if the length of the loop is between three and five residues; loop score >4, if the length of the loop is between six and seven residues.

Not allowed: more than one residue from the “h” group, if the length of the loop is between three and five residues; more than three residues from the h group, if the length of the loop is between six and seven residues.

6. *Strand #3:* minimum length is seven residues. In almost all structures (~99%), Cys residue is found in the strand.

Required: position #3 is occupied by the residues from the “h” group (1bm3 H: LKLSCAA). Position #5 is occupied by the residue from the “h+C, A” group (1rur H, VKISCKA; 3lrg A, VTISVSG).

Residues at positions #1 and 5 are correlated: if Cys residue is absent at position #5, then residue from the “h group” occupies the position #1 (4w68 A: LRLSATA).

Residues at positions #1 and 7 are correlated: at least one of these two positions is occupied by the residue from the “h+A” group (1baf H: QSLTCTV).

Not allowed: more than six hydrophobic residues (4orz C: LRLFCAA); more than two Ala residues (2hrp A: RKLSCAA); more than two Gly residues (1w72 L: ARITCGG); more than one Pro residue (1pko A: AELPCRI); both Pro and Gly residues; Pro and Gly at positions #1, 3, and 5; more than four residues from the “h” group (1kgc D: VHLPCNH).

7. *The loop between strands #3 and #4:* length 6–13 residues. The total length of all preceding loops varies between 29 and 39 residues.

Required: loop score >2, if the length of the loop is between six and eight residues; loop score >5, if the length of the loop is more than nine residues.

Not allowed: more than two residues from the h group if the length of the loop is six residues; more than three residues from the h group if the length of the loop is varied between seven and eight residues; more than five residues from the h group if the length of the loop is more than eight residues.

8. *Strand 4:* minimum length is six residues.

Require: at least one hydrophobic residue from the h group; at least two residues from the group “h+A,Y,C” in the middle of the strand (1yc7 A: TGWYRQ; 1hkf A: KGWCKE).

Position #3 is occupied by the residue from the “h+A,Y” group. In the overwhelming number of structures, about 98%, Trp residue is found in this position.

Not allowed: more than five residues from the “h+A,Y,C” group (1plg L: LYWYLQ); more than one Pro or Gly residue (1u3h A: FPWYQQ); residue Pro or Gly at positions #1 and 3;

more than one Ala residue (3R4D A: FAWYKG); more than two residues from the NotFav group (1u3h A: FDYFPW).

9. *The loop between strands #4 and #5:* length 4–11 residues.

Required: loop score >2 if the length of the loop is between four and six residues; loop score >3 if the length of the loop is between seven and eight residues; loop score >5 if the length of the loop is nine residues; loop score >9, if the length of the loop is more than nine residues.

Not allowed: more than two hydrophobic residues from the “h” group.

10. *Strand #5:* minimum length—five residues.

Require: at least one residue from the “h” group at position #1, 3, or 5 (3rnq A: EQAAF); at least two residues from the “h+A,Y” group.

Not allowed: more than two residues from the NotFav group; more than one Pro residue.

11. *The loop between strands 5 and 6:* length 2–11 residues.

Required: loop score >0 if the length of the loop is between four and five residues; loop score >1 if the length of the loop is between six and seven residues; loop score >3 if the length of the loop is more than seven residues.

12. *Strand #6:* minimum length is two residues. The sequence of the strand is very variable (2atp B: VL; 1f4x H: TS; 1pz5 A: NR).

There are two SSS variants: (1) the edge strand #6 forms H-bond to strand 5 (Fig. 1a) or (2) the edge strand #6 forms H-bond to the strand #7 (Fig. 1b). In both variants, the edge strand #6 may have residue-residue contacts to both strands 5 and 7, so it is difficult to predict the preferred location of the strand #6.

Not allowed: Gly and Pro residues; more than one residue from the NotFav group;

13. *The loop between strands #6 and #7:* length is 6–9 residues.

Required: loop score >2.

Not allowed: more than two residues from the “h+A,Y,C” group.

14. *Strand #7:* minimum length is four residues.

Required: position #1—occupied by residue from the h+A,Y,C group or from the ST group.

Position #3: occupied by the residue from the “h+A,Y,C” group, or from the “ST” group, or by residue G.

Residues at positions #1 and 3 are correlated: if residue S or T is at position #1, then a residue from the h group is at position #3 (1mfa H: TKLTA); if residue S, T, or G is at position #3, then residues from the h group is at position 1 (3ab0 B: FTTSR); if residue Y, A, or C is at position

1, then residues from the h+A,Y,C group are at position #3 (2aq2 YEAS, 4ios D: CAAS).

Not allowed: residue Pro; more than two residues from the NotFav group.

15. *The loop between strands #7 and #8:* length is 3–10 residues.

Required: loop score >2.

Not allowed: more than one residue from the h group, if the length of the loop is between three and five residues; more than two residues from the h group if the length of the loop is more than five residues.

16. *Strand #8:* minimum length is five residues.

Required: at least one residue from the h group at position #1, 3, or 5; at least two residues from the “h+A,Y,C” group at position 1, 3, or 5 (1ghf H: AYKQI).

Correlation of residues at positions #1, 3, and 5: if a residue at position #1 is not from the h+A,Y group, then residues at positions #3 and 5 are from the h group (1kgc E: STLKI).

Not allowed: Cys residue; more than one Pro or Gly residue (1fo0 A: IGLII; 1oga E: FPLTV}; Pro and Gly residues at positions #1, 3, and 5; more than four residues from the h+A,Y group; more than two residues from the NotFav group; more than three Ala residues (1qfw H: AAMQA); more than two Tyr residues (1gpo H: YYLDL).

17. *The loop between strands 8 and 9:* length is 5–13 residues.

Required: loop score >1 if the length of the loop is between 5 and 6 residues; loop score >2, if the length of the loop is between 6 and 7 residues; loop score >3, if the length of the loop is between 8 and 13 residues.

Not allowed: more than 3 residues from the h group, if the length of the loop is less than 6 residues; more than 4 residues from the h group, if the length of the loop varies between 6 and 13 residues.

18. *Strand #9:* minimum length is seven residues.

Required: at least three residues from the h+A,Y,C group (1xed A: GRYKCGL).

Position #3: occupied by a residue from the “h+Y” group.

Position #5: occupied by a residue from the h+A,C group. In almost all structures, there is Cys–Cys bond between strands #3 and #9. In other structures this position is hydrophobic (3lrg A: ADYYAAT).

Correlation between residues at positions #1 and 7: at least one of these inward-directed positions is occupied by a residue from h+A group (3r4d A: GVYTLDM).

Not allowed: Pro residue at any positions except position #7 (1kxv C: AMYYCKP); more than two Gly residues (1dlf H:

GIYYCTG); both Pro and Gly residues; more than four residues from NotFav group (1jnl L: **GNYYCHH**).

19. *The loop between strands #9 and #10:* length is 2–23 residues.

Required: loop score >0 if the length of the loop is between 6 and 11 residues; loop score >2 if the length of the loop is between 12 and 13 residues; loop score >6 if the length of the loop is between 14 and 15 residues; loop score >7 if the length of the loop is between 16 and 17 residues; loop score >8 if the length of the loop is between 18 and 19 residues; loop score >9 if the length of the loop is between 20 and 23 residues.

There is at least one residue from the h group at position 2 or 10. If the length of the loop is less than 4 residues, then at least 1 hydrophobic residue from the h group is presented; more than 3 hydrophobic residues if the length of the loop is between 4 and 5 residues; more than 5 hydrophobic residues if the length of the loop is between 6 and 8 residues; more than 6 hydrophobic residues if the length of the loop is between 9 and 10 residues; more than 7 hydrophobic residues if the length of the loop is between 11 and 23 residues.

20. *Strand #10:* there are two variants of the strand.

Variant 1. The strand with a bulge; there is one residue at position #3' as shown in Fig. 1a, b.

Variant 2. The strand without a bulge (Fig. 1c).

Characteristics common to both variants:

Required: at least two residues from the h group at position #2, 7, or 9; at least one residue from the h group or from the “ST” group at position #5.

Not allowed: more than one Pro residue (3zkq D: **RRGP~~G~~GTQVTV**).

Characteristics specific to variant 1: minimum length is 10 residues.

Required: at least one residue from the h group at position #2 or 10; at least one residue Gly at position #3 or 4; at least two residues from the NotFav group at positions #3, 3', and 4 (4x7e C: **SRGR~~G~~GTQVTV**).

Not allowed: more than four Gly residues (2rhe A: **GF~~GGG~~GTLLTV**); the residue Gly at positions #6–10; more than one residue from the h group at positions #3, 3', and 4.

Characteristics specific to variant 2. The minimum length is nine residues.

Required: at least three residues from the h group at inward-directed positions #2, 4, 6, 8, and 10 (3rrq A: **KESLRAELRV**).

Not allowed: Gly residue at positions #3–10.

4 Notes

1. Analysis of all beta-sandwich structures with two beta sheets (per the SCOP classification) revealed a certain constraint for localization of consecutive strands in strandons and beta sheets [31, 32]. There are always at least two pairs of consecutive strands $i, i + 1$ and $j, j + 1$ such that strand i forms H-bond contacts with strand j in one beta sheet and strand $i + 1$ forms H-bond with strand $j + 1$ in another beta sheet. This specific configuration of four strands was termed an “interlock,” because the two pairs of strands are “interlocked.” This four-strand configuration is found in most sandwich-like proteins, but not in any other beta structures. The strands $i, i + 1, j, j + 1$ correspond to strands 3, 4, 8, and 9 in the target proteins discussed in this paper (Fig. 1). Thus, the interlock is the structural invariant substructure of sandwich-like beta proteins. We suppose that formation of H-bond contacts between non-consecutive strands in the interlock is due to the specific distribution of specific residues in these strands that favor to formation of this stable substructure.
2. The number of hydrophobic residues in most proteins with the SSS described in Fig. 1 varies from 30% to 40% of all residues in the sequences. Most of these residues (60–70%) are located at inward-oriented positions within the ten strands. About 50% of the residues that form hydrophobic core are found in four interlock strands. This observation supports the conclusion about the crucial role of residues within interlock strands for stability of sandwich-like beta proteins.

References

1. Lazaridis T, Karplus M (2000) Effective energy functions for protein structure prediction. *Curr Opin Struct Biol* 10:139–145
2. Huang PS, Boyken SE, Baker D (2016) The coming of age of de novo protein design. *Nature* 537(7620):320–327
3. Adhikari B, Cheng J (2018) CONFOLD2: improved contact-driven *ab initio* protein structure modeling. *BMC Bioinformatics* 19 (1):22
4. Pace CN, Shirley BA, McNutt M, Gajiwala K (1996) Forces contributing to the conformational stability of proteins. *FASEB J* 10:75–83
5. Jothi A (2012) Principles, challenges and advances in *ab initio* protein structure prediction. *Protein Pept Lett* 9:1194–1204
6. Fogolari F, Corazza A, Esposito G (2018) Free energy, enthalpy and entropy from implicit solvent end-point simulations. *Front Mol Biosci* 5:11
7. Nguyen H, Maier J, Huang H, Perrone V, Simmerling C (2014) Folding simulations for proteins with diverse topologies are accessible in days with a physics-based force field and implicit solvent. *J Am Chem Soc* 136 (40):13959–13962
8. Chatzou M, Magis C, Chang JM et al (2016) Multiple sequence alignment modeling: methods and applications. *Brief Bioinform* 17 (6):1009–1023
9. Le Q, Sievers F, Higgins DG (2017) Protein multiple sequence alignment bench-marking through secondary structure prediction. *Bioinformatics* 33(9):1331–1337
10. Xiang Z (2006) Advances in homology protein structure modeling. *Curr Protein Pept Sci* 7 (3):217–227

11. Eaton KV, Anderson WJ, Dubrava MS et al (2015) Studying protein fold evolution with hybrids of differently folded homologs. *Protein Eng Des Sel* 28(8):241–250
12. He Y, Chen Y, Alexander P et al (2008) NMR structures of two designed proteins with high sequence identity but different fold and function. *Proc Natl Acad Sci U S A* 105 (38):14412–14417
13. Schubert HL, Raux E, Wilson KS et al (1999) Common chelatase design in the branched tetrapyrrole pathways of heme and anaerobic cobalamin synthesis. *Biochemistry* 38 (33):10660–10669
14. Ma J, Wang S (2014) Algorithms, applications, and challenges of protein structure alignment. *Adv Protein Chem Struct Biol* 94:121–175
15. Sadowski MI, Taylor WR (2012) Evolutionary inaccuracy of pairwise structural alignments. *Bioinformatics* 28(9):1209–1215
16. Kolodny R, Pereyaslavets L, Samson AO et al (2013) On the universe of protein folds. *Annu Rev Biophys* 42:559–582
17. Bernal J (1939) Structure of proteins. *Nature* 143:663–667
18. Bresler SE, Talmud DL (1944) On the nature of globular proteins. II. Some consequences of a new hypothesis. *Dokl Akad Nauk SSSR* (in Russian) 43:326–330
19. Kauzmann W (1959) Some factors in the interpretation of protein denaturation. *Adv Protein Chem* 14:1–63
20. Nick Pace C, Scholtz JM, Grimsley GR (2014) Forces stabilizing proteins. *FEBS Lett* 588 (14):2177–2184
21. Berman HM, Westbrook J, Feng Z et al (2000) The protein data bank. *Nucleic Acids Res* 28 (1):235–242
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. Fox NK, Brenner SE, Chandonia JM (2014) SCOPe: Structural Classification of Proteins—extended, integrating SCOP and ASTRAL data and classification of new structures. *Nucleic Acids Res* 42:D304–D309
24. de Beer TAP, Berka K, Thornton JM et al (2014) PDBsum additions. *Nucleic Acids Res* 42:D292–D296
25. Sobolev V, Eyal E, Gerzon S et al (2005) SPACE: a suite of tools for protein structure prediction and analysis based on complementarity and environment. *Nucleic Acids Res* 33:W39–W43
26. Costantini S, Colonna G, Facchiano AM (2006) Amino acid propensities for secondary structures are influenced by the protein structural class. *Biochem Biophys Res Commun* 342(2):441–451
27. Chou PY, Fasman GD (1974) Conformational parameters for amino acids in helical, beta sheet, and random coil regions calculated from proteins. *Biochemistry* 13(2):211–222
28. Fujiwara K, Toda H, Ikeguchi M (2012) Dependence of α -helical and β -sheet amino acid propensities on the overall protein fold type. *BMC Struct Biol* 12:18
29. Nilsson I, Johnson AE, von Heijne G (2003) How hydrophobic is alanine? *J Biol Chem* 278 (32):29389–29393
30. Nagano N, Ota M, Nishikawa K (1999) Strong hydrophobic nature of cysteine residues in proteins. *FEBS Lett* 458(1):69–71
31. Kister AE, Finkelstein AV, Gelfand IM (2002) Common features in structures and sequences of sandwich-like proteins. *Proc Natl Acad Sci U S A* 99:14137–14141
32. Kister AE (2015) Amino acid distribution rules predict protein fold: protein grammar for beta-strand sandwich-like structures. *Biomolecules* 5:41–59



Chapter 17

Homology Searches Using Supersecondary Structure Code

Hiroshi Izumi

Abstract

Supersecondary structure code (SSSC), which is represented as the combination of α -helix-type (SSSC: **H**), β -sheet-type (SSSC: **S**), the other (SSSC: **T**), and disorder residue or C-terminal (SSSC: **D**) patterns, has been produced by the developed concept of Ramachandran plot, in addition, with the ω angle and with the specification of positions of torsion angles in a protein by the registration of codes for torsion angles of each amino acid peptide unit, derived from the fuzzy search of structural code homology using the template patterns **3a5c4a** (SSSC: **H**) and **6c4a4a** (SSSC: **S**) with conformational codes. The DSSP (Dictionary of Secondary Structure in Proteins) method assigns the secondary structure including hydrogen bond well. In contrast, supersecondary structure code is very sensitive to the supersecondary structures of proteins. In this chapter, the protocol of homology search methods, the sequence alignment using supersecondary structure code, the assignment of supersecondary structure code **T**, the fuzzy search using supersecondary structure code, and the exact search using supersecondary structure code are described. Supersecondary structure code is variable with the conformational change. If possible, many Protein Data Bank (PDB) data of similar main chains of proteins should be used for the homology searches. The thorough check of SSSC sequences is also useful to reveal the role of target pattern.

Key words Supersecondary structure code, Ramachandran plot, DSSP, Multiple sequence alignment, Conformation

1 Introduction

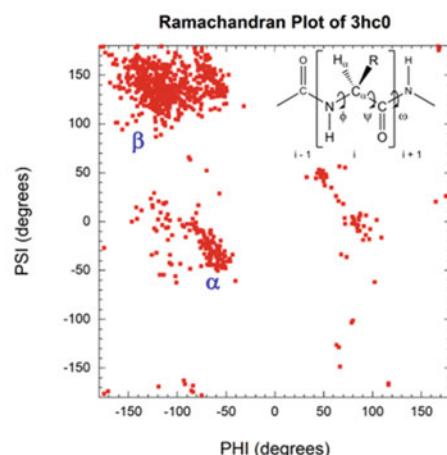
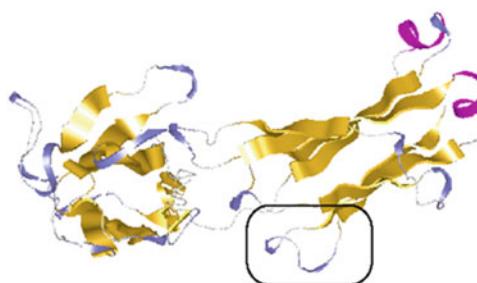
Structural regions of proteins such as $\alpha\alpha$ corner (right-angled connection of α -helices), which cannot be classified as secondary structures (mainly loop regions), are called supersecondary structure motifs [1, 2]. We have developed the codification and fuzzy search techniques of supersecondary structure homology in X-ray structures of proteins and have shown that supersecondary structure motifs closely contribute to the protein functions [3]. Supersecondary structure code (SSSC) is represented as the combination of α -helix-type (SSSC: **H**), β -sheet-type (SSSC: **S**), the other (SSSC: **T**), and disorder residue or C-terminal (SSSC: **D**) patterns. In this chapter, the practical homology search methods using supersecondary structure code are introduced.

1.1 Supersecondary Structure Code and Ramachandran Plot

Ramachandran plot is the two-dimensional (2D) plot of the ϕ - ψ torsion angles of a protein backbone for a simple view of the conformation of protein [4–7]. The ϕ - ψ angles cluster into distinct regions in the Ramachandran plot is suitable to recognize the particular secondary structure. However, the difference of characteristic supersecondary structure motifs between a IgG immunoglobulin and a IgM rheumatoid factor in the black frames cannot be distinguished from the Ramachandran plots (Fig. 1).

Supersecondary structure code was made by the developed concept of Ramachandran plot, in addition, with the ω angle. Further, supersecondary structure code contains the specification of positions of torsion angles in a protein by the registration of codes for torsion angles of each amino acid peptide unit and is derived from the fuzzy search of structural code homology using the template patterns **3a5c4a** (α -helix-type) and **6c4a4a** (β -sheet-

a IgG Immunoglobulin



b IgM Rheumatoid Factor

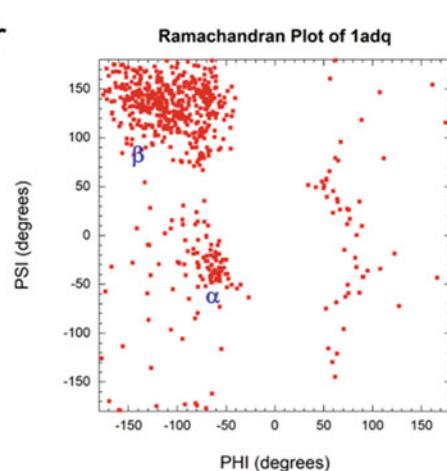
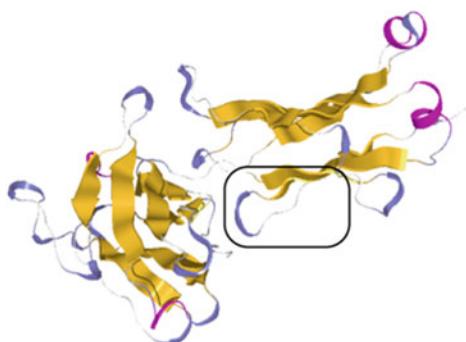


Fig. 1 Ramachandran plots of (a) IgG immunoglobulin (3hc0) and (b) IgM rheumatoid factor (1adq). The supersecondary structure motifs in the black frames cannot be distinguished from the plots (Adapted with permission from ref. 3. Copyright 2013 American Chemical Society)

type) with conformational codes (Fig. 2) [8] for each amino acid peptide unit [3].

Namely, the conformational code is composed of the combination of the codes of regional angle locations and the 12-divided segments (conformational elements) and is available for the description of the local structures of various molecules (Fig. 2) [8]. The comparison of structural code homology of the main chains using the conformational elements with the 12-divided segments extracted the fine conformational difference such as the case of 2imm and 2mcp that the strict structural code homology was 11.5% [3]. On the other hand, the comparison of 2imm and 2mcp using the fuzzy search of structural code homology, in which it was judged as the high homology if the conformational elements were included in a range of 90° , and the structural code homology of the main chain for each amino acid peptide unit was calculated as the logical conjunction of structural code homology at the angle locations **A**, **B**, and **C** in Fig. 2, which indicated that the structural code homology of the main chains was 94.7% [3]. The residual fragments (5.3%) showed the large conformational difference such as the *cis-trans* relationship at angle location **B** of Ser [3].

Most protein structures except α -helix and β -sheet also consist of the combination of α -helix-type (SSSC: **H**) and β -sheet-type (SSSC: **S**) fragment patterns, and the other (SSSC: **T**) patterns are few [3]. Therefore, all the conformational patterns of main chain for each amino acid peptide unit can be classified by using this supersecondary structure code, which is suitable for the description of supersecondary structures of proteins.

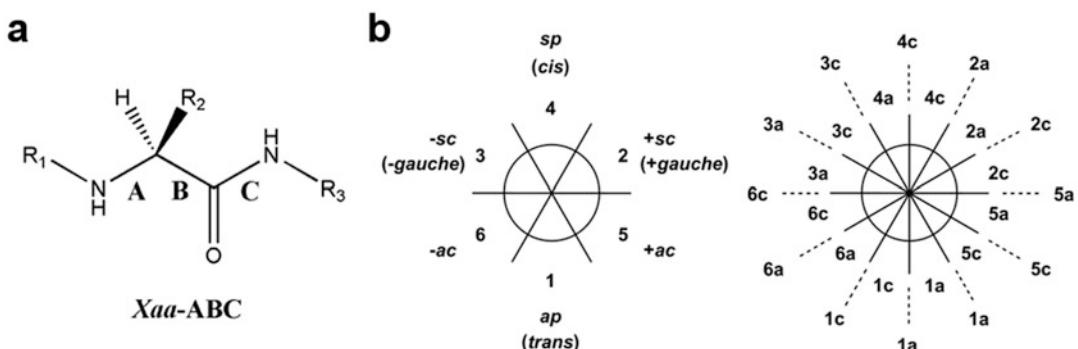


Fig. 2 Definition of angle locations and conformational elements for conformational code of main chains of proteins. (a) Angle locations consist of prefix of amino acids (Xaa) and symbols indicating the bond locations (ABC). (b) Conformational elements represent classification of dihedral angles, and the elements **1**, **2**, **3**, **4**, **5**, and **6** correspond to conformational terms *ap* (*antiperiplanar*), *+sc* (*+synclinal*), *-sc* (*-synclinal*), *sp* (*synperiplanar*), *+ac* (*+anticlinal*), and *-ac* (*-anticlinal*), respectively. The terms, **c** and **a**, in the conformational elements mean clockwise and anticlockwise, respectively (Adapted with permission from ref. 3. Copyright 2013 American Chemical Society)

1.2 Supersecondary Structure Code (SSSC) and Dictionary of Secondary Structure in Proteins (DSSP)

The DSSP (Dictionary of Secondary Structure in Proteins) program has been used to standardize secondary structure assignments (and much more) for all protein entries in the Protein Data Bank (PDB) widely [9, 10]. The algorithm identifies hydrogen bonds between main chain carbonyl and amide groups. The DSSP method assigns the secondary structure including hydrogen bond well, but some blanks which correspond to loop or are irregular are found (Fig. 3) [3]. In this way, the DSSP program is not suitable for homology search of supersecondary structure of proteins.

On the other hand, supersecondary structure code is very sensitive to the supersecondary structures of proteins [3]. This code can detect the difference of characteristic fragment structures between IgG immunoglobulin (SSSC: SHHSHSS) and IgM rheumatoid factor (SSSC: TTTSSSS) (Fig. 4). Further, GM-CSF auto-antibody, TSHR autoantibody, CD1D, and MIC-A,B indicate similar patterns of IgM rheumatoid factor (SSSC: TTTSSSS), and supersecondary structure code is thus available for the classification of group of autoantibodies.

In the next section, the protocol of homology search methods using supersecondary structure code is shown.

2 Materials

The following computer programs are necessary for the homology search methods using supersecondary structure code.

2.1 SSSC Program

1. Python 2.7: Install Python 2.7 for Windows [11].
2. Biopython: Install Biopython [12].
3. SSSC (*see Note 1*): Download and unzip the “SSSC.zip” file [13]. Put the “SSSC” folder under the “C” drive. Save the original “Polypeptide.py” file at C:\Python27\Lib\site-packages\Bio\PDB to the safety space. Replace the “Polypeptide.py” file in the “SSSC” folder to the original “Polypeptide.py” file at C:\Python27\Lib\site-packages\Bio\PDB.

2.2 SSSC Analysis Program

1. SSSC Analysis: Download and unzip the “SSSC_Analysis.zip” file [14]. Put the “SSSC_Analysis” folder under the “C” drive. The “Alignment_input_1_0_1.py” and “ssc_to_seq_1_0_2.py” files are used in the sequence alignment using supersecondary structure code (*see Subheading 3.2*). The “ConfCode_m-chain_1_0_2.py” and “ConfCode_T_homology_1_0_2.py” files are used in the assignment of supersecondary structure code T (*see Subheading 3.3*). The “Struct_homology_sssc_1_0_4.py” file is used in the fuzzy search using supersecondary structure code (*see Subheading 3.4*). The

Num.	2bsr ^a	1a6z ^a	3hc0 ^a	2w9e ^a	1adq ^a	2bsr ^b	1a6z ^b	3hc0 ^b	2w9e ^b	1adq ^b	2bsr ^c	1a6z ^c	3hc0 ^c	2w9e ^c	1adq ^c
238	ARG	GLY	SER	SER	ALA	T	T	S	S	S	S	S	S	E	E
239	THR	THR	LEU	MET	ALA	S	S	S	S	S			E	E	E
240	PHE	TYR	SER	SER	SER	S	S	S	S	S	E	E	E	E	E
241	GLN	GLN	SER	SER	SER	S	S	S	S	S	E	E	E	E	E
242	LYS	GLY	THR	THR	TYR	S	S	S	S	S	E	E	E	E	E
243	TRP	TRP	LEU	LEU	LEU	S	S	S	S	S	E	E	E	E	E
244	ALA	ILE	THR	THR	SER	S	S	S	S	S	E	E	E	E	E
245	ALA	THR	LEU	LEU	LEU	S	S	S	S	S	E	E	E	E	E
246	VAL	LEU	SER	THR	THR	S	S	S	S	S	E	E	E	E	
247	VAL	ALA	LYS	LYS	PRO	S	S	H	H	H	E	E	H	H	H
248	VAL	VAL	ALA	ASP	GLU	S	S	H	H	H	E	E	H	H	H
249	PRO	PRO	ASP	GLU	GLN	S	S	H	H	H	E		H	H	H
250	SER	PRO	TYR	TYR	TRP	S	S	H	H	H	T	T	H	H	H
251	GLY	GLY	GLU	GLU	LYS	T	T	H	H	H	T	T	H	H	H
252	GLU	GLU	LYS	ARG	SER	H	H	H	H	T	G	T	H	T	
253	GLU	GLU	HIS	HIS	HIS	H	H	S	S	S	G	G			
254	GLN	GLN	LYS	ASN	LYS	H	H	H	H	H	G	G			S
255	ARG	ARG	VAL	SER	SER	H	H	S	S	S	G	G	E		
256	TYR	TYR	TYR	TYR	TYR	S	S	S	S	S	E	E	E	E	E
257	THR	THR	ALA	THR	SER	S	S	S	S	S	E	E	E	E	E
258	CYS	CYS	CYS	CYS	CYS	S	S	S	S	S	E	E	E	E	E
259	HIS	GLN	GLU	GLU	GLN	S	S	S	S	S	E	E	E	E	E
260	VAL	VAL	VAL	ALA	VAL	S	S	S	S	S	E	E	E	E	E
261	GLN	GLU	THR	THR	THR	S	S	S	S	S	E	E	E	E	E
262	HIS	HIS	HIS	HIS	HIS	S	S	S	S	T				E	
263	GLU	PRO	GLN	LYS	GLU	H	H	H	H	T	T	T	T	S	T
264	GLY	GLY	GLY	THR	GLY	H	H	H	H	T	T	T	T	S	T
265	LEU	LEU	LEU	SER	SER	S	S	S	S	S		S			E
266	PRO	ASP	SER	THR	THR	H	H	H	H	S	S	S	S	S	E
267	LYS	GLN	SER	SER	VAL	S	S	S	S	S	S	S	S	S	E
268	PRO	PRO	PRO	PRO	GLU	S	S	S	S	S					E
269	LEU	LEU	VAL	ILE	LYS	S	S	S	S	S	E	E	E	E	E
270	THR	ILE	THR	VAL	THR	S	S	S	S	S	E	E	E	E	E
271	LEU	VAL	LYS	LYS	VAL	S	S	S	S	S	E	E	E	E	E
272	ARG	ILE	SER	SER	ALA	S	S	S	S	S	E	E	E	E	
273	TRP	TRP	PHE	PHE	PRO	S	T	S	S	H	E	E	E	E	S
274	GLU		ASN	ASN	THR	S		S	S	S	E				
275	PRO		ARG	ARG	GLU	T		S	T	H	T		T	S	
276		GLY	ASN	CYS				T	H	T			T		
277		GLU	SER					T	T	T					

Fig. 3 Comparison of SSSC with DSSP for MHC class I (2bsr; 1a6z, human) and light chains of immunoglobulins (3hc0; 1adq, human; 2w9e, mouse). All amino acid peptide units are finely assigned using SSSC.
^aAmino acid sequence. ^bSSSC (H, α -helix-type; S, β -sheet-type; and T, other type). ^cDSSP [H, α -helix; E, extended strand, participates in beta ladder; B, residue in isolated β -bridge; G, 3-helix (3/10 helix); I, 5 helix (π -helix); T, hydrogen-bonded turn; and S, bend] (Adapted with permission from ref. 3. Copyright 2013 American Chemical Society)

“Fragment_search_1_0_2.py” file is used in the exact search using supersecondary structure code (see Subheading 3.5).

- MAFFT: Install MAFFT optionally [15–17] or other multiple sequence alignment programs.
- Clustal W: Install Clustal W optionally [18, 19] or other multiple sequence alignment programs.

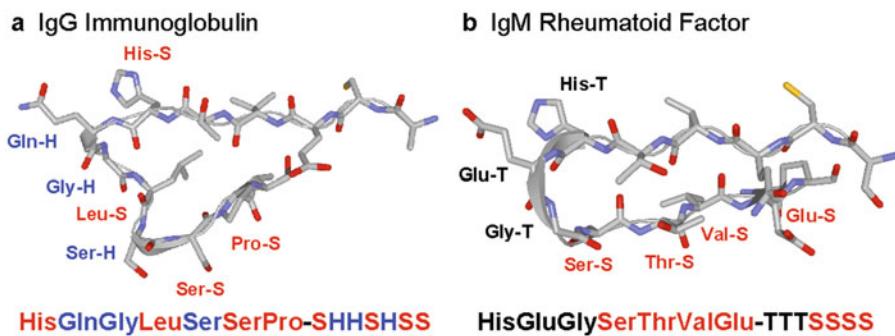


Fig. 4 Fragment structures of main chains (258–270 aa) for light chains of immunoglobulins: (a) IgG immunoglobulin (3hc0, human), (b) IgM rheumatoid factor (1adq, human). Characteristic fragment patterns (SSSC: SHHSHSS) sandwiched between two identical amino acid sequences His and Pro are protruded on the molecular surfaces. Of particular interest, the IgM rheumatoid factor loses the fragment pattern (SSSC: TTTSSSS) (Adapted with permission from ref. 3. Copyright 2013 American Chemical Society)

3 Methods

All input data must be putted as text files at the proper directories in the “SSSC” folder (*see* Subheading 2.1) or “SSSC_Analysis” folder (*see* Subheading 2.2) for the SSSC or SSSC analysis programs.

3.1 Derivation of Supersecondary Structure Code

1. Copy PDB files (pdbxxxx.ent) [20] into the “PDB_data” folder in the “SSSC” folder (*see* Subheading 2.1).
2. Run the “SSSC_1_0_2.py” module.
3. The converted SSSC files are created into the “FASTA_sscc_data” folder in the “SSSC” folder.

3.2 Sequence Alignment Using Supersecondary Structure Code

1. Create the converted SSSC files with FASTA format into the “FASTA_sscc_data” folder in the “SSSC” folder (*see* Subheading 3.1).
2. Run the “Alignment_input_1_0_1.py” module in the “SSSC_Analysis” folder (*see* Subheading 2.2).
3. The input files of SSSC and amino acid sequence, “ssc.txt” and “seq.txt,” are created into the “SSSC_Analysis” folder.
4. Align the SSSC sequences by using the proper multiple sequence alignment programs such as MAFFT and/or Clustal W (Fig. 5).
5. If conversion of the SSSC sequences to the amino acid sequences is needed, copy the converted SSSC files with FASTA format (*see* Subheading 3.1) to the “ssc_to_seq” folder in the “SSSC_Analysis” folder.
6. Put the output file of multiple sequence alignment with SSSC into the “SSSC_Analysis” folder. Rename the file name to “mafft.txt.”

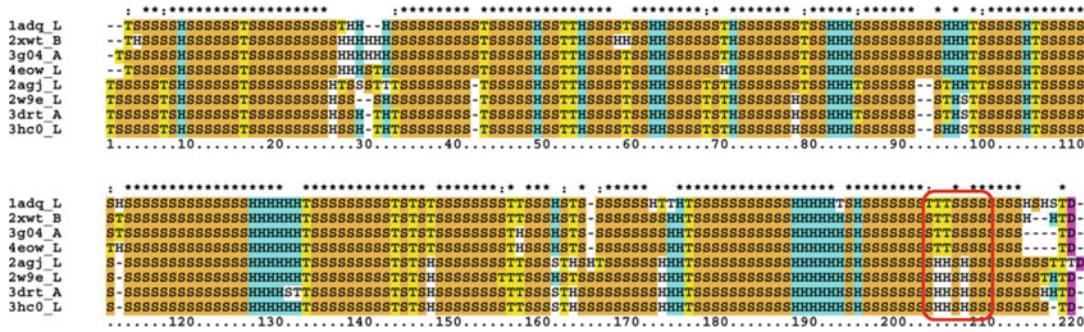


Fig. 5 Alignment result of the SSSC sequences for light chains of immunoglobulins (red frame: characteristic fragment patterns SHHSHSS and TTTSSSS)

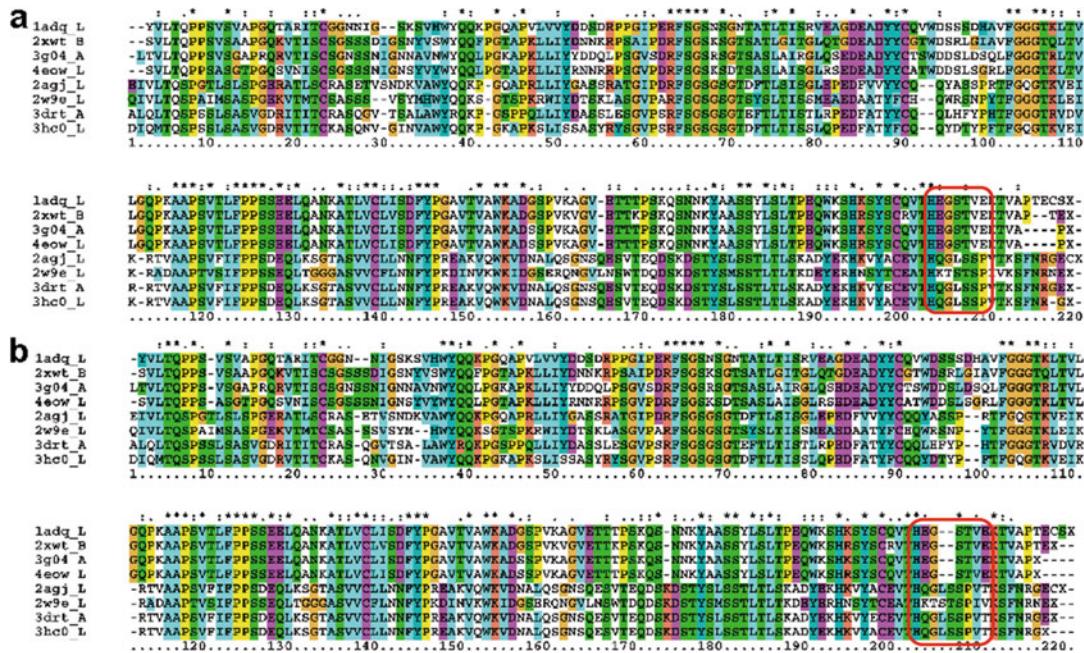


Fig. 6 Alignment result of the amino acid sequences for light chains of immunoglobulins (red frame: characteristic fragment patterns SHHSHSS and TTTSSSS). (a) Sequence alignment derived from SSSC. (b) ClustalX multiple sequence alignment

7. Run the “ssc_to_seq_1_0_2.py” module in the “SSSC_Analysis” folder (*see Subheading 2.2*).
8. The converted “ssc_to_seq.txt” file is created (Fig. 6) into the “SSSC_Analysis” folder (*see Note 2*).

