RICE UNIVERSITY

Ab initio methods for protein structure prediction

by

Athanasios Dimitri Dousis

A Thesis Submitted
in Partial Fulfillment of the
Requirements for the Degree

Doctor of Philosophy

Approved, Thesis Committee:

Jianpeng Ma, Ph.D., Chair
Professor of Bioengineering
Lodwick T. Bolin Professor of
Biochemistry, Baylor College of Medicine

Michael Diehl, Ph.D.
Assistant Professor of Bioengineering
Assistant Professor of Chemistry

Bruce Johnson, Ph.D.
Distinguished Faculty Fellow of
Chemistry
Executive Director, Rice Quantum
Institute

Houston, Texas

July, 2010

ABSTRACT

Ab initio methods for protein structure prediction

by

Athanasios Dimitri Dousis

Recent breakthroughs in DNA and protein sequencing have unlocked many secrets of molecular biology. A complete understanding of gene function, however, requires a protein structure in addition to its sequence. Modern protein structure determination methods such as NMR, cryo-EM and X-ray crystallography are woefully unable to keep pace with automated sequencing techniques, creating a serious gap between available sequences and structures. This thesis describes several *ab initio* computational methods designed in the near-term to facilitate structure determination experiments, and in the long-term goal to predict protein structure completely and reliably. First, VecFold is a novel method for predicting the global tertiary structure topologies of proteins. VecFold applies fragment assembly to construct structural models from a target sequence by folding a chain of predicted secondary structure elements; these elements are represented either as C^{α}-based rigid bodies or as vectors. The knowledge-based energy function OPUS-Ca or a knowledge-based geometric packing potential is used to guide the folding process. The newest version of VecFold is demonstrated to modestly outperform Rosetta, one of the leading *ab initio* predictors, on the CASP8 benchmark set. In our protein domain boundary prediction method OPUS-Dom, VecFold generates a large ensemble of folded structure models, and the domain boundaries of each model are labeled by a domain parsing algorithm. OPUS-Dom

then derives consensus domain boundaries from the statistical distribution of the putative boundaries; the original version is also aided by three empirical sequence-based domain profiles. The latest version of OPUS-Dom outperformed, in terms of prediction sensitivity, several state-of-the-art domain prediction algorithms over various multi-domain protein sets. Even though many VecFold-generated structures contain large errors, collectively these structures provide a more robust delineation of domain boundaries. The success of OPUS-Dom suggests that the arrangement of protein domains is more a consequence of limited coordination patterns per domain arising from tertiary packing of secondary structure segments, rather than sequence-specific constraints. Finally, the knowledge-based energy function OPUS-Core was applied to the problem of protein folding core prediction, and it was shown to outpredict two leading computational methods on a benchmark set of 29 well-characterized protein targets.

Acknowledgments

First, I wish to thank my thesis advisor and graduate school mentor, Dr. Jianpeng Ma, for his invaluable wisdom, support and direction over the past six years. Dr. Ma is a brilliant scientist who leads by example through his hard work and drive to succeed, and I have grown a lot under his tutelage. I am grateful to Dr. Qinghua Wang and Dr. Pernilla Wittung-Stafshede for their guidance and help in my earlier collaborations with them, as well as my committee members Dr. Michael Diehl and Dr. Bruce Johnson for their helpful suggestions and insights regarding this thesis. I also want to thank Dr. Kyriacos Athanasiou, Dr. Ariel Fernandez, and Dr. Jack Gill for their kind advice over the years.

Dr. Ma often makes the point that he is just the conductor, and that it is his students who make the music. In the Ma Lab, I have had the privilege of working with several of the brightest young "scientific musicians" in the world. The list is long, but I want to specifically acknowledge a few individuals who I have worked with closely. First, Dr. Billy Poon helped me readjust to academic life as my office-mate for my first two years at Rice, and he also built the computer cluster on which I did a substantial amount of my research. Much of my research originated from the hard work of Dr. Yinghao Wu, Dr. Mingzhi Chen, and Jialin Li, and I deeply appreciate their continued guidance and help. Finally, Dr. Mingyang Lu has been an inspiration, a teacher, and a good friend throughout my time in graduate school.

The music that I have helped compose and play would go unheard if not for the incredible support staff at the Keck Center of the Gulf Coast Consortia, the Rice Department of Bioengineering, and the BCM Department of Biochemistry and Molecular Biology. Lisa Blinn, Karen Ethun, Melissa Glueck, and Deborah Miller have served the Houston computational biology community exceptionally well, and I am very grateful for their assistance. I also want to thank the Bioengineering Department Coordinator, Gayle Schroeder, for her help with administrative questions and issues that I have, as well as past coordinators Cindy Wilkes, Elaine Carrasco,

and Ginger Wright.

Furthermore, I could not have succeeded in my computational experiments without the outstanding support of the Rice Research Computing Support Group (RCSG), led by Dr. Kim Andrews. They provided the cutting-edge musical instruments and kept them working around the clock.

My research was financially supported in part by the National Library of Medicine Computational Biology and Medicine Training Program (NLM Grant No. 5T15LM07093) via the Keck Center of the Gulf Coast Consortia, and the National Science Foundation IGERT Program (Grant No. DGE-0114264) via the Rice Institute for Biosciences and Bioengineering. High-performance computer resources were provided by the Rice Terascale Cluster funded by NSF (Grant No. EIA-0216467), Intel, and HP, and the Shared University Grid at Rice funded by NSF (Grant No. EIA-0216467) and a partnership between Rice University, Sun Microsystems, and Sigma Solutions, Inc.

Finally, I am very thankful for the support and patience of my friends, especially Dr. Michael Wakin, despite my anti-social tendencies over the past two years. Most of all, I appreciate the love and unwavering support of my parents, Eleni and Dimitri, and my sister and roommate, Angelique. I dedicate this thesis to them.

Contents

Illustrations

Tables

Chapter 1

Background and significance

Many recent breakthroughs in DNA and protein sequencing have unleashed a torrent of sequence data, sowing new fields of science such as genomics and unlocking many secrets of molecular biology. A complete understanding of gene function, however, requires a protein structure in addition to its sequence. Protein structures and their databases are therefore vital for functional studies, as well as for practical applications such as development of pharmaceuticals and industrial enzymes. Despite major advances in biophysical techniques, structure determination methods such as NMR, cryo-EM and X-ray crystallography are woefully unable to keep pace with sequencing techniques, creating a serious gap between available sequences and structures. With new single-molecule sequencers on the horizon that are capable of reading a human genome for less than $1000 [96], the rate of new genomic data generation will only accelerate.

This enormous sequence-structure gap compels the use of computers to predict protein structure from amino acid sequence. In the four decades since Christian Anfinsen proposed that protein structure is determined by its amino acid sequence, and that proteins will fold into an ensemble of states at their free energy minimum [5], the field of protein structure prediction has been intensely active and has made substantial progress, as evidenced by the bi-annual Critical Assessment of Techniques for Protein Structure Prediction (CASP) competition [103]. At CASP, the structure prediction categories have grown to include tertiary structure, secondary structure, residue-residue contacts, domain boundaries, and so forth. Objective functions have evolved from purely physics-based quantum-chemical molecular mechanics force fields

to knowledge-based potential functions to hybrid potentials that incorporate physical models as well as statistics extracted from known structure databases. The most successful sampling methods are based on fragment assembly, which was first made popular by the Rosetta prediction suite [131, 117]. Even so, the structure prediction problem is far from solved.

Predicting three-dimensional protein structure, or tertiary structure, from a one-dimensional amino acid sequence is a fundamentally important challenge for science (and engineering, in the case of *de novo* protein design) and has been actively pursued for several decades. Comparative prediction methods such as homology modeling have already demonstrated much success, yet these are constrained by the size of the protein databases from which they draw their predictive power. *Ab initio* methods do not share this restriction, but because of the sheer size and physical complexity of the protein folding problem, these methods at best can predict low resolution structures of little more than 100 residues [52].

There are three general classes of protein structure prediction. The first is the class of *ab initio* methods, which apply physical principles to determine the native state of the protein. The second and third classes, comparative (homology) modeling [123] and fold recognition [109, 134], approach structure prediction through evolutionary rather than physical principles. Comparative modeling predicts structure by searching for similar amino acid sequences within a database of known structures. Fold recognition, on the other hand, uses a library of known fold motifs, also derived from a protein structure database, as a template of available folds for the target protein. Thus, both homology modeling and fold recognition are limited by the requirement that the native structure of the target sequence must have already been solved [170]. Homology modeling so far provides the most accurate models, yet it cannot predict new folds.

The lines between these classes are blurring rapidly, as complex new algorithms integrate physical principles, sequence and structure information, and experimental data to overcome the limitations of each separate class. Kolinski & Bujnicki [79]

combine fold recognition and *ab initio* folding methods, while Cheng & Baldi [26] use sequence homology and threading in their machine learning-based algorithm. Similarly, small-angle X-ray scattering (SAXS) data is incorporated into the structure prediction methods of Zheng & Doniach [173, 174] and Wu *et al* [162].

This thesis describes two related methods to predict protein tertiary structure, called VecFold and VecFold2, followed by two versions of OPUS-Dom, a domain prediction scheme that relies on structure models generated by VecFold or VecFold2. Finally, this thesis presents an application of the knowledge-based potential function OPUS-Core to protein folding core prediction. Before jumping into the methods, it is necessary to introduce some fundamental concepts of protein structure modeling and prediction.

1.1 *Ab initio* protein structure prediction

Ab initio protein structure prediction (i.e., protein folding) methods attempt to determine the tertiary structure "from scratch." The general assumptions in *ab initio* methods, which were originally postulated by Anfinsen [5], are that:

- the tertiary structure of a protein is uniquely determined by its amino acid sequence.
- the native ensemble of protein conformations will fall around a global free-energy minimum for that sequence [12].

The free-energy "landscape" is very large and rugged, and the process of folding is often described as a descent into a funnel-like free-energy well, where the protein eventually settles into a region of minimum energy.

Ab initio folding requires: (a) a representation of the protein molecule; (b) an objective (or scoring) function, typically in the form of a potential energy function from which a force field may be derived; and (c) a sampling method for searching the space of possible protein states.

1.2 Reduced complexity structure models

In terms of computational cost, an N-atom protein model suffers from two characteristics:

1. order N degrees of freedom available to the protein state function, which translates into an enormously large free energy landscape; and

2. order N^2 calculations to evaluate the free energy.

Together, these two factors make the conformation space search prohibitively expensive to compute for large molecules, thereby limiting the reach of all-atom *ab initio* methods.

Fortunately, cleverly-designed protein models can greatly reduce computational complexity with little loss in accuracy. One general coarse-graining strategy is to combine groups of atoms into single interaction centers, e.g., represent a multi-atom residue by a single pseudo-atom centered at the site of the alpha carbon. The second is to reduce the degrees of freedom, e.g., by eliminating stiff degrees of freedom such as bond lengths that have little effect on the global protein conformation (so discounting them makes the model simpler and also eliminates the need to compute very small time step motions) [54].

In addition, it is possible to apply a hierarchy of models to balance efficiency with resolution, as Oldziej *et al.* demonstrate when they search conformation space using a coarse-grained model, and then convert the lowest-energy coarse-grained structure into an all-atom model and refine it using an all-atom force field [105].

1.2.1 Lattice models

Lattice models restrict the geometry of the model to a subset of coordinates, thereby reducing the available states of the system and simplifying coordinate-based calculations such as the evaluation of energies. The coordinate constraints, however, also limit the ability of lattice models to correctly predict certain geometries like helices.

Kolinski & Skolnick apply a side-chain-only (SICHO) lattice model in TOUCHSTONE [80, 71], and Zhang *et al.* replace SICHO with a C^{α}, C^{β}, and side-group (CABS) model in TOUCHSTONE II [170]. Xia *et al.* [163] apply a hierarchical approach, such that the lattice becomes increasingly finer-grained as the algorithm iterates.

