UNIVERSITY OF CALIFORNIA, SAN DIEGO

Structure-Preserving Rearrangements: Algorithms for Structural Comparison and Protein Analysis

A dissertation submitted in partial satisfaction of the
requirements for the degree
Doctor of Philosophy

in

Bioinformatics and Systems Biology

by

Spencer Edward Bliven

Committee in charge:

Professor Philip E. Bourne, Chair
Professor Milton H. Saier, Co-Chair
Professor Russell F. Doolittle
Professor Michael K. Gilson
Professor Adam Godzik

2015

The dissertation of Spencer Edward Bliven is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

_____

_____

_____

_____
Co-Chair

_____
Chair

University of California, San Diego

2015

DEDICATION

To my father, whose curiosity and enthusiasm for both nature and technology I wholeheartedly share.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SUPPLEMENTAL FILES

ACKNOWLEDGEMENTS

First and foremost, thanks to my wife Christine, who stuck with me through the difficulties of long-distance dating and bravely moved down to San Diego to support my doctoral dreams. You've always been there for me, putting up with the long workdays and the terrible science puns. Thanks for all the support, even at the end when you were working hard yourself on our own little developmental biology experiment.

Thanks to my Mom and Dad for encouraging my curiosity growing up and supporting my many years of education, and for their help in preparing my thesis. To my siblings Hunter and Hillary, for their love and support. And to my San Diego cousins, Ben Willis, Matt Hall, and Megan Jones, for making life more fun.

My advisor, Phil Bourne, has been an incredible mentor through this. I've always appreciated your friendly manner, which made your lab a great place to work. Thanks for giving me the freedom to work independently, and for supporting my project even as your own duties broadened from one lab to the campus to the nation.

Andreas Prlić has been a great mentor ever since my first research rotation with the Bourne lab. You are always the first one I want to share an exciting new idea with, and without your regular supervision and contributions this thesis would have never gotten off the ground. Thanks for all the rapid bug fixes and last minute proofreads for my papers and abstracts over the years, and for the help preparing for my defense.

A hearty thanks to my UCSD committee. To Milton Saier, for being my co-advisor despite my knowing next to nothing about transporters when I asked him. Russ Doolitle, for your heartfelt advice and for demonstrating clearly the joy of doing science. Mike Gilson, for your dependability and interest in my success. And to Adam Godzik, for asking the hard questions and sharing your practical advice.

Many thanks to all the other people at UCSD who have helped me: Peter Rose, Cole Christie, and the rest of the RCSB; Crystal Ku, Jian Wang, Nathan Mih, Roger

Chang, and the other students in the lab; Åke Vastermark, Liz Brunk, and my other collaborators; Jeremy Davis-Turak, Matt Schultz, Steve Federowicz, and all the other Bioinformatics students who travelled the PhD path along side me.

I had the privilege to be affiliated with several fine institutes. Guido Capitani has been a wonderful mentor at Paul Scherrer Institute. You have gone far beyond the expected support for a guest. Your team has been as pleasant and interesting a place as one could hope for. Thanks also for the diligent help editing and proofreading this thesis. Thanks to Jose Duarte, Kumaran Baskaran, Aleix Lafita, Chris Somody, Charleen Don, Sandra Markovic-MÏler, and the rest of the Laboratory for Biomolecular Research. Thanks also to David Landsman and the staff at NCBI for helping me navigate the NIH bureaucracy, and to Philippe Youkharibache for the stimulating talks about symmetry and structure.

A warm thanks to my coauthors for allowing me to include material from the following papers in this dissertation. Individual contributions are addressed in the acknowledgement sections of the relevant chapters.

- Chapter 2:

  A. Prlić, A. Yates, S. E. Bliven, P. W. Rose, J. Jacobsen, P. V. Troshin, M. Chapman, J. Gao, C. H. Koh, S. Foisy, R. Holland, G. Rimša, M. L. Heuer, H. Brandstätter-Müller, P. E. Bourne, and S. Willis. BioJava: an open-source framework for bioinformatics in 2012. *Bioinformatics*, 28 (20):2693–2695, Oct. 2012

- Chapter 3:

  A. Prlić, S. E. Bliven, P. W. Rose, W. F. Bluhm, C. Bizon, A. Godzik, and P. E. Bourne. Pre-calculated protein structure alignments at the RCSB PDB website. *Bioinformatics*, 26(23):2983–2985, Dec. 2010

- Chapter 5:

  S. E. Bliven and A. Prlić. Circular permutation in proteins. *PLoS Comput Biol*, 8(3):e1002445, Mar. 2012

- Chapter 6:

  S. E. Bliven, P. E. Bourne, and A. Prlić. Detection of circular permutations within protein structures using CE-CP. *Bioinformatics*, 31(8): 1316–1318, Apr. 2015

- Chapter 7:

  D. Myers-Turnbull, S. E. Bliven, P. W. Rose, Z. K. Aziz, P. Youkharibache, P. E. Bourne, and A. Prlić. Systematic Detection of Internal Symmetry in Proteins Using CE-Symm. *J Mol Biol*, 426(11):2255–2268, May 2014

Funding: My initial training was funded through the Bioinformatics and Systems Biology training grant from the National Institutes of Health, USA (National Institutes of Health grant T32GM8806). Work with the RCSB PDB was supported by the RCSB PDB grant from the National Science Foundation [grant numbers DBI 0829586 and 1338415] Finally, work since April 2014 has been supported through the Intramural Research Program of the National Center for Biotechnology Information, National Library of Medicine, National Institutes of Health. Computation was provided in part by the Open Science Grid.

VITA

2015	Doctor of Philosophy in Bioinformatics, University of California San Diego, La Jolla, CA
2014 - 2015	Guest Scientist, Paul Scherrer Institute, Villigen, Switzerland
2014 - 2015	Predoctoral Fellow, National Center for Biotechnology Information, National Library of Medicine, National Institutes of Health, Bethesda, MD
2009	Bachelor of Science in Computer Science and Biochemistry, University of Washington, Seattle, WA
2007-2008	Study Abroad, Eidgenössische Technische Hochschule, Zürich, Switzerland

PUBLICATIONS

A. Prlić, **S. E. Bliven**, P. W. Rose, W. F. Bluhm, C. Bizon, A. Godzik, and P. E. Bourne. Pre-calculated protein structure alignments at the RCSB PDB website. *Bioinformatics*, 26(23):2983–2985, Dec. 2010

S. E. Bliven and A. Prlić. Circular permutation in proteins. *PLoS Comput Biol*, 8(3):e1002445, Mar. 2012

A. Prlić, A. Yates, **S. E. Bliven**, P. W. Rose, J. Jacobsen, P. V. Troshin, M. Chapman, J. Gao, C. H. Koh, S. Foisy, R. Holland, G. Rimša, M. L. Heuer, H. Brandstätter-Müller, P. E. Bourne, and S. Willis. BioJava: an open-source framework for bioinformatics in 2012. *Bioinformatics*, 28(20):2693–2695, Oct. 2012

D. Myers-Turnbull, **S. E. Bliven**, P. W. Rose, Z. K. Aziz, P. Youkharibache, P. E. Bourne, and A. Prlić. Systematic Detection of Internal Symmetry in Proteins Using CE-Symm. *J Mol Biol*, 426(11):2255–2268, May 2014

K. Baskaran, J. M. Duarte, N. Biyani, **S. E. Bliven**, and G. Capitani. A PDB-wide, evolution-based assessment of protein–protein interfaces. *BMC Struct Biol*, 14(1):22, Oct. 2014

S. E. Bliven, P. E. Bourne, and A. Prlić. Detection of circular permutations within protein structures using CE-CP. *Bioinformatics*, 31(8):1316–1318, Apr. 2015

G. Capitani, J. M. Duarte, K. Baskaran, **S. E. Bliven**, J. C. Somody (2015) Understanding the Fabric of Protein Crystals: Computational Classification of Biological Interfaces and Crystal Contacts *Submitted to PLOS Comp Biol*. 2015.

A. Vastermark, A. Driker, D. Allohn, **S. Bliven**, J. Weng, X. Li, J. Wang and M. H. Saier, Jr. Refinement of the group translocation model for the L-ascorbate transporter of *E. coli*. *In preparation*. 2015.

E. Brunk, N. Mih, J. Monk, K. Chen, Z. Zhang, E. J. O'Brien, **S. Bliven**, R. L. Chang, P. E. Bourne, and B. O. Palsson. Comparative Systems Biology of the Structural Proteome. *In preparation*. 2015.

S. Markovic-Mueller, E. Stuttfeld, M. Asthana, T. Weinert, K. Kisko, **S. Bliven**, K. N. Goldie, G. Capitani and K. Ballmer-Hofer Structural and biochemical evidence for allosteric regulation in VEGFR-1. *In preparation*. 2015.

ABSTRACT OF THE DISSERTATION

Structure-Preserving Rearrangements: Algorithms for Structural Comparison and Protein Analysis

by

Spencer Edward Bliven

Doctor of Philosophy in Bioinformatics and Systems Biology

University of California, San Diego, 2015

Professor Philip E. Bourne, Chair
Professor Milton H. Saier, Co-Chair

Protein structure is fundamental to a deep understanding of how proteins function. Since structure is highly conserved, structural comparison can provide deep information about the evolution and function of protein families. The Protein Data Bank (PDB) continues to grow rapidly, providing copious opportunities for advancing our understanding of proteins through large-scale searches and structural comparisons. In this work I present several novel structural comparison methods for specific applications, as well as apply structure comparison tools systematically to better understand global properties of

protein fold space.

Circular permutation describes a relationship between two proteins where the N-terminal portion of one protein is related to the C-terminal portion of the other. Proteins that are related by a circular permutation generally share the same structure despite the rearrangement of their primary sequence. This non-sequential relationship makes them difficult for many structure alignment tools to detect. Combinatorial Extension for Circular Permutations (CE-CP) was developed to align proteins that may be related by a circular permutation. It is widely available due to its incorporation into the RCSB PDB website.

Symmetry and structural repeats are common in protein structures at many levels. The CE-Symm tool was developed in order to detect internal pseudosymmetry within individual polypeptide chains. Such internal symmetry can arise from duplication events, so aligning the individual symmetry units provides insights about conservation and evolution. In many cases, internal symmetry can be shown to be important for a number of functions, including ligand binding, allostery, folding, stability, and evolution.

Structural comparison tools were applied comprehensively across all PDB structures for systematic analysis. Pairwise structural comparisons of all proteins in the PDB have been computed using the Open Science Grid computing infrastructure, and are kept continually up-to-date with the release of new structures. These provide a network-based view of protein fold space. CE-Symm was also applied to systematically survey the PDB for internally symmetric proteins. It is able to detect symmetry in $\sim$20% of all protein families. Such PDB-wide analyses give insights into the complex evolution of protein folds.

Chapter 1

Introduction

One of the key problems in bioinformatics has always been detecting homologous genes and proteins (Waterman, 1995; Jones and Pevzner, 2004). With the rapid rise of available data and computational resources, increasingly sophisticated methods have been developed to detect homology across large evolutionary distances. For sequences, this has been accomplished by shifting from pairwise sequence alignment methods (Smith and Waterman, 1981; Needleman and Wunsch, 1970; Altschul et al., 1990) to more sensitive multiple sequence alignment methods (Altschul et al., 1997; Soding, 2005). Even more distant relationships can be determined by comparing protein structure, which tends to be more conserved than sequence (Illergård et al., 2009).

Pairwise structural comparisons are useful for highlighting similarities and differences in related structures, as well as visualizing the alignments for further structural analysis. The Combinatorial Extension (CE) (Shindyalov and Bourne, 1998; Jia et al., 2004) and FATCAT algorithms are available as part of the BioJava software library, as described in Chapter 2. FATCAT was used for systematic pairwise comparison across the Protein Data Bank (PDB), which involved significant computational effort (Chapter 3). These comparisons are updated automatically as new structures are released, and are

available to users on the RCSB PDB website (Berman et al., 2000; Rose et al., 2011). The availability of a comprehensive database of structural similarities and visualization tools enables users to easily analyze related structures and more distant homologues.

The comprehensive structural similarity information also provides a view of protein fold space. Understanding the set of all protein folds and the evolutionary pressures that shape which folds are observed in nature is an important problem in structural biology. The very concept of a protein fold implies a discrete classification of fold space, yet similarities between distant folds are abundant enough that many have argued that protein structures form a continuum, of which the known protein structures are but a sample (Sadreyev et al., 2009). Chapter 4 gives our analysis of protein fold space based on a graph representation. This incorporates some characteristics of both continuous and discrete interpretations of fold space.

One difficulty with identifying homology is dealing with the variety of scales at which nature reuses proteins. At a local scale, evolution can be well approximated by the familiar processes of insertion, deletion, and point mutation. However, at a larger scale many types of rearrangements can occur. Genes are duplicated, domains are rearranged and recombined, interfaces form and dissolve, creating new complexes. While standard structural comparison tools can accurately account for the local kind of mutation, identifying larger rearrangements is more difficult.

While gene mutations provide the mechanism for evolution, natural selection pressures are influenced by the expressed proteins and gene products. For proteins, this entails the full protein complex present under physiological conditions, as well as any binding sites for ligands or regulatory partners. The remaining chapters address several types of modifications that rearrange the underlying sequence, but that preserve the global structure. Chapter 5 describes the process of circular permutation in proteins, where the first portion of one protein is related to the second portion of another protein,

and vice versa. This type of rearrangement typically preserves the overall structure of the protein. Circular permutations can occur naturally (Ponting and Russell, 1995) and are also created artificially to study protein folding (Capraro et al., 2008; Viguera et al., 1996; Zhang and Schachman, 1996), increase thermostability (Topell et al., 1999), decrease proteolytic susceptibility (Whitehead et al., 2009), and otherwise manipulate protein function. Chapter 6 describes the CE-CP algorithm for aligning protein structures that are related by a circular permutation. Since several mechanisms by which circular permutations can arise are understood, detecting circularly permuted structures can shed light on the evolutionary history of protein families.

(a) (b) (c)

Figure 1.1: Members of the glyoxalase family with various repeat organizations, colored by chain. All proteins bind a metal cofactor and contain four structural repeats, but with different chain connectivity. (a) Glyoxalase I from *Clostridium acetobutylicum* [PDB:3HDP] (Nickel; Dimer). (b) Glyoxalase I from *E. coli* [PDB:1F9Z] (Nickel; Dimer). (c) 1,2-dihydroxy-naphthalene dioxygenase from *Pseudomonas* sp. strain C18 [PDB:2EHZ] (Iron; Octamer). No ion is observed at the lower beta sheet, but the overall RMSD is still 3.6 Å over nearly the whole protein.

Another structural feature that is discussed is that of structural repeats within individual polypeptide chains. These can come about naturally through duplication and fusion events, or can occur through convergent evolution resulting in similar substructures (Andrade et al., 2001; Abraham et al., 2009; Blaber et al., 2012). The most frequent type of repeat is internal pseudosymmetry. Internal symmetry is quite common, appearing in $\sim$24 % of protein families. A close relationship exists between internal symmetry within

individual chains and the quaternary symmetry common in biological assemblies. Many examples are known where internally symmetric proteins are thought to evolve from oligomeric complexes while preserving the overall structure of the biological assembly (Lee and Blaber, 2011; Blaber et al., 2012; Kelman and O'Donnell, 1995). Figure 1.1 shows one such example. The CE-Symm algorithm was developed to automatically detect internal symmetry in proteins (Chapter 7 and 8). Using CE-Symm, a systematic search for internally symmetric proteins was performed. The tool was also useful in characterizing the functional implications of symmetry in a number of cases, including quantifying symmetry around ligand-binding sites and the enrichment of symmetry in membrane proteins.

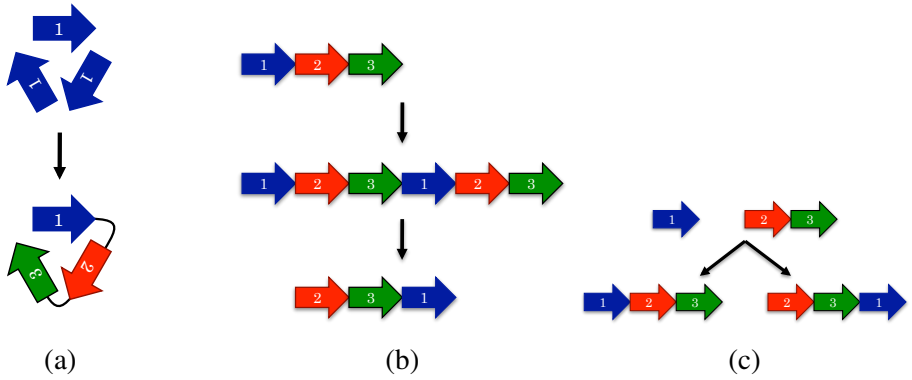


Figure 1.2: Evolution of symmetry and circular permutation. (a) Duplication and fusion mechanism for internal symmetry. (b) Permutation by duplication mechanism for circular permutation, involving a duplicated intermediate. (c) Fission and Fusion mechanism for circular permutation, where independent proteins fuse in two different orders.

Both circular permutations and internal symmetry can evolve via duplication and fusion events that significantly change the structure of the gene (Figure 1.2). However, these rearrangements do not entail correspondingly large changes in the structure of the overall biological assembly. Thus such events incur relatively little fitness cost, while opening up potentially useful new avenues for evolution. For instance, the fusion of a dimeric transcription factor, which would bind palindromic sequences, allows the

recognition of asymmetric binding sites, such as in TATA-binding protein (Juo et al., 1996). The novel algorithms developed here allow the detection and alignment of proteins with complex relationships that would typically be missed by algorithms which assume a sequential correspondence of structures. This gives a richer and more nuanced view of structural similarities, with the hope that this translates into a more accurate assessment of protein homology.

As a matter of principle, I am committed to open and accessible research. All research described here was published in open access journals. The tools and source code are made available under open source licenses as specified in subsequent chapters.

1.1 Bibliography

A.-L. Abraham, J. Pothier, and E. P. C. Rocha. Alternative to homo-oligomerisation: the creation of local symmetry in proteins by internal amplification. *J Mol Biol*, 394(3): 522–534, Dec. 2009.

S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman. Basic local alignment search tool. *J Mol Biol*, 215(3):403–410, Oct. 1990.

S. F. Altschul, T. L. Madden, A. A. Schäffer, J. Zhang, Z. Zhang, W. Miller, and D. J. Lipman. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res*, 25(17):3389–3402, Sept. 1997.

M. A. Andrade, C. Perez-Iratxeta, and C. P. Ponting. Protein Repeats: Structures, Functions, and Evolution. *J Struct Biol*, 134(2-3):117–131, May 2001.

H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne. The Protein Data Bank. *Nucleic Acids Res*, 28(1): 235–242, Jan. 2000.

M. Blaber, J. Lee, and L. Longo. Emergence of symmetric protein architecture from a simple peptide motif: evolutionary models. *Cell. Mol. Life Sci.*, 69:3999–4006, July 2012.

D. T. Capraro, M. Roy, J. N. Onuchic, and P. A. Jennings. Backtracking on the folding landscape of the beta-trefoil protein interleukin-1beta? 105(39):14844–14848, Sept. 2008.

K. Illergård, D. H. Ardell, and A. Elofsson. Structure is three to ten times more conserved than sequence-A study of structural response in protein cores. *Proteins*, 77(3):499–508, Nov. 2009.

Y. Jia, T. G. Dewey, I. N. Shindyalov, and P. E. Bourne. A new scoring function and associated statistical significance for structure alignment by CE. *J Comput Biol*, 11(5): 787–799, 2004.

N. C. Jones and P. Pevzner. An introduction to bioinformatics algorithms. 2004.

Z. S. Juo, T. K. Chiu, P. M. Leiberman, I. Baikalov, A. J. Berk, and R. E. Dickerson. How proteins recognize the TATA box. *J Mol Biol*, 261(2):239–254, Aug. 1996.

Z. Kelman and M. O'Donnell. Structural and functional similarities of prokaryotic and eukaryotic DNA polymerase sliding clamps. *Nucleic Acids Res*, 23(18):3613–3620, Sept. 1995.

J. Lee and M. Blaber. Experimental support for the evolution of symmetric protein architecture from a simple peptide motif. 108(1):126–130, Jan. 2011.

S. B. Needleman and C. D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J Mol Biol*, 48(3):443–453, Mar. 1970.

C. P. Ponting and R. B. Russell. Swaposins: circular permutations within genes encoding saposin homologues. *Trends Biochem Sci*, 20(5):179–180, May 1995.

P. W. Rose, B. Beran, C. Bi, W. F. Bluhm, D. Dimitropoulos, D. S. Goodsell, A. Prlić, M. Quesada, G. B. Quinn, J. D. Westbrook, J. Young, B. Yukich, C. Zardecki, H. M. Berman, and P. E. Bourne. The RCSB Protein Data Bank: redesigned web site and web services. *Nucleic Acids Res*, 39(Database issue):D392–401, Jan. 2011.

R. I. Sadreyev, B.-H. Kim, and N. V. Grishin. Discrete-continuous duality of protein structure space. *Curr Opin Struct Biol*, 19(3):321–328, June 2009.

I. N. Shindyalov and P. E. Bourne. Protein structure alignment by incremental combinatorial extension (CE) of the optimal path. *Protein Eng*, 11(9):739–747, Sept. 1998.

T. F. Smith and M. S. Waterman. Identification of common molecular subsequences. *J Mol Biol*, 147(1):195–197, Mar. 1981.

J. Soding. Protein homology detection by HMM-HMM comparison. *Bioinformatics*, 21 (7):951–960, Mar. 2005.

S. Topell, J. Hennecke, and R. Glockshuber. Circularly permuted variants of the green fluorescent protein. *FEBS Lett*, 457(2):283–289, Aug. 1999.

A. R. Viguera, L. Serrano, and M. Wilmanns. Different folding transition states may result in the same native structure. *Nat Struct Biol*, 3(10):874–880, Oct. 1996.

M. S. Waterman. *Introduction to Computational Biology*. Maps, Sequences and Genomes. CRC Press, June 1995.

T. A. Whitehead, L. M. Bergeron, and D. S. Clark. Tying up the loose ends: circular permutation decreases the proteolytic susceptibility of recombinant proteins. *Protein Eng Des Sel*, 22(10):607–613, Oct. 2009.

P. Zhang and H. K. Schachman. In vivo formation of allosteric aspartate transcarbamoylase containing circularly permuted catalytic polypeptide chains: implications for protein folding and assembly. *Protein Sci*, 5(7):1290–1300, July 1996.

Chapter 2

BioJava: an open-source framework for bioinformatics in 2012

2.1 Forward

BioJava is an open source library for computational biology (http://www.biojava.org) (Holland et al., 2008b; Prlić et al., 2012). It includes functionality for many common bioinformatics problems, including genomics, sequence alignment, structural biology, and phylogenetics. BioJava is supported by the Open Bioinformatics Foundation and has an active user community. Source code is available under the GNU Lesser General Public License (LGPL) and is available from https://github.com/biojava/biojava.

Over the course of my thesis I have contributed significantly to BioJava, particularly in the core algorithm and structural biology algorithms. Most of the algorithms developed for this thesis are included in the latest version of BioJava. Since BioJava 3 I have been an active maintainer for the project, and I am currently the second most frequent committer to the project out of a list of ~ 50 contributors (after project leader Andreas Prlić).

The remaining text of chapter 2 is a reprint of the material from:

A. Prlić, A. Yates, S. E. Bliven, P. W. Rose, J. Jacobsen, P. V. Troshin, M. Chapman, J. Gao, C. H. Koh, S. Foisy, R. Holland, G. Rimša, M. L.

Heuer, H. Brandstätter-Müller, P. E. Bourne, and S. Willis. BioJava: an open-source framework for bioinformatics in 2012. *Bioinformatics*, 28(20): 2693–2695, Oct. 2012

2.2 Abstract

Motivation: BioJava is an open-source project for processing of biological data in the Java programming language. We have recently released a new major version (V.3.0.3), which is a major update of the code base and that greatly extends its functionality.

Results: BioJava now consists of several independent modules that provide state of the art tools for protein structure comparison, pairwise and multiple sequence alignments, working with DNA and protein sequences, analysis of amino acid properties, detection of protein modifications, and prediction of disordered regions in proteins, as well as parsers for common file formats using a biologically meaningful data model.

Availability: BioJava is an open-source project distributed under the Lesser GPL (LGPL). BioJava can be downloaded from the BioJava website (http://www.biojava.org). BioJava requires Java 1.6 or higher.

Contact: andreas.prlic@gmail.com All inquiries should be directed to the BioJava mailing lists. Details are available at http://biojava.org/wiki/BioJava:MailingLists

2.3 Introduction

BioJava is an established open source project which is driven by an active developer community (Holland et al., 2008a). It provides a framework for processing of commonly used biological data and has seen contributions from more than 60 developers in the 12 years since its creation. The supported data ranges in scope from DNA and protein sequence information up to the level of 3D protein structures. BioJava provides various file parsers, data models and algorithms to facilitate working with the standard data formats and enable rapid application development and analysis.

The project is hosted by the Open Bioinformatics Foundation (OBF, http://www.open-bio.org), which provides the source code repository, bug tracking

database, and email mailing lists. It also supports the related BioPerl (Stajich et al., 2002), BioPython (Cock et al., 2009), BioRuby (Goto et al., 2010), and a number of other projects.

2.4 Methods

Over the last 2 years, large parts of the original code base have been rewritten. BioJava 3 is a clear departure from the version 1 series. It now consists of several independent modules built using Maven (http://maven.apache.org). The original code has been moved into a separate biojava-legacy project, which is still available for backwards compatibility. In the following, we describe several of the new modules and highlight some of the new features that are included in the latest version of BioJava.

2.4.1 Core Module

The core module provides classes to model nucleotide and amino acid sequences and their inherent relationships. Emphasis was placed on using Java classes and method names to describe sequences that would be familiar to the biologist and provide a concrete representation of the steps in going from a gene sequence to a protein sequence to the computer scientist.

BioJava 3 leverages recent innovations in Java. A sequence is defined as a generic interface, allowing the framework to build a collection of utilities, which can be applied to any sequence such as multiple ways of storing data. In order to improve the framework's usability to biologists, we also define specific classes for common types of sequences, such as DNA and proteins. One area that highlights this work is the translation engine, which allows the interconversion of DNA, RNA and amino acid sequences. The engine can handle details such as choosing the codon table, converting start codons to a methionine, trimming stop codons, specifying the reading frame and handling ambiguous sequences ('R' for purines, for example). Alternatively, the user can manually override

defaults for any of these.

The storage of sequences is designed to minimize memory usage for large collections using a 'proxy' storage concept. Various proxy implementations are provided that can store sequences in memory, fetch sequences on demand from a web service such as UniProt or read sequences from a FASTA file as needed. The latter two approaches save memory by not loading sequence data until it is referenced in the application. This concept can be extended to handle very large genomic datasets, such as NCBI GenBank or a proprietary database.

2.4.2 Protein Structure Modules

The protein structure modules provide tools for representing and manipulating 3D biomolecular structures, with the particular focus on protein structure comparison. It contains Java ports of the FATCAT algorithm (Ye and Godzik, 2003) for flexible and rigid body alignment, a version of the standard Combinatorial Extension (CE) algorithm (Shindyalov and Bourne, 1998) as well as a new version of CE that can detect circular permutations in proteins (Bliven and Prlić, 2012). These algorithms are used to provide the RCSB Protein Data Bank (PDB) (Rose et al., 2011) Protein Comparison Tool as well as systematic comparisons of all proteins in the PDB on a weekly basis (Prlić et al., 2010).

Parsers for PDB and mmCIF file formats (Bernstein et al., 1977; Fitzgerald et al., 2006) allow the loading of structure data into a reusable data model. Notably, this feature is used by the SIFTS project to map between UniProt sequences and PDB structures (Velankar et al., 2005). Information from the RCSB PDB can be dynamically fetched without the need to manually download data. For visualization, an interface to the 3D viewer Jmol (Hanson, 2010) (http://www.jmol.org) is provided. Work is underway for better interaction with the RCSB PDB viewers (Moreland et al., 2005).

2.4.3 Genome and Sequencing Modules

The genome module is focused on the creation of gene sequence objects from the core module by supporting the parsing of GTF files generated by GeneMark (Besemer and Borodovsky, 2005), GFF2 files generated by GeneID (Blanco and Abril, 2009) and GFF3 files generated by Glimmer (Kelley et al., 2011). The gene sequences can then be written out as a GFF3 format for importing into GMOD (Stein et al., 2002). A separate sequencing module provides memory efficient, low level and streaming I/O support for several common variants of the FASTQ file format from next generation sequencers (Cock et al., 2010).

2.4.4 Alignment Module

The alignment module supplies standard algorithms for sequence alignment and establishes a foundation to perform progressive multiple sequence alignments. For pairwise alignments, an implementation of the Needleman–Wunsch algorithm computes the optimal global alignment (Needleman and Wunsch, 1970) and the Smith–Waterman algorithm calculates local alignments (Smith and Waterman, 1981). In addition to these standard pairwise algorithms, the module includes the Guan–Uberbacher algorithm to perform global sequence alignment efficiently using only linear memory (Guan and Uberbacher, 1996). This routine also allows predefined anchors to be manually specified that will be included in the alignment produced. Any of the pairwise routines can also be used to perform progressive multiple sequence alignment. Both pairwise and multiple sequence alignments output to standard alignment formats for further processing or visualization.

2.4.5 ModFinder Module

The ModFinder module provides new methods to identify and classify protein modifications in protein 3D structures. More than 400 different types of protein modifications (phosphorylation, glycosylation, disulfide bonds metal chelation, etc.) were

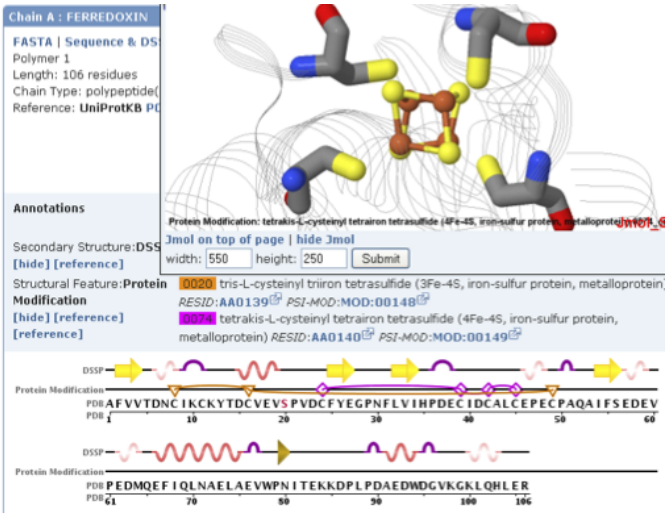


Figure 2.1: An example application utilizing the ModFinder module and the Protein Structure module. Protein modifications are mapped onto the sequence and structure of ferredoxin I (PDB ID 1GAO, (Chen et al., 2002)). Two possible iron-sulfur clusters are shown on the protein sequence (3Fe-4S (F3S): orange triangles/lines; 4Fe-4S (SF4): purple diamonds/lines). The 4Fe-4S cluster is displayed in the Jmol structure window above the sequence display.

collected and curated based on annotations in PSI-MOD (Montecchi-Palazzi et al., 2008), RESID (Garavelli, 2004), and PDB (Berman et al., 2000). The module provides an API for detecting protein modifications within protein structures. Figure 2.1 shows a web-based interface for displaying modifications, which was created using the ModFinder module. Future developments are planned to include additional protein modifications by integrating other resources such as UniProt (Farriol-Mathis et al., 2004).

2.4.6 Amino Acid Properties Module

The goal of the amino acid properties module is to provide a range of accurate physicochemical properties for proteins. The following peptide properties can currently be calculated: molecular weight, extinction coefficient, instability index, aliphatic index, grand average of hydropathy, isoelectric point and amino acid composition.

To aid proteomic studies, the module includes precise molecular weights for common isotopically labeled or post-translationally modified amino acids. Additional

types of PTMs can be defined using simple XML configuration files. This flexibility is especially valuable in situations where the exact mass of the peptide is important, such as mass spectrometry experiments.

2.4.7 Protein Disorder module

BioJava now includes a port of the Regional Order Neural Network (RONN) predictor (Yang et al., 2005) for predicting disordered regions of proteins. BioJava's implementation supports multiple threads, making it ∼3.2-times faster than the original C implementation on a modern quad-core machine.

The protein disorder module is distributed both as part of the BioJava library and as a standalone command line executable. The executable is optimized for use in automated analysis pipelines to predict disorder in multiple proteins. It can produce output optimized for either human readers or machine parsing.

2.4.8 Web Service Access Module

More and more bioinformatics tools are becoming accessible through the web. As such, BioJava now contains a web services module that allows bioinformatics services to be accessed using REST protocols. Currently, two services are implemented: NCBI Blast through the Blast URLAPI (previously known as QBlast) and the HMMER web service at http://hmmer.janelia.org (Finn et al., 2011).

2.5 Conclusion

The BioJava 3 library provides a powerful API for analyzing DNA, RNA and proteins. It contains state-of-the-art algorithms to perform various calculations and provides a flexible framework for rapid application development in bioinformatics. The library also provides lightweight interfaces to other projects that specialize in visualization tools.

The transition to Maven made managing external dependencies much easier, allowing the use of external libraries without overly complicating the installation procedure for users.

The BioJava project site provides an online cookbook that demonstrates the use of all modules through short recipes of common tasks. We are looking forward to extending the BioJava 3 library with more functionality over the coming years and welcome contributions of novel components by the community.

2.6 Acknowledgements

The authors thank everybody who contributed code, documentation or ideas, in particular A. Al-Hossary, R. Thornton, J. Warren, A. Draeger, G. Waldon and G. Barton. Each contribution is appreciated, although the total list of contributors is too long to be reproduced here. They also thank the Open Bioinformatics Foundation for project hosting.

This chapter was originally published as:

> A. Prlić, A. Yates, S. E. Bliven, P. W. Rose, J. Jacobsen, P. V. Troshin, M. Chapman, J. Gao, C. H. Koh, S. Foisy, R. Holland, G. Rimša, M. L. Heuer, H. Brandstätter-Müller, P. E. Bourne, and S. Willis. BioJava: an open-source framework for bioinformatics in 2012. *Bioinformatics*, 28(20): 2693–2695, Oct. 2012

Author Contributions: Andreas Prlić is the project leader for BioJava 3 and higher. He is the primary author of the protein structure module, including the jCE and jFATCAT ports of structure alignment algorithms. Of the features discussed in this chapter, I contributed bug fixes and features to the core and structure packages. Additional major features that I added to BioJava, such as CE-CP, are discussed in subsequent chapters. For a full record of the contributions of each coauthor, refer to http://github.org/biojava/biojava.

Funding: The RCSB PDB (NSF DBI 0829586 to A.P., P.W.R. and P.E.B.); Google Summer of Code in 2010 and 2011 (to J.G., M.C. and C.H.K.) and Scottish Universities Life Sciences Alliance (SULSA) (to P.T.).

2.7 Bibliography

H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne. The Protein Data Bank. *Nucleic Acids Research*, 28: 235–242, 2000.

F. C. Bernstein, T. F. Koetzle, G. J. Williams, E. F. Meyer, M. D. Brice, J. R. Rodgers, O. Kennard, T. Shimanouchi, and M. Tasumi. The Protein Data Bank: a computer-based archival file for macromolecular structures. *J Mol Biol*, 112(3):535–542, May 1977.