3.3 Assignment of Supersecondary Structure Code T

1. If target patterns contain supersecondary structure code T (*see Note 2*), the characterization of T is necessary.
2. Run the “ConfCode_mchain_1_0_2.py” module in the “SSSC_Analysis” folder (*see Subheading 2.2*).

3. The converted conformational code files (Fig. 2) for main chains of proteins are created into the “ConfCode_mchain” folder in the “SSSC_Analysis” folder.
4. Copy and paste the line with the target codes in the converted conformational code file such as “LYS135, T, 2a, 6a, 4a” to the “ConfCode_T.txt” file in the “SSSC_Analysis” folder.
5. Run the “ConfCode_T_homology_1_0_2.py” module in the “SSSC_Analysis” folder (*see Subheading 2.2*).
6. The “ConfCode_T_list.txt” file is created into the “SSSC_Analysis” folder. If a file name exists in the “ConfCode_T_list.txt” file, a similar motif to the target pattern is contained in the main chain of protein. In this way, the homology of SSSC sequences with code T can be confirmed (*see Note 2*).

3.4 Fuzzy Search Using Supersecondary Structure Code

1. Copy the converted SSSC files with FASTA format (*see Subheading 3.1*) to the “FASTA_sscc_database” folder in the “SSSC_Analysis” folder.
2. Rename the file name of the converted SSSC file with the target pattern to “protein.txt” in the “SSSC_Analysis” folder.
3. Run the “Struct_homology_sscc_1_0_4.py” module [3] in the “SSSC_Analysis” folder (*see Subheading 2.2*).
4. The “Struct_homology_value.txt” file is created into the “SSSC_Analysis” folder. If a number near 1.0 with a file name exists in the “Struct_homology_value.txt” file, a similar motif to the target pattern is contained in the main chain of protein (*see Note 3*).

3.5 Exact Search Using Supersecondary Structure Code

1. Copy the converted SSSC files with FASTA format (*see Subheading 3.1*) to the “FASTA_sscc_database” folder in the “SSSC_Analysis” folder.
2. Copy and paste the target pattern in the converted SSSC file such as “HHHTTSHHH” (*see Note 4*) to the “Fragment_code.txt” file in the “SSSC_Analysis” folder.
3. Run the “Fragment_search_1_0_2.py” module in the “SSSC_Analysis” folder (*see Subheading 2.2*).
4. The “Fragment_search.txt” file is created into the “SSSC Analysis” folder. Count a number of file names. If a file name exists in the “Fragment_search.txt” file, the same motif of the target pattern is contained in the main chain of protein (*see Note 4*). If the number of file names is few, the target pattern is a rare motif (*see Note 3*).

4 Notes

1. The SSSC program was constructed by using Python [11] and Biopython [12] in reference to the description of “Generating Ramachandran (phi/psi) plots for Proteins” [21].
 2. The conversion of SSSC sequences to amino acid sequences is especially useful to find a characteristic supersecondary structure motif such as **HHHTTSHHH** of interferon- α (3oq3), which cannot be obtained by the usual multiple sequence alignments (Fig. 7).

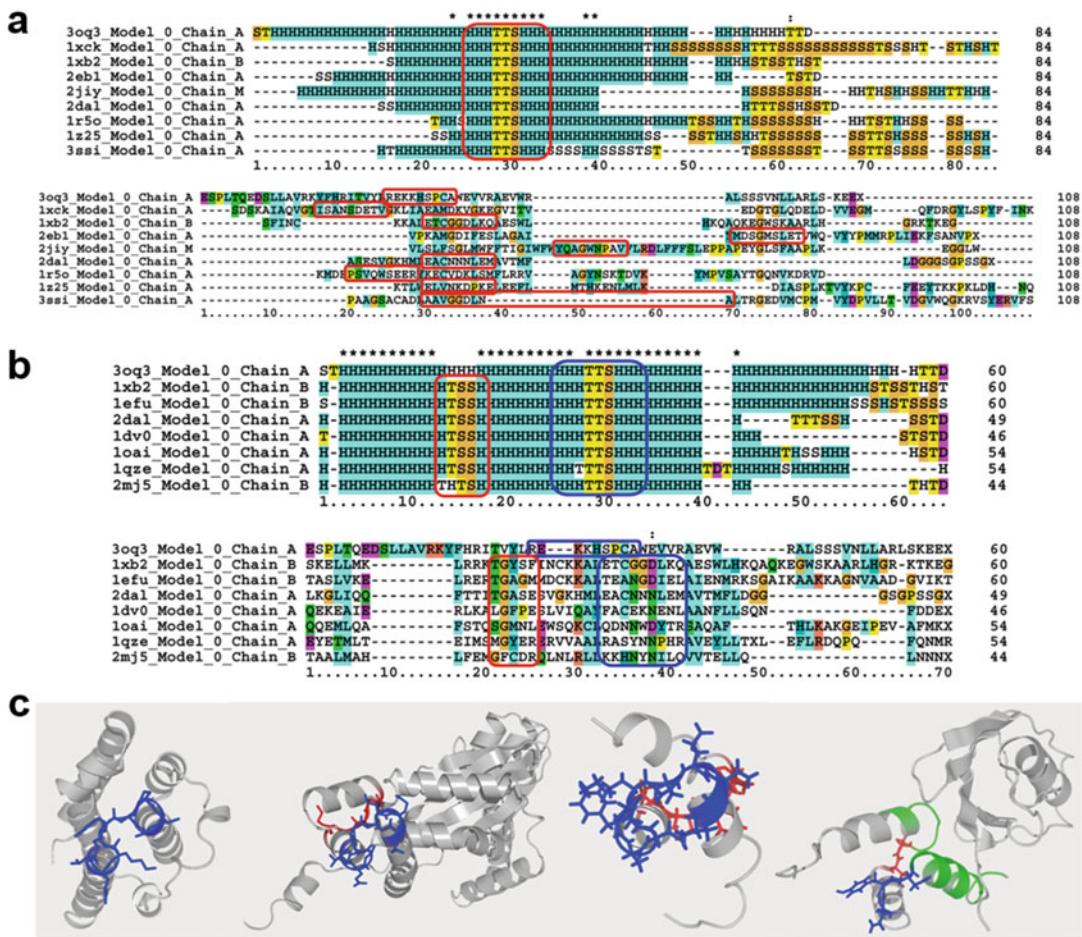


Fig. 7 Comparison of sequence alignments for characteristic fragment pattern **HHHTTSHHH** of interferon- α between SSSC sequences and amino acid sequences. **(a)** Interferon- α and proteins with **HHHTTSHHH** (3oq3, interferon- α ; 1xck, 60 kDa chaperonin; 1xb2, elongation factor Ts; 2eb1, endoribonuclease dicer; 2jy, photosynthetic reaction center; 2dal, FAS associated factor 1; 1r50, translation release factor 3; 1z25, argonaute; 3ssi, *Streptomyces* subtilisin inhibitor). **(b)** Interferon- α (3oq3) and ubiquitin-associated domains. **(c)** Supersecondary structure motifs of interferon- α and ubiquitin-associated domains

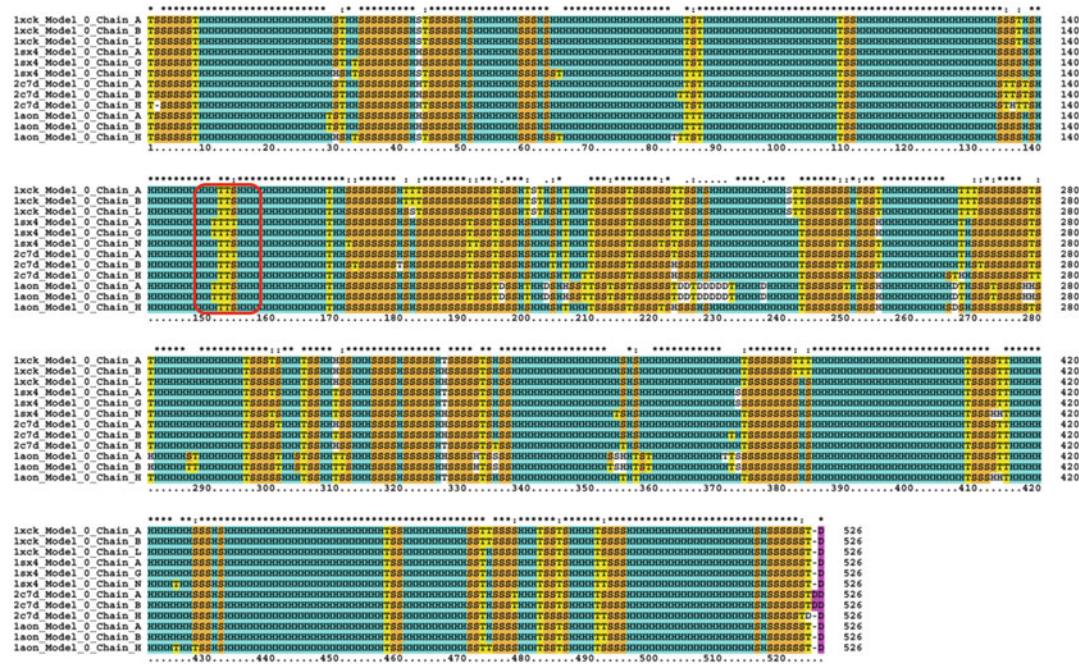


Fig. 8 Alignment result of the SSSC sequences for 60 kDa chaperonin (red frame: characteristic fragment pattern **HHHTTSHHH** and its conformational change)

- The fuzzy search using supersecondary structure code [3] should be used to give it a try. The target pattern can be found by the exact search using supersecondary structure code and the count of file number more efficiently than the fuzzy search (see Subheading 3.5).
 - Supersecondary structure code is variable with the conformational change. If possible, many PDB data of similar main chains of proteins should be used for the homology searches using supersecondary structure code, and the accuracy should be confirmed (Fig. 8) such as the case of 60 kDa chaperonin [22]. The thorough check of SSSC sequences is also helpful for elucidation of the role of target pattern like a hook (Fig. 9) [23].

Acknowledgments

This work was partly supported by JSPS KAKENHI Grant Number JP16K05711. The author thanks Dr. Rina K. Dukor and Professor Laurence A. Nafie for the discussion of supersecondary structure code.

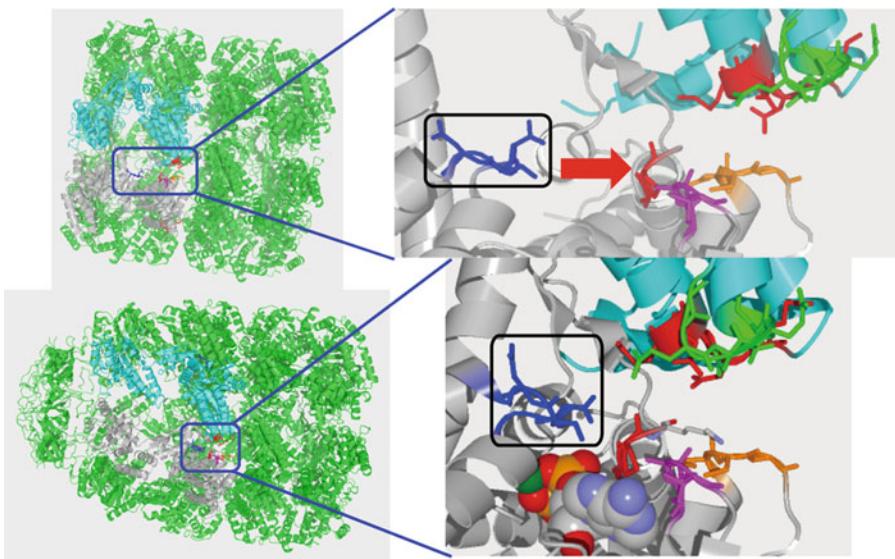


Fig. 9 Supersecondary structure motif (black frame: **HHHTTSHHH**) in 60 kDa chaperonin (1xck). The motif acts like a hook in molecular assembly of GroEL and GroES (1sx4)

References

- Andreeva NS, Gustchina AE (1979) On the supersecondary structure of acid proteases. *Biochem Biophys Res Commun* 87:32–42. [https://doi.org/10.1016/0006-291X\(79\)91643-7](https://doi.org/10.1016/0006-291X(79)91643-7)
- Richards FM, Kundrot CE (1988) Identification of structural motifs from protein coordinate data—secondary structure and 1st-level supersecondary structure. *Proteins* 3:71–84. <https://doi.org/10.1002/prot.340030202>
- Izumi H, Wakisaka A, Nafie LA, Dukor RK (2013) Data mining of supersecondary structure homology between light chains of immunoglobulins and MHC molecules: absence of the common conformational fragment in the human IgM rheumatoid factor. *J Chem Inf Model* 53:584–591. <https://doi.org/10.1021/ci300420d>
- Ramachandran GN, Ramakrishnan C, Sasisekharan V (1963) Stereochemistry of polypeptide chain configurations. *J Mol Biol* 7:95–99
- Kleywegt GJ, Jones TA (1996) Phi/psi-ology: ramachandran revisited. *Structure* 4:1395–1400. [https://doi.org/10.1016/s0969-2126\(96\)00147-5](https://doi.org/10.1016/s0969-2126(96)00147-5)
- Lovell SC, Davis IW, Adrendall WB, de Bakker PIW, Word JM, Prisant MG, Richardson JS, Richardson DC (2003) Structure validation by C alpha geometry: phi, psi and C beta deviation. *Proteins* 50:437–450. <https://doi.org/10.1002/prot.10286>
- Ho BK, Brasseur R (2005) The Ramachandran plots of glycine and pre-proline. *BMC Struct Biol* 5:14. <https://doi.org/10.1186/1472-6807-5-14>
- Izumi H, Nafie LA, Dukor RK (2016) Three-dimensional chemical structure search using the conformational code for organic molecules (CCOM) program. *Chirality* 28:370–375. <https://doi.org/10.1002/chir.22596>
- Touw WG, Baakman C, Black J, te Beek TAH, Krieger E, Joosten RP, Vriend G (2015) A series of PDB-related databanks for everyday needs. *Nucleic Acids Res* 43:D364–D368. <https://doi.org/10.1093/nar/gku1028>
- Kabsch W, Sander C (1983) Dictionary of protein secondary structure—pattern-recognition of hydrogen-bonded and geometrical features. *Biopolymers* 22:2577–2637. <https://doi.org/10.1002/bip.360221211>
- Python Software Foundation (2018) Python. <https://www.python.org>. Accessed 26 Jan 2018
- Open Bioinformatics Foundation (2018) Biopython. <http://biopython.org>. Accessed 26 Jan 2018

13. Izumi H (2016) SSSC. http://researchmap.jp/muqq8vge-2132135/#_2132135. Accessed 26 Jan 2018
14. Izumi H (2017) SSSC analysis. http://researchmap.jp/mufl0vbz5-2132135/#_2132135. Accessed 26 Jan 2018
15. Katoh K (2013) MAFFT version 7. <http://mafft.cbrc.jp/alignment/software/>. Accessed 26 Jan 2018
16. Yamada KD, Tomii K, Katoh K (2016) Application of the MAFFT sequence alignment program to large data-reexamination of the usefulness of chained guide trees. *Bioinformatics* 32:3246–3251. <https://doi.org/10.1093/bioinformatics/btw412>
17. Katoh K, Misawa K, Kuma K, Miyata T (2002) MAFFT: a novel method for rapid multiple sequence alignment based on fast Fourier transform. *Nucleic Acids Res* 30:3059–3066. <https://doi.org/10.1093/nar/gkf436>
18. Higgins D, Sievers F, Dineen D, Wilm A (2014) Clustal W/Clustal X. <http://www.clustal.org/clustal2>. Accessed 26 Jan 2018
19. Larkin MA, Blackshields G, Brown NP, Chenna R, McGettigan PA, McWilliam H, Valentin F, Wallace IM, Wilm A, Lopez R et al (2007) Clustal W and clustal X version 2.0. *Bioinformatics* 23:2947–2948. <https://doi.org/10.1093/bioinformatics/btm404>
20. Protein Data Bank Japan (2018) PDBj. <https://pdbj.org>. Accessed 26 Jan 2018
21. Molecular Organisation and Assembly in Cells (2006) Generating Ramachandran (phi/psi) plots for proteins. http://www2.warwick.ac.uk/fac/sci/moac/people/students/peter_cock/python/ramachandran. Accessed 26 Jan 2018
22. Bartolucci C, Lamba D, Grazulis S, Manakova E, Heumann H (2005) Crystal structure of wild-type chaperonin GroEL. *J Mol Biol* 354:940–951. <https://doi.org/10.1016/j.jmb.2005.09.096>
23. Chaudhry C, Horwich AL, Brunger AT, Adams PD (2004) Exploring the structural dynamics of the E-coli chaperonin GroEL using translation-libration-screw crystallographic refinement of intermediate states. *J Mol Biol* 342:229–245. <https://doi.org/10.1016/j.jmb.2004.07.015>



Chapter 18

Beyond Supersecondary Structure: Physics-Based Sequence Alignment

S. Rackovsky

Abstract

Traditional approaches to sequence alignment are based on evolutionary ideas. As a result, they are prebiased toward results which are in accord with initial expectations. We present here a method of sequence alignment which is based entirely on the physical properties of the amino acids. This approach has no inherent bias, eliminates much of the computational complexity associated with methods currently in use, and has been shown to give good results for structures which were poorly predicted by traditional methods in recent CASP competitions and to identify sequence differences which correlate with structural and dynamic differences not detectable by traditional methods.

Key words Sequence alignment, Amino acid physical properties, Physics-based alignment, Evolution-free alignment, Homology modeling, Protein structure prediction

1 Introduction

Protein sequence alignment is one of the most commonly applied techniques in modern molecular biology. There are two circumstances in which sequence alignment is useful:

- (a) The establishment of an evolutionary relationship between sequences of interest.
- (b) The establishment of a structural relationship between sequences of interest.

It has been recognized since the earliest days of protein science [1] that, in organisms which are evolutionarily related, corresponding proteins will exhibit similar sequences and evolutionary clocks have been constructed based on the analysis of that sequence similarity.

It is also a fundamental tenet of protein structure studies, as currently practiced, that similarity between the structures of two proteins arises from similarity between their sequences. This is the basis for homology modeling [2–4], which remains an important

method for structure prediction. This belief collides with two well-known observations in the field:

- The “distant-homology” problem [5], in which it is observed that pairs of proteins exist which have closely similar structures but have sequences which are not related by any available criterion
- The existence of conformational switches [6], in which a minor change in amino acid sequence leads to a major change in the fold of the molecule

As a result, the processes of sequence comparison and homology modeling have become increasingly convoluted, as algorithms are devised which attempt to circumvent these phenomena. The basic problem, however, cannot be eliminated by algorithmic modifications. It arises from a fundamental confusion as to the definition of sequence similarity. All current methods for homology searching are based on evolutionary considerations [4] and use similarity metrics constructed from predefined sets of sequences which are known to be related. Protein folding, however, is a physics-based phenomenon, and there is no reason to believe that only evolutionarily related molecules will assume any given fold. Only a basic change in the definition of sequence similarity can lead to progress in this field.

In recent work we have developed an approach to sequence comparison, the property factor method (PFM), based on ideas which differ fundamentally from current methods.

- (a) Sequences are defined by the physical properties of the amino acids.
- (b) The degree of similarity between sequences arises solely from the degree to which the physical characteristics of the sequences are similar.
- (c) Because the signals which determine the fold of a protein are distributed throughout the molecule, we have developed alignment methods that are used in which a fundamental length scale (the size of the fragments which are compared) is definable by the investigator. This is in contrast to current algorithms, which are restricted to a single-residue comparison length.

The property factor method was compared to PSI-BLAST, the leading standard alignment algorithm [7]. It was shown that the PFM approach is able to outperform PSI-BLAST when sequence lengths are comparable, using targets from the CASP10 and CASP11 prediction competitions. It was also shown that PFM outperforms PSI-BLAST on a set of informatically challenging targets.

In a further study [8], the PFM was used to compare the sequences of a challenging set of homologs, selected from the CheY-like proteins. The three proteins studied exhibit both structural and functional similarity, but low sequence identity. A set of structural and dynamic differences between CheY and the remaining two proteins in the dataset were correlated with sequence differences measured by PFM—differences which are not indicated by standard sequence comparisons. The reader is referred to ref. 8 for examples of sequence comparison results.

We summarize the comparison approach. It was demonstrated by Kidera et al. [9, 10] that the known physical properties of the 20 naturally occurring amino acids are well represented by a set of 10 property factors, which together encode 86% of the variance of the entire property database (comprising 186 sets of physical properties). We have used this representation extensively in analyses of protein sequence-structure relationships [11–17]. Using these property factors, a protein sequence of length N can be written as a set of ten numerical strings, which together trace the quantitative course of the property factors from N- to C-terminus.

One can view this as a $(10 \times N)$ matrix representation of the sequence of interest, written entirely in terms of the physical properties of the amino acids. Comparison of two sequences of length N , which we denote X and Υ , is then equivalent to the comparison of the corresponding $10 \times N$ matrices \mathbf{X} and $\mathbf{\Upsilon}$. For this purpose we use a fast normalized cross correlation (FNCC) algorithm, adapted from image processing applications. This gives a cosine-line correlation coefficient between the two sequences:

$$\gamma(j) = \frac{\sum_{i=1}^{10} \sum_{m=1}^N [X(i, m) - \bar{X}] [\Upsilon(i-j, m) - \bar{\Upsilon}(j)]}{\left\{ \sum_{i=1}^{10} \sum_{m=1}^N [X(i, m) - \bar{X}]^2 \right\}^{1/2} \left\{ \sum_{i=1}^{10} \sum_{m=1}^N [\Upsilon(i-j, m) - \bar{\Upsilon}(j)]^2 \right\}^{1/2}}.$$

Here $X(i, m)$ is the (i, m) th element of the matrix \mathbf{X} , an overbar denotes an average of all the elements in the matrix in question, and j is a sequence offset index, which indicates the possibility of comparing N -residue fragments which are not in exact positional correspondence. Alignment then consists of stepping through values of j , so that the sequence X is tested against the entire sequence of Υ by comparing with every N -residue subsequence of Υ .

In order to ascribe statistical significance to calculated values of γ , it is necessary to compare those values to the values which would arise from randomly permuted sequences with the same compositions as X and Υ . We have derived analytic formulae for $\langle \gamma \rangle$, the average of γ , and the associated standard deviation, $\sigma(\gamma)$, over the double ensemble of all possible permutations of X and Υ . The

derivation is extremely tedious, and the formulae are not compact, and therefore we do not give the explicit results here. However, the calculations have been implemented numerically. It should be noted that the availability of analytic formulae for these statistical quantities makes it possible to determine a significance for an observed value of γ which depends only on the *specific linear sequences* of the fragments being compared and is independent of their amino acid compositions.

Execution of this algorithm is extremely rapid. All calculations in this area to date, including those on large sequence databases, have been carried out on desktop computers in time frames on the order of minutes to hours.

2 Materials

The “materials” required for this method consist of software and databases. The software for carrying out the sequence 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.
3. **PFM Alignment 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 [9] for the 20 amino acids, together with the numerical identification code for the amino acids.
 - (b) **cheyseq.dat:** This is a file which contains the two sequences to be compared, written in single-letter code.
 - (c) **seqconvert.par:** This is a parameter file which contains the chain lengths of the two sequences being converted by **seqconvert.f**.
 - (d) **seqconvert.f:** This program reads a pair of sequences in single-letter code from **cheyseq.dat** and converts them into numerical values suitable for use by the alignment program, writing to the file **seqconvert.dat**.

- (e) **pfmstat2.par**: This is a parameter file which contains the chain lengths of the two sequences being compared and the length N of the sequence fragment of interest in computing γ .
- (f) **pfmstat2.f**: This program reads the sequences to be compared from **seqconvert.dat** and calculates values of γ , $\langle \gamma \rangle$, and $\sigma(\gamma)$. The results are given for every alignment over a range specified by the user, as values of γ and of the Z function

$$Z = [\gamma - \langle \gamma \rangle] / \sigma(\gamma)$$

which indicates the statistical significance of the value of γ relative to the ensemble of randomly permuted sequences.

3 Methods

1. Place all files in the same directory, in order to avoid the use of long pathnames.
2. Compile the program files (**seqconvert.f**, **pfmstat2.f**).
3. Place the sequences of interest in the file **cheyseq.dat**.
4. Run the program **seqconvert.f**, giving the output file **seqconvert.dat**, which contains the sequence data in a format ready to be read by the alignment program.
5. Run the program **pfmstat2.f**, which gives the output file **pfmstat2.dat** containing the sequence alignment data.

4 Notes

1. The file **cheyseq.dat** is designed to read sequences up to 132 residues in length. The programs are readily adapted to handle longer chain lengths if necessary. The format for this file in the current version is as follows:
 - Line 1: Name of the first sequence, alphanumeric, up to 80 characters.
 - Line 2: Single-letter specification of the first sequence, up to 132 residues.
 - Line 3: Blank.
 - Line 4: Name of the second sequence, alphanumeric, up to 80 characters.
 - Line 5: Single-letter specification of the second sequence, up to 132 residues.

2. The parameter file **seqconvert.par** should have two lines. Each line contains the chain length of the corresponding (first/second) sequence, in I3 format.
3. The file **pfmstat2.par** should have three lines. The first two lines contain the chain lengths of the corresponding (first/second) sequences, in I3 format. The third line contains the length of the chain fragments to be compared, also in I3 format.

References

1. Dayhoff MO, Schwartz RM, Orcutt BC (1978) A model of evolutionary change in proteins. In: *Atlas protein sequence and structure*, vol 5. National Biomedical Research Foundation, Washington, DC, pp 345–352
2. Lewis PN, Scheraga HA (1971) Prediction of structural homology between bovine α -lactalbumin and hen egg white lysozyme. *Arch Biochem Biophys* 144:584–588
3. Warme PK, Momany FA, Rumball SV, Scheraga HA (1974) Computation of structures of homologous proteins. α -Lactalbumin from lysozyme. *Biochemistry* 13:768–782
4. Saxena A, Sangwan RS, Mishra S (2013) Fundamentals of homology modeling steps and comparison among important bioinformatics tools: an overview. *Forensic Sci Int* 1:237–252
5. Ben-Hur A, Brutlag D (2003) Remote homology detection: a motif based approach. *Bioinformatics* 19(Suppl. 1):i26–i33
6. Alexander PA, He Y, Chen Y, Orban J, Bryan PN (2009) A minimal sequence code for switching protein structure and function. *Proc Natl Acad Sci U S A* 106:21149–21154
7. He Y, Rackovsky S, Yin Y, Scheraga HA (2015) Alternative approach to protein structure prediction based on sequence similarity of physical properties. *Proc Natl Acad Sci U S A* 112:5029–5032
8. He Y, Maisuradze GG, Yin Y, Kachlishvili K, Rackovsky S, Scheraga HA (2017) Sequence-, structure- and dynamics-based comparisons of structurally homologous CheY-like proteins. *Proc Natl Acad Sci U S A* 114:1578–1583
9. Kidera A, Konishi Y, Oka M, Ooi T, Scheraga HA (1985) Statistical analysis of the physical properties of the 20 naturally occurring amino acids. *J Protein Chem* 4:23–55
10. Kidera A, Konishi Y, Ooi T, Scheraga HA (1985) Relation between sequence similarity and structure similarity in proteins: role of important properties of amino acids. *J Protein Chem* 4:265–297
11. Rackovsky S (1998) “Hidden” sequence periodicities and protein architecture. *Proc Natl Acad Sci U S A* 95:8580–8584
12. Rackovsky S (2006) Characterization of architecture signals in proteins. *J Phys Chem B* 110:18771–18778
13. Rackovsky S (2009) Sequence physical properties encode the global organization of protein structure space. *Proc Natl Acad Sci U S A* 106:14345–14348
14. Rackovsky S (2011) Spectral analysis of a protein conformational switch. *Phys Rev Lett* 106:248101
15. Rackovsky S (2013) Sequence determinants of protein architecture. *Proteins* 81:1681–1685
16. Scheraga HA, Rackovsky S (2015) Nonlinearities in protein space limit the utility of informatics in protein biophysics. *Proteins* 83:1923–1928
17. Scheraga HA, Rackovsky S (2016) Global informatics and physical property selection in protein sequences. *Proc Natl Acad Sci U S A* 113:1808–1810



Chapter 19

Secondary and Supersecondary Structure of Proteins in Light of the Structure of Hydrophobic Cores

Mateusz Banach, Leszek Konieczny, and Irena Roterman

Abstract

The traditional classification of protein structures (with regard to their supersecondary and tertiary conformation) is based on an assessment of conformational similarities between various polypeptide chains and particularly on the presence of specific secondary structural motifs. Mutual relations between secondary folds determine the overall shape of the protein and may be used to assign proteins to specific families (such as the immunoglobulin-like family). An alternative means of conducting structural assessment focuses on the structure of the protein's hydrophobic core. In this case, the protein is treated as a quasi-micelle, which exposes hydrophilic residues on its surface while internalizing hydrophobic residues. The accordance between the actual distribution of hydrophobicity in a protein and its corresponding theoretical ("idealized") distribution can be determined quantitatively, which, in turn, enables comparative analysis of structures regarded as geometrically similar (as well as geometrically divergent structures which are nevertheless regarded as similar in the sense of the fuzzy oil drop model). In this scope, the protein may be compared to an "intelligent micelle," where local disorder is often intentional and related to biological function—unlike traditional surfactant micelles which remain highly symmetrical throughout and do not carry any encoded information.

Key words Tertiary structure, Supersecondary structure, Secondary structure, Hydrophobicity, Water environment, Hydrophobic core, Micelle

1 Introduction

No discussion of the supersecondary structure of proteins would be complete without a reference to publicly available structural databases, including CATH [1, 2], SCOP [3, 4], and Pharm [5, 6]. These databases tend to follow the same similarity metrics, rooted in molecular geometry and dependent on the placement of specific atoms (such as C-alpha). They differ, however, in terms of the criteria used to compare secondary and supersecondary folds.

CATH (comprehensive structural and functional annotations for genome sequences) assigns proteins to four distinct categories: class, based on the presence of secondary structural motifs in the analyzed domain; architecture, structural similarity (without regard

to homology); topology/fold, identification of structural forms; and homologous superfamily, acknowledging the homologies between proteins.

SCOP (structural classification of proteins) applies the following classification scheme: (1) all alpha proteins; (2) all beta proteins; (3) alpha and beta proteins in two forms (a/b) and (4) (a + b); (5) multi-domain proteins (alpha and beta); (6) membrane and cell surface proteins and peptides; (7) small proteins; (8) coiled-coil proteins; (9) low-resolution protein structures; (10) peptides; (11) designed proteins; and (12) artifacts.

The Pfam database exploits multiple sequence alignment and hidden Markov models to define distinct protein families. The following classification is applied:

- Family: a collection of related protein regions.
- Domain: a structural unit.
- Repeat: a short unit which is unstable in isolation but forms a stable structure when multiple copies are present.
- Motifs: a short unit found outside globular domains.
- Coiled coil: regions that predominantly contain coiled-coil motifs, regions that typically contain alpha-helices that are coiled together in bundles of 2–7.
- Disordered: regions that are conserved yet are either shown or predicted to contain bias sequence composition and/or are intrinsically disordered (non-globular).

An interesting phenomenon formerly acknowledged by the PDB database [7, 8] is the notion of “new fold” proteins with peculiar topologies [8]. Identification of structures which eluded attempts at classification forced the authors of classification schemes to revise their models, eventually leading to a situation where no forms are considered as being “outside” of the given scheme. Such unusual conformations included solenoids (also referred to as the beta-helix; *see*, for example, 2PEC), beta-propellers (e.g., 2BAT), or the so-called horseshoe structure (e.g., 1BNH) [9].

The aim of this chapter is to propose an alternative structural classification of proteins, based on the structure of their hydrophobic cores without regard to the actual shape (secondary and/or supersecondary structure) adopted by each protein. Our criterium reflects the mandatory presence of the aqueous solvent without which proteins cannot express their biological activity. Here, the structure of the protein’s hydrophobic core is viewed as a result of the solvent acting upon the protein chain and guiding the folding process—in other words, the protein itself behaves like a micelle, albeit a nonsymmetrical one (unlike micelles composed of identical unit molecules, which often exhibit pronounced symmetry and tend to adopt spherical or cylindrical shapes). The structure of

this “quasi-micelle,” formed by the polypeptide chain, may become quite complex as a result of variable hydrophobicity exhibited by individual amino acids and also due to the presence of covalent bonds between residues, limiting the number of their degrees of freedom. Consequently, the polypeptide chain cannot produce a perfectly symmetrical (spherical or cylindrical) structure and very frequently (in majority) includes certain deformations which can be treated as encoded information. The protein therefore becomes an “intelligent micelle” [10, 11].

Quantitative assessment of the degree of structural ordering (or disorder) enforced by the aqueous environment is possibly by referring to the 3D Gaussian, which represents the idealized (theoretical) distribution of hydrophobicity in a protein—with a highly hydrophobic center and low-surface hydrophobicity, enabling entropically advantageous contact with the surrounding water. The 3D Gaussian is also a highly symmetrical distribution, reflecting conditions which exist in a surfactant micelle.

For these reasons the fuzzy oil drop model can be used to determine the accordance between the observed and theoretical distribution of hydrophobicity for a given protein and the degree to which the protein deviates from theoretical values (implying the presence of encoded information). It turns out that such accordance/discordance is frequently independent of the secondary and supersecondary structure of the protein under consideration. The degree of discordance is actually a measure of the amount of information carried by the structure making it “intelligent micelle” in contrast to surfactant micelle which as highly symmetrical (high degree of determination) makes deprived of any form of information.

The proteins selected in this chapter are expected to convince the reader to accept this interpretation.

2 Materials

The presented model assumes the form of a bioinformatics software package. The entire research process represents the *in silico* approach (*see* the description in Subheading 3). The program takes as input the 3D conformations of proteins available in the Protein Data Bank [8]. Computations are strongly dependent on the intrinsic hydrophobicity of each amino acid residue; however, any available scale may be applied [12, 13]. Comparative analysis of a scale which depends on the position of each residue in the protein chain [12] and of the Kyte-Doolittle scale [12, 13] indicates that results are largely similar, with no major qualitative differences.

When analyzing the 3D conformation of proteins, we may disregard H atoms since the model depends on the positions of the so-called effective atoms, i.e., geometric centers of each residue

forming the input chain. This simplification is justified by the fact that intrinsic hydrophobicity applies to the residue as a whole, and not to individual atoms or groups.