1.2.2 Discrete state off-lattice models

Discrete state off-lattice models fix certain degrees of freedom, typically bond lengths and side chain degrees of freedom [12]. In the UNRES model (Figure 1.1), Liwo *et al.* [88] represent a protein as a chain of C^{α} atoms with attached side chain pseudo-atoms. A peptide pseudo-atom is located in the center of each virtual bond connecting consecutive C^{α} atoms, and it serves as an interaction site along with the side chain pseudo-atoms, while the purpose of the C^{α} atoms is limited to defining the geometry. All the bond lengths are fixed, but the side-chain angles and bond angles are allowed to change.

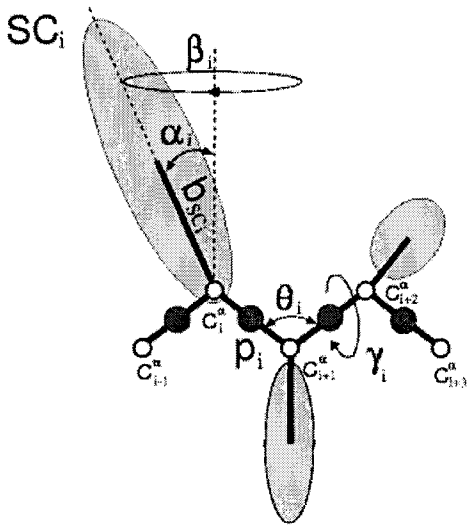

Figure 1.1 : The UNRES model [88].

The Rosetta model [131] identifies the structures of short amino acid subsequences, or fragments, using the PDB as a reference, and then assembles these structures using free energy optimization [42].

Another unique coarse-grained model was devised for the methods OPUS-PSP [91] and OPUS-Rota [90], in which the atoms of a residue are grouped into rigid body blocks based on their functional groups, as illustrated in Figure 1.2. This model is an all-atom model but with degrees of freedom limited to side-chain torsional angles.

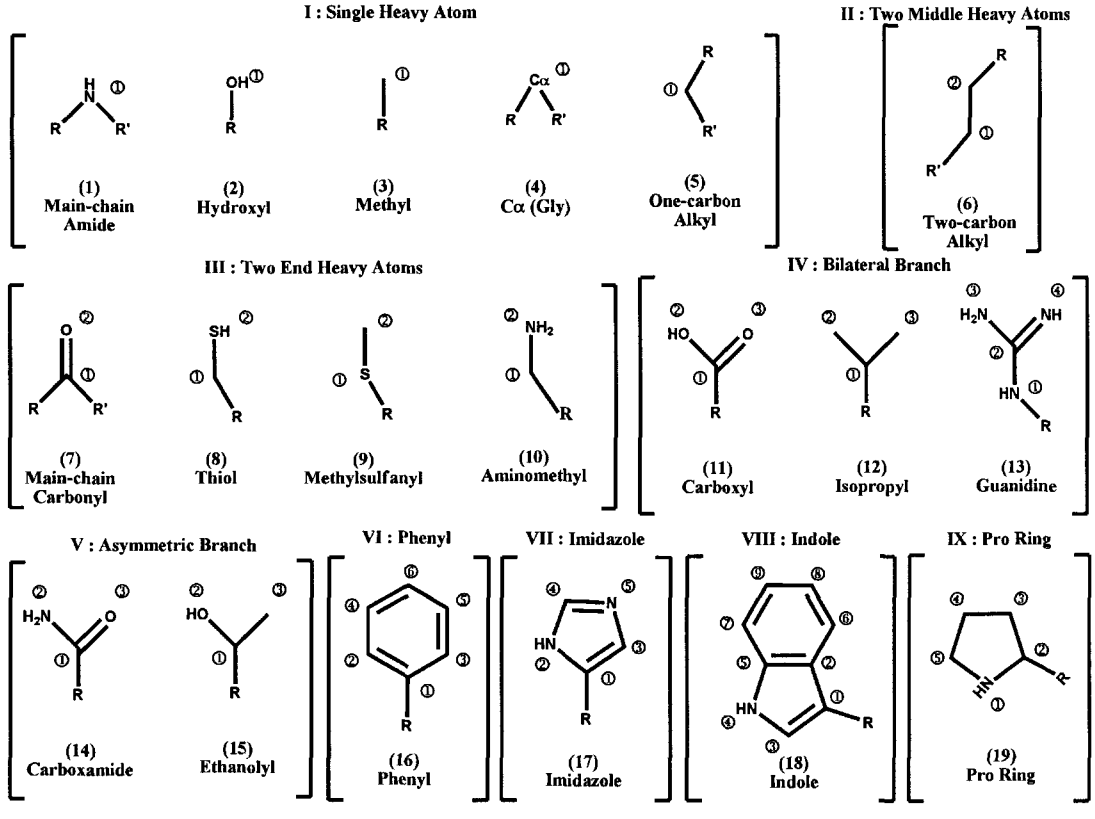

Figure 1.2 : The OPUS-PSP rigid body blockset [91].

1.3 Objective functions

Objective functions, also known as scoring functions, energy functions or force fields, fall on a spectrum from statistics-based (knowledge-based) to physics-based potential

functions. Knowledge-based objective functions infer structural information from a database of known protein structures, whereas physics-based scoring functions are derived from first principles, although they may be parameterized using protein databases. Objective functions are composed of several terms that model various atomic interactions, such as steric, electrostatic, hydrogen bonding, and hydrophobic effects. Typically, a potential function includes non-bonded interaction energies (e.g., van der Waals, electrostatic) and bonded interaction terms (e.g., bond length, bond angle, dihedral angle).

1.3.1 Physics-based potential functions

All-atom physics-based potential functions

Among the most commonly used all-atom force fields are CHARMM (Chemistry at HARvard Macromolecular Mechanics) [93], AMBER (Assisted Model Building and Energy Refinement) [33], ECEPP [102], and GROMOS (GROningen MOlecular Simulation) [125].

Coarse-grained physics-based potential functions

A wide variety of coarse-grained force fields, often partly derived from the all-atom force fields, are used in structure prediction. One of the first such force fields is UNRES [88] by the Scheraga group. The energy of the side chain-backbone model is expressed as

$$\begin{aligned} U = & \sum_{i<j} U_{SC_i SC_j} + \sum_{i \neq j} U_{SC_i p_j} + w_{el} \sum_{i<j-1} U_{p_i p_j} \\ & + w_{\text{tor}} \sum_i U_{\text{tor}}(\gamma_i) + w_{\text{loc}} \sum_i [U_b(\theta_i) + U_{\text{rot}}(\alpha_{SC_i}, \beta_{SC_i}] \\ & + w_{\text{corr}} U_{\text{corr}} \end{aligned}$$

where $U_{SC_i SC_j}$ is a side-chain hydrophobic interaction term, $U_{SC_i p_j}$ is an excluded-volume term for side-chain–peptide-group interactions, $U_{p_i p_j}$ is a peptide group in-

teraction potential corresponding to backbone hydrogen bond formation, U_{tor}, U_b, and U_{rot} are local terms accounting for virtual-dihedral angle torsions, virtual-angle bending, and side-chain rotamers, U_{corr} is a multibody correlation term, and the w's are the weights.

TOUCHSTONE II [170] employs a 19-parameter force field encapsulated in 10 energy terms

$$E = E_{\text{short}} + E_{\text{stiffness}} + E_{\text{HB}} + E_{\text{pair}} + E_{\text{burial}} + E_{\text{electro}} + E_{\text{profile}} + E_{\text{COCN}} + E_{\text{distmap}} + E_{\text{contact}}$$

representing short range interactions (E_{short}), local conformational stiffness ($E_{\text{stiffness}}$), hydrogen bonds (E_{HB}), local distance restraints (E_{distmap}), long-range pairwise interactions (E_{pair}), burial interactions (E_{burial}), electrostatic interactions (E_{electro}), the contact environment of individual residues (E_{profile}), contact order and contact number (E_{COCN}), and tertiary contact restraints (E_{contact}).

1.3.2 Knowledge-based potential functions

At the heart of knowledge-based, or statistical, potential functions is the assumption that the distribution of native structures, in particular their structural features and interaction patterns, is related to the canonical ensemble in statistical mechanics [126, 132]. In other words, we can relate the probability of a structural interaction pattern to an energy function by the Boltzmann relation:

$$E(\mathbf{x}) = -k_B T \ln \left(\frac{p^{\text{obs}}(\mathbf{x})}{p^{\text{ref}}(\mathbf{x})} \right)$$

where $k_B T$ is the product of the Boltzmann constant and Boltzmann temperature, $\mathbf{x}$ is some arbitrary state variable representing an interaction pattern, p^{obs} is the probability of observing that interaction pattern in a database of known interaction patterns or structures, and p^{ref} is a non-trivial reference state probability.

An example of a statistical potential function is OPUS-PSP [91], which was developed for side-chain modeling. At the core of this objective function is a simple

expression:

$$E(\Omega_{ab}, a, b) = -k_B T \ln \left(\frac{p^{\text{obs}}(\Omega_{ab}, a, b)}{p_{\text{ref}}(\Omega_{ab}, a, b)} \right)$$

where Ω_{ab} represents a relative orientation of two rigid body block types a and b.

1.4 Sampling methods

In protein structure prediction, sampling methods such as Monte Carlo are used to search state space for conformations of interest, specifically those that minimize the global free energy.

1.4.1 Molecular Dynamics

In Molecular Dynamics, for each state of a molecular system, the force field is used to compute the forces on each degree of freedom of the system, and Newton's equations of motion are integrated for a single time step to yield a new state at the new time point. This process is repeated until the desired measurements are complete.

1.4.2 Monte Carlo sampling

The Monte Carlo algorithm consists of a random walk and an acceptance criterion, typically in the form introduced by Metropolis & Ulam [100] and further developed by Metropolis *et al.* [99]. The random walk may be biased (e.g., by normal mode analysis) or constrained (e.g., to a lattice). For a computed energy difference $\Delta E = E(\mathbf{r}_{\text{new}}) - E(\mathbf{r}_{\text{old}})$, the Metropolis acceptance rule is:

$$\text{acc}(\text{old} \to \text{new}) = \begin{cases} e^{-\Delta E/T} & \Delta E \geq 0 \\ 1 & \Delta E < 0 \end{cases}$$

where T is the Boltzmann temperature of the system. In other words, we always accept moves that result in a lower score, or energy, and accept "uphill" moves with a probability that decreases exponentially for more positive energy differences.

Simulated annealing [76] may be applied to improve the global minimization process. Annealing in simulation is conceptually similar to its physical counterpart: when annealing a physical system, the system is first heated, and then it is very gradually allowed to cool, allowing the system to initially sample states of high energy (avoid getting trapped in local minima) and eventually settle into the global energy minimum.

Rosetta [131] employs Metropolis Monte Carlo sampling, whereas TOUCHSTONE II [170] uses the parallel hyperbolic sampling (PHS) algorithm of Zhang *et al* [169], which is a Monte Carlo scheme that "logarithmically flattens" local high-energy barriers to allow the simulation to traverse the energy landscape more efficiently while preserving the local energy minima.

1.5 Important structure databases

Template libraries, or fold libraries, are the set of motifs used to match sequence to structure. Among the most common template libraries are the SCOP (Structural Classification of Proteins) fold library [104], the FSSP/Dali (Distance matrix alignment) Domain Classification library [58], and the CATH (Class, Architecture, Topology, Homologous superfamily) Domain Structure Database [108].

1.6 Organization of thesis

The rest of this thesis is organized into six chapters, divided into four parts. Each of the next five chapters contains two sections, one describing a method for protein structure prediction and a second summarizing the results of the corresponding computational experiments. The first part describes the tertiary structure prediction schemes VecFold1 and VecFold2 in Chapters 2 and 3. Part II, consisting of Chapters 4 and 5, describes two versions of the domain prediction method OPUS-Dom, which are based on VecFold. Part III describes the application of the potential function OPUS-Core to the prediction of protein folding cores. The last part is a concluding

discussion of the work in the first three parts. The appendix contains a glossary and mathematical definitions and notations used in this document.