J. Besemer and M. Borodovsky. GeneMark: web software for gene finding in prokaryotes, eukaryotes and viruses. *Nucleic Acids Research*, 33(Web Server issue):W451–W454, 2005. URL http://www.ncbi.nlm.nih.gov/pubmed/15980510.

E. Blanco and J. F. Abril. Computational gene annotation in new genome assemblies using GeneID. *Methods In Molecular Biology*, 537(1):243–261, 2009. URL http://www.ncbi.nlm.nih.gov/pubmed/19378148.

S. Bliven and A. Prlić. Circular Permutation in Proteins. *PLoS Computational Biology*, 8 (3):e1002445, Mar. 2012. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1002445. URL http://dx.plos.org/10.1371/journal.pcbi.1002445.

K. Chen, Y.-S. Jung, C. A. Bonagura, G. J. Tilley, G. S. Prasad, V. Sridhar, F. A. Armstrong, C. D. Stout, and B. K. Burgess. Azotobacter vinelandii ferredoxin I: a sequence and structure comparison approach to alteration of [4Fe-4S]2+/+ reduction potential. *The Journal of Biological Chemistry*, 277(7):5603–5610, 2002. URL http://www.ncbi.nlm.nih.gov/pubmed/11704670.

P. J. A. Cock, T. Antao, J. T. Chang, B. A. Chapman, C. J. Cox, A. Dalke, I. Friedberg, T. Hamelryck, F. Kauff, B. Wilczynski, and M. J. L. De Hoon. Biopython: freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics*, 25(11):1422–1423, 2009. URL http://www.ncbi.nlm.nih.gov/pubmed/19304878.

P. J. A. Cock, C. J. Fields, N. Goto, M. L. Heuer, and P. M. Rice. The Sanger FASTQ file format for sequences with quality scores, and the Solexa/Illumina FASTQ variants. *Nucleic Acids Research*, 38(6):1767–1771, 2010. URL http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2847217&tool=pmcentrez&rendertype=abstract.

N. Farriol-Mathis, J. S. Garavelli, B. Boeckmann, S. Duvaud, E. Gasteiger, A. Gateau, A.-L. Veuthey, and A. Bairoch. Annotation of post-translational modifications in the Swiss-Prot knowledge base. *Proteomics*, 4(6):1537–1550, 2004. URL http://www.ncbi.nlm.nih.gov/pubmed/15174124.

R. D. Finn, J. Clements, and S. R. Eddy. HMMER web server: interactive sequence similarity searching. *Nucleic Acids Research*, 39(Web Server issue): W29–W37, 2011. URL http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3125773&tool=pmcentrez&rendertype=abstract.

P. M. D. Fitzgerald, J. D. Westbrook, P. E. Bourne, B. McMahon, K. D. Watenpaugh, and H. M. Berman. Macromolecular dictionary (mmCIF). In S. R. Hall and B. McMahon, editors, *International Tables for Crystallography*, pages 295–443. 2006.

J. S. Garavelli. The RESID Database of Protein Modifications as a resource and annotation tool. *Proteomics*, 4(6):1527–1533, 2004. URL http://www.ncbi.nlm.nih.gov/pubmed/15174122.

N. Goto, P. Prins, M. Nakao, R. Bonnal, J. Aerts, and T. Katayama. BioRuby: bioinformatics software for the Ruby programming language. *Bioinformatics*, 26(20):2617–2619, 2010. URL http://bioinformatics.oxfordjournals.org/content/26/20/2617.abstract.

X. Guan and E. C. Uberbacher. Alignments of DNA and protein sequences containing frameshift errors. *Computer applications in the biosciences CABIOS*, 12(1):31–40, 1996. URL http://www.ncbi.nlm.nih.gov/pubmed/8670617.

R. M. Hanson. Jmol – a paradigm shift in crystallographic visualization. *Journal of Applied Crystallography*, 43(5):1250–1260, 2010. ISSN 00218898. doi: 10.1107/S0021889810030256. URL http://scripts.iucr.org/cgi-bin/paper?S0021889810030256.

R. C. G. Holland, T. A. Down, M. Pocock, A. Prlić, D. Huen, K. James, S. Foisy, A. Dräger, A. Yates, M. Heuer, and M. J. Schreiber. BioJava: an open-source framework for bioinformatics. *Bioinformatics*, 24(18):2096–7, 2008a. ISSN 13674811. doi: 10.1093/bioinformatics/btn397. URL http://www.ncbi.nlm.nih.gov/pubmed/18689808.

R. C. G. Holland, T. A. Down, M. Pocock, A. Prlić, D. Huen, K. James, S. Foisy, A. Dräger, A. Yates, M. Heuer, and M. J. Schreiber. BioJava: an open-source framework for bioinformatics. *Bioinformatics*, 24(18):2096–2097, Sept. 2008b.

D. R. Kelley, B. Liu, A. L. Delcher, M. Pop, and S. L. Salzberg. Gene prediction with Glimmer for metagenomic sequences augmented by classification and clustering. *Nucleic Acids Research*, 40(1):1–12, 2011. ISSN 03051048. doi: 10.1093/nar/gkr1067. URL http://www.nar.oxfordjournals.org/cgi/doi/10.1093/nar/gkr1067.

L. Montecchi-Palazzi, R. Beavis, P.-A. Binz, R. J. Chalkley, J. Cottrell, D. Creasy, J. Shofstahl, S. L. Seymour, and J. S. Garavelli. The PSI-MOD community standard

for representation of protein modification data., 2008. URL http://www.ncbi.nlm.nih.gov/pubmed/18688235.

J. L. Moreland, A. Gramada, O. V. Buzko, Q. Zhang, and P. E. Bourne. The Molecular Biology Toolkit (MBT): a modular platform for developing molecular visualization applications. *BMC Bioinformatics*, 6(1):21, 2005. URL http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=548701&tool=pmcentrez&rendertype=abstract.

S. B. Needleman and C. D. Wunsch. A general method applicable to the search for similarities in the amino acid sequences of two proteins. *J. Mol. Biol.*, 48:443–453, 1970.

A. Prlić, S. Bliven, P. W. Rose, W. F. Bluhm, C. Bizon, A. Godzik, and P. E. Bourne. Pre-calculated protein structure alignments at the RCSB PDB website. *Bioinformatics*, 26(23):2983–2985, Dec. 2010. doi: 10.1093/bioinformatics/btq572. URL http://dx.doi.org/10.1093/bioinformatics/btq572.

A. Prlić, A. Yates, S. E. Bliven, P. W. Rose, J. Jacobsen, P. V. Troshin, M. Chapman, J. Gao, C. H. Koh, S. Foisy, R. Holland, G. Rimša, M. L. Heuer, H. Brandstätter-Müller, P. E. Bourne, and S. Willis. BioJava: an open-source framework for bioinformatics in 2012. *Bioinformatics*, 28(20):2693–2695, Oct. 2012.

P. W. Rose, B. Beran, C. Bi, W. F. Bluhm, D. Dimitropoulos, D. S. Goodsell, A. Prlic, M. Quesada, G. B. Quinn, J. D. Westbrook, J. Young, B. Yukich, C. Zardecki, H. M. Berman, and P. E. Bourne. The RCSB Protein Data Bank: redesigned web site and web services. *Nucleic Acids Res*, 39(Database issue):D392—-D401, Jan. 2011. doi: 10.1093/nar/gkq1021. URL http://dx.doi.org/10.1093/nar/gkq1021.

I. N. Shindyalov and P. E. Bourne. Protein structure alignment by incremental combinatorial extension {(CE)} of the optimal path. *Protein Eng.*, 11:739–747, 1998. URL http://peds.oxfordjournals.org/cgi/content/short/11/9/739.

T. F. Smith and M. S. Waterman. Identification of common molecular subsequences. *J. Mol. Biol.*, 147:195–197, 1981.

J. E. Stajich, D. Block, K. Boulez, S. E. Brenner, S. A. Chervitz, C. Dagdigian, G. Fuellen, J. G. Gilbert, I. Korf, H. Lapp, H. Lehvaslaiho, C. Matsalla, C. J. Mungall, B. I. Osborne, M. R. Pocock, P. Schattner, M. Senger, L. D. Stein, E. Stupka, M. D. Wilkinson, and E. Birney. The Bioperl toolkit: Perl modules for the life sciences. *Genome Research*, 12(10):1611–1618, 2002. ISSN 10889051. doi: 10.1101/gr.361602.1. URL http://dx.doi.org/10.1101/gr.361602.

L. D. Stein, C. Mungall, S. Shu, M. Caudy, M. Mangone, A. Day, E. Nickerson, J. E. Stajich, T. W. Harris, A. Arva, and S. Lewis. The Generic Genome Browser: A Building Block for a Model Organism System Database. *Genome Research*, 12(10):1599–1610, 2002. URL http://www.ncbi.nlm.nih.gov/pubmed/12368253.

S. Velankar, P. McNeil, V. Mittard-Runte, A. Suarez, D. Barrell, R. Apweiler, and K. Henrick. E-MSD: an integrated data resource for bioinformatics. *Nucleic Acids Res*, 33(Database issue):D262–5, Jan. 2005.

Z. R. Yang, R. Thomson, P. McNeil, and R. M. Esnouf. RONN: the bio-basis function neural network technique applied to the detection of natively disordered regions in proteins. *Bioinformatics*, 21(16):3369–3376, 2005. URL http://www.ncbi.nlm.nih.gov/pubmed/15947016.

Y. Ye and A. Godzik. Flexible structure alignment by chaining aligned fragment pairs allowing twists. *Bioinformatics*, 19 Suppl 2:II246—-II255, Oct. 2003.

Chapter 3

Precalculated Protein Structure Alignments at the RCSB PDB website

3.1 Forward

The Protein Data Bank (PDB) is the worldwide repository for all atomic-resolution protein structures. The RCSB PDB is one of three access point to this data and is visited by hundreds of thousands of unique visitors per year (Berman et al., 2000; Rose et al., 2011, 2012).

The RCSB website includes several tools for comparing structures. This includes access to BioJava's CE and FATCAT implementations directly from the website, as well as several external services for pairwise structure and sequence comparison.

To facilitate rapid comparison of protein structures, pairwise structural comparisons were performed for a non-redundant set of all proteins in the PDB. This comparison was originally performed between protein chains, as described below (Prlić et al., 2010).

After the initial publication, the comparison was re-run using structural domains. Each protein was split into domains using the SCOP classification (Murzin et al., 1995; Andreeva et al., 2008b). For newer structures or where SCOP classifications were otherwise unavailable, domains were assigned automatically using the Protein Domain

Parser (Alexandrov and Shindyalov, 2003). This resulted in all-vs-all comparisons between $\sim$19 500 representative domains.

The domain comparison data has been updated weekly as part of the normal RCSB release procedure for new structures. This allows a list of all structurally similar domains to be made available for each structure. As of June 2015, the database contains 27 322 chain representatives and 41 838 domains.

The remaining text of chapter 3 is a reprint of material from:

> A. Prlić, S. E. Bliven, P. W. Rose, W. F. Bluhm, C. Bizon, A. Godzik, and P. E. Bourne. Pre-calculated protein structure alignments at the RCSB PDB website. *Bioinformatics*, 26(23):2983–2985, Dec. 2010

3.2 Abstract

Summary: With the continuous growth of the RCSB Protein Data Bank (PDB), providing an up-to-date systematic structure comparison of all protein structures poses an ever growing challenge. Here, we present a comparison tool for calculating both 1D protein sequence and 3D protein structure alignments. This tool supports various applications at the RCSB PDB website. First, a structure alignment web service calculates pairwise alignments. Second, a stand-alone application runs alignments locally and visualizes the results. Third, pre-calculated 3D structure comparisons for the whole PDB are provided and updated on a weekly basis. These three applications allow users to discover novel relationships between proteins available either at the RCSB PDB or provided by the user.

Availability and Implementation: A web user interface is available at http://www.rcsb.org/pdb/workbench/workbench.do. The source code is available under the LGPL license from http://www.biojava.org. A source bundle, prepared for local execution, is available from http://source.rcsb.org.

3.3 Introduction

At its core, the *RCSB PDB Protein Comparison Tool* contains a new implementation of the two structure alignment algorithms Combinatorial Extension (CE) (Shindyalov and Bourne, 1998) and FATCAT (both rigid body and flexible versions) (Ye and Godzik, 2003).

Both the CE and FATCAT algorithms detect aligned fragment pairs (AFPs) during the alignment process. These AFPs are based on similarities in local geometry. There is a difference in how initial AFPs are combined in order to calculate an optimal alignment. CE applies the process of 'Combinatorial Extension' to find possible continuous alignment paths leading to an optimal alignment. The resulting alignment is a 'rigid-body' based alignment. In contrast to this, FATCAT allows the introduction of 'twists' into the

alignment with the consequence that different regions of a protein structures can undergo different geometric transformations. This is required in order to be able to deal with protein flexibility.

A protein that undergoes significant domain re-arrangement during iron binding is transferrin. It consists of two domains that can move relative to each other. FATCAT in its flexible mode can easily detect an alignment between the apo and holo forms that covers 95% of both protein chains. (e.g. PDB ID 1IEJ chain A and PDB ID 1BTJ chain A). However, using rigid body superposition only a partial alignment is possible (both CE and FATCAT using the rigid mode only). One drawback of the flexible mode is that if distantly related proteins are being aligned, sometimes twists between unrelated regions can be introduced (e.g. alignment of PDB ID 1CDG chain A and PDB ID 1TIM chain A), in which case it is better to run the alignments in rigid mode.

A limitation of both CE and FATCAT in their original versions is that they compute sequence order-dependent alignments. A number of difficult to detect relationships between proteins have been published, some of which require sequence order independence for a correct alignment (Andreeva et al., 2006). An algorithm that can detect such order-independent alignments is Triangle Match (Bachar et al., 1993; Nussinov and Wolfson, 1991). Dali in its early versions also could detect permuted proteins; however, this feature seems to have been lost in its recent implementations, (Holm and Sander, 1993). We have recently improved CE to be able to detect circularly permuted alignments (Chapter 6). This implementation is available as an option of the RCSB Protein Comparison Tool.

3.4 Approach

The CE and FATCAT algorithms have been re-implemented in the Java programming language, which is indicated by a lower-case j in front of the new names, jCE and jFATCAT. Several components were added to these implementations.

First, the alignment algorithms were integrated into the RCSB PDB website

(Berman et al., 2000) to provide a novel structure alignment service. Second, a stand-alone application can be run using Java Web Start technology. Third, a client-server architecture was developed for calculating large-scale comparisons using compute clouds. Finally, a software bundle is provided that allows local installation of the tool and to run custom comparisons. We describe some of these components in more detail.

Pairwise Sequence and Structure Alignment

The comparison tool allows pairwise comparison of protein sequences and 3D structures. For sequence comparison the Smith–Waterman (Smith and Waterman, 1981), Needleman–Wunsch (Needleman and Wunsch, 1970), and blast2seq (Tatusova and Madden, 1999) algorithms are provided. Support for structure comparisons includes the new implementations of CE and FATCAT and links to some of the most prominent external protein structure alignment services: the original FATCAT server (Ye and Godzik, 2004), Mammoth (Ortiz et al., 2002), TM-align (Zhang and Skolnick, 2005), and Topmatch (Sippl and Wiederstein, 2008; Sippl, 2008). Other available structure alignment software can be found at Wikipedia http://en.wikipedia.org/wiki/Structural_alignment_software.

All alignments that are run using jCE and jFATCAT are calculated server-side on the fly and cached for future retrieval using XML files. If the alignment is requested again later, it can be instantly returned by reading the XML file. A web user interface provides access to the alignment results. Alternatively, a Java Web Start client application, based on Jmol (http://www.jmol.org, 2010), and BioJava (Holland et al., 2008), provides a novel 3D visualization tool that allows the investigation of sequence–structure relationships between two aligned proteins. See Figure 3.1 for an example alignment.

Systematic structure alignments across the PDB

Sequence database searches are a frequently used tool to identify closely related proteins within a database. However, with decreasing sequence similarity relationships

between proteins become harder to detect. In order to enable identification of relationships across the PDB, even if sequence similarity is low, we are providing systematic and precomputed alignments.

The procedure providing pre-computed structure comparisons across the PDB is split into two steps.

First, the goal is to reduce the complexity of the problem by identifying representative protein chains for clusters of related proteins. BLASTClust (Altschul and et al, 2004) is used to cluster all protein chains by sequence similarity. We require 90% overlap between all sequences in a cluster. Therefore, a shorter fragment (e.g. a single domain) of a longer sequence (e.g. a multi-domain protein) will usually not be in the same cluster as the whole sequence. Within clusters, sequences are ranked by experimental method, resolution and release date. While the RCSB PDB website provides sequence comparisons for various levels of sequence identity within a cluster, structural comparisons are only provided based on clusters with 40% sequence identity; currently approximately 16 000 representative protein chains.

Second, the rigid version of jFATCAT is used to calculate all-against-all 3D structure comparisons across all representative protein chains. This requires a significant amount of CPU time. Specifically, a client-server architecture has been developed that allows the user to easily run a large number of jobs in parallel (for details see http://www.renci.org/publications/techreports/TR-09-03.pdf). A total of 122 million alignments were calculated on the Open Science Grid, taking approximately 102 000 CPU hours. Another 18 million alignments were calculated on the San Diego Supercomputing Center (SDSC) Triton Cluster and local RCSB PDB servers. The alignment results were stored in $\sim$1 terabyte of XML files.

Weekly Updates

Incremental updates to the all-against-all comparisons are run weekly using in-house RCSB PDB servers at the same time the PDB itself is updated. Every week new sequence clusters are calculated and missing alignments for newly added representative

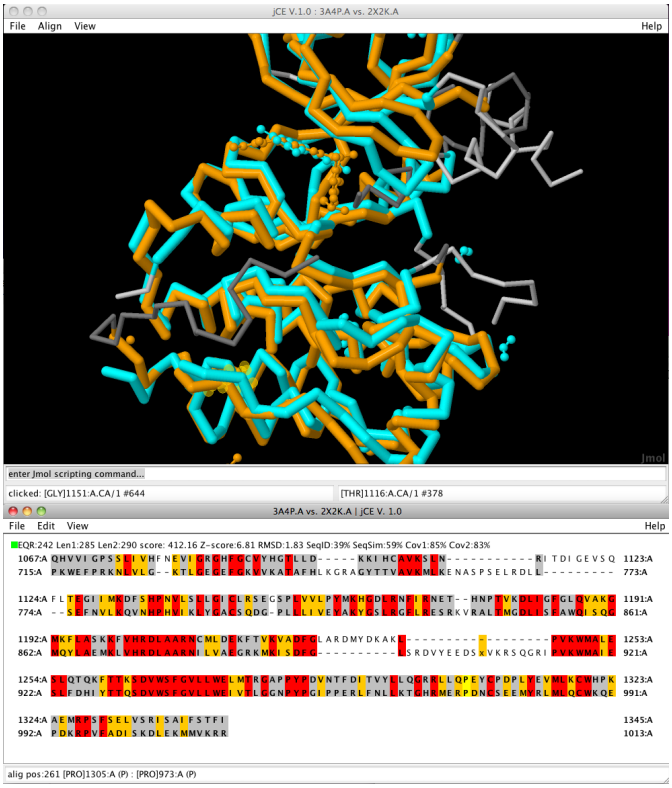


Figure 3.1: A new user interface for jCE and jFATCAT structure alignments allows the investigation of sequence and 3D structure relationships. Here, the alignment of two kinases, the Hepatocyte growth factor receptor PDB ID 3A4P and the Proto-Oncogene Tyrosine-Protein Kinase Receptor RET PDB ID 2X2K. If the structures contain ligands they are also superimposed and displayed. The coloring for the sequence representation of the structure alignment represents the sequence conservation: red: identical residues, orange: similar and grey: structurally equivalent, but sequence mismatch.

chains are calculated. At present, an average weekly update requires the calculation of about 1 million structure alignments.

3.5 Discussion

The RCSB PDB website now provides a state of the art protein comparison tool that can be used as a web service and for local access. Furthermore, pre-calculated all-against-all comparisons of sequences and 3D protein structures, respectively, are provided on a weekly basis. Currently, the calculations are done on a whole-chain basis. We are working on another set of comparisons using domain-based comparisons of all

chains with an enhanced version of jCE that includes handling of circular permutations. An alternative is to use TOPS++FATCAT (Veeramalai et al., 2008), which provides a 10-fold speed up as compared to FATCAT. The domain assignment problem is non-trivial, and for newly released protein structures results are not immediately available from classifications like SCOP (Andreeva et al., 2008a), or CATH (Cuff et al., 2009). Hence, we are investigating whether consensus based approaches like pDomains (Alden et al., 2010), can guide which domain assignments to use for the automated calculations.

3.6 Acknowledgements

The text of this chapter was original published as an open access article as:

A. Prlić, S. E. Bliven, P. W. Rose, W. F. Bluhm, C. Bizon, A. Godzik, and P. E. Bourne. Pre-calculated protein structure alignments at the RCSB PDB website. *Bioinformatics*, 26(23):2983–2985, Dec. 2010

Computational time was provided by the Open Science Grid and supported by the OSG Engagement group. The RCSB team maintains weekly updates to the structural comparisons.

Author Contributions: Andreas Prlić and Chris Bizon implemented the server/client architecture for distributing alignments (Bizon and Prlić, 2009). Andreas Prlić and I managed the clients and monitored the OSG jobs during the three month computation and analyzed the final results. Adam Godzik created the FATCAT algorithm. Peter Rose, Andreas Prlić, and other RCSB members coordinated the incorporation of the results into the PDB website and continue to support the weekly update pipeline. Philip Bourne provided advice and support throughout the project. All authors contributed to the manuscript.

Funding: The RCSB PDB is managed by two members of the RCSB: Rutgers and UCSD, and is funded by National Science Foundation (NSF), National Institute of

General Medical Sciences, Department of Energy (DOE), National Library of Medicine, National Cancer Institute, National Institute of Neurological Disorders and Stroke and National Institute of Diabetes and Digestive and Kidney Diseases. The RCSB PDB is a member of the wwPDB. This work was supported by the RCSB PDB grant NSF DBI 0829586. Computation provided in part by the Open Science Grid funded by NSF and DOE, supported by OSG Engagement under NSF award number 0753335.

3.7 Bibliography

K. Alden, S. Veretnik, and P. E. Bourne. dconsensus: a tool for displaying domain assignments by multiple structure-based algorithms and for construction of a consensus assignment. *BMC Bioinformatics*, 11:310, 2010. doi: 10.1186/1471-2105-11-310. URL http://dx.doi.org/10.1186/1471-2105-11-310.

N. Alexandrov and I. Shindyalov. PDP: protein domain parser. *Bioinformatics*, 19(3): 429–430, Feb. 2003.

Altschul and et al. BLASTClust, 2004. URL http://www.ncbi.nlm.nih.gov/Web/Newsltr/Spring04/blastlab.html.

A. Andreeva, A. Prlic, T. J. P. Hubbard, and A. G. Murzin. SISYPHUS–structural alignments for proteins with non-trivial relationships. *Nucleic Acids Res*, 0:0, Oct 2006. doi: 10.1093/nar/gkl746. URL http://dx.doi.org/10.1093/nar/gkl746.

A. Andreeva, D. Howorth, J.-M. Chandonia, S. E. Brenner, T. J. P. Hubbard, C. Chothia, and A. G. Murzin. Data growth and its impact on the SCOP database: new developments. *Nucleic Acids Res*, 36(Database issue):D419–D425, Jan 2008a. doi: 10.1093/nar/gkm993. URL http://dx.doi.org/10.1093/nar/gkm993.

A. Andreeva, D. Howorth, J.-M. Chandonia, S. E. Brenner, T. J. P. Hubbard, C. Chothia, and A. G. Murzin. Data growth and its impact on the SCOP database: new developments. *Nucleic Acids Res*, 36(Database issue):D419–25, Jan. 2008b.

O. Bachar, D. Fischer, R. Nussinov, and H. Wolfson. A computer vision based technique for 3-d sequence-independent structural comparison of proteins. *Protein Eng*, 6(3): 279–288, Apr 1993.

H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne. The Protein Data Bank. *Nucleic Acids Res*, 28(1): 235–242, Jan. 2000.

C. Bizon and A. Prlić. Calculating All Pairwise Similarities from the RCSB Protein Data Bank: Client/Server Work Distribution on the Open Science Grid. *RENCI Technical Report Series*, pages 1–19, Dec. 2009.

A. Cuff, O. C. Redfern, L. Greene, I. Sillitoe, T. Lewis, M. Dibley, A. Reid, F. Pearl, T. Dallman, A. Todd, R. Garratt, J. Thornton, and C. Orengo. The CATH hierarchy revisited-structural divergence in domain superfamilies and the continuity of fold space. *Structure*, 17(8):1051–1062, Aug 2009. doi: 10.1016/j.str.2009.06.015. URL http://dx.doi.org/10.1016/j.str.2009.06.015.

R. C. G. Holland, T. A. Down, M. Pocock, A. . A. Prlić, D. Huen, K. James, S. Foisy, A. Dräger, A. Yates, M. Heuer, and M. J. Schreiber. BioJava: an open-source framework for bioinformatics. *Bioinformatics*, 24(18):2096–2097, Sep 2008. doi: 10.1093/bioinformatics/btn397. URL http://dx.doi.org/10.1093/bioinformatics/btn397.

L. Holm and C. Sander. Protein structure comparison by alignment of distance matrices. *J.Mol.Biol.*, 233:123 – 138, 1993.

http://www.jmol.org. Jmol: an open-source Java viewer for chemical structures in 3D., 2010. URL http://www.jmol.org.

A. G. Murzin, S. E. Brenner, T. Hubbard, and C. Chothia. SCOP: a structural classification of proteins database for the investigation of sequences and structures. *J Mol Biol*, 247 (4):536–540, Apr. 1995.

S. B. Needleman and C. D. Wunsch. A general method applicable to the search for similarities in the amino acid sequences of two proteins. *J. Mol. Biol.*, 48:443–453, 1970.

R. Nussinov and H. J. Wolfson. Efficient detection of three-dimensional structural motifs in biological macromolecules by computer vision techniques. *Proc Natl Acad Sci U S A*, 88(23):10495–10499, Dec 1991.

A. R. Ortiz, C. E. M. Strauss, and O. Olmea. MAMMOTH (matching molecular models obtained from theory): an automated method for model comparison. *Protein Sci*, 11 (11):2606–2621, Nov 2002. doi: 10.1110/ps.0215902. URL http://dx.doi.org/10.1110/ps.0215902.

A. Prlić, S. E. Bliven, P. W. Rose, W. F. Bluhm, C. Bizon, A. Godzik, and P. E. Bourne. Pre-calculated protein structure alignments at the RCSB PDB website. *Bioinformatics*, 26(23):2983–2985, Dec. 2010.

P. W. Rose, B. Beran, C. Bi, W. F. Bluhm, D. Dimitropoulos, D. S. Goodsell, A. Prlić, M. Quesada, G. B. Quinn, J. D. Westbrook, J. Young, B. Yukich, C. Zardecki, H. M. Berman, and P. E. Bourne. The RCSB Protein Data Bank: redesigned web site and web services. *Nucleic Acids Res*, 39(Database issue):D392–401, Jan. 2011.

P. W. Rose, C. Bi, W. F. Bluhm, C. H. Christie, D. Dimitropoulos, S. Dutta, R. K. Green, D. S. Goodsell, A. Prlić, M. Quesada, G. B. Quinn, A. G. Ramos, J. D. Westbrook, J. Young, C. Zardecki, H. M. Berman, and P. E. Bourne. The RCSB Protein Data Bank: new resources for research and education. *Nucleic Acids Res*, 41(D1):D475–D482, Dec. 2012.

I. N. Shindyalov and P. E. Bourne. Protein structure alignment by incremental combinatorial extension (CE) of the optimal path. *Protein Eng.*, 11:739–747, 1998. URL http://peds.oxfordjournals.org/cgi/content/short/11/9/739.

M. J. Sippl. On distance and similarity in fold space. *Bioinformatics*, 24(6):872–873, Mar 2008. doi: 10.1093/bioinformatics/btn040. URL http://dx.doi.org/10.1093/bioinformatics/btn040.

M. J. Sippl and M. Wiederstein. A note on difficult structure alignment problems. *Bioinformatics*, 24:426–427, Jan 2008. doi: 10.1093/bioinformatics/btm622. URL http://dx.doi.org/10.1093/bioinformatics/btm622.

T. F. Smith and M. S. Waterman. Identification of common molecular subsequences. *J. Mol. Biol.*, 147:195–197, 1981.

T. A. Tatusova and T. L. Madden. BLAST 2 Sequences, a new tool for comparing protein and nucleotide sequences. *FEMS Microbiol Lett*, 174(2):247–250, May 1999.

M. Veeramalai, Y. Ye, and A. Godzik. TOPS++FATCAT: fast flexible structural alignment using constraints derived from TOPS+ Strings Model. *BMC Bioinformatics*, 9:358, 2008. doi: 10.1186/1471-2105-9-358. URL http://dx.doi.org/10.1186/1471-2105-9-358.

Y. Ye and A. Godzik. Flexible structure alignment by chaining aligned fragment pairs allowing twists. *Bioinformatics*, 19 Suppl 2:II246–II255, Oct 2003.

Y. Ye and A. Godzik. FATCAT: a web server for flexible structure comparison and structure similarity searching. *Nucleic Acids Res*, 32(Web Server issue):W582–W585, Jul 2004. doi: 10.1093/nar/gkh430. URL http://dx.doi.org/10.1093/nar/gkh430.

Y. Zhang and J. Skolnick. TM-align: a protein structure alignment algorithm based on the TM-score. *Nucleic Acids Res*, 33(7):2302–2309, 2005. doi: 10.1093/nar/gki524. URL http://dx.doi.org/10.1093/nar/gki524.

Chapter 4

Structural Comparison Networks

4.1 Introduction

The domain-based PDB-wide structural comparison resulted in a database of $\sim$190 million structural similarity scores. In addition to allowing users to quickly access structurally similar structures on the RCSB website, this database also provides an overview of known fold space.

The question of the nature of protein fold space has captured the attention of numerous structural and computational biologists. The number of possible protein sequences is so large as to be practically infinite, yet proteins with completely different primary sequences may fold into nearly identical structures (Sippl, 2009). Since the structure of a protein is essential to performing its function, understanding the nature of what structures are possible and how evolution has sampled the space of possible structures can have far reaching consequences.

A number of questions regarding the nature of protein fold space remain open. One of the more controversial questions is whether fold space is composed primarily of discrete protein folds, or whether folds are connected by a continuum of possible but unobserved folds (Sadreyev et al., 2009; Skolnick et al., 2009; Shindyalov and Bourne, 2000; Orengo et al., 1997; Rost, 2002). This question has practical implications on the

designability of proteins, the utility of structural genomics initiatives, and the design of structure classification methods.

With the proliferation of protein structures, several schemes for classifying proteins into discrete categories emerged. Notable protein classifications include SCOP (Murzin et al., 1995) and CATH (Orengo et al., 1997), DALI (Holm and Sander, 1993), and the recent ECOD (Cheng et al., 2014). Such classifications are undeniably useful as a description of the possible folds observed in nature. However, numerous examples exist of proteins with clear structural similarity, yet classified as discrete folds by these methods. For example, Grishin (2001) describes a sequence of structurally similar proteins leading from an all-β to an all-α protein. Such observations led to the view of fold space as a continuum. In this view, protein classifications are more like clusters of closely related structures. Some structures may lie near the edge of multiple clusters, incorporating structural features of each. Efforts have been made to formalize the notion of continuous fold space by defining rigorous distance functions for pairwise comparisons and computing all-against-all pairwise comparisons of known proteins (Sippl, 2008; Shindyalov and Bourne, 2000; Marsden and Abagyan, 2004). Multidimensional scaling can then be used to embed protein folds in a Euclidean space (Orengo et al., 1993; Holm and Sander, 1996; Hou et al., 2003). The problem with such approaches is that they are generally able to distinguish protein classes, but cannot capture the finer classifications of fold and superfamily. Thus, they have limited utility in predicting evolutionary relationships or functional characteristics.

It now seems that neither the discrete nor the continuous views of protein fold space can fully explain the relationships between protein folds (Sadreyev et al., 2009). Hybrid methods for fold space classification have been developed, such as SCOP2, which attempts to classify proteins not using a hierarchical tree, but with an acyclic graph including many types of relationships between folds (Andreeva et al., 2014). An alternative approach is to treat protein similarities as a graph of pairwise similarity relationships. Friedberg and Godzik (2005) applied this approach to identify shared structural fragments.

More recently, Nepomnyachiy et al. (2014) used a structural similarity network to analyze structural similarity at a domain level. Their results are remarkably similar to those described below, which were presented as a poster at the Intelligent Systems for Molecular Biology Conference in 2011 (Bliven et al., 2011).

4.2 Methods

The structural similarity network consists of nodes representing non-redundant structures, connected by an edge where significant structural similarity exists. Prior to the PDB-wide structural comparison, protein chains were clustered with BLASTClust (Altschul et al., 2004, 1990). The chain-based structure comparison was created using an early 2010 version of the PDB, while the domain-based comparison used a 2011 version. Chains were then optionally split into domains using either SCOP or the Protein Domain Parser (Alexandrov and Shindyalov, 2003).

Following the all-vs-all computation with jFATCAT (Chapter 3), comparisons were retained as significant if they had p-value ≤ 0.001, contained over 25 aligned residues, and covered over 50% of both aligned chains or domains.

Since proteins with $< 40\%$ sequence identity can nonetheless have very similar structures, the resulting network was still highly redundant at a structural level. To reduce this redundancy, agglomerative hierarchical clustering was performed to iteratively merge proteins with the highest degree of structural similarity. Changing the clustering threshold changes the degree of structural diversity represented in the graph, with higher thresholds being analogous to superfamilies and lower threshold clustering together whole folds.

This process results in an undirected graph consisting of the protein structural clusters and weighted edges indicating the maximum p-value between members of each cluster. All edge weights fall between the significance threshold (0.001) and the clustering threshold, which varied between 10^{-6} and 10^{-10} in our analyses.

The graph was visualized using Cytoscape (Shannon, 2003; Smoot et al., 2011). A number of properties were mapped onto the clusters for visualization and analysis

purposes. SCOP version 1.75 classifications were mapped to the representative proteins. In cases where multiple SCOP superfamilies mapped to the cluster, the majority annotation was used. The Transporter Classification Database (TCDB) provides a functional classification for many membrane transporter proteins (Saier et al., 2006, 2009, 2014). TCDB classifications were mapped onto clusters, with each cluster annotated with the union of all member annotations.

4.3 Results

The full chain-based network is presented in Figure 4.1. For stringent clustering thresholds, most proteins fall into a single large connected component, with only a few disconnected "orphan" folds. Reducing the significance threshold removes less significant edges and causes this component to break up into smaller disjoint networks. Thus, fold space can be considered as either mostly continuous or mostly discrete depending on the significance threshold considered.

The clustering threshold controls the granularity of the network. At extremely low thresholds, only identical proteins are clustered together, whereas at higher thresholds families and folds begin to be clustered. Thus, the clustering threshold can be thought of analogously to the various levels of hierarchical classifications. It was found that thresholds of $p < 10^{-6}$, $p < 10^{-7}$, and $p < 10^{-10}$ correspond roughly to SCOP class, fold, and superfamily levels (Figure 4.2). At these thresholds, around 90% of edges link clusters with the same SCOP classification, with the remaining 10% of edges representing structural similarities between discrete SCOP groupings.