Lack of specific information concerning chain fragments (i.e., inability to determine the correct fold for a given fragment such as an uncoiled loop) does not restrict the model's applicability. Excluding an uncoiled loop from computations does not preclude analysis of the compact globular portion of the protein—and may occasionally prove beneficial. Note that hydrophobic core analysis may involve the entire protein chain or be limited to specific fragments, depending on the researcher's intentions (although—of course—exclusion of any fragment from analysis should be properly justified).

3 Methods

The fuzzy oil drop model has been thoroughly described in numerous publications [12]; however, given the scope of the publishing series and its focus on methodologies, we will again provide a broader description of the model's theoretical underpinnings.

The fuzzy oil drop model [14] may be treated as an evolution of Kauzmann's original oil drop model [15], which posited the presence of a centrally located hydrophobic core encapsulated in a hydrophilic shell. Kauzmann's model itself was based on observing the behavior of an oil drop immersed in water—in an attempt to minimize its surface area, the hydrophobic drop adopts a spherical shape. In contrast to an oil drop, proteins contain polar residues and may expose them on the surface, thus minimizing (or even eliminating) contacts between hydrophobic residues and the solvent. This phenomenon gives rise to an outer layer—whether complete or partial—separating the hydrophobic core from water.

Kauzmann's original model was discrete and recognized only two possible states: hydrophilic (observed in the outer layer) and hydrophobic (observed in the core). In contrast, the fuzzy oil drop model introduces a continuous gradient of hydrophobicity, peaking at the center of the protein body and then gradually sloping to near-zero values on the surface. This distribution may be mathematically expressed by a 3D Gaussian form.

A 3D Gaussian form is superimposed onto the protein molecule in such a way as to encapsulate its entire volume. This is done by adjusting the σ coefficients of the Gaussian—according to the so-called three-sigma rule, values of the Gaussian can be assumed as equal to 0 at a distance of 3σ along each direction from the center of the distribution. Thus, the σ coefficients indirectly determine the volume of the protein. The geometric center of the protein is assumed to coincide with the origin of the coordinate system, and then the molecule is aligned in such a way that the greatest

separation which exists between any two effective atoms determines one of the principal axes of the coordinate system (e.g., X). Having aligned the molecule along the X axis, the positions of effective atoms are projected on the YZ plane, and, again, the greatest separation between any two atoms is identified in order to orient the molecule along the second axis (in this case— Y). Finally, the greatest and lowest values of Z are calculated, yielding sufficient data to determine all three σ coefficients (as $1/6$ of the maximum distance between any two effective atoms along each axis).

When calculating σ values, the volume of the protein body may be slightly increased in each direction (e.g., by 3, 4, or 5 Å). This is done to derive an ellipsoid which encompasses all atoms belonging to the molecule—including those which do not coincide with its effective atoms.

Having computed all σ coefficients, we may proceed with the calculation of the protein's theoretical (idealized) hydrophobicity distribution. This is done simply by computing the value of the Gaussian for each effective atom. This distribution, denoted T , expresses the expected hydrophobicity of each residue given its position within the ellipsoid capsule. A distinct value of T_i is assigned to each effective atom i , as expressed by Eq. 1.

$$\tilde{H}t_j = \frac{1}{\tilde{H}t_{\text{sum}}} \exp\left(\frac{-(x_j - \bar{x})^2}{2\sigma_x^2}\right) \exp\left(\frac{-(y_j - \bar{y})^2}{2\sigma_y^2}\right) \exp\left(\frac{-(z_j - \bar{z})^2}{2\sigma_z^2}\right) \quad (1)$$

The value of Gaussian function calculated for any point (in the capsule) expresses the distribution of probability in an ellipsoid capsule, the size of which is determined by the values of σ along each cardinal direction. If we fine-tune these σ values in such a way that the Gaussian envelope completely encapsulates the molecule, then the value of the function will express the expected (theoretical) distribution of hydrophobicity at each point within the protein body. The parameters $\bar{x}, \bar{y}, \bar{z}$ reflect the placement of the center of the ellipsoid—at the origin of the coordinate system, all three values become equal to 0. σ values are calculated as $1/3$ of the greatest distance between an effective atom and the origin of the system once the molecule has been oriented in such a way that its greatest spatial extension coincides with a system axis (for each axis separately).

The $1/Ht_{\text{sum}}$ coefficient ensures normalization of both distributions (empirical and theoretical) and therefore enables comparative analysis. The point j th represents any point in ellipsoid; however, only positions of effective atoms are taken into account.

The theoretical distribution is not necessarily realized by any actual protein. Rather, we have to deal with the so-called observed distribution (denoted O), which depends on local hydrophobic

interactions between residues, i.e., the intrinsic hydrophobicity of each residue and their mutual separation.

The observed distribution may be calculated for each residue by applying Levitt's formula [16]:

$$\tilde{H}_{0j} = \frac{1}{\tilde{H}_{0\text{sum}}} \sum_{i=1}^N (H_i^r + H_j^r) \begin{cases} \left[1 - \frac{1}{2} \left(7 \left(\frac{r_{ij}}{c} \right)^2 - 9 \left(\frac{r_{ij}}{c} \right)^4 + 5 \left(\frac{r_{ij}}{c} \right)^6 - \left(\frac{r_{ij}}{c} \right)^8 \right) \right] & \text{for } r_{ij} \leq c \\ 0 & \text{for } r_{ij} > c \end{cases} \quad (2)$$

In Eqs. 1 and 2, the j index denotes the averaged position of atoms present in j th residue. H_{0j} is the collected value describing the interactions with the neighboring residues (labeled with the i index) at distances not greater than 9 Å (this distance— c —is treated as the cutoff value for the hydrophobic interactions, following the original model [16]). Applying a cutoff value implies that hydrophobic interactions are considered local and depend on the location of each residue. This function is empirically determined and, according to [16], expresses the strength of hydrophobic interactions. H_i^r and H_j^r represent the hydrophobicity parameter (constant for each residue) ascribed to each amino acid using a predetermined scale, which can be arbitrary (in our study, the relevant scale is derived from [12, 13]). The r_{ij} is the distance between the i th and the j th residue, while N is the total number of residues in the chain.

Following [16] we apply a cutoff distance of 9 Å, beyond which no hydrophobic interactions are expected to occur.

The above calculations result in a pair of values for each residue: T_i and O_i . In order to facilitate meaningful comparisons, these two distributions must be normalized—consequently, each value of T_i is divided by the sum of all T_i for the given chain (and likewise for O_i).

While assessment of the similarity of any two input distributions may be based on subjective criteria, the divergence entropy formula [17] enables us to provide an objective comparison.

$$D_{\text{KL}}(p|p^0) = \sum_{i=1}^N p_i \log_2(p_i/p_i^0) \quad (3)$$

where p_i is the probability at the point i th; p_i^0 , probability value in target distribution; and N , number of points.

It is, however, important to note that D_{KL} is a measure of entropy, and therefore its value cannot be interpreted on its own. To properly assess the status of the input chain, we require two reference distributions. Since T is already available as a boundary case which corresponds to a perfect monocentric distribution (similar to that observed in a surfactant micelle), we now proceed with the introduction of the opposite boundary case—referred to as unified (R)—where each residue is assigned the same value of

hydrophobicity ($1/N$ where N is the number of residues in the chain). The result is a distribution devoid of any concentration of hydrophobicity at any point in the protein body.

Having defined the two boundary cases, we can now determine whether the observed distribution (O) is more closely aligned with T or with R . This is done by computing D_{KL} for the O/T

$$O | T = \sum_{i=1}^N O_i \log_2(O_i / T_i) \quad (4)$$

and O/R

$$O | R = \sum_{i=1}^N O_i \log_2(O_i / R_i) \quad (5)$$

relations, respectively. The protein is assumed to match T when the distance between T and O is lower than the corresponding distance between O and R (and the other way around). In order to avoid having to deal with two separate distance values, we introduce the so-called relative distance parameter, given as

$$RD = O | T / (O | T + O | R) \quad (6)$$

$RD > 0.5$ is indicative of the observed distribution deviating in favor of R , i.e., the lack of a prominent hydrophobic core. The degree of this deviation is given directly by the value of RD, with higher values corresponding to greater deviation. The distribution representing the intrinsic hydrophobicity of each residue can also be taken as reference distribution. Thus we distinguish two forms of RD parameter: $T-O-R$ and $T-O-H$ relations.

Figure 1 provides a visual representation of RD.

The above description presents one of the possible variants of fuzzy oil drop analysis, carried out for a single-protein molecule. Similar analysis may, however, be performed for a protein complex (quaternary structure) as long as the 3D Gaussian encapsulates the complex as a whole. In this case, we can inquire whether the complex contains a shared hydrophobic core formed by its participating protein chains [18].

Many other potential modifications are possible in the context of FOD analysis. For example, having assessed the status of the complex, we may inquire about the status of any individual chain belonging to that complex. In order to do so, we need to again normalize a specific fragment of the distribution (corresponding to the given chain). This normalization is then followed by calculation of a new RD parameter expressing the status of the unit chain and its contribution to the shared hydrophobic core in the complex. We may also determine the status of smaller structural units—including supersecondary or secondary folds—in the context of arbitrary higher-tier structures (complexes or chains).

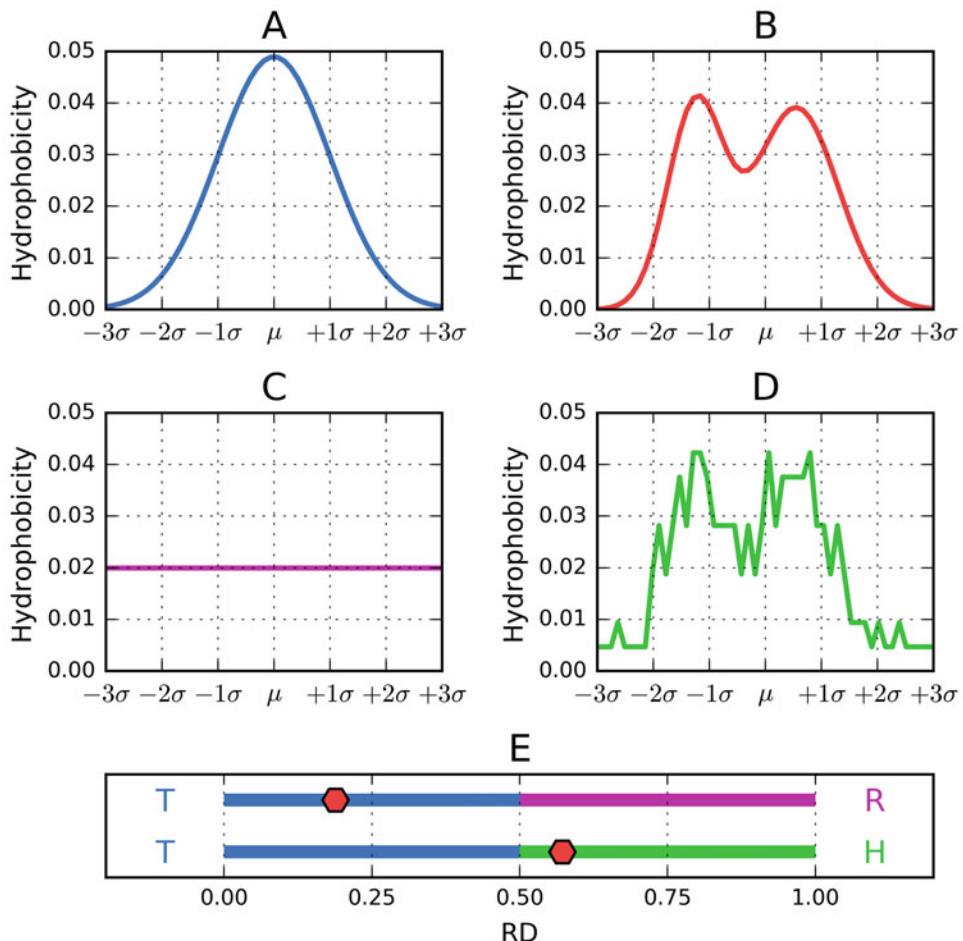


Fig. 1 Graphic interpretation of the fuzzy oil drop-based parameters. (a) Theoretical distribution (T) expressed by Gaussian function (for simplicity presentation is reduced to one dimension). (b) Observed distribution (O). (c) Unified distribution where all residues represent same hydrophobicity level. (d) Distribution based on the intrinsic hydrophobicity (H) of each residue. (e) Interpretation of RD values: Upper line—red dot visualizes the status of the O distribution for the relation $T-O-R$ and Lower line—red dot visualizes the status of O distribution for the relation $T-O-H$. The interpretation of the RD values is as follows: $T-O-R$, the O distribution is accordant with the T distribution (closer to T than to R); $T-O-H$, the O distribution is similar to H distribution—the intrinsic residual hydrophobicity influences the O distribution

Real-world proteins do not always follow the theoretical distribution, even though the presence of a hydrophobic core is an important determinant of tertiary structural stability (along with disulfide bonds) and regardless of the fact that interaction with water would seem to favor this type of distribution.

Deviations between T and O may be either local or global. As already remarked, when dealing with higher-tier structures, we may consider each chain (or chain fragment) separately and determine whether it contributes to the shared hydrophobic core. In the

course of this analysis, we may successively eliminate discordant residues until the remainder of the input chain is found to be consistent with the theoretical model. This allows identification of parts of the chain responsible for structural stabilization (commonly taking the hydrophobic core as tertiary structure stabilization) and which ones disrupt this stabilization—as it turns out, often in a targeted and intentional manner. The degree to which the chain diverges from its corresponding theoretical distribution may be regarded as a measure of the quantity of information it encodes.

The notion of “global discordance” between O and T usually refers to an entirely different structural ordered pattern which does not involve a monocentric hydrophobic core.

All examples discussed in this section are also presented in detail in Subheading 4, which illustrates (among others) the theoretical, observed, and intrinsic distributions of hydrophobicity for various types of proteins.

Biological activity is an important concept in protein studies. We intend to reveal the relation between the structure of the hydrophobic core (or, more specifically, between its specific asymmetry) and the location of active sites such as substrate-binding cavities. Biological activity may also refer to the presence of a ligand which is permanently bound to the protein—the effect which such ligands exert upon the protein’s hydrophobic core explains the binding mechanism for nonprotein ligands as well as for external proteins.

4 Results

This part of the publication is intended as an introduction to problems which may emerge when applying the fuzzy oil drop model to assess the structure of the hydrophobic core. Rather, however, than enumerating these problems, we have chosen to present a selection of varied proteins and try to determine the structure of the hydrophobic cores, taking into account secondary and supersecondary structural variability. This section is not devoted to any specific biological phenomenon—instead, it outlines the diversity of structures to which the fuzzy oil drop model can be applied, along with an interpretation of results. Note that this subject has already been addressed in [19] where, based on textbook data [20], we subjected various supersecondary folds to FOD analysis.

4.1 Proteins Where O Remains Consistent with T

2OTR provides an example of a micelle-like protein. This structure, synthesized and crystallized in the framework of the Structural Genome Project [21], is a member of a family of proteins involved in plasmid stabilization (although its specific function has not yet

Table 1
RD values for the entire protein, for individual secondary folds, and for selected supersecondary structures present in 2OTR

Secondary str.	Fragment	RD
	1–90	0.393
Beta	3–5	0.657
Helix	7–22	0.425
Helix	24–37	0.318
Beta	47–49	0.786
Loop	50–55	0.661
Beta	56–62	0.393
Beta	65–72	0.246
Beta	76–84	0.409
Helix	85–90	0.315
Beta-sheet	All beta above	0.376
Helices	All helices above	0.392

been unambiguously established). According to the CATH classification, its supersecondary structure is described as an alpha-beta two-layer sandwich.

In light of the fuzzy oil drop model, this structure is highly consistent with the theoretical distribution of hydrophobicity, as confirmed by its RD coefficient and by direct comparison of T and O .

Despite overall strong consistency between T and O , suggesting micelle-like ordering (Table 1), there are also local deviations, exhibited by two short beta folds and the loop at 50–55, as presented in Fig. 2. One shall note that the discordant fragments are highly exposed to environment. Thus their discordance may be the effect of dynamics of these fragments.

In the authors' experience, the type of titin immunoglobulin domain is among the most accordant structures, both with regard to the entire protein and to its individual beta folds as well as the beta-sheet system [22].

4.2 Proteins Exhibiting Local Deviations from T

In some proteins local disagreements between the observed distribution of hydrophobicity and the theoretical Gaussian form are caused by the presence of a ligand, an enzymatic active site, or local structural adjustments associated with bound ions, some of which exhibit a high degree of coordination and thus enforce a specific local distribution.

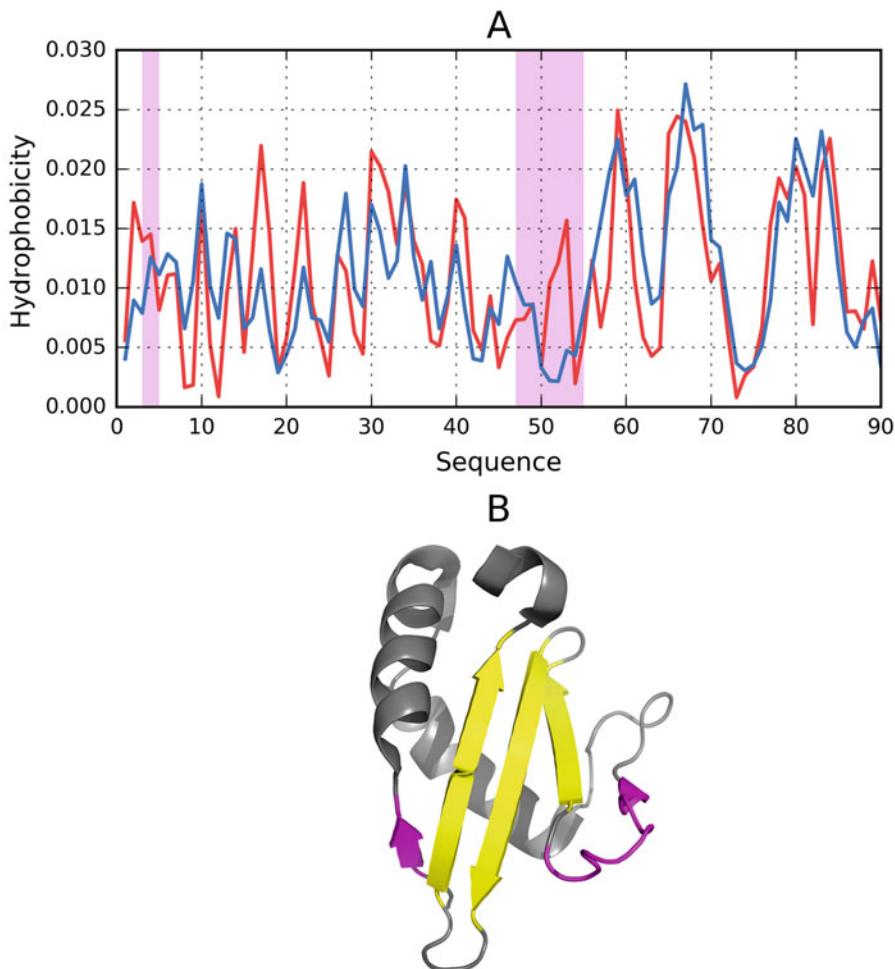


Fig. 2 Status of 2OTR. **(a)** Theoretical (T) and observed (O) distributions of hydrophobicity for 2OTR, revealing good agreement between both distributions despite the presence of short-discordant fragments (distinguished as pink). **(b)** 3D presentation with locally discordant fragments highlighted (pink fragments)

4.2.1 Ligand-Binding Cavity: Ion/Enzyme

These properties are evident in 1Y3X-trypsin inhibitor complex: Ec. 3.4.21.4 [23]. The structure listed in PDB also includes a ligand—UIB—(1r,3as,4r,8as,8br)-4-{5-(Phenyl[1,3]dioxol-5-Ylmethyl)-4-Ethyl-2,3,3-Trimethyl-6-Oxo-Octahydro-Pyrrolo[3,4-C]pyrrol-1-Yl}-Benzamidine, along with a Ca^{2+} ion. The protein is composed of two domains. We have performed an analysis for the complete chain, for each domain, and for the substrate-binding cavity while also determining the effects exerted by the ion (Ca^{2+}) and the ligand. The protein includes five catalytic residues.

1Y3X is also characterized by the presence of numerous covalent disulfide bonds, which significantly alter the shape of the molecule and the structure of its hydrophobic core.

Table 2 provides a summary of the FOD status of 1Y3X.

Table 2
Parameters describing 1Y3X and the secondary and supersecondary structural units

1Y3X	Fragment	Secondary structure	RD
Complete molecule	16–245	Two beta-barrels Ligand binding No ligand binding Ion binding No ion binding	0.511 0.618 0.499 0.818 0.509
2xE	22–157	SS bond	0.502
3xE	42–58	SS bond	0.318
3xE	127–232	SS bond	0.552
	136–201	SS bond	0.564
	168–182	SS bond	0.469
3xE	191–220	SS bond	0.493
Domain 1			
Domain 1	16–27, 141–232 20–22 134–140 155–162 164–172 176–183 197–201 204–215 226–231 Beta folds together 22–157 136–201 168–182 191–220 175, 189, 190, 192, 214–216, 219, 220 193–196	Beta-barrel Beta Beta Beta Helix Beta Beta Beta Beta Beta Beta Beta Beta Beta Beta Beta Ligand binding No ligand binding Catalytic residues No catalytic residues	0.469 0.449 0.509 0.493 0.390 0.261 0.611 0.308 0.318 0.457 0.513 0.433 0.240 0.328 0.526 0.443 0.526 0.467
Domain 2			
Domain 2	28–120, 233–245 29–37 39–48 50–54 55–59 63–69 81–91 103–109 234–244 Beta folds together	Beta barrel Beta Beta Beta Helix Beta Beta Beta Beta Beta	0.443 0.267 0.481 0.032 0.359 0.163 0.474 0.206 0.292 0.322

(continued)

Table 2
(continued)

1Y3X	Fragment	Secondary structure	RD
	42–58	SS bond	0.391
		Ligand binding	0.899
		No ligand binding	0.412
		Ion binding	0.887
		No ion binding	0.414
		No catalytic residues	0.443

The RD > 0.5 is given in bold. The number of E in left column—number of catalytic residues in given fragment

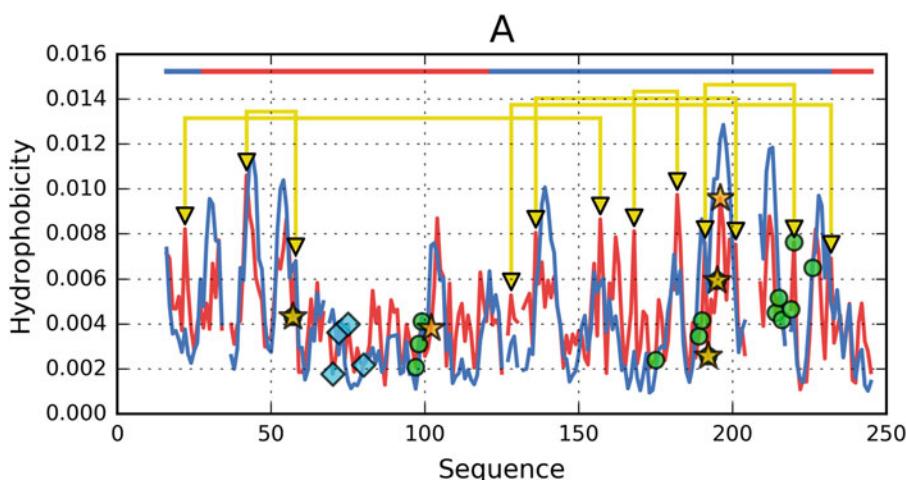


Fig. 3 Profiles of T (blue) and O (red) distributions in 1Y3X visualizing the similarity/dissimilarity in two-domain chains (domains distinguished by horizontal line on top). The positions of disulfide bonds are shown as yellow clamps

Interpretation of the structure of the hydrophobic core in this molecule is based on RD values listed in Table 2 and on a direct comparison between T and O (Figs. 3 and 4).

The molecule as a whole lacks a clearly defined monocentric hydrophobic core. Elimination of residues responsible for ligand binding and complexation lowers the overall RD value, indicating that these residues cause local discordances (note also the high values of RD calculated specifically for such residues).

Analysis of disulfide bonds reveals that the fragments which include bonds 42–58, 168–182, and 191–220 are consistent with the model, while the remaining SS fragments diverge from it.

Both beta-barrel domains follow the theoretical distribution of hydrophobicity, which means that a monocentric hydrophobic core is present in each domain. The aggregate status of beta folds in each domain appears consistent with theoretical expectations even

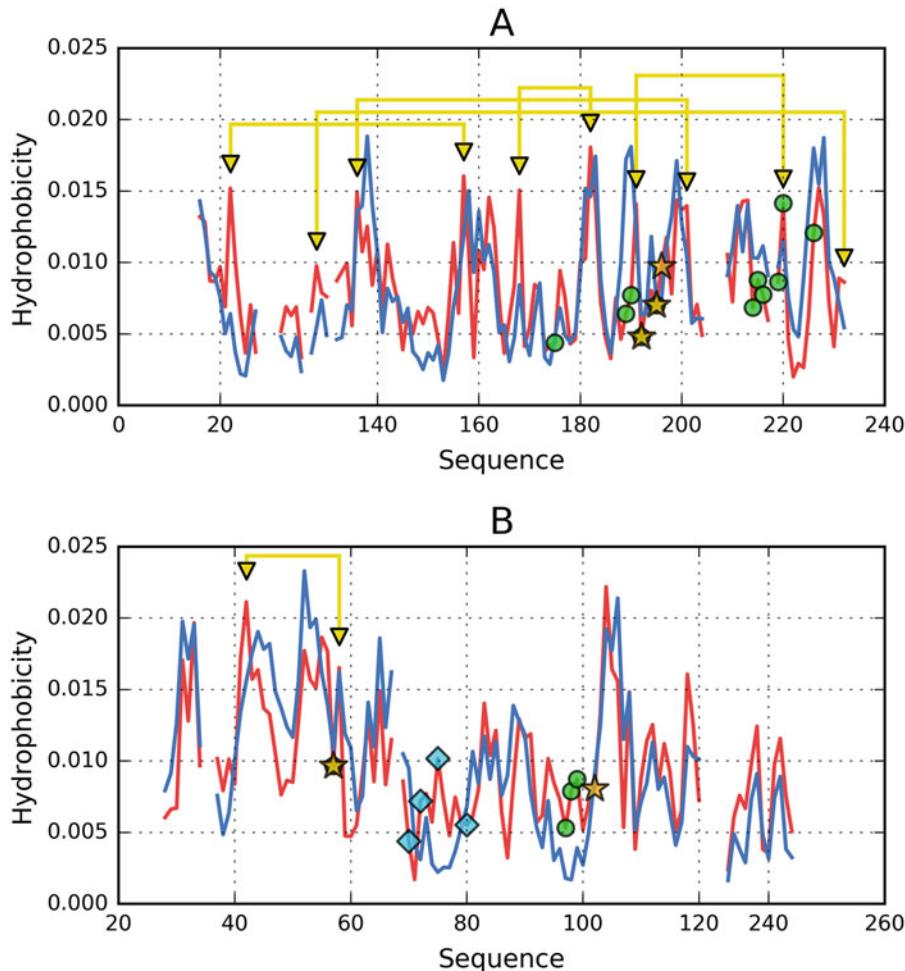


Fig. 4 The T and θ distributions calculated for each domain individually in 1Y3X. (a) Domain 1 distinguished as red in Fig. 3. (b) Domain 2 distinguished as blue in Fig. 3

though some fragments exhibit $RD > 0.5$. Eliminating catalytic and ligand-/ion-binding residues lowers the RD value, confirming that the peptide is locally adapted to the presence of external molecules (including its intended substrate). Disulfide bonds appear to promote structural ordering in each domain.

Trypsinogen (1Y3X) is a good example of a molecule where alignment between the theoretical and observed distributions of hydrophobicity is locally disrupted by a number of factors. Table 2 reveals that RD decreases following elimination of residues which mediate interactions with ligands and ions, as well as residues which form the protein's catalytic active site (Fig. 5).

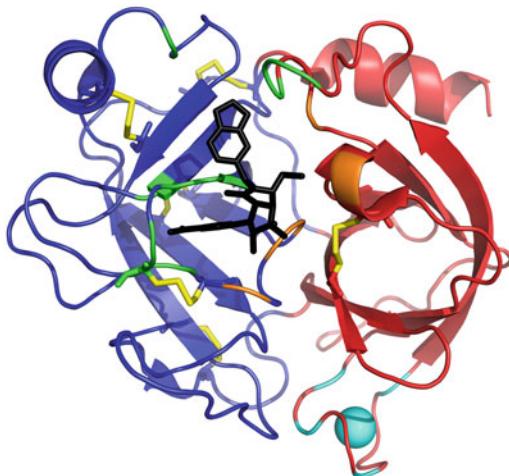


Fig. 5 The 3D structure of the complete chain with two domains (distinguished as red and blue) with SS bonds (yellow). Ligand present in this complex shown as black. The position of Ca^{2+} turquoise ball. The residues engaged in enzymatic activity shown in green. SS bonds—yellow. Domain 1, red; domain 2, blue

4.2.2 Protein-Protein Complexation Interface

The immunoglobulin-like domain which forms part of 1HNG-A02 provides an example of the effects of complexation interfaces upon the structure of the hydrophobic core [24]. This protein, as listed in PDB, is a homodimer consisting of two chains, each of which is further composed of two domains. Our analysis concerns the A2 domain. Taken as a whole, this domain (which is a beta-sandwich, according to CATH) has RD = 0.483, while the RD value computed solely for interface residues is 0.676. Elimination of residues responsible for P–P complexation reduces the RD value to 0.460. This highlights the influence of the adjacent chain, where the local (domain-wide) hydrophobic core is distorted in the process of complexation (Fig. 6).

The beta-sandwich—particularly when appearing as the so-called immunoglobulin-like domain—has been the focus of numerous scientific studies, owing to its ubiquity and functional variability of molecules in which it participates. Much effort has gone into predicting the supersecondary conformation of these structures given their composition [25–27]. It is also worth noting that immunoglobulin-like domains exhibit a high degree of geometric similarity despite major differences in the status of their hydrophobic cores [22].

4.2.3 Disulfide Bond System

The influence of disulfide bonds upon the hydrophobic core structure has already been discussed in the context of 1Y3X. Another interesting protein, 1C01 [28], is a beta-sandwich, which, despite its limited chain length (76), contains three disulfides.

Results listed in Table 3 and Fig. 7 suggest that disulfide bonds play an important role in structural stabilization. None of the

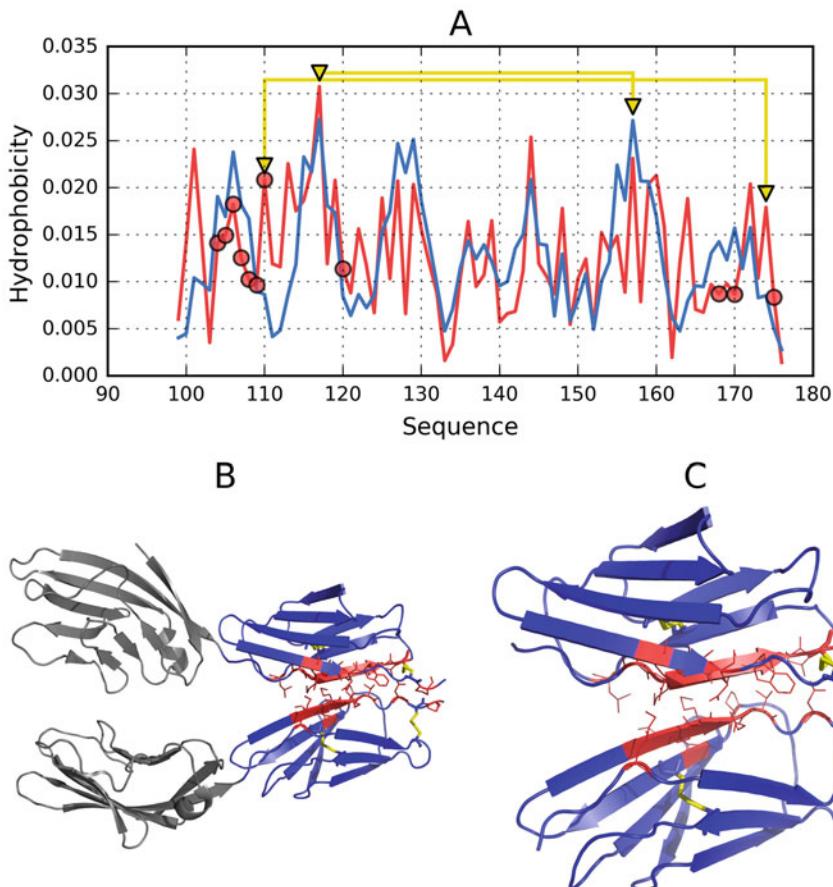


Fig. 6 The status of the domain A02 (chain A domain 2) in 1HNG-T-lymphocyte adhesion glycoprotein CD2. **(a)** Distribution *T* (blue) and *O* (red) in the domain. The residues engaged in P-P interaction distinguished by red circles. It visualizes the local discordance of status of these residues. SS bonds shown as yellow clamps. **(b)** 3D presentation of the complete dimer. Only blue domains are discussed. The residues engaged in P-P interaction distinguished as red. **(c)** domain A02 in form of dimer. Residues distinguished in red - residues engaged in P-P interaction

fragments bracketed by CYS residues which participate in SS bonds are consistent with the theoretical distribution. This indicates that disulfides may stabilize the structure in a state which deviates from the protein's natural tendency to follow the theoretical distribution of hydrophobicity. Such local deviations, reinforced by disulfides, are likely related to the protein's function and subject to strict control, even though they do not reflect the theoretical distribution.

A protein which retains high accordance with the FOD model in spite of disulfide bonds has been discussed in [29]. The protein in question is 1QLL—neurotoxin myotoxin bothropstoxin phospholipase (mainly alpha up-down bundle, according to CATH),

Table 3

RD values for 1C01—a beta-sandwich which contains three disulfide bonds despite its limited length (76 aa)

Fragment	Secondary	RD	Comments
1–76	Beta-sandwich	0.552	
2–6	Beta I	0.782	
7–15	Loop	0.306	
16–18	Beta I	0.044	
19–21	Loop	0.558	
22–26	Beta II	0.727	SS bond
27–30	Loop	0.401	
31–35	Beta I	0.514	
36–39	Loop	0.451	
40–45	Beta II	0.272	
46–50	Helix	0.393	SS bond
52–57	Loop	0.847	SS bond
58–69	Beta II	0.458	
70–75	Beta II	0.354	
	Beta I	0.490	
	Beta II	0.507	
	Beta I + II	0.511	
	SS bonds		
11–64		0.550	
21–76		0.529	
23–49		0.508	

which consists of 127 residues and contains no fewer than 7 disulfide bonds, with all fragments bracketed by these bonds remaining consistent with the theoretical distribution of hydrophobicity. Notably, the protein chain as a whole is also consistent with the model.