Part I

Tertiary structure prediction

Tertiary structure predictors can be broadly divided into template-based (comparative modeling and threading) and template-free (*ab initio*) methods, in addition to meta-predictors that can span both categories. Successful tertiary structure predictors at CASP7 and CASP8 include Rosetta [131], pro-Sp3-TASSER [176], MUFOLD [168], MULTICOM [153], and I-TASSER ("Zhang-Server" at CASP) [158]. One important characteristic of the top CASP predictors is that they incorporate multiple techniques into a single package. These predictors range from fully-integrated software suites such as Rosetta and MUFOLD to meta-predictors such as METATASSER [175] that apply several independent methods separately and then find a consensus prediction from the various results.

For cases where the sequence is short or sequence homology is high, the low-resolution structure prediction problem is mostly solved [172, 72, 133]. The challenge lies in template-free modeling, especially of large proteins. VecFold1 [160] and VecFold2 were developed specifically for template-free modeling. Inspired by Rosetta and FRAGFOLD [64], the VecFold methods assemble multi-secondary-structure-element fragments, called supersecondary structure motifs (SSSMs), and apply simulated annealing to generate compact tertiary structures.

Chapter 2

VecFold1

VecFold is predicated on the idea that conformation space may be sampled more broadly and rapidly by limiting our initial move-set to secondary structure elements (SSEs). The SSEs are essentially treated as rigid body elements. We extract small contiguous sets of SSEs from the structure database to populate our fragment library. The fragments have the form "XLX...LX", where "L" represents a loop and "X" represents an α-helix or β-strand. The fragments replace regions of the query chain that correspond to the predicted secondary structure.

In the following section, we first describe the vector geometry of VecFold1. Then, we explain the simulated annealing Monte Carlo scheme used to fold vector models into compact tertiary structures. Finally, we introduce the non-redundant structure database used for the geometric scoring function and SSSM (super secondary structure motif) fragment library described previously.

2.1 Methods

2.1.1 Structure model: vector representation of protein conformation

In the VecFold1 algorithm, a protein is modeled as a highly coarse-grained chain of vectors representing secondary structure elements (SSEs) [159], such that no atomic coordinates are needed during folding. For an α-helix, the vector is defined on the primary axis of the helix. The direction is calculated from the center of the first four alpha carbons (C^{α}) in the helix to the center of the last four C^{α} atoms in the helix. The length of the helix vector is $(N_{\alpha} - 1) \times 1.5$ Å, where N_{α} is the number of residues in the helix. The midpoint of the vector resides on the helix center of mass. Each β-

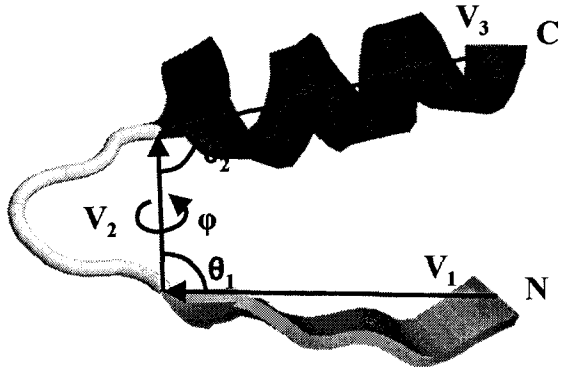

Figure 2.1 : Geometry of two consecutive secondary structures connected by a loop. An α-helix is represented by a vector along the cylindrical axis of the helix directed from the N terminus to the C terminus. For a loop or β-strand, the vector runs from the first C^{α} atom to the last C^{α} atom of the loop or strand. For any three vectors, we define the packing angle θ_1 with respect to vectors $\mathbf{v}_1$ and $\mathbf{v}_2$, the packing angle θ_2 with respect to vectors $\mathbf{v}_2$ and $\mathbf{v}_3$, and the dihedral angle ϕ with respect to all three vectors $\mathbf{v}_1$, $\mathbf{v}_2$, and $\mathbf{v}_3$.

strand is represented by two vectors, an axial vector that connects the two terminal C^{α} atoms of the strand, and a normal vector describing the orientation of the strand. For a loop, the vector connects the end-points of the two adjacent SSE vectors. With this vector representation, illustrated in Figure 2.1, a protein conformation is described by a set of local internal coordinates: vector lengths, packing angles between pairs of adjacent vectors, and dihedral angles for sets of three adjacent vectors.

2.1.2 Objective function: geometric packing potential

In order to evaluate the tertiary structure packing quality of VecFold1 vector models, a scoring function based on the statistical distribution of different types of long-range SSE packing is derived from the non-redundant structural database. The geometric scoring function E_{Geometry} consists of four terms:

$$E_{\text{Geometry}} = E_{\text{HH}} + E_{\text{SS}} + E_{\text{SH}} + E_{\text{Rg}} \tag{2.1}$$

where E_{HH}, E_{SS}, and E_{SH} are helix-helix, strand-strand, and strand-helix packing terms, respectively, and E_{Rg} is a radius of gyration term.

For the helix-helix packing term E_{HH}, two helices are considered in contact if there is at least one pair of side-chain atoms, one atom on each helix, within the 5Å cutoff distance. E_{HH} consists of three terms representing the packing distance d_{ij}, the packing angle ϕ_{ij}, and the coupling effect of helix lengths L_i and L_j:

$$E_{\mathrm{HH}}(d_{ij},\phi_{ij},L_i,L_j) = -RT\ln\frac{N_{\mathrm{bin}}^{d}N_{\mathrm{obs}}(d_{ij})}{\sum_{d_{k\ell}}N_{\mathrm{obs}}(d_{k\ell})} - RT\ln\frac{N_{\mathrm{bin}}^{\phi}N_{\mathrm{obs}}(\phi_{ij})}{\sum_{\phi_{k\ell}}N_{\mathrm{obs}}(\phi_{k\ell})} - RT\ln\frac{N_{\mathrm{bin}}^{d,\phi,L}N_{\mathrm{obs}}(d_{ij},\phi_{ij},L_i,L_j)}{\sum_{d_{k\ell},\phi_{k\ell},L_k,L_\ell}N_{\mathrm{obs}}(d_{k\ell},\phi_{k\ell},L_k,L_\ell)}. \quad (2.2)$$

Here, N_{obs} is the number of observed occurrences of a specific coarse-grained parameter (e.g., d_{ij}) in the non-redundant structural database, and $N_{\mathrm{bin}}^{\{d,\phi,L\}}$ is the number of bins for a set or subset of parameters $\{d,\phi,L\}$. For helix-helix interactions, the distance d_{ij} is divided evenly into 30 bins from 0 to 15Å (distances beyond the cutoff are not considered in contact and therefore ignored), and the dihedral angle ϕ_{ij} is divided into 36 equal bins spanning -180^o to 180^o. For the coupling term, the helix length L is divided into 3 types: *short* (less than 15 residues), *medium* (15 to 22 residues) and *long* (more than 22 residues). Figure 2.2(a) illustrates the definitions of d_{ij}, ϕ_{ij}, L_i and L_j. For helices H_i and H_j defined by helix vectors $\mathbf{v}_i$ and $\mathbf{v}_j$, d_{ij} equals the transversal distance (i.e., the minimal distance between the two vectors) [47], ϕ_{ij} is the dihedral angle between $\mathbf{v}_i$ and $\mathbf{v}_j$, and $L_i = \|\mathbf{v}_i\|_2$ and $L_j = \|\mathbf{v}_j\|_2$ are the lengths of the helices. Figures 2.3(a) and (b) show the packing score as a function of helix-helix packing angle or distance, and the minima in the packing angle curve (Figure 2.3(a)) correspond to the orthogonal and up-down packing preferences.

For the strand-strand packing term E_{SS} in Equation 2.1, two strands are considered in contact if there is at least one hydrogen bond between the backbones of these two strands. Like E_{HH}, E_{SS} includes three terms representing the packing (transversal) distance d_{ij}, packing angle ϕ_{ij}, and the coupling effect of normal angles β_i and

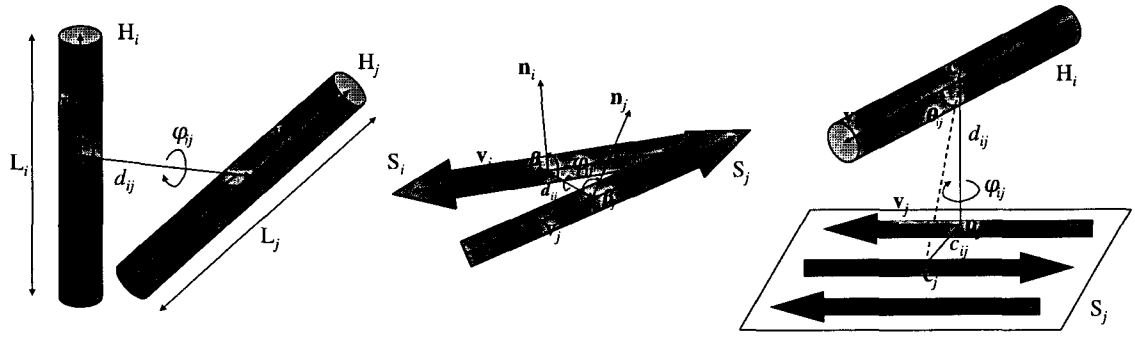

Figure 2.2 : Schematic illustration of secondary-structure packing geometry and geometric parameters used to describe the packing. (a) helix-helix packing; (b) strand-strand packing; (c) helix-sheet packing.

β_j:

$$E_{\mathrm{SS}}(d_{ij},\phi_{ij},\beta_i,\beta_j) = -RT\ln\frac{N^d_{\mathrm{bin}}N_{\mathrm{obs}}(d_{ij})}{\sum_{d_{k\ell}}N_{\mathrm{obs}}(d_{k\ell})} - RT\ln\frac{N^\phi_{\mathrm{bin}}N_{\mathrm{obs}}(\phi_{ij})}{\sum_{\phi_{k\ell}}N_{\mathrm{obs}}(\phi_{k\ell})} - RT\ln\frac{N^{d,\phi,\beta}_{\mathrm{bin}}N_{\mathrm{obs}}(d_{ij},\phi_{ij},\beta_i,\beta_j)}{\sum_{d_{k\ell},\phi_{k\ell},\beta_k,\beta_\ell}N_{\mathrm{obs}}(d_{k\ell},\phi_{k\ell},\beta_k,\beta_\ell)}. \quad (2.3)$$

The transversal distance d_{ij} is divided equally into 60 bins spanning 0 to 6Å, and the dihedral angle ϕ_{ij} is divided equally into 36 bins spanning -180° to 180°. For the coupling term, the angle β between the normal vector and the transversal is equally divided into 18 bins spanning 0° to 180°, d_{ij} is divided equally into 3 bins spanning 0 to 6Å, and ϕ_{ij} is equally divided into 12 bins spanning -180° to 180°.

Figure 2.2(b) illustrates the definitions of d_{ij}, ϕ_{ij}, β_i and β_j. In addition to the direction vectors $\mathbf{v}_i$ and $\mathbf{v}_j$, strands S_i and S_j require normal vectors $\mathbf{n}_i$ and $\mathbf{n}_j$ (perpendicular to the sheet plane) to define their orientations. Figures 2.3(c) and (d) show the packing score as a function of strand-strand packing angle or distance, and the minima in the packing angle curve (Figures 2.3(c)) correspond to the parallel and anti-parallel strand-strand packing preferences.