The relationship between structure and function can be investigated by mapping various functional attributes onto the structural similarity graph. As an example of this, TCDB annotations were mapped onto the domain-based network to analyze structural conservation of transporter proteins. The subgraph of the domain-based network corresponding to proteins annotated as transporters is shown in Figure 4.3. A relatively stringent significance cutoff of TM-Score ≥ 0.5 was used to identify close structural

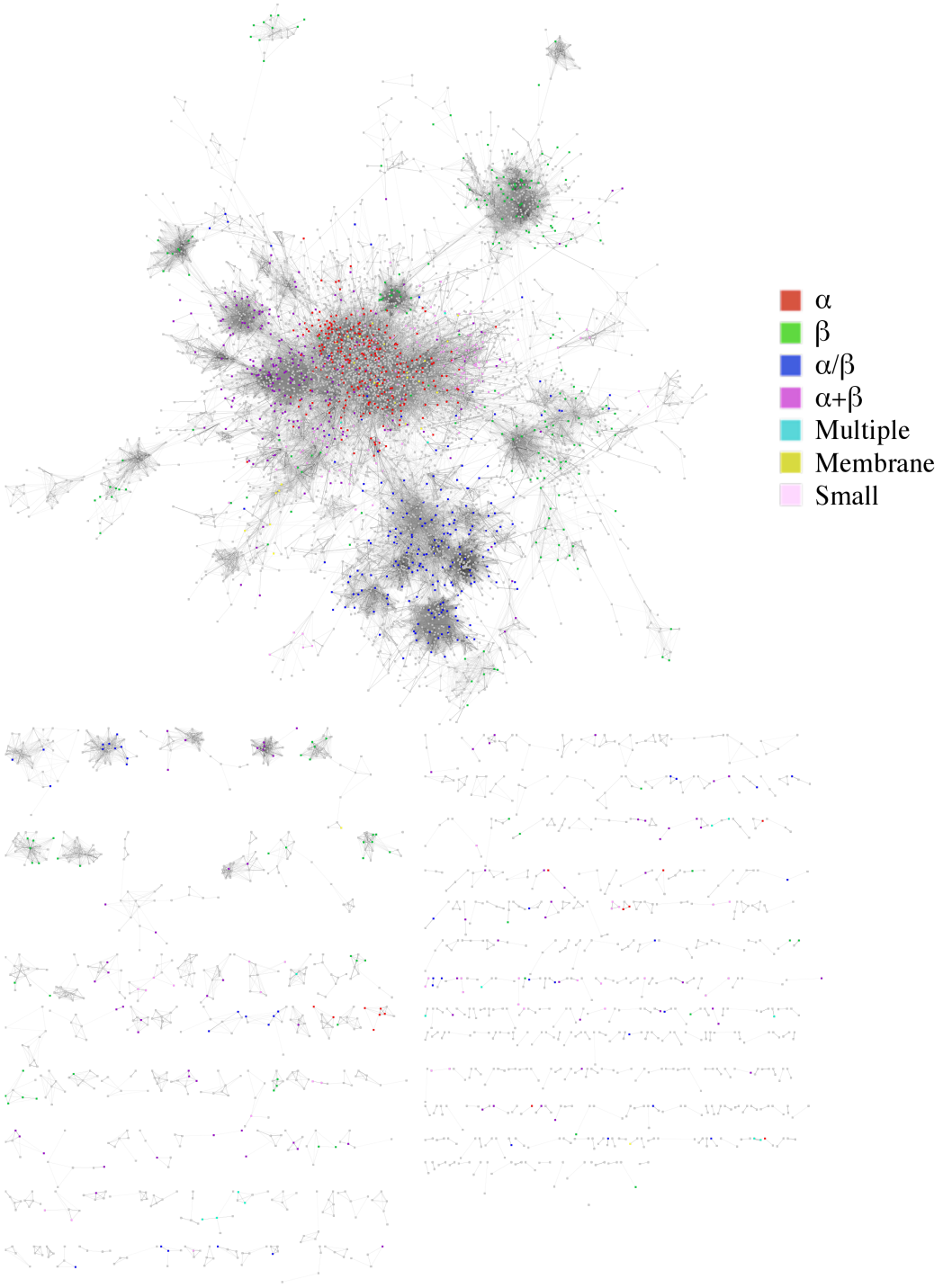


Figure 4.1: Structural similarity network for all protein chains. Structural clusters are colored according to their dominant SCOP class, or grey if no SCOP annotations were available.

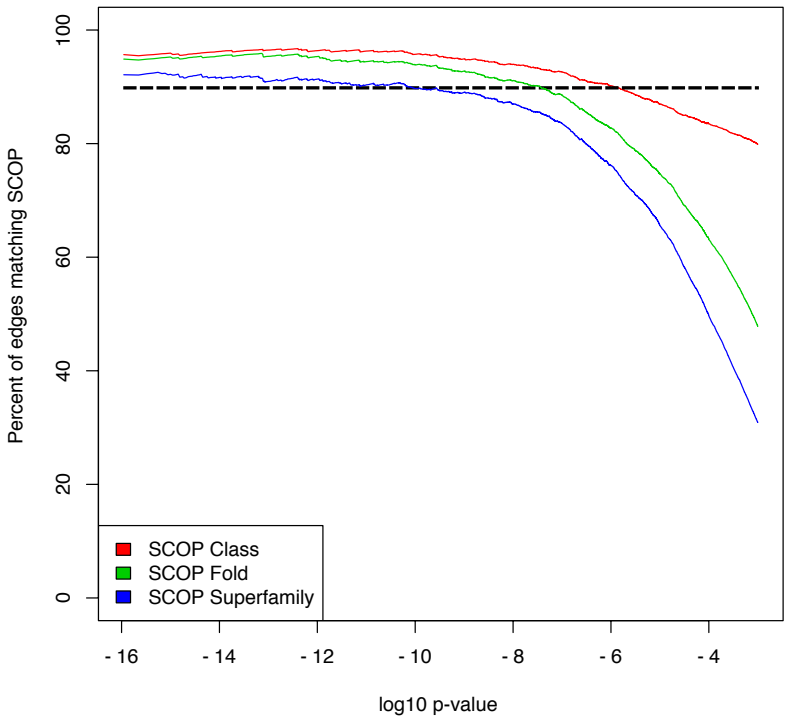


Figure 4.2: The agreement between the chain-based similarity network and SCOP categories. For a given clustering threshold, the plot gives the percentage of edges which connect two nodes of the same SCOP category. The dashed line shows that thresholds of $p < 10^{-6}$, $p < 10^{-7}$, and $p < 10^{-10}$ can represent SCOP class, fold, and superfamily classifications with 90% accuracy.

similarities. It can be seen that most nodes fall into a small number of discrete clusters that correspond to single TCDB classes. A few larger connected components contain several transporter classes with significant structural similarity, such as the alpha-helical membrane proteins. One structural class with notable functional diversity is shown in Figure 4.3b, where four of the five main transporter classes are present. These were found to be Rossmann-like folds, which are common as accessory domains for membrane proteins.

Properties such as protein symmetry can also be used to annotate the network. Figure 4.4 shows the β-propeller domain-level structural similarities. Most propellers form a single densely connected graph, indicating the strong stuctural similarity within

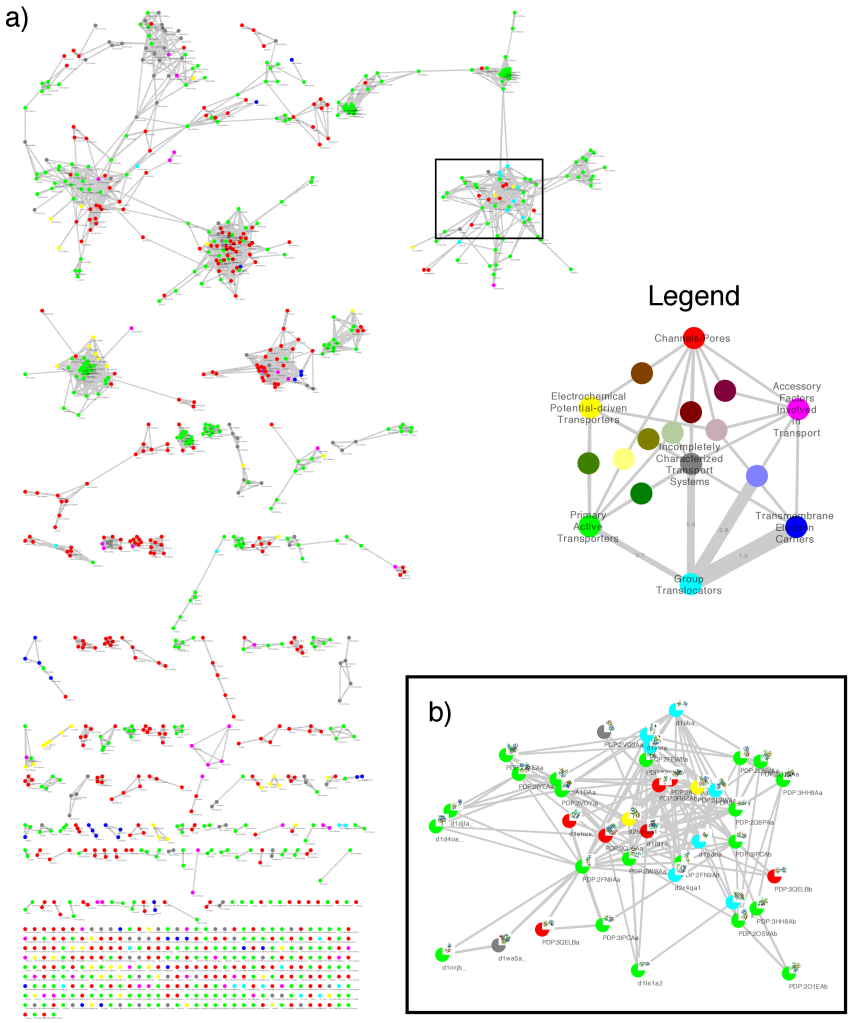


Figure 4.3: TCDB structural similarity network. (a) Structural similarities between domains of transmembrane proteins, as classified by the Transporter Classification Database (TCDB). A moderate significance threshold of TM-Score ≥ 0.5 was used, leaving the network fairly sparse. The nodes are colored according to their TCDB classification, with intermediate colors representing clusters with multiple TCDB annotations. The maximum TM-Score between clusters is indicated by the edge thickness. (b) Enlargement of the boxed section of the graph, with 3D structures overlaid. All structures are similar to the Rossmann fold, but several are classified in different SCOP folds (e.g. PTS system IIB component-like [SCOP:d1iiba_]. A wide diversity of TCDB functions are annotated to this subgraph.

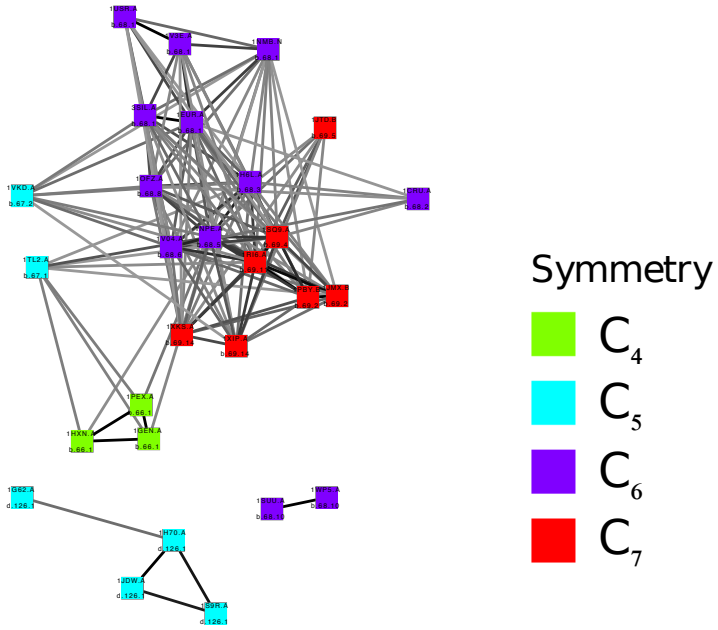


Figure 4.4: The β-propeller subgraph. Nodes are colored according to propeller symmetry.

this family. Since the structural comparisons were performed at the level of domains, proteins with the same order of internal symmetry tend to cluster tightly together. Decomposition of the structures into their internal repeats using CE-Symm (Chapter 7) would presumably improve the alignment between structures with different order and identify similarities and differences in the core blade motif.

4.4 Discussion and Conclusion

Structural similarity networks are a valuable tool for visualizing and understanding fold space. Rather than imposing either discrete clusters or a continuous metric on fold space, the network is able to capture both discrete and continuous features by varying the thresholds chosen for clustering and significance.

Annotating the network with various types of functional data can highlight correlations between structure and function. It is also useful for identifying cases where function

does not follow structure, either in the case of promiscuous folds with many functions (such as the Rossmann-like folds), or for functions that can be found in numerous folds.

One challenge in combining pairwise structural comparisons is that the alignments between three related proteins are not transitive due to the possibility of aligning different parts of the structures (Sadreyev et al., 2009). One method to combat this is to enforce high coverage in the alignments, but this misses cases with large insertions. Reducing the aligned subunits to the most evolutionarily relevant unit is important for mitigating this. Manual inspection of the alignments from the domain-based comparison showed that they tended to better capture the structural relationships than in the original chain-based comparison. Further reducing the subunits, for example through the use of CE-Symm to detect internal repeats or CE-CP (Chapter 6) to compare proteins with a circular permutation, would be expected to give additional gains in the ability of the network to fully capture structural relationships.

4.5 Acknowledgments

Andreas Prlić provided advice and mentoring for this research. Figures were generated using Cytoscape (Shannon, 2003; Smoot et al., 2011) and R (Wickham, 2009; R Development Core Team).

4.6 Bibliography

N. Alexandrov and I. Shindyalov. PDP: protein domain parser. *Bioinformatics*, 19(3): 429–430, Feb. 2003.

S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman. Basic local alignment search tool. *J Mol Biol*, 215(3):403–410, Oct. 1990.

S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman. BLASTclust, 2004. URL http://www.ncbi.nlm.nih.gov/Web/Newsltr/Spring04/blastlab.html.

A. Andreeva, D. Howorth, C. Chothia, E. Kulesha, and A. G. Murzin. SCOP2 prototype: a new approach to protein structure mining. *Nucleic Acids Res*, 42(Database issue): D310–4, Jan. 2014.

S. E. Bliven, A. Prlić, and P. Bourne. A comprehensive review of protein fold space and the correlation of structure with function. *F1000Posters*, 2(1292), Aug. 2011.

H. Cheng, R. D. Schaeffer, Y. Liao, L. N. Kinch, J. Pei, S. Shi, B.-H. Kim, and N. V. Grishin. ECOD: An Evolutionary Classification of Protein Domains. *PLoS Comput Biol*, 10(12):e1003926, Dec. 2014.

I. Friedberg and A. Godzik. Connecting the protein structure universe by using sparse recurring fragments. *Structure*, 13(8):1213–1224, Aug. 2005.

N. V. Grishin. Fold change in evolution of protein structures. *J Struct Biol*, 134(2-3): 167–185, Apr. 2001.

L. Holm and C. Sander. Protein structure comparison by alignment of distance matrices. *J Mol Biol*, 233(1):123–138, Sept. 1993.

L. Holm and C. Sander. Mapping the protein universe. *Science*, 273(5275):595–603, Aug. 1996.

J. Hou, G. E. Sims, C. Zhang, and S.-H. Kim. A global representation of the protein fold space. 100(5):2386–2390, Mar. 2003.

B. Marsden and R. A. Abagyan. SAD–a normalized structural alignment database: improving sequence-structure alignments. *Bioinformatics*, 20(15):2333–2344, Oct. 2004.

A. G. Murzin, S. E. Brenner, T. Hubbard, and C. Chothia. SCOP: a structural classification of proteins database for the investigation of sequences and structures. *J Mol Biol*, 247 (4):536–540, Apr. 1995.

S. Nepomnyachiy, N. Ben-Tal, and R. Kolodny. A GLOBAL VIEW OF THE PROTEIN UNIVERSE. In *3DSIG: Structural Bioinformatics and Computational Biophysics*, page 19, July 2014.

C. A. Orengo, T. P. Flores, W. R. Taylor, and J. M. Thornton. Identification and classification of protein fold families. *Protein Eng*, 6(5):485–500, July 1993.

C. A. Orengo, A. D. Michie, S. Jones, D. T. Jones, M. B. Swindells, and J. M. Thornton. CATH–a hierarchic classification of protein domain structures. *Structure*, 5(8):1093–1108, Aug. 1997.

R Development Core Team. R: A language and environment for statistical computing. Vienna, Austria.

B. Rost. Did evolution leap to create the protein universe? *Curr Opin Struct Biol*, 12(3): 409–416, June 2002.

R. I. Sadreyev, B.-H. Kim, and N. V. Grishin. Discrete-continuous duality of protein structure space. *Curr Opin Struct Biol*, 19(3):321–328, June 2009.

M. H. Saier, C. V. Tran, and R. D. Barabote. TCDB: the Transporter Classification Database for membrane transport protein analyses and information. *Nucleic Acids Res*, 34(Database issue):D181–6, Jan. 2006.

M. H. Saier, M. R. Yen, K. Noto, D. G. Tamang, and C. Elkan. The Transporter Classification Database: recent advances. *Nucleic Acids Res*, 37(Database issue): D274–8, Jan. 2009.

M. H. Saier, V. S. Reddy, D. G. Tamang, and A. Västermark. The transporter classification database. *Nucleic Acids Res*, 42(Database issue):D251–8, Jan. 2014.

P. Shannon. Cytoscape: A Software Environment for Integrated Models of Biomolecular Interaction Networks. *Genome Res*, 13(11):2498–2504, Nov. 2003.

I. N. Shindyalov and P. E. Bourne. An alternative view of protein fold space. *Proteins*, 38(3):247–260, Feb. 2000.

M. J. Sippl. On distance and similarity in fold space. *Bioinformatics*, 24(6):872–873, Mar. 2008.

M. J. Sippl. Fold space unlimited. *Curr Opin Struct Biol*, 19(3):312–320, June 2009.

J. Skolnick, A. K. Arakaki, S. Y. Lee, and M. Brylinski. The continuity of protein structure space is an intrinsic property of proteins. 106(37):15690–15695, Sept. 2009.

M. E. Smoot, K. Ono, J. Ruscheinski, P.-L. Wang, and T. Ideker. Cytoscape 2.8: new features for data integration and network visualization. *Bioinformatics*, 27(3):431–432, Feb. 2011.

H. Wickham. *ggplot2*. Elegant Graphics for Data Analysis. Springer New York, 2009.

Chapter 5

Circular Permutations

5.1 Forward

Circular permutation is the simplest type of rearrangement which can occur within a protein. Proteins that undergo a circular permutation typically retain their overall structure, which provides a view into the evolutionary history of the protein. Circular permutation is linked to a wide range of functional changes, both in natural proteins and in artificially created permutations.

Research into CE-CP (Chapter 6) had begun in 2011 when PLOS Computational Biology began discussing Topic Pages. These were inspired by the observation that while Wikipedia is an extremely popular reference for all topics, few incentives existed to motivate scientists to contribute to the project. Topic Pages were envisioned as a way to improve the quality of computational biology articles on Wikipedia (Wodak et al., 2012). They are review-style articles suitable for the general audience, and published simultaneously as a peer reviewed, static copy of record for the page in PLOS Computational Biology and as a living, community-edited Wikipedia article. This process is possible because the open access license that PLOS Computational Biology uses is compatible with Wikipedia's Creative Commons license.

Due to Philip Bourne's position as editor in chief of PLOS Computational Biology

at the time, we learned about the Topic Page concept when it was first implemented. Our article on Circular Permutation in Proteins was the first published Topic Page. As part of the article, I also set up a private MediaWiki instance to allow drafting of topic pages, which I still maintain on behalf of PLOS Computational Biology.

The remaining text of chapter 5 is a reprint of the material from:

> S. E. Bliven and A. Prlić. Circular permutation in proteins. *PLoS Comput Biol*, 8(3):e1002445, Mar. 2012

The most current version of the article is available on Wikipedia at https://en.wikipedia.org/wiki/Circular_permutation_in_proteins.

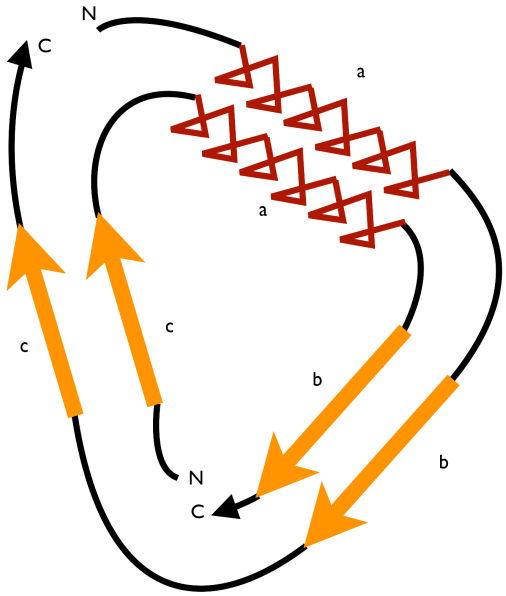


Figure 5.1: Schematic representation of a circular permutation in two proteins. The first protein (outer circle) has the sequence a-b-c. After the permutation the second protein (inner circle) has the sequence c-a-b. The letters N and C indicate the location of the amino- and carboxy-termini of the protein sequences and how their positions change relative to each other.

Circular permutation describes a type of relationship between proteins, whereby the proteins have a changed order of amino acids in their protein sequence, such that the sequence of the first portion of one protein (adjacent to the N-terminus) is related to that of the second portion of the other protein (near its C-terminus), and vice versa (see Figure 5.1). This is directly analogous to the mathematical notion of a cyclic permutation over the set of residues in a protein.

Circular permutation can be the result of evolutionary events, post-translational modifications, or artificially engineered mutations. The result is a protein structure with different connectivity, but overall similar three-dimensional (3D) shape. The homology between portions of the proteins can be established by observing similar sequences between N- and C-terminal portions of the two proteins, structural similarity, or other methods.

5.2 History

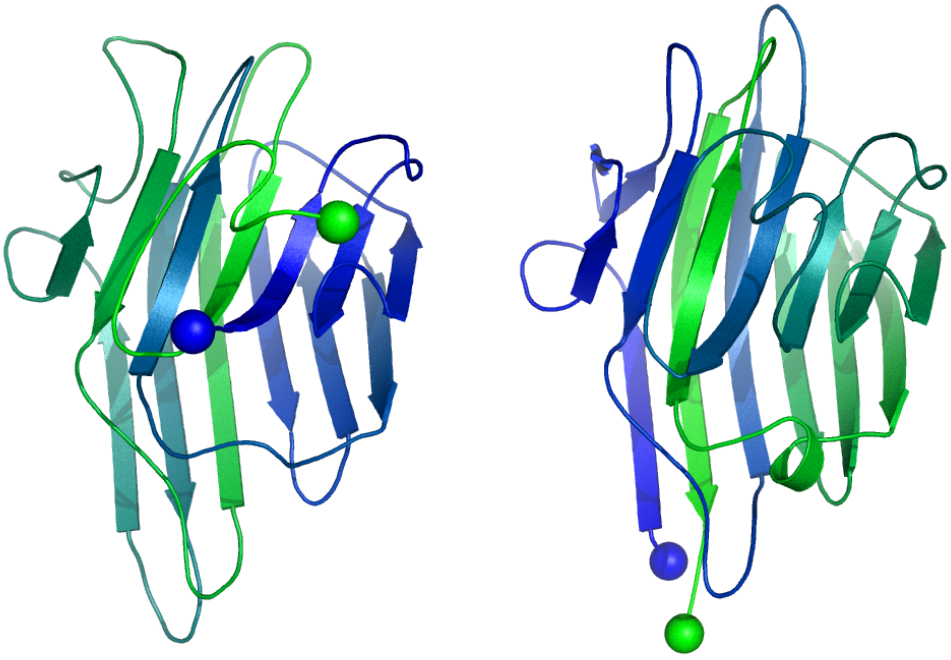


Figure 5.2: Two proteins that are related by a circular permutation. Concanavalin A (left), from the Protein Data Bank [PDB:3CNA], and peanut lectin (right), from [PDB:2PEL], which is homologous to favin. The termini of the proteins are highlighted by blue and green spheres, and the sequence of residues is indicated by the gradient from blue (N-terminus) to green (C-terminus). The 3D fold of the two proteins is highly similar; however, the N- and C-termini are located on different positions of the protein (Cunningham et al., 1979).

In 1979, Bruce Cunningham and his colleagues discovered the first instance of a circularly permuted protein in nature (Taylor, 2007; Cunningham et al., 1979). After determining the peptide sequence of the lectin protein favin, they noticed its similarity to a known protein—concanavalin A—except that the ends were circularly permuted (see Figure 5.2). Later work confirmed the circular permutation between the pair (Einspahr et al., 1986) and showed that concanavalin A is permuted post-translationally (Carrington et al., 1985) through cleavage and an unusual protein ligation (Bowles and Pappin, 1988).

After the discovery of a natural circularly permuted protein, researchers looked

for a way to emulate this process. In 1983, David Goldenberg and Thomas Creighton were able to create a circularly permuted version of a protein by chemically ligating the termini to create a cyclic protein, then introducing new termini elsewhere using trypsin (Goldenberg and Creighton, 1983). In 1989, Karolin Luger and her colleagues introduced a genetic method for making circular permutations by carefully fragmenting and ligating DNA (Luger et al., 1989). This method allowed for permutations to be introduced at arbitrary sites, and is still used today to design circularly permuted proteins in the lab.

Despite the early discovery of post-translational circular permutations and the suggestion of a possible genetic mechanism for evolving circular permutants, it was not until 1995 that the first circularly permuted pair of genes were discovered. Saposins are a class of proteins involved in sphingolipid catabolism and lipid antigen presentation in humans. Christopher Ponting and Robert Russell identified a circularly permuted version of a saposin inserted into plant aspartic proteinase, which they nicknamed swaposin (Ponting and Russell, 1995). Saposin and swaposin were the first known case of two natural genes related by a circular permutation.

Hundreds of examples of protein pairs related by a circular permutation were subsequently discovered in nature or produced in the laboratory. The Circular Permutation Database contains 2,238 circularly permuted protein pairs with known structures, and many more are known without structures (Lo et al., 2009). The CyBase database collects proteins that are cyclic, some of which are permuted variants of cyclic wild-type proteins (Kaas and Craik, 2010). SISYPHUS is a database that contains a collection of hand-curated manual alignments of proteins with non-trivial relationships, several of which have circular permutations (Andreeva et al., 2007).

5.3 Evolution

There are two main models that are currently being used to explain the evolution of circularly permuted proteins: *permutation by duplication* and *fission and fusion*. The two models have compelling examples supporting them, but the relative contribution of

each model in evolution is still under debate (Weiner and Bornberg-Bauer, 2006). Other, less common, mechanisms have been proposed, such as ”cut and paste” (Bujnicki, 2002) or ”exon shuffling”.

5.3.1 Permutation by Duplication

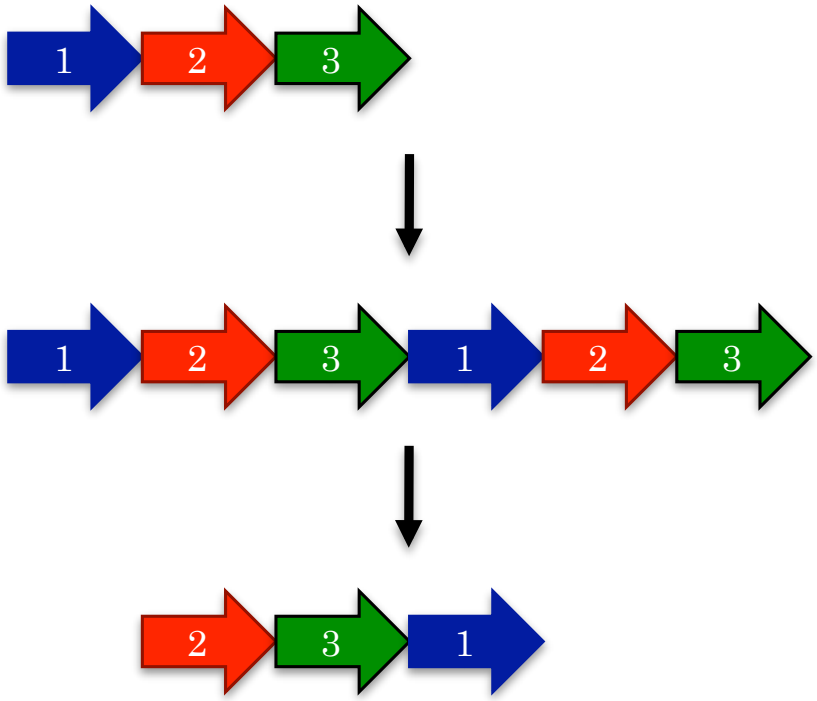


Figure 5.3: The permutation by duplication mechanism for producing a circular permutation. First, a gene is duplicated in place. Next, start and stop codons are introduced, resulting in a circularly permuted gene.

The earliest model proposed for the evolution of circular permutations is the permutation by duplication mechanism (Cunningham et al., 1979). In this model, a precursor gene first undergoes a duplication and fusion to form a large tandem repeat. Next, start and stop codons are introduced at corresponding locations in the duplicated gene, removing redundant sections of the protein (Figure 5.3).

One surprising prediction of the permutation by duplication mechanism is that intermediate permutations can occur. For instance, the duplicated version of the protein

should still be functional, since otherwise evolution would quickly select against such proteins. Likewise, partially duplicated intermediates where only one terminus was truncated should be functional. Such intermediates have been extensively documented in protein families such as DNA methyltransferases (Jeltsch, 1999).

Saposin and Swaposin

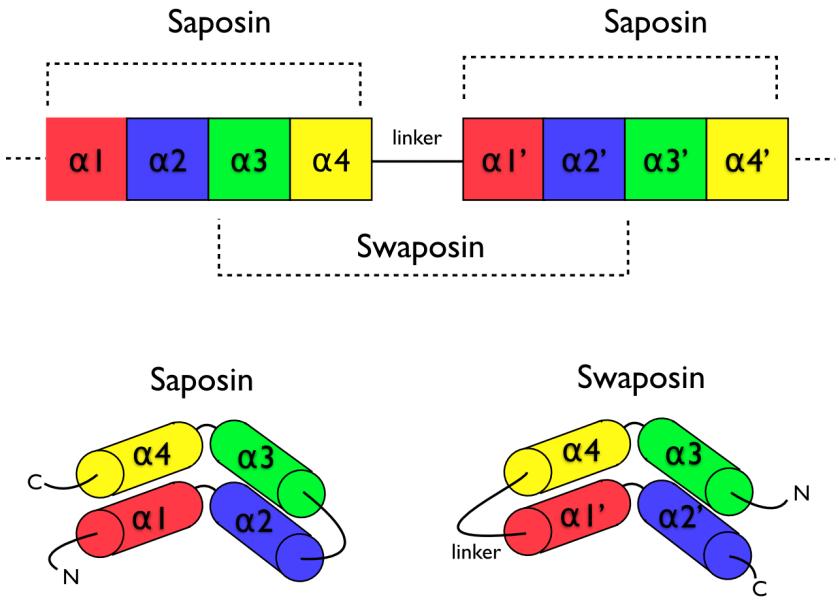


Figure 5.4: Suggested relationship between saposin and swaposin. They could have evolved from a similar gene (Ponting and Russell, 1995). Both consist of 4 alpha helices with the order of helices being permuted relative to each other.

An example for permutation by duplication is the relationship between saposin and swaposin. Saposins are highly conserved glycoproteins that consist of an approximately 80 amino acid residue long protein forming a four alpha helical structure. They have a nearly identical placement of cysteine residues and glycosylation sites. The cDNA sequence that codes for saposin is called prosaposin. It is a precursor for four cleavage products, the saposins A, B, C, and D. The four saposin domains most likely arose from two tandem duplications of an ancestral gene (Hazkani-Covo et al., 2002). This repeat

suggests a mechanism for the evolution of the relationship with the plant-specific insert (PSI) (see Figure 5.4). The PSI is a domain exclusively found in plants, consisting of approximately 100 residues and found in plant aspartic proteases (Guruprasad et al., 1994). It belongs to the saposin-like protein family (SAPLIP) and has the N- and C-termini ”swapped”, such that the order of helices is 3-4-1-2 compared with saposin, thus leading to the name ”swaposin” (Ponting and Russell, 1995). For a review on functional and structural features of saposin-like proteins, see Bruhn (2005).

5.3.2 Fission and Fusion

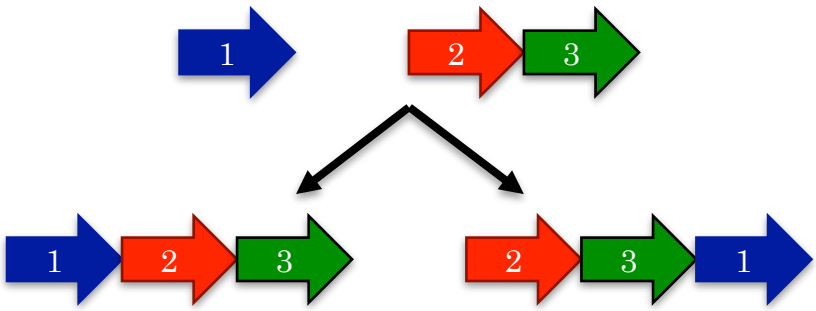


Figure 5.5: The fission and fusion mechanism of circular permutation. Two separate genes arise (potentially from the fission of a single gene). If the genes fuse together in different orders in two orthologues, a circular permutation occurs.

Another model for the evolution of circular permutations is the fission and fusion model. The process starts with two partial proteins. These may represent two independent polypeptides (such as two parts of a heterodimer), or may have originally been halves of a single protein that underwent a fission event to become two polypeptides (see Figure 5.5).

The two proteins can later fuse together to form a single polypeptide. Regardless of which protein comes first, this fusion protein may show similar function. Thus, if a fusion between two proteins occurs twice in evolution (either between paralogues within the same species or between orthologues in different species) but in a different order, the resulting fusion proteins will be related by a circular permutation.

Evidence for a particular protein having evolved by a fission and fusion mechanism can be provided by observing the halves of the permutation as independent polypeptides in related species, or by demonstrating experimentally that the two halves can function as separate polypeptides (Lee and Blaber, 2011).

Transhydrogenases

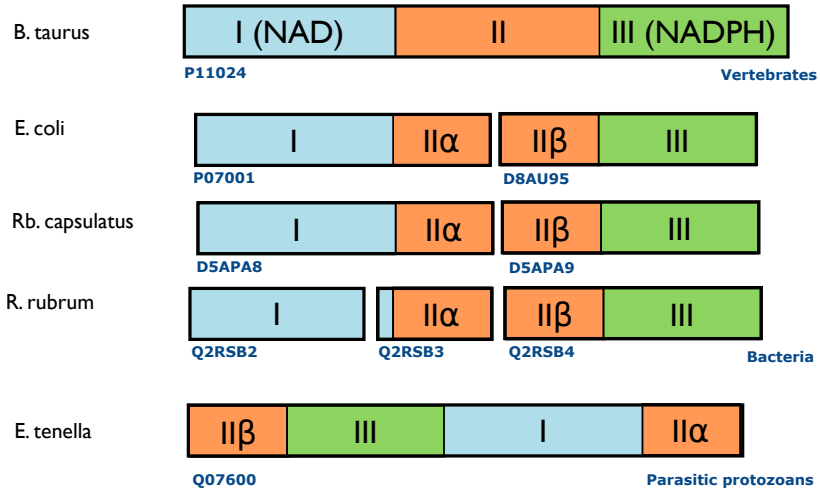


Figure 5.6: Transhydrogenases in various organisms can be found in three different domain arrangements. In cattle, the three domains are arranged sequentially. In the bacteria E. coli, Rb. capsulatus, and R. rubrum, the transhydrogenase consists of two or three subunits. Finally, transhydrogenase from the protist E. tenella consists of a single subunit that is circularly permuted relative to cattle transhydrogenase (Hatefi and Yamaguchi, 1996).