4.3 Supersecondary Structures and the Fuzzy Oil Drop Model

In this section we present a protein consisting of three domains, each of which represents a different supersecondary structure (according to CATH), mainly alpha orthogonal bundle, mainly beta-beta barrel, and mainly alpha up-down bundle, respectively. The sample protein is human glutaryl-CoA dehydrogenase (1SIQ) E.C. 1.3.8.6—glutaryl-CoA dehydrogenase (ETF) [30]. Only one

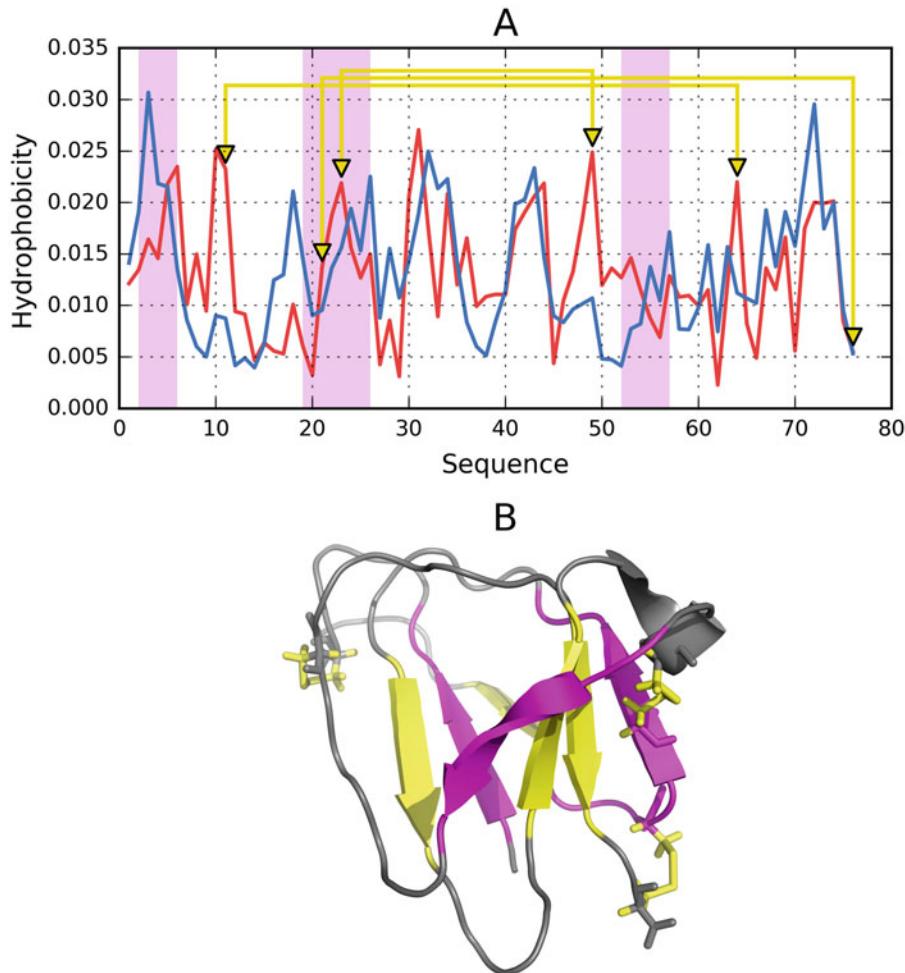


Fig. 7 Status of 1C01. (a) *T* (blue) and *O* (red) distributions in 1C01. SS bonds shown as yellow clamps. Pink fragments—discordant. (b) 3D presentation of 1C01 with pink fragments distinguished according to the presentation in (a)

of these three domains (specifically, domain no. 3) (Table 4) is consistent with the fuzzy oil drop model. The FOD status and conformation of individual fragments are affected by a multitude of factors (ligand binding, enzymatic cavity) and differ greatly from the properties of identical domains appearing in other proteins. It turns out that the same geometric construct may represent various stages of conformance/discordance depending on the function of the given protein (as evidenced by immunoglobulin-like domains [22] and flavodoxin folds [31]).

Given the sample protein, it is interesting to look for fragments which follow the theoretical distribution despite the lack of such conformance for the protein as a whole. As it turns out, eliminating the following residues—293–246, 276–284, 303–306, 316–318,

Table 4

RD values for the three-domain structure, with each domains adopting a different supersecondary conformation

1SIQ	Fragment	Second. str.	RD
Complete molecule	3–392		0.597
Domain 1	3–130	Alpha orthog. Bundle	0.454
	11–16	Helix	0.477
	17–35	Helix	0.418
	36–46	Helix	0.499
	50–59	Helix	0.222
	66–68	Beta	0.366
	71–73	Beta	0.462
	75–88	Helix	0.480
	90–103	Helix	0.692
	104–112	Helix	0.740
	113–126	Helix	0.330
	Helix bundle	Helices together	0.506
	Beta folds together		0.359
Domain 2	131–236	Beta barrel	0.449
	132–134	Beta	0.551
	143–147	Helix—lig. Bind	0.455
	149–154	Beta	0.578
	158–171	Beta	0.394
	172–175	Helix—lig. Bind	0.237
	176–185	Beta	0.401
	188–196	Beta	0.557
	200–202	Beta	0.528
	All beta fr.	Beta barrel	0.451
		Ligand bind	0.500
		No ligand bind	0.442
Domain 3	238–392	Alpha up-down bundle	0.684
	239–275	Helix—1 cat. Res.	0.630
	276–277	Beta	0.421
	280–282	Beta	0.956
	285–315	Helix	0.481
	319–345	Helix	0.596
	346–351	Helix	0.232
	352–355	Helix	0.319
	356–367	Helix	0.520
	372–386	Helix—proximate to 1 cat. Res.	0.814
		Beta folds together	0.636
		Helix bundle	0.675
		Ligand binding	0.817
		No ligand binding	0.682

327–340, 348–351, 368–371, and 378–392—reduces the RD value from domain 3 to 0.493, indicating that the remainder of the domain is consistent with the model, i.e., adopts a shape which is entropically favorable given its interaction with the solvent

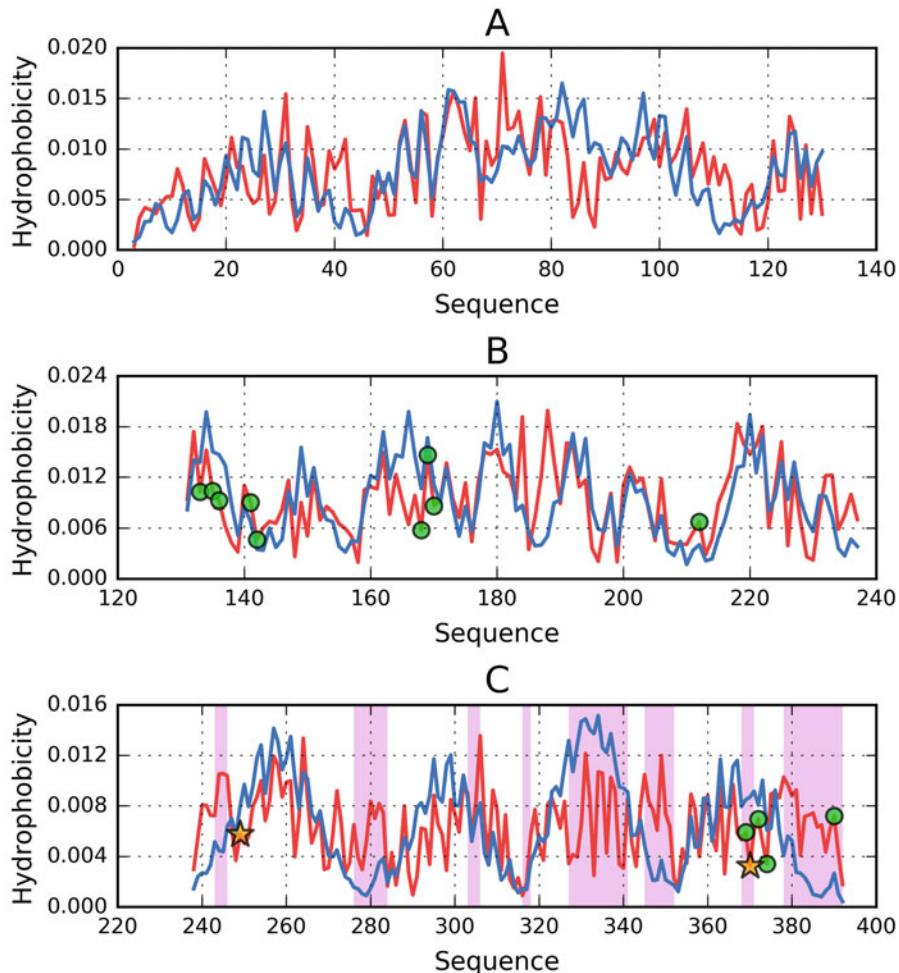


Fig. 8 Profiles of T (blue) and O (red) hydrophobicity distribution in domains. (a–c) Individual domains as described in text. (c) Pink fragments—fragments discordant in respect to T distribution

(Figs. 8 and 9) and therefore stabilizes the protein's tertiary conformation. Notably, the eliminated fragments include the protein's catalytic and ligand-binding residues.

4.4 Proteins Which Contain Chameleon Sequences

In the context of the fuzzy oil drop model, the question concerning the role of each secondary fold assumes a peculiar importance. It turns out that certain fragments may participate in the generation of a common hydrophobic core regardless of their secondary conformation. Attempts at protein structure prediction based on the assumption that specific sequences tend to adopt the same conformations are complicated by the presence of the so-called chameleon sequences: short peptides (6–11 residues) which may appear as beta folds or as helices depending on their host protein. In order to explain this phenomenon, we have assembled a set of proteins

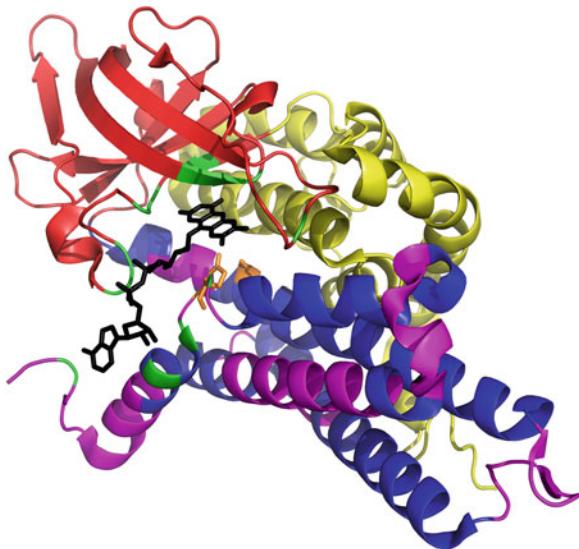


Fig. 9 3D presentation of 1SIQ ligand: black, residues interacting with ligand; green, residues eliminated to reach the RD value <0.5 are given in pink (according to Fig. 8)

containing 7-residue sequences which adopt helical conformations in some proteins and beta conformations in others. These proteins have been selected at random from ChSeq [32]—and while the set presented here is fairly small, a much larger database of proteins containing chameleon sequences (over 200 in total) with lengths between 6 and 11 aa is discussed in [33], showing that secondary conformation is merely a means to an end, rather than an end unto itself. Regardless of their secondary structural properties, the analyzed sequences turn out to share major similarities with respect to their involvement in hydrophobic core generation (within their host domain or the entire chain as appropriate).

Results listed in Table 5 and visualized in Fig. 10 reveal that the FOD status of a given fragment is not directly dependent on its secondary conformation.

The correlation coefficient for the spread of RD values in protein pairs (taking into account the status of the entire chain; Fig. 10a) is 0.139. When considering individual domains (where applicable), the coefficient increases to 0.227 (Fig. 10b). The corresponding coefficients for chameleon sequences are 0.676 (Fig. 10c) and 0.569 (Fig. 10d), respectively. The former two coefficients, while low, should be regarded as interesting, given the relatively small sample size. They indicate that the structures being compared differ with regard to the status of their hydrophobic cores. Despite this structural diversity, chameleon sequences exhibit strong correlation (upward of 0.6)—we can therefore assume that they play a similar role in shaping the hydrophobic

Table 5

Values of RD parameters for chains of length L and for structural units (SU—if the given chain includes a domain, its status is presented in a separate row)

PDB ID	STR. UNIT		RD		Sequence	RD		STR. UNIT		POS	L	PDB-ID
	L	POS	SU	FR		FR	SU	POS	L			
2GDF-A [34]	226	90	0.401	0.702	VGLSATT	0.442	0.761	417	492	4IKV-A [35]		
2DXQ-A [36]	147	59	0.507	0.847	VATATLL	0.943 0.908	0.645 0.498	141	185	3DCF-A [37]		
3CSL-A [38]	753	346	0.771 0.761	0.669 0.695	LEFYYDK	0.291	0.628	118	156	1EYV-A [39]		
2BLN-A [40]	298	138	0.740 0.487	0.614 0.537	VAQLRIA	0.406	0.484	110	260	1XWY-A [41]		
2QY2-A [42]	224	133	0.590	0.605	IKSTEEI	0.566 0.611	0.644 0.670	242	258	3RKG-A [43]		
4IMA-A [44]	522	513	0.7030 0.503	0.502 0.570	NIMRVLS	0.829	0.668	84	548	4HPF-A [45]		
3C8L-A [46]	121	110	0.609	0.711	AIAAVTV	0.535	0.514	95	131	1J23-A [47]		
1VDH-A [48]	247	105	0.519 0.535	0.505 0.595	FYSVVEL	0.321 0.431	0.464 0.422	100	231	4FJV-A [49]		
1MTP-A [50]	320	312	0.710 0.453	0.955 0.116	ATAAMML	0.611	0.774	134	430	4F35-A [51]		
1QFG-A [52]	707	565	0.760 0.712	0.849 0.837	EAKAALS	0.377	0.591	45	403	1Z8P [53]		
4E27-A [54]	164	85	0.795 0.627	0.590 0.491	YKVVGNL	0.311	0.393	27	90	2OTR [21]		

POS starting position of the chameleon sequence (according to ChSeq notation), FR RD value for the fragment whose sequence is listed in the central column

core in each chain (or domain). This leads us to the conclusion that a given residue sequence, rather than being a determinant of secondary conformation, instead plays a consistent role in ensuring that the resulting structure either contains or lacks a prominent hydrophobic core.

Our conjecture is borne out by analysis of a much larger set of sample proteins (over 200 different chains) [33].

4.5 Globally Discordant Structures

In this section we will discuss proteins/domains which diverge entirely from the monocentric distribution of hydrophobicity and instead adopt different conformations. The term “globally discordant” concerns the new form of ordering different than unicentric

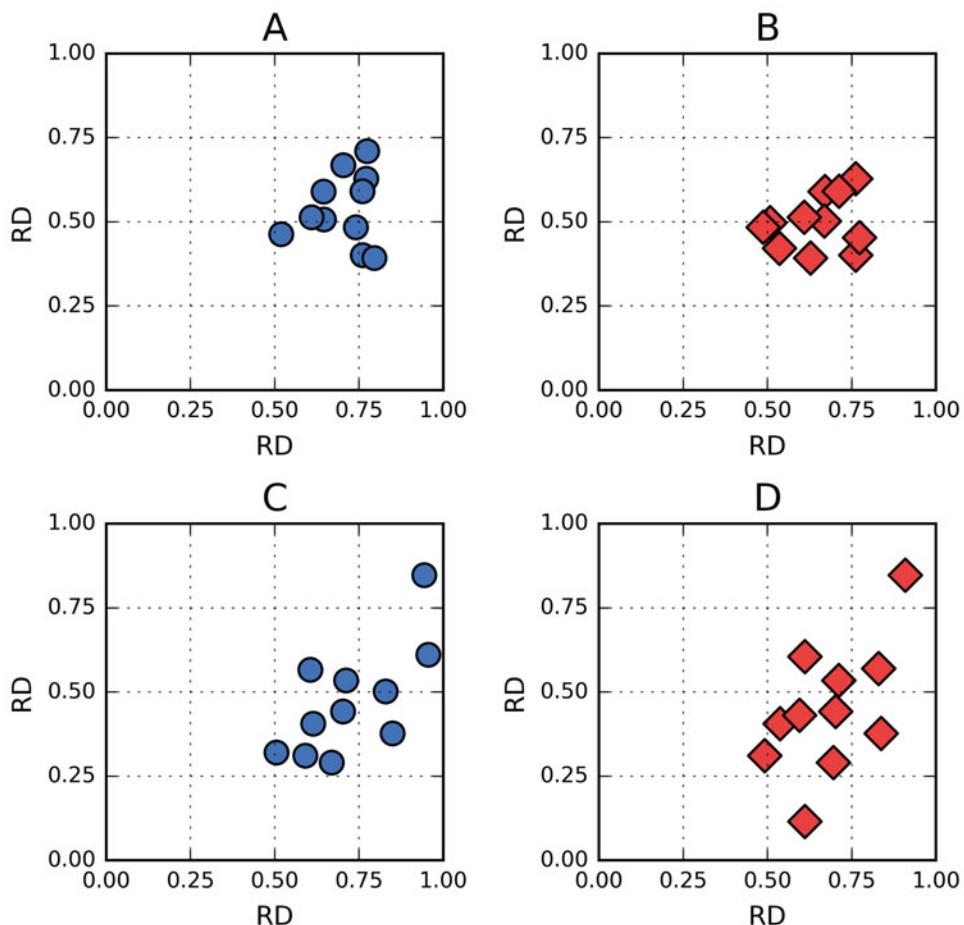


Fig. 10 Characteristics of chameleon fragments. **(a)** Relation of the RD status of structural units—chains—correlation coefficient, 0.139. **(b)** Relation of the RD status of structural units—chains and domains (if present)—correlation coefficient, 0.227. **(c)** Relation of RD status of chameleon fragments in units as shown in **(a)** (in chains)—correlation coefficient, 0.676. **(d)** Relation of RD status of chameleon fragments in units as shown in **(b)** (in chains or domains if present)—correlation coefficient, 0.569

one. This new ordering form is the linear propagation formed by alternating bands of high and low hydrophobicity.

This group contains both biologically active and pathological proteins. The former category of proteins includes supersecondary structures which exhibit linear arrangement, along with other fragments (“caps”) which, in some cases, make the protein as a whole consistent with the model.

The second group—the pathological proteins (amyloids)—is linear structures capable of unrestricted propagation. They lack structural “caps” and are therefore unable to produce a monocentric hydrophobic core while exposing hydrophilic residues on their surface.

4.5.1 Biologically Active Proteins Containing Solenoid Fragments

Solenoid fragments provide an example of a supersecondary unit which strongly deviates from the monocentric distribution of hydrophobicity. They typically exhibit linear arrangement of alternating bands of high and low hydrophobicity propagating along their main axis, entirely divergent from the theoretical Gaussian form. Rather than a local disruption, this phenomenon should instead be regarded as an entirely different structural pattern. An example of a protein which contains a solenoid fragment is a lyase-gamma-carbonic anhydrases from *Methanosarcina thermophila* [55].

The fragment in question is a beta-s-solenoid also referred to as a beta-helix. The protein as a whole is characterized by $RD = 0.476$ (Table 6), which suggests that it is soluble (contains a hydrophilic surface layer) and that it also includes elements which counteract the effects of linear propagation of hydrophobicity. By subjecting it to FOD analysis, we can identify factors which cause the protein (as a whole) to conform to the theoretical model ($RD < 0.5$).

The presented values of RD ($T-O-R$) clearly show that the solenoid—along with the beta-sheets which comprise it—diverges from the monocentric distribution of hydrophobicity. It is also interesting to note the status of the helix which runs alongside the solenoid. This helix is accordant with the model, suggesting that it may mediate contact with the aqueous solvent and also act as a “cap,” preventing unrestricted propagation. Similar structural components may be found in other solenoid-containing proteins [56] and provide leads in the search for novel pharmaceutical agents counteracting amyloid aggregation [57].

Table 6
RD values for $T-O-R$ and $T-O-H$ relations, along with HvT, TvO, and HvO correlation coefficients

1QRM	RD			Correlation coefficient		
	$T-O-R$	$T-O-H$	Fragment	HvT	TvO	HvO
Complete chain	0.476	0.471	5–213	0.363	0.661	0.737
Solenoid	0.601	0.448	Beta-sheets	0.207	0.389	0.716
Beta-sheet I	0.709	0.477	10–12 ^a	0.026	0.200	0.675
Beta-sheet II	0.572	0.425	33–35 ^a	0.162	0.463	0.751
STOP signal	0.436	0.471	175–181	0.317	0.549	0.868
Helix	0.426	0.346	193–211	0.278	0.597	0.764
Loops	0.570	0.446	13–25	-0.121	0.300	0.684
	0.489	0.432	83–106	0.482	0.582	0.838

^aThe beta-sheet is identified by listing its first constituent beta fold

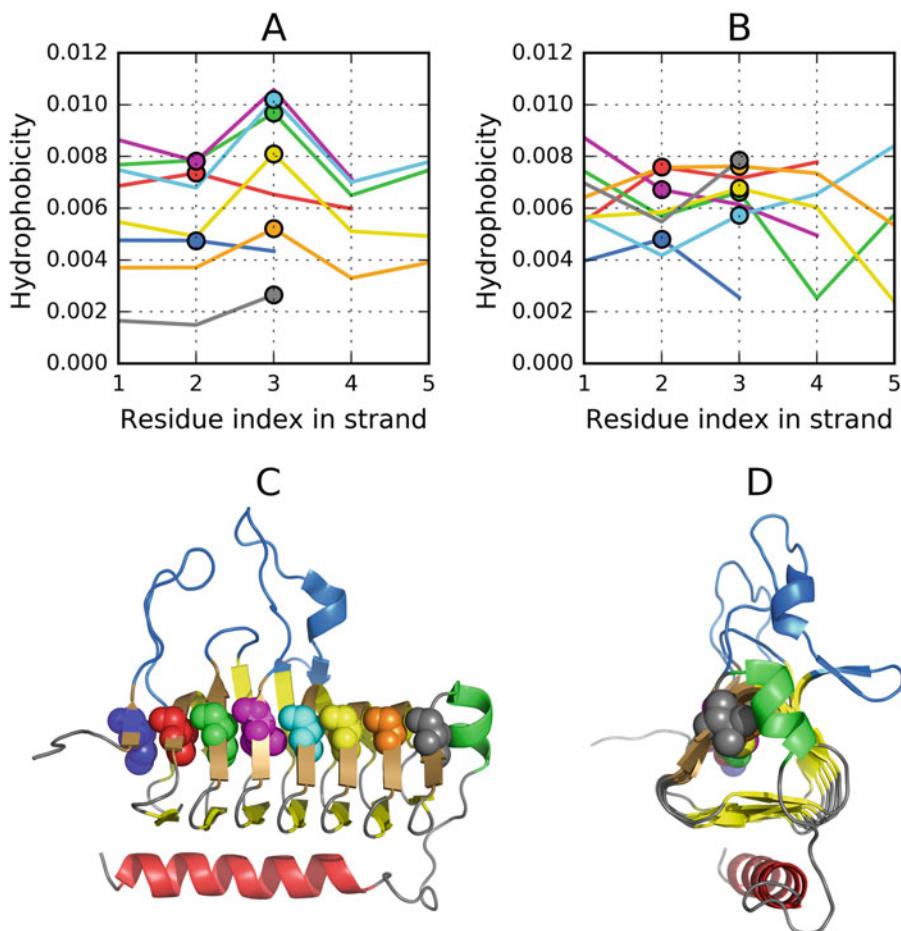


Fig. 11 Status of 1QRM. (a) T profiles for beta-structural fragments participating in beta-sheet generation—local maximum expected in position 3. (b) O profiles for beta-fragments as present in (a). Similar level on position 3 represents linear propagation of local maximum accompanied with linearly propagated band of similar hydrophobicity level. (c) 3D presentation of 1QRM—helix parallel to solenoid, red (its status, accordant with model); “capping helix,” green (status accordant with the model); loops, blue; beta-sheet (part of solenoid), colors according to lines shown in (a) and (b). (d) 3D presentation visualizing the linear propagation of residues discordant versus model. 3D structure of 1QRM with highlighted fragments: amphipathic helix parallel to the solenoid, red; “capping” helix, green; outlying loops, blue

Figure 11 provides a visual representation of all fragments discussed in this section—the helix, the cap, the solenoid fragment, and the outlying loops.

The listed correlation coefficients suggest close correspondence between the observed and the intrinsic distribution of hydrophobicity (O and H , respectively). These values may be compared to the corresponding values in an amyloid fibril—notably, the alignment between both distributions is less pronounced in the biologically active protein.

Linear propagation becomes evident when we analyze the distribution of hydrophobicity in the beta-sheet which forms part of the solenoid (Fig. 11). It can also be observed in the 3D representation (Fig. 11).

4.5.2 Pathological Proteins

For the purposes of this section, we have selected the beta amyloid ($\text{A}\beta 11\text{--}42$) (2MXU [58]). This pathological protein is characterized by strong propagation of alternating bands of high and low hydrophobicity propagating along the axis of the fibril and—in some areas—presenting a distribution of hydrophobicity which is a polar opposite of the expected values.

The 2MXU complex, as presented in PDB, can be characterized as follows: RD values computed for the entire complex are 0.680 ($T\text{-}O\text{-}R$) and 0.756 ($T\text{-}O\text{-}H$), respectively. Both values indicate significant differences between T and O , with the observed distribution more closely approximating R and—particularly— H . The conclusion is that the conformational properties of the fibril are dominated by the intrinsic hydrophobicity of each residue rather than by any tendency to produce a shared hydrophobic core. This suggestion is supported by analysis of correlation coefficients: 0.246, 0.363, and 0.821 for HvT, TvO, and HvO, respectively. The dominant role of intrinsic hydrophobicity in the amyloid fibril is therefore evident. In some areas the observed distribution can be described as the “opposite” of T (with negative correlation coefficients)—this applies, e.g., to residues no. 22–25 (given the reading frame size of 5 aa, it implies that the observation holds for the entire fragment at 22–29; see Fig. 12).

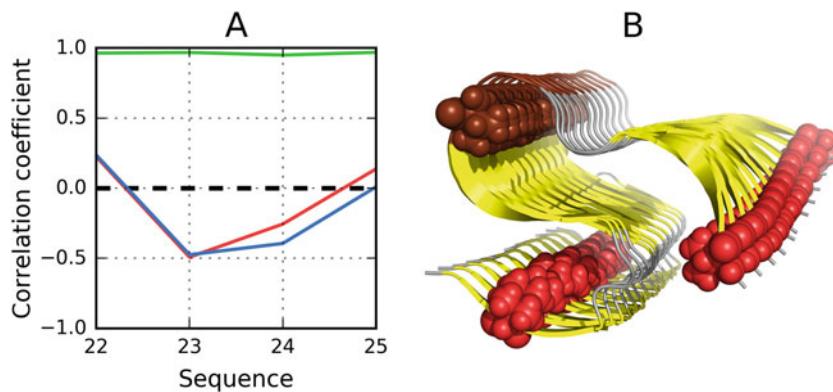


Fig. 12 Status of amyloid as observed in 2MXU. The 22–25 fragment observed as representing status opposite versus the Gaussian distribution in favor of a linear model. **(a)** Correlation coefficients for 5 aa chain fragments (first positions in pentapeptide are given on the bottom). Relation HvO, green line; TvO, red line; TvH, blue line. This identifies positions in observed distribution opposed the Gaussian distribution in favor of a linear model. **(b)** Linear propagation of positions representing similar status highly discordant versus the expected one (called as actively discordant). The brown residues—position of the fragment status of which is shown in **(a)**

Comparing the linear distribution of hydrophobicity in a biologically active protein with the one observed in the amyloid fibril reveals significant similarities. However, active proteins include mechanisms which keep this unusual distribution under control—in the presented case, this is the role of the amphipathic helix which runs parallel to the solenoid and mediates contact with the aqueous solvent (notably, the “capping” helix is also amphipathic). Amyloid proteins lack such safety measures and are therefore prone to unrestricted linear propagation, resulting in the formation of fibrillary deposits [10, 11].

5 Summary

In our to-date presentation of the fuzzy oil drop model, we have focused on the analysis of 3D structures derived from databases (PDB). The model may, however, be applied to the folding process itself. Folding simulations conducted in an environment whose properties are modeled by the 3D Gaussian acknowledge the active participation of the environment in determining the protein’s final structure, where hydrophobic residues are directed toward the central part of the emerging globule. While other models also account for the presence of water, they typically do so by treating the solvent as a collection of individual molecules interacting with constituent residues of the input chain. Such interaction is highly localized, and its effects are also local in scope. In contrast, the fuzzy oil drop model treats the solvent as a global force field acting upon the protein chain as a whole (as expressed by the 3D Gaussian). Changes in the properties of this environment—e.g., with regard to its ionic potential, pH, presence of other nonpolar molecules such as presence of membranes, etc.—may result in amyloid transformation [59, 60]. Shaking—a somewhat “nonscientific” factor—is a known promoter of amyloidogenesis, and it may not be coincidental that shaking greatly increases the surface area of the water/air phase transition space. While the physical structure of water in its natural state is not well described, we cannot speculate about the structural changes caused by the abovementioned factors. In particular, the effects of ice crystal formation and substances which lower the freezing point of water (antifreeze factors) await a proper scientific study which would help explain the mechanisms of various phenomena known to occur in an aqueous environment.

Recent research into the effects of hydrophobic and hyperhydrophobic surfaces, capable of inducing levitation of water particles, may help explain the properties of the aqueous environment as such [61–63].

The fuzzy oil drop model is based on a hypothesis concerning the effect of the solvent upon the structural properties of polypeptide chains. Such chains, unlike surfactant micelles, are characterized by varying hydrophobicity and limited conformational freedom, which may explain the occasional local exposure of excess hydrophobicity on the protein surface, or areas where hydrophobicity is lower than expected.

The structural properties of water become evident in the course of micellization, where polar—but highly similar (or even identical)—molecules self-assemble into symmetrical shapes such as spheres or cylinders. From the point of view of the fuzzy oil drop model, the protein is also a globular micelle, whose regularity (or lack thereof) can be explained by the variable hydrophobicity of its constituent residues, as well as by the presence of other factors, such as peptide bonds, which do not apply to surfactant micelles.

It seems that the effects of the internal force field (a term used to refer to nonbinding interactions between atoms which comprise the protein) are comparable to that of the continuous external field generated by the aqueous solvent. In this sense, attempting to model the environment as a collection of individual molecules appears insufficient. Simulating the folding process in conditions described by the 3D Gaussian appears to solve this issue, although it requires proper multicriteria optimization techniques [64]. Such simulations would reveal the relative importance of both force fields (external and internal). It is expected that this proportion will vary, depending on the properties of the solvent and of the residue sequence itself—including the locations of its hydrophobic and hydrophilic residues. Reduced dependence on the external field—which is postulated by structural analysis of amyloid fibrils—seems to promote a linear distribution of hydrophobicity, dominated by the intrinsic properties of each residue. Referring once again to the concept of an “intelligent micelle,” we can state that the amyloid more closely resembles a surfactant micelle, with linear (ribbon-like or cylindrical) symmetry and no evidence of encoded information (i.e., no local deviations from the overarching structural pattern).

If we accept Anfinsen’s statement that “the native conformation (of a protein) is determined by the totality of interatomic interactions and hence by the amino acid sequence, in a given environment” [65], it logically follows that the conformation should be obtainable *in silico* by optimizing interatomic interactions. However, the latter part of the statement—referring to the environment—is often neglected or underestimated. Most models treat water as a collection of individual molecules, with all the attendant properties, such as density (number of molecules per unit of volume), ionic potential (concentrations of ions such as Na and Cl), and pH. The solvent is not, however, thought to produce a global external force field. We furthermore lack a clear

picture of the structural properties of water (except ice structure). Such properties may be elucidated by analyzing the effects of the solvent of other molecules. Micellization (at least in the context of surfactants) is a well-understood phenomenon and serves as a clear proof that the solvent should be treated as a continuum, directing nonpolar portions of the surfactant toward the center of the micelle while exposing polar fragments. The micelle is a very simple system: in most cases it consists of identical molecules with known physicochemical characteristics. Consequently, micelles exhibit a high degree of symmetry (as spheres or cylinders). In contrast, the polypeptide is far more complex, given the varied hydrophobicity of its unit residues. Further restrictions are associated with the volume of each residue and with the presence of covalent bonds, limiting conformational freedom (unlike in surfactant micelles, held together by nonbinding interactions). Nevertheless, the protein molecule can be characterized as a quasi-micelle owing to the way it interacts with the aqueous environment. Other reports on this subject can be found [66–68].

The presented fuzzy oil drop model treats the polypeptide as a peculiar micelle, whose structure depends on the proportions and relations between its hydrophobic and hydrophilic residues. Introducing a force field described by a 3D Gaussian immediately determines the structure and even the size of a surfactant micelle. In a similar manner, applying the Gaussian field to a polypeptide chain should, at least in part, explain the dynamics of the folding process, leading to the emergence of a hydrophobic core. On the other hand, the specific properties of certain unit sequences often prevent the protein from becoming a uniform micelle. The protein may thus be described as an “intelligent micelle,” carrying information which is encoded as structural deformations. This can be referred to as “specificity”—local deviations from the theoretical distribution of hydrophobicity frequently correspond to active sites, ligand-binding pockets, or complexation interfaces. The folding process must account for such deformations—in this context, secondary and supersecondary folds can be treated as means to an end, i.e., they represent a mechanism by which the protein may deviate from a perfectly symmetrical micelle in a predetermined and tightly controlled manner.

Acknowledgments

The authors are indebted to Piotr Nowakowski and Anna Śmietańska for their editorial and technical help. This research was supported by Jagiellonian University Medical College grant no. K/ZDS/006363.