For the strand-helix packing term E_{SH}, the helix and strand are in contact if there

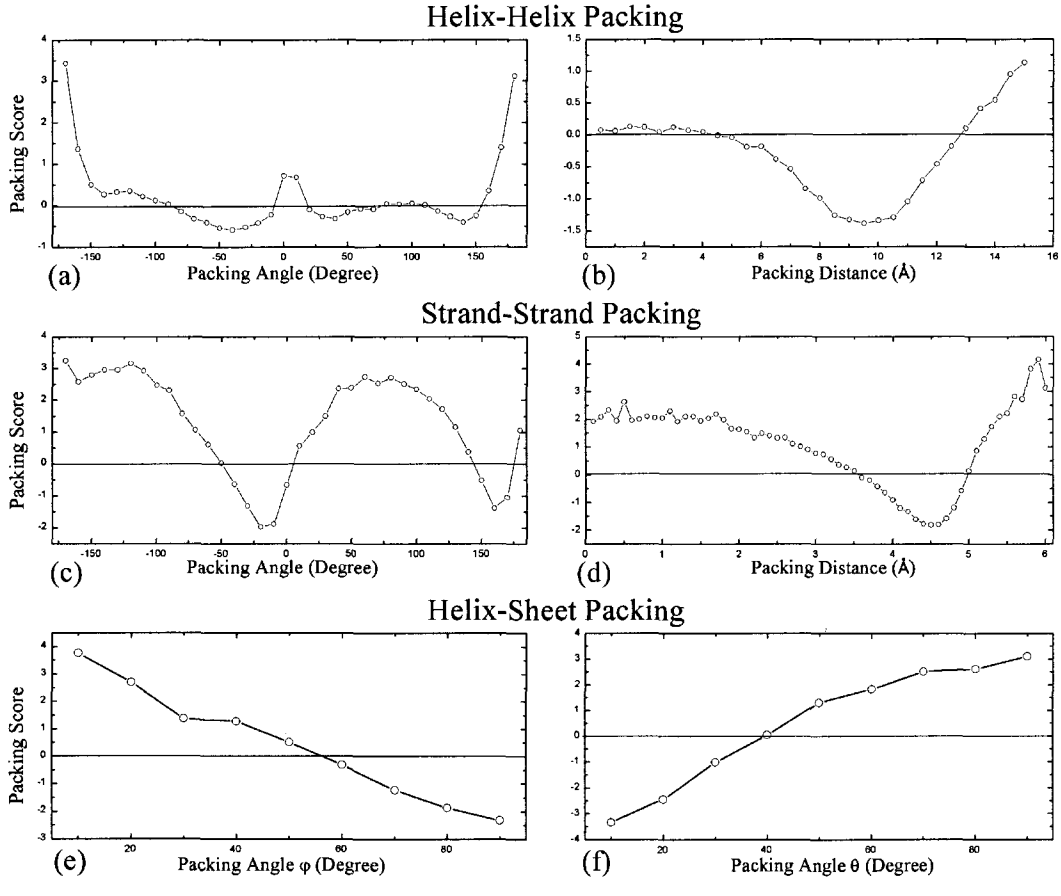

Figure 2.3 : Statistical behavior of the packing scoring function: for helix-helix packing with respect to the packing angle (a) and packing distance (b); for strand-strand packing with respect to the packing angle (c) and packing distance (d); for helix-sheet packing with respect to the packing angle ϕ (e) and packing angle θ (f).

is at least one pair of side-chain atoms, one atom on the helix and the other on the sheet region, within the 5Å distance cutoff. Strand-helix packing is illustrated in Figure 2.2(c), where $d_{ij} = \|\mathbf{c}_i - \mathbf{p}_j\|_2^2$ is the distance between the center of mass $\mathbf{c}_i$ of helix H_i and its projection $\mathbf{p}_j$ onto the plane of sheet S_j, θ_{ij} is the angle between the helix vector $\mathbf{v}_i$ and the vector represented by d_{ij} (connecting $\mathbf{c}_i$ and $\mathbf{p}_j$), $c_{ij} = \|\mathbf{c}_j - \mathbf{p}_j\|_2^2$ is the distance between the center of mass $\mathbf{c}_j$ of the sheet plane and $\mathbf{p}_j$, and ϕ_{ij} is the dihedral angle between consensus sheet vector $\mathbf{v}_j$ and $\mathbf{v}_i$ (i.e., the angle between $\mathbf{v}_j$ and the vector projection of $\mathbf{v}_i$ in the sheet plane). The strand-helix packing energy also includes three terms corresponding to θ_{ij}, ϕ_{ij}, and the coupling effect of distances c_{ij} and d_{ij}:

$$\begin{aligned} E_{\mathrm{SH}}(\theta_{ij}, \phi_{ij}, c_{ij}, d_{ij}) = &- RT \ln \frac{N^{\theta}_{\mathrm{bin}} N_{\mathrm{obs}}(\theta_{ij})}{\sum_{\theta_{k\ell}} N_{\mathrm{obs}}(\theta_{k\ell})} - RT \ln \frac{N^{\phi}_{\mathrm{bin}} N_{\mathrm{obs}}(\phi_{ij})}{\sum_{\phi_{k\ell}} N_{\mathrm{obs}}(\phi_{k\ell})} \\ &- RT \ln \frac{N^{c,d}_{\mathrm{bin}} N_{\mathrm{obs}}(c_{ij}, d_{ij})}{\sum_{c_{k\ell}, d_{k\ell}} N_{\mathrm{obs}}(c_{k\ell}, d_{k\ell})}. \end{aligned} \tag{2.4}$$

Figures 2.3(e) and (f) show the packing score as a function of angle ϕ_{ij} or θ_{ij}. The angles θ_{ij} and ϕ_{ij} are each equally divided into 9 bins spanning 0^o to 90^o. For the coupling term, the distances c_{ij} and d_{ij} are each equally divided into 30 bins spanning 0 to 30Å.

In our study, we use a harmonic scoring function based on the radius of gyration R_g to promote compactness in the global shape of the VecFold1 models:

$$E_{R_g} = \frac{\epsilon}{2} \left(\frac{R_{g,\mathrm{Calculated}} - R_{g,\mathrm{Empirical}}}{\sigma} \right). \tag{2.5}$$

where weight parameter ϵ and standard deviation σ are set to 10.0Å and 5.0Å, respectively, based on the statistical data, $R_{g,\mathrm{Calculated}}$ is the calculated radius of gyration, and $R_{g,\mathrm{Empirical}}$ is the radius of gyration of an N-residue protein that obeys the empirical expression by Bahar & Jernigan [48]:

$$\log(R_{g,\mathrm{Empirical}})^2 = \frac{2}{3} \log N + 0.92 \tag{2.6}$$

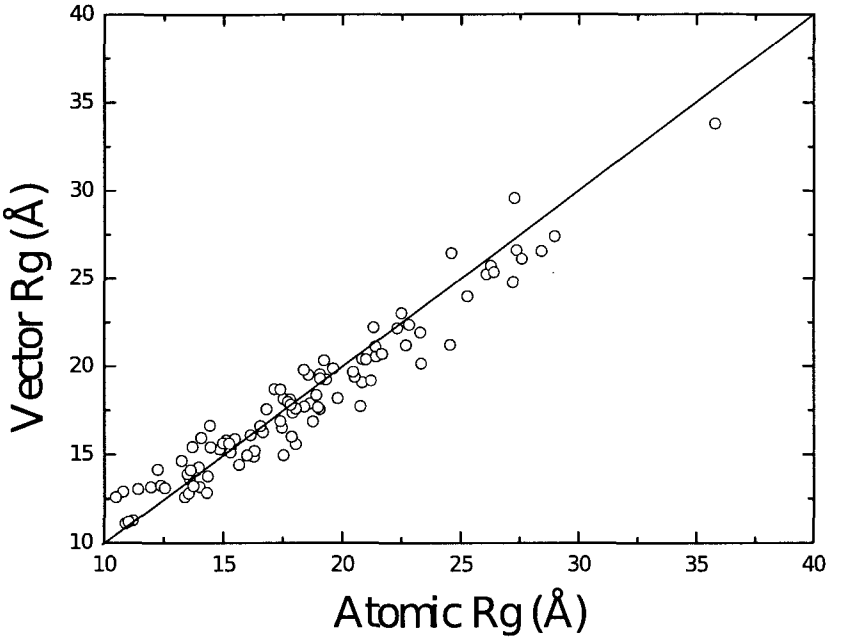

Figure 2.4 : Correlation between protein radii of gyration calculated by the vector-based and traditional atom-based methods. The correlation coefficient is 0.936.

For a series of consecutive vectors packed in space, the radius of gyration may be calculated as:

$$R_{g,\text{Calculated}} = \sqrt{\frac{1}{N}\sum_i n_i \quad \frac{v_i^2}{12 + \|\mathbf{r}_{\text{MC}} - \mathbf{r}_{m,i}\|_2^2}} \tag{2.7}$$

where N is the number of residues in the query protein, i is the index over all vectors $\{\mathbf{v}_i\}$, n_i is the number of residues in $\mathbf{v}_i$, $\mathbf{r}_{\text{MC}}$ is the protein center of mass, and $\mathbf{r}_{m,i}$ is the midpoint of $\mathbf{v}_i$. We choose 100 test proteins, given in Table 2.1, to validate the accuracy of vector-based $R_{g,\text{Calculated}}$. These test proteins are well-packed and lack long disordered regions. We calculate the radius of gyration for these proteins using both the vector approach and the traditional atomic method, and we plot these values in Figure 2.4. The strong correlation between these two methods (correlation coefficient 0.936) suggests that our vector-based method approximates the traditional method well.

1BVC 0	153	1GVN A	87	2CRO 0	65	1XLY A	224	1J5X A	312
1CTF 0	68	1UPK A	310	1OO0 A	144	1T82 A	138	1SBY A	254
1W2W A	207	1VM0 A	89	1DP7 P	76	1PUO A	141	1NZ0 A	108
1VHY A	237	1U00 A	227	1K94 A	165	1R0U A	142	1KKO A	401
1JYO A	130	1V74 B	87	1K8U A	87	1IOM A	374	1IZM A	168
1GKM A	509	1GS9 A	144	1R0D A	189	1YX1 A	248	1U0F A	556
1XDZ A	238	1T06 A	235	1ALU 0	157	1YM3 A	193	1XPP A	99
1EW6 A	137	1WWJ A	99	1GL2 C	60	1J5W A	270	1PYF A	309
1MQV A	123	1SMB A	147	1GL2 D	55	1Y6X A	86	1Y6Z A	242
1MG7 A	351	1JYS A	226	1HH8 A	192	1OYJ A	228	2TRX A	108
1UEK A	268	1MR8 A	90	1WHZ A	67	1TU9 A	131	1UIX A	68
1Y9W A	140	2EBO A	74	1YB3 A	163	1S7Z A	101	1DLW A	116
1IO7 A	366	1E2K A	308	1W07 A	655	1RY9 A	133	1XTE A	116
1W2Y A	226	1DOW A	200	1T7R A	250	2MLT A	26	1VMG A	80
1P0K A	306	1VLS 0	146	1S4B P	654	1VME A	394	1N97 A	385
1VPD A	279	1NFV A	169	1TFE 0	142	1OK7 A	366	1TQG A	105
1R7J A	90	1OXX K	352	1TU7 A	208	1LS1 A	289	1NRI A	240
1RJ1 A	148	1U02 A	222	1N1J B	78	1N8V A	101	1S12 A	94
1EVY A	346	1NG6 A	148	1JDH A	508	1IDP A	147	1QJP A	137
1X6I A	89	1IRQ A	48	1JL1 A	152	1DPJ B	29	1KL1 A	405

Table 2.1 : Test set for radius of gyration validation. The entries in the table include the PDB code (4 characters), domain ID (1 character), and domain size (in number of residues)

2.1.3 Sampling method: super-secondary structure motif (SSSM) vector fragment assembly

An SSSM is a mesoscale protein fragment consisting of three consecutive SSEs and their two connecting loops. VecFold1 folds protein tertiary structures by drawing fragments as Monte Carlo moves from an ensemble of SSSM candidates.

Prior to executing VecFold1, an SSSM template library is constructed from the non-redundant structure database. Each structure is divided into overlapping SSSM windows based on the SSEs identified in the RCSB Protein Data Bank records, and an SSE vector model is built to match the structure. In parallel, a position-specific sequence profile is generated from the query sequence by three iterations of PSI-BLAST with an E-value below 0.001 (the profile is normalized to unity at each residue position). The sequence and secondary structure profiles and SSSM vector coordinates (vector lengths, packing angles, and dihedrals) are recorded in the template library for each SSSM.

To generate an ensemble of SSSM candidates, VecFold1 first generates a secondary structure profile from the query sequence using PSI-PRED [63], and then it separates the resulting P SSEs into $P-2$ SSSM windows, as is illustrated in Figure 2.5. Next, just as for the template library, a PSSM-based sequence profile is generated from the query sequence by three iterations of PSI-BLAST with an E-value below 0.001 (the profile is again normalized to unity at each residue position). The structural candidates for each query SSSM window are generated by aligning the sequence profile and secondary structure information to all SSSMs in the template library. We use Smith-Waterman local affine-gap dynamic programming to align the profiles [137]. The gap opening and gap extension parameters are -4.5 and -0.5, respectively.