An example for the fission and fusion mechanism can be found in nicotinamide nucleotide transhydrogenases (Hatefi and Yamaguchi, 1996). These are membrane-bound enzymes that catalyze the transfer of a hydride ion between NAD(H) and NADP(H) in a reaction that is coupled to transmembrane proton translocation. They consist of three major functional units (I, II, and III) that can be found in different arrangement in bacteria, protozoa, and higher eukaryotes (see Figure 5.6). Phylogenetic analysis suggests that the three groups of domain arrangements were acquired and fused independently (Weiner and Bornberg-Bauer, 2006).

5.3.3 Other Processes that can Lead to Circular Permutations

Post-Translational Modification

The two evolutionary models mentioned above describe ways in which genes may be circularly permuted, resulting in a circularly permuted mRNA after transcription. Proteins can also be circularly permuted via post-translational modification, without permuting the underlying gene. Circular permutations can happen spontaneously through auto-catalysis, as in the case of concanavalin A (see Figure 5.2) (Bowles and Pappin, 1988). Alternately, permutation may require restriction enzymes and ligases (Goldenberg and Creighton, 1983).

5.4 The Role of Circular Permutations in Protein Engineering

Many proteins have their termini located close together in 3D space (Thornton and Sibanda, 1983; Yu and Lutz, 2011). Because of this, it is often possible to design circular permutations of proteins. Today, circular permutations are generated routinely in the lab using standard genetics techniques (Luger et al., 1989). Although some permutation sites prevent the protein from folding correctly, many permutants have been created with nearly identical structure and function to the original protein.

The motivation for creating a circular permutant of a protein can vary. Scientists may want to improve some property of the protein, such as

- **Reduce proteolytic susceptibility.** The rate at which proteins are broken down can have a large impact on their activity in cells. Since termini are often accessible to proteases, designing a circularly permuted protein with less accessible termini can increase the lifespan of that protein in the cell (Whitehead et al., 2009).

- **Improve catalytic activity.** Circularly permuting a protein can sometimes increase the rate at which it catalyzes a chemical reaction, leading to more efficient proteins

(Cheltsov et al., 2001).

- **Alter substrate or ligand binding.** Circularly permuting a protein can result in the loss of substrate binding, but can occasionally lead to novel ligand binding activity or altered substrate specificity (Qian and Lutz, 2005).

- **Improve thermostability.** Making proteins active over a wider range of temperatures and conditions can improve their utility (Topell et al., 1999).

  Alternately, scientists may be interested in properties of the original protein, such as

- **Fold order.** Determining the order in which different parts of a protein fold is challenging due to the extremely fast time scales involved. Circularly permuted versions of proteins will often fold in a different order, providing information about the folding of the original protein (Viguera et al., 1996; Capraro et al., 2008; Zhang and Schachman, 1996).

- **Essential structural elements.** Artificial circularly permuted proteins can allow parts of a protein to be selectively deleted. This gives insight into which structural elements are essential or not (Huang et al., 2011).

- **Modify quaternary structure.** Circularly permuted proteins have been shown to take on different quaternary structure than wild-type proteins (Beernink et al., 2001).

- **Find insertion sites for other proteins.** Inserting one protein as a domain into another protein can be useful. For instance, inserting calmodulin into green fluorescent protein (GFP) allowed researchers to measure the activity of calmodulin via the florescence of the split-GFP (Baird et al., 1999). Regions of GFP that tolerate the introduction of circular permutation are more likely to accept the addition of another protein while retaining the function of both proteins.

- **Design of novel biocatalysts and biosensors.** Introducing circular permutations can be used to design proteins to catalyze specific chemical reactions (Turner, 2009; Cheltsov et al., 2001), or to detect the presence of certain molecules using proteins. For instance, the GFP-calmodulin fusion described above can be used to detect the level of calcium ions in a sample (Baird et al., 1999).

5.5 Algorithmic Detection of Circular Permutations

Many sequence alignment and protein structure alignment algorithms have been developed assuming linear data representations and as such are not able to detect circular permutations between proteins. Two examples of frequently used methods that have problems correctly aligning proteins related by circular permutation are dynamic programming and many hidden Markov models. As an alternative to these, a number of algorithms are built on top of non-linear approaches and are able to detect topology-independent similarities, or employ modifications allowing them to circumvent the limitations of dynamic programming. Table 5.1 is a collection of such methods.

The algorithms are classified according to the type of input they require. *Sequence*-based algorithms require only the sequence of two proteins in order to create an alignment. Sequence methods are generally fast and suitable for searching whole genomes for circularly permuted pairs of proteins. *Structure*-based methods require 3D structures of both proteins being considered. They are often slower than sequence-based methods, but are able to detect circular permutations between distantly related proteins with low sequence similarity. Some structural methods are *topology independent*, meaning that they are also able to detect more complex rearrangements than circular permutation.

Table 5.1: Algorithmic Detection of Circular Permutations

Name	Type	Description	Reference
FBPLOT	Sequence	Draws dot plots of suboptimal sequence alignments	Zuker (1991)

continued ...

Table 5.1: (Continued) Algorithmic Detection of Circular Permutations

Name	Type	Description	Reference
Bachar et al	Structure, topology independent	Uses geometric hashing for the topology independent comparison of proteins	Bachar et al. (1993)
Uliel et al	Sequence	First suggestion of how a sequence comparison algorithm for the detection of circular permutations can work	Uliel et al. (1999)
SHEBA	Structure	Duplicates a sequence in the middle; uses SHEBA algorithm for structure alignment; determines new cut position after structure alignment	Jung and Lee (2001)
Multiprot	Structure, Topology independent	Calculates a sequence order independent multiple protein structure alignment	Shatsky et al. (2004)
RASPODOM	Sequence	Modified Needleman & Wunsch sequence comparison algorithm	Weiner et al. (2005)
CPSARST	Structure	Describes protein structures as one-dimensional text strings by using a Ramachandran sequential transformation (RST) algorithm. Detects circular permutations through a duplication of the sequence represention and ”double filter-and-refine” strategy.	Lo and Lyu (2008)
GANGSTA +	Structure	Works in two stages: Stage one identifies coarse alignments based on secondary structure elements. Stage two refines the alignment on residue level and extends into loop regions.	Schmidt-Goenner et al. (2010)
SANA	Structure	Detect initial aligned fragment pairs (AFPs). Build network of possible AFPs. Use random-mate algorithm to connect components to a graph.	Wang et al. (2010)
CE-CP	Structure	Built on top of the combinatorial extension algorithm. Duplicates atoms before alignment, truncates results after alignment	Prlić et al. (2010); Bliven et al. (2015)

5.6 Further Reading

- David Goodsell (2010) "Concanavalin A and Circular Permutation." *Research Collaboratory for Structural Biology (RCSB) Protein Data Bank (PDB)*. Molecule of the Month. April 2010. http://www.rcsb.org/pdb/101/motm.do?momID=124
- Yu and Lutz (2011), for a review of the use of circular permutation in protein design.
- Weiner and Bornberg-Bauer (2006), for a review of evolutionary mechanisms for circular permutations.
- Cyclic permutation on Wikipedia (https://en.wikipedia.org/wiki/Cyclic_permutation)

5.7 Acknowledgements

A stub of this article authored primarily by Andreas Prlić was previously published on Wikipedia. The authors built on the version of 00:53, 29 May 2011 (http://en.wikipedia.org/w/index.php?title=Circular_permutation_in_proteins&oldid=431415525) and were careful not to include the work of others for licensing reasons and rather provide a complete re-write of the first version. Still, they would like to thank individuals from the Wikipedia community for subsequent contributions to the original stub.

Daniel Mietchen and Shoshana Wodak, the editors for the Topic Page section, were very helpful with finding a publication procedure which worked with both PLOS and Wikipedia. The PLOS IT staff were also very responsive when setting up the Wiki server. The Wikipedia community was quite helpful explaining the different expectations for an encyclopedic article compared to a normal review, particularly WikiProject Computational Biology. Finally, thanks to all the Wikipedia users who contributed to improve the article after it's publication.

This chapter was originally published as:

S. E. Bliven and A. Prlić. Circular permutation in proteins. *PLoS Comput Biol*, 8(3):e1002445, Mar. 2012

Author Contributions: Both authors contributed to writing this manuscript. I was additionally responsible for setting up the public MediaWiki server for drafting the manuscript and other topic page manuscripts.

5.8 Bibliography

A. Andreeva, A. Prlić, T. J. P. Hubbard, and A. G. Murzin. SISYPHUS–structural alignments for proteins with non-trivial relationships. *Nucleic Acids Res*, 35(Database issue):D253–9, 2007.

O. Bachar, D. Fischer, R. Nussinov, and H. Wolfson. A computer vision based technique for 3-D sequence-independent structural comparison of proteins. *Protein Eng*, 6(3): 279–288, Apr. 1993.

G. S. Baird, D. A. Zacharias, and R. Y. Tsien. Circular permutation and receptor insertion within green fluorescent proteins. 96(20):11241–11246, Sept. 1999.

P. T. Beernink, Y. R. Yang, R. Graf, D. S. King, S. S. Shah, and H. K. Schachman. Random circular permutation leading to chain disruption within and near alpha helices in the catalytic chains of aspartate transcarbamoylase: effects on assembly, stability, and function. *Protein Sci*, 10(3):528–537, Mar. 2001.

S. E. Bliven and A. Prlić. Circular permutation in proteins. *PLoS Comput Biol*, 8(3): e1002445, Mar. 2012.

S. E. Bliven, P. E. Bourne, and A. Prlić. Detection of circular permutations within protein structures using CE-CP. *Bioinformatics*, 31(8):1316–1318, Apr. 2015.

D. J. Bowles and D. J. Pappin. Traffic and assembly of concanavalin A. *Trends Biochem Sci*, 13(2):60–64, Feb. 1988.

H. Bruhn. A short guided tour through functional and structural features of saposin-like proteins. *Biochem. J.*, 389(Pt 2):249–257, July 2005.

J. M. Bujnicki. Sequence permutations in the molecular evolution of DNA methyltransferases. *BMC Evol. Biol.*, 2:3, Mar. 2002.

D. T. Capraro, M. Roy, J. N. Onuchic, and P. A. Jennings. Backtracking on the folding landscape of the beta-trefoil protein interleukin-1beta? 105(39):14844–14848, Sept. 2008.

D. M. Carrington, A. Auffret, and D. E. Hanke. Polypeptide ligation occurs during post-translational modification of concanavalin A. *Nature*, 313(5997):64–67, 1985.

A. V. Cheltsov, M. J. Barber, and G. C. Ferreira. Circular permutation of 5-aminolevulinate synthase. Mapping the polypeptide chain to its function. *J Biol Chem*, 276(22):19141–19149, June 2001.

B. A. Cunningham, J. J. Hemperly, T. P. Hopp, and G. M. Edelman. Favin versus concanavalin A: Circularly permuted amino acid sequences. 76(7):3218–3222, July 1979.

H. Einspahr, E. H. Parks, K. Suguna, E. Subramanian, and F. L. Suddath. The crystal structure of pea lectin at 3.0-A resolution. *J Biol Chem*, 261(35):16518–16527, Dec. 1986.

D. P. Goldenberg and T. E. Creighton. Circular and circularly permuted forms of bovine pancreatic trypsin inhibitor. *J Mol Biol*, 165(2):407–413, Apr. 1983.

K. Guruprasad, K. Törmäkangas, J. Kervinen, and T. L. Blundell. Comparative modelling of barley-grain aspartic proteinase: a structural rationale for observed hydrolytic specificity. *FEBS Lett*, 352(2):131–136, Sept. 1994.

Y. Hatefi and M. Yamaguchi. Nicotinamide nucleotide transhydrogenase: a model for utilization of substrate binding energy for proton translocation. *FASEB J*, 10(4): 444–452, Mar. 1996.

E. Hazkani-Covo, N. Altman, M. Horowitz, and D. Graur. The evolutionary history of prosaposin: two successive tandem-duplication events gave rise to the four saposin domains in vertebrates. *J Mol Evol*, 54(1):30–34, Jan. 2002.

Y.-M. Huang, S. Nayak, and C. Bystroff. Quantitative in vivo solubility and reconstitution of truncated circular permutants of green fluorescent protein. *Protein Sci*, 20(11): 1775–1780, Nov. 2011.

A. Jeltsch. Circular permutations in the molecular evolution of DNA methyltransferases. *J Mol Evol*, 49(1):161–164, July 1999.

J. Jung and B. Lee. Circularly permuted proteins in the protein structure database. *Protein Sci*, 10(9):1881–1886, Sept. 2001.

Q. Kaas and D. J. Craik. Analysis and classification of circular proteins in CyBase. *Biopolymers*, 94(5):584–591, 2010.

J. Lee and M. Blaber. Experimental support for the evolution of symmetric protein architecture from a simple peptide motif. 108(1):126–130, Jan. 2011.

W.-C. Lo and P.-C. Lyu. CPSARST: an efficient circular permutation search tool applied to the detection of novel protein structural relationships. *Genome Biol*, 9(1):R11, 2008.

W.-C. Lo, C.-C. Lee, C.-Y. Lee, and P.-C. Lyu. CPDB: a database of circular permutation in proteins. *Nucleic Acids Res*, 37(Database issue):D328–32, 2009.

K. Luger, U. Hommel, M. Herold, J. Hofsteenge, and K. Kirschner. Correct folding of circularly permuted variants of a beta alpha barrel enzyme in vivo. *Science*, 243(4888): 206–210, Jan. 1989.

C. P. Ponting and R. B. Russell. Swaposins: circular permutations within genes encoding saposin homologues. *Trends Biochem Sci*, 20(5):179–180, May 1995.

A. Prlić, S. E. Bliven, P. W. Rose, W. F. Bluhm, C. Bizon, A. Godzik, and P. E. Bourne. Pre-calculated protein structure alignments at the RCSB PDB website. *Bioinformatics*, 26(23):2983–2985, Dec. 2010.

Z. Qian and S. Lutz. Improving the Catalytic Activity of CandidaantarcticaLipase B by Circular Permutation. *J Am Chem Soc*, 127(39):13466–13467, Oct. 2005.

T. Schmidt-Goenner, A. Guerler, B. Kolbeck, and E. W. Knapp. Circular permuted proteins in the universe of protein folds. *Proteins: Structure, Function, and Bioinformatics*, 78(7):1618–1630, May 2010.

M. Shatsky, R. Nussinov, and H. J. Wolfson. A method for simultaneous alignment of multiple protein structures. *Proteins: Structure, Function, and Bioinformatics*, 56(1): 143–156, July 2004.

W. R. Taylor. Evolutionary transitions in protein fold space. *Curr Opin Struct Biol*, 17 (3):354–361, June 2007.

J. M. Thornton and B. L. Sibanda. Amino and carboxy-terminal regions in globular proteins. *J Mol Biol*, 167(2):443–460, June 1983.

S. Topell, J. Hennecke, and R. Glockshuber. Circularly permuted variants of the green fluorescent protein. *FEBS Lett*, 457(2):283–289, Aug. 1999.

N. J. Turner. Directed evolution drives the next generation of biocatalysts. *Nat. Chem. Biol.*, 5(8):567–573, Aug. 2009.

S. Uliel, A. Fliess, A. Amir, and R. Unger. A simple algorithm for detecting circular permutations in proteins. *Bioinformatics*, 15(11):930–936, Nov. 1999.

A. R. Viguera, L. Serrano, and M. Wilmanns. Different folding transition states may result in the same native structure. *Nat Struct Biol*, 3(10):874–880, Oct. 1996.

L. Wang, L.-Y. Wu, Y. Wang, X.-S. Zhang, and L. Chen. SANA: an algorithm for sequential and non-sequential protein structure alignment. *Amino Acids*, 39(2):417–425, July 2010.

J. Weiner and E. Bornberg-Bauer. Evolution of circular permutations in multidomain proteins. *Mol. Biol. Evol.*, 23(4):734–743, Apr. 2006.

J. Weiner, G. Thomas, and E. Bornberg-Bauer. Rapid motif-based prediction of circular permutations in multi-domain proteins. *Bioinformatics*, 21(7):932–937, Apr. 2005.

T. A. Whitehead, L. M. Bergeron, and D. S. Clark. Tying up the loose ends: circular permutation decreases the proteolytic susceptibility of recombinant proteins. *Protein Eng Des Sel*, 22(10):607–613, Oct. 2009.

S. J. Wodak, D. Mietchen, A. M. Collings, R. B. Russell, and P. E. Bourne. Topic pages: PLoS Computational Biology meets Wikipedia. *PLoS Comput Biol*, 8(3):e1002446, 2012.

Y. Yu and S. Lutz. Circular permutation: a different way to engineer enzyme structure and function. *Trends Biotechnol.*, 29(1):18–25, Jan. 2011.

P. Zhang and H. K. Schachman. In vivo formation of allosteric aspartate transcarbamoylase containing circularly permuted catalytic polypeptide chains: implications for protein folding and assembly. *Protein Sci*, 5(7):1290–1300, July 1996.

M. Zuker. Suboptimal sequence alignment in molecular biology. Alignment with error analysis. *J Mol Biol*, 221(2):403–420, Sept. 1991.

Chapter 6

Detection of Circular Permutations within Protein Structures using CE-CP

6.1 Forward

One factor used to distinguish protein structural comparison algorithms is their ability to align residues which occur in a difference order in each of the sequences, referred to as nonsequential or topology-independent alignments. Both CE and FATCAT are unable to detect nonsequential alignments, but several topology-independent algorithms exist for that purpose (Nguyen and Madhusudhan, 2011; Dundas et al., 2007; Abyzov and Ilyin, 2007; Guerler and Knapp, 2008).

The simplest form of nonsequential alignment is circular permutation, where the proteins consist of just two aligned blocks with different order. Circular permutation is an attractive relationship to detect because it has a clear evolutionary basis, as described below. Limiting the number of permutation sites also avoids problems of overfitting which are possible with more general topology-independent methods. Thus, algorithms specifically designed to detect circular permutations are desirable.

The CE-CP algorithm for aligning circularly permuted proteins was initially developed in 2011. Later development led to its incorporation into the RCSB PDB

website, and to its eventual publication.

The remaining text of chapter 6 is a reprint of the material from:

S. E. Bliven, P. E. Bourne, and A. Prlić. Detection of circular permutations within protein structures using CE-CP. *Bioinformatics*, 31(8):1316–1318, Apr. 2015

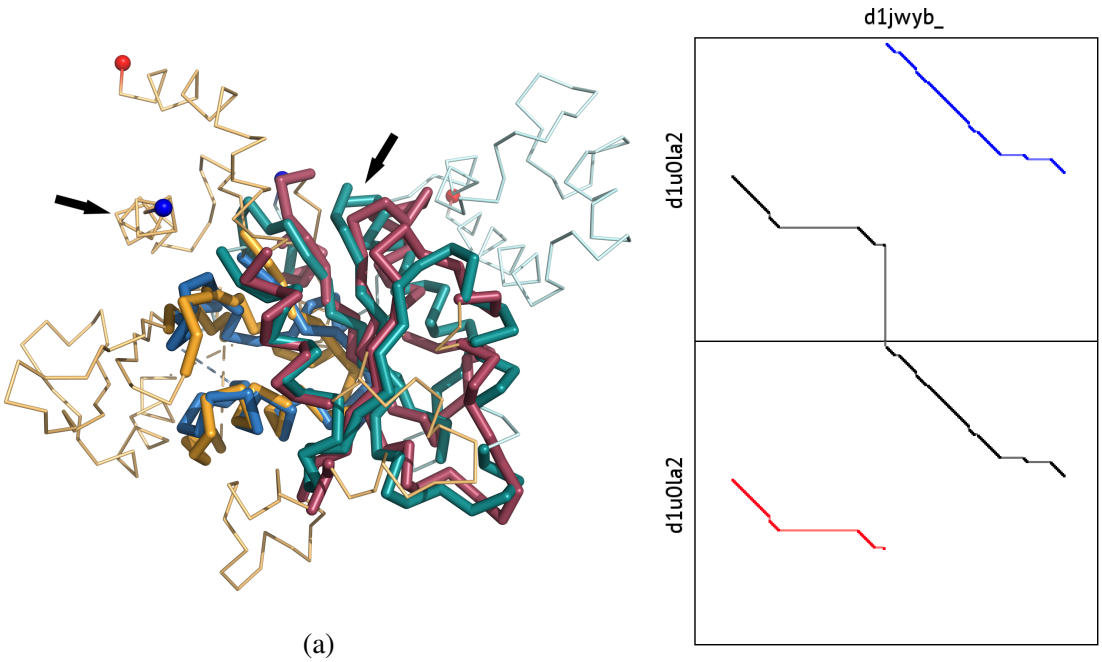
6.2 Abstract

Motivation: Circular permutation is an important type of protein rearrangement. Natural circular permutations have implications for protein function, stability, and evolution. Artificial circular permutations have also been used for protein studies. However, such relationships are difficult to detect for many sequence and structure comparison algorithms and require special consideration.

Results: We developed a new algorithm, called Combinatorial Extension for Circular Permutations (CE-CP), which allows the structural comparison of circularly permuted proteins. CE-CP was designed to be user friendly and is integrated into the RCSB Protein Data Bank. It was tested on two collections of circularly permuted proteins. Pairwise alignments can be visualized both in a desktop application or on the web using Jmol and exported to other programs in a variety of formats.

Availability: The CE-CP algorithm can be accessed through the RCSB website at http://www.rcsb.org/pdb/workbench/workbench.do. Source code is available under the LGPL 2.1 as part of BioJava 3 (http://biojava.org; urlhttp://github.com/biojava/biojava).

Contact: info@rcsb.org



(a) (b)

Figure 6.1: CE-CP algorithm for aligning circular permuted proteins. (a) CE-CP alignment of the dynamin A GTPase domain (yellow and red, SCOP:d1jwyb_) and the YjeQ protein (blue and green, SCOP:d1u0la2). N- and C-termini are shown with blue and red spheres. Arrows indicate the positions of the circular permutation. (b) Dotplot of the alignment with YjeQ duplicated. The optimal alignment is shown in black, with the inferred equivalent positions in blue and red.

6.3 Introduction

Circular permutation describes a relationship between two proteins where the N-terminal portion of one protein is related to the C-terminal portion of the other. While the order of amino acids changes, circularly permuted proteins are generally found to assume the same structure. Circular permutation has been documented to naturally occur in a number of protein families, such as lectins (Cunningham et al., 1979) and DNA methyltransferases (Jeltsch, 1999). Two general mechanisms for the evolution of circularly permuted proteins are known, so detecting such events can shed light on the evolutionary history of protein families (Weiner and Bornberg-Bauer, 2006). Circular permutation can influence protein folding, dynamics, and function (Bliven and Prlić, 2012). Synthetic circular permutatants have been engineered to alter activity, control

regulation, and improve stability (Yu and Lutz, 2011; Whitehead et al., 2009; Ostermeier, 2005).

Natural circular permutants generally have quite low sequence similarity, with previous studies finding less than 0.3–2.6% of proteins share >30% identity (Jung and Lee, 2001; Lo and Lyu, 2008). Thus, including structural information is essential for detecting circular permutations. Many structural alignment algorithms are unable to detect rearrangements in sequence, while general sequence-order independent methods lack a clear evolutionary mechanism by which complex rearrangements could occur. Therefore, algorithms that specifically search for circular permutations are needed. Several existing algorithms have been reported, including SHEBA (Jung and Lee, 2001) and CPSARST (Lo and Lyu, 2008).

Here we describe a method, Combinatorial Extension with Circular Permutations (CE-CP), for the identification of circular permutations based on protein structure.

6.4 Methods

Combinatorial Extension (CE) is a rigid-body structural comparison algorithm (Shindyalov and Bourne, 1998). It uses dynamic programming to identify regions of local similarity between the alpha-carbons of two protein structures, followed by iterative refinement to find a global superposition with low RMSD and high number of aligned residues.

To adapt CE to quickly find circular permutations, we use an algorithm analogous to that proposed by Uliel et al. (1999) for detecting circular permutations by sequence similarity. The atoms of the shorter structure are virtually duplicated. This allows the alignment of the first protein to wrap around from the carboxyl terminus to the amino terminus of the second protein (see Figure 6.1b). Thus, the path with optimal structural similarity will contain some residues from each copy of the duplicated protein, allowing the permutation site to be identified. In the case of structures that are not circular permuted, two equivalent paths are possible.

After identifying the highest scoring alignment to the duplicated protein, the result is processed to map the alignment onto the original query. While this is generally unambiguous for high-scoring alignments, it is possible that a single residue in the duplicated protein will align to multiple residues. In this case, a single aligned residue is chosen such that the total alignment length is maximized.

This technique is agnostic to the details of the actual alignment algorithm. Thus, it could be easily adapted to allow other sequence-order dependent alignment algorithms to detect circular permutations. For instance, CE-CP was used in our recent tool CE-Symm for identifying internally symmetric structures (Myers-Turnbull et al., 2014).

To reduce computational time, CE by default limits gap sizes to 30 residues. Since terminal insertions are common in protein structures, due to both biological variability in tail regions and experimental tags and artifacts, this limit is often restrictive for circularly permuted proteins, and all gaps are considered used in CE-CP by default. This ensures that the optimal path can be found regardless of insertions and deletions.

6.5 Results and Discussion

CE-CP is integrated into the RCSB PDB Comparison Tool, along with several other algorithms for structural alignment (Prlić et al., 2010). Two user interfaces are available: a web version, and a standalone Java application that can be downloaded or run via Java Web Start.

CE-CP results are presented to the user graphically, using the Jmol visualization program, and as a pairwise alignment. The portions before and after the permutation site are displayed in different colors if a permutation is found. The alignment is also available for export in a variety of formats, including a parsable text format or a two-model PDB file containing the two superimposed structures. The standalone application provides additional features, such as the ability to compare custom PDB files and perform full database searches.

As shown in Figure 6.1, CE-CP is able to identify the conserved structural core

between highly divergent structures. It is robust to insertions and deletions, making it suitable for the detection of circular permutations in multi-domain structures.

No large, balanced benchmarks of circularly permuted structures are available. However, CE-CP performance was evaluated on the small but accurate RIPC benchmark (Mayr et al., 2007), as well as compared to results from the semi-automated Circular Permutation Database (CPDB) (Lo et al., 2009).

The RIPC dataset is a small benchmark of "challenging" manual alignments, due to the presence of insertions, conformational variability, and permutations. All 11 pairs of circularly permuted proteins from the dataset were correctly identified by CE-CP, with most residues matching the reference alignment within 0–4 residues.

The CPDB contains 4169 pairs of circularly permuted proteins, as identified by the CPSARST algorithm, followed by manual screening for false positives. Thus, entries contain a plausible circular permutation but are not verified as evolutionarily related. CE-CP identified a circular permutation in 3666 (88%) of CPDB pairs. Of the cases where a permutation was not detected, many are internally pseudosymmetric structures that have reasonable sequential alignments. Since both circularly permuted proteins and internally symmetric proteins can evolve through duplication and fusion mechanisms, the high correlation between the two phenomena should be unsurprising. A portion of these symmetric cases may prove to be false positives from CPSARST given additional evolutionary or functional data.

CE-CP is a readily available and easy to use tool for detecting circular permutations from protein structures. It is incorporated into the RCSB PDB Comparison Tool, which allows the comparison of structures through a variety of methods both on the RCSB PDB website and via a Java Webstart executable. CE-CP is available as part of the BioJava open source project (Prlić et al., 2012).

6.6 Acknowledgements

We would like to thank Guido Capitani for help proofreading the manuscript, and Wei-Cheng Lo and Ping-Chiang Lyu for providing access to the CPDB alignments.

Material from this chapter was originally published as:

> S. E. Bliven, P. E. Bourne, and A. Prlić. Detection of circular permutations within protein structures using CE-CP. *Bioinformatics*, 31(8):1316–1318, Apr. 2015

Author Contributions: I implemented the CE-CP algorithm as a rotation project in 2010. Andreas Prlić assisted with understanding the BioJava CE implementation and correctly implementing the dynamic programming scores in CE-CP. Philip Bourne provided advice and mentoring.

Funding: This work was supported by the National Science Foundation [grant number DBI-1338415], the Intramural Research Program of the National Center for Biotechnology Information, National Library of Medicine, National Institutes of Health [grant number T32GM8806], and the Department of Energy.

6.7 Bibliography

A. Abyzov and V. A. Ilyin. A comprehensive analysis of non-sequential alignments between all protein structures. *BMC Struct Biol*, 7:78, 2007.

S. E. Bliven and A. Prlić. Circular permutation in proteins. *PLoS Comput Biol*, 8(3): e1002445, Mar. 2012.

S. E. Bliven, P. E. Bourne, and A. Prlić. Detection of circular permutations within protein structures using CE-CP. *Bioinformatics*, 31(8):1316–1318, Apr. 2015.

B. A. Cunningham, J. J. Hemperly, T. P. Hopp, and G. M. Edelman. Favin versus concanavalin A: Circularly permuted amino acid sequences. 76(7):3218–3222, July 1979.

J. Dundas, T. A. Binkowski, B. DasGupta, and J. Liang. Topology independent protein structural alignment. *BMC Bioinformatics*, 8:388, 2007.

A. Guerler and E. W. Knapp. Novel protein folds and their nonsequential structural analogs. *Protein Sci*, 17(8):1374–1382, Aug. 2008.

A. Jeltsch. Circular permutations in the molecular evolution of DNA methyltransferases. *J Mol Evol*, 49(1):161–164, July 1999.

J. Jung and B. Lee. Circularly permuted proteins in the protein structure database. *Protein Sci*, 10(9):1881–1886, Sept. 2001.

W.-C. Lo and P.-C. Lyu. CPSARST: an efficient circular permutation search tool applied to the detection of novel protein structural relationships. *Genome Biol*, 9(1):R11, 2008.

W.-C. Lo, C.-C. Lee, C.-Y. Lee, and P.-C. Lyu. CPDB: a database of circular permutation in proteins. *Nucleic Acids Res*, 37(Database issue):D328–32, 2009.

G. Mayr, F. S. Domingues, and P. Lackner. Comparative analysis of protein structure alignments. *BMC Struct Biol*, 7:50, 2007.

D. Myers-Turnbull, S. E. Bliven, P. W. Rose, Z. K. Aziz, P. Youkharibache, P. E. Bourne, and A. Prlić. Systematic Detection of Internal Symmetry in Proteins Using CE-Symm. *J Mol Biol*, 426(11):2255–2268, May 2014.

M. N. Nguyen and M. S. Madhusudhan. Biological insights from topology independent comparison of protein 3D structures. *Nucleic Acids Res*, 39(14):e94, Aug. 2011.

M. Ostermeier. Engineering allosteric protein switches by domain insertion. *Protein Eng Des Sel*, 18(8):359–364, Aug. 2005.

A. Prlić, S. E. Bliven, P. W. Rose, W. F. Bluhm, C. Bizon, A. Godzik, and P. E. Bourne. Pre-calculated protein structure alignments at the RCSB PDB website. *Bioinformatics*, 26(23):2983–2985, Dec. 2010.

A. Prlić, A. Yates, S. E. Bliven, P. W. Rose, J. Jacobsen, P. V. Troshin, M. Chapman, J. Gao, C. H. Koh, S. Foisy, R. Holland, G. Rimša, M. L. Heuer, H. Brandstätter-Müller, P. E. Bourne, and S. Willis. BioJava: an open-source framework for bioinformatics in 2012. *Bioinformatics*, 28(20):2693–2695, Oct. 2012.

I. N. Shindyalov and P. E. Bourne. Protein structure alignment by incremental combinatorial extension (CE) of the optimal path. *Protein Eng*, 11(9):739–747, Sept. 1998.

S. Uliel, A. Fliess, A. Amir, and R. Unger. A simple algorithm for detecting circular permutations in proteins. *Bioinformatics*, 15(11):930–936, Nov. 1999.

J. Weiner and E. Bornberg-Bauer. Evolution of circular permutations in multidomain proteins. *Mol. Biol. Evol.*, 23(4):734–743, Apr. 2006.

T. A. Whitehead, L. M. Bergeron, and D. S. Clark. Tying up the loose ends: circular permutation decreases the proteolytic susceptibility of recombinant proteins. *Protein Eng Des Sel*, 22(10):607–613, Oct. 2009.

Y. Yu and S. Lutz. Circular permutation: a different way to engineer enzyme structure and function. *Trends Biotechnol.*, 29(1):18–25, Jan. 2011.

Chapter 7

Systematic detection of internal symmetry in proteins using CE-Symm

7.1 Forward

Proteins that are composed of multiple copies of a structural motif are said to have internal symmetry. Unlike the quaternary symmetry commonly observed in protein crystals or oligomeric complexes, internal symmetry need not be perfect, but may be pseudosymmetry from structurally similar substructures.

Internal symmetry is thought to evolve via duplications, similarly to the mechanisms for circular permutation discussed in Chapter 5. This suggests that the structural repeats could share a common ancestor. Internal symmetry is thought to often evolve from complexes with quaternary symmetry. Like circular permutations, this process conserves the overall structure despite the significant changes in gene lengths.

The CE-Symm algorithm was developed to detect internal symmetry. It is able to identify the top-scoring alignment between a protein and a rotated copy of itself. The original CE-Symm version was able in some cases to identify the order of symmetry in a protein as well. Other methods for detecting the order are discussed in Chapter 8.

While CE-Symm was originally created to find symmetry in protein structures, it

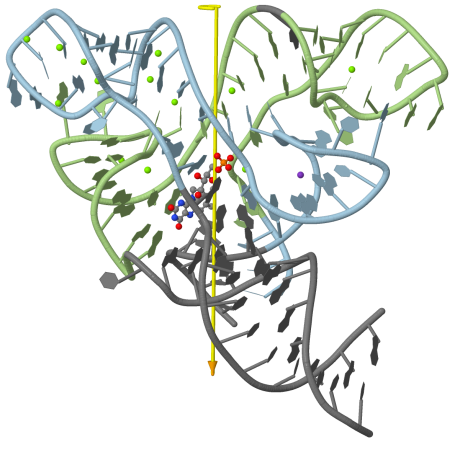


Figure 7.1: CE-Symm alignment of FMN Riboswitch [PDB:3F4E]. The alignment has 52% sequence identity.

can also be run on DNA, RNA, and other polymers. For instance, the FMN riboswitch contains internal symmetry (Jones and Ferré-D'Amaré, 2015). Intriguingly, the flavin molecule is located directly on the axis of symmetry (Figure 7.1).