References

1. Sillitoe I, Lewis TE, Cuff AL, Das S, Ashford P, Dawson NL, Furnham N, Laskowski RA, Lee D, Lees J, Lehtinen S, Studer R, Thornton JM, Orengo CA (2015) CATH: comprehensive structural and functional annotations for genome sequences. *Nucleic Acids Res* 43:D376–D381. <https://doi.org/10.1093/nar/gku947>
2. <http://www.cathdb.info/>
3. Fox NK, Brenner SE, Chandonia JM (2014) SCOPe: Structural Classification of Proteins—extended, integrating SCOP and ASTRAL data and classification of new structures. *Nucleic Acids Res* 42:D304–D309
4. <http://scop.berkeley.edu/>
5. Finn RD, Coggill P, Eberhardt RY, Eddy SR, Mistry J, Mitchell AL, Potter SC, Punta M, Qureshi M, Sangrador-Vegas A, Salazar GA, Tate J, Bateman A (2016) The Pfam protein families database: towards a more sustainable future. *Nucleic Acids Res* 44(Database Issue): D279–D285
6. <https://pfam.xfam.org/>
7. Berman HM, Westbrook J, Feng Z, Gilliland G, Bhat TN, Weissig H, Shindyalov IN, Bourne PE (2000) The protein data bank. *Nucleic Acids Res* 28:235–242
8. www.rcsb.org
9. Sternberg MJE (1996) Protein structure prediction—principles and approaches. In: Sternberg MJE (ed) *Protein structure prediction: a practical approach*. IRL Press, Oxford
10. Roterman I, Banach M, Kalinowska B, Konieczny L (2016) Influence of the aqueous environment on protein structure—a plausible hypothesis concerning the mechanism of amyloidogenesis. *Entropy* 18(10):351
11. Roterman I, Banach M, Konieczny L (2017) Application of the fuzzy oil drop model describes amyloid as a ribbonlike micelle. *Entropy* 19(4):167
12. Kalinowska B, Banach M, Konieczny L, Roterman I (2015) Application of divergence entropy to characterize the structure of the hydrophobic core in DNA interacting proteins. *Entropy* 17(3):1477–1507
13. Kyte J, Doolittle RF (1982) A simple method for displaying the hydropathic character of a protein. *J Mol Biol* 157:105–132
14. Konieczny L, Brylinski M, Roterman I (2006) Gauss function based model of hydrophobicity density in proteins. In *Silico Biol* 6:15–22
15. Kauzmann W (1959) Some factors in the interpretation of protein denaturation. *Adv Protein Chem* 14:1–63
16. Levitt M (1976) A simplified representation of protein conformations for rapid simulation of protein folding. *J Mol Biol* 104:59–107
17. Kullback S, Leibler RA (1951) On information and sufficiency. *Ann Math Stat* 22:79–86
18. Dygut J, Kalinowska B, Banach M, Piwowar M, Konieczny L, Roterman I (2016) Structural interface forms and their involvement in stabilization of multidomain proteins or protein complexes. *Int J Mol Sci* 17(10):E1741
19. Kalinowska B, Banach M, Wiśniowski Z, Konieczny L, Roterman I (2017) Is the hydrophobic core a universal structural element in proteins? *J Mol Model* 23(7):205
20. Devlin TM (2011) *Textbook of biochemistry with clinical correlations*, vol 7. Wiley, New York
21. Han KD, Park SJ, Jang SB, Lee BJ (2008) Solution structure of conserved hypothetical protein HP0892 from *Helicobacter pylori*. *Proteins* 70(2):599–602
22. Banach M, Konieczny L, Roterman I (2014) The fuzzy oil drop model, based on hydrophobicity density distribution, generalizes the influence of water environment on protein structure and function. *J Theor Biol* 359:6–17
23. Fokkens J, Klebe G (2006) A simple protocol to estimate differences in protein binding affinity for enantiomers without prior resolution of racemates. *Angew Chem Int Ed Engl* 45(6):985–989
24. Jones EY, Davis SJ, Williams AF, Harlos K, Stuart DI (1992) Crystal structure at 2.8 Å resolution of a soluble form of the cell adhesion molecule CD2. *Nature* 360(6401):232–239
25. Kister A (2015) Amino acid distribution rules predict protein fold: protein grammar for beta-strand sandwich-like structures. *Biomolecules* 5:41–59
26. Fokas AS, Papatheodorou TS, Kister AE, Gelfand IM (2005) A geometric construction determines all permissible strand arrangements of sandwich proteins. *PNAS* 102(44):15851–15853
27. Fokas AS, Gelfand IM, Kister AE (2004) Prediction of the structural motifs of sandwich proteins. *PNAS* 101(48):16780–16783
28. McManus AM, Nielsen KJ, Marcus JP, Harrison SJ, Green JL, Manners JM, Craik DJ

- (1999) MiAMP1, a novel protein from *Macadamia integrifolia* adopts a Greek key beta-barrel fold unique amongst plant antimicrobial proteins. *J Mol Biol* 293(3):629–638
29. Banach M, Kalinowska B, Konieczny L, Roterman I (2016) Role of disulfide bonds in stabilizing the conformation of selected enzymes—an approach based on divergence entropy applied to the structure of hydrophobic core in proteins. *Entropy* 18(3):67
30. Fu Z, Wang M, Paschke R, Rao KS, Frerman FE, Kim JJ (2004) The crystal structure and mechanism of human glutaryl-CoA dehydrogenase. *Biochemistry* 43:9674–9684
31. Banach M, Prudhomme N, Carpentier M, Duprat E, Papandreou N, Kalinowska B, Chomilier J, Roterman I (2015) Contribution to the prediction of the fold code: application to immunoglobulin and flavodoxin cases. *PLoS One* 10(4):e0125098
32. Li W, Kinch LN, Karplus PA, Grishin NV (2015) ChSeq: a database of chameleon sequences. *Protein Sci* 24(7):1075–1086. <https://doi.org/10.1002/pro.2689>
33. Kalinowska B, Banach M, Konieczny L, Roterman I (2016) Chameleon sequences—sequence-to-structure relation in proteins. *J Proteomics Bioinform* 9:264–275
34. Gallego F, Sol D, Chornet JJC, Cavada BS. PDB
35. Doki S, Kato HE, Solcan N, Iwaki M, Koyama M, Hattori M, Iwase N, Tsukazaki T, Sugita Y, Kandori H, Newstead S, Ishitani R, Nureki O (2013) Structural basis for dynamic mechanism of proton-coupled symport by the peptide transporter POT. *Proc Natl Acad Sci U S A* 110(28):11343–11348
36. Binkowski TA, Xu X, Edwards A, Savchenko A, Joachimiak A. Midwest Center for Structural Genomics (MCSG)—PDB
37. Joint Center for Structural Genomics (JCSG)—PDB
38. Krieg S, Huché F, Diederichs K, Izadi-Pruneyre N, Lecroisey A, Wandersman C, Delapelaire P, Welte W (2009) Heme uptake across the outer membrane as revealed by crystal structures of the receptor-hemophore complex. *Proc Natl Acad Sci U S A* 106(4):1045–1050
39. Gopal B, Haire LF, Cox RA, Jo Colston M, Major S, Brannigan JA, Smerdon SJ, Dodson G (2000) The crystal structure of NusB from *Mycobacterium tuberculosis*. *Nat Struct Biol* 7(6):475–478
40. Williams GJ, Breazeale SD, Raetz CR, Naismith JH (2005) Structure and function of both domains of ArnA, a dual function decarboxylase and a formyltransferase, involved in 4-amino-4-deoxy-L-arabinose biosynthesis. *J Biol Chem* 280(24):23000–23008
41. Malashkevich VN, Xiang DF, Rauschel FM, Almo SC, Burley SK. New York Sgx Research Center For Structural Genomics (Nsgxrc)—PDB
42. Benarroch D, Smith P, Shuman S (2008) Characterization of a trifunctional mimivirus mRNA capping enzyme and crystal structure of the RNA triphosphatase domain. *Structure* 16(4):501–512
43. Khan MB, Sponder G, Sjöblom B, Svidová S, Schweyen RJ, Carugo O, Djinović-Carugo K (2013) Structural and functional characterization of the N-terminal domain of the yeast Mg²⁺ channel Mrs2. *Acta Crystallogr D Biol Crystallogr* 69(Pt 9):1653–1664
44. Holyoak T, Zhang B, Deng J, Tang Q, Prasanan CB, Fenton AW (2013) Energetic coupling between an oxidizable cysteine and the phosphorylatable N-terminus of human liver pyruvate kinase. *Biochemistry* 52(3):466–476
45. Leonetti MD, Yuan P, Hsiung Y, Mackinnon R (2012) Functional and structural analysis of the human SLO3 pH- and voltage-gated K⁺ channel. *Proc Natl Acad Sci U S A* 109(47):19274–19279
46. Joint Center for Structural Genomics (JCSG). Crystal structure of ftsz-like protein of unknown function (zp_00109722.1) from *nostoc punctiforme* pcc 73102 at 1.22 Å resolution—PDB
47. Nishino T, Komori K, Ishino Y, Morikawa K (2003) X-ray and biochemical anatomy of an archaeal XPf/Rad1/Mus81 family nuclease: similarity between its endonuclease domain and restriction enzymes. *Structure* 11(4):445–457
48. Ebihara A, Okamoto A, Kousumi Y, Yamamoto H, Masui R, Ueyama N, Yokoyama S, Kuramitsu S (2005) Structure-based functional identification of a novel heme-binding protein from *Thermus thermophilus* HB8. *J Struct Funct Genom* 6(1):21–32
49. Altun M, Walter TS, Kramer HB, Herr P, Iphöfer A, Boström J, David Y, Komsany A, Ternette N, Navon A, Stuart DI, Ren J, Kessler BM (2015) The human otubain2-ubiquitin structure provides insights into the cleavage specificity of poly-ubiquitin-linkages. *PLoS One* 10(1):e0115344
50. Irving JA, Cabrita LD, Rossjohn J, Pike RN, Bottomley SP, Whisstock JC (2003) The 1.5 Å crystal structure of a prokaryote serpin: controlling conformational change in a heated environment. *Structure* 11(4):387–397

51. Mancusso R, Gregorio GG, Liu Q, Wang DN (2012) Structure and mechanism of a bacterial sodium-dependent dicarboxylate transporter. *Nature* 491(7425):622–626
52. Ferguson AD, Welte W, Hofmann E, Lindner B, Holst O, Coulton JW, Diederichs K (2000) A conserved structural motif for lipo-polysaccharide recognition by prokaryotic and eucaryotic proteins. *Structure* 8(6):585–592
53. Nagano S, Cupp-Vickery JR, Poulos TL (2005) Crystal structures of the ferrous dioxygen complex of wild-type cytochrome P450eryF and its mutants, A245S and A245T: investigation of the proton transfer system in P450eryF. *J Biol Chem* 280(23):22102–22107
54. Craig TK, Abendroth J, Lorimer D, Burgin AB Jr, Segall A, Rohwer F. Crystal structure of a pentameric capsid protein isol from metagenomic phage sequences solved by iodide sad phasing—PDB
55. Iverson TM, Alber BE, Kisker C, Ferry JG, Rees DC (2000) A closer look at the active site of gamma-class carbonic anhydrases: high-resolution crystallographic studies of the carbonic anhydrase from Methanoscarcina thermophila. *Biochemistry* 39(31):9222–9231
56. Roterman I, Banach M, Konieczny L (2017) Propagation of fibrillar structural forms in proteins stopped by naturally occurring short polypeptide chain fragments. *Pharmaceuticals (Basel)* 10(4):89
57. Roterman I, Banach M, Konieczny L (2018) Towards the design of anti-amyloid short peptide helices. *Bioinformation* 14(1):1–7
58. Xiao Y, Ma B, McElheny D, Parthasarathy S, Long F, Hoshi M, Nussinov R, Ishii Y (2015) A beta (1–42) fibril structure illuminates self-recognition and replication of amyloid in Alzheimer’s disease. *Nat Struct Mol Biol* 22:499–505
59. Chiti F, Dobson CM (2017) Protein misfolding, amyloid formation and human disease; a summary of progress over the last decade. *Annu Rev Biochem* 86:27–68
60. Gremer L, Schölzel D, Schenk C, Reinartz E, Labahn J, Ravelli RBG, Tusche M, Lopez-Iglesias C, Hoyer W, Heise H, Willbold D, Schröder GF (2017) Fibril structure of amyloid- β (1–42) by cryo-electron microscopy. *Science* 358(6359):116–119
61. Biedermann F, Nau WM, Schneider H-J (2014) The hydrophobic effect revisited—studies with supramolecular complexes imply high-energy water as a noncovalent driving force. *Angew Chem* 53:11158–11171
62. Schutzius TM, Jung S, Maitra T, Graeber G, Köhme M, Poulikakos D (2015) Spontaneous droplet trampolining on rigid superhydrophobic surfaces. *Nature* 527(7576):82–85
63. Kim KH, Späh A, Pathak H, Perakis F, Mariedahl D, Amann-Winkel K, Sellberg JA, Lee JH, Kim S, Park J, Nam KH, Katayama T, Nilsson A (2017) Maxima in the thermodynamic response and correlation functions of deeply supercooled water. *Science* 358(6370):1589–1593
64. Konieczny L, Roterman I (2012) Conclusions. In: Roterman-Konieczna I (ed) *Protein folding in silico*. Elsevier, Oxford, pp 191–203
65. Anfinsen CB (1973) Principles that govern the folding of protein chains. *Science* 181:223–230
66. Kim W, Xiao J, Chaikof EL (2011) Recombinant amphiphilic protein micelles for drug delivery. *Langmuir* 27(23):14329–14334
67. Kim W, Brady C, Chaikof EL (2012) Amphiphilic protein micelles for targeted in vivo imaging. *Acta Biomater* 8(7):2476–2482
68. Schott H (1968) On the similarity between micelles of nonionic detergents and globular proteins. *J Am Oil Chem Soc* 45(11):823–824



Chapter 20

Hands On: Using Tryptophan Fluorescence Spectroscopy to Study Protein Structure

Nadja Hellmann and Dirk Schneider

Abstract

Fluorescence spectroscopy is well suited to obtain information about the structure and function of proteins. The major advantage of this spectroscopic technique is the pronounced dependence of the fluorescence emission characteristics of fluorophores on their distinct local environment and the rather inexpensive equipment required. In particular, the use of intrinsic tryptophan fluorescence offers the possibility to study structure and function of proteins without the need to modify the protein. While fluorescence spectroscopy is technically not demanding, a number of factors can artificially alter the results. In this article, we systematically describe the most common applications in fluorescence spectroscopy of proteins, i.e., how to gain information about the local environment of tryptophan residues and how to employ changes in the environment to monitor an interaction with other substances. In particular, we discuss pitfalls and wrong and/or misleading interpretations of gained data together with potential solutions.

Key words Tryptophan, Intrinsic fluorescence, Inner filter effect, Quenching, Energy transfer, Protein fluorescence

1 Introduction

Fluorescence spectroscopy is frequently used to obtain information about the structure and function of proteins. The major advantage of this spectroscopic technique is the pronounced dependence of the fluorescence emission characteristics of fluorophores on their distinct local environment and the rather inexpensive equipment required. Yet, in many cases, proteins have to be modified prior to analysis. However, if, e.g., fluorescent dyes are covalently attached to proteins, such modifications can affect the structure and activity of proteins.

Tryptophan residues, which are naturally present in most proteins, are intrinsically fluorescent, and the particular advantage of monitoring tryptophan fluorescence is the possibility to gain information about a protein without any preceding protein modification. Although the aromatic amino acids tyrosine and

phenylalanine are also fluorescent, the quantum yield of free phenylalanine is very low (0.04, [1]) compared to free tryptophan (0.2, [1]). While the quantum yield of free tyrosine in water is similar to that of tryptophan (0.14 [1]), tryptophan's extinction coefficient is substantially higher ($5600\text{ M}^{-1}\text{ cm}^{-1}$ compared to $1400\text{ M}^{-1}\text{ cm}^{-1}$ for the free amino acid [1]). Thus, if present in a protein, tryptophan will typically dominate a fluorescence emission spectrum, while the fluorescence emission of tyrosine residues is masked.

The characteristics of a tryptophan fluorescence emission spectrum markedly depend on the tryptophan's environment. When a tryptophan is completely exposed to the polar aqueous environment, the emission maximum is around 350 nm, as found for the free amino acid in water, whereas the emission maximum can be as low as 310 nm in a very hydrophobic environment [2] (a well-known example is azurin [3]). Thus, the position of the fluorescence emission maximum reflects the polarity of a tryptophan's adjacent environment (Fig. 1). Statistical analyses of a large number of protein fluorescence emission spectra have indicated that five major classes of tryptophan residues, which differ in their local environment, can be distinguished [4, 5]. These correlate well with distinct tryptophan classes differing in their accessibilities for externally added soluble quenchers [6]. Furthermore, tryptophan residues were also classified based on their intimate environment in proteins possessing known crystal structures [6]. Together, such analyses have led to an integrated approach, allowing to obtain information about a protein's structure from tryptophan fluorescence measurements [6]. Thus, the properties of a fluorescence

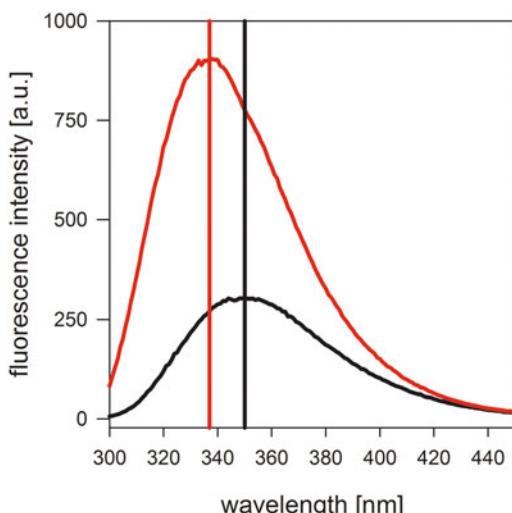


Fig. 1 Influence of solvent polarity on tryptophan's emission properties. Black, tryptophan in water; red, tryptophan in methanol. The vertical lines indicate the position of the maximum

emission spectrum can indirectly report on the position of a tryptophan residue within a protein, i.e., the proteins' super-secondary, tertiary, and/or quaternary structure, as well as on changes of a protein structure, e.g., due to ligand binding.

Apart from the position of the emission maximum, the local environment also modulates the intensity of the fluorescence emission. Typically, the quantum yield (i.e., the fraction of excited molecules emitting fluorescence light) is higher in an apolar environment, as, e.g., found in the core of folded proteins, than in polar environments [2]. While the fluorescence quantum yield clearly depends on the polarity of the tryptophan environment and the polarity of the environment, changes e.g., when proteins fold/unfold, the quantum yield is also influenced in folded proteins by fluorescence quenching processes. In principle, the tryptophan fluorescence might be quenched by proton transfer from charged amino groups, as well as by electron transfer from/to protonated carboxyl groups, disulfides, and/or amides [2]. Thus, residues such as cysteine, lysine, histidine, and/or the protein backbone potentially lower the fluorescence quantum yield of tryptophan [2, 7]. Furthermore, fluorescence energy transfer between multiple tryptophans or between tyrosine and tryptophan residues might occur [2]. Thus, there is no clear and simple correlation between the position of a fluorescence emission maximum and the quantum yield of tryptophan residues in proteins. Yet, changes in the intensity of a fluorescence emission clearly reflect changes in a tryptophan's intimate environment.

Together, monitoring tryptophan fluorescence is suitable to gain information about the local environment of tryptophan residues in folded proteins, as well as to follow changes of a tryptophan's environment, e.g., during folding/unfolding, structural rearrangements, or binding of substances, allowing to quantify such processes. In contrast, while the aromatic amino acids tyrosine and phenylalanine are also fluorescent (as mentioned above), the emission characteristics of these residues do not considerably depend on the local environment. Thus, solely tryptophan is suitable to gain information about the structure of and structural adjustments within proteins. However, the tryptophan fluorescence emission spectrum monitored for a particular protein is the sum of the individual fluorescence emission spectra of all tryptophans present in the protein, which might differ in their spectral characteristics. Thus, analyses using tryptophan fluorescence emission spectra are most meaningful when peptides or small proteins are investigated, as the number of tryptophan residues is limited here, which makes interpretation of the spectra and spectral changes more straightforward than in case of proteins containing multiple tryptophans.

Although the spectroscopic measurement itself is technically rather straightforward, a number of experimental and technical

issues need to be considered prior to and/or during the actual measurements, which are outlined in this chapter. A proper interpretation of fluorescence emission spectra is only possible if a spectrum is not artificially distorted, which could occur due to unexpected impurities, inadequate temperature control, bleaching, inner filter effect, or improper measurement settings.

Here, we describe the general procedure to obtain and to meaningfully interpret tryptophan fluorescence emission spectra, and we also describe what kind of information can be gained from such analyses.

2 Materials

2.1 The “First and Always...”

2.1.1 Reagents

2.1.2 Equipment

1. Buffer

The buffer used for the measurement is largely defined by the protein. From the technical point of view, no limitations apply, except the use of buffers that contain substances absorbing in the far-UV, which leads to inner filter effects (as discussed in Subheading 3.3.3). Always use analytical grade chemicals, in order to reduce the chance of having fluorescent contaminations (*see Note 1*). Buffers containing imidazole, as typically used for purification of his-tagged proteins, should not be used.

2. Milli-Q water (potentially).

1. The fluorescence spectrometer

Any well-calibrated (*see Note 2*) research fluorescence spectrometer can be used, and no special requirements apply, except that the optics allows measurements in the UV range. Yet, the intensity of the lamp might be important for the tryptophan fluorescence measurement. Standard fluorescence spectrometers are equipped with a 150 W lamp. Spectrometers with higher sensitivities are often equipped with a 450 W lamp. Thus, under otherwise identical settings, such instruments will give rise to increased danger of photo-bleaching (*see Note 3*). In this case, the excitation slit width needs to be reduced (*see Note 4*).

Typically, the cuvette holder is tempered, and temperature control should be used due to the intrinsic temperature dependence of fluorescence emission (*see Note 5*).

2. Cuvettes

The cuvettes used for the measurements need to transmit light in the UV region. Thus, either quartz or special, UV-transmissible plastic fluorescence cuvettes are required. With respect to thermal equilibration, plastic cuvettes are less favorable (*see Note 5*), since more time is required to thermally equilibrate solutions in plastic ware.

A good compromise between large volumes and easy washing/sample removal is to use half-micro cuvettes, which have a path length of 1 cm in one direction and 0.4 cm in the other. These have the additional advantage that the path length can be shortened by rotation of the cuvette in case inner filter effects become an issue (*see* Subheading 3.3.3). However, should the sample volume be an issue, low-volume cuvettes can also be used. Yet, cuvettes having very low volumes might be less favorable when solutions are mixed within the cuvette (*see* Note 6). Solutions should always be mixed with special care within the cuvette in order to avoid formation of bubbles, which would lead to light scattering.

3. What else?

Dust-free paper tissues should always be used to dry the outside of cuvettes. Do not use dry paper in case of quartz cuvettes; always use paper wetted with a bit of ethanol to clean the outside (reduces chance of scratches).

Use pure ethanol (without additive) and water for cleaning the cuvettes. Do not use compressed air from technical sources, as this often contains oil from the compressor. Use compressed air only together with a special filter, or use clean compressed air from a gas cylinder. Nitrogen gas can also be used for drying.

2.2 Analysis of Trp Fluorescence of the Protein (as It Is)

2.2.1 Reagents

1. Buffers: *see item 1* in Subheading 2.1.1.

2. The protein sample

Well-purified samples are required to monitor meaningful spectra. In particular, when a studied protein contains only a single tryptophan residue, already a little contamination with proteins having a high tryptophan content could severely distort the fluorescence emission spectrum. Furthermore, fractions of denatured and/or aggregated protein might lead to increased scattering. Thus, remove protein aggregates prior to the measurement, e.g., by filtration or ultracentrifugation.

A typical protein concentration used in fluorescence measurements is a few micromolar (*see* Subheading 3.2.1). Ideally, the concentration of a protein stock solution should be higher than the concentration in the measured solution, in order to be able to perform dilutions from stock. If the protein tends to stick to the wall of the container, prolonged storage, in particular at low protein concentrations, might lead to a reduced concentration of protein in solution (*see* also Subheading 3.3.2).

2.2.2 Equipment

1. *See* items in Subheading 2.1.2.

2. Potentially: syringe filter, pore size 0.2 µm, to remove scattering particles.

2.3 Unintended Changes in Fluorescence Characteristics Upon Addition of Substances

2.3.1 Reagents

2.3.2 Equipment

1. Buffers: *see item 1* in Subheading 2.1.1.
2. Tryptophan dissolved in buffer as test substance. The fluorescence emission intensity should be similar to the one observed in the experiment, for which the test is performed.

2.4 Solvent Accessibility of Tryptophan Residues Tested via Fluorescence Quenching

2.4.1 Reagents

1. *See items in Subheading 2.1.2.*
2. A standard research photometer is needed in order to determine the absorption of the added substances in case inner filter effects are suspected.

1. Buffers: *see item 1* in Subheading 2.1.1.
2. Protein solution: *see item 2* in Subheading 2.2.1.
3. Iodide solution

Potassium iodide (KI) solutions can be prepared up to a concentration of 1430 g/L in water (http://www.chemicalbook.com/ChemicalProductProperty_EN_CB3125298.htm, August 31, 2018) by dissolving the corresponding weighted amount into the buffer. Thus, the maximum concentration is approximately 9 M. When iodide is used as a fluorescence quencher, it is important to keep the ionic strength of the protein solution constant, for example, by addition of KCl to the quencher solution. Furthermore, the formation of I_2 has to be inhibited, for example, by addition of $Na_2S_2O_3$. Otherwise, reactive species might develop which can partition into the nonpolar regions of proteins and membranes [2]. Also, keeping the solution on ice helps to inhibit I_2 formation for some time. However, the time to reach thermal equilibrium after addition of the solution to a cuvette might be prolonged (*see Note 5*).

4. Acrylamide solution

Acrylamide is prepared by dissolving the weighted amount into buffer. Solubility in water is at least 2.5 g/10 mL (https://www.sigmapelab.com/content/dam/sigma-aldrich/docs/Sigma/Product_Information_Sheet/a3553pis.pdf, August 31, 2018), resulting in an approximately 3.5 M solution.

Note that liquid acrylamide is considered to be carcinogenic and neurotoxic.

2.4.2 Equipment

1. *See items in Subheading 2.1.2.*
2. If the quencher causes inner filter effects, *see item 2* in Subheading 2.3.2.

3 Methods

3.1 The “First and Always...”

Prior to a measurement, it is advisable to verify the correct settings of the instrument (based on the position of the Raman peak, *see Note 7*) and to check whether the buffers are not contaminated.

1. Fill a cuvette with tempered buffer (*see Note 5*), and measure a fluorescence emission spectrum ($\lambda_{\text{ex}} = 280 \text{ nm}$, $\lambda_{\text{em}} = 300\text{--}450 \text{ nm}$). *See Notes 4, 8, 9* for other settings like photomultiplier voltage, slit width and scan speed.
 - (a) In the spectrum solely the water Raman peak at 310 nm should be visible (Fig. 2, black line). If the Raman peak has a different emission maximum or is not seen at all, then the monochromator most likely is decalibrated. See the instrument’s manual for recalibration.
 - (b) If larger particles are present in the buffer, indicated by “humps” (Fig. 2), these can be removed via filtration using a syringe filter with $0.22 \mu\text{m}$ pore size.
 - (c) Additional fluorescence bands arising from contaminations are usually not removable by filtration. In this case, fresh buffer solutions need to be prepared.

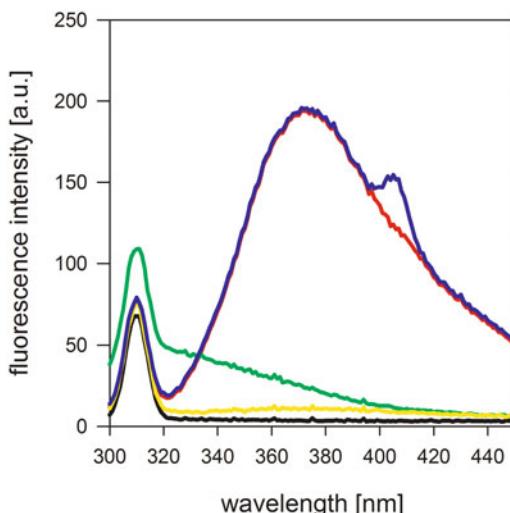


Fig. 2 Fluorescence emission spectra of different solutions. Black, clean buffer or pure water; yellow, little (acceptable) fluorescent contamination; green and red, unacceptable contaminations of buffers with fluorescent molecules; blue, typical signature of scattering particles visible as “spikes” within the fluorescence emission spectrum. Such spikes are usually found at a different wavelength, when the same spectrum is measured a second time. The red curve could be a contamination with proteins. Note: the contaminations visible in the yellow and green curves are typically only seen upon excitation in the UV. Therefore, such analysis is performed with excitation at 280 nm

- (d) If unsure about the quality of a buffer, fresh water can be measured as a reference (*see Note 1*).
- 2. Finally, measure also a spectrum of the pure buffer under the experimental conditions used in the final experiments, if the settings deviate from the settings above. Thereby, an appropriate baseline spectrum is recorded.

3.2 Analysis of Trp Fluorescence of a Protein (as It Is)

3.2.1 Standard Measurement of a Fluorescence Emission Spectrum

Experimental Procedure

The total concentration of tryptophans used in the experiment should be at least around 5 μM for measurement in a fluorimeter of standard sensitivity. Thus, if the protein, e.g., contains two tryptophans, the protein concentration should be at least 2–3 μM . The tryptophan fluorescence emission of the sample should be significantly higher than the Raman peak under all experimental conditions (*see*, e.g., Subheading 3.4.1).

1. Fill a cuvette with buffer, and measure a fluorescence emission spectrum ($\lambda_{\text{ex}} = 280 \text{ nm}$, $\lambda_{\text{em}} = 300\text{--}450 \text{ nm}$) if not already done.
2. Fill a cuvette with buffer containing the protein at an appropriate concentration.
3. Allow sufficient time for thermal equilibration or have the solutions pre-equilibrated, e.g., in a water bath (*see Note 5*).
4. Measure a fluorescence emission spectrum ($\lambda_{\text{ex}} = 280 \text{ nm}$, $\lambda_{\text{em}} = 300\text{--}450 \text{ nm}$). *See Notes 4, 8, 9* for other settings.
5. If the spectrum is too noisy, there are several possibilities for improvement:
 - (a) Increase the protein concentration. The upper limit here is the occurrence of the inner filter effect (*see* Subheading 3.3.3).
 - (b) Increase the width of the emission and/or excitation slits (*see Note 4* for limitations).
 - (c) Scan more slowly (*see also Note 9*).

Important: solely 5(a) will increase the protein fluorescence *relative* to the background. In all other cases, the background (buffer) must be measured again with the altered settings, since the intensity of Raman scattering is also altered.
6. Subtract the buffer spectrum from the protein spectrum.

Interpretation of the Spectrum

Isolated tryptophan in water has an emission maximum at approximately 350 nm (Fig. 1). Thus, if a fluorescence emission spectrum has a maximum around this value, the tryptophans are largely solvent accessible. Deeply buried tryptophans show an emission maximum at around 320 nm. In most cases, the fluorescence emission maximum is between 330 and 340 nm. If multiple

tryptophans are present in the protein, they have not necessarily the same environment and the same spectral shape (width and peak positions). Thus, the overall spectrum might be rather broad, if the differences are large. The suggested different classes of tryptophan spectra (*see* Subheading 1) are characterized by defined widths of the respective spectrum [5]. Thus, the spectral width of a measured spectrum might be compared to the suggested typical spectral widths [5]. A significantly broader spectrum might indicate different environments of multiple tryptophans of a protein. This can, e.g., be further investigated by quenching experiments (*see* Subheading 3.4).

3.2.2 Determining the Contribution of Tyrosine to the Emission Spectrum

Experimental Procedure

The relative contribution of tyrosine fluorescence emission to a total protein fluorescence emission spectrum can be determined by the procedure described in the following. Noteworthy, the method shows reliable results solely when only a single tryptophan is present or when the extinction coefficients of multiple tryptophans are all very similar at 275 and 295 nm (*see* also comments in Subheading 3.2.2). An example can be found in Weinberg [8].

1. Fill a cuvette with buffer, and measure two fluorescence emission spectra, one with $\lambda_{\text{ex}} = 275 \text{ nm}$ and $\lambda_{\text{em}} = 310\text{--}450 \text{ nm}$ and one with $\lambda_{\text{ex}} = 295 \text{ nm}$ and $\lambda_{\text{em}} = 310\text{--}450 \text{ nm}$.
2. Fill a cuvette with buffer containing the protein at an appropriate concentration (*see* Subheading 3.2.1). The concentration should be high enough to have a fluorescence emission well above the background (buffer spectrum) at 390 nm.
3. Thermally equilibrate the said sample, or use pre-equilibrated solutions, e.g., pre-equilibrated in a water bath (*see* Note 5).
4. Measure two fluorescence emission spectra, one with $\lambda_{\text{ex}} = 275 \text{ nm}$ and $\lambda_{\text{em}} = 310\text{--}450 \text{ nm}$ (spectrum S_{275}) and one with $\lambda_{\text{ex}} = 295 \text{ nm}$ and $\lambda_{\text{em}} = 310\text{--}450 \text{ nm}$ (spectrum S_{295}).
5. Subtract the respective buffer spectra from the protein spectra.
6. Determine the ratio of the intensities of S_{275} and S_{295} (S_{275}/S_{295}) at a wavelength, where only tryptophan emits, e.g., at 390 nm, yielding a scaling factor (F_{scale}).
7. Multiplication of S_{295} with F_{scale} yields the spectrum S_{trp} (*see* Note 10). S_{trp} reflects the contribution of tryptophan fluorescence emission to the fluorescence emission spectrum of the protein.
8. Subtraction of spectrum S_{trp} from spectrum S_{275} yields the spectrum S_{diff} . This spectrum reflects the spectrum of tyrosine contributing to the measured spectrum (S_{tyr}). An example is shown in Fig. 3.

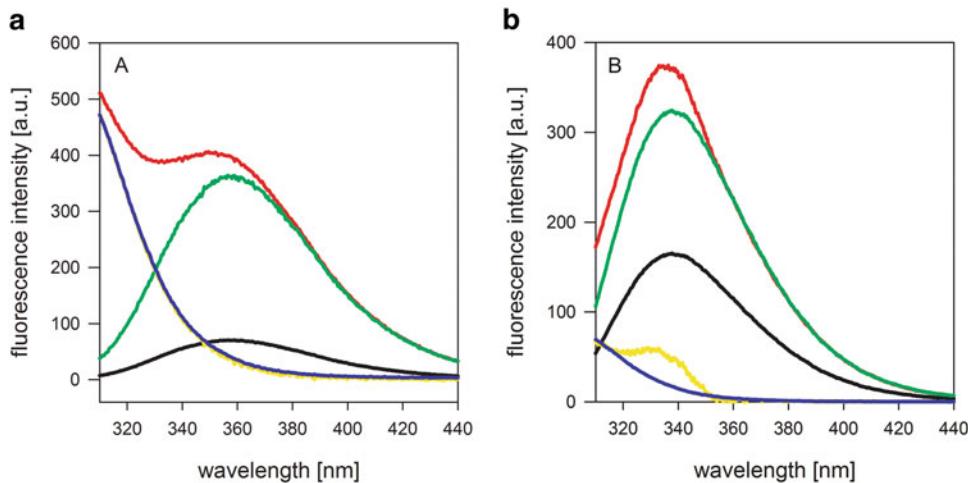


Fig. 3 The individual contributions of tryptophan and tyrosine fluorescence emissions to a spectrum (**a**) of a mixture of both amino acids and (**b**) of a protein. Fluorescence emission spectra were monitored upon excitation at 295 nm (S_{295} , black) or 275 nm (S_{275} , red). Green: tryptophan fluorescence emission spectrum calculated via rescaling (S_{trp}). The difference $S_{\text{diff}} = S_{275} - S_{\text{trp}}$ (yellow) represents the theoretical tyrosine contribution. Blue: measured tyrosine spectrum rescaled to S_{diff} at 310 nm. In (**a**) the shape of the theoretical and the true tyrosine spectrum is identical; in (**b**) a “hump” is observed

Interpretation of the Spectrum

When a mixture of pure tryptophan and tyrosine is analyzed as a positive control, the described process reveals a true tyrosine fluorescence emission spectrum (Fig. 3a). While possible, a similar spectrum is typically not observed when “real” proteins are analyzed: the calculated difference spectrum S_{diff} often does not have the same shape as a tyrosine spectrum but contains additional spectral features, which look like a “hump” on the tyrosine spectrum (Fig. 3b). This “hump” stems from contributions from the tryptophan fluorescence spectrum, which could not be predicted based on the procedure described above. Thus the calculated S_{trp} does not contain the complete tryptophan emission spectrum. The reason for the faulty S_{trp} might be a red edge excitation shift, multiple tryptophans differing in excitation and emission properties, or Förster resonance energy transfer.

Red Edge Excitation Shift

Red edge excitation shift (REES) describes the phenomenon of observing a shift in a fluorescence emission spectrum toward higher wavelengths when the excitation wavelength is increased (e.g., [9]). Thus, the tryptophan fluorescence emission spectrum obtained upon excitation at 275 nm has a different shape than the one obtained upon excitation at 295 nm. Thus, the spectrum obtained at 275 nm cannot be obtained from the latter by just multiplication by a scaling factor.

Multiple Tryptophans with Different Absorption Characteristics

When more than one tryptophan is present in a protein and when these differ in their extinction coefficients, the appropriate scaling factor is different for the different tryptophan species. In the absence of FRET and REES, the scaling factor is only determined by the ratio of the extinction coefficients at the 275 and 295 nm. Since these different scaling factors cannot be determined by the procedure described above, the calculated spectrum S_{trp} does not represent the real tryptophan fluorescence emission spectrum (*see* comments below in “Förster Resonance Energy Transfer”).

Förster Resonance Energy Transfer

Finally, Förster resonance energy transfer (FRET) between tyrosine and tryptophan residues can be a reason for the “hump,” as the effective quantum yield of tryptophan is different at the two excitation wavelengths when FRET occurs. When tyrosines are excited, part of the energy might be transferred via FRET to nearby tryptophan residues, from which the energy is emitted as fluorescence. This adds to the emission originating from direct excitation of tryptophan. Resonance energy transfer between tyrosine and tryptophan residues occurs frequently [2], but proving its existence unambiguously based on steady-state fluorescence without modification of the protein is difficult. The procedure described in Eisinger [10] has been applied a couple of times [11, 12]. Yet, the predicted and the measured normalized spectra do not always coincide.

The method is based on comparing the shape of the fluorescence excitation monitored at an emission wavelength where only tryptophan contributes and the absorption spectrum of the protein. The major bottleneck of this method is that the fractional absorption spectra of the tyrosine and tryptophan residues in the protein need to be available. These spectra have to be determined based on the overall absorption spectrum of the protein. Yet, the spectral properties of both amino acids depend to some degree on their local environment, and thus far no algorithm exists (to the best of our knowledge) which allows to recompose a full protein absorption spectrum based on superposition of the corresponding spectra of free tyrosine and tryptophan. At around 280 nm, where both spectra are flat, the observed absorption correlates with the tryptophan, tyrosine, and phenylalanine content, as used in [13] for obtaining a theoretical extinction coefficient from the amino acid sequence of the protein. In contrast, the exact values of the extinction coefficients of a tryptophan and therefore also the intensities in the excitation spectrum in the region around 290 nm, namely, at the flank of the spectra, are much more variable (Fig. 4).

Tryptophan fluorescence emission might change when substances are added to a protein solution and the substances interact with the protein (e.g., ligand binding) and/or alter the protein (e.g., protein denaturation or fluorescence quenching (*see* Subheading 3.4)).