For sequence position i and template position j, the alignment score consists of two terms, the first related to sequence profiles and the second related to secondary

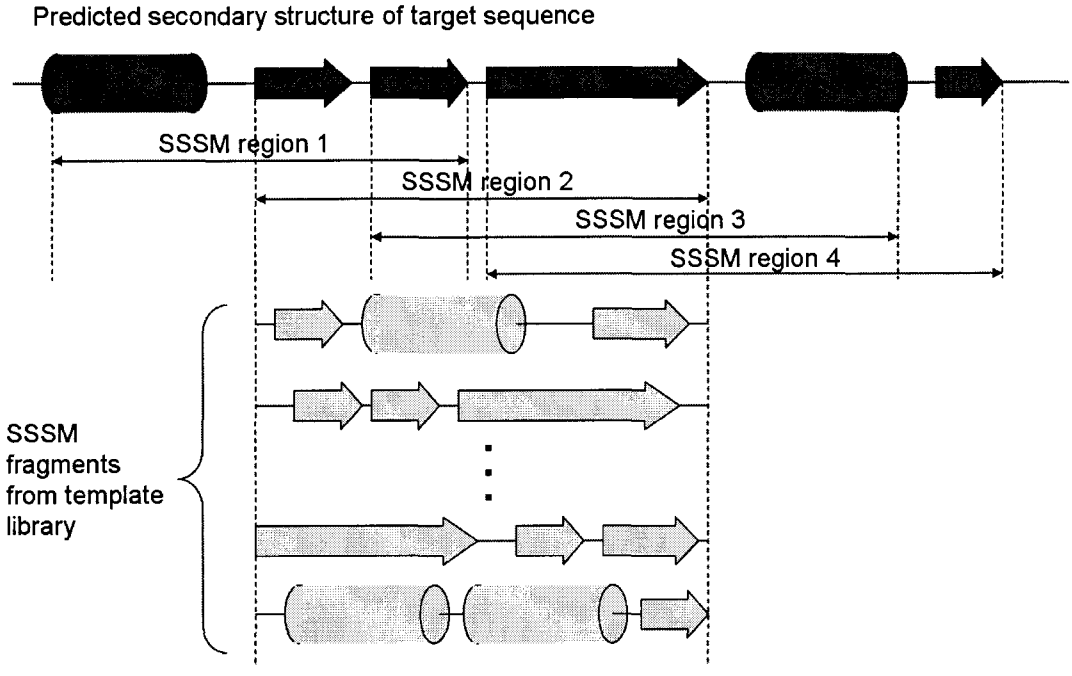

Figure 2.5 : SSSM fragment generation in VecFold1. First, PSI-PRED predicts the secondary structure of the target sequence, and this information is used to parse the target into SSSM regions or windows. Then for each SSSM window, fragments that contain only 3 non-loop SSEs and fit within the sequence length of the window are extracted from a template library of non-redundant structures, and each fragment is then aligned to the window and scored by Equation 2.8. The 100 best scoring fragments for each window are stored in a fragment library for use during fragment assembly.

structures:

$$S_1(i,j) = -w_{\text{seq}}(\mathbf{F}^{\text{seq}}_{\text{query}}(i))^{\text{T}}\mathbf{M}^{\text{seq}}_{\text{templ}}(j) - w_{\text{2nd}}s_{i,j}(k) \tag{2.8}$$

where $\mathbf{F}^{\text{seq}}_{\text{query}}(i)$ is the 20-length sequence-based frequency profile of the ith residue of the query sequence, $\mathbf{M}^{\text{seq}}_{\text{templ}}(j)$ is the 20-length log-odd profile of the jth residue of the template fragment, and $s(i,j)$ is unity when the secondary structure tags are equal at residue i of the query sequence and residue j residue of the template fragment, and -1 otherwise. Both weights w_{seq} and w_{2nd} are set to unity. Note that when the sequence homology is low, the secondary structure profile dominates the alignment score.

For each query SSSM window, the ten template SSSMs with highest alignment scores are selected as structure candidates. Note that among these top ten candidates, the SSE content may vary, as the secondary structure information is but one of three terms in the alignment. Moreover, the number of residues per SSE may vary, as Smith-Waterman includes the processes of residue insertion and deletion. Therefore, the residue lengths of the target SSSM window and the chosen template SSSM may differ. However, the number of SSEs per SSSM is always three.

Generating tertiary structures by simulated annealing

Armed with our geometric packing scoring function and the ensembles of SSSM candidates, we apply Monte Carlo simulated annealing to fold the query protein into a compact tertiary structure in vector form. We begin with the vector chain in its fully extended conformation, such that each vector packing angle is 120^o and each dihedral is 180^o. Then the chain conformation is adjusted using SSSM-based fragment assembly. In each simulation step, a SSSM region in the query sequence is randomly chosen, and the conformation of that query SSSM is replaced by the conformation of a template structure candidate selected randomly from the top 10 candidates for that SSSM. The geometric packing score is evaluated at each step and the SSSM substitution is accepted or rejected by the Metropolis criterion. The effective temperature factor T is adjusted linearly from 20 to 0.5 over the course of $5000P$ steps, where P is the number of SSEs in the query sequence.

At the end of each individual run, a C^α trace is reconstructed from the vector chain. First, ideal C^α traces of helices and strands are superimposed onto the target vectors as rigid bodies, and then the C^α atoms of the loop regions are built in a crude but very fast way. The C^α atoms in a loop region are assigned to an isosceles triangle on the plane defined by two vectors, (i) the vector representing the loop connecting the previous and next SSE, and (ii) the resultant of the normalized vectors representing the previous and next SSE. The loop C^α atoms are spaced 3.8Å apart (the pseudo-

bond length) and bridged by 3.8Å to the C^{α} atoms in adjoining SSEs.

2.1.4 Non-redundant structure database

The template library and geometric scoring function described above are constructed from a non-redundant structure database, which is generated by PISCES [150] and consists of 2701 non-homologous proteins with homologies less than 20% and resolutions better than 1.8Å.

2.2 Results

To illustrate the results of VecFold1, several target proteins are chosen as examples and final models for these proteins are selected from 10^4 independently-converged trajectories. The structures of these models together with their corresponding native structures are shown in Figure 2.6.

Figure 2.6(a) is the C-MYB DNA-binding domain (PDB code: 1MSE). A cartoon of the native structure is shown on the far right and consists of two domains according to its Molecular Modeling Database (MMDB) definition [19]. The N-terminus domain is in black, while the C-terminus domain is in white. The leftmost two structure models are generated by VecFold1 using the same color index as for the native domain assignment. Despite having different tertiary topologies, the VecFold1 models share similar domain features. Such models are generated frequently and repeatedly by VecFold1.

The copper chaperone for superoxide dismutase (PDB code:1QUP) and human heart short chain L-3-hydroxyacyl COA dehydrogenase (PDB code: 2HDH) are two examples with relatively larger sizes than 1MSE. Their native structures and VecFold1 models with similar domain arrangements are shown in Figures 2.6(b) and (c). All these examples (1MSE, 1QUP, 2HDH) indicate that VecFold1 is capable of obtaining conformations that are similar to the native protein at the domain level. Such a finding inspired us to apply statistical analysis to VecFold-based folding results to

identify domain boundary regions.

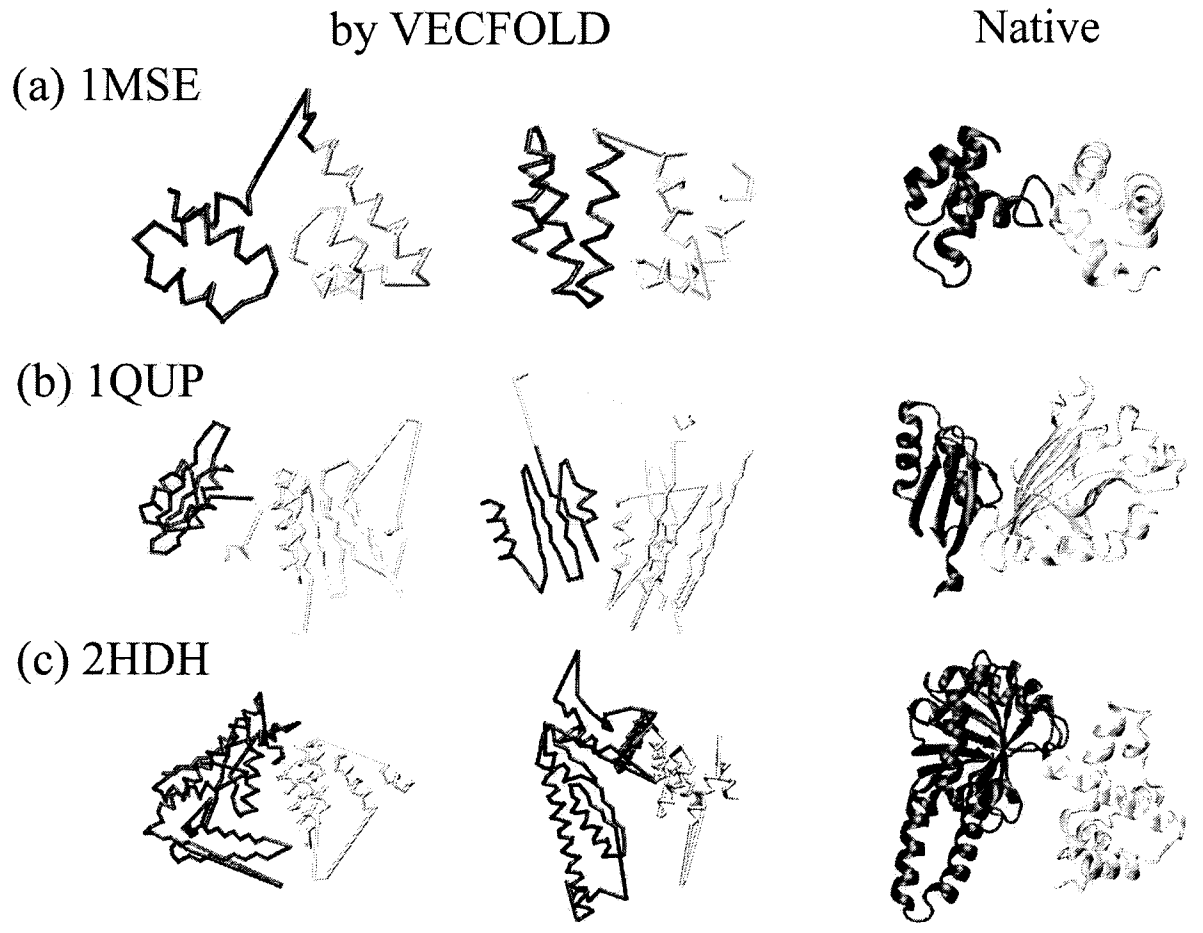

Figure 2.6 : (a) The C-MYB DNA-binding domain (PDB code 1MSE). (b) The copper chaperone for superoxide dismutase (PDB code 1QUP). (c) The human heart short-chain L-3-hydroxyacyl COA dehydrogenase (PDB code 2HDH). A cartoon of the native structure is shown on the far right and consists of two domains specified by the MMDB. The N-terminus domain is shaded black and the C-terminus domain is in white. The two left structure models were generated by VecFold1, and the color index is the same as for the native domain assignment.

Chapter 3

VecFold2

VecFold2 (or VF2) is derived from the original VecFold1 (VF1) [160] but differs in two fundamental ways:

1. VecFold2 operates entirely on C^{α} coordinates; the original operates in SS-vector space. As a result, VecFold2 is able to utilize the more detailed and accurate scoring function OPUS-Ca [161]. In addition, VF2 models the loops much more accurately than VF1.

2. VecFold2 can change the number and sequence positions of secondary structure elements; VF1 is restricted to secondary structure (SS) regions determined by the initial SS prediction. This allows VF2 to sample more broadly, and it also makes VF2 less sensitive to the initial SS prediction.