The remaining text of Chapter 7 is a reprint of the material from:

D. Myers-Turnbull, S. E. Bliven, P. W. Rose, Z. K. Aziz, P. Youkharibache, P. E. Bourne, and A. Prlić. Systematic Detection of Internal Symmetry in Proteins Using CE-Symm. *J Mol Biol*, 426(11):2255–2268, May 2014

7.2 Abstract

Symmetry is an important feature of protein tertiary and quaternary structure that has been associated with protein folding, function, evolution and stability. Its emergence and ensuing prevalence has been attributed to gene duplications, fusion events, and subsequent evolutionary drift in sequence. This process maintains structural similarity and is further supported by this study. To further investigate the question of how internal symmetry evolved, how symmetry and function are related, and the overall frequency of internal symmetry, we developed an algorithm, CE-Symm, to detect pseudo-symmetry within the tertiary structure of protein chains. Using a large manually curated benchmark

of 1007 protein domains, we show that CE-Symm performs significantly better than previous approaches. We use CE-Symm to build a census of symmetry among domain superfamilies in SCOP and note that 18% of all superfamilies are pseudo-symmetric. Our results indicate that more domains are pseudo-symmetric than previously estimated. We establish a number of recurring types of symmetry–function relationships and describe several characteristic cases in detail. Using the Enzyme Commission classification, symmetry was found to be enriched in some enzyme classes but depleted in others. CE-Symm thus provides a methodology for a more complete and detailed study of the role of symmetry in tertiary protein structure.

Availability: CE-Symm can be run from the web at http://source.rcsb.org/jfatcatserver/symmetry.jsp. Source code and software binaries are also available under the GNU Lesser General Public License (v. 2.1) at https://github.com/rcsb/symmetry. An interactive census of domains identified as symmetric by CE-Symm is available from: http://source.rcsb.org/jfatcatserver/scopResults.jsp.

7.3 Introduction

Many proteins have a high degree of symmetry in both their tertiary and quaternary structures. This observation dates back to the determination of the quaternary structure of hemoglobin in 1960 (Perutz et al., 1960), which was discovered to contain symmetric pairs of subunits. Subsequently, symmetry has been found to be important for understanding protein evolution (Lee and Blaber, 2011), DNA binding (Juo et al., 1996; Waldrop, 2011), allosteric regulation (Monod et al., 1965; Changeux and Edelstein, 2005), cooperative enzyme effects (Goodsell and Olson, 2000), and folding (Wolynes, 1996). The relationships between protein symmetry, evolution, and function are reviewed in (Giraldo and Ciruela, 2013; Broom et al., 2012; Matthews et al., 2012; Goodsell and Olson, 2000; Kinoshita et al., 1999).

Symmetry is characterized by an alignment between equivalent substructures.

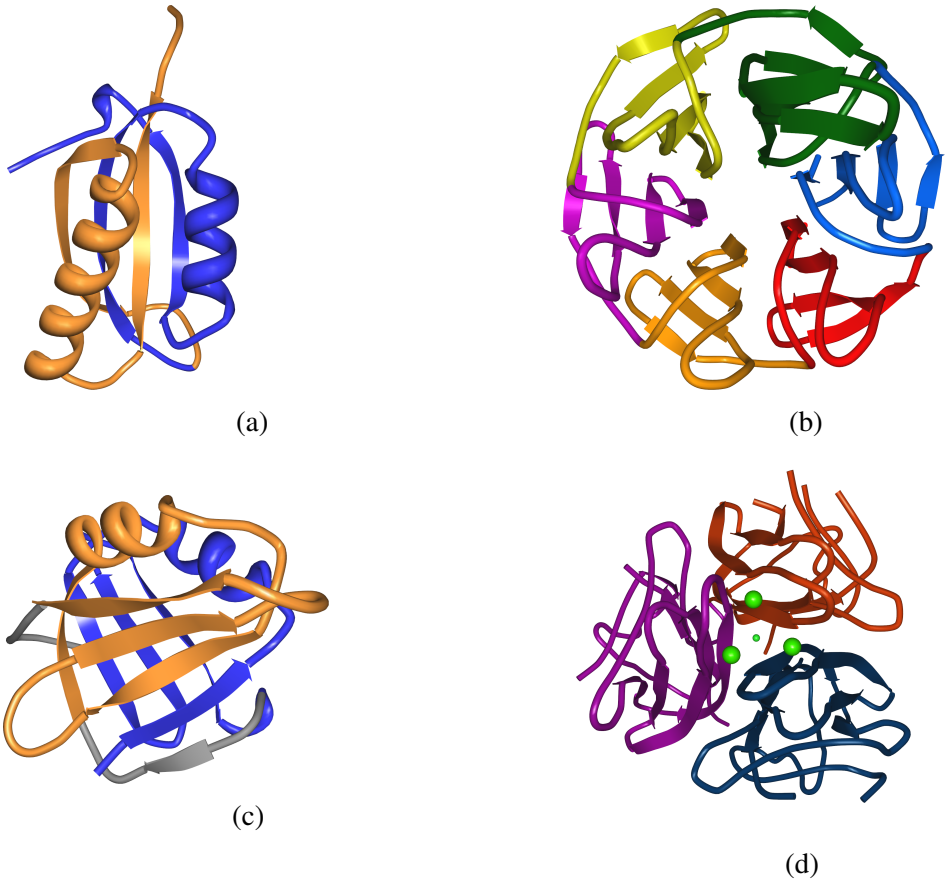


Figure 7.2: Several protein domains with internal symmetry that CE-Symm detects. Coloring is by symmetry unit. (a) A ferredoxin-like fold with two-fold symmetry. (SCOP ID: d2j5aa1) (b) A 6-bladed β propeller. Each blade contains a Kelch sequence motif (Adams et al., 2000), which is also found in some 7-bladed β-propellers (SCOP ID: d1u6dx_) (c) A single DNA clamp domain of a human proliferating cell nuclear antigen (PCNA). The full biological assembly contains 6 of these domains arranged with six-fold symmetry as a trimer of PCNA chains (SCOP ID: d1vyma1) (d) Adiponectins normally assemble into homotrimers of 3 single-domain chains. Shown here (PDB ID: 4DOU) is a designed single-chain three-fold symmetric repeat of an adiponectin globular domain that folds much like an adiponectin trimer (Min et al., 2012). The construct was found to increase insulin sensitivity in mice (Ge et al., 2010).

In the case of quaternary symmetry, these substructures are defined by the inherent equivalence of interactions between identical chains, and often can be determined from the space group of the crystal for X-ray structures. However, this equivalence can be relaxed to allow for evolutionary divergence, revealing pseudo-symmetric arrangements within individual polypeptide chains (internal symmetry) or that span two or more non-identical chains. Figure 7.2 contains examples of proteins with such symmetry within a single chain. This study will focus on internal pseudo-symmetry.

7.3.1 Symmetry and protein evolution

Considering all proteins in the Protein Data Bank (PDB) (Berman et al., 2000; Rose et al., 2012) that contain at least two chains in the annotated biological assembly, we find that approximately 80% of all protein complexes contain quaternary structural symmetry (unpublished, see http://www.rcsb.org). Large symmetric oligomers are thought to have been present in primordial life (Goodsell and Olson, 2000; Koshland, 1976), and symmetry continues to be an important feature of proteins.

One model explaining the evolution of internal symmetry has been described by Andrade et al. (2001) and Abraham et al. (2009). They proposed gene duplication and fusion as a model for the emergence of symmetric protein chains from complexes with quaternary symmetry. These architectures are then subject to evolutionary drift, but their overall symmetric architectures are preserved. An alternative hypothesis, the emergent architecture model, posits that symmetric architectures arise primarily via convergent evolution (Blaber et al., 2012). Most likely both mechanisms are correct for different protein families. Another possible driving force for the evolution of symmetry could be random chance, driven by negative selection against destabilizing mutations (Bershtein et al., 2012).

Well-known cases of symmetry include TIM barrels, β-trefoils, β-propellers, ferredoxin-like proteins, pentein propellers, and immunoglobulin proteins.

TIM barrels consist of eight pairs of alternating α-helices and β-sheets that

interact in parallel to form a cylinder. The TIM barrel fold is extremely versatile and supports a wide diversity of enzymatic reactions (Nagano et al., 2002). Canonical TIM barrels have eight-fold symmetry around the central channel. However, the overall structure is robust to changes in the $(\beta\alpha)_8$ sequence: functional TIM barrels are known with single anti-parallel sheets, with deleted $(\beta\alpha)$ subunits (Grishin, 2001; Sadreyev et al., 2009), and even as a dimer of $(\beta\alpha)_4$ chains (Fortenberry et al., 2011).

The β-trefoil fold has three-fold symmetry and similarly spans a wide range of functions. Several studies have investigated the role of symmetry in β-trefoils by creating β-trefoils with perfect three-fold symmetry (Broom et al., 2012; Lee and Blaber, 2011; Blaber et al., 2012). Both studies found that perfect trimeric β-trefoils are highly stable. One of these constructs—a synthetic glycosidase carbohydrate binding domain—not only retained its function, but was found to have increased binding activity. However, a similar construct of an FGF-1 protein showed none of its normal binding activity. This suggests that exact symmetry improves the function of some proteins, while the normal function of other proteins requires imperfect symmetry.

Adiponectin is a hormone involved in metabolic regulation (Hug and Lodish, 2005) whose normal functioning has been associated with increased insulin sensitivity (Maeda et al., 2002; Shklyaev et al., 2003; Combs et al., 2004; Min et al., 2012). The protein normally assembles as a homotrimer with three-fold crystallographic symmetry (PDB ID: 1C3H). Ge et al. (2010) constructed a single-chain repeat of an Adiponectin globular domain (Figure 7.2d), which folded into a perfectly three-fold symmetric monomer with a structure similar to that of its multimeric counterpart. Expression of the protein construct increased insulin sensitivity in mice and is hoped to be useful in the treatment of diabetes. Given the contribution of symmetry to protein stability, symmetry may become important in protein design, similar to the increased importance of circular permutations (Bliven and Prlić, 2012).

7.3.2 Algorithms that detect symmetry

The examples described in the previous section provide a compelling reason to accurately establish and classify symmetry in protein tertiary structure. Many symmetry-detection algorithms have been developed, including COSEC2 (Mizuguchi and Go, 1995; Kinoshita et al., 1999), DAVROS (Murray et al., 2004), OPAAS (Shih and Hwang, 2004; Shih et al., 2006), Swelfe (Abraham et al., 2008), RQA (Chen et al., 2009), GANGSTA+ (Guerler et al., 2009), and SymD (Kim et al., 2010).

Some of the early methods are based on the alignment of secondary structure elements. These are sensitive to secondary structure assignment, which limits their power to detect some cases of pseudo-symmetry. Moreover, several of these approaches are no longer available. One algorithm, SymD, is still being actively developed. It aligns proteins at the residue level, detecting symmetry by systematically performing a structural alignment for all possible circular permutations of a protein. This results in the determination of protein symmetry, including the detection of multiple axes of symmetry for some cases. Using SymD, Kim et al. (2010) estimated that 10–15% of known protein domains are symmetric.

7.3.3 Symmetry detection using structural alignment

We have previously developed the Combinatorial Extension (CE) algorithm for global three-dimensional protein structure alignment (Shindyalov and Bourne, 1998; Jia et al., 2004) and integrated it into the RCSB PDB as part of the Protein Comparison Tool (Prlić et al., 2010). CE is a well-established protein structure comparison algorithm that has been used in a number of benchmarks as one of the reference methods in terms of alignment accuracy (Mayr et al., 2007; Zhang and Skolnick, 2005; Ye and Godzik, 2003). Here, the intention is to use our experience in performing protein structure alignments using CE and employ it to detect symmetry in protein tertiary structure using a new variation of CE, called CE-Symm.

With several algorithms for the detection of symmetry available, it is surprising

that no reference benchmark to evaluate and compare the quality of these algorithms has been introduced previously. Here we present a manually curated benchmark containing 1007 protein domains.

In the following sections we describe CE-Symm and the benchmark, and we use both to demonstrate that CE-Symm is currently the leading method for the detection of symmetry. Finally, we systematically apply CE-Symm to establish a census of symmetry found in superfamilies as defined by SCOPe 2.03 (formerly SCOP 1.75C) (Fox et al., 2014; Murzin et al., 1995; Andreeva et al., 2008).

7.4 Results

To evaluate the accuracy of CE-Symm and competing methods overall, a total of 1100 SCOP superfamilies from SCOPe 2.01 (SCOP 1.75A)[1] were initially sampled at random, with one domain arbitrarily selected as the representative structure. Sampling superfamilies rather than domains was intended to reduce the effect of bias in the PDB towards easily crystallized or heavily studied proteins. Repeated motifs were classified as cyclic symmetry, dihedral symmetry, linear repeats, helical symmetry, or superhelical. For explanations of these types of symmetry, see Detailed evaluation.

The presence and type of symmetry for each of these domains was determined manually, resulting in a table of SCOP IDs with their corresponding space groups presented in Supplemental File 7.1. When testing algorithms against the benchmark, we considered only cyclic and dihedral symmetry to be cases of symmetry.

7.4.1 Evaluating CE-Symm

CE-Symm performed well on the benchmark set, and fared particularly well at higher thresholds for specificity (fewer false positives). While maintaining a false-

[1]The census (described in the preceding paragraph) originally used SCOPe 2.01 but was updated to SCOPe 2.03 when that version was released. The benchmark was fixed at SCOPe 2.01. No differences at the level of superfamily or higher exist between the two versions.

positive rate (FPR) of just 3.3%, it correctly identified 86% of the symmetric domains in the benchmark set. Among true-positive results, CE-Symm determined the correct order of symmetry 83% of the time. In 96% of cases, it reported either the correct order or an integral multiple or divisor of it.

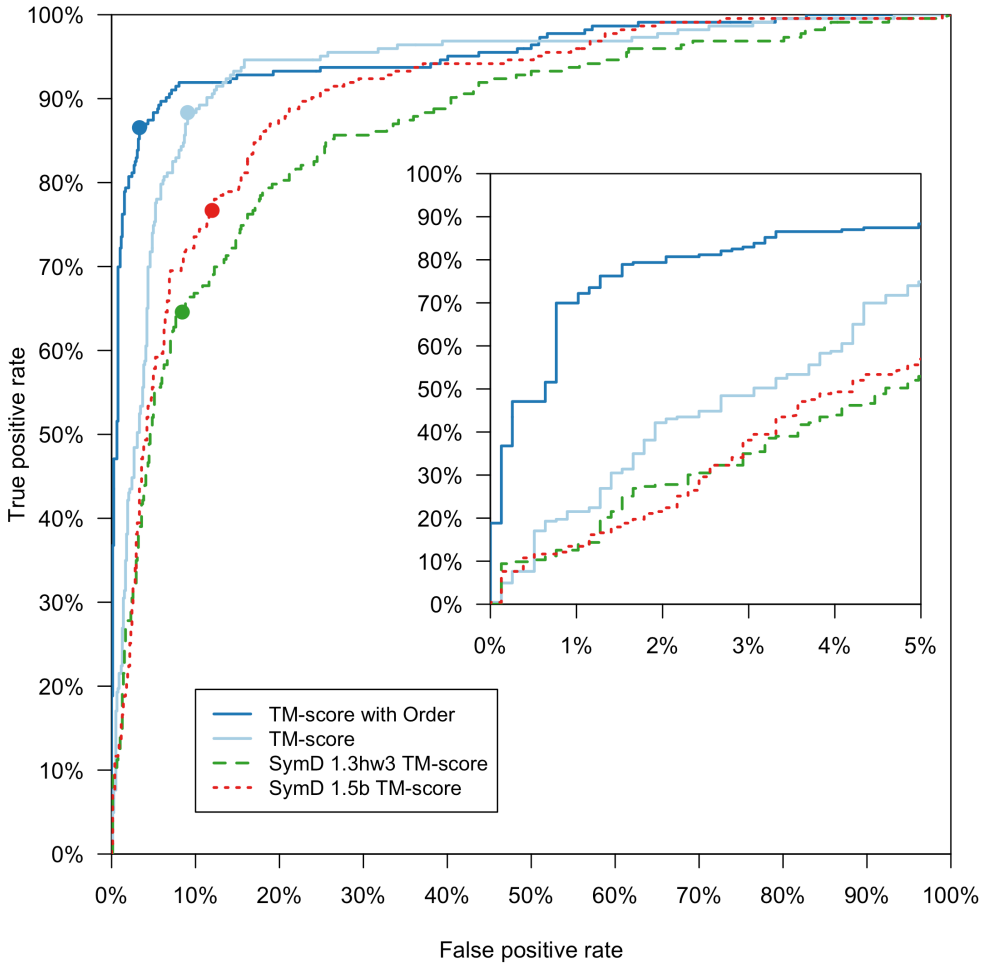


Figure 7.3: Receiver Operating Characteristic curves for CE-Symm and SymD on a benchmark set of 1007 SCOP domains. Two curves for CE-Symm are shown: using only TM-score for scoring (light blue), and using TM-score and the order-method described in Methods (dark blue, solid). Two curves for SymD are shown, one for SymD 1.3hw3 (green), and one for the unpublished version 1.5b (red). The thresholds used for determining symmetry (refer to the footnotes in Table 7.1) are indicated with circles.

To compare CE-Symm against what we considered the best previously available method, we also ran results from SymD (version 1.3hw3) against our benchmark set. Kim et al. (2010) provided us with a copy of an unpublished update to SymD (version 1.5b), which we also benchmarked. For comparison, SymD 1.3hw3 found only 39% of symmetric domains while maintaining the same FPR of 3.3%. The two algorithms are compared in the receiver operating characteristic (ROC) curves shown in Figure 7.3.

The ROC curve for CE-Symm (dark blue) results had an area under curve (AUC) of 0.95, and this value was 0.87 for SymD (version 1.3hw3; orange). The difference between these values was determined to be highly statistically significant (P-value $= 2.2 \times 10^{-5}$) using StAR (Vergara et al., 2008). Therefore, overall CE-Symm performs much better than SymD. We also benchmarked an alternate scoring system for CE-Symm (light blue).

Based on these results, suggested thresholds for the binary decision of symmetric/asymmetric using CE-Symm were established (Table 7.1). Thresholds for SymD are included for reference.

7.4.2 Folds with well-known symmetry

In the interest of continuing the benchmark by Kim et al. (2010), which compared SymD against the secondary structure-based symmetry detection algorithm GANGSTA+, we ran CE-Symm on a set of 8 SCOP folds that are known to be symmetric (Table 7.1). This evaluation is useful to compare CE-Symm with GANGSTA+, and CE-Symm with SymD for selected cases; however, we emphasize that this table contains only a limited and arbitrary choice of folds compared to the more comprehensive benchmark described above. CE-Symm was at least as likely to classify a domain as symmetric than either SymD and GANGSTA+ in 7 of 8 cases. It was 6 times as likely to find symmetry among immunoglobulin-like β-sandwiches than SymD, and 23 times as likely as GANGSTA+ to find symmetry among TIM barrels.

Table 7.1: Folds with known symmetry. Percentage of domains determined to be symmetric according to different decision methods. Data for SymD 1.3hw3 and GANGSTA+, in addition to the list of SCOP domains, is taken from supplemental 3 of Kim et al. (2010). The best-performing methods for each fold are in bold.

ID	**Fold**	**No.**[1]	**CE-Symm (%)**		**SymD (%)**			**GANG (%)**
			Ord [2]	TM [3]	Z_8 [4]	Z_{10} [5]	1.5b [6]	FSAR [7]
d.58	Ferredoxin-like	59	**73**	**73**	19	5.0	43	23
b.1	Immunoglobulin-like	28	**61**	**61**	8.9	0.54	26	8.4
b.42	β-trefoil	8	98	98	**100**	95	98	56
a.24	Four-helical bundle	24	60	**71**	51	25	56	25
d.131	DNA clamp	1	**100**	**100**	91	73	96	64
b.69	7-bladed β-propeller	14	94	**100**	**100**	**100**	**100**	37
c.1	TIM barrel	33	70	**88**	83	69	70	3.7
b.11	γ-Crystallin-like	1	**92**	**92**	75	58	**92**	83

[1] The number of superfamilies in the fold
[2] CE-Symm using TM-score ≥ 0.4 and requiring order ≥ 2
[3] CE-Symm using TM-score ≥ 0.4
[4] SymD using Z-score ≥ 8 (recommended by authors)
[5] SymD using Z-score ≥ 10 (recommended by authors)
[6] The unpublished SymD version 1.5b using TM-score ≥ 0.4
[7] GANGSTA+ using FSAR(fraction of sequentially aligned residues) ≥ 0.8, which the authors recommend (Guerler et al., 2009)

7.4.3 Detailed evaluation

We analyzed a number of cases where CE-Symm determined symmetry correctly but SymD did not, and vice versa. Generally, we found that CE-Symm was more robust to insertions and small structural differences than SymD. For example, CE-Symm correctly identified C2 symmetry in the Ferredoxin-like domain d1r0bl1 and C8 symmetry in the β/α barrel domain d2i5ia1.

One strength of SymD is its superior order-detection capabilities, due to its systematic consideration of all circular permutation points. The order-detection methods used by CE-Symm are useful for eliminating many asymmetric cases and for estimating the order of symmetry. However, the methods are heuristics and sometimes incorrectly report the order, particularly among structures with order greater than 8 or those whose order has no small factors (Supplemental Table 7.S3). The order-detection heuristic can also fail for proteins with variable-length subunits, such as some β-barrels. For example, CE-Symm's order-detection incorrectly reports C1 for the autotransporter domain (SCOP ID: d1uyox_), but CE-Symm is able to correctly classify it as symmetric based on TM-Score alone. A complete listing of predictions on the benchmark set by CE-Symm and SymD is available in Supplemental File 7.2.

CE-Symm and SymD were found to have comparable computation times. Both SymD 1.3hw3 and CE-Symm with order-detection completed in about 2 seconds per domain when run on the benchmark set in a single-threaded environment on a 64-bit Mac OS system with a 2.8Ghz Intel Core i7 processor and 16GB RAM. On the same system, SymD 1.5b required about 4 seconds per domain; however, we note that this version has not been released publicly.

7.4.4 Symmetry order

The types of symmetry identified in the benchmark set are given in Table 7.2). We found that 23.9% of the superfamilies sampled contained some form of structural repeat. Of these, cyclic symmetry was by far the most common (91.3%). Two-fold symmetry

Table 7.2: Benchmark symmetry by order. Types of symmetry found in the benchmark.

Order	Superfamilies		Example Folds
Asymmetric			
	766	76.1%	
Rotational			
2	166	16.5%	Immunoglobulin-like, Ferredoxin-like, Rossmann, γ-crystallin-like, DNA clamp, up-down 4-helical bundle
3	10	1.0%	β-Trefoil, β-Prism, Flavodoxin-like
4	2	0.2%	4-bladed β-propeller, Streptavidin-like, Prealbumin-like, OMPA-like
5	3	0.3%	5-bladed β-propeller, Pentein β/α-propeller, PT-barrel
6	9	0.9%	6-hairpin glycosidases, 6-bladed β-propeller, auto-transporter
7	9	0.9%	7-bladed β-propeller, 7-bladed α/α-toroid, 7-hairpin glycosidase
8	21	2.1%	TIM barrel, 8-bladed β-propeller
Dihedral			
2	2	0.2%	Transmembrane β-barrels, Streptavidin-like
4	1	0.1%	Streptavidin-like
Helical			
2	9	0.9%	Leucine-rich repeat, β-helix, α-α superhelix
3	2	0.2%	α-α superhelix, β-helix
Non-integral	2	0.2%	α-α superhelix, Triple-stranded β-helix
Superhelical	2	0.2%	α-α superhelix
Translational			
	3	0.3%	Ankyrin repeat, β-helix, Bacteriochlorophyll A protein

was the most common type of cyclic symmetry (75.5%), followed by eight-fold cyclic symmetry.

Dihedral symmetry, helical symmetry, and translational repeats accounted for the remainder, about 2.1%. Linear repeats have translational symmetry, which is given by the repeated application of a translation but no rotation. In most helically symmetric structures, rotating by $360^\circ/k$ for some integer k is equivalent to no rotation; such a structure is said to have helical symmetry of order k. For some structures, no such integer exists; we labeled this type of symmetry "non-integral helical". Superhelical symmetry is the unusual symmetry seen in domains such as in Leucine-rich repeats.

7.4.5 A census of symmetry in SCOP

A census of symmetry in the tertiary structure of domains was created by running CE-Symm on every domain in each superfamily in SCOPe 2.03 Fox et al. (2014); Murzin et al. (1995). This version of SCOP is an update by John-Marc Chandonia, Naomi K. Fox, and Steven E. Brenner; it is available at http://scop.berkeley.edu.

SCOPe 2.03 contains 1766 superfamilies over 5 main classes: all-α, all-β, α/β, $\alpha+\beta$, and trans-membrane. We constructed a census of symmetry over these superfamilies by running CE-Symm (with order detection enabled) on every domain in each superfamily and normalizing by the number of domains per superfamily. We found that 18.0% of these superfamilies are symmetric. This percentage of symmetric superfamilies is slightly higher than the percentage of symmetric domains in SCOP among ASTRAL 40 representatives (Chandonia et al., 2004) found by SymD, which was 10–15% (Kim et al., 2010). Figure 7.2 shows some examples of symmetric proteins identified by CE-Symm.

Interestingly, symmetric $\alpha+\beta$ superfamilies are disproportionately rare (Table 7.3). $\alpha+\beta$ folds consist of α and β regions that are physically separated in sequence; we hypothesize that this separation limits the number of viable symmetric architectures. In contrast, all-β proteins are enriched for symmetry. This class contains a number of common symmetric folds, such as β-barrels and β-propellers. The extended hydrogen-

Table 7.3: Symmetry by SCOP class. Percentage of superfamilies identified as symmetric by CE-Symm. Note that, to maintain a low false-discovery rate, CE-Symm underestimates the number of symmetric superfamilies in SCOP by about 27% (see Figure 7.3).

Class	Total Number	% Symmetric
α	507	18.5%
β	354	24.6%
α/β	244	16.8%
α+β	551	14.3%
Multi-domain[1]	66	4.5%
Membrane	109	23.8%
Overall	1831	18.0%

[1] These are large protein chains that have only been observed in their entirety.

bonding networks in β-sheets may also contribute to this enrichment, as planar structures are inherently more likely to be symmetric due to their reduced dimensionality.

Symmetry is also disproportionately frequent among membrane superfamilies, in agreement with previous observations (Klingenberg, 1981; Choi et al., 2008). Membrane proteins often contain additional quaternary symmetry in addition to the internal symmetry within individual domains. The axis of symmetry is typically perpendicular to the membrane plane, although some cases are known with the axis of symmetry parallel to the plane (Goodsell and Olson, 2000). The symmetric arrangement of subunits in membrane proteins minimizes the lipid interface for each subunit, and the gap formed at the axis of symmetry often forms the channel for membrane transporters.

7.4.6 Sequence conservation

Using all superfamilies in the census, we calculated the percentage identity of the alignment given by CE-Symm. In the case of self-alignments given by CE-Symm, the percentage identity is defined as the percentage of amino acids that are conserved when the domain is superimposed on itself following a rotation about the axis of symmetry. Percent identity was graphed separately for symmetric and asymmetric superfamilies

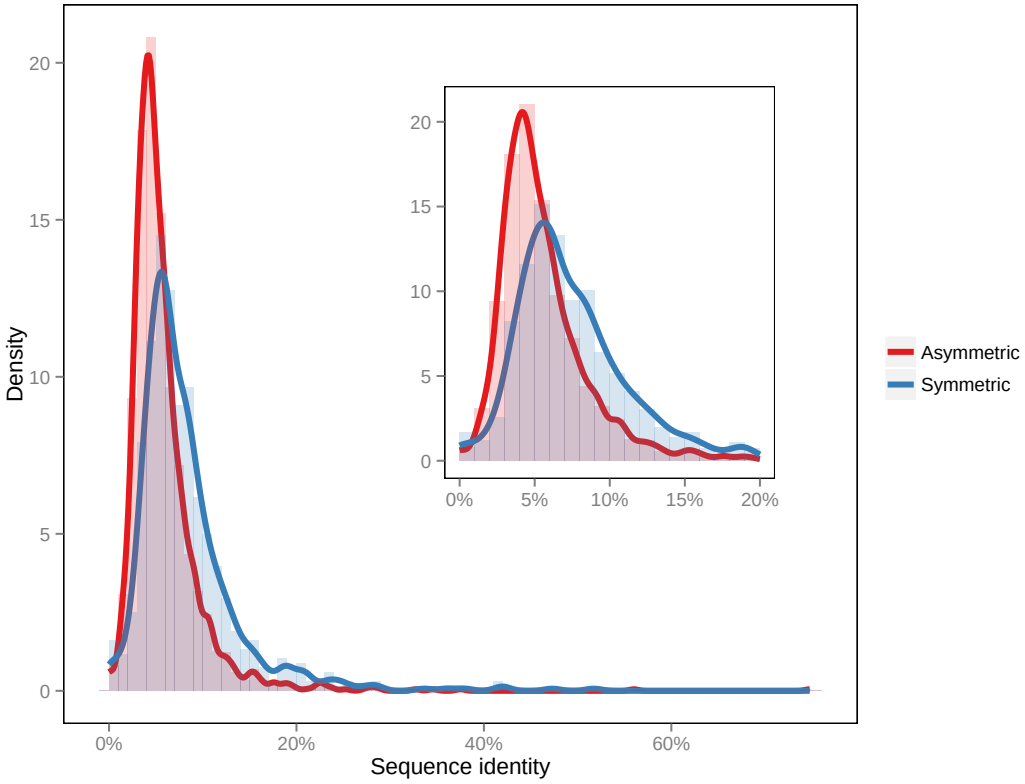


Figure 7.4: Sequence identity between symmetry units. Distribution of sequence identity between aligned subunits for symmetric superfamilies (blue). For comparison, the distribution of percentage identity among asymmetric superfamilies (red). Most CE-Symm alignments of asymmetric proteins represent random alignments, although a few examples contain translational repeats or helical symmetry.

(Figure 7.4).

Surprisingly, the distributions in Figure 7.4 are very similar. Indeed, the mean %id among symmetric results is 8.2%, not substantially higher than the mean %id among asymmetric results, 5.8%. Moreover, there are few symmetric domains with greater than 16 %id. Considering amino acid similarity rather than identity produces similar results (see Supplemental Figure 7.S1). This lack of sequence conservation between the structural units that give rise to the symmetry (*symmetry units*) could indicate (a) that the majority of internally symmetric superfamilies arose following ancient duplication events, (b) that convergent evolution between subunits is a more significant contributor

to internally symmetric proteins than previously thought, or (c) that the relationship between sequence and structural motifs is relatively flexible, making it difficult to detect sequence similarities based on structure-based methods such as CE-Symm. A similar observation has also been made by Wright et al. (2005), where a low sequence identity between proteins might be associated with the inhibition of misfolding and aggregation of proteins in the crowded environment of a living cell.

7.4.7 Enzyme function

To investigate the relationship between symmetry and protein function, we grouped symmetric superfamilies by their Enzyme Commission (EC) numbers (Bairoch, 2000)). Consistent with our methodology for the census, we normalized by the number of domains per superfamily to mitigate bias in the PDB. A superfamily was assigned an EC number if it contained a domain having that EC number, meaning that multiple enzyme classes can be assigned to a single superfamily.

Analysis of the top-level EC classes proved difficult due to the breadth of structures which provide scaffolds for each type of reaction. Isomerases were enriched for internal symmetry (24% symmetric), while oxidoreductases and ligases contained fewer symmetric domains than average (each 15%; see Supplemental Figure 7.S2). Oxidoreductases span a broad range of evolutionarily and structurally disparate folds (148 in the analysis), and the distribution of folds and the distribution of superfamilies over these folds are both diffuse. Therefore, the low level of symmetry cannot be ascribed to the class having a constrained set of viable folds.

Considering second-level EC subclasses allows the relationship between symmetry and function to be more clearly established. The number of symmetric superfamilies for selected EC subclasses is given in Table 7.4 and is fully detailed in Supplemental Table 7.S1. Although the number of superfamilies annotated with each subclass is fairly small, enrichment for symmetry also could not be explained by a lack of structural diversity in enzymes with each function.

Table 7.4: Percentage of superfamilies found to be symmetric for selected second-level Enzyme Commission numbers. The most and least symmetric 5 EC subclasses containing at least 20 superfamilies are shown. See Table 7.S1 for the complete list.

EC	Description	%S[1]	NSf[2]
5.1	Isomerases: racemases and epimerases	38	21
5.3	Isomerases: intramolecular oxidoreductases	26	34
4.1	Lyases: carbon–carbon lyases	26	57
2.5	Transferases: transferring alkyl or aryl groups, other than methyl groups	23	31
3.4	Hydrolases: acting on peptide bonds (peptide hydrolases)	21	95
6.3	Ligases: forming carbon–nitrogen bonds	11	74
1.8	Oxidoreductases: acting on a sulfur group of donors	10	29
4.2	Lyases: carbon–oxygen lyases	10	79
1.10	Oxidoreductases: acting on diphenols and related substances as donors	10	20
1.4	Oxidoreductases: acting on the CH-NH(2) group of donors	8.3	24

[1] Percentage of superfamilies that are symmetric
[2] The number of superfamilies

One of the most enriched subclasses for symmetry is that of the racemases and epimerases (EC 5.1). While perfect symmetry would be unexpected in racemase active sites based on their need to bind multiple sterioisomers equally well, pseudo-symmetric scaffolds may be amenable to these types of function (Whitman et al., 1985). Many racemases exhibit quaternary symmetry, in addition to the internal symmetry considered for the census. Several oxidoreductase subclasses are significantly below average for symmetry. Oxidoreductases often contain multiple cofactors for electron transport, which may be less easily supported by symmetric protein scaffolds. Thus, certain enzymatic reactions may support or preclude symmetry.

7.5 Discussion

To further investigate potential relationships between symmetry and protein function, we analyzed a large number of proteins to ascertain their symmetry–function relationships. Based on this analysis, we identified recurring types of symmetry–function

relationships.

7.5.1 Symmetry around ligand-binding sites

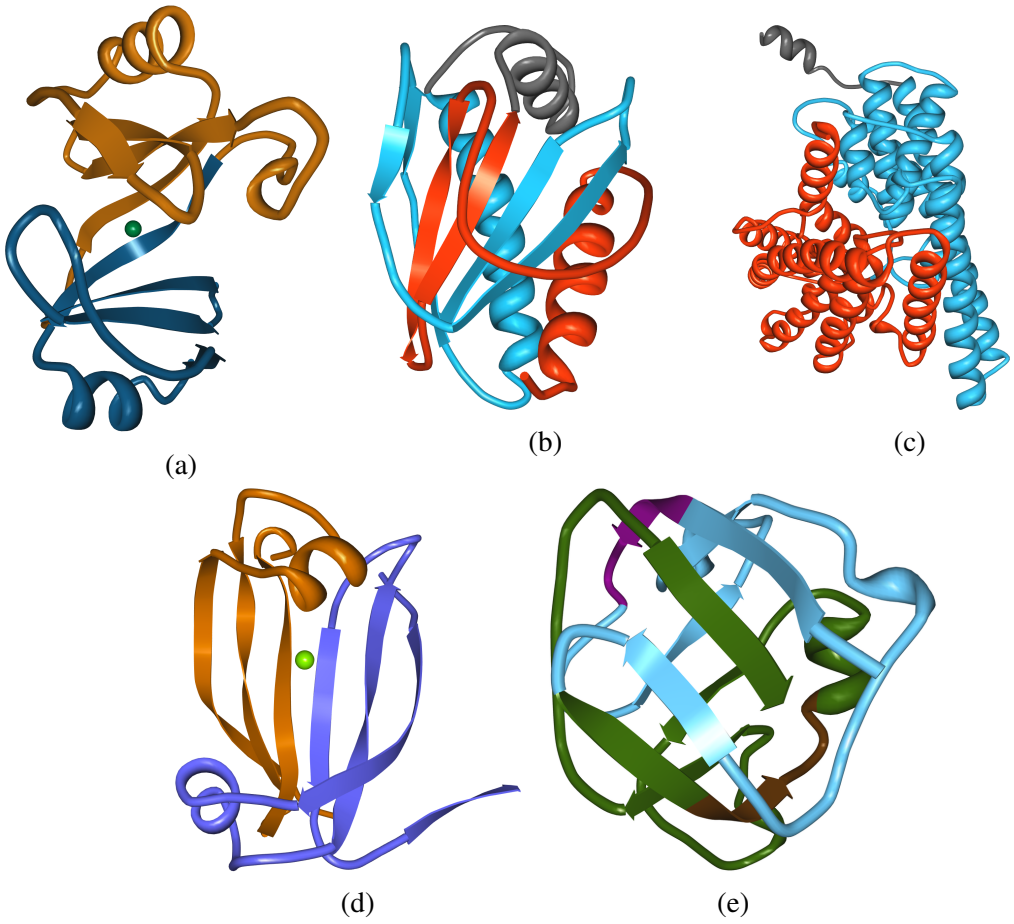


Figure 7.5: Examples of proteins with symmetry and function relationships. (a) Glyoxalase I contains a duplication around the nickel-binding active site (PDB ID: 3HDP). (b) CheX protein contains two identical active sites (PDB ID: 1SQU). (c) CLC-ec1 chloride carrier, where ions are thought to flow along its symmetric interface (PDB ID: 2FEE). (d) A chorismate lyase-like protein with a two-fold symmetry that is not clearly related to its little-understood function (PDB ID: 3DDV). (e) PTSIIA/GutA-like domain (PDB ID: 2F9H). Both subdomains of the symmetry contain the same 8-amino-acid sequence (residues 9–16, shown in purple and 67-74, shown in brown).