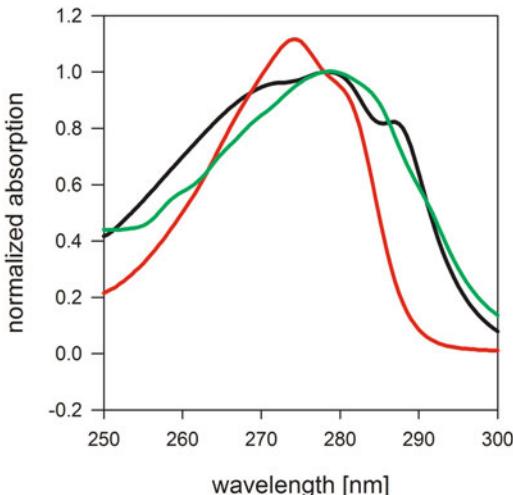


Fig. 4 Impact of the local environment on the tryptophan absorption spectrum. Normalized absorption spectra of tryptophan (black), tyrosine (red), as well as protein containing 5 tryptophans and 11 tyrosines (green). Note the pronounced area above 285 nm in the tryptophan absorption spectrum, which does not reconcile with the shape of free tryptophan. These spectral properties are due to altered environments of the tryptophan residues within the protein

3.3 Unintended Changes in Fluorescence Characteristics Upon Addition of Substances

3.3.1 A Substance Is (Unexpectedly) Fluorescent

However, there might be a more trivial reason for observing a changed fluorescence emission. The following possibilities should be considered prior to drawing wrong conclusions:

This problem might occur if, e.g., the interaction of a substance that binds with low affinity is measured. Here, the substance has to be used at high concentrations, where the intrinsic fluorescence emission of the substance might become detectable, even in cases of very low quantum yields. The measured fluorescence emission spectra can be corrected by the method described in the following (for an alternative, see **Note 11**):

1. Perform the titration experiment exactly like the “real” experiment, but replace the protein by buffer (dummy experiment).
2. Subtract the value determined in the dummy experiment from the value obtained in the actual experiment at each substance concentration.

Comment: In this case, the buffer baseline does not need to be subtracted from the protein and the substance spectrum prior to subtracting the latter from the protein spectrum.

3.3.2 Sticky Proteins

If proteins tend to stick to surfaces such as glass or plastic, the relative amount of protein lost while preparing an experiment is in particular high at low protein concentrations and increases by mechanical treatments such as mixing. Thus, if a protein solution in the cuvette is titrated with, e.g., a ligand (or a quencher), and thus the whole solution is mixed several times, an increasing amount of protein might stick to the cuvette walls. This will decrease the amount of protein in the detection volume. Thus, if the tryptophan fluorescence emission is expected to decrease upon addition of the ligand, this might look like a real binding event. Typically, a decrease observed due to sticky proteins does not saturate but proceeds in a rather linear manner with the number of titrations. Thus, in particular when an observed decrease in fluorescence intensity is small and nearly linear, the possibility of proteins sticking to the cuvette wall should be taken into account. Sticking can be tested by the following method (for an alternative, see Note 12):

1. Measure a fluorescence emission spectrum as described in Subheading 3.2.1.
2. Titrate buffer instead of the substance exactly like the substance in the actual experiment.
3. Correct the spectra for dilution, and plot the intensity at the wavelength used in the real experiments against the added volume.
4. If a significant change in fluorescence emission (intensity and/or position of the maximum) is observed, protein instability due to repeated mixing and/or sticking might be a problem.

If the protein does not tolerate repeated mixing, you have to use a fresh protein solution for each concentration of added substance (ligand, quencher, etc.).

3.3.3 Inner Filter Effects

The term “inner filter effect” describes the phenomenon of the light (excitation or emission) being partially “filtered away” in a cuvette and thus not being available anymore. This occurs if substances are present in a protein solution, which absorb light, either the excitation light or the emitted fluorescence light. Both will lead to reduced fluorescence emission intensities. Thus, the absorption properties of these substances determine whether an inner filter effect distorts a fluorescence emission and/or excitation spectrum. As a rule of thumb, an absorption of $A < 0.05$ is not problematic.

Since $A = \epsilon cd$, with d being the optical path length of the cuvette, ϵ the molar extinction coefficient, and c the concentration of the substance, the extent of the inner filter effect can be reduced by reducing the concentration of the absorbing substance or the

path length by changing the type of cuvette or by rotating it or by selecting a different excitation wavelength.

Up to $A = 0.3$, the inner filter effect can simply be corrected by the following theoretically derived equation:

$$F_{\text{corr}} = \frac{F_{\text{meas}}}{\text{CF}} \quad \text{CF} = 10^{-0.5(A_{\text{ex}} + A_{\text{em}})} \quad (1)$$

Here, F_{corr} is the intensity that would have been observed in the absence of the inner filter effect, and F_{meas} is the measured fluorescence intensity. A_{ex} and A_{em} are the absorptions of the sample at the excitation and emission wavelengths, respectively. In order to employ this equation simply based on the known absorption properties of the substances added to the protein solution, the effective path length is required, which is determined by the volume of the cuvette from which the fluorescence light is collected by the optics [14]. However, this depends on geometric properties specific for the instrument, and is usually not readily available. Thus, if possible, CF should be determined experimentally (*see* below), and only in case this is not possible, variant 4.15 should be employed.

The inner filter effect can arise due to the sample itself or upon addition of substances.

Inner Filter Effect due to Self-Absorption

At low concentrations, a protein's fluorescence increases linearly with concentration. Due to the inner filter effect, this will not be the case anymore when the fluorophore is used at high concentrations, since light reaching the fluorophores further down the optical path is already reduced in intensity due to absorption by other fluorophores. If measurements at high protein concentrations are required, this should be kept in mind.

Inner Filter Effect due to Addition of Substances

If a substance titrated to a protein solution does absorb at the excitation and/or emission wavelength, the fluorescence emission has to be corrected for the inner filter effect. The appropriate correction factor (CF in Eq. 1) can be experimentally determined as described in the following, if the substance does not interact with tryptophan. *See Notes 13 and 14 for alternative experimental methods and Note 15 for calculations based on absorption values.*

Experimental Determination of CF

Comment: Even if the CF can be determined experimentally for any absorption values, we discourage to perform experiments having a high level of inner filter effects. Should the corrections be much larger than the actual change to be measured, the intensities obtained after correction have large errors.

The simplest way to perform an experiment to determine CF is described below.

1. Prepare a tryptophan solution with about the same fluorescence intensity as the protein solution in the actual experiment.

2. Perform the experiment as performed with the protein, e.g., titrate the substance as described for quenchers in Subheading 3.4.2, employing the same concentration as in the “real” experiment.
3. The corrected fluorescence is obtained by dividing the fluorescence measured with the protein by the relative fluorescence change obtained with the tryptophan solution for each substance concentration (x).

$$F_{\text{prot,corr}}(x) = \frac{F_{\text{prot}}(x)}{\text{CF}} \quad \text{CF} = \frac{F_{\text{Trp}}(x)}{F_{\text{Trp}}(0)} \quad (2)$$

Normalizing both, the tryptophan and the protein titration, to the fluorescence in the absence of substance (to have both curves starting with the value of 1) and comparing the shape of the two curves allow to quickly assess whether there is an effect of the substances on the protein beyond the inner filter effect (Fig. 5).

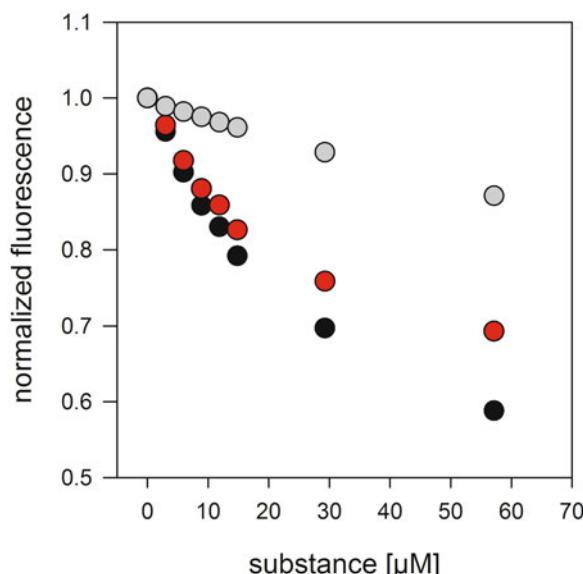


Fig. 5 Effect of inner filter effect on binding data. A tryptophan solution (gray) and a protein solution (black) were titrated with an absorbing substance. The values were normalized to the starting value (set as 1.0). Since the curve obtained for tryptophan (black curve), shows a clear trend, a correction for inner filter effect is necessary, yielding the red curve. In order to correct for the inner filter effect the values determined with tryptophan were used to correct the measured data following the procedure described in Subheading 3.3.3

3.4 Solvent Accessibility of Tryptophan Residues Tested via Fluorescence Quenching

Conformational changes induced by altered buffer conditions or binding of a ligand sometimes lead to shifts in the fluorescence emission maximum, but only to small changes of the fluorescence intensity. Since determination of emission maxima is cumbersome, one might take advantage of the possibility to probe changes in tryptophan's environment by changes in accessibility toward aqueous quenchers. In this process, the energy absorbed by the fluorophore is transferred to another substance during collisions (collisional quenching) or because a complex is transiently formed with the quencher (static quenching).

A number of substances can quench tryptophan fluorescence: iodide, acrylamide, oxygen, cesium, succinimide, dichloroacetamide, dimethylformamide, pyridinium hydrochloride, imidazolium hydrochloride, methionine, europium, and silver. Typically, iodide and acrylamide are used to probe the accessibility of tryptophans in aqueous protein solution.

3.4.1 Optimizing the Experiment

Typically, the fluorescence quenchers are used at rather high concentrations. Since the tryptophan fluorescence emission is significantly reduced at high quencher concentrations, fluorescent contaminations (*see Note 1*) or the Raman peak of water (*see Note 7*) can become prominent in the spectra. Thus, it is advisable to first measure a protein spectrum with the highest quencher concentration to find out whether the fluorescence intensity is still high enough and a spectrum of solely the quencher (without protein) to check for quencher fluorescence (*see Subheading 3.3.1*).

3.4.2 Experimental Procedure

1. Measure a fluorescence spectrum as described in Subheading 3.2.1, or measure the fluorescence intensity at a defined wavelength (*see Note 16*).
2. Repeat the measurement at increasing quencher concentrations. Typical concentration ranges are 0–0.5 M. As quenching typically is observed immediately after mixing the solutions, extended incubation times are not required. However, sufficient time for temperature equilibration has to be taken into account (*see Note 5*).

Typically, small aliquots of a quencher solution are added to a protein solution (e.g., $10 \times 10 \mu\text{L}$ in 1 mL solution) before the measurement. However, the solutions need to be properly mixed prior to the measurement without losing part of the volume (*see Note 6*).

3.4.3 Standard Analysis of Quenching Data

1. In case a full spectrum was measured at each quencher concentration, select a single wavelength for the analysis (*see Note 17*).

2. In case a sequential titration was performed, correct the fluorescence intensity at the selected wavelength for dilution of the protein concentration due to addition of the quencher solution.
3. In case the quencher gives rise to an inner filter effect, correct the intensity correspondingly (Subheading 3.3.3). This could, e.g., present a problem with acrylamide, depending on the excitation wavelength.
4. Divide the initial intensity (no quencher) by the corrected intensity at each quencher concentration, yielding (F_o/F).
5. Plotting (F_o/F) versus the quencher concentration yields a so-called *Stern-Volmer* plot, which, in the simplest case, can be fitted with a single straight line. This applies regardless whether collisional or static quenching occurs.

Deviations from a straight line could occur due to two reasons: (1) if the quencher exhibits simultaneously collisional and static quenching, the *Stern-Volmer* plot will show an upward curvature. Then a modified version of the *Stern-Volmer* plot should be used (see Note 18). (2) Another possibility is the occurrence of tryptophan residues with different accessibilities for the quencher. Here, a downward curvature will be observed. In the most extreme case, some of the tryptophans are completely buried and not accessible for the quencher. If this is the case, the fluorescence intensity does not continue to decrease with increasing quencher concentrations but remains on a certain level. Then, the fraction of accessible and shielded tryptophans can be determined by the following procedure:

3.4.4 Determining the Fractions Accessible vs. Buried Tryptophans

The fraction of accessible and completely shielded tryptophans within a given protein can be determined by employing the following method [2] (see Note 19):

1. Plotting $F_o/(F_o - F)$ versus $1/Q$ should yield a straight line.
2. The intercept of this line with the y -axis is the inverse of the fraction of accessible tryptophans f_a (intercept = $1/f_a$), while the slope yields $1/(f_a * K_{Qa})$, with K_{Qa} being the quenching constant for the accessible fraction (see supplementary file in [15] for an example).

Note that, if the tryptophan residues of the “shielded fraction” are not completely inaccessible or tryptophans with a range of different accessibilities are present, this analysis can only provide rough estimates.

4 Notes

1. Water kept for a prolonged time in plastic bottles is often slightly fluorescent. Thus, to measure a reference spectrum for clean buffer, use freshly tapped water. In addition, sometimes fluorescence substances are found in buffer solutions, possibly from the chemicals and the storing container or due to prolonged storage. Therefore, it is advisable to always check the buffers for unwanted fluorescence.
2. Obviously, proper wavelength calibration of the spectrometer is mandatory for comparing spectral properties of fluorescence, in particular when compared with other methods, such as absorption. Furthermore, the use of emission and excitation correction factors, recorded for each instrument, is mandatory. These take the wavelength dependence of the instrument's response into account. For further details, see the manual of the instrument.
3. A fluorophore can be excited and returns to the ground state only for a limited number of times. Thus, prolonged excitation, in particular with high light intensities, can lead to loss of fluorescent molecules. Such photo-bleaching is enhanced in the presence of oxygen, and thus, anaerobic conditions can increase the lifetime of fluorescent molecules.
4. Standard values for excitation and emission slit widths are 2–5 nm in both cases. When excitation spectra are monitored, the excitation slit should rather be at 2 nm than at 5 nm. The optimal value also depends on the lamp intensity and the fluorophore concentration. Reducing the slit width will reduce the intensity of the measured light. Furthermore, narrowing the excitation slit will reduce exposure of the sample to light, which can reduce photo-bleaching (*see Note 3*). This might be desirable, e.g., when long kinetics are measured. Enlarging the emission slit too much will distort the features of the emission spectrum, and thus, the value should not be larger than 10 nm. If distortion of the spectrum is suspected, the fluorescence emission spectrum should be checked for distortions by comparing the shape of the spectrum with a spectrum measured with narrower emission slits (e.g., after normalization, *see Note 10*).
5. The quantum yield is determined by the ratio of the rate of fluorescence relaxation (the rate by which the excited molecules lose their energy via fluorescence) relative to the rate of fluorescence relaxation plus the sum of all other possible processes [2]:

$$Q = \frac{k_f}{k_f + \sum k_i}.$$

Since the rate for collisional relaxation (loss of absorbed energy upon collision, e.g., with solvent molecules) increases with increasing temperature, the quantum yield Q and therefore the measured fluorescence intensity will decline. Thus, in order to obtain reproducible results, proper thermal control is crucial. This is especially important, when solutions have to be kept on ice and large volumes are added. In particular, when plastic cuvettes with limited contact to the cuvette holder are used, the required equilibration time will increase. If in doubt, measure a spectrum twice and check for superposition. Preferably, the buffer is kept tempered to the measuring temperature, e.g., in the water bath used for keeping the cuvette holder at constant temperature.

6. Proper mixing is essential but should not be conducted too rigorously in order to avoid formation of air bubbles. These will scatter the light and lead to an increased noise level or even distortions (“humps” on the spectrum) within the spectrum (Fig. 2).

If possible, solutions should be mixed outside of a cuvette. When performing a titration experiment, the cuvette should remain in the fluorimeter, and the solution should be gently pipetted up and down, e.g., with a plastic pipette. Multiple mixing events might lead to loss of protein (*see* Subheading 3.3.2).

In particular if viscous solutions are added, or with high concentration (e.g., typical quenchers), the success of the mixing should be checked by visually inspecting the solution for schlieren.

7. The Raman scattering peak (short: Raman peak) occurs due to inelastic scattering of the exciting light. Thus, the position depends on the excitation wavelength, and thus a buffer background emission spectrum has to be recorded separately for each excitation wavelength.
8. In the software controlling operation of a typical fluorescence spectrometer, a number of parameters have to be set by the user. In the following, typical values, which are usually compatible with the measurement of tryptophan fluorescence, are summarized (*see also* Subheading 2.1).

Avoid using conditions, where excitation and emission wavelength are too close together; here Rayleigh scattering can be harmful for very sensitive detectors. A difference of 20 nm has turned out to be a good choice ($\lambda_{\text{em,min}} = \lambda_{\text{ex}} + 20 \text{ nm}$).

The voltage for the photomultiplier can be set continuously or is selected from various values. Usually, a value in the

middle of the available range is appropriate for tryptophan fluorescence emission measurements, when the concentration is selected accordingly (see Subheading 3.2.1)

9. A standard value for the scan speed is 100 nm/min. In case a spectrum is too noisy, the speed should be reduced. If faster recording is required, for example, because successive spectra in a kinetic experiment are measured, the speed can be increased. The faster the scan speed, the lower the signal/noise ratio.
10. When normalized, a whole spectrum is multiplied by a constant. If, for example, one spectrum (S_A) has to be normalized to match another spectrum (S_B) at a certain wavelength (λ_{norm}), for each wavelength, the intensity of spectrum A ($I_A(\lambda)$) is multiplied by the ratio of the intensities of the two spectra at λ_{norm} : $I_{A,\text{new}}(\lambda) = I_A(\lambda) \frac{I_B(\lambda_{\text{norm}})}{I_A(\lambda_{\text{norm}})}$.

If a spectrum has to be normalized to the intensity at a certain wavelength, the operation is

$$I_{A,\text{new}}(\lambda) = \frac{I_A(\lambda)}{I_A(\lambda_{\text{norm}})}, \text{ thus at } \lambda = \lambda_{\text{norm}} \text{ one gets } I_{A,\text{new}}(\lambda) = 1.$$

11. Alternative procedure to correct for contribution of fluorescence of added substance:
 - Record fluorescence emission spectra at different concentrations of solely the substances in the range used in the actual binding experiment.
 - Plot the values at the wavelength employed for the “real” titration experiment against the substance concentration, and fit the data with a function that describes the curve well (e.g., linear, exponential).
 - Calculate the contribution of the substances alone to the whole fluorescence for each concentration in the actual experiment, based on the fitted function, and subtract the value from the measured fluorescence intensity.
12. Alternative procedure to test for sticking of protein during titration:
 - Measure a fluorescence emission spectrum of the sample as described in Subheading 3.2.1.
 - Select a ligand concentration, where a significant effect is seen, and add the volume containing the respective concentration in six aliquots. Mix between the individual additions and record a spectrum after each addition.
 - Start with a new sample of protein, add the total volume in one step, and mix and record a spectrum.

If the spectrum monitored in step 2 has a significantly lower intensity or a shifted emission maximum compared to the spectrum obtained in step 3, you might have a problem with sticky proteins or another mixing-induced alteration of the protein structure.

13. Alternatively, if the experiments with the tryptophan solution were performed at different concentrations, the curve determined with free tryptophan can be represented by an adequate fitting function (e.g., an exponential decrease), and based on this, the values for any substance concentration can be interpolated (Fig. 5, similar to the approach in **Note 11**).
14. The described procedure for experimental determination of the correction factor CF (Eq. 2) is only meaningful when the substance does not directly interact with tryptophan. If a substance does directly interact with tryptophan, leading to changes in the tryptophan fluorescence intensity, binding and inner filter effect cannot be simply separated. However, since only the absorption properties of the substance is of interest for the inner filter effect, a different substance, which does not interact with tryptophan but absorbs at the same wavelength of interest, can be used instead. The change in fluorescence upon titration of this dummy substance is then plotted against the concentration of the real ligand at the same absorption.
15. If experimentally not accessible, the next best choice to correct for inner filter effects is to calculate the CF via absorption characteristics of the substance. In this model, it is assumed that the effective optical path length corresponds to the cuvettes' path length, which is often a bit larger than the effective optical path length. Thus, the inner filter effect might be overestimated.
 - Determine the extinction coefficients of the substance causing the inner filter effect at the wavelength of interest (excitation wavelength and/or emission wavelength of the actual experiment) and the corresponding path length(s) d of the cuvette.
 - For each concentration c of the substance, calculate the absorption $A_{\text{ex}} (=c\varepsilon_{\text{ex}}d_{\text{ex}})$ at the excitation wavelength employed and the absorption $A_{\text{em}} (=c\varepsilon_{\text{em}}d_{\text{em}})$ at the emission wavelength employed in the actual experiment.
 - Use Eq. 1, employing these values, to calculate the correction factor CF.
16. In certain experiments, e.g., when titrating quencher, one could also use single wavelength measurements (e.g., $\lambda_{\text{ex}} = 280 \text{ nm}$, $\lambda_{\text{em}} = 340 \text{ nm}$ if the protein has an emission maximum at 340 nm). In this case, one should ensure to

measure the fluorescence intensity for a sufficiently long period of time, in order to be able to notice scattering from dust or air bubbles introduced by mixing. The emission slit width can be increased in such a case compared to the standard settings, if no shifts in the emission maximum are observed. This would increase the signal/noise ratio.

17. If the intensity of a single wavelength is to be plotted against an experimental variable (e.g., quencher concentration), but full spectra were recorded, it is often useful to not just pick the corresponding wavelength but take the average over the nearest five wavelengths or the like. In most cases, the spectra are broad; thus the intensity does not change significantly within this range. Yet, the signal/noise ratio increases by this procedure.
18. If the quenching mechanism involves both static (e.g., complex formation, quenching constant K_s) and collisional quenching (K_c), instead of $\frac{F}{F_o}$, the simple Stern-Volmer plot will exhibit upward curvature. Then, a modified value is plotted versus the quencher concentration Q :

$$\left(\frac{F}{F_o} - 1\right) \frac{1}{[Q]} = K_s + K_c + K_s K_c [Q].$$
This will yield a straight line with a slope of the product of both quenching constants and an intercept with the y-axis representing the sum of both.
19. If, e.g., only one of the two tryptophans present in a protein can be quenched, the Stern-Volmer plot will show downward curvature. Then the dependence of the fluorescence on the quencher concentration is given by

$$F(Q) = F_{bo} + \frac{F_{ao}}{1 + K_{Qa}[Q]},$$

with F_{ao} and F_{bo} being the fluorescence of the accessible and the buried tryptophan, respectively, in the absence of a quencher. Rearrangement leads to the following equation [2]:

$$\frac{F_o}{F_o - F(q)} = \frac{1}{f_a K_{Qa}[Q]} + \frac{1}{f_a}$$

with $F_o = F_{bo} + F_{ao}$ and f_a being the fractional contribution of the accessible species to the total spectrum $f_a = \frac{F_{ao}}{F_{bo} + F_{ao}}$.

References

1. Winter R, Noll F (1998) Methoden der Biophysikalischen Chemie. Springer, Stuttgart
2. Lakowicz JR (2006) Principles of fluorescence spectroscopy. Springer, New York
3. Tognotti D, Gabellieri E, Morelli E et al (2013) Temperature and pressure dependence of azurin stability as monitored by tryptophan fluorescence and phosphorescence. The case of F29A mutant. *Biophys Chem* 182:44–50. <https://doi.org/10.1016/j.bpc.2013.06.005>
4. Burstein EA, Vedenkina NS, Ivkova MN (1973) Fluorescence and the location of tryptophan residues in protein molecules. *Photochem Photobiol* 18(4):263–279

5. Reshetnyak YK, Burstein EA (2001) Decomposition of protein tryptophan fluorescence spectra into log-normal components. II. The statistical proof of discreteness of tryptophan classes in proteins. *Biophys J* 81(3):1710–1734. [https://doi.org/10.1016/S0006-3495\(01\)75824-9](https://doi.org/10.1016/S0006-3495(01)75824-9)
6. Shen C, Menon R, Das D et al (2008) The protein fluorescence and structural toolkit: Database and programs for the analysis of protein fluorescence and structural data. *Proteins* 71(4):1744–1754. <https://doi.org/10.1002/prot.21857>
7. Chen Y, Barkley MD (1998) Toward understanding tryptophan fluorescence in proteins. *Biochemistry* 37(28):9976–9982
8. Weinberg RB, Cook VR (2010) Distinctive structure and interfacial activity of the human apolipoprotein A-IV 347S isoprotein. *J Lipid Res* 51(9):2664–2671. <https://doi.org/10.1194/jlr.M007021>
9. Santos NC, Prieto M, Castanho MA (1998) Interaction of the major epitope region of HIV protein gp41 with membrane model systems. A fluorescence spectroscopy study. *Biochemistry* 37(24):8674–8682. <https://doi.org/10.1021/bi9803933>
10. Eisinger J (1969) Intramolecular energy transfer in adrenocorticotropin. *Biochemistry* 8(10):3902–3908
11. Kaylor J, Bodner N, Edridge S et al (2005) Characterization of oligomeric intermediates in alpha-synuclein fibrillation: FRET studies of Y125W/Y133F/Y136F alpha-synuclein. *J Mol Biol* 353(2):357–372. <https://doi.org/10.1016/j.jmb.2005.08.046>
12. Eisenhawer M, Cattarinussi S, Kuhn A et al (2001) Fluorescence resonance energy transfer shows a close helix–helix distance in the transmembrane M13 procoat protein. *Biochemistry* 40(41):12321–12328. <https://doi.org/10.1021/bi0107694>
13. Pace CN, Vajdos F, Fee L et al (1995) How to measure and predict the molar absorption coefficient of a protein. *Protein Sci* 4(11):2411–2423. <https://doi.org/10.1002/pro.5560041120>
14. Gu Q, Kenny JE (2009) Improvement of inner filter effect correction based on determination of effective geometric parameters using a conventional fluorimeter. *Anal Chem* 81(1):420–426. <https://doi.org/10.1021/ac801676j>
15. Root KT, Glover KJ (2016) Reconstitution and spectroscopic analysis of caveolin-1 residues 62–178 reveals that proline 110 governs its structure and solvent exposure. *Biochim Biophys Acta* 1858(4):682–688. <https://doi.org/10.1016/j.bbamem.2016.01.007>



Chapter 21

Structural Characterization of Membrane Protein Dimers

António J. Preto, Pedro Matos-Filipe, Panagiotis I. Koukos, Pedro Renault, Sérgio F. Sousa, and Irina S. Moreira

Abstract

Membrane proteins are essential vessels for cell communication both with other cells and noncellular structures. They modulate environment responses and mediate a myriad of biological processes. Dimerization and multimerization processes have been shown to further increase the already high specificity of these processes. Due to their central role in various cell and organism functions, these multimers are often associated with health conditions, such as Alzheimer's disease (AD), Parkinson's disease (PD), and diabetes, among others.

Understanding the membrane protein dimers' interface takes advantage of the specificity of the structure, for which we must pinpoint the most relevant interfacial residues, since they are extremely likely to be crucial for complex formation. Here, we describe step by step our own *in silico* protocol to characterize these residues, making use of known experimental structures. We detail the computational pipeline from data acquisition and pre-processing to feature extraction. A molecular dynamics simulation protocol to further study membrane dimer proteins and their interfaces is also illustrated.

Key words Membrane protein dimers, Machine learning, Feature extraction, Interfacial residues, Protein-protein interaction, Molecular dynamics

1 Introduction

Membranes are essential structures for life assuming many functions within cells, such as mobility and nutrient intake. To add to these, energy transduction, biosynthesis, and immunologic and nerve response are displayed in higher organisms [1]. These actions are often controlled by membrane proteins (MPs), which play essential roles such as ion and nutrient transport, communication with the extracellular environment, and signal transduction [2]. These proteins are also ubiquitous: 20–30% of genes of most organisms code for MPs [3]. Understanding these cellular

António J. Preto and Pedro Matos-Filipe contributed equally with all other contributors.

functions requires detailed knowledge of MPs' 3D structures and interactions. MPs frequently assemble as dimers or even higher-order oligomers. These higher-order assemblies can have specific roles that not necessarily coincide with those of their monomeric constituents [4, 5]. This makes the structural biology of MPs even more complex and demands the development of new experimental and theoretical methods to elucidate their function.

Experimental characterization of MPs is difficult as the membrane imposes obstacles to its manipulation, notably its purification and crystallization. As such MP structural studies can greatly benefit from *in silico* tools, since they provide useful approaches that complement, make use of, and add to the experimental results. In spite of the difficulties mentioned above, progress in experimental techniques has generated a growing body of structural information. For instance, the *mpstruc*—Membrane Proteins of Known 3D Structure—database from the Stephen White Laboratory at UC Irvine (available at <http://blanco.biomol.uci.edu/mpstruc/>) [6] now lists 817 unique membrane proteins whose 3D structures are known (as of August 30, 2018).

Dimers or higher-order assemblies of MPs are often the subject of computational studies [7]. These typically aim at predicting protein-protein interactions (PPIs) or hot spots (HS), interfacial residues that upon alanine mutation generate a binding free energy difference of 2.0 kcal/mol. We have recently developed a web server, SpotOn, for the prediction of HS in a soluble complex [8, 9]. However, we are still lacking reliable computational approaches that target the understanding of multimeric MPs. Here, we describe in detail our protocol to analyze a variety of biological and physic-chemical characteristics of interfacial interactions within MPs. By following this protocol, the reader has access to a comprehensive set of tools that target the understanding of MP dimerization and that can be used to construct any possible database regarding these biological systems. In the interest of readers, we also revise and explain the basics of machine learning (ML) and molecular dynamics (MD) techniques, which could potentially be used to further describe the MP interfacial residues. This tool contributes to further understanding the interaction of MP complexes and should be a valuable addition to the repertoire of methods/tools that aim to elucidate MP structure and function.

2 Materials

The goal of this chapter is to introduce a variety of tools to help readers characterize interfacial residues between two transmembrane monomers of a MP system. To achieve this, we built a pipeline of different scripts and tools able to process protein database files (.PDB) containing the two monomeric chains. We also

provide tools to retrieve a large amount of key features. In the end, the users can apply ML to this database to attain a predictive algorithm or MD if their main interest is to depict the mechanism of a particular system.

2.1 Machine Learning

ML has been defined in several different ways, yet there is a common concept of ML as the science of “getting computers to act without being explicitly told how to do so.” This means that ML is appropriate to solve real life problems in which there is no tool to deduct an answer as ML focuses on learning from experience. Usually, this means that the predictions from a ML model are not absolute; they improve when gathering more data [10]. Recently, many fields have experienced an increase in the accessible data. In particular, in the realm of biological problems where scarce data is many times a big obstacle, this was also true [11].

When referring to data, we also use the term instances, the available “samples” that we can feed to a predictor. Each of these instances is associated to the characteristic we want to predict and to the descriptors (features) that are associated or can be extracted from it. Furthermore, the features are components of the dataset instances that can be used to predict the target characteristic.

2.1.1 Supervised and Unsupervised Learning

ML is typically divided in two subfields, depending on its relation to the data: supervised and unsupervised learning (although there can also be the concept of semi-supervised learning, which can make use of both the previous approaches). This partition implies that the methods used to construct the prediction models are usually distinct for each different type of learning [12]. Supervised learning is the case in which the data fed to the prediction model is constituted of both input and output information. This means that every instance has a label. The labels inform the prediction model of the possible outputs, since they are the known values of the target prediction. A supervised learning model will make use of the labeled instances to predict cases in which the entries have unknown output values. The input information is constituted by all the features that characterize the instance, not including the output information. The output information on the data of a supervised learning model can lead to classification or regression models. A classification model is generated when the output is limited to a discrete number of possible values (classes). When there are only two possible classes, the problem is referred to as a binary problem. Regression models allow an infinite number of possible values. Unsupervised learning models do not have associated output values. This means that it is impossible to label an unknown new entry according to the starting data. However, these models are useful to identify patterns, since they can group instances according to their input information (features).

2.1.2 Dataset Construction: Instances

From a general point of view, the construction of a dataset is firstly conducted by gathering instances. Overall, instances are every entry that can be characterized and constitute a data point on a ML deployment pipeline. Gathering more data points will yield more information for the model to learn from. Usually, a dataset with more data points leads to stronger and more generalized models than its smaller counterparts. Regarding the type of data, instances can be many things, as long as they are able to be standardized along each other and can yield a pattern that relates towards the target prediction. In the case of classification models, it is preferable if the number of instances for each class is similar. This sometimes requires that the dataset is balanced to equilibrate the number of instances in each class. There are several sample (instances) selection processes such as up-sampling (artificially augmenting the lower populated classes) and down-sampling (lowering number of instances in the overpopulated classes). Another possibility is filtering out the irrelevant instances. For all these processes, there are well-developed mathematical approaches that are available in most ML-centered software [13].

2.1.3 Dataset Construction: Features

The number and quality of instances are certainly determinants for the quality of the upcoming predictions. What is associated or generated from those instances, however, can be equally important. The descriptors that we associate with instances are called features. Features are all the characteristics that can be associated to a data point. These features need to be relevant for the output prediction and be independent among each other. If this relationship is missing, the features can introduce biases, noise or overall weakening of the prediction capability of the model. What makes a feature relevant, however, is not always straightforward. Although there are approaches that can test the dataset for the most relevant features, the scientific/technical knowledge on the dataset is certainly an important factor in the selection and analysis of features. Freely available data from databases or, in some cases, data collected by the researchers, does not always comprise all the necessary information to generate strong models. For example, sometimes the problem of missing values must be addressed; in some cases, several approaches artificially generate values where they are not available. Nevertheless, it is always preferable to first mine alternative data sources that can yield the corresponding values. Feature extraction is the process in which, for the original raw data or instance data points, alternative features are generated to better describe the entries. Feature extraction is highly dependent on the type of data under focus.

2.1.4 Splitting the Dataset

“Generalized” is a term that has been mentioned several times to address the quality of a model. A ML model is said to be generalized if it can give accurate predictions about upcoming, unknown, instances. Indeed, a classification model may appear to

be highly reliable and yield good accuracy, but when faced with new information, it can output unreliable predictions due to its bias towards the input data. In order to overcome these issues, the data should be split into training and test sets. The training set should then be used to train the model, while the test set should not be present in the learning phase. This is usually performed several times, in a process called cross-validation (CV). When deploying CV, the computer is not informed of the specific instances that will be used for the training and test sets. Rather, it is told to split the dataset into two sets with a given percentage each time (commonly 70–30%), performing the training on the larger dataset and testing the model on the small test set for each case [14]. CV is usually performed several times for each run. Each time, the data undergoes randomized resampling, which leads to different training and test sets, in order to achieve an unbiased and generalized model [15].

2.1.5 Predictive Model Deployment

The application of predictive models on the dataset is probably the step for which ML is more commonly identified. A ML model engulfs a predictive approach that makes use of a mathematical or logical model to predict an outcome with a given degree of certainty. The specific model, however, differs for classification, regression, or clustering (unsupervised). Although some approaches are displaying consistently positive results for a wide array of problems, such as deep learning [16], there is not a perfect model to fit all possible problems. A thorough knowledge on the models and the data is the best way to maximize the use of both on the construction of a good predictor. Furthermore, there are approaches that allow the combination of several models, which are referred to as ensemble models. Ensemble models have been displaying competitive results in comparison to single complex models, even if sometimes the models that make up ensemble models are themselves simple. Such is the case of our own SpotOn predictor [8, 9].

2.1.6 Model Evaluation

The final evaluation of the models is one of the most important steps, since it can lead to the drawback of the process until the very start. Evaluating a ML model means assessing its validity upon unknown outcomes. There are many available metrics, but most supervised learning approaches rely on the relation between the predicted outcome and the actual outcome. This ratio is yielded from the test and validation sets when in comparison with the outcome predicted by the trained model. In the case of classification models, several common metrics derive from a confusion matrix (Table 1). Sensitivity (Eq. 1), specificity (Eq. 2), precision (Eq. 3), negative predictive value (NPV, Eq. 4), and F1-score (Eq. 7) are calculated directly from the values attained from the

Table 1
Confusion matrix

	Predicted: no	Predicted: yes
Actual: no	True negative (TN)	False positive (FP)
Actual: yes	False negative (FN)	True positive (TP)

confusion matrix. The false discovery rate (FDR, Eq. 5), although it can be calculated independently, can also be seen as the inverse of precision. Similarly, the false-negative rate (FNR, Eq. 6) is the inverse of sensitivity. The area under the receiver operating characteristic curve (AUROC, Eq. 8) depends on the true-positive rate (TPR, Eq. 1) and the false discovery rate (FDR, Eq. 5). By including different metrics on all the evaluated set of data points, AUROC constitutes a good metric for binary classification models [17]. All this can be attained by computing a confusion matrix (Table 1). The equations listed below (1)–(8) are all dependent on these values and can be used to address the particularities of a dataset.