Figure 3.1 illustrates the benefits of the improved method using the CASP8 target T0428. Figure 3.1(a) is a structure model generated by VecFold1, and (b) by VecFold2, using the same template library. In the VecFold1 model, the loops are very crudely modeled, but more importantly, the broader sampling enabled by varying the number of SSEs allows VecFold2 to sample the correct structure, which in this case has a C^{α} RMSD from native of 1.25Å and a TM-score of 0.96.

The intended audience for VecFold2 is mainly computational biologists who wish to augment their own fragment-based structure predictors with a new sampling scheme and effective scoring function. Thus, we show that VecFold2 is capable of results comparable to the popular Rosetta [131], and that combining the predictions of VecFold2 and Rosetta can yield modestly better results.

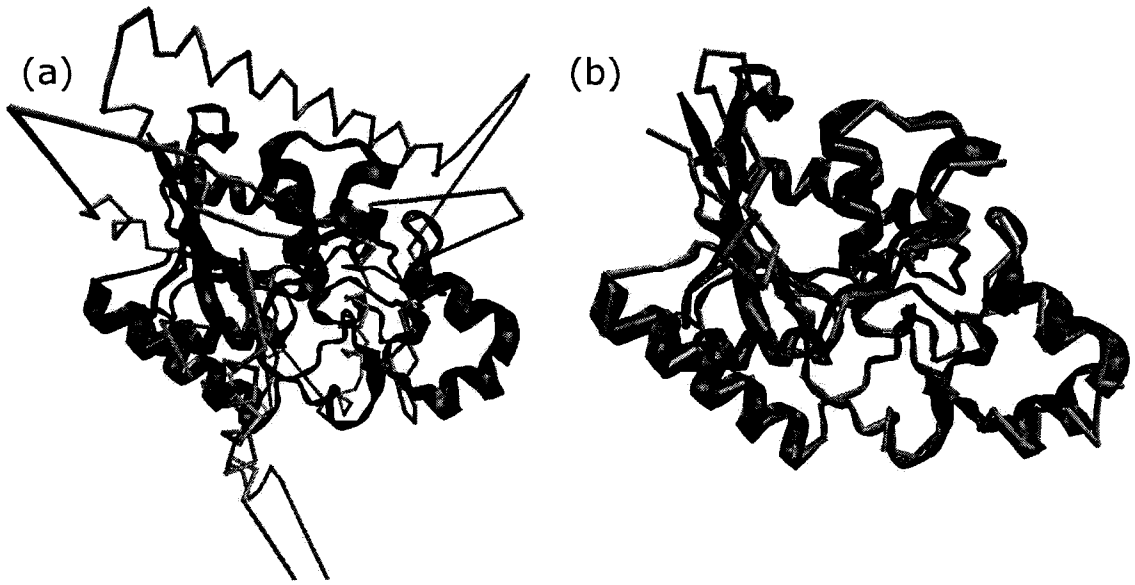

Figure 3.1 : VecFold1 structure model of CASP8 target T0428 (a) compared to structure model generated by VecFold2 (b). The structure in (b) has an RMSD from native of 1.25Å and a TM-score of 0.96.

3.1 Methods

The basis for VecFold2 is the common C-alpha (C^{α}) trace, by which each residue in the protein chain is represented by a pseudo-atom centered at its alpha carbon coordinate. Our method therefore seeks to determine the structure of the protein backbone, which is represented primarily in internal coordinates. Our method is knowledge-based, as we derive our potential energy function and sampling move set from a structure database.

The objective function is based on the OPUS-Ca potential, a unique knowledge-based potential energy function that requires only C^{α} Cartesian coordinates as input, coupled with a simple Lennard-Jones potential term to account for steric contact. The sampling scheme is a variant of fragment assembly, by which the fragments are based on secondary structure boundaries.

3.1.1 Objective function: OPUS-Ca potential

OPUS-Ca is a coarse-grained knowledge-based potential function that is described in detail by Wu *et al.* [161]. OPUS-Ca requires only C^{α} coordinates as inputs, and it consists of seven terms. For speed, we use only the four single or pairwise terms of OPUS-Ca, which are also the most significant, plus a repulsive Lennard-Jones term to disfavor steric clash:

$$\begin{aligned} E = w_{\text{solvent}} \sum_{i=1}^{L} E_{\text{solvent}}(A_i, \zeta_i) + \sum_{i=1}^{L} \sum_{j=1, j\neq i}^{L} \Big[& w_{\text{pairwise}} E_{\text{pairwise}}(A_i, A_j, \Omega_{ij}, r_{ij}^{\beta}) \\ & + w_{\text{Hbond}} E_{\text{Hbond}}(r_{ij}^{\text{C-N}}, \theta_{ij}^{\text{C-O-N}}) \\ & + w_{\text{packing}} E_{\text{packing}}(A_i, A_j, B_{ij}) \\ & + w_{\text{repulsive}} E_{\text{repulsive}}(r_{ij}) \Big] \end{aligned} \tag{3.1}$$

where $A_i, A_j \in \{\text{Glu}, \text{Lys}, \text{Arg}, \ldots, \text{Tyr}\}$ are the residue types for residue indices i and j, respectively. The first term on the right hand side, $E_{\text{solvent}}(A_i, \zeta_i)$, is the solvation energy term based on solvent accessible surface (SAS), where ζ_i is a coarse-grained approximation of the SAS.

The second term, $E_{\text{pairwise}}(A_i, A_j, \Omega_{ij}, r_{ij}^{\beta})$, is an orientation-and-distance-dependent pairwise potential, where $\Omega_{ij} = \text{sgn}\left((\mathbf{r}_i^{\alpha} - \mathbf{r}_i^{\beta}) \cdot (\mathbf{r}_i^{\beta} - \mathbf{r}_j^{\beta})\right)$ is a relative orientation term for C^{β} position $\mathbf{r}_i^{\alpha}$ of residue i and C^{β} positions $\mathbf{r}_i^{\beta}, \mathbf{r}_j^{\beta}$ of residues i and j (sgn is the signum function), and $r_{ij}^{\beta} = \left\|\mathbf{r}_i^{\beta} - \mathbf{r}_j^{\beta}\right\|_2$ is the C^{β}-C^{β} Euclidian distance between residues i and j.

The third term, $E_{\text{Hbond}}(r_{ij}^{\text{C-N}}, \theta_{ij}^{\text{C-O-N}})$, represents main-chain hydrogen bond energy, where $r_{ij}^{\text{C-N}} = \left\|\mathbf{r}_i^{\text{C}} - \mathbf{r}_j^{\text{N}}\right\|_2$ is the Euclidian distance between main-chain C and N atoms in residues i and j, and $\theta_{ij}^{\text{C-O-N}}$ is the angle between main-chain coordinates $\mathbf{r}_i^{\text{C}}$, $\mathbf{r}_j^{\text{O}}$, and $\mathbf{r}_j^{\text{N}}$.

The fourth term, $E_{\text{packing}}(A_i, A_j, B_{ij})$, is a pair-wise secondary structure packing energy potential, where B_{ij} represents a residue interaction pair in parallel or anti-parallel β-sheets.

The fifth term, $E_{\text{repulsive}}(r_{ij})$, is a simple Lennard-Jones 12-6 potential with a linear region:

$$E_{\text{repulsive}}(r_{ij}) = \begin{cases} E_{\text{LJ}}(r_{\text{linear}}) + (r_{ij} - r_{\text{linear}}) \left.\frac{dE_{\text{LJ}}}{dr}\right|_{r=r_{\text{linear}}}, & r_{ij} \leq r_{\text{linear}} \\ E_{\text{LJ}}(r_{ij}), & r_{\text{linear}} < r_{ij} < r_{\text{cutoff}} \\ 0, & r_{ij} > r_{\text{cutoff}} \end{cases} \tag{3.2}$$

where $r_{ij} = \left\|\mathbf{r}_i^{\alpha} - \mathbf{r}_j^{\alpha}\right\|_2$ is the C^{α} distance between the residues i and j, $r_{\text{linear}} = 0.5$, and $r_{\text{cutoff}} = 3.0$. The full Lennard-Jones term is defined as

$$E_{\text{LJ}}(r) = \epsilon \left[\left(\frac{r_{\text{min}}}{r}\right)^{12} - 2\left(\frac{r_{\text{min}}}{r}\right)^{6} \right],$$

where $r_{\text{min}} = 5.0$ and $\epsilon = 0.01$.

All the statistics are obtained by using a structural non-redundant database of non-homologous soluble proteins. The weights for the individual energy terms were optimized against a training set of 25 proteins.

3.1.2 Sampling method: super-secondary structure motif (SSSM) fragment assembly

VecFold2 samples protein state space by fragment assembly, a Monte Carlo sampling method that uses the fragment library as the move set. At each step, the objective function OPUS-Ca is evaluated.

Non-redundant structure database

The SSSM fragments are extracted from a non-redundant template library consisting of 7417 non-homologous soluble proteins. This template library was constructed in August 2006.

Sequence profiles for alignment

The first step in VecFold2 is to generate sequence profiles for alignment. The template library contains structure information (amino acid name, secondary structure type, Cartesian coordinates for the alpha and beta carbons), a frequency profile, and a log-odd position-specific substitution matrix (PSSM) profile. The secondary structure information for the template is determined from the author annotations in the RCSB Protein Data Bank (PDB) [10] records. The query sequence profile also contains frequency and log-odd profiles, as well as secondary structure tags identified by PSI-PRED v2.6 [63].

PSI-PRED used BLAST v.2.2.21 [2] and the BLAST *nr* structure database downloaded from the NCBI* on September 6, 2009. The default PSI-PRED parameters were used for secondary structure prediction. BLAST was run for a maximum of 3 iterations with an e-value threshold of 0.001, and the output of PSI-PRED was smoothed once with no bias on any of the secondary structure predictions.

Alignment score

Template-query alignments are ranked by a score based on SP^3 [178] and similar to Equation 2.8:

$$S_2(i,j) = -w_{\text{str}}(\mathbf{M}^{\text{seq}}_{\text{query}}(i))^{\text{T}}\mathbf{F}^{\text{str}}_{\text{templ}}(j) - w_{\text{seq}}(\mathbf{F}^{\text{seq}}_{\text{query}}(i))^{\text{T}}\mathbf{M}^{\text{seq}}_{\text{templ}}(j) - w_{\text{2nd}}s_{i,j}(k) \tag{3.3}$$

where $\mathbf{M}^{\text{seq}}_{\text{query}}(i)$ is the 20-length log-odd profile of the ith residue of the query sequence, $\mathbf{F}^{\text{str}}_{\text{templ}}(j)$ is the 20-length structure-based frequency profile of the jth residue of the template fragment, $\mathbf{F}^{\text{seq}}_{\text{query}}(i)$ is the 20-length sequence-based frequency profile of the ith residue of the query sequence, $\mathbf{M}^{\text{seq}}_{\text{templ}}(j)$ is the 20-length log-odd profile of the jth residue of the template fragment, and $s(i,j)$ is unity when the secondary structure tags are equal at residue i of the query sequence and residue j residue of

*ftp://ftp.ncbi.nlm.nih.gov/blast/db/

the template fragment, and -1 otherwise. The weights are $w_{\text{str}} = 0.5$, $w_{\text{seq}} = 0.5$, and $w_{\text{2nd}} = 0.4$.

Aligning fragments to super-secondary structure motif windows

Let M be the number of predicted non-loop SSEs (alpha helix or beta strand) in the query sequence. We always assume that loops exist between each non-loop SSE, even if the loop is zero-length. We then divide the query sequence into windows that contain three non-loop SSEs, and we include any leading and trailing loop segments ("LXL...LXL" format, where "L" is a loop segment and "X" is a non-loop SSE), such that the total number of SSSM windows is $M - 2$ (see Figure 3.2). If the entire query sequence contains less than three non-loop SSEs, then only one SSSM window is allocated.

Let L_m be the chain length of an SSSM-window indexed by $m \in M - 2$. Template fragments are extracted in the format "XL...LX" such that they contain at least three non-loop SSEs, and their structure profiles are aligned with the SSSM-window sequence profile and scored. The alignment is gapless, which corresponds to a convolution sum.