Symmetry around ligand-binding sites is the most basic symmetry–function relationship. For example, glyoxalase I (Figure 7.5a) is a two-fold symmetric protein with a metal-binding site at its center. (Bergdoll et al., 1998). Searching systematically

in our census and counting only one domain per superfamily, we found that 22% of symmetric, ligand-containing domains contained a ligand within 5Å from the centroid of the domain. Unaligned residues, such as insertions, were excluded from the calculation of the centroid.

7.5.2 Function along symmetric interfaces

Many symmetric proteins have function at the interface between symmetry units, the repeated structural units that describe the symmetry. This differs from symmetry around a ligand–binding site, described above, in that the functional site can occur anywhere along the axis of symmetry. An example of this relationship is the chloride channel, in which the symmetric interface between the two symmetry units forms a gate at the core of the channel (Dutzler et al., 2003). Interestingly, the chloride channel is thought to be moderately rigid compared to other channels, such as potassium ion channels or bacterial leucine transporters, both of which are activated by the rotation of subunits relative to each other (Dutzler et al., 2003; Forrest and Rudnick, 2009). Currently, it seems that only the movement of one side chain at the core of the gate is responsible for letting Cl^- ions pass. Using the same systematic, preliminary analysis we applied to find ligands near the centroids of domains, we found that 63% of symmetric, ligand-containing domains contained a ligand within 5Å from the axis of symmetry. This number was 37% within a mere 1Å of the axis.

7.5.3 Duplication of ligand-binding sites

Duplication of ligand-binding sites is another common feature of symmetric proteins. For example, it occurs in the chemotaxis protein CheC (Figure 7.5b), which is a globular α/β protein that functions in bacterial chemotaxis and is involved in flagella movement. The protein is two-fold symmetric. Each of the two units of symmetry contains a dephosphorylation center comprising asparagine and glutamate residues. Gene duplication followed by domain swapping has been proposed as an evolutionary process

for the emergence of CheC (Park et al., 2004).

7.5.4 Unknown functions

Besides examples such as those listed above, there are many symmetric domains with no obvious relationship between their symmetry and their function. The chorismate lyase-like protein (Figure 7.5d) consists of a two-fold internally symmetric domain. Its biologically active form is a dimer such as PhnF from *E. coli* (PDB ID: 2FA1) or YurK from *B. subtilis* (PDB ID: 2IKK).

7.5.5 Conserved sequence motifs

In some cases we can identify conserved sequence motifs shared between symmetry units. The PTSIIA/GutA-like domain is an antiparallel β-barrel fold with highly conserved two-fold symmetry. The overall sequence identity of this symmetry is 16%. Little is known about this protein structure since it is a novel fold and does not have an associated publication. Similarly, not much is known about its sequence, with Uniprot only listing a manuscript that describes the larger genomic region covering the gene encoding this structure. However, by investigating the symmetric alignment, we can identify a motif that corresponds to equivalent residues in the structure and that is observable in the Pfam domain (PF03829) (Punta et al., 2012), which contains a conserved [IV]XX[IV]GXX[VA] motif at the corresponding positions (Figure 7.5e). Sequence homology between the subunits can be established using the protein sequence alone. However, the analysis of symmetry reveals structural homology and shows that the two types of homology correspond. Based on this correspondence, we postulate that these residues are important functionally, and that they can serve as a guide for further experimental analysis.

7.5.6 Relationship between tertiary and quaternary symmetry

We also suggest that there is a relationship between symmetry of proteins and their biological assemblies. It has been speculated that this can be related to mono- and oligomerization events during evolution that keep the biologically active assembly essentially unmodified (Abraham et al., 2009). We can confirm this finding and identify several domains with complex relationships between symmetry in the biological assembly and internal symmetry in tertiary structure. An example of this is the DNA clamp. In eukaryotes (PDB ID: 1VYM), it exists as a three-chain symmetric biological assembly. Each chain consists of two protein domains, which in turn have two-fold symmetry (Figure 7.2c). Thus, the overall assembly has six-fold pseudo-symmetry. The overall symmetry is highly conserved in the bacterial DNA clamp, which has only two chains in the biological assembly, but with each chain consisting of three internally symmetric domains (PDB ID: 1MMI; Kelman and O'Donnell (1995)).

Another example with an interesting relationship between the biological assembly and internal pseudo-symmetry is the vitamin B12 transporter BtuCD-F (PDB ID: 4FI3; Korkhov et al. (2012)). It consists of three components: BtuC, BtuD, and BtuF. BtuC and BtuD are present as a dimer and bound to BtuF, which is a monomer in the biological assembly. However, BtuF has has internal pseudo-symmetry, giving the whole complex pseudo-twofold symmetry. For a classification of symmetry in structural complexes of proteins see (Levy et al., 2006).

7.5.7 Types of symmetry CE-Symm identifies

The modifications described in the Methods section enable CE-Symm to detect rotational pseudo-symmetry within protein backbones. It can also detect non-rotational repeats, such as linear repeats, helical proteins, and β-helices. Rotational symmetry can be easily distinguished from other repeats using geometric criteria (see Materials and Methods).

Because CE-Symm uses dynamic programming, it is limited to finding alignments

that contain at most a single circular permutation. Types of symmetry that contain more than one axis of symmetry (dihedral, tetrahedral, octahedral, or icosahedral) require multiple changes in sequence topology to align. In such cases, CE-Symm typically will identify one axis of rotation, though additional axes may be found by rerunning CE-Symm on just one of the symmetric domains identified by the first run.

CE-Symm is also limited to returning the single highest-scoring alignment. This may not correspond to the smallest rotational symmetry present in the protein. For instance, in proteins with four-fold pseudo-symmetry, the alignment corresponding to the 180° rotation may score higher than the 90° or 270° alignments. This sometimes leads to the protein being identified as containing two-fold pseudo-symmetry, which incompletely describes the relationships within the protein. More broadly, accurate detection of order of symmetry is a current limitation in CE-Symm which we expect to rectify in a future version.

7.6 Conclusions

In this study we introduced a new method for determining pseudo-symmetry in protein structure and used it to build a census of symmetry over domains in SCOP. We also established a reliable benchmark set containing SCOP domains for which both presence and type of symmetry was determined manually. We used this benchmark set to compare our algorithm and previously published symmetry-detection algorithms and demonstrated that our algorithm is more suitable than other methods for detecting symmetry at high specificity. The benchmark set can be used to verify the accuracy of results from other methods for symmetry detection or classification.

By systematically applying CE-Symm on many protein domains we found that more proteins contain internal symmetry than previously estimated. The symmetry of most domains lacks any sequence signal that CE-Symm readily detects. However, clear sequence signals were found for certain folds, such as β-propellers (Chaudhuri et al., 2008).

We also found symmetry to be more associated with some types of enzymatic activity than with others and suggest that certain enzymatic functions preclude or hinder symmetry. We note that in several cases there is a clear relationship between protein symmetry and function, which may explain why certain domains are symmetric.

The analysis of symmetry and pseudo-symmetry in protein structures leads to a deeper understanding of protein function and evolution. Besides detecting pseudo-symmetry in protein structures, CE-Symm allows also the detection of conserved sequence motifs in symmetry units. This can provide insight useful for further analysis of a protein. This is particularly important if the function or active sites of the protein are unknown.

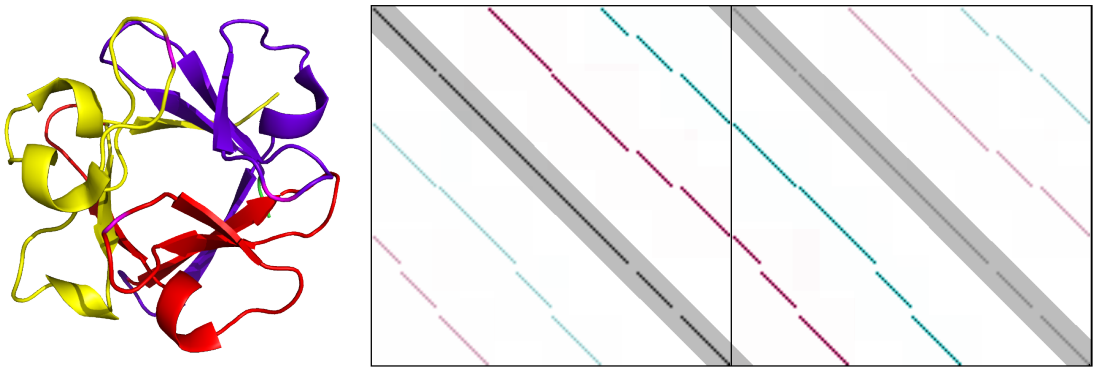
7.7 Materials and Methods

7.7.1 CE-Symm Algorithm

The Combinatorial Extension (CE) algorithm operates by using a geometric distance score to evaluate the local structural similarity between two proteins around each residue (Shindyalov and Bourne, 1998). Dynamic programming is used to identify high-scoring paths in the dynamic programming matrix, corresponding to regions of local structural similarity. An iterative algorithm then heuristically combines local fragments to identify a high-scoring global superposition of the two proteins.

Building on the CE concept, and to identify self-similar regions within a protein, CE-Symm compares a protein structure to itself. It runs CE to compare two copies of the input protein, with the following modifications:

1. **Prohibit alignments near the diagonal.** To prevent the algorithm from finding trivial identity similarity, the distance score between residues less than δ residues apart was defined as infinity, preventing the optimal path from traversing the region near the diagonal in the dynamic programming matrix (black line in Figure 7.6). $\delta = 8$ performed well in practice.



(a) (b)

Figure 7.6: Self-similarity in FGF-1, a three-fold symmetric protein. (a) 3D structure of FGF-1 (PDB ID: 3JUT), colored to highlight the three analogous portions of the protein. (b) Dot plot showing corresponding residues within the single chain. Three alignments are possible, corresponding to rotations of 0° (black), 120° (magenta), and 240° (cyan)

2. **Allow circular permutations.** When comparing a protein to a rotated copy of itself, the aligned sequence of the rotated copy will appear to be circularly permuted relative to the original protein. This can be seen in Figure 7.6b as discontinuities in the magenta and cyan alignments. To detect circular permutations we apply an approach similar to Uliel et al. (1999). The dynamic programming matrix is duplicated in one direction (see Figure 7.6) and CE is run normally. This allows the full length of a symmetric protein to be aligned. The results are then post-processed to map the alignment back onto the single protein. While it is possible with this technique that single residues may be aligned twice, this is rare in practice. In cases where it does occur, alignment length is used as a heuristic to choose which residues to include in the final alignment.

7.7.2 Identifying symmetry order

CE-Symm identifies self-similar structures within a protein. Rotational symmetry is the most abundant form of structural repeat, but linear repeats with high self-similarity can also be found (concentric turns of β-helices, for example). To filter out such cases,

we developed an algorithm to estimate the symmetry order of a self-alignment. Proteins with order 1 (no rotational symmetry) were removed from the results.

The algorithm considers a self-alignment to be a function from the set of residues in a protein to itself. We say that $f(x) = y$ if CE-Symm aligned residues x and y. If CE-Symm identified rotational symmetry within the protein, then the repeated function composition $f^k(x)$ corresponds to repeated rotations. When the function is applied a number of times equal to the order of the underlying CE-Symm alignment, k^*, then $f^{k^*}(x) \approx x$, corresponding to a rotation by 360°. To identify the order of a self-alignment, successively larger values of k are tried and the root mean squared deviation (RMSD) found for each according to the formula:

$$RMSD = \sqrt{\sum_i \left(f^k(x_i) - x_i\right)^2}$$

The correct order is determined by identifying large decreases in RMSD. In practice, a threshold of 40% decreases was found to correctly identify the order in most cases. If no such drops are identified for k of 8 or less, an order of 1 (no rotational symmetry) is assumed.

We also employed a secondary method to determine order based on the angle between aligned subunits. The rotation axis and angle of rotation is first calculated based on the procedure in Kim et al. (2010). We then compare the angle of rotation, θ, to the ideal angles for proteins with low orders of rotational symmetry.

$$\varepsilon(\theta) = \min_{2 \le k \le 8} \left| \frac{2\pi}{k} - \theta \right|$$

If this angle is below a threshold, τ, we label the protein as symmetric with order k. For this study we used a stringent threshold of $\tau = 1°$.

Initial tests found two methods to be complementary. Method 1 is more robust to geometrical distortions, while method 2 is more robust to inaccuracies in the alignment. Thus, proteins were classified as symmetric if either method determined the symmetry

order to be greater than one.

7.7.3 Scoring schemes

Several alternate scoring schemes were considered, both for optimizing the alignment and for detecting the presence of symmetry. By default, the CE scoring scheme is used to judge the quality of alignments (Shindyalov and Bourne, 1998). This is a purely structural scoring which attempts to maximize the alignment length while maintaining a low RMSD. We also implemented an alternate score that incorporates sequence similarity in addition to the structural alignment. Sequence similarity is quantified using the structure-derived substitution matrix (Prlić et al., 2000), which is optimized for the alignment of distantly related proteins. The relative weight of structure and sequence scores can be adjusted with a configuration parameter.

A number of features were considered for classifying proteins as either rotationally symmetric or asymmetric, including RMSD, TM-score, Z-score (as reported by CE), alignment length, and sequence identity. Of these, the TM-score gave the best performance on the ROC curves. A variant of TM-score that incorporates order information was also evaluated, in which 1.0 was added to the TM-score if either method for determining symmetry order determined an order of symmetry greater than 1. This ensures that rotationally symmetric structures always have scores strictly greater than asymmetric ones, reducing false positives especially from helical symmetry and translational repeats. To classify the structure as symmetry or asymmetric, a threshold of ≥ 1.4 is applied to the sum. This last method yielded the best performance and is recommended by the authors.

7.8 Supplemental Methods

7.8.1 Scoring Functions

To compare CE-Symm with competing algorithms, we also ran predictions by SymD against the benchmark set. We wanted to compare the two algorithms using their respective best scoring methods. Therefore, for SymD we considered TM-score as well as T-score and Z-score, two scores reported by SymD. T-score is similar to TM-score in the sense that both are essentially length-normalized RMSD. It is defined (Kim et al., 2010) by:

$$T = \sum_{ij:|i-j|>3} \frac{1}{1 + d_{ij}/d_0^2}$$

where $d_0 = 2.0$Å is a normalization constant. The Z-score is then defined over a distribution of T-scores as:

$$Z = \frac{T - \overline{T}(N)}{\sigma(N)}$$

It is important to note that the moments $\overline{T}(N)$ and $\sigma(N)$ are both functions of the structure length N. We found TM-score to be the best scoring method for both SymD 1.3hw3 and the unpublished update (version 1.5b). Interestingly, Z-score performed almost equally well as TM-score for SymD 1.3hw3, but performed quite poorly for 1.5b. For SymD 1.5b, we used the TM-score that version calculates; however 1.3hw3 does not output a TM-score, so we recalculated this from the FASTA alignment. We note that, in 52 cases of the 1007 domains on the benchmark, there was a non-negligible (≥ 0.001) difference between our recalculated TM-score and the TM-score that SymD 1.5b provided. Although we were unable to determine a precise explanation for this discrepancy, we believe that our recalculation correctly matches the definition by Zhang and Skolnick (2004). Of the differing cases we examined, the alignment by SymD uses the trivial alignment (that is, aligns residues to themselves) entirely or in part.

7.8.2 Symmetry and order of symmetry for superfamilies

Because superfamilies can contain many domains, the determination of symmetry for superfamilies is ambiguous.

We deemed a superfamily (or fold) to be symmetric if the mean TM-score over the domains within that superfamily (or fold) was at least 0.4, and an order of symmetry greater than 1 was identified in at least 50% of domains.

7.8.3 Symmetric folds benchmark

For each fold in Table 7.1, we generated a list of SCOP domains by taking the intersection of SCOP 1.73 and ASTRAL40 1.73, as done in Kim et al. (2010). However, we found 5 Ferredoxin-like domains and 2 4-helical bundles that Kim et al. (2010) did not: d1aopa1, d1fxda_, d2bv3a4, d1zhva1, d2fdna_, d1okkd1, and d1st6a6.

7.9 Supplemental Figures

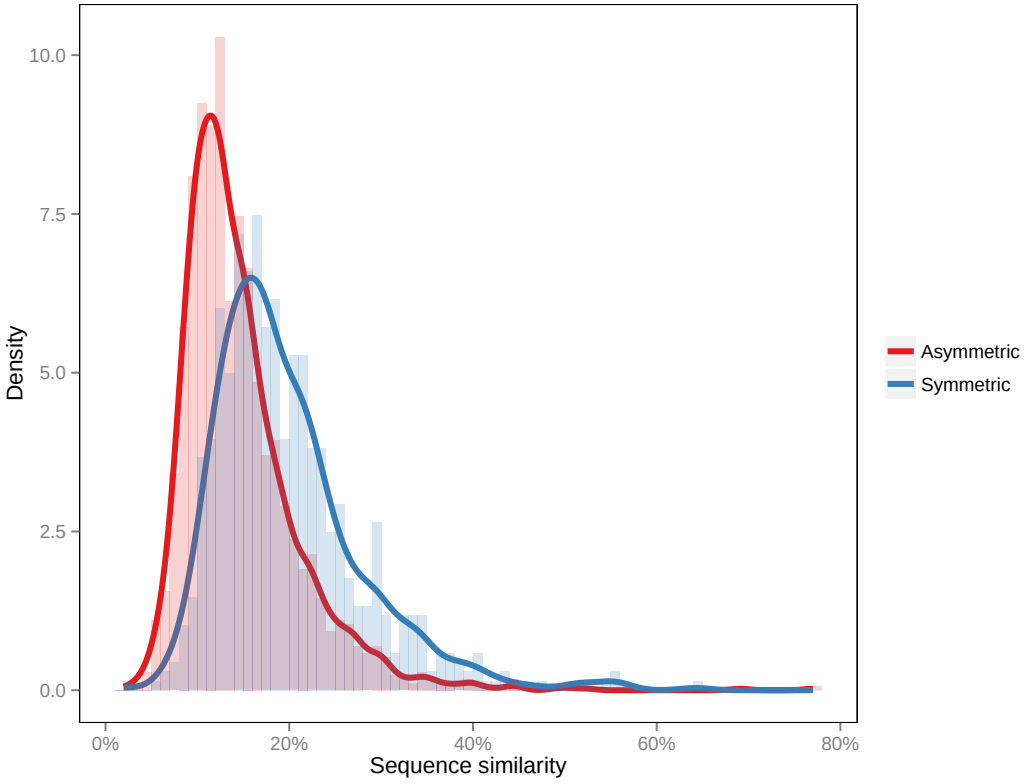


Figure 7.S1: Distribution of sequence similarity among symmetric (blue) and asymmetric (red) SCOP superfamilies. The two curves are normalized to have the same area.

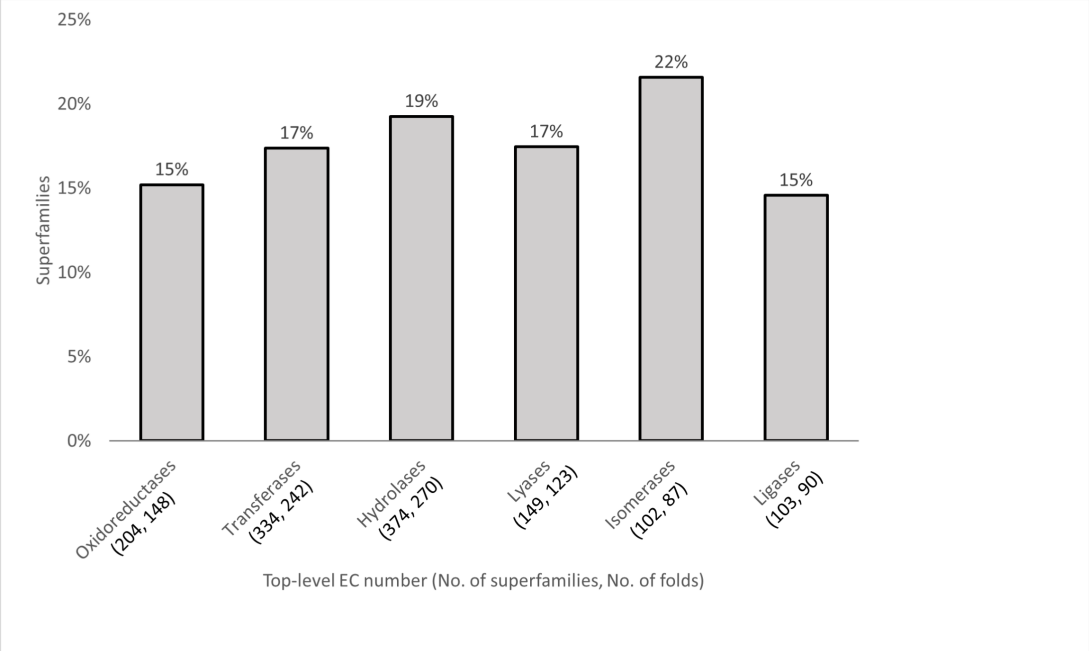


Figure 7.S2: Percentage of superfamilies that are symmetric by top-level Enzyme Commission Numbers (Webb and IUBMB, 1992). For each EC number class, the number of SCOP folds and the number of superfamilies are given in parenthesis. Note that some superfamilies are included in more than one category.

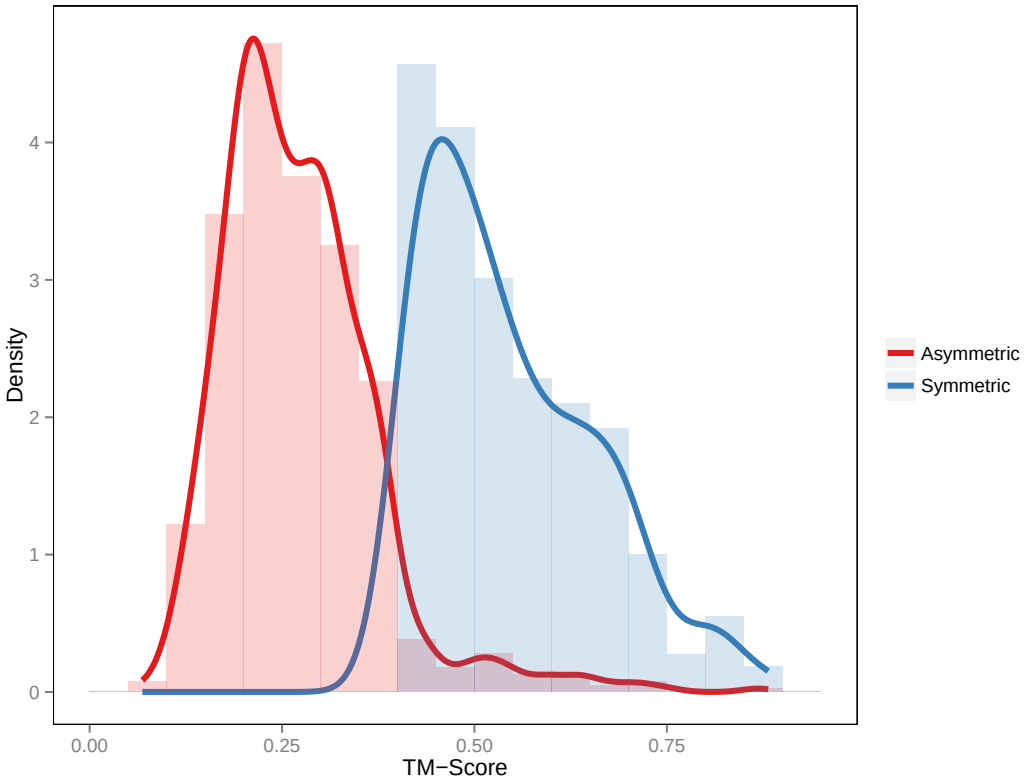


Figure 7.S3: TM-score among symmetric (solid blue) and asymmetric (dashed black) domains in the benchmark set. Although the distribution among symmetric domains is significantly right-shifted, an arbitrary result with TM-score of 0.5 is as likely to be asymmetric as symmetric. The two curves are normalized to have the same area.

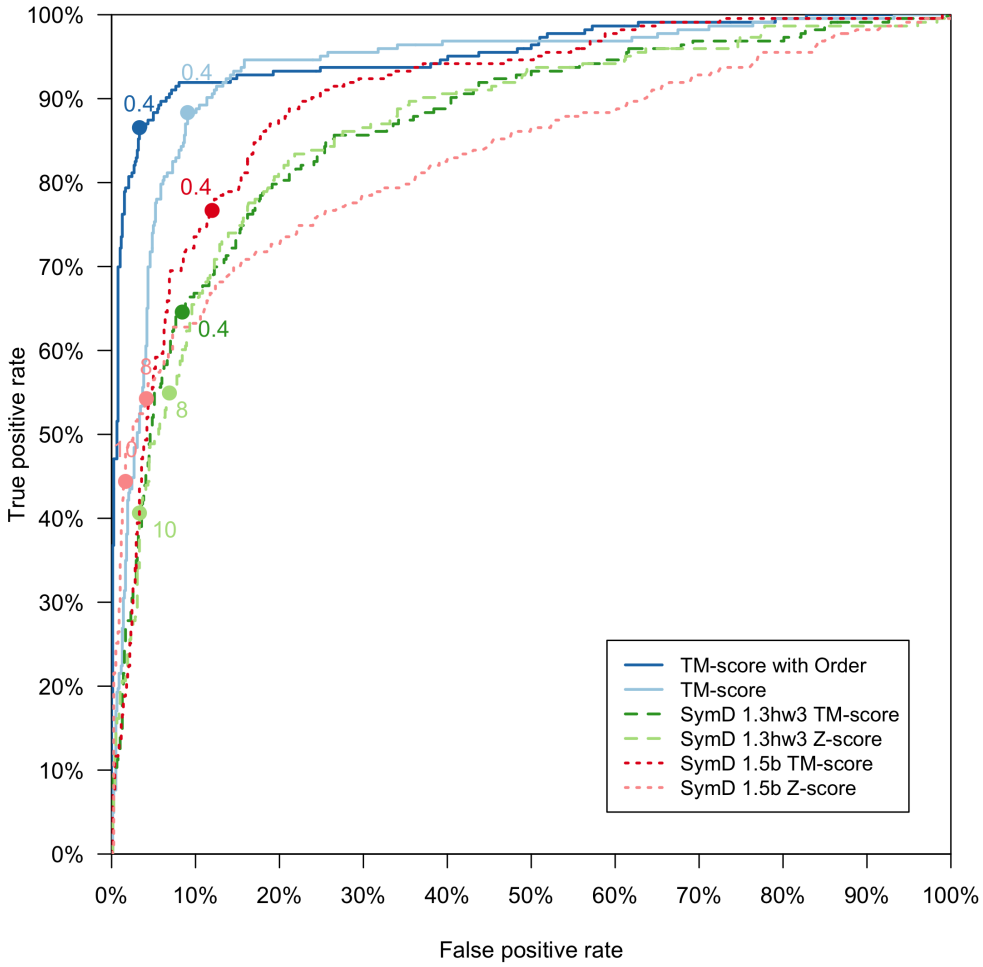


Figure 7.S4: Receiver Operating Characteristic curves for CE-Symm and SymD. In addition to the scoring functions shown in Figure 7.3, SymD performance was also analyzed using the Z-score reported by the program. The thresholds used for determining symmetry (see Table 7.1) are indicated with circles.

7.10 Supplemental Tables

Table 7.S1: Second-level Enzyme Commission numbers and their percentage symmetry among SCOP superfamilies.

EC	Description	% S[1]	NSf[2]
1.10	Oxidoreductases: acting on diphenols and related substances as donors	10	20
1.13	Oxidoreductases: acting on single donors with incorporation of molecular oxygen	33	15
1.14	Oxidoreductases: acting on paired donors, with incorporation or reduction of molecular oxygen	16	32
1.18	Oxidoreductases: acting on iron–sulfur proteins as donors	13	8
1.1	Oxidoreductases: acting on the CH-OH group of donors	16	38
1.20	Oxidoreductases: acting on phosphorus or arsenic in donors	25	4
1.21	Oxidoreductases: acting on x-H and y-H to form an x-y bond	0	1
1.2	Oxidoreductases: acting on the aldehyde or oxo group of donors	16	25
1.3	Oxidoreductases: acting on the CH-CH group of donors	16	32
1.4	Oxidoreductases: acting on the CH-NH(2) group of donors	8.3	24
1.5	Oxidoreductases: acting on the CH-NH group of donors	18	28
1.7	Oxidoreductases: acting on other nitrogenous compounds as donors	18	22
1.8	Oxidoreductases: acting on a sulfur group of donors	10	29
2.10	Transferases: transferring molybdenum- or tungsten -containing groups	33	3
2.2	Transferases: transferring aldehyde or ketone residues	20	5
2.5	Transferases: transferring alkyl or aryl groups, other than methyl groups	23	31

continued . . .

[1] Percentage of superfamilies that are symmetric

[2] The number of superfamilies

Table 7.S1: (Continued) Second-level Enzyme Commission numbers and their percentage symmetry among SCOP superfamilies.

EC	Description	% S[1]	NSf[2]
2.7	Transferases: transferring phosphorous-containing groups	17	170
2.8	Transferases: transferring sulfur-containing groups	6.3	16
2.9	Transferases: transferring selenium-containing groups	0	1
3.4	Hydrolases: acting on peptide bonds (peptide hydrolases)	21	95
3.5	Hydrolases: acting on carbon–nitrogen bonds, other than peptide bonds	12	60
1.11	Oxidoreductases: acting on a peroxide as acceptor	6.3	16
1.12	Oxidoreductases: acting on hydrogen as donor	13	8
1.15	Oxidoreductases: acting on superoxide as acceptor	40	5
1.16	Oxidoreductases: oxidizing metal ions	0	3
1.17	Oxidoreductases: acting on CH or CH(2) groups	19	16
1.23	Oxidoreductases: reducing C-O-C group as acceptor	100	1
1.6	Oxidoreductases: acting on NADH or NADPH	16	25
1.9	Oxidoreductases: acting on a heme group of donors	6.3	16
1.97	Oxidoreductases: other oxidoreductases	40	5
2.1	Transferases: transferring one-carbon groups	13	53
2.3	Transferases: acyltransferases	17	53
2.4	Transferases: glycosyltransferases	16	45
2.6	Transferases: transferring nitrogenous groups	22	9
3.1	Hydrolases: acting on ester bonds	14	142
3.11	Hydrolases: acting on carbon–phosphorus bonds	33	3
3.13	Hydrolases: acting on carbon-sulfur bonds	100	1
3.2	Hydrolases: glycosylases	20	56

continued . . .

[1] Percentage of superfamilies that are symmetric

[2] The number of superfamilies

Table 7.S1: (Continued) Second-level Enzyme Commission numbers and their percentage symmetry among SCOP superfamilies.

EC	Description	% S[1]	NSf[2]
3.3	Hydrolases: acting on ether bonds	11	9
3.6	Hydrolases: acting on acid anhydrides	19	99
3.7	Hydrolases: acting on carbon–carbon bonds	0	5
3.8	Hydrolases: acting on halide bonds	0	3
4.1	Lyases: carbon–carbon lyases	26	57
4.2	Lyases: carbon–oxygen lyases	10	79
4.3	Lyases: carbon–nitrogen lyases	18	11
4.4	Lyases: carbon–sulfur lyases	11	9
4.5	Lyases: carbon–halide lyases	33	3
4.6	Lyases: phosphorus–oxygen lyases	33	9
4.99	Lyases: other lyases	33	6
5.1	Isomerases: racemases and epimerases	38	21
5.2	Isomerases: cis-trans-isomerases	17	12
5.3	Isomerases: intramolecular oxidoreductases	26	34
5.4	Isomerases: intramolecular transferases (mutases)	21	28
5.5	Isomerases: intramolecular lyases	20	10
5.99	Isomerases: other isomerases	17	12
6.1	Ligases: forming carbon–oxygen bonds	17	24
6.2	Ligases: forming carbon-sulfur bonds	50	4
6.3	Ligases: forming carbon–nitrogen bonds	11	74
6.4	Ligases: forming carbon–carbon bonds	0	5
6.5	Ligases: forming phosphoric ester bonds	33	9
6.6	Ligases: forming nitrogen–metal bonds	0	1

[1] Percentage of superfamilies that are symmetric

[2] The number of superfamilies

Table 7.S2: Percentages of superfamilies that are symmetric among folds with significant symmetry. Specifically, folds with at least 30% symmetry, excluding those containing with fewer than 10 domains. % Symm is calculated by normalizing by the number of domains per superfamily.

Fold	% Symm	No. Superfamilies	No. Domains
a	18	507	24891
a.102	67	6	536
a.124	100	1	27
a.126	100	1	334
a.129	100	1	201
a.132	100	1	214
a.139	100	1	12
a.174	100	1	22
a.194	100	1	18
a.2	60	20	562
a.213	100	1	16
a.24	57	28	681
a.246	33	3	35
a.25	50	6	2021
a.26	100	1	271
a.28	33	3	145
a.281	100	1	12
a.29	33	15	480
a.30	62	8	65
a.40	67	3	90
a.41	100	1	48
a.42	100	1	90
a.56	100	1	59

continued . . .

Table 7.S2: (Continued) Percentages of superfamilies that are symmetric among folds with significant symmetry. Specifically, folds with at least 30% symmetry, excluding those containing with fewer than 10 domains. % Symm is calculated by normalizing by the number of domains per superfamily.

Fold	% Symm	No. Superfamilies	No. Domains
a.65	100	1	87
a.66	100	1	69
a.7	31	16	272
a.77	100	1	57
b	24	354	39497
b.1	39	28	11419
b.11	100	1	116
b.111	100	1	12
b.114	100	1	10
b.12	100	1	58
b.123	100	1	25
b.129	50	2	32
b.138	100	1	24
b.15	100	1	72
b.159	50	2	59
b.2	30	10	667
b.23	67	3	38
b.42	100	8	580
b.44	100	2	90
b.45	33	3	174
b.49	67	3	207
b.52	50	2	173
b.58	100	1	133

continued . . .