Sensitivity formula

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (1)$$

Specificity formula

$$\text{TNR} = \frac{\text{TN}}{\text{FP} + \text{TN}} \quad (2)$$

Precision formula

$$\text{PPV} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (3)$$

Negative predictive value

$$\text{NPV} = \frac{\text{FP}}{\text{FP} + \text{TN}} \quad (4)$$

False discovery rate

$$\text{FDR} = \frac{\text{FP}}{\text{FP} + \text{TP}} = 1 - \text{PPV} \quad (5)$$

False-negative rate

$$\text{FNR} = \frac{\text{FN}}{\text{FN} + \text{TP}} = 1 - \text{TPR} \quad (6)$$

F1-score

$$\text{F1-score} = \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}} \quad (7)$$

AUROC

$$\text{AUROC} = \int_{-\infty}^{\infty} \text{TPR}(T)\text{FDR}'(T) dT \quad (8)$$

2.2 Molecular Dynamics Simulations

Molecular dynamics (MD) simulations are now a standard tool in the study of biomolecules. Following the publication of the first study describing an MD simulation of a protein (the bovine pancreatic trypsin inhibitor) already 40 years ago [18], this field has been one of the strong and enthusiastic developments, taking advantage of the astonishing computational progress that has characterized the past decades and of the good parallelization efficiency of most modern MD algorithms and more recently of the efficient use of GPUs. The dynamic properties of a protein have a profound effect upon its functional behavior. This is even more important when dealing with protein-protein interfaces. MD simulations allow the study of the dynamic properties of a system. They enable the complex and dynamic processes that take place in biological systems to be analyzed and provide atomistic detail concerning the individual particle motion as a function of time. Typical examples of application include the study of phenomena such as protein stability, molecular recognition, conformational changes, protein folding, and ion transport in biological systems.

MD methods normally used when dealing with such systems are based on the classic equations of motion, which are at the cornerstone molecular mechanics (MM). MM methods represent the energy of a system as a parametric function of the nuclear coordinates. These methods neglect both electrons and the quantum aspects of the nuclear motion and are based on classical Newtonian mechanics. They typically consider a rather simplified scheme of the interactions within a system. A “ball and spring” model is usually employed, in which the atoms are described as charged spheres of different sizes, whereas the bonds are described as springs with different degrees of stiffness. The van der Waals interactions are also an important step in the interaction between the modeled biomolecules and the solvent [19]. The neglect of the concept of electron forecloses any direct study of processes involving the formation or breaking of chemical bonds.

The energy of the system is split into a sum of contributions from different processes, including the stretching of bonds, the opening and closing of angles, rotations around simple bonds, etc. Each of these contributing processes is described by an individual expression, parameterized for a given set of standard atom types. Hence, a MM method is characterized not only by its functional form but also by the corresponding parameters, the two of which form a single entity termed force field. The parameters involved are typically derived from experimental data and/or

from calculations with higher-level methods (e.g., density functional theory (DFT)) for small molecules. The accuracy of the parameterization protocol is of paramount importance to the reliability of the force field, and special care should be taken when calculating properties other than those included in the parameterization process.

2.2.1 Biomolecular Force Fields

When preparing an MD simulation, one of the most critical choices to be made is the selection of a force field. As previously mentioned, the term force field encompasses the functional form, and the parameter sets are used to calculate the potential energy of a system of atoms or coarse-grained particles in molecular mechanics and molecular dynamics simulations. The development of a general force field, able to yield accurate results for a plethora of chemically different compounds, is a particularly hard, complex, and ungrateful task. To obtain high-accuracy calculations on such chemically different compounds, a careful parameterization of an extremely diverse and complete set of reference molecules is required. This is, in practice, an impossible mission. Hence, it is not surprising that currently available general force fields had to sacrifice accuracy for a wider applicability. Improved quality is normally achieved by developing specialized force fields, ensuring accurate calculations to be performed, albeit in a much more limited class of compounds. The limited structural diversity, in terms of building blocks, that characterizes most biological systems of relevance, including proteins, lipids, carbohydrates, and nucleotides, renders the development of specialized force fields for each one of these large and important classes of biological macromolecules a particular interesting and valuable strategy, with an almost infinite number of applications, given the large number of combinations of the correspondent biological structural basic elements that can be found in nature [20–24].

Different levels of detail can also be achieved using different types of force fields, including coarse-grained, united-atom, and all-atom force fields. All-atom (i.e., atomistic) force fields have explicit parameters for all the atoms in a system, including hydrogen atoms. United-atom force fields treat the hydrogen and carbon atoms in each methyl group (terminal methyl) and each methylene bridge as one interaction center, providing a cruder representation. Coarse-grained force fields, which are often used in long-time simulations of macromolecules such as lipids, proteins, nucleic acids, and multi-component complexes, provide even cruder representations for higher computing efficiency.

All-atom force fields are generally the most accurate, as they retain virtually all atomic-level interactions and can use time steps in the femtosecond range. While this makes them quite slow and computationally expensive compared with the other alternatives, the wide range of carefully tested parameters available for these

models, including proteins, lipids, nucleic acids, and small organic molecules, makes them reliable when it comes to quantitative prediction of properties such as motional time scales or interaction strengths, showing that this type of simulations has advantages, over those with a lower level of detail. They are also the most appropriate type of force field for simulating the interactions involving membrane proteins, as they provide an explicit representation of all the atoms and interactions at the interface, including those involving hydrogen atoms.

2.2.2 Simulating Biomembranes

In the particular case of membrane proteins, MD simulations offer an unparalleled way to analyze from a dynamic perspective the interactions established between MPs when inserted in the membrane, taking also into account the particularities of the water/membrane interface. Performing MD on membrane proteins requires the use of force fields for the representation of the protein, the water, and a model of the biomembrane.

AMBER, CHARMM, GROMOS, and OPLS are the most popular molecular mechanics force field families devised to describe biomolecular systems [25]. A common characteristic to these force fields is that the potential energy function is a function of pairs of atoms (it is two-body additive). Most force fields used in biological simulations apply the same form for the energy function, with harmonic terms for bonds and angles, Fourier series for torsions, and pairwise van der Waals and Coulombic interactions between atoms that are separated by three or more bonds. However, they are parameterized in conceptually different ways. Hence, individual parameters from different force fields should not be compared, as the parameterization scheme varies from force field to force field. Comparisons have to focus on the ability to reproduce observable data for a given system. Each of these force field families has specialized versions for the treatment of proteins and lipids. However, while for the treatment of proteins, accurate atomistic force field variations have been commonly in use with great success in a wide range of problems for a couple of decades, options to simulate lipids have remained for many years some steps behind. Furthermore, when combining membranes and proteins, it is important to take into account that the parameters used should be consistent, which means that the same general protocol should have been followed in the parameterization of all the associated molecules. This is especially important in the treatment of the non-bonded interactions (particularly charges, which decay slower with the distance), as the interactions between atoms within the protein or within the lipid bilayers have to be handled in the same fashion, and so should be the ones involving atoms in the protein with those in the bilayer. Such requirement is critical for an accurate representation of the interaction between the different partners.

More recently, dedicated force field extensions for the treatment of lipids have also been made available within all the major biomolecular force fields, levelling both fields and contributing to accurate representations of both the protein and the biomembrane [24, 26]. Within GROMOS, a number of variations have been made available through the years [27], including the parameter sets 45A3 [28], G53A [29], and G54A [30] and the popular Berger lipid FF [31], based on the original GROMOS non-bonded parameters and adopting a united-atom representation. More recent and improved versions include the 43A1-S3 [32] and the G53A6 [33] parameter sets. CHARMM [34, 35] included several parameter sets for atomistic simulations of lipids, including the CHARMM22 set (C22) [36], CHARMM27 (C27 and C27r) [17, 18, 37, 38], and the more recent CHARMM36 (C36) parameter set [39]. An extension for cholesterol has also been made available (C36c) [40]. Within AMBER, lipid simulations were done through many years with sets of lipid parameters based on re-parameterizations of the general AMBER force field [21, 41–43]. A specialized AMBER parameter set for lipids, called LIPID11 [44], was reported in 2012, followed by LIPID14 [45]. OPLS-AA also included parameters for lipids containing the DPPC bilayer [46]. Other common force field examples include MARTINI [47, 48], a coarse-grained force field, and Slipids [49].

The other critical partner is water, which also plays a fundamental role in mediating the interactions between different proteins and of these with the biomembrane. For the representation of the water molecules [50], common choices include the TIP3P (transferable intermolecular potential 3P) [51, 52], SPC (simple point charge) [53], and the SPC/E (extended simple point charge) [54] water models.

In spite of the increase in computational power that has characterized the past decades and advance in the technical sophistication of the software packages and force fields available, knowledge by the user still represents the single most determinant factor for a detailed simulation [55], particular of a complex problem such as this which involves the interaction between the protein, the membrane, and the water molecules. It is also important to consider that the length of the simulation is always a critical issue when discussing an MD simulation. Different chemical phenomena involve different time scales, and even when considering only proteins, it is important to keep in mind that their various characteristic types of motion have very different time scales, ranging from the fast and localized motion characteristic of atomic fluctuations to the slow and large-scale motions that involve rearrangements on the full protein. The length of the simulation should therefore be adequate to the type of motion under study. In addition, it is also important to retain that the different types of motion are interdependent and

coupled to one another, although for some practical applications, some types may be regarded as independent. In general, these motions can span over 20 orders of magnitude in terms of time scale, from femtoseconds (e.g., vibrations of bonds) to several seconds and even hours. Membrane protein recognition and membrane interaction in particular normally require a minimum simulation length from 20 to 100 ns for proper sampling of the properties associated [56].

3 Methods

All the python-associated methods of the work pipeline for this protocol are based on Python version 3.6 and its respective packages. Manually curated changes and visualization were performed with PyMOL [57] unless otherwise indicated. The methods and databases referred along the text can be consulted in Table 2, in Subheading 4. The overall workflow is depicted in Fig. 1.

3.1 Dataset

The final biological dataset should be made of protein dimers that obey a predetermined set of criteria and for which a variety of features can be calculated. Residues should also be labeled as interfacial and non-interfacial, a binary positive and negative class, to be used by the reader to effectively train a ML model.

3.1.1 Raw Data

We began by accessing the *mpstruc* [6] database in which all MPs are associated to a .PDB file corresponding to the experimentally determined structure, mostly through X-ray crystallography and more rarely by nuclear magnetic resonance (NMR). The list of MP protein identification codes is made available at <http://blanco.biomol.uci.edu/mpstruc/>, by means of Extensible Markup Language (.xml) files. The .xml files retrieved are available on the “XML representations” section of the website. We downloaded “XML for the β -barrel proteins” and “XML for the α -helical membrane proteins,” since the only remaining structures (monotopic MP) do not comply to one of the requirements for this database: constituting a transmembrane protein. The files were read with the python package *ElementTree*, and the final structures were retrieved from PDB [58] with an inbuilt method that employs Biopython [63] through a python pipeline. The structures were downloaded with the code below:

Table 2
List of methods or databases referred along the text

Method or database	URL	Description	Ref.
Mpstruc	http://blanco.biomol.uci.edu/mpstruc/	Known membrane protein structures	[6]
SpotOn	http://milou.science.uu.nl/cgi/services/SPOTON/spoton/	Soluble protein complexes, hot spot detection	[9]
Protein data bank	https://www.rcsb.org/	Known protein structures	[58]
AMBER	http://ambermd.org/	Biomolecular molecular dynamics simulation software	[59]
CHARMM	https://www.charmm.org/charmm/	Biomolecular molecular dynamics simulation software	[60]
GROMOS	http://www.gromos.net/	Biomolecular molecular dynamics simulation software	[61]
OPLS		Molecular dynamics force field for liquid simulations	[62]
PyMOL	https://pymol.org/2/	Molecular visualization software	[57]
ElementTree	https://pypi.org/project/elementtree/	Python package for XML files handling	
Biopython	https://biopython.org/	Python-based biological computational tools	[63]
MODELLE R	https://salilab.org/modeller/	Protein structures homology modeling tool	[64]
VMD	https://www.ks.uiuc.edu/Research/vmd/	Molecular modeling and visualization tool	[65]
PyDPI	https://pypi.org/project/pydapi/	Python-based chemoinformatics and bioinformatics package	[66]
iFeature	http://ifeature.erc.monash.edu/	Python-based package for protein feature extraction	[67]
DSSP	https://swift.cmbi.umcn.nl/gv/dssp/index.html	Protein secondary structure dictionary. Can be accessed via Biopython	[68]
Rosetta	https://www.rosettacommons.org/software	Molecular modeling program	[69]
Psiblast	https://www.ebi.ac.uk/Tools/sss/psiblast/	Protein sequence alignment tool	[70]
LipidBuilder	http://lipidbuilder.epfl.ch/home	Lipid creation, storage, and sharing	[37]
MemBuilder	http://bioinf.modares.ac.ir/software/mb2/	Membrane model initial configuration tool	[71]

(continued)

Table 2
(continued)

Method or database	URL	Description	Ref.
Insane	http://www.cgmartini.nl/index.php/insane	Lipid bilayer system setup tool	[39]
Packmol	http://m3g.iqm.unicamp.br/packmol/home.shtml	Molecular dynamics simulations initial configuration tool	[40]
InflateGRO	https://github.com/fuentesdt/MembraneProtein/blob/master/inflategro.pl	Biomembrane lipid simulation tool	[21]
Griffin		AMBER force field development	[41]
Alchemical	https://github.com/philipwfowler/alchemical-tutorial	Tool for incorporating multiple proteins into lipids	[72]
SHAKE		Molecular dynamics simulation box	[73]
LINCS		Molecular simulation constraint solver	[74]

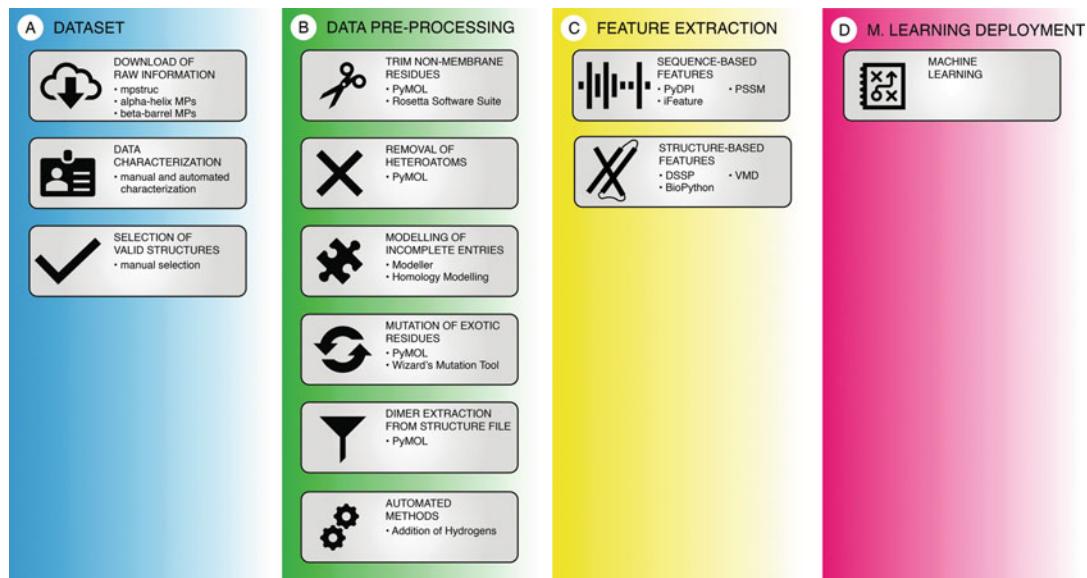


Fig. 1 Overall graphical depiction of the work pipeline for in silico characterization of MP interfacial residues database

```

import Bio
from Bio.PDB import *
import os

pdbl = PDBList()

###Append the PDB files to the following list before continuing
PDBlist2=[]

for pdb_entry in PDBlist2:
    try:
        print pdb_entry
        pdbl.retrieve_pdb_file(pdb_entry, pdir='PDB')
    except:
        continue

```

3.1.2 Data Characterization

To more easily and thoroughly select and manipulate the structures for the database, we performed an initial analysis. From the .xml files, by once again using the *ElementTree* package in python, we organized tables that characterized each structure with its PDB identification code, protein name, organism species and taxonomic domain, resolution of the structure, digital object identifier (DOI), protein subgroup name, and description. When the information was not available for all the referred fields, we attempted to retrieve it manually. In addition to the previous information, the number of chains, chains' biological names, in-file chain names (according to alphabetical labeling), and number of non-repeated chains were retrieved by analyzing the complexes with the Biopython [63] package. Stoichiometry was retrieved from PDB [58] with python through the selenium package for web automation. However, due to a high number of failing processes, this information was also partially manually retrieved and fully manually confirmed. Finally, all structures were manually analyzed to determine the complex class and the oligomer state. Regarding the complex class, there were the following possibilities: single chain, protein-ligand, protein-antibody, protein-protein, protein-peptide, and pore. When considering oligomer states, the options considered were single chain, multimer, multimer with at least one soluble protein (multimer*), membrane protein dimer (m-m), membrane protein-soluble protein dimer (m-s), membrane protein and soluble protein dimer (m-m*), or membrane protein dimer with soluble protein (both). All these characteristics were documented in separate tables for β -barrel and α -helical MPs. We also constructed a joint table to document all the complexes adding a descriptor to characterize them as β -barrel or α -helical. From the final characteristics, some were especially determinant on the selection of the structures: number of chains, complex class, and oligomer state.

3.1.3 Selection of Valid Dimeric Structures

At this point, we had a fully characterized set of MP structures, which needed to be further analyzed to ensure that the protein structures selected contributed positively to the purpose of this

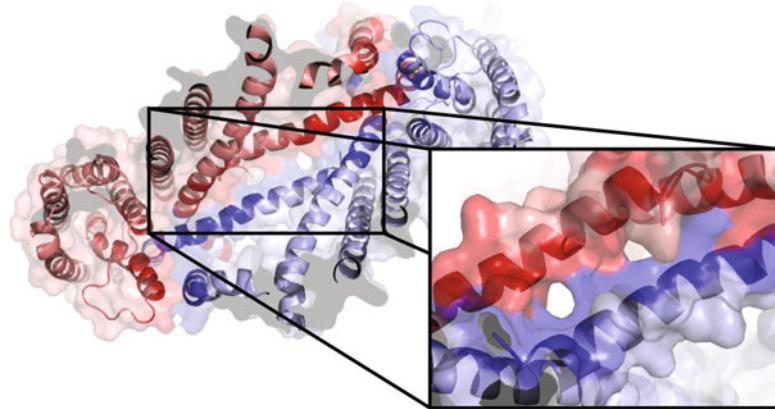


Fig. 2 Representation of an MP example (PDBid 5sy1 [75]) with the interface illustrated in the close-up

dataset. Although not referred extensively at the time, the choice of not using the monotopic membrane proteins was already under the scope of selection criteria. First of all, as said before, there are basic characteristics that this dataset must comply to: the structures must be MP and must contain more than one transmembrane chain. By choosing the *mpstruc* database, we ensured only MP would be present, and by excluding monotopic MP, we already excluded some non-transmembrane or single-chained proteins.

The characterization of the downloaded structures allowed us to automatically exclude, from the complex class and oligomer state, single chains, dimers in which one of the chains was a soluble protein (m-s), single MPs interacting with soluble small peptides (protein-peptide), pores, protein-antibodies (since antibodies are soluble proteins), and proteins with small organic or nonorganic ligands (protein-ligand). None of these structures had truly a PPI between two transmembrane MPs that contributed decisively to the upcoming prediction of interfacial residues. Regarding the remaining structures, not all were considered. The second round of selection excluded structures in which an excessive number of residues were unknown or incomplete. Furthermore, although some structures had two or more transmembrane chains, there was no clear PPI, in most cases due to a significantly large distance between them (much higher than the typical 5 Å). Finally, a very high amount of lipids between the chains was in some cases determinant for the exclusion, since it would clearly interfere in the interface. Figure 2 illustrates a typical MP dimer.

3.1.4 Data Pre-processing

Pre-processing treatment of any dataset is necessary and mandatory to ensure the success of ML application. First, since the quality of the data is vital to the training of the model, we describe the steps

employed to guarantee the viability of the models from the dataset built (3.1.4.i–3.1.4.v). Next, an additional pre-processing step from our automated pipeline is also described (3.1.4.vi).

(i) Trim Non-transmembrane Residues

The α -helical and β -barrel structures attained for the dataset were representative of transmembrane dimers but had nevertheless non-transmembrane residues or motifs. Visual inspection of the .PDB as well as information contained in these files was used to identify the transmembrane domain in order to posteriorly remove residues outside these regions. The process of identifying non-transmembrane residues is, on the automated part of this pipeline, underlined by the use of an inbuilt deployment of Rosetta [69].

(ii) Remove Heteroatoms

The downloaded structures were experimentally determined and, in many cases, present structural water molecules, metal ions, or small molecules. Also, since these are MP proteins, lipids are often found, even if not in sufficient amounts to exclude the structure for interfacial interference. For the purpose of this pipeline, these atoms would introduce unnecessary error and often intervene negatively with the measurements from the upcoming steps. Hence, a very simple but yet important step was to remove these atoms. They are listed as heteroatoms in the .PDB files, and PyMOL [57] scripts allow their easy removal. The PyMOL code snippet below displays a possible protocol for PyMOL visualization and heteroatom removal after loading all structures:

```
load pdb_file.pdb
bg_color white
set depth_cue, off
set fog, off
hide lines
show cartoon
util.cbc(selection='(all)',first_color=7,quiet=1,legacy=0,_self=cmd)
remove hetatm
```

(iii) Mutate Exotic Amino Acids

Besides heteroatoms, there were still other atoms capable of introducing error or nullifying some of our methods. In particular, many feature extraction methods are not prepared to deal with amino acids out of the ordinary set of 20. Selenomethionine residues are an example of amino acids that raise this problem. Furthermore, since these amino acids stand in the backbone of the protein, they could not be simply erased, as the heteroatoms were. To avoid this, such residues were mutated to their more usual counterparts (selenomethionines, e.g., were mutated to methionine) by using the PyMOL [57] “Mutagenesis” tool, available at

the “Wizard” section, and choosing the rotamer with the lowest number of crashes.

(iv) Model Incomplete Structures

Structures from the raw dataset that contained residues with many missing atoms were excluded, as referred before. However, in some cases, only a few residues were incomplete, and so the structure was kept. This being the case, homology modelling was used to rebuild the full residues. Following the extraction of the sequence of the original structure in the form of a FASTA file, we employed MODELLER [76], to generate an alignment file with both the original and the target sequences, which are the same in this case.

```
from modeller import *
import os

def align(input_pdb):

    pdb_name = input_pdb[0:-4]
    env = environ()
    aln = alignment(env)
    first_chain = "FIRST:A"
    last_chain = "LAST:B"
    fasta_name = pdb_name + ".fasta"
    mdl = model(env, file=pdb_name, model_segment=(first_chain, last_chain))
    aln.append_model(mdl, align_codes=pdb_name, atom_files=input_pdb)
    aln.append(file=fasta_name, align_codes=pdb_name)
    aln.align2d()
    q_ali_name = pdb_name + ".ali"
    aln.write(file=q_ali_name, alignment_format='PIR')
```

Also using MODELLER [76] with the previously generated alignment and the original structure as template, we generated models of the structure with the full residues (where they were previously incomplete). These models, since the template was the protein itself, had very little difference in structure, but they were complete and apt to be properly included in the dataset.

```
from modeller import *
from modeller.automodel import *
from modeller import soap_protein_od

def generate_model(input_ali):

    input_name = input_ali[0:-4]
    env = environ()
    a = automodel(env, alnfile=input_ali,
                  knowns=input_name, sequence=input_name,
                  assess_methods=(assess.DOPE,
                                  assess.GA341))

    a.starting_model = 1
    a.ending_model = 5
    a.make()
```

(v) Dimer Extraction from the Structure Files

Having standardized all the models to meet the criteria for the dataset purpose, the final step was the extraction of the

dimers from the structural files in which there were more than two valid possible dimer options. This was performed by visual inspection with PyMOL [57]. The following steps of the protocol are fully automated.

(vi) Add Hydrogens

Most structures available do not include hydrogen atoms, but their explicit representation is important since they can be involved in hydrogen bonds that contribute to stabilize the structure and, in particular, the interfaces. To add them, we employed the visual molecular dynamics (VMD) [65] software in a fully automated manner. Using a template (.tpl) file which stores the necessary commands that VMD [65] needs to properly add hydrogens to a .PDB file with two chains (see the code snippet below), a new file was generated specifying the commands for the specific structure under scope. This output file was then run, also from inside python, employing the python “os.system” in-built function.

```

package require psfgen
topology top_na.inp
alias residue HIS HSD
alias residue HOH TIP3
alias residue ZN ZN2
alias atom ILE CD1 CD
alias atom HOH O OH2
pdalias residue DG GUA
pdalias residue DC CYT
pdalias residue DA ADE
pdalias residue DT THY
foreach bp { GUA CYT ADE THY URA } {
    pdalias atom $bp "O5\*" 05'
    pdalias atom $bp "C5\*" C5'
    pdalias atom $bp "O4\*" 04'
    pdalias atom $bp "C4\*" C4'
    pdalias atom $bp "C3\*" C3'
    pdalias atom $bp "O3\*" 03'
    pdalias atom $bp "C2\*" C2'
    pdalias atom $bp "O2\*" 02'
    pdalias atom $bp "C1\*" C1'
}
segment A {
    pdb chain_A.pdb
    first none
    last none
}
segment B {
    pdb chain_B.pdb
    first none
    last none
}
coordpdb chain_A.pdb A
coordpdb chain_B.pdb B

guesscoord
writepdb final_file_HS.pdb
quit
exit

```

3.2 Feature Extraction

In order to perform ML on a given dataset, more than the instances (in these cases the MP structures' residues), there is a need to associate them with features, which, as already mentioned, are descriptors that characterize the instances. For this dataset, we engulf as many features as possible, provided they are reliable and their extraction/calculation can be automated. In the case of proteins, many of the features relate to their different hierarchical structures: primary (or sequence-based), secondary, and tertiary features. Furthermore, other features can be associated with the interfacial interaction or the proteins' evolutionary profile.

3.2.1 Sequence-Based Features

i) PyDPI

PyDPI [66] is a python package developed toward chemoinformatics, bioinformatics, and chemogenomics studies. We focused on PyPro, a PyDPI [66] sub-module that mines protein structural files in order to retrieve sequence-based features. To perform this, the sequences of both chains of each dimer were retrieved from the files with the aid of Biopython [63]. A “PyPro()” object was initialized, allowing for the next steps. This object read the sequences employing the *ReadProteinSequence()* method. Finally, we employed the *GetALL()* method on the same object, to retrieve a python dictionary in which the keys were the name of the feature and the values were the computed results. Notice that these features are not residue specific but rather associated to the whole sequence; hence, all the residues in one chain will have the same associated score. The features provided by this method are:

- 20 Amino Acid Composition (AAC) descriptors—the amount of each amino acid residue in the sequence.
- 400 Dipeptide Composition (DPC) descriptors—the amount of possible combinations of two subsequent amino acids.
- 240 Moreau-Broto autocorrelation (MBauto) descriptors.
- 240 Moran autocorrelation (Moranauto) descriptors.
- 240 Geary autocorrelation (Gearyauto) descriptors.
- 21 Composition descriptors.
- 21 Transition descriptors.
- 105 Distribution descriptors.
- 100 Quasi-Sequence Order (QSO) descriptors.
- 777 Pseudo Amino Acid Composition (PAAC), Amphiphilic Pseudo Amino Acid Composition (APAAC) and Conjoint Triad (CT) descriptors.

```

from pydipi import pypro
from pydipi import protein
from pydipi.pypro import GetAAIndex1, GetAAIndex23
from pydipi.pypro import PyPro
from pydipi.protein import getpdb, AACComposition
from pydipi.pypro import CTD

def amino_sequence_pypro(input_pdb):

    """Retrieves amino acid sequence from biopython structure object
    structure = pdb_parser(input_pdb)[0]
    seq_type = "ATOM"
    for model in structure:
        for chain in model:
            seq = ""
            for residue in chain:
                ## The test below checks if the amino acid
                ## is one of the 20 standard amino acids
                ## Some proteins have "UNK" or "XXX", or other symbols
                ## for missing or unknown residues
                if is_aa(residue.get_resname(), standard=True):
                    seq = seq + (str(three_to_one(residue.get_resname())))
                else:
                    continue
    return seq,seq_type

def pypro_features(input_pdb, chain_name):

    """All the sequence based features come from here
    path_to_pdb = main + "/" + input_pdb + "_HS_" + chain_name + ".pdb"
    res1, res2 = amino_sequence_pypro(path_to_pdb)
    protein = PyPro()
    protein.ReadProteinSequence(res1)
    all_features = protein.GetALL()
    key_list = []
    value_list = []
    for key, value in all_features.items():
        key_list.append(round_number(key))
        value_list.append(round_number(value))
    return key_list,value_list

```

i) iFeature

iFeature [67] is another package developed for python applications which encompasses several tools for bioinformatics deployment. Namely, it allows feature extraction. Similar to PyDPI [66], we used iFeature [67] to extract sequence-based features. iFeature [67] was called from inside the main script. To use it, the features must be called separately and computed from the sequence (FASTA file), since there is no unified function. Similar to PyDPI [66], the scores do not characterize specific residues, but rather the proteins' chains. Hence, all the residues from the same chain have the same associated score. The features retrieved from iFeature are:

- 240 Normalized Moreau-Broto (NMBroto) descriptors.
- 240 Moran descriptors.
- 39 Composition features.
- 60 Sequence-Order-Coupling Numbers (SOCNumber).
- 100 Quasi-Sequence-Order (QSOrder) descriptors.
- 50 Pseudo Amino Acid Composition (PAAC) descriptors.
- 80 Amphiphilic Pseudo Amino Acid Composition (APAAC) descriptors.

Even considering that some features are similar between PyDPI [66] and iFeature [67], we kept all of them since posterior ML methods chosen by readers will be able to rule out or ignore redundant features.

```
import os

executeCommands = True
def RunInOS(command):
    if executeCommands:
        os.system(command)

def write_iFeature(pdb_input, iFeature_path, feature_type):

    #####Write a .txt file with
    out_name = pdb_input[0:-4] + "_" + feature_type + ".txt"
    fasta_path = pdb_input[0:-4] + ".fasta"
    new_command = "python " + '"' + iFeature_path + '"' + " --file " + '"' + fasta_path + '"' + " --type "
+ feature_type + " --out " + out_name
    RunInOS(new_command)

def read_iFeature(input_feature_txt, feature_type):

    #####Read the previously written iFeature .txt
    read_file = open(input_feature_txt, "r").readlines()
    count = 0
    header = []
    chain_A = []
    chain_B = []
    for row in read_file:
        row = row.split()
        for cell in row[1:]:
            if count == 0:
                feature_name = feature_type + "_" + cell
                header.append(feature_name)
            if count == 1:
                chain_A.append(round_number(cell))
            if count == 2:
                chain_B.append(round_number(cell))
        count = count + 1
    return header, chain_A, chain_B
```

3.2.2 Secondary and Tertiary Features

In order to predict secondary structure, and later secondary and tertiary structure derived features, we employed DSSP (Database of Secondary Structure assignments for all Protein entries), from PDB [58]. The approach that we developed was to use the “sys” package from python to call DSSP from the shell, using the command:

```
dssp -i input_pdb > input_pdb_name_dssp.txt
```

This command generates a text file from which several features can be obtained. Before extracting the features, however, we used DSSP to attain the secondary structure prediction, which could then be manipulated for obtaining amino acid propensity in secondary structure motifs, as explained below. Regarding the features extracted with DSSP, they are residue specific and are:

- Relative accessible surface area (ASA).
- Phi angle.