Special case: aligning fragments to 9-residue windows

In one of the assessments, VecFold2 with SSSM fragment assembly was compared to VecFold2 with 9-mer fragment assembly. For this 9-mer case, if L is the chain length of the query sequence, we fully divide the query sequence into all overlapping windows of chain length nine, resulting in $L - 9 + 1$ possible windows. The structure profiles of all 9-residue fragments from the template library are aligned with the query window sequence profiles and scored.

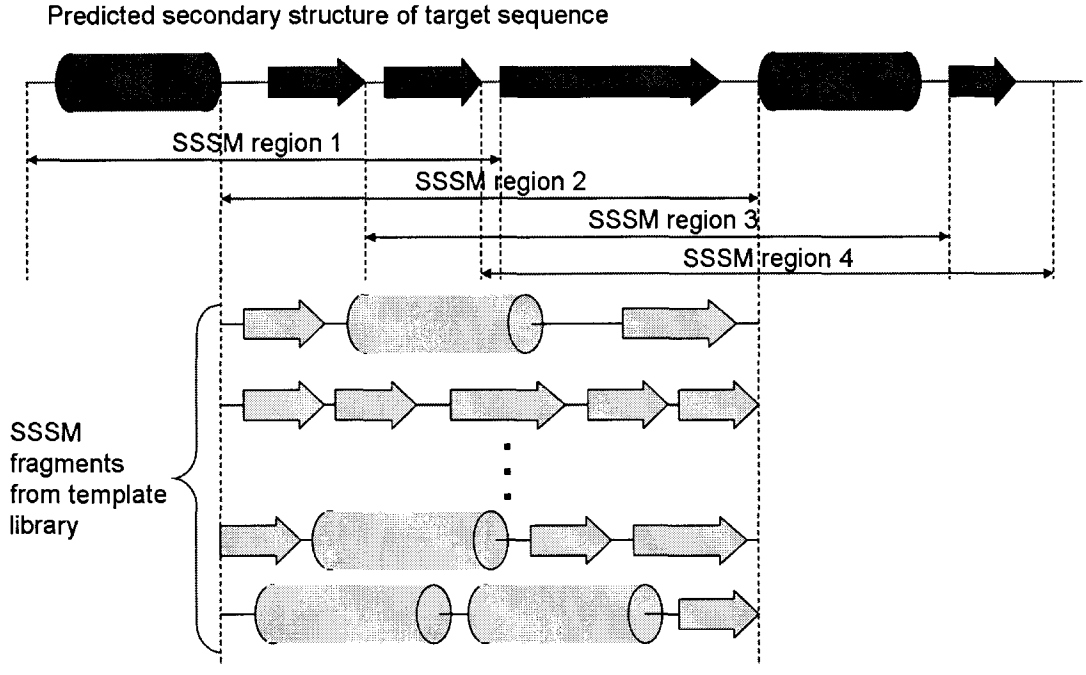

Figure 3.2 : SSSM fragment generation in VecFold2. Note that the number of SSEs per SSSM can vary in VecFold2, whereas they were fixed at three SSEs in VecFold1. As in VecFold1, VecFold2 parses the target sequence into SSSM regions or windows based on a secondary structure profile predicted by PSI-PRED. Then for each SSSM window, fragments that contain only three or more non-loop SSEs and fit within the sequence length of the window are extracted from a template library of non-redundant structures, and each fragment is then aligned to the window and scored by Equation 3.3. The 200 best scoring fragments for each window are stored in a fragment library for use during fragment assembly.

Converting alignment scores to fragment selection probability distribution

For each query window m, the best aligned $K = 200$ candidate fragments are saved, and each template alignment score is converted into a relative probability by a form of the Boltzmann relation

$$P(k,m) = \frac{\exp(-F(k,m)/R)}{\sum_{\ell=1}^{K} \exp(-F(\ell,m)/R)} \tag{3.4}$$

where $F(k,m) = \sum_{j=1}^{L_m} S_{2,k}(j+s_m, j)$, L_m is the chain length and s_m the shift operator for the mth query window, $S_{2,k}(\cdot,\cdot)$ is the residue-pair alignment score function for the

kth candidate fragment as described in Equation 3.3, and R is a scaling parameter set to 0.5. These probabilities are then summed into a cumulative distribution function (CDF), which is used in fragment assembly.

Fragment selection

For each simulation step (objective function evaluations), we randomly choose one of the $M-2$ query windows (M is the number of predicted non-loop SSEs), and then we choose an SSSM fragment at random but weighted by the CDF of that query window. Let m be the index of the query window and k the index of the template fragment for that window, and let $P_{\text{lo}}(k,m)$ and $P_{\text{hi}}(k,m)$ be the lower and upper bounds of the CDF for fragment k in window m. Then in other words, we sample a uniformly distributed pseudo-random variable $r = [0,1]$, and then we search through the K template fragments corresponding to the query window and choose the fragment for which $P_{\text{lo}}(k,m) \leq r < P_{\text{hi}}(k,m)$.

Fragment replacement

A move consists of replacing internal coordinates in our model with those from a fragment selected from the fragment library. Then the objective function (OPUS-Ca potential) is re-evaluated and the move is accepted or rejected by the Metropolis criterion:

$$\text{acc}(\text{old} \rightarrow \text{new}) = \begin{cases} e^{-\Delta E/T}, & \Delta E \geq 0 \\ 1, & \Delta E < 0 \end{cases}$$

where ΔE is the change in energy before and after a move, and T is the Boltzmann temperature. T is set according to a linear annealing schedule

$$T = T_{\text{ini}} - n \cdot (T_{\text{ini}} - T_{\text{final}})/N \tag{3.5}$$

where T_{ini} and T_{final} are the initial and final temperatures, respectively, n is the simulation step index, and N is the total number of simulation steps. In other words,

any move that is downhill in energy is always accepted, and uphill moves are accepted only with a probability that decreases as the energy barrier increases.

Simulated annealing Monte Carlo

The target model is initialized in the extended conformation, in which all bond angles are set to $\theta = 2\pi/3$ and dihedrals to $\phi = 175\pi/180$. The initial and final temperatures are $T_{\text{ini}} = 20$ and $T_{\text{final}} = 0.5$, and the number of simulation steps is 200 times the number of non-loop SSEs, i.e., $N = 200M$.

3.1.3 Tertiary structure prediction using Rosetta 3.1

The Abinitio Relax ("AbRelax") protocol of the open source prediction software suite Rosetta 3.1 [131, 117], downloaded in October 2009, was used as a benchmark for VecFold2. Fragments were generated from the `vall.dat.2006-05-05` template library, and secondary structures were predicted using PSI-PRED v2.6 [63] and BLAST v2.2.23 [2]. Structure profiles were generated from the BLAST *nr* database (downloaded on September 6, 2009) and the VALL Blast database (`vall.blast.2006-05-05`). AbRelax combines the classic Rosetta *ab initio* 9-mer and 3-mer fragment assembly method with a series of refinement steps by the "Relax" protocol. Default parameters and scoring weights were used to generate 1000 candidate structures for each target.

Clustering and selection of candidate structures was achieved using a protocol supplied in Rosetta 2.3.0 (`scripts/abinitio/bin/cluster.pl`), as such scripted protocols were missing from Rosetta 3.1. In short, Rosetta selects a representative structure from each of the top 5 largest clusters as described by Bonneau *et al* [13].

3.1.4 Benchmark test sets used in assessment

For assessing tertiary structure and domain prediction, we use several benchmark test sets. The first three benchmarks sets were used in the authors' previous assessment of domain boundary prediction [160]: GM, consisting of 29 targets; Miyazaki, with

74 targets; and MMDB, with 211 targets. The fourth benchmark is a set of 113 targets from the CASP8 experiment [81]; the 113 were originally chosen by Ezkurdia *et al.* [41].

3.1.5 Assessment methods

We calculate two types of distance measures in our assessment of tertiary structure accuracy: C^{α} root mean square distance (RMSD) and the Template Modeling Score (TM-score) [171]. The latter measure is useful for assessing correct topology because it weights closer residue-pair matches higher than distant matches and is length independent; TM-scores range from 0 to 1, with 1 being assigned to perfectly overlapping structures. RMSD is calculated after pairwise superposition of the candidate C^{α} atoms to the native target structure. The TM-score is calculated using source code obtained from the Zhang Lab[†].

3.2 Results

3.2.1 Performance of VecFold2 versus VecFold1

In Figure 3.3, we compare VecFold2 to the original VecFold1 to show how the new sampling scheme and scoring function have improved the predictive power of the VecFold idea. Figure 3.3 compares the top 5 models by energy score for VecFold2 and VecFold1, using the CASP8 benchmark set; the top 2 panels, left to right, are histograms and scatter plots, respectively, for RMSD, and the bottom two panels are for TM-score. The vertical lines in the histograms on the left represent the arithmetic means of RMSD and TM-score for the selected models, and the diagonal lines in the scatter plots are iso-lines. The bottom right panel in Figure 3.3 shows a dramatic TM-score shift favoring VecFold2, which is not surprising given its more sophisticated sampling method and detailed scoring function.

[†]http://zhanglab.ccmb.med.umich.edu/TM-score/

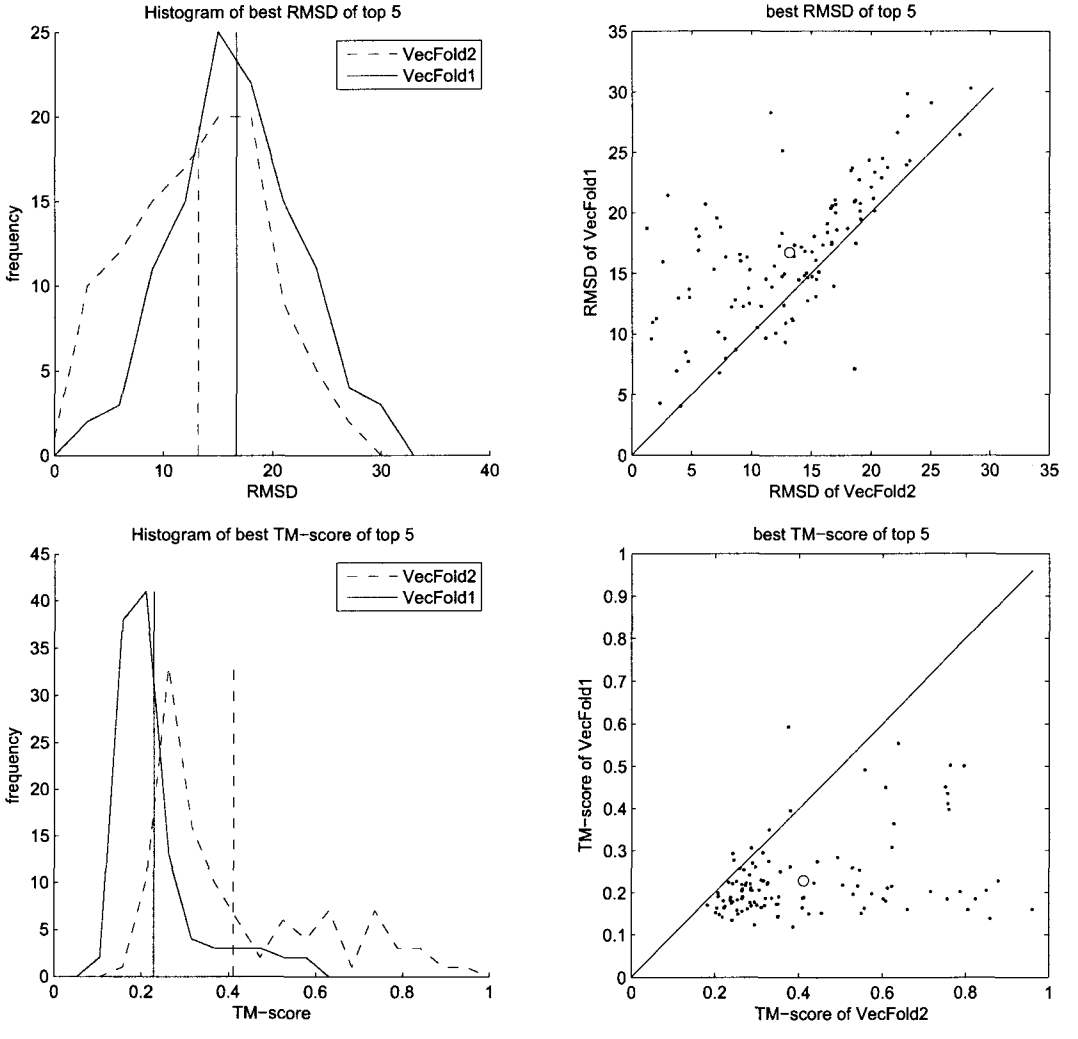

Figure 3.3 : Comparison of best RMSD and TM-score of top 5 energy-score-ranked structure models for each CASP8 target, generated by VecFold2 and VecFold1. The top panels plot C^{α} RMSD-to-native and the bottom panels plot the TM-score, and each of these groups of panels are split into a histogram on the left and the corresponding scatter plot on the right. Each point in the scatter plot represents a target in the CASP8 testset. In this case, VecFold2 shows improvement in mean RMSD (lower value) and mean TM-score (higher value) versus VecFold1. This is also reflected in the scatter plots, as the mass of points are skewed above the iso-line in the RMSD plot and below the iso-line in the TM-score plot.