Table 7.S2: (Continued) Percentages of superfamilies that are symmetric among folds with significant symmetry. Specifically, folds with at least 30% symmetry, excluding those containing with fewer than 10 domains. % Symm is calculated by normalizing by the number of domains per superfamily.

Fold	% Symm	No. Superfamilies	No. Domains
b.61	50	8	461
b.66	100	1	21
b.67	67	3	103
b.68	82	11	535
b.69	50	14	375
b.70	33	3	305
b.77	100	3	130
b.78	100	1	65
b.8	100	1	78
b.86	100	1	16
b.9	100	1	26
c	17	244	46576
c.1	36	33	7442
c.121	100	1	67
c.129	100	1	29
c.135	100	1	24
c.2	100	1	4088
c.25	100	1	154
c.26	33	3	787
c.32	100	1	131
c.34	100	1	42
c.44	50	2	75
c.54	100	1	33

continued . . .

Table 7.S2: (Continued) Percentages of superfamilies that are symmetric among folds with significant symmetry. Specifically, folds with at least 30% symmetry, excluding those containing with fewer than 10 domains. % Symm is calculated by normalizing by the number of domains per superfamily.

Fold	% Symm	No. Superfamilies	No. Domains
c.57	100	1	119
c.59	100	1	32
c.65	100	1	80
c.77	100	1	302
c.92	67	3	247
c.93	100	1	233
d	14	551	39793
d.127	100	1	143
d.131	100	1	254
d.137	100	1	31
d.151	100	1	84
d.152	100	1	10
d.156	100	1	20
d.160	100	1	46
d.18	100	1	42
d.19	100	1	802
d.190	100	1	51
d.215	100	1	11
d.240	100	1	97
d.32	100	1	389
d.323	100	1	29
d.37	100	1	109
d.52	30	10	150

continued . . .

Table 7.S2: (Continued) Percentages of superfamilies that are symmetric among folds with significant symmetry. Specifically, folds with at least 30% symmetry, excluding those containing with fewer than 10 domains. % Symm is calculated by normalizing by the number of domains per superfamily.

Fold	% Symm	No. Superfamilies	No. Domains
d.58	58	59	3765
d.60	100	1	16
d.61	100	1	30
d.64	50	2	16
d.74	80	5	237
d.76	50	2	11
d.80	100	1	356
d.95	50	2	95
e	4.6	66	3827
e.23	100	1	80
f	24	109	2848
f.14	100	1	88
f.17	80	5	138
f.19	100	1	42
f.20	100	1	54
f.24	100	1	89
f.28	100	1	43
f.34	100	1	14
f.4	50	6	207
f.44	100	1	19
f.54	100	1	29
f.55	100	1	10
all	18	1831	157432

Table 7.S3: Errors in order detection. Each domain in the benchmark is binned according to its manually determined order (rows) and CE-Symm-determined order (columns).

		Reported order							
		1	2	3	4	5	6	7	8
True order	1	37	21	0	3	0	1	0	1
	2	9	134	1	5	1	0	0	0
	3	1	0	10	0	0	0	0	0
	4	0	1	0	2	0	0	0	0
	5	0	0	0	0	3	0	0	0
	6	2	1	4	0	0	2	0	0
	7	0	0	0	0	0	0	9	0
	8	2	3	0	8	0	0	0	7

7.11 Supplemental Files

File 7.1: Types of symmetry for domains in the benchmark, manually annotated.

File 7.2: A table of predictions by CE-Symm and SymD on the benchmark set.

File 7.3: A compressed XML file of CE-Symm results over all domains in SCOP 2.03, restricted to classes a–f. The full alignment, alignment scores, and the axis of symmetry are included for each result.

7.12 Acknowledgements

We would like to thank Jean-Pierre Changeux for inspiring discussions about protein symmetry and Chin-Hsien (Emily) Tai for providing access to SymD, as well as stimulating discussion of symmetry. We would also like to thank Jean-Pierre Changeux for inspiring discussions about protein symmetry.

This chapter was originally published as:

D. Myers-Turnbull, S. E. Bliven, P. W. Rose, Z. K. Aziz, P. Youkharibache, P. E. Bourne, and A. Prlić. Systematic Detection of Internal Symmetry in Proteins Using CE-Symm. *J Mol Biol*, 426(11):2255–2268, May 2014

Author Contributions: Andreas Prlić and myself wrote the original CE-Symm program. Douglas Myers-Turnbull also contributed to programming and did much of the comparison to other methods, as well as systematic application for the census. Zaid Aziz did much of the work for the manual benchmark data, with checks by the other authors. Philippe Youkharibache provided numerous examples of the functional significance of internal symmetry. Phil Bourne provided advice and oversight. All authors contributed to structure analysis and to the manuscript.

Funding: The RCSB PDB is managed by two members of the RCSB: Rutgers and UCSD, and it is funded by National Science Foundation (NSF), National Institute of General Medical Sciences, Department of Energy (DOE), National Library of Medicine, National Cancer Institute, National Institute of Neurological Disorders and Stroke and National Institute of Diabetes and Digestive and Kidney Diseases. The RCSB PDB is a member of the wwPDB. This work was supported by the RCSB PDB grant NSF DBI 0829586. SB was also supported by a grant from the National Institutes of Health, USA (NIH grants T32GM8806). Computation provided in part by the Open Science Grid (Sfiligoi et al., 2009; Pordes et al., 2007)

7.13 Bibliography

A.-L. Abraham, E. P. C. Rocha, and J. Pothier. Swelfe: a detector of internal repeats in sequences and structures. *Bioinformatics*, 24(13):1536–1537, July 2008.

A.-L. Abraham, J. Pothier, and E. P. C. Rocha. Alternative to homo-oligomerisation: the creation of local symmetry in proteins by internal amplification. *J Mol Biol*, 394(3): 522–534, Dec. 2009.

J. Adams, R. Kelso, and L. Cooley. The kelch repeat superfamily of proteins: propellers of cell function. *Trends Cell Biol.*, 10(1):17–24, Jan. 2000.

M. A. Andrade, C. Perez-Iratxeta, and C. P. Ponting. Protein Repeats: Structures, Functions, and Evolution. *J Struct Biol*, 134(2-3):117–131, May 2001.

A. Andreeva, D. Howorth, J.-M. Chandonia, S. E. Brenner, T. J. P. Hubbard, C. Chothia, and A. G. Murzin. Data growth and its impact on the SCOP database: new developments. *Nucleic Acids Res*, 36(Database issue):D419–25, Jan. 2008.

A. Bairoch. The ENZYME database in 2000. *Nucleic Acids Res*, 28(1):304–305, Jan. 2000.

M. Bergdoll, L. D. Eltis, A. D. Cameron, P. Dumas, and J. T. Bolin. All in the family: structural and evolutionary relationships among three modular proteins with diverse functions and variable assembly. *Protein Sci*, 7(8):1661–1670, Aug. 1998.

H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne. The Protein Data Bank. *Nucleic Acids Res*, 28(1): 235–242, Jan. 2000.

S. Bershtein, W. Mu, W. Wu, and E. I. Shakhnovich. Soluble oligomerization provides a beneficial fitness effect on destabilizing mutations. 109(13):4857–4862, Mar. 2012.

M. Blaber, J. Lee, and L. Longo. Emergence of symmetric protein architecture from a simple peptide motif: evolutionary models. *Cell. Mol. Life Sci.*, 69:3999–4006, July 2012.

S. E. Bliven and A. Prlić. Circular permutation in proteins. *PLoS Comput Biol*, 8(3): e1002445, Mar. 2012.

A. Broom, A. C. Doxey, Y. D. Lobsanov, L. G. Berthin, D. R. Rose, P. L. Howell, B. J. McConkey, and E. M. Meiering. Modular evolution and the origins of symmetry: reconstruction of a three-fold symmetric globular protein. *Structure*, 20(1):161–171, Jan. 2012.

J.-M. Chandonia, G. Hon, N. S. Walker, L. Lo Conte, P. Koehl, M. Levitt, and S. E. Brenner. The ASTRAL Compendium in 2004. *Nucleic Acids Res*, 32(Database issue): D189–92, 2004.

J.-P. Changeux and S. J. Edelstein. Allosteric mechanisms of signal transduction. *Science*, 308(5727):1424–1428, June 2005.

I. Chaudhuri, J. Söding, and A. N. Lupas. Evolution of the β-propeller fold. *Proteins: Structure, Function, and Bioinformatics*, 71(2):795–803, May 2008.

H. Chen, Y. Huang, and Y. Xiao. A simple method of identifying symmetric substructures of proteins. *Comput Biol Chem*, 33(1):100–107, Feb. 2009.

S. Choi, J. Jeon, J.-S. Yang, and S. Kim. Common occurrence of internal repeat symmetry in membrane proteins. *Proteins: Structure, Function, and Bioinformatics*, 71(1):68–80, Apr. 2008.

T. P. Combs, U. B. Pajvani, A. H. Berg, Y. Lin, L. A. Jelicks, M. Laplante, A. R. Nawrocki, M. W. Rajala, A. F. Parlow, L. Cheeseboro, Y.-Y. Ding, R. G. Russell, D. Lindemann, A. Hartley, G. R. C. Baker, S. Obici, Y. Deshaies, M. Ludgate, L. Rossetti, and P. E. Scherer. A transgenic mouse with a deletion in the collagenous domain of adiponectin displays elevated circulating adiponectin and improved insulin sensitivity. *Endocrinology*, 145(1):367–383, Jan. 2004.

R. Dutzler, E. B. Campbell, and R. MacKinnon. Gating the selectivity filter in ClC chloride channels. *Science*, 300(5616):108–112, Apr. 2003.

L. R. Forrest and G. Rudnick. The rocking bundle: a mechanism for ion-coupled solute flux by symmetrical transporters. *Physiology (Bethesda)*, 24:377–386, Dec. 2009.

C. Fortenberry, E. A. Bowman, W. Proffitt, B. Dorr, S. Combs, J. Harp, L. Mizoue, and J. Meiler. Exploring symmetry as an avenue to the computational design of large protein domains. *J Am Chem Soc*, 133(45):18026–18029, Nov. 2011.

N. K. Fox, S. E. Brenner, and J.-M. Chandonia. SCOPe: Structural Classification of Proteins–extended, integrating SCOP and ASTRAL data and classification of new structures. *Nucleic Acids Res*, 42(1):D304–9, Jan. 2014.

H. Ge, Y. Xiong, B. Lemon, K. J. Lee, J. Tang, P. Wang, J. Weiszmann, N. Hawkins, J. Laudemann, X. Min, D. Penny, T. Wolfe, Q. Liu, R. Zhang, W.-C. Yeh, W. Shen, R. Lindberg, Z. Wang, J. Sheng, and Y. Li. Generation of novel long-acting globular adiponectin molecules. *J Mol Biol*, 399(1):113–119, May 2010.

J. Giraldo and F. Ciruela, editors. *Oligomerization in Health and Disease*. Progress in Molecular Biology and Translational Science. Academic Press, May 2013.

D. S. Goodsell and A. J. Olson. Structural symmetry and protein function. *Annu Rev Biophys Biomol Struct*, 29:105–153, 2000.

N. V. Grishin. Fold change in evolution of protein structures. *J Struct Biol*, 134(2-3): 167–185, Apr. 2001.

A. Guerler, C. Wang, and E. W. Knapp. Symmetric structures in the universe of protein folds. *J Chem Inf Model*, 49(9):2147–2151, Sept. 2009.

C. Hug and H. F. Lodish. The role of the adipocyte hormone adiponectin in cardiovascular disease. *Curr Opin Pharmacol*, 5(2):129–134, Apr. 2005.

Y. Jia, T. G. Dewey, I. N. Shindyalov, and P. E. Bourne. A new scoring function and associated statistical significance for structure alignment by CE. *J Comput Biol*, 11(5): 787–799, 2004.

C. P. Jones and A. R. Ferré-D'Amaré. RNA quaternary structure and global symmetry. *Trends Biochem Sci*, Mar. 2015.

Z. S. Juo, T. K. Chiu, P. M. Leiberman, I. Baikalov, A. J. Berk, and R. E. Dickerson. How proteins recognize the TATA box. *J Mol Biol*, 261(2):239–254, Aug. 1996.

Z. Kelman and M. O'Donnell. Structural and functional similarities of prokaryotic and eukaryotic DNA polymerase sliding clamps. *Nucleic Acids Res*, 23(18):3613–3620, Sept. 1995.

C. Kim, J. Basner, and B. Lee. Detecting internally symmetric protein structures. *BMC Bioinformatics*, 11:303, 2010.

K. Kinoshita, A. Kidera, and N. Go. Diversity of functions of proteins with internal symmetry in spatial arrangement of secondary structural elements. *Protein Sci*, 8(6): 1210–1217, June 1999.

M. Klingenberg. Membrane protein oligomeric structure and transport function. *Nature*, 290(5806):449–454, Apr. 1981.

V. M. Korkhov, S. A. Mireku, and K. P. Locher. Structure of AMP-PNP-bound vitamin B12 transporter BtuCD-F. *Nature*, 490(7420):367–372, Oct. 2012.

D. E. Koshland. The evolution of function in enzymes. *Fed. Proc.*, 35(10):2104–2111, Aug. 1976.

J. Lee and M. Blaber. Experimental support for the evolution of symmetric protein architecture from a simple peptide motif. 108(1):126–130, Jan. 2011.

E. D. Levy, J. B. Pereira-Leal, C. Chothia, and S. A. Teichmann. 3D complex: a structural classification of protein complexes. *PLoS Comput Biol*, 2(11):e155, Nov. 2006.

N. Maeda, I. Shimomura, K. Kishida, H. Nishizawa, M. Matsuda, H. Nagaretani, N. Furuyama, H. Kondo, M. Takahashi, Y. Arita, R. Komuro, N. Ouchi, S. Kihara, Y. Tochino, K. Okutomi, M. Horie, S. Takeda, T. Aoyama, T. Funahashi, and Y. Matsuzawa. Diet-induced insulin resistance in mice lacking adiponectin/ACRP30. *Nat. Med.*, 8(7):731–737, July 2002.

J. M. Matthews, M. Sunde, D. A. Gell, R. P. Grant, J. P. Mackay, S. Jones, S. H. MacKenzie, A. C. Clark, M. D. Griffin, and J. A. Gerrard. *Protein Dimerization and Oligomerization in Biology*. Landes Bioscience and Springer Science+Business Media, LLC, May 2012.

G. Mayr, F. S. Domingues, and P. Lackner. Comparative analysis of protein structure alignments. *BMC Struct Biol*, 7:50, 2007.

X. Min, B. Lemon, J. Tang, Q. Liu, R. Zhang, N. Walker, Y. Li, and Z. Wang. Crystal structure of a single-chain trimer of human adiponectin globular domain. *FEBS Lett*, 586(6):912–917, Mar. 2012.

K. Mizuguchi and N. Go. Comparison of spatial arrangements of secondary structural elements in proteins. *Protein Eng*, 8(4):353–362, Apr. 1995.

J. Monod, J. Wyman, and J.-P. Changeux. On the Nature of Allosteric Transitions: A Plausible Model. *J Mol Biol*, 12:88–118, May 1965.

K. B. Murray, W. R. Taylor, and J. M. Thornton. Toward the detection and validation of repeats in protein structure. *Proteins: Structure, Function, and Bioinformatics*, 57(2): 365–380, Nov. 2004.

A. G. Murzin, S. E. Brenner, T. Hubbard, and C. Chothia. SCOP: a structural classification of proteins database for the investigation of sequences and structures. *J Mol Biol*, 247 (4):536–540, Apr. 1995.

D. Myers-Turnbull, S. E. Bliven, P. W. Rose, Z. K. Aziz, P. Youkharibache, P. E. Bourne, and A. Prlić. Systematic Detection of Internal Symmetry in Proteins Using CE-Symm. *J Mol Biol*, 426(11):2255–2268, May 2014.

N. Nagano, C. A. Orengo, and J. M. Thornton. 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, Aug. 2002.

S.-Y. Park, X. Chao, G. Gonzalez-Bonet, B. D. Beel, A. M. Bilwes, and B. R. Crane. Structure and function of an unusual family of protein phosphatases: the bacterial chemotaxis proteins CheC and CheX. *Mol Cell*, 16(4):563–574, Nov. 2004.

M. F. Perutz, M. G. Rossmann, A. F. Cullis, H. Muirhead, G. Will, and A. C. North. Structure of haemoglobin: a three-dimensional Fourier synthesis at 5.5-A. resolution, obtained by X-ray analysis. *Nature*, 185(4711):416–422, Feb. 1960.

R. Pordes, D. Petravick, B. Kramer, D. Olson, M. Livny, A. Roy, P. Avery, K. Blackburn, T. Wenaus, F. Würthwein, I. Foster, R. Gardner, M. Wilde, A. Blatecky, J. McGee, and R. Quick. The open science grid. *J. Phys.: Conf. Ser.*, 78(1):012057, Aug. 2007.

A. Prlić, F. S. Domingues, and M. J. Sippl. Structure-derived substitution matrices for alignment of distantly related sequences. *Protein Eng Des Sel*, 13(8):545–550, Aug. 2000.

A. Prlić, S. E. Bliven, P. W. Rose, W. F. Bluhm, C. Bizon, A. Godzik, and P. E. Bourne. Pre-calculated protein structure alignments at the RCSB PDB website. *Bioinformatics*, 26(23):2983–2985, Dec. 2010.

M. Punta, P. C. Coggill, R. Y. Eberhardt, J. Mistry, J. Tate, C. Boursnell, N. Pang, K. Forslund, G. Ceric, J. Clements, A. Heger, L. Holm, E. L. L. Sonnhammer, S. R. Eddy, A. Bateman, and R. D. Finn. The Pfam protein families database. *Nucleic Acids Res*, 40(Database issue):D290–301, Jan. 2012.

P. W. Rose, C. Bi, W. F. Bluhm, C. H. Christie, D. Dimitropoulos, S. Dutta, R. K. Green, D. S. Goodsell, A. Prlić, M. Quesada, G. B. Quinn, A. G. Ramos, J. D. Westbrook, J. Young, C. Zardecki, H. M. Berman, and P. E. Bourne. The RCSB Protein Data Bank: new resources for research and education. *Nucleic Acids Res*, 41(D1):D475–D482, Dec. 2012.

R. I. Sadreyev, B.-H. Kim, and N. V. Grishin. Discrete-continuous duality of protein structure space. *Curr Opin Struct Biol*, 19(3):321–328, June 2009.

I. Sfiligoi, D. C. Bradley, B. Holzman, P. Mhashilkar, S. Padhi, and F. Wurthwein. The Pilot Way to Grid Resources Using glideinWMS. *Computer Science and Information Engineering, 2009 WRI World Congress on*, 2:428–432, 2009.

E. S. C. Shih and M.-J. Hwang. Alternative alignments from comparison of protein structures. *Proteins: Structure, Function, and Bioinformatics*, 56(3):519–527, Aug. 2004.

E. S. C. Shih, R.-c. R. Gan, and M.-J. Hwang. OPAAS: a web server for optimal, permuted, and other alternative alignments of protein structures. *Nucleic Acids Res*, 34(Web Server issue):W95–8, July 2006.

I. N. Shindyalov and P. E. Bourne. Protein structure alignment by incremental combinatorial extension (CE) of the optimal path. *Protein Eng*, 11(9):739–747, Sept. 1998.

S. Shklyaev, G. Aslanidi, M. Tennant, V. Prima, E. Kohlbrenner, V. Kroutov, M. Campbell-Thompson, J. Crawford, E. W. Shek, P. J. Scarpace, and S. Zolotukhin. Sustained peripheral expression of transgene adiponectin offsets the development of diet-induced obesity in rats. 100(24):14217–14222, Nov. 2003.

S. Uliel, A. Fliess, A. Amir, and R. Unger. A simple algorithm for detecting circular permutations in proteins. *Bioinformatics*, 15(11):930–936, Nov. 1999.

I. A. Vergara, T. Norambuena, E. Ferrada, A. W. Slater, and F. Melo. StAR: a simple tool for the statistical comparison of ROC curves. *BMC Bioinformatics*, 9:265, 2008.

G. L. Waldrop. The role of symmetry in the regulation of bacterial carboxyltransferase. *BioMolecular Concepts*, 2(1-2), 2011.

E. C. Webb and IUBMB. *Enzyme nomenclature 1992. Recommendations of the Nomenclature Committee of the International Union of Biochemistry and Molecular Biology on the Nomenclature and Classification of Enzymes.* Academic Press, 1992.

C. P. Whitman, G. D. Hegeman, W. W. Cleland, and G. L. Kenyon. Symmetry and asymmetry in mandelate racemase catalysis. *Biochemistry*, 24(15):3936–3942, July 1985.

P. G. Wolynes. Symmetry and the energy landscapes of biomolecules. 93(25):14249–14255, Dec. 1996.

C. F. Wright, S. A. Teichmann, J. Clarke, and C. M. Dobson. The importance of sequence diversity in the aggregation and evolution of proteins. *Nature*, 438(7069):878–881, Dec. 2005.

Y. Ye and A. Godzik. Flexible structure alignment by chaining aligned fragment pairs allowing twists. *Bioinformatics*, 19 Suppl 2:ii246–55, Oct. 2003.

Y. Zhang and J. Skolnick. Scoring function for automated assessment of protein structure template quality. *Proteins: Structure, Function, and Bioinformatics*, 57(4):702–710, Dec. 2004.

Y. Zhang and J. Skolnick. TM-align: a protein structure alignment algorithm based on the TM-score. *Nucleic Acids Res*, 33(7):2302–2309, 2005.

Chapter 8

Order Detection Methods in CE-Symm

8.1 Introduction

One of the most interesting properties of internally symmetric proteins is their order of symmetry, but this is a property which has been hard to automatically determine. CE-Symm is a structure alignment program, and outputs a single alignment between a symmetric protein and a rotated copy of itself. However, structural biologists tend to think of symmetric proteins as a multiple alignment between the repeated substructures. Thus, we have tried to develop tools to reflect this view of symmetric proteins.

One of the difficulties of determining the order comes from inconsistencies in how humans would assign it. Assuming an evolutionary mechanism for internal symmetry, the ideal order of an internally symmetric protein would reflect a series of duplication and rearrangement events that led from the primordial set of protodomains to the folds observed today. However, such a model ignores cases of convergent substructures, and even in homologous cases the alignment between symmetric repeats is rarely clear or unambiguous. Thus we are forced to rely on structural similarity between substructures, which leads to a difficulty in choosing thresholds for similarity. For instance, TIM barrels contain 8 βα repeats, so they are often considered to have C8 symmetry. However, structural distortions can lead to significantly better alignments at some rotation angles

than others, which can make the correct order ambiguous (see fig. 8.1). It has been suggested that the ancestral motif for TIM barrels may be a $(\beta\alpha)_2$ subunit that underwent four-fold duplication, which would imply a C4 symmetry rather than C8 (Nagano et al., 2002).

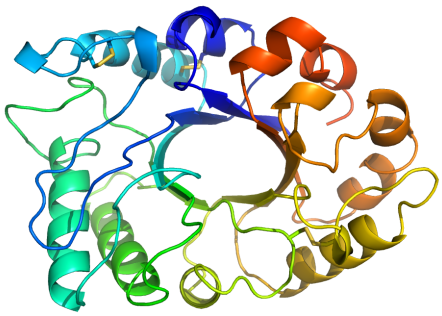


Figure 8.1: Structural distortion in the TIM barrel protein, Hevamine [PDB:2HVM]. The $1/2$ turn rotation has significantly lower RMSD than the $1/4$ or $1/8$ turn rotations, leading one to potentially favor a C2 classification over the C8.

Given these difficulties, several approaches to order detection have been developed. These can be grouped into three broad categories: those based on the rotation angle, those based on the rotation axis, and those based on the alignment itself.

8.2 Rotation Angle

Perhaps the most straightforward method for detecting the order is to measure the rotation angle for the alignment. The order is then chosen based on the error between the observed angle and the ideal rotation angles for a given order. The angular error for each order as a function of the observed angle is shown in Figure 8.2. Angles can then be partitioned into discrete order determinations when the error to an ideal angle falls below some threshold.

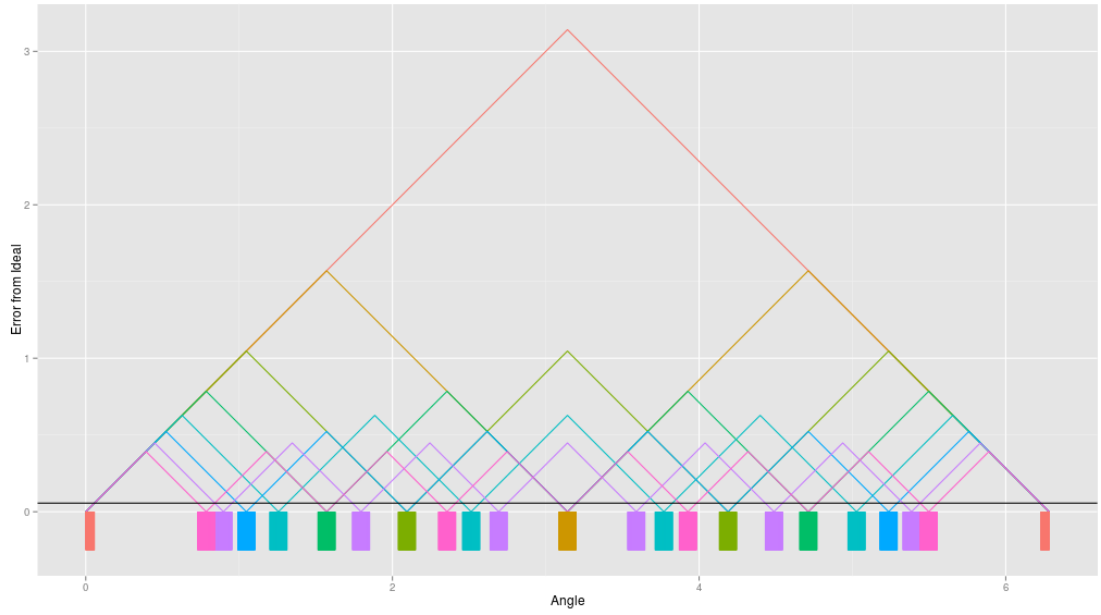


Figure 8.2: Distance from a given angle to the closest ideal angle for orders 1–8. The black horizontal line indicates an error of $1/112$ of a turn (.056 radians), or half the difference in angle between a C7 and C8 rotation. Color swatches below the axis give order predictions for this error rate; uncolored regions default to C1 order.

Note that the original CE-Symm publication (Chapter 7) used an incorrect equation for determining the angular error. The correct equation for observed angle θ and order k is a triangle wave with period $1/k$:

$$\delta(\theta,k) = \frac{\tau}{k}\left|\left|\frac{\theta k}{\tau} - \frac{1}{2}\right| \bmod 1 - \frac{1}{2}\right| \tag{8.1}$$

The difficulty with this scheme is that high angular accuracy is required to differentiate between high orders. For instance, rotations by $1/8$ and $1/7$ turns differ by only 0.056 radians.[1] Significant CE-Symm alignments of symmetric proteins have an

[1] Angles are given in turns or radians, and may use $\tau = 2\pi$ for ease of interconversion.

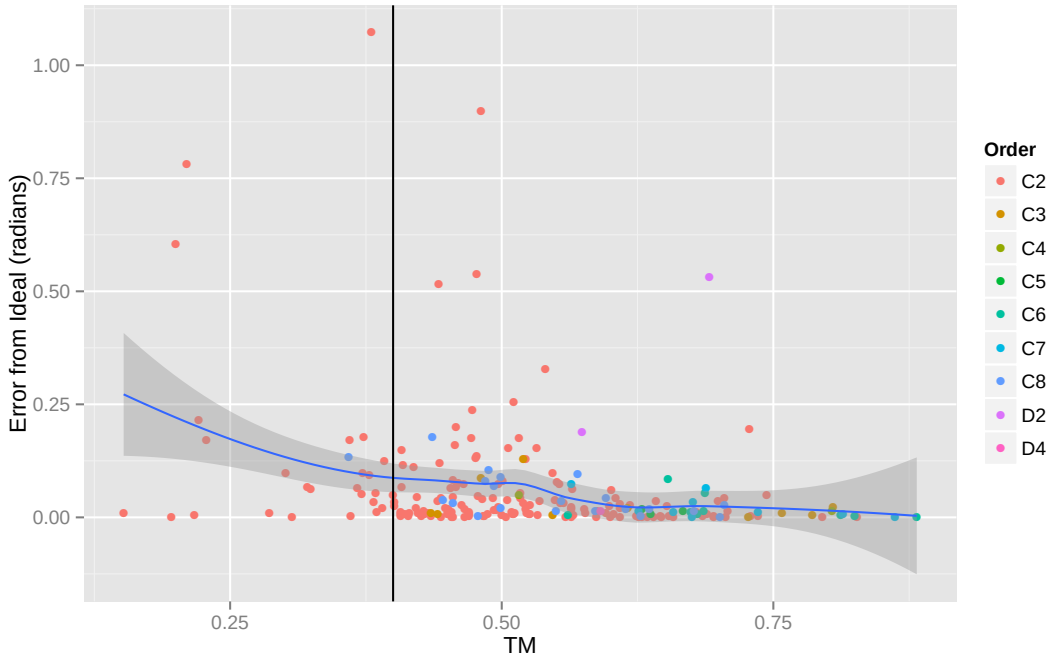


Figure 8.3: Observed error from an ideal angle for rotationally symmetric benchmark cases. The black line indicates the threshold of $TM \geq .4$ used for determining significance.

average error of 0.05 radians (see fig. 8.3), so even considering only angles up to 8 pushes the bounds of feasibility. While this could potentially be minimized by optimizing which angular intervals are assigned to each order, it can never be fully obviated. One such attempt is the normalized scheme in Figure 8.4, which omits the initial τ/k term from equation 8.1. This scheme favors lower orders (following the general trend observed in internally symmetric proteins in the PDB) by broadening the angles assigned to lower orders.

In addition to practical limitations from errors in the CE-Symm superposition, rotation angle methods are also liited to finding the order of the CE-Symm alignment, and are unable to find higher orders that might be compatible with the given alignment.

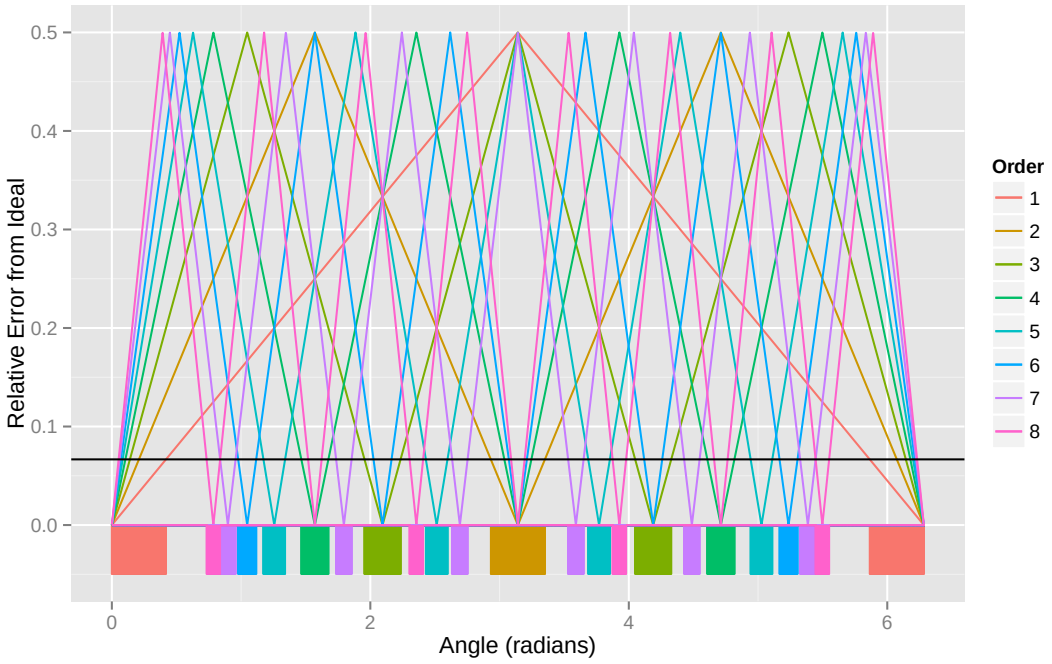


Figure 8.4: An alternate method for assigning order based on the angle. Errors are normalized based on the ideal rotation angle for a given order. The black line indicates an error of 6.67% of the ideal rotation angle, corresponding to the differences between C7 and C8, as in fig. 8.2.

8.3 Rotation Axis

While determining the order for a CE-Symm alignment is difficult, the rotation axis can be determined fairly accurately for significant alignments. The rotation axis methods work by rotating the protein along the alignment and comparing the structural similarity at each angle.

While it is possible to efficiently find an optimal pairwise alignment between proteins for a given orientation (Poleksic, 2011), we instead opt for a simpler alignment-free heuristic. The coordinates of α-carbon atoms of the protein are first transformed by rotating them around the axis by the desired angle, θ. For every transformed atom, the nearest atom in the original protein is identified, and vice versa. This does not form a pairwise alignment, since an atom may be the closest partner to several transformed atoms. The average distance over all C_α atom in both the original and transformed

structures is used, which provides an estimate of the goodness of fit between the original and the rotated structure.

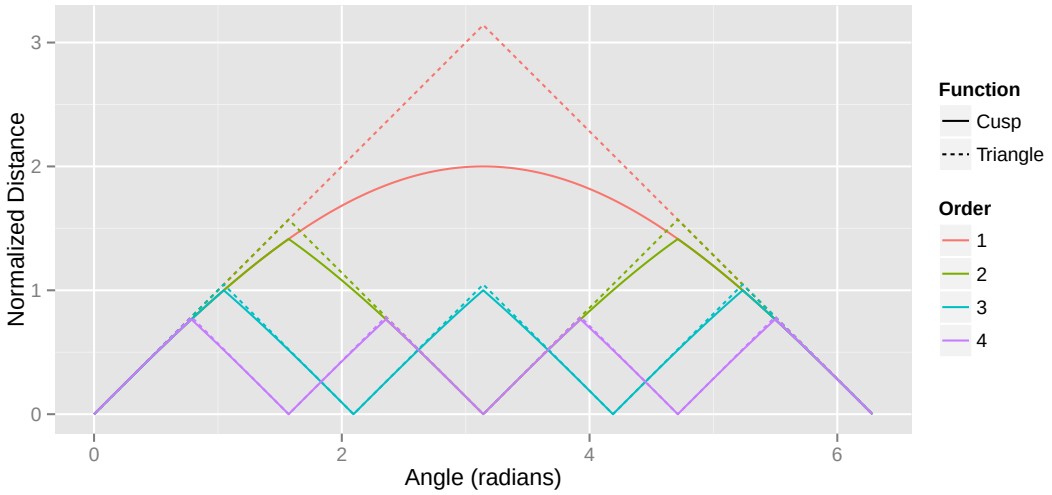


Figure 8.5: Cusp function. Graph of equation 8.3 for orders 1–4, with a perfect triangle wave for comparison.