- Psi angle.
- NH–O1 energy and relaxation (2 features).
- O–NH1 energy and relaxation (2 features).
- NH–O2 energy and relaxation (2 features).
- O–NH2 energy and relaxation (2 features).

```
def DSSP_features(input_pdb, feature_number):
    """Retrieves the features 0-13 described below, from biopython structure object
    ###0          DSSP index
    ###1          Amino acid
    ###2          Secondary structure
    ###3          Relative ASA
    ###4          Phi
    ###5          Psi
    ###6          NH-->O_1_relidx
    ###7          NH-->O_1_energy
    ###8          O-->NH_1_relidx
    ###9          O-->NH_1_energy
    ###10         NH-->O_2_relidx
    ###11         NH-->O_2_energy
    ###12         O-->NH_2_relidx
    ###13         O-->NH_2_energy
    to_break = [7,8,9,10]
    structure = pdb_parser(input_pdb)[0]
    dssp_name = input_pdb[0:-4] + "_dssp.txt"
    opened_file = open(dssp_name, "r").readlines()
    chain_SS_sequences = []
    useful = False
    feature_residues_A = {}
    feature_residues_B = {}
    residues_A_count = 0
    residues_B_count = 0
    feature_gaps = {"0": [0, 5], "1": [5, 10], "2": [10, 12], "3": [12, 14], "4": [14, 22], "5": [22, 33],
                    "6": [34, 38], "7": [38, 50], "8": [50, 61], "9": [61, 72], "10": [72, 83], "11": [83, 91], "12": [91, 97],
                    "13": [97, 103], "14": [103, 109], "15": [109, 115], "16": [115, 122], "17": [122, 129], "18": [129, 136], "19": [136,
                    150]}
    for row in opened_file:
        if useful == True:
            if row[feature_gaps["2"][-1]:feature_gaps["2"][-1]].replace(" ", "") == "A":
                residues_A_count = residues_A_count + 1
            if feature_number in to_break:
                feature_to_store = row[feature_gaps[str(feature_number)][-1]:feature_gaps[str(fea-
ture_number)][1]].replace(" ", "").split(",")
                feature_value = round_number(feature_to_store[-1])
                feature_residues_A[residues_A_count] = feature_value
            else:
                feature_value = round_number(row[feature_gaps[str(feature_number)][0]:fea-
ture_gaps[str(feature_number)][1]].replace(" ", ""))
                feature_residues_A[residues_A_count] = feature_value
            if row[feature_gaps["2"][-1]:feature_gaps["2"][-1]].replace(" ", "") == "B":
                residues_B_count = residues_B_count + 1
            if feature_number in to_break:
                feature_to_store = row[feature_gaps[str(feature_number)][0]:feature_gaps[str(fea-
ture_number)][1]].replace(" ", "").split(",")
                feature_value = round_number(feature_to_store[-1])
                feature_residues_B[residues_B_count] = feature_value
            else:
                feature_value = round_number(row[feature_gaps[str(feature_number)][0]:fea-
ture_gaps[str(feature_number)][1]].replace(" ", ""))
                feature_residues_B[residues_B_count] = feature_value
            if row[feature_gaps["0"][-1]:feature_gaps["0"][-1]].replace(" ", "") == "#":
                useful = True
    chain_SS_sequences.append(feature_residues_A)
    chain_SS_sequences.append(feature_residues_B)
    return chain_SS_sequences
```

Using the Biopython [63] module, we extracted the B-factor values from the complexes and constructed a windowed function that averages, for each residue, its values in a radius of 5 residues, generating a new feature (please check **Note 2** for further information).

```

import Bio
from Bio.PDB import *

def pdb_parser(input_pdb):

    ##### Parses pdb from .pdb file
    parser = PDBParser()
    pdb_name = input_pdb[0:-4]
    structure = parser.get_structure(pdb_name, input_pdb)
    return structure, pdb_name

def b_factor(input_pdb, input_atom = "CA"):

    ##### Returns b-factor for the input .pdb atoms, alpha carbon as default
    structure = pdb_parser(input_pdb)[0]
    chain_tagger = []
    for model in structure:
        for chain in model:
            count = 0
            b_factors = {}
            for residue in chain:
                count = count + 1
                for atom in residue:
                    if atom.get_name() == input_atom:
                        B = atom.get_bfactor()
                        feature_value = round_number(B)
                        b_factors[count] = feature_value
            chain_tagger.append(b_factors)
    return chain_tagger

def window(input_dicts, user_function, window_size = 5):

    ##### Iterates over previously achieved scores and builds new values by sliding a window of twice the size of the argument
    chain_storer = []
    for chain in input_dicts:
        output_dict = {}
        for entry in chain.keys():
            value_list = []
            current_value = chain[entry]
            value_list.append(current_value)

            for new_value in range(1, window_size):
                try:
                    value_list.append(chain[int(entry) + new_value])
                except:
                    continue
                try:
                    value_list.append(chain[int(entry) - new_value])
                except:
                    continue
            final_value = user_function(value_list)
            output_dict[entry] = final_value
        chain_storer.append(output_dict)
    return chain_storer

```

```

for new_value in range(1, window_size):
    try:
        value_list.append(chain[int(entry) + new_value])
    except:
        continue
    try:
        value_list.append(chain[int(entry) - new_value])
    except:
        continue
    final_value = user_function(value_list)
    output_dict[entry] = final_value
chain_storer.append(output_dict)
return chain_storer

```

Using the secondary structure predicted with DSSP, we created an amino acid propensity feature that associates to each of the possible secondary structural motifs the frequency of occurrence of each amino acid.

```

def amino_acid_propensity(sequence, secondary_sequence, secondary_structure_tag):
    #####Fetches the amino acid counts by secondary_structure:
    all_chains = []
    for chain_simple, chain_second in zip(sequence, secondary_sequence):
        PC_dict = {'G': 0, 'A': 0, 'V': 0, 'L': 0, 'M': 0, 'I': 0, 'F': 0,
                   'Y': 0, 'W': 0, 'S': 0, 'T': 0, 'C': 0, 'P': 0, 'N': 0,
                   'Q': 0, 'K': 0, 'R': 0, 'H': 0, 'D': 0, 'E': 0, 'X': 0}
        count = 0
        for residue, residue_TM in zip(chain_simple, chain_second):
            if residue_TM == secondary_structure_tag:
                count = count + 1
                PC_dict[residue] = PC_dict[residue] + 1
        for entry in PC_dict:
            try:
                PC_dict[entry] = float(PC_dict[entry])/float(count)
            except:
                continue
        new_PC_dict = {}
        new_count = 0
        for new_residue in chain_simple:
            new_count = new_count + 1
            feature_value = round_number(PC_dict[new_residue])
            new_PC_dict[new_count] = feature_value

    all_chains.append(new_PC_dict)
return all_chains

```

Furthermore, we used VMD [65] to find surface residues using the code below, in which the individual solvent-accessible surface area values considered were the ones described in Miller et al. [77].

```

mol new file_name.pdb
set allsel [atomselect top "all and chain name"]
set chain A
set tot_sasa [dict create ARG 241 TRP 259 TYR 229 LYS 211 PHE 218 MET 204 GLN 189 HIS 194 GLU 183 LEU 180
ILE 182 ASN 158 ASP 151 CYS 140 VAL 160 THR 146 PRO 143 SER 122 ALA 113 GLY 85]
set residlist [lsort -unique $allsel get resid]
set surf_list [list]
foreach r $residlist {
    set sel [atomselect top "resid $r and chain $chain"]
    set temp_rsasa [measure sasa 1.4 $allsel -restrict $sel]
    set temp_name [lsort -unique [$sel get resname]]
    set temp_id [lsort -unique [$sel get resid]]
    set temp_tot [dict get $tot_sasa $temp_name]
    set rsasa [expr $temp_rsasa/$temp_tot]
    if {$rsasa > 0.2} {lappend surf_list "$temp_id $temp_name"}
}

set filename "residues_surface_chain_name"
set fileId [open $filename "w"]
puts $fileId $surf_list
close fileId
exit

```

This code was then run on VMD, from the python main frame by issuing the command:

```
vmd -dispdev text -e get_surf_residues_chain_name.tcl
```

The surface residues are then associated with 1, while the non-surface residues are associated with 0, in the data table for ML deployment. Similar to surface residues, we calculated the interface residues, also using VMD [65], with specific model

scripts. The interfacial characterization (also binary) was the class we used to perform the ML deployment. The code used to perform this is displayed below.

```
mol file_name.pdb
set outfile [open "residues_number_according_to_chain" w]
set sel1 [atomselect top "protein and (chain A) and within 5 of (chain B)"]
$sel1 get {resid resname}
set sel2 [lsort -unique {$sel1 get {resid resname}}]
puts $outfile "$sel_number_according_to_chain"
close $outfile
quit
exit
```

In addition to the individual solvent-accessible surface area, lipid accessibility by residue was extracted as well. For this procedure, we used the “mp_lipid_acc”—an application included in the Rosetta Software Suite [69]. The following command was used, creating a new model .PDB file from information in the original .PDB file in addition to the span file. The obtained model contains a binary score column with the values 0 and 50, depending in the lipid accessibility—0 corresponding to inaccessible and 50 to accessible.

```
./rosetta_bin_linux_2018.09.60072_bundle/main/source/bin/mp_lipid_acc.static.linuxgccrelease -database
./rosetta_bin_linux_2018.09.60072_bundle/main/database -in:file:s [input_pdb_file] -mp:setup:spanfiles
[general span file, built in the previous command] -ignore_unrecognized_res
```

Like stated in section 3.1.4, it is essential to restrict the analysis exclusively to residues in the transmembrane region. This procedure was performed using the “mp_span_from_pdb” application, from the Rosetta Software Suite [69]. The following commands, having downloaded and built the Rosetta Software Suit in the same directory as the input files, output a span file, which lists the transmembrane regions of the chain under analysis. Below, is highlighted the command needed to attain this output which will generate a set of span files, one for each chain in the input .PDB file.

```
./rosetta_bin_linux_2018.09.60072_bundle/main/source/bin/mp_span_from_pdb.static.linuxgccrelease -
in:file:s [input_pdb_file] -ignore_unrecognized_res
```

An example of the output file, obtained from the sucrose-specific porin (PDBid: 1A0T) [78], is displayed below:

```
Rosetta-generated spanfile from SpanningTopology object
19 413
antiparallel
n2c
 3 11
 49 58
 64 72
 91 98
111 118
```

123	128
139	147
151	157
178	183
186	190
221	228
234	241
270	276
281	287
308	314
319	324
352	357
369	373
407	412

This procedure was performed using a Linux operating system. The Rosetta build may vary depending on the operating system in use.

3.2.3 Sequence Comparison Features

In this pipeline, there is also the possibility of adding position-specific scoring matrix (PSSM) features. To do this, the option must be specifically selected, since these features increase significantly the time of the process. While without PSSM features the in-house process can run within 3 min, for one dimer, it can take up to a few hours if this option is chosen, due to the alignment process. This happens because of the use of the “psiblast” [70] alignment for PSSM extraction. Please confer **Note 3** to learn how to employ “psiblast” to extract PSSM features.

3.3 Molecular Dynamics Simulations

3.3.1 Building the Protein-Membrane Model

A critical step before initiating any MD simulation involving a membrane protein is the creation of reasonable conformation containing the protein and the biomembrane model [21, 79]. Such structure has to represent or enable within a reasonable simulation time a realistic packing of the protein and lipids [35].

The choice of the biomembrane model represents an obvious approximation into the biology of the problem. Biomembranes are complex systems comprised by a wide range of different molecules, the balance of which determines its physical properties [80]. The relative composition of different molecular types in biomembranes can vary significantly between different organelles and cell types, ranging from 20% to 60% proteins, 30% to 80% lipids, and up to 10% carbohydrates. Among lipids, phospholipids, sphingolipids, and sterols are the major components present, in a ratio that determines a variety of properties, including surface charge, thickness, packing order, curvature, etc. [81]. However, most simulations represent the biomembrane as small bilayer patches containing just a few lipid moieties, often manually prepared.

Sometimes, a limited number of cholesterol molecules are introduced into the simulation to partially account for the heterogeneity of the bilayer. However, when preparing the biomembrane model, it is also important to take into account that building the biomembrane model as a simple assembly of the selected lipids is not enough, as extensive sampling would be required to transform the modeled bilayer structure into a reasonable biomembrane model [35]. Rather, a variety of solutions are currently available in current MD packages which contain ensembles of pre-equilibrated lipids [45, 82]. This approach is, however, not flexible, as it enables the simulation of only the limited set of alternatives explicitly present.

More recently a variety of membrane builder applications have been made available, enabling customization by the user in building biomembranes with custom compositions. These can be grouped into two main categories: web servers and distributed software. Web server membrane builder applications are normally user-friendly and relatively fast in distributing different components along the model generated. Examples include CHARMM-GUI [36], LipidBuilder [37], and MemBuilder [38]. However, the membranes generated are normally a long way from equilibrium, as optimizing biomembrane distribution and interactions can be an intensive process. Also, these servers are normally limited to specific molecular components and lipid types. Software-based application includes programs such as Insane [39] and Packmol [40] that can be downloaded and installed locally. They can generate any kind of densely packed structures, with the components and physical requirement imposed by the user and taking into account density optimization. However, such alternatives are still in their infancy, and generally consider only a crude description of the properties of the individual molecules in the density optimization.

i) Insertion of the Protein in the Membrane

Choosing the exact orientation and position of the membrane protein within the bilayer model can be a challenging task, as there is often no experimental data to guide how the protein should be placed relative to the bilayer. Nevertheless, an analysis on the amino acid residues defining the surface of the membrane protein can provide valuable clues, taking into account the amphipathic nature of the biomembrane. In fact, for many membrane proteins, there is a clear distinction between the type of amino acid residues located in the surface interacting the hydrophobic core of the biomembranes and those that interact with the lipid polar heads or with the solvent. The surface of the membrane protein interacting with the solvent or with the polar heads results from a careful balance between hydrophilic and hydrophobic amino acid side chains, with the first being prevalent [83]. However, the surface of the membrane protein that interacts with the hydrophobic core of the biomembrane presents an almost total absence of polar or

charged amino acid residues, being composed almost entirely by amino acid residues with hydrophobic side chains. Taken together, these tendencies normally help to define a perpendicular axis for membrane protein insertion in the biomembrane, which will be clearer for membrane proteins with a high degree of insertion in the biomembrane or for transmembrane proteins. These observations also help to identify an axis along the membrane protein that will be parallel to the biomembrane surface and that differentiates the predominately hydrophilic from the predominant hydrophobic surface regions of the protein and that will tend to be aligned with hydrophilic/hydrophobic regions of the membrane.

Once an orientation for the membrane protein is selected, the next step is its insertion into the membrane. Early alternatives relied on building the membrane around the protein or deleting a certain amount of lipids from a pre-equilibrated bilayer, creating a void into which the membrane protein could be placed. Both approaches tend to lead to excessive perturbations into the overall structure of the biomembrane and require careful equilibration. More recently, several specialized methods have been developed to ensure a smoother insertion of the membrane protein. Examples include the InflateGRO methods [21, 41], GRIFFIN [42], and GROMACS-based approaches such as *Mdrun_hole* [43] and *G_membed* [44]. Alchembed [72] is another popular alternative for membrane protein insertion, making use of soft-core potentials to slowly push the lipids away from the membrane protein during insertion. CHARMM-GUI also contains a functionality that enables the inclusion of one membrane protein per biomembrane. Its wide range of functionalities coupled to the ability to generate membranes with different compositions make CHARMM-GUI a popular starting point for generating custom biomembranes and inserting the protein.

ii) Selection of Force Fields

As described above, several force fields are currently available for the simulation of membrane proteins inserted in a bilayer. The study of membrane protein interactions with detail greatly encourages the use of atomistic force fields, and these enable the inclusion of all the main interactions formed along the protein interface with detail, enabling also the inclusion of the effect of the membrane and water. As reviewed previously, the major limitation in terms of atomistic force fields has been lipid representation. Presently, several atomic-level force fields able to describe a variety of lipid molecular types with accuracy have been made available, including CHARMM36, Lipid14, and SLipids, just to cite some of the most popular and recent. Although the level of accuracy of different alternatives can differ when performing extensive MD simulation on lipid properties, for the interactions between membrane proteins, alternatives as the ones mentioned would provide excellent results. Consistency in the force field selection for protein

and biomembrane is therefore presently the main issue to take into account, as the parameters used to describe the amino acid residues and lipid molecules should have been developed using a similar approach, based on the same type of overall principles. Hence, mixing different force field families and classes for protein and membrane is highly discouraged.

Final choice often emerges from the specific software package available to the user and his working knowledge. While some software packages like GROMACS and NAMD offer the user the possibility to choose from several different force fields, others like CHARMM and AMBER initially only supported their own specific force fields. Alternatives to convert topology files and input parameters from one software package to another have been increasingly made available, making it possible to use specific force fields in other software packages. However, this process is often difficult to master for the non-expert user, often still limiting the final choice.

iii) Simulating the Protein by Molecular Dynamics

Once the model system is prepared and force field and software are selected, the molecular dynamics simulation can be performed. This is normally run in a cuboid box, with periodic boundary conditions along the biomembrane plane (*xy*), typically with an integration time step of 1 or 2 fs (if bonds involving hydrogen are kept constrained with specialized algorithms as SHAKE [73] or LINCS [74]), and with a cutoff of 10–12 Å for the treatment of the non-bonded interactions. Given the complexity of the model, and the two phases that it comprises (water and biomembrane), special care must be taken when starting the simulation. First, to prevent disruption of the model system in the initial stages of the simulation, a set of MM minimizations are normally recommended. These normally start with a preliminary MM minimization in which all heavy atoms are frozen and only hydrogen atoms are allowed to optimize. Typically, in a second stage, the water molecules are optimized, while the gross of the biomembrane and protein is kept frozen. Subsequent steps involve the progressive release of the constraints imposed in the system (e.g., protein side chains, biomembrane tails, protein backbone, biomembrane heads, etc.), ending in a fully free MM minimization of the full system. Only after this stage is the system ready for MD simulation.

This normally starts with a stage in an NVT ensemble starting a 0 K, during which the temperature of the system is gradually increased up to the desired simulation temperature (typically 298 or 310 K). The densities of the water and biomembrane are evaluated through time, until equilibrated. The system is then switched to an NPT ensemble (or a variation), and the simulation continues at the desired temperature and pressure. The structural stability of the components analyzed is monitored through time through a RMSd analysis. Total simulation lengths of 20–500 ns after equilibration are normally pursued.

3.3.2 Analyzing the Interactions

Depending on the specific software chosen, a variety of tools can be used to analyze membrane protein interactions through MD simulations. Common examples include the analysis of hydrogen bonds, distances, radial distribution function, and solvent-accessible surface areas. An important feature is the analysis of the hydrogen interactions along the protein interface. Contrary to the static representation of systems, MD simulations offer the opportunity to assess the prevalence of specific hydrogen bonds during an entire simulation, enabling the determination of dynamic properties including average length and angle and their standard deviation, average time during which the hydrogen bond is kept, maximum occupancy, alternative hydrogen bonds involving the same group, etc. Similar procedures can be used to analyze other interactions or lengths, including the overall length and width of the protein, difference between centers of mass of different proteins, differences between average axis of α -helices, etc. [46].

Radial distribution function analysis is often applied to sample the accessibility of specific functional groups along the interface to solvent molecules, or the atoms from other molecular components. In a radial distribution analysis, a number of increasingly larger circles are traced around atoms or groups of reference, with increasing size (typically by 0.1 Å) typically covering a range of different radius from as much as 0–10 Å. Within each increasing circle, the number of interacting molecules (e.g., water) is determined for each recorded conformation of the simulation trajectory. From these analyses, a probability density for the type of interactions evaluated with distance emerges.

Another common property is the solvent-accessible surface area. This property can be used to analyze a simulation of a protein-protein biomembrane complex and determine the area of a given amino acid residue that is in contact with the solvent or with the biomembrane. SASA tools can also normally be adjusted to estimate the area of a specific amino acid residue that is in contact with other protein, the potential SASA lost upon protein-protein interaction, or the percentage of surface of an amino acid residue that is employed in the interaction, always from a dynamic perspective, as these quantities oscillate during a simulation. VMD [65] is a popular molecular visualization tool used to analyze molecular dynamics simulations. It contains a selection of built-in tools for automated analysis of these and other properties. AMBER, CHARMM, GROMACS also contain specific commands to analyze these properties.

Here, we described two identifiable processes. The first, assembling the dataset prepared to characterize a MP database and to potentially train an interfacial residues predictor. The processed forms of the original .PDB files, the final dimer database, and the description of the used structures from their original files will be available for use and constitute a landmark for protein dimer study.

The automated pipeline for the study itself is hereby explained in its individual steps and will be made fully available for any user to access it in an easy way. The second process was focused on special techniques and advices when applying MD to extra characterization of structural and mechanistic features of membrane proteins of particular interest for the user.

4 Notes

1. When adding hydrogens bonds, PyMOL [57] can also be used in a simpler manner. To do this the PDB file must be loaded, and the method “add_h” is called, adding the hydrogens. This method was not employed due to not being as thorough as VMD [65] and being of difficult employment on a Python-integrated pipeline; however, it can be used for simpler modifications.
2. Regarding the windowed function used to compute *B*-factors with the influence of surrounding residues, it can also be used for other purposes. The feature under scope can be different from the *B*-factor. The window radius of residues and the function that is employed on the values can be changed. This aims at reproducing the influence of other residues on a given residue.
3. The following command, having installed psiblast and downloaded the “non-redundant” proteins dataset, outputs a PSSM for one chain.

```
psiblast_path -query file.fasta -evalue 0.001 -num_iterations
2 -db nr -outfmt 5 -out pssm_file_ chain_name.txt -out-
ascii_pssm pssm_file_chain_name.pssm -num_threads 6
```

The output file can then be read to retrieve 42 PSSM-derived features.

Acknowledgments

Irina S. Moreira acknowledges support by the Fundação para a Ciência e a Tecnologia (FCT) Investigator programme—IF/00578/2014 (co-financed by European Social Fund and Programa Operacional Potencial Humano). This work was also financed by the European Regional Development Fund (ERDF), through the Centro 2020 Regional Operational Programme under project CENTRO-01-0145-FEDER-000008: BrainHealth 2020. We also acknowledge the grants POCI-01-0145-FEDER-031356 and

PTDC/QUI-OUT/32243/2017 financed by national funds through the FCT/MCTES and co-financed by the European Regional Development Fund (ERDF), namely, under the following frameworks: “Projetos de Desenvolvimento e Implementação de Infraestruturas de Investigação inseridas no RNIE”; “Programa Operacional Competitividade e Internacionalização—POCI”; “Programa Operacional Centro2020”; and/or State Budget.

References

1. Israelachvili JN, Marcelja S, Horn RG (1980) Physical principles of membrane organization. *Q Rev Biophys* 13(2):121–200
2. Chiu ML 2012 Introduction to membrane proteins. *Curr Protoc Protein Sci Chapter 29: Unit 29.1*
3. Gromiha MM, Ou YY (2014) Bioinformatics approaches for functional annotation of membrane proteins. *Brief Bioinform* 15 (2):155–168
4. Papadopoulos DK et al (2012) Dimer formation via the homeodomain is required for function and specificity of Sex combs reduced in *Drosophila*. *Dev Biol* 367(1):78–89
5. Damian M et al (2018) GHSR-D2R heteromerization modulates dopamine signaling through an effect on G protein conformation. In: *Proceedings of the National Academy of Sciences*
6. Moraes I et al (2014) Membrane protein structure determination - the next generation. *Biochim Biophys Acta* 1838(1 Pt A):78–87
7. Almeida JG et al (2017) Membrane proteins structures: a review on computational modeling tools. *Biochim Biophys Acta* 1859 (10):2021–2039
8. Melo R et al (2016) A machine learning approach for hot-spot detection at protein-protein interfaces. *Int J Mol Sci* 17(8):1215
9. Moreira IS et al (2017) SpotOn: high accuracy identification of protein-protein interface hot-spots. *Sci Rep* 7(1):8007
10. Bastanlar Y, Ozuysal M (2014) Introduction to machine learning. *Methods Mol Biol* 1107:105–128
11. Cook CE et al (2016) The European Bioinformatics Institute in 2016: data growth and integration. *Nucleic Acids Res* 44(Database issue): D20–D26
12. Greene CS et al (2016) Big data bioinformatics. *Methods* (San Diego, CA) 111:1–2
13. Gopinath RA, Burrus CS (1994) On upsampling, downsampling, and rational sampling rate filter banks. *IEEE Trans Signal Process* 42(4):812–824
14. Browne MW (2000) Cross-validation methods. *J Math Psychol* 44(1):108–132
15. Schumacher M, Hollander N, Sauerbrei W (1997) Resampling and cross-validation techniques: a tool to reduce bias caused by model building? *Stat Med* 16(24):2813–2827
16. Min S, Lee B, Yoon S (2017) Deep learning in bioinformatics. *Brief Bioinform* 18 (5):851–869
17. Hajian-Tilaki K (2013) Receiver operating characteristic (ROC) curve analysis for medical diagnostic test evaluation. *Caspian J Intern Med* 4(2):627–635
18. McCammon JA, Gelbin BR, Karplus M (1977) Dynamics of folded proteins. *Nature* 267:585
19. Mori T et al (2016) Molecular dynamics simulations of biological membranes and membrane proteins using enhanced conformational sampling algorithms. *Biochim Biophys Acta Biomembr* 1858(7, Part B):1635–1651
20. Neves RPP et al (2013) Parameters for molecular dynamics simulations of manganese-containing metalloproteins. *J Chem Theory Comput* 9(6):2718–2732
21. Coimbra JT et al (2014) Biomembrane simulations of 12 lipid types using the general Amber force field in a tensionless ensemble. *J Biomol Struct Dyn* 32(1):88–103
22. Sousa SF, Fernandes PA, Ramos MJ (2007) General performance of density functionals. *J Phys Chem A* 111(42):10439–10452
23. Comba P, Remenyi R (2003) Inorganic and bioinorganic molecular mechanics modeling—the problem of the force field parameterization. *Coord Chem Rev* 238–239:9–20
24. Nerenberg PS, Head-Gordon T (2018) New developments in force fields for biomolecular simulations. *Curr Opin Struct Biol* 49:129–138
25. Lopes PEM, Guvench O, MacKerell AD (2015) Current status of protein force fields

- for molecular dynamics. *Methods Mol Biol* (Clifton, NJ) 1215:47–71
26. Lyubartsev AP, Rabinovich AL (2016) Force field development for lipid membrane simulations. *Biochim Biophys Acta* 1858 (10):2483–2497
 27. Eichenberger AP et al (2011) GROMOS++ software for the analysis of biomolecular simulation trajectories. *J Chem Theory Comput* 7 (10):3379–3390
 28. Chandrasekhar I et al (2003) A consistent potential energy parameter set for lipids: dipalmitoylphosphatidylcholine as a benchmark of the GROMOS96 45A3 force field. *Eur Biophys J* 32(1):67–77
 29. Oostenbrink C et al (2004) A biomolecular force field based on the free enthalpy of hydration and solvation: the GROMOS force-field parameter sets 53A5 and 53A6. *J Comput Chem* 25(13):1656–1676
 30. Poger D, Van Gunsteren Wilfred F, Mark Alan E (2009) A new force field for simulating phosphatidylcholine bilayers. *J Comput Chem* 31 (6):1117–1125
 31. Berger O, Edholm O, Jähnig F (1997) Molecular dynamics simulations of a fluid bilayer of dipalmitoylphosphatidylcholine at full hydration, constant pressure, and constant temperature. *Biophys J* 72(5):2002–2013
 32. Chiu S-W et al (2009) An improved united atom force field for simulation of mixed lipid bilayers. *J Phys Chem B* 113(9):2748–2763
 33. Jämbbeck JP, Lyubartsev AP (2012) Derivation and systematic validation of a refined all-atom force field for phosphatidylcholine lipids. *J Phys Chem B* 116(10):3164–3179
 34. Pastor RW, MacKerell AD (2011) Development of the CHARMM force field for lipids. *J Phys Chem Lett* 2(13):1526–1532
 35. Zhu X, Lopes PEM, Mackerell AD (2012) Recent developments and applications of the CHARMM force fields. *Wiley Interdiscip Rev Comput Mol Sci* 2(1):167–185
 36. Feller SE et al (1997) Molecular dynamics simulation of unsaturated lipid bilayers at low hydration: Parameterization and comparison with diffraction studies. *Biophys J* 73 (5):2269–2279
 37. Feller SE, MacKerell AD Jr (2000) An improved empirical potential energy function for molecular simulations of phospholipids. *J Phys Chem B* 104(31):7510–7515
 38. Klauda JB et al (2005) An ab initio study on the torsional surface of alkanes and its effect on molecular simulations of alkanes and a DPPC bilayer. *J Phys Chem B* 109(11):5300–5311
 39. Klauda JB et al (2010) Update of the CHARMM all-atom additive force field for lipids: validation on six lipid types. *J Phys Chem B* 114(23):7830–7843
 40. Lim JB, Rogaski B, Klauda JB (2012) Update of the cholesterol force field parameters in CHARMM. *J Phys Chem B* 116(1):203–210
 41. Wang J et al (2004) Development and testing of a general Amber force field. *J Comput Chem* 25(9):1157–1174
 42. Dickson CJ et al (2012) GAFFlipid: a General Amber Force Field for the accurate molecular dynamics simulation of phospholipid. *Soft Matter* 8(37):9617–9627
 43. Ogata K, Nakamura S (2015) Improvement of parameters of the AMBER potential force field for phospholipids for description of thermal phase transitions. *J Phys Chem B* 119 (30):9726–9739
 44. Skjekvik AA et al (2012) LIPID11: a modular framework for lipid simulations using amber. *J Phys Chem B* 116(36):11124–11136
 45. Dickson CJ et al (2014) Lipid14: the amber lipid force field. *J Chem Theory Comput* 10 (2):865–879
 46. Maciejewski A et al (2014) Refined OPLS all-atom force field for saturated phosphatidylcholine bilayers at full hydration. *J Phys Chem B* 118(17):4571–4581
 47. Marrink SJ et al (2007) The MARTINI force field: coarse grained model for biomolecular simulations. *J Phys Chem B* 111 (27):7812–7824
 48. Marrink SJ, De Vries AH, Mark AE (2004) Coarse grained model for semiquantitative lipid simulations. *J Phys Chem B* 108 (2):750–760
 49. Jämbbeck JPM, Lyubartsev AP (2012) Derivation and systematic validation of a refined all-atom force field for phosphatidylcholine lipids. *J Phys Chem B* 116(10):3164–3179
 50. Demerdash O, Wang LP, Head-Gordon T (2018) Advanced models for water simulations. *Wiley Interdiscip Rev Comput Mol Sci* 8(1):e1355
 51. Jorgensen WL et al (1983) Comparison of simple potential functions for simulating liquid water. *J Chem Phys* 79(2):926–935
 52. Neria E, Fischer S, Karplus M (1996) Simulation of activation free energies in molecular systems. *J Chem Phys* 105(5):1902–1921
 53. Berweger CD, van Gunsteren WF, Müller-Plathe F (1995) Force field parametrization by weak coupling. Re-engineering SPC water. *Chem Phys Lett* 232(5–6):429–436

54. Berendsen HJC, Grigera JR, Straatsma TP (1987) The missing term in effective pair potentials. *J Phys Chem* 91(24):6269–6271
55. Wong-Ekkabut J, Karttunen M (2016) The good, the bad and the user in soft matter simulations. *Biochim Biophys Acta Biomembr* 1858 (10):2529–2538
56. Khalili-Araghi F et al (2013) Molecular dynamics simulations of membrane proteins under asymmetric ionic concentrations. *J Gen Physiol* 142(4):465–475
57. DeLano WL (2002) The PyMOL molecular graphics system. Delano Scientific, San Carlos, CA
58. Berman HM et al (2000) The protein data bank. *Nucleic Acids Res* 28(1):235–242
59. Case DA et al (2005) The Amber biomolecular simulation programs. *J Comput Chem* 26 (16):1668–1688
60. Brooks BR et al (2009) CHARMM: the biomolecular simulation program. *J Comput Chem* 30(10):1545–1614
61. Christen M et al (2005) The GROMOS software for biomolecular simulation: GROMOS05. *J Comput Chem* 26(16):1719–1751
62. Das A, Ali SM (2018) Molecular dynamics simulation for the test of calibrated OPLS-AA force field for binary liquid mixture of tri-isooamyl phosphate and n-dodecane. *J Chem Phys* 148(7):074502
63. Cock PJA et al (2009) Biopython: freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics* 25 (11):1422–1423
64. Webb B, Sali A (2014) Protein structure modeling with MODELLER. *Methods Mol Biol* 1137:1–15
65. Humphrey W, Dalke A, Schulten K (1996) VMD: visual molecular dynamics. *J Mol Graph* 14(1):33–38
66. Cao DS et al (2013) PyDPI: freely available python package for chemoinformatics, bioinformatics, and chemogenomics studies. *J Chem Inf Model* 53(11):3086–3096
67. Chen Z et al (2018) iFeature: a python package and web server for features extraction and selection from protein and peptide sequences. *Bioinformatics* 34(14):2499–2502
68. Kabsch W, Sander C (1983) Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers* 22(12):2577–2637
69. Leaver-Fay A et al (2011) ROSETTA3: an object-oriented software suite for the simulation and design of macromolecules. *Methods Enzymol* 487:545–574
70. Altschul SF et al (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res* 25 (17):3389–3402
71. Ghahremanpour MM et al (2014) Mem-Builder: a web-based graphical interface to build heterogeneously mixed membrane bilayers for the GROMACS biomolecular simulation program. *Bioinformatics* 30 (3):439–441
72. Jefferys E et al (2015) Alchembed: a computational method for incorporating multiple proteins into complex lipid geometries. *J Chem Theory Comput* 11(6):2743–2754
73. Ruymgaart AP, Elber R (2012) Revisiting molecular dynamics on a CPU/GPU system: Water Kernel and SHAKE parallelization. *J Chem Theory Comput* 8(11):4624–4636
74. Hess B, Bekker H, Berendsen HJC, Fraaije JG (1997) LINCS: a linear constraint solver for molecular simulations. *J Comput Chem* 18:1463–1472
75. Chen Y et al (2016) Structure of the STRA6 receptor for retinol uptake. *Science* 353 (6302):aad8266
76. Eswar N et al (2006) Comparative protein structure modeling using modeller. *Curr Protoc Bioinformatics Chapter 5:Unit 5.6*
77. Miller S et al (1987) Interior and surface of monomeric proteins. *J Mol Biol* 196 (3):641–656
78. Forst D et al (1998) Structure of the sucrose-specific porin ScrY from *Salmonella typhimurium* and its complex with sucrose. *Nat Struct Biol* 5:37
79. Chavent M, Duncan AL, Sansom MSP (2016) Molecular dynamics simulations of membrane proteins and their interactions: from nanoscale to mesoscale. *Curr Opin Struct Biol* 40:8–16
80. Goñi FM (2014) The basic structure and dynamics of cell membranes: an update of the Singer–Nicolson model. *Biochim Biophys Acta Biomembr* 1838(6):1467–1476
81. van Meer G, Voelker DR, Feigenson GW (2008) Membrane lipids: where they are and how they behave. *Nat Rev Mol Cell Biol* 9 (2):112–124
82. Kulig W, Pasenkiewicz-Gierula M, Rog T (2015) Topologies, structures and parameter files for lipid simulations in GROMACS with the OPLS-aa force field: DPPC, POPC, DOPC, PEPC, and cholesterol. *Data Brief* 5:333–336
83. Lee AG (2005) How lipids and proteins interact in a membrane: a molecular approach. *Mol BioSyst* 1(3):203–212

INDEX

A

- Amino acid
flexibility properties 106
physicochemical properties 105, 106
Anfinsen, C.B. 16, 74, 102, 374

B

- β - α - β motifs 104–106, 117, 118
 β -graph 50
 β -hairpin motifs 26, 104
 β -hairpins vii, xi, 16, 22, 24, 26, 31, 38, 73, 75, 87–91, 93, 102–106, 110, 112–116, 193, 206, 221, 227
 β -ladders 50, 58, 62
 β -sheets 6, 9, 24, 33, 50, 56, 58, 62, 247, 248, 287, 297, 305

C

- CAR T-cell therapies 188
CATH database 1, 4, 6, 8, 48, 50–52, 284, 347
CD28 188
CD3z 188
CD8 188, 196, 215
CE-symm 196–198, 211
Chain identifier 50, 52, 53, 59, 176
Cn3D 195–197, 199, 212, 213
Complex supersecondary structures 31, 32, 34
Cytokine receptors 205, 208

D

- Database of simple super-secondary structure elements-ArchDB2014 31
DSSP-Database of Secondary Structure assignments 423

F

- FN3 domains 198, 203, 205, 207, 208
Fragment assembly approach
QUARK 19, 36
ROSETTA 19, 21
Fragment library 20, 37, 283–292

G

- Gene duplications ix, 188, 222
GPCRs 56, 68, 189, 199, 203, 204, 208–213, 215

H

- Homology modeling 1, 10, 17, 39, 82, 84, 86, 87, 102, 174, 180, 341, 342, 419

I

- iCn3D 193, 196–198, 200, 201, 216
Immunoglobulin (Ig) ix–xi, 188, 189, 193–195, 197, 198, 201–203, 205–208, 210, 316, 321, 330, 332, 334, 335, 356, 361
Immunoglobulin fold (Ig-fold) 188, 192, 198, 202, 203, 207
Immunotoxins 188

J

- Jmol 196, 198

M

- Matthews correlation coefficient (MCC) 109, 112–114
Membrane proteins (MPs) xi, 34, 38, 189, 203, 209, 212, 403–433
Molecular dynamics (MD) x, 102, 134, 284, 297–310, 404, 405, 408–413, 424–433

P

- Pfam database 50–53, 68, 348
Protein structure prediction x, 15–40, 147–162, 263, 285, 291, 364
Protodomains xi, 188–216
Pseudosymmetry 188–192, 194–199, 202–205, 208, 209, 211–213, 215
PyMOL-molecular visualization tool 51, 181

Q

- Quaternary symmetry 189, 191, 192, 196, 198, 202, 204

R

- RNA protodomains 201, 203, 211
 Rossmann fold vii, 1–11, 188

S

- Secondary structures
 annotation 47–70, 76
- Secondary structure assignment (SSA) 35, 50, 51, 55, 56, 59–64, 69, 75, 332, 423
- Secondary structure elements (SSE) vii–x, 2, 6, 21, 22, 31, 33, 35, 36, 47–70, 73, 74, 77, 84, 90, 102, 124, 191, 192, 194, 204, 207, 213, 215, 263
- Secondary structure prediction
 SPIDER3 81–83
 SPINE X 22, 81, 102, 106
 SSpro 81, 82, 87, 92
- Secondary structure prediction methods
 DISSPred 82, 85
 Frag1D 81, 82, 85
 Jpred 22, 81–83
 PCI-SS 81, 82, 85
 PORTER 22, 81, 82, 84, 85
 PROTEUS 81, 82, 86, 92
 PSIPRED 81, 82, 84, 289
 RaptorX 81–83
 SABLE 81, 82, 87
 SCORPION 22, 81, 82, 84
 YASPIN 81, 82, 86
 YASSPP 81, 82, 86
- Secondary structure states 76, 77, 79, 81
- Self-assembly 189–191, 203, 204, 207, 238
- SH3-like topology 200, 210
- Simple super-secondary structure 16, 22, 24–31, 33, 40
- Sm fold 189, 192, 200, 203, 204, 210, 212
- Supersecondary structures (SSSs)
 β-barrel motif 24, 34, 264
 β-propellers motif 24, 33, 34
 CCPLUS database 78
 coiled-coil motif design 22, 24, 27, 30, 104, 264
 CCbuilder 2.0 30
 MPs 34

symmetry detection (SymD)

- program 24, 33, 34

TOPS database 78

Supersecondary structures (SSSs) detection

- CSI3.0 35

Supersecondary structures (SSSs) prediction

- βαβ motif 25, 32

BhairPred 25, 26, 88, 89

coiled-coil motif

- AAFreqCoil 25, 28

LOGICOIL 24, 25, 27, 28, 30

Multicoil2 24, 25, 27, 30, 89

RFCoil 25, 27, 28, 30

SCORER 2.0 25, 27, 28, 30

GYM 88–90

helix-turn-helix motif 75, 78

machine learning (ML) algorithm

- Extra Tree (ET) Classifier 106, 110

Gradient Boosting Classifier

- (GBC) 106, 110

K Nearest Neighbor (KNN) Classifier 106, 110

Logistic regression (LogReg) 108, 110

Random Decision Forest (RDF) 108, 110

MultiCoil2 24, 25, 27, 30, 89

Rossmann Fold 25, 32

strand-loop-helix-loop-strand (βαβ) 24

the stacking-based ML approach 103, 109

StackSSSPred 101–119

using predicted secondary structure 90–91, 117

Supersecondary structures (SSSs) visualization

HERA 35, 49, 62

PROMOTIF 35, 49, 62

Pro-origami 49, 198

PTGL 35

Symmetry detection (SymD) 24, 25,

33, 34, 197

T

- T-cell surface proteins 188
- Template-Based Modeling (TBM) approaches
 I-TASSER 18, 25
 ModBase 18, 25
 MODELLER 18
- TIM barrels ix, 33, 188, 221–233
- Toll-like receptor 8 189
- Translational symmetry 213