In symmetric structures, the average distance is a periodic function of the rotation angle. A number of methods were used to fit analytical functions to the calculated distances and determine the order. Initial experiments used the "harmonic" function for a given order k:

$$f_k(\theta) = \sin^2\left(\frac{k\theta}{2}\right) \tag{8.2}$$

Later, the "cusp" function was used. It is derived from the expected distance between a perfectly symmetric point as it is rotated around the axis of symmetry. For

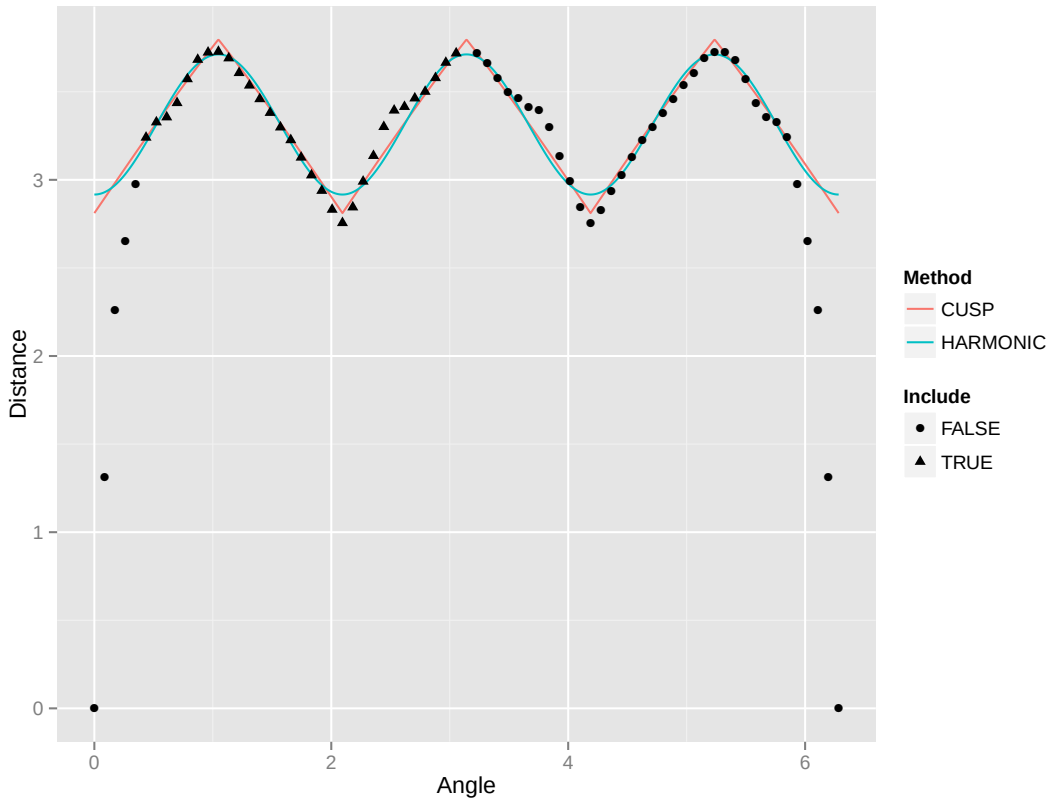


Figure 8.6: Harmonic and cusp functions of order 3 fit to the C3 ubiquinol oxidase protein (SCOP:d1ffta_). Triangular points were used for fitting the functions.

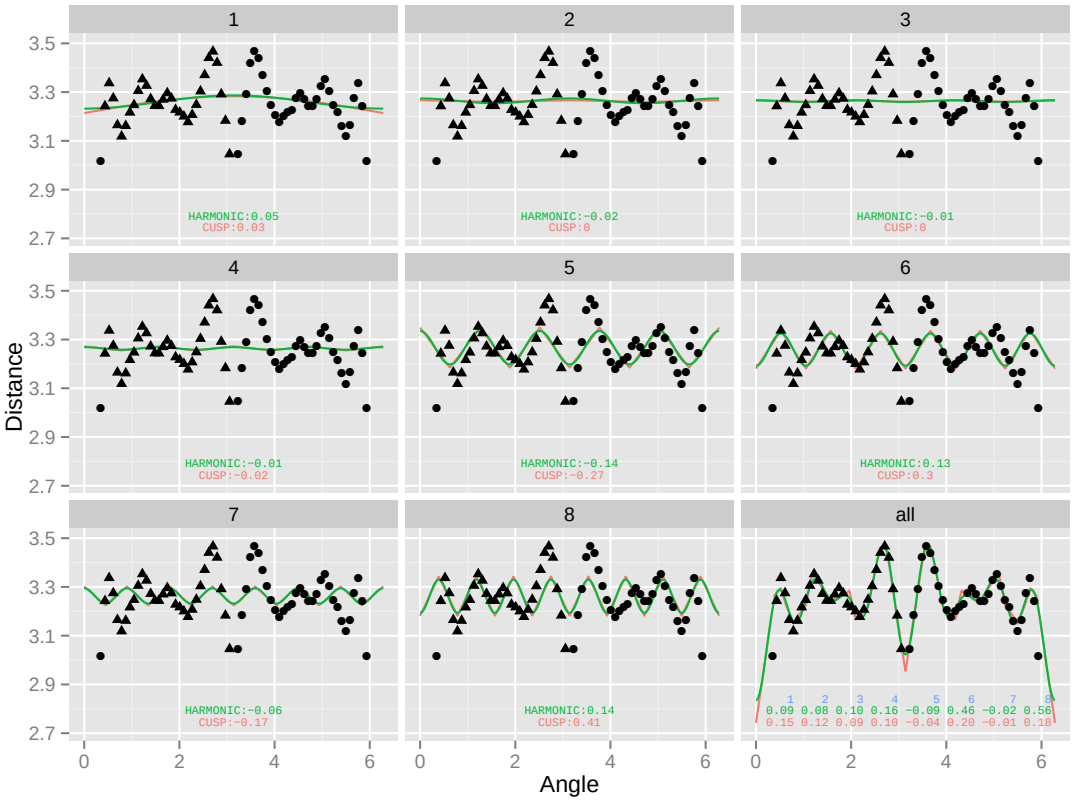


Figure 8.7: Functions fit to the observed distances for triose phosphate isomerase (PDB:1TIM). The 1–8 categories fit single orders, while the "all" graph fits a linear sum of orders 1–8. The amplitude of the wave is printed for each order. Both methods have the highest amplituted for the correct order of C8.

higher orders, the cusp function converges to a triangle wave (Figure 8.5).

$$\begin{aligned} f_k(\theta) &= \min_{i\in\mathbb{N}} \sqrt{2-2\cos\left(\theta+\frac{i\tau}{k}\right)}. \\ &= \sqrt{2-2\cos\left(\frac{\tau}{k}\left|\left|\frac{k\theta}{\tau}-\frac{1}{2}\right| \mod 1-\frac{1}{2}\right|\right)} \end{aligned} \tag{8.3}$$

To determine the order, the observed data were fit to each of the orders under consideration (typically $k \in \{1,\ldots,8\}$) using linear regression. The order with the lowest sum squared error to the observed data were chosen (Figure 8.7).

The PeakCounting method used a LOESS fit to smooth data, followed by a count of the number of local maxima to determine the order.

8.4 Alignment Map

Rather than relying on geometric properties of the superposition, the third method for determining the order relies only on the alignment itself. This method is described in detail in Myers-Turnbull et al. (2014).

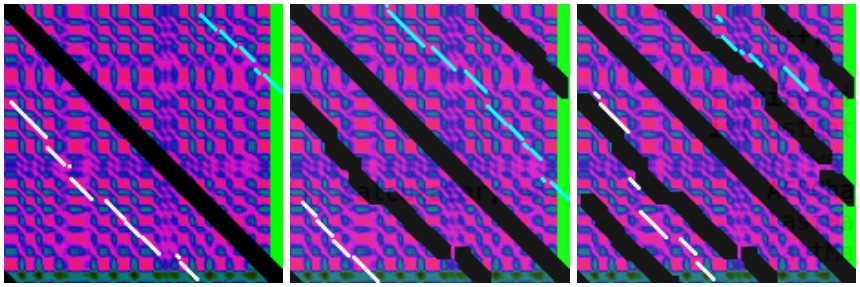
The algorithm considers a self-alignment to be a function from the set of residues in a protein to itself. We say that $f(x) = y$ if CE-Symm aligned residues x and y. If CE-Symm identified rotational symmetry within the protein, then the repeated function composition $f^k(x)$ corresponds to repeated rotations. When the function is applied a number of times equal to the order of the underlying CE-Symm alignment, k^*, then $f^{k^*}(x) \approx x$, corresponding to a rotation by one turn. To identify the order of a self-alignment, we try successively larger values of k and find the root-mean-square deviation (RMSD) found for each according to the formula:

$$RMSD = \sqrt{\sum_i \left(f^k(x_i) - x_i\right)^2}$$

The correct order is determined by identifying large decreases in RMSD. In practice, a threshold of 40% decrease was found to correctly identify the order in most cases. If no such drops are identified for k of 8 or less, an order of 1 (no rotational symmetry) is assumed.

8.4.1 Multipass Map

One possible way to improve the alignment map method would be to generalize it to include alignments for multiple rotation angles. The existing method takes the single top alignment and uses that to generate the map. By preventing CE-Symm from finding previous alignments via blacking out the relevant portions of the dynamic programming matrix, it is possible for CE-Symm to find slightly lower-scoring paths corresponding to other valid rotations (see Figure 8.8). This does come at the cost of significant computational overhead, since CE-Symm must be run a number of times proportionate



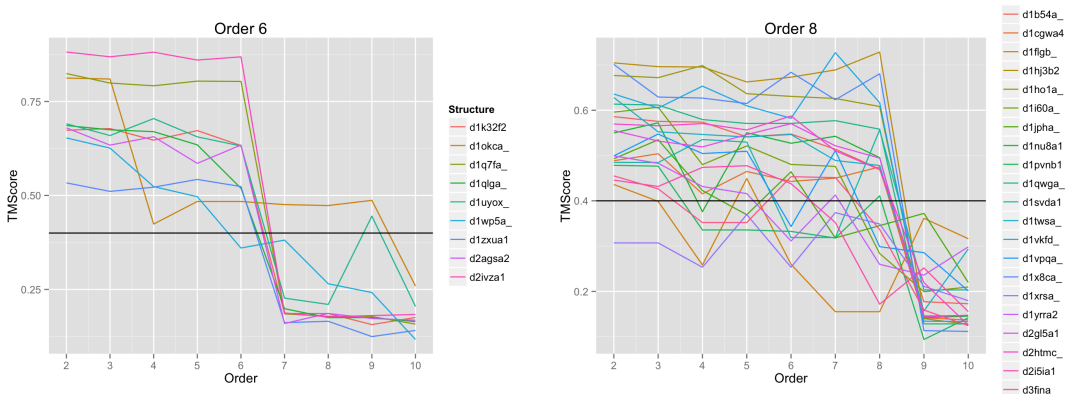
(a) (b) (c)

Figure 8.8: Sequential passes of CE-Symm on the C3 protein interleukin-1 beta [PDB:1ITB.A]. (a) In the first pass, only the diagonal is disallowed, yielding the top-scoring alignment. (b) Later passes disallow earlier alignments, forcing CE-Symm to find new rotations. (c) After alignments for all rotations have been identified, only low-scoring paths are available, leading to very low alignment scores.

to the order of the structure.

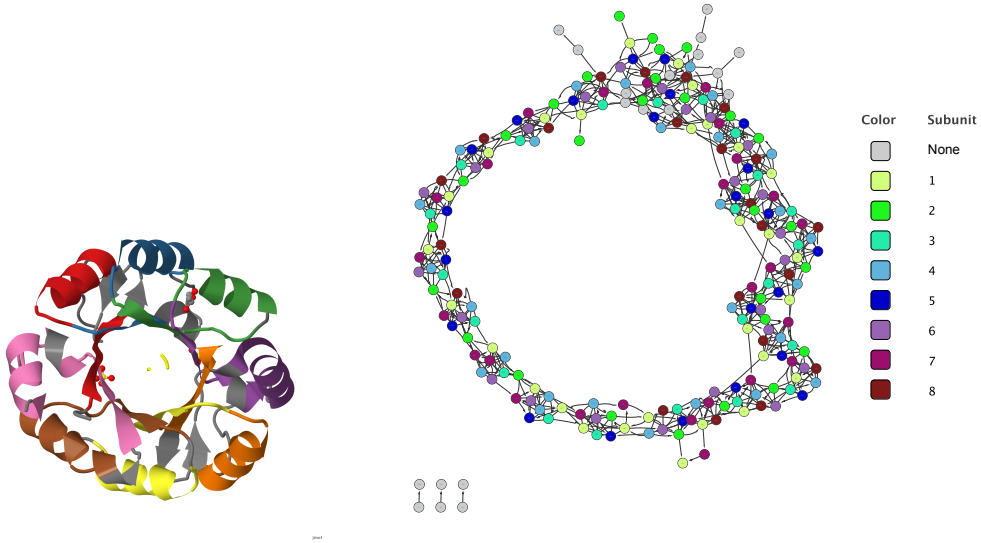
This multipass mode of executing CE-Symm provides a heuristic method for order detection. In proteins with clear symmetry, the alignments corresponding to all valid rotations give scores above the standard threshold of TM-Score ≥ 0.4. However, after all valid alignments have been found and forbidden, further attempts to find an alignment result in erroneous alignments with low scores (Figure 8.8c). Thus, the order can in some cases be detected due to the precipitous drop in score for the pass following the correct order (Figure 8.9a). However, for cases where the repeated subunits have significant divergence, this method becomes less reliable. This is particularly noticeable for proteins with high order (Figure 8.9a). Thus, the additional computational cost may not be justified compared to other methods for order detection.

Together, the multipass alignments contain redundant information about the pairwise relationships between the repeats of the internally symmetric protein. In principle, this information should be usable to increase the accuracy of the alignment map method, since correctly aligned residues will tend to reinforce relative to incorrectly aligned residues. However, the alignment map technique has not yet been generalized from the simple graphs from single alignments to the hypergraphs for high-order proteins. For difficult cases such as in Figure 8.10, these hypergraphs can consist of well connected



(a) (b)

Figure 8.9: Alignment scores over multiple passes. The alignment scores for selected (a) C6 and (b) C8 SCOP domains are shown. In most cases, the scores drop below the TM-Score ≥ 0.4 threshold after the pass corresponding to the correct order. Cases with inconsistent scores such as SCOP:d1flgb_ (lower yellow line in (b)) typically have significant structural divergence between repeats.



(a) (b)

Figure 8.10: Graph of aligned residues from multiple passes of PDB:1VZW. (a) Structure colored by subunit. (b) Graph showing residues of the protein, roughly colored by subunit. Edges are drawn between residues that align in any of the eight rotations. An ideal alignment would contain disjoint cliques of eight vertices each.

modules with sparse interconnections due to incorrect assignments. Additional algorithmic development is required to optimally decompose such networks into cliques of consistent size.

8.5 Comparison of Order Detection Methods

The CE-Symm benchmark includes information about the order of symmetry over 1007 proteins (Myers-Turnbull et al., 2014). Proteins were selected uniformly at random from SCOP superfamilies, so this benchmark provides a broad, non-redundant sample of protein space, including many cases with significant divergence among proteins. Since order detection occurs as a step after the generation of the initial CE-Symm alignment and after determination of whether the protein is symmetric, the benchmark was filtered to 197 cases with closed symmetry (cyclic or dihedral symmetry) and where the initial alignment passed the scoring threshold of TM-Score ≥ 0.4.

An additional benchmark consisting of all 1098 β-propeller proteins from SCOPe 2.0.1 was generated (Fox et al., 2014). β-propellers range from C4 to C8 symmetry. They tend to be "easy" cases for symmetry detection due to the high degree of similarity between structural repeats. The inclusion of all SCOP domains in this benchmark does lead to a high degree of redundancy, which could potentially increase the noise in the results. Thus the balanced CE-Symm benchmark should be considered more trustworthy.

Figure 8.11 shows the performance of various methods on the two benchmarks. Two metrics are considered for comparison. The *accuracy* is the more stringent metric, and is defined as the percentage of structures where the correct order is found. To account for cases such as TIM barrels where some ambiguity exists in the correct order, the *lenient accuracy* was defined as the percentage of structures where either the correct order or a non-trivial divisor thereof is found. Under this metric, detecting 2-fold symmetry in a protein annotated as C4, for example, would still be counted as a correct annotation. Our goal is to create an order detection algorithm with good accuracy, but analyzing the lenient accuracy can highlight reasons for poor performance.

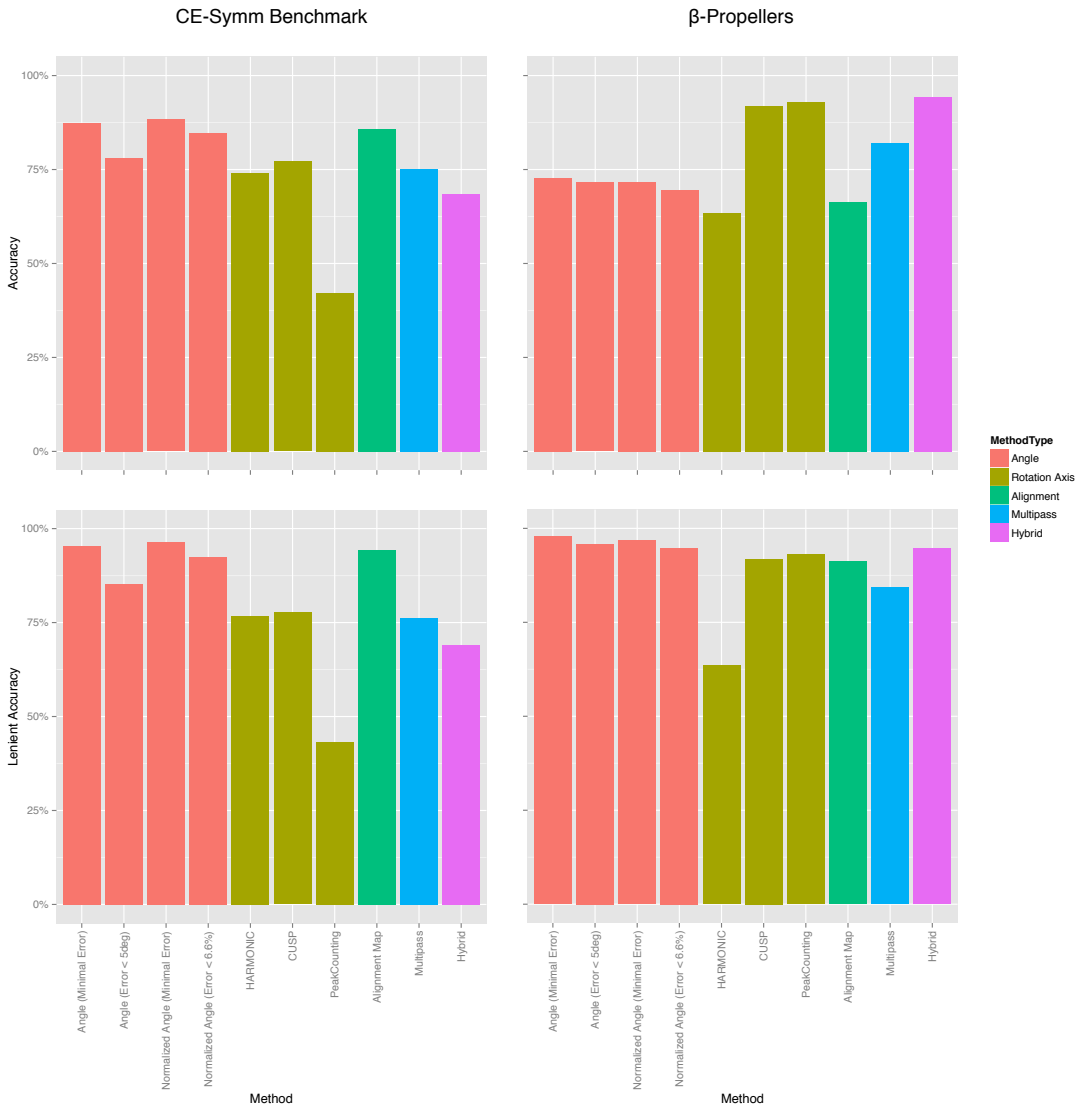


Figure 8.11: Comparison of the various order detection algorithms. Performance was assessed for symmetric, significant results from running CE-Symm on (left) symmetric structures from the CE-Symm benchmark, and (right) β-propeller structures annotated in SCOPe 2.01. Two metrics of accuracy are used: (top) *accuracy*, the percentage of structures reporting the correct order, and (bottom) *lenient accuracy*, the percentage of structures reporting a divisor of the correct order.

Methods based on the angle or rotation had the best overall performance. The angle method had the best performance when the order with minimal error from the ideal angle was used. Requiring that the observed angle be within a narrow window of the ideal angle only decreased the performance, even though most alignments do fall near the ideal. The normalization scheme from Figure 8.4 did not significantly impact the results. The angle methods had better performance on the CE-Symm benchmark than on the β-propellers. This may be explained by much higher average order of the propeller proteins. Since the angle order detector is limited to detecting the order of the top-scoring alignment, in a significant number of cases a divisor of the correct order is returned. This can only happen in high-order proteins with multiple factors, so it is more likely to occur in the propeller benchmark. This type of error is reflected in the very high lenient accuracy of all the angle methods, particularly on the propeller dataset.

The rotation axis methods performed similarly, with the best accuracy coming from fitting the cusp function and choosing the order with the highest amplitude. Measuring lenient accuracy gave little to no performance increase. This is expected since the methods depend on information from the full range of rotation, and are thus less likely to erroneously choose a particular sub-order at the expense of the global order. Instead, errors typically arise from noise in the distance function being fit. Errors in the placement of the rotation axis and deviations from the perfectly symmetric arrangement of repeats can mask the periodic signal in the average distance (e.g. Figure 8.7). Incorrectly assigned orders are thus not strongly biased towards divisors of the true order. The PeakCounting method was particularly prone to such errors, and often found one more or fewer peaks than should have been present according to the correct order. While optimizing the smoothing parameter in the LOESS regression could potentially improve the performance, the extreme difference in performance between the CE-Symm benchmark and the propeller dataset indicates that PeakCounting is unlikely to perform well on difficult cases with significant deviation from perfect symmetry.

It was hoped that the high lenient accuracy of the angle method could be combined with the insensitivity of the rotation axis method to the original alignment. This resulted

in a hybrid method, where a set of possible orders was determined based on the angle, followed by rotation axis analysis to determine the best scoring multiple of the base order. This was found to work well on the propeller benchmark, but failed on the unbiased benchmark. Inspection showed that this procedure generally overestimated the order.

The alignment map method, which was the primary method used in the original CE-Symm publication, performed well, with similar results to the angle-based method. The map is also limited to finding the order of the top-scoring alignment, so it has very high lenient accuracy. One additional benefit of the map method is that it is closely related to the algorithm for refining pairwise CE-Symm refinements into a consistent multiple alignment, so the order returned is particularly useful when such refinement is desired.

Surprisingly, the preliminary multipass method did not perform better than the other methods, despite requiring significantly more computation time. This appears to be due to significant noise in the scores for latter alignments for proteins with lower similarity between repeats. Comparing the performance using lenient accuracy showed that the error was not significantly biased to divisors of the true order. However, it is likely that with additional algorithmic development the additional information available to the multipass method could be used to create an improved order detector.

8.6 Conclusion

Determining the order of internal symmetry for a protein is an important goal for automated internal symmetry detection. Additionally, correctly determining this order is a prerequisite for the automatic decomposition of proteins into repeats. Thus, significant effort has been expended attempting to improve this step.

The existing alignment map method for order detection was among the best performing methods. The simple angle-based method also performed well. However, both these methods suffer from bias towards a divisor of the correct order because of their reliance on the initial pairwise CE-Symm alignment.

The rotation axis method was less sensitive to this bias, but had lower overall accuracy. Attempts to combine the methods have so far not resulted in a method with comparable accuracy to manual inspection. Future algorithmic advances for hybrid methods or novel multipass methods may lead to further improvements.

8.7 Acknowledgments

Many thanks to Douglas Myers-Turnbull for engaging discussions during the development of the alignment map method, and for the initial implementation of the angle method. Aleix Lafita implemented the multipass features of CE-Symm and contributed data for the associated figures. Figures were generated using PyMOL (Schrödinger, LLC, 2015), the ggplot library for R (Wickham, 2009; R Development Core Team), and Cytoscape (Shannon, 2003; Smoot et al., 2011).

8.8 Bibliography

N. K. Fox, S. E. Brenner, and J.-M. Chandonia. SCOPe: Structural Classification of Proteins–extended, integrating SCOP and ASTRAL data and classification of new structures. *Nucleic Acids Res*, 42(1):D304–9, Jan. 2014.

D. Myers-Turnbull, S. E. Bliven, P. W. Rose, Z. K. Aziz, P. Youkharibache, P. E. Bourne, and A. Prlić. Systematic Detection of Internal Symmetry in Proteins Using CE-Symm. *J Mol Biol*, 426(11):2255–2268, May 2014.

N. Nagano, C. A. Orengo, and J. M. Thornton. 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, Aug. 2002.

A. Poleksic. Optimal pairwise alignment of fixed protein structures in subquadratic time. *J Bioinform Comput Biol*, 9(3):367–382, June 2011.

R Development Core Team. R: A language and environment for statistical computing. Vienna, Austria.

Schrödinger, LLC. The PyMOL Molecular Graphics System, Version˜1.7. 2015.

P. Shannon. Cytoscape: A Software Environment for Integrated Models of Biomolecular Interaction Networks. *Genome Res*, 13(11):2498–2504, Nov. 2003.

M. E. Smoot, K. Ono, J. Ruscheinski, P.-L. Wang, and T. Ideker. Cytoscape 2.8: new features for data integration and network visualization. *Bioinformatics*, 27(3):431–432, Feb. 2011.

H. Wickham. *ggplot2*. Elegant Graphics for Data Analysis. Springer New York, 2009.

Chapter 9

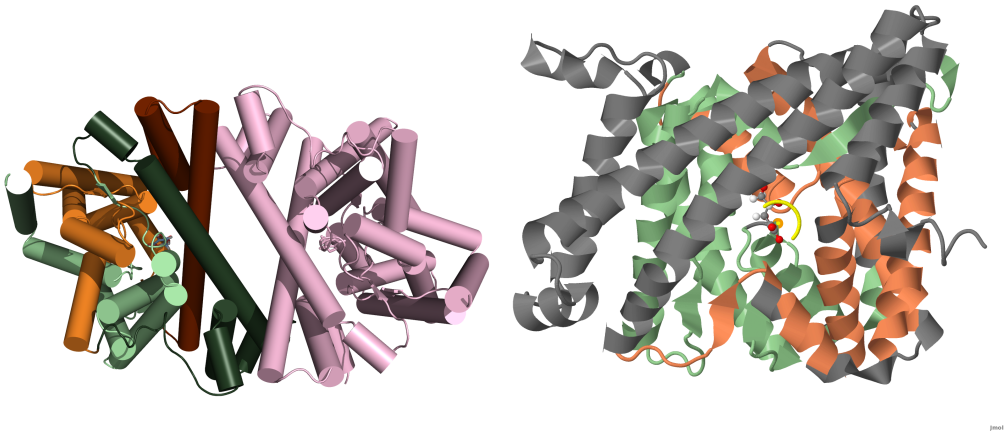
Conclusion

The various structural alignment algorithms developed here form a toolkit for analyzing protein fold space. These tools are useful both for detailed comparison of individual cases and for automated, systematic analysis of PDB-wide properties. Correlating structural similarity with protein function provides new insights into evolutionary pressures, physical folding constraints, and functional mechanisms.

The all-vs-all structural comparison network shows that fold space is highly connected at fold-level similarity thresholds, but that a more discrete nature appears at more stringent levels of similarity. This is consistent with the growing consensus that protein fold space has both continuous and discrete attributes. Mapping functional attributes such as transporter classifications shows how functions are conserved in a given fold. Identifying outliers where the function or structure is not conserved are particularly interesting, as they highlight areas of divergence and evolution. This could also be used to identify incorrect annotations for families with highly conserved functions.

Applying CE-Symm systematically shows that internal symmetry is quite common among proteins. However, the reasons for this are still unclear. Internal symmetry is tied to the function of many proteins, including binding ligands near the active site and increasing cooperativity between subunits. It may also have benefits for folding and thermostability, which could cause evolution to converge on symmetric structures.

Running CE-CP systematically for the whole PDB proved to be computationally infeasible. However, previous searches for circularly permuted protein pairs have found thousands (Lo et al., 2009), although this may be overinflated to to the presence of internal symmetry. Given additional resources, the combination of CE-CP and CE-Symm would be able to distinguish true circular permutations from other types of structural repeats.

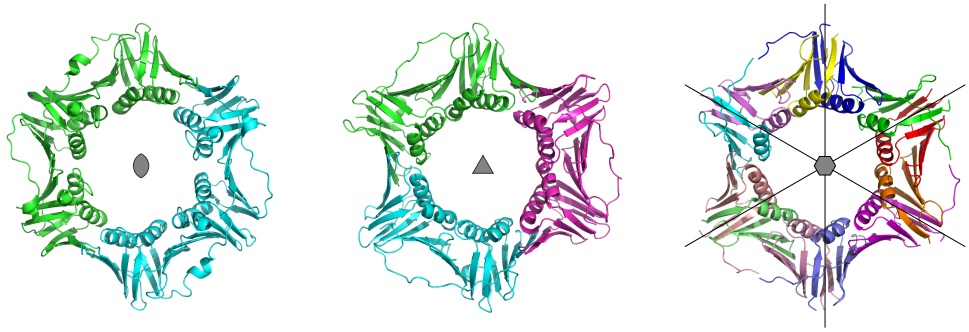


(a) (b)

Figure 9.1: Symmetry in the vitamin C transporter UlaA (Luo et al., 2015). (a) Overview of the dimer complex along the two-fold crystallographic axis. The left chain is colored to show internal pseudosymmetry between the V-motifs (dark green; dark red) and the core motifs (light green; orange). (b) CE-Symm alignment of one chain, shown from the dimerization interface along the 2-fold pseudosymmetry axis.

For individual proteins, deeper analysis is possible. Many membrane proteins exhibit internal and quaternary symmetry (Forrest, 2015). The inherent polarity imposed by the membrane may make symmetry more favorable by limiting the degrees of freedom for association. UlaA transports vitamin C in *E. coli* as part of the phosphoenolpyruvate-dependent phosphotransferase system (PTS). It is a symmetric dimer with the axis of symmetry perpendicular to the membrane (Figure 9.1a). However, each chain consists of an internal duplication of five transmembrane segments (TMS). The duplication of an odd number of helices leads to an inversion of the second repeat, giving a pseudosymmetry axis parallel to the membrane. While the full mechanism of transport in UlaA is not

well understood, it is interesting to speculate about how the ancestral protein might have functioned immediately following the five TMS duplication. This hypothetical ancestor would be symmetric across the membrane, suggesting that it would function as a passive transporter. Furthermore, the mechanism of transport would also be symmetric across the membrane. The halves of UlaA have diverged significantly (16% sequence identity; 49% similarity) and the existing structure has noticeable asymmetry in the position of the dimerization V-motif (gray portion of Figure 9.1b). However, without further experimental evidence a reasonable hypothesis for the mechanism of action could be to assume that the inward and outward conformations of the protein are similar, reflecting the vestiges of the original symmetry. This hypothesis would mean that the mechanism of transport involves the movement of the V-motifs across the ascorbic acid binding site to expose the ligand to the other side of the membrane. In the absence of a structure for the inner conformation, evidence for this hypothesis could even be gained computationally by building a homology modelling for UlaA using the CE-Symm alignment to invert the orientation of the protein in the membrane and analyzing the stability of the resulting structure.



(a) (b) (c)

Figure 9.2: Symmetric repeats in DNA clamps. (a) A dimeric bacterial clamp, DNA polymerase IIβ from *E. coli* [PDB:1MMI]. (b) A trimeric eukaryotic clamp, proliferating cell nuclear antigen from *H. sapiens* [PDB:1VYM]. (c) The trimer structure, colored to show the 12 structural repeats. Lines show the two-fold pseudosymmetry axes within each domain.

The orthogonal quaternary and pseudosymmetric axes in UlaA represent one

area of CE-Symm where manual intervention is still required. Such mixed quaternary and internal symmetry is common and can come about in intermediate stages between oligomeric and fully internally symmetric structures, or where duplications proceed in different ways in several species. One example of this phenomenon are the DNA clamps. Clamps consist of six copies of processivity fold that assemble in a ring around a DNA strand during replication. However, bacterial clamps consist of dimers with three domains per chain, while eukaryotic clamps are trimers with two domains per chain (Sippl and Wiederstein, 2012). Furthermore, each domain contains a twofold pseudosymmetry axis perpendicular to the DNA strand, for a total of 12 structural repeats arranged with D2 symmetry (Figure 9.2). The overall structure and function of DNA clamps is preserved across the full tree of life, but the exact order of duplications in prokaryotes and eukaryotes differs. Automated structural comparisons are as yet unable to detect such multi-level similarities. The ability to detect multiple symmetry axes and synthesis them into a picture of how biological assemblies evolve would be extremely powerful. CE-Symm provides a first step towards that goal.

Questions about the role of internal symmetry in the evolution of new protein folds remain. Although CE-Symm can detect internal symmetry with high accuracy, comparing repeats in proteins with different symmetry can best be done using single representatives of repeats. This is currently the focus of a project by Aleix Lafita to convert CE-Symm results into multiple structural alignments between the repeated subunits. His approach is based on the refinement procedure used by the alignment map order detector strategy, followed by optimization of the multiple alignment. Using this procedure, representative symmetric repeats could be compared using a similar methodology to the FATCAT comparison. With this, CE-Symm would be able to track changes in quaternary structure and internal symmetry together and allow quantification of the prevalence of duplications and rearrangements in the evolution of new folds.

Decomposing proteins into structural domains was essential for capturing the structural similarities between proteins, since domains are frequently reused between proteins as functional building blocks.(Campbell and Downing, 1994). In the same

way, the tools in this thesis could be used to identify the fundamental protodomains that evolution has reused in the evolution of folds. As a more distant goal, the repeats identified by CE-Symm and CE-CP could be combined with tools and concepts from phylogenetics to gain an even better understanding of the events that led to the current, wonderful variety of folds and symmetries found in the Protein Data Bank.

9.1 Acknowledgments

The UlaA example came out of discussions with Åke Vastermark and Milton Saier. The treatment of symmetry as multiple alignments was implemented by Aleix Latifa, who also wrote the improved visualization software used to create Figure 9.1b.

9.2 Bibliography

I. D. Campbell and A. K. Downing. Building protein structure and function from modular units. *Trends Biotechnol.*, 1994.

L. R. Forrest. Structural Symmetry in Membrane Proteins. *Annu Rev Biophys*, 44(1): 311–337, June 2015.

W.-C. Lo, C.-C. Lee, C.-Y. Lee, and P.-C. Lyu. CPDB: a database of circular permutation in proteins. *Nucleic Acids Res*, 37(Database issue):D328–32, 2009.

P. Luo, X. Yu, W. Wang, S. Fan, X. Li, and J. Wang. Crystal structure of a phosphorylation-coupled vitamin C transporter. *Nat. Struct. Mol. Biol.*, 22(3):238–241, Mar. 2015.

M. J. Sippl and M. Wiederstein. Detection of spatial correlations in protein structures and molecular complexes. *Structure*, 20(4):718–728, Apr. 2012.