In addition, we give a more detailed breakdown of prediction performance in Tables 3.1 and 3.2, showing total TM-scores and average RMSDs for two classes of selection criteria and various subsets of CASP8 targets, as well as all the prediction methods and variants included in this study. The selection classification "best of top 5" is standard in protein structure assessments and represents the best model, in terms of RMSD or TM-score, from a set of 5 models that are submitted by the predictor for assessment. The second classification, "best of 1000", represents the very best model out of the 1000 that were generated by each prediction scheme for this study. The "best of 1000" class is included to isolate the protein folding methods from their model selection methods. The "all" columns represent 111 of the original 113 targets of Ezkurdia *et al* [41], as 2 targets (T0471 and T0492) were excluded from the study for technical reasons. The "multi" column represents 35 multi-domain targets, and the "hard" column represents 22 multi-domain targets where at least one domain is considered a template-free modeling target by the CASP8 assessors or where no domain is part of the template-based high-accuracy category. In terms of total TM-score for the best of the top 5, VecFold2 yields approximately 70-80%, 70%, and 40-45% better results than VecFold1 for all, multi-domain, and hard targets, respectively.

3.2.2 Performance of VecFold2 versus Rosetta

In order to evaluate the performance of VecFold2 and determine its strengths and weaknesses, we compare it to Rosetta [131, 117], which is consistently among the most successful tertiary structure predictors at the CASP experiments.

Like Figure 3.3, Figure 3.4 compares the top 5 models for Rosetta and VecFold2, again using the CASP8 benchmark set. Though the shapes of the histograms are similar, VecFold2 has a 2-3Å advantage in terms of average RMSD and is roughly 20% better in total TM-score. In the scatter plots, the points are clustered around the iso-lines but are often off-diagonal. This indicates that VecFold2 predicts certain

	Best of top 5			Best of 1000		
	All*	Multi	Hard	All*	Multi	Hard
VecFold2	45.68	12.84	6.34	54.01	15.33	7.97
VecFold1	25.35	7.61	4.41	31.31	9.14	5.54
VecFold2 (with 9-mer fragments)	36.59	9.68	5.32	44.33	11.72	6.61
VecFold2 (with 2010 library)	59.23	14.91	7.74	66.53	17.83	9.25
VecFold2 (with Rosetta library)	46.62	12.86	6.56	54.25	15.51	8.06
Rosetta	37.53	9.30	5.56	45.84	11.58	6.81
VecFold2 + Rosetta	46.13	13.06	6.49	56.01	15.44	8.14

* Targets T0471 and T0492 were excluded from the study for technical reasons.

Table 3.1 : Total TM-scores for various tertiary structure predictors using 3 CASP8 benchmark subsets. "Best of top 5" represents the best model in terms of TM-score from a set of 5 models submitted for assessment. "Best of 1000" represents the very best model out of the 1000 that were generated by each prediction scheme. The "all" columns represent 111 CASP8 targets, the "multi" columns represent 35 multi-domain targets, and the "hard" columns represent 22 multi-domain targets where at least one domain is considered a template-free modeling target by the CASP8 assessors or where no domain is part of the template-based high-accuracy category [41].

	Best of top 5			Best of 1000		
	All*	Multi	Hard	All*	Multi	Hard
VecFold2	13.19	16.38	17.72	10.15	12.91	14.37
VecFold1	16.69	19.55	20.16	13.51	15.50	15.51
VecFold2 (with 9-mer fragments)	14.73	18.07	18.34	11.92	15.31	15.91
VecFold2 (with 2010 library)	10.96	15.39	16.37	8.09	11.33	13.14
VecFold2 (with Rosetta library)	12.78	16.24	17.02	10.16	12.92	14.32
Rosetta	15.07	19.70	20.32	12.09	16.44	16.66
VecFold2 + Rosetta	13.02	16.01	17.08	9.93	13.10	14.26

* Targets T0471 and T0492 were excluded from the study for technical reasons.

Table 3.2 : Mean RMSDs for various tertiary structure predictors using 3 CASP8 benchmark subsets. Similar to Table 3.1, "Best of top 5" represents the best model in terms of RMSD from a set of 5 models submitted for assessment. "Best of 1000" represents the very best model out of the 1000 that were generated by each prediction scheme. "All", "multi", and "hard" represent 111, 35, and 22 CASP8 targets, respectively [41].

targets better than Rosetta, and vice versa.

In order to understand the differences in performance between VecFold2 and Rosetta, we attempted to isolate several of the key design and implementation features that separate the two methods. The most important of these features are: (a) the template library; (b) the profile alignment and fragment library selection scheme; (c) the sampling method; and (d) the scoring function. In this study, we focused on testing (a) and (c), leaving the profile alignment and scoring function tests for future studies, as these would require, for example, implementing the Rosetta scoring function in VecFold2 or vice versa.

The choice of template library is very important in fragment-based structure prediction methods, and the newer the library, the more likely it will contain templates with high homology to a particular target. Thus, to keep this study fair to the earlier referenced assessments, VecFold2 by default extracts fragments from a template library generated in 2006. To assess the impact of the template library, we generated another set of VecFold2 models for the CASP8 targets using the Rosetta VALL

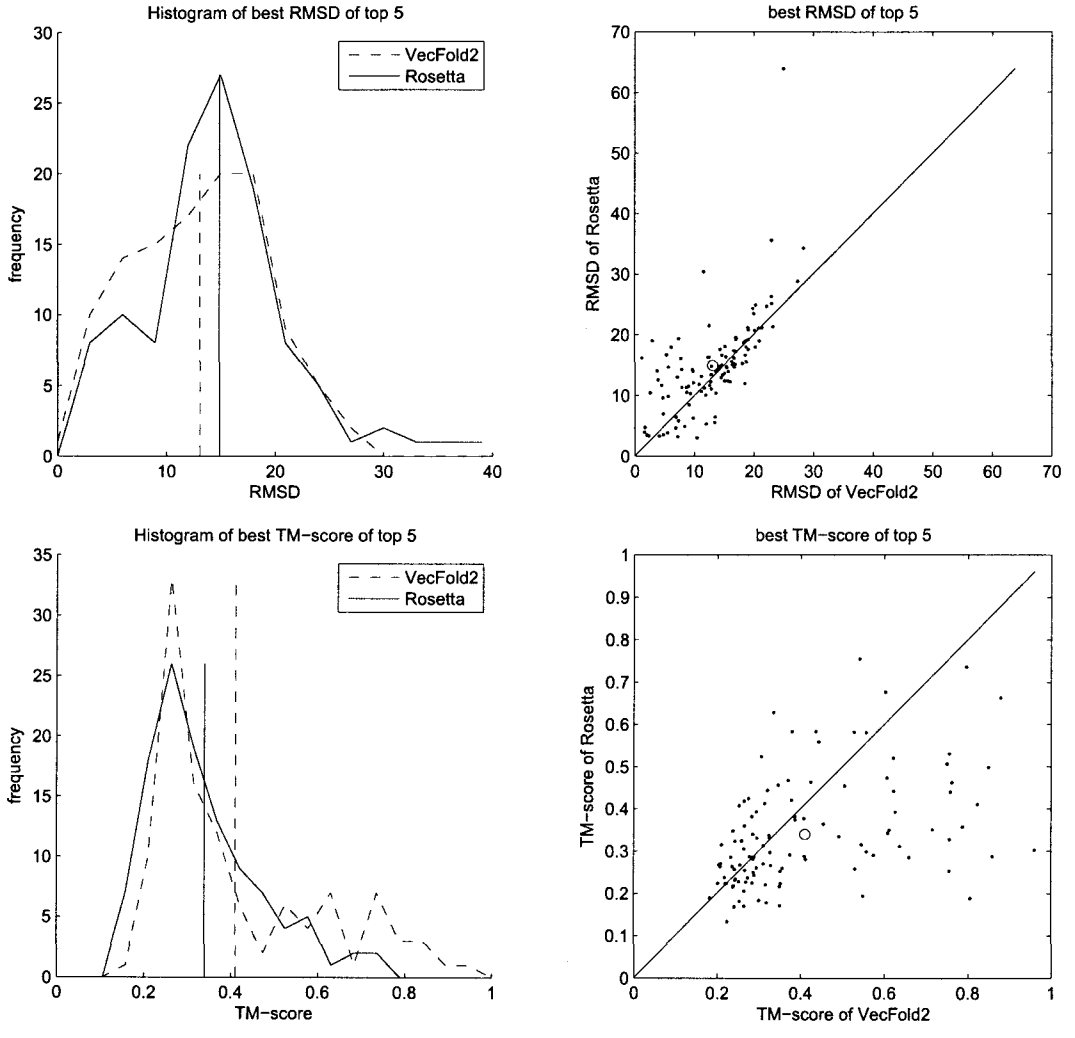

Figure 3.4 : Comparison of best RMSD and TM-score of top 5 structure models for each CASP8 target, generated by VecFold2 and Rosetta. VecFold2 shows a modest improvement in mean RMSD and TM-score versus Rosetta. The dispersion around the iso-lines in the scatter plots suggests that VecFold2 significantly outperforms Rosetta on some targets, and vice versa.

template library (`vall.dat.2006-05-05`). Figure 3.5 shows the same types of plots as Figure 3.3, but with VecFold2 using the default and Rosetta template libraries. Using the Rosetta library improves the VecFold2 results very slightly over the default 2006 template library (see Tables 3.1 and 3.2).

In addition, we generated a new VecFold2 template library based on a precompiled PISCES [150, 151] culled PDB list of 7473 protein chains with percent identity cutoff of 40%, resolution cutoff of 2.0Å, and R-factor cutoff of 0.25 , downloaded June 20, 2010. The comparison of VecFold2 with the 2006 and 2010 libraries is illustrated in Figure 3.6. The improvement in prediction performance is significant, with 15-30% increase in total TM-score for all categories of CASP8 targets.

To test the importance of the sampling method, we compared the VecFold2 structures generated using the SSSM-based sampling scheme with another set of structures generated by VecFold2 but using 9-residue ("9-mer") fragments. This comparison is illustrated in Figure 3.7, and as before, the total TM-scores are included in Table 3.1 and mean RMSD in Table 3.2. SSSM-based fragment assembly yields TM-scores that are roughly 25% better than those by 9-mer fragment assembly. In addition, the summed TM-scores for 9-mer fragment assembly by VecFold2 are roughly equal to those of Rosetta; the difference in mean RMSDs is even smaller.

3.2.3 Performance of VecFold2 and Rosetta combined

One important observation regarding the structure models generated by VecFold2 vis-à-vis those by Rosetta is that there are significant differences in model quality on a target-by-target basis, even though the summary statistics for the whole benchmark set might be comparable. The obvious application of these differences is to combine the models of VecFold2 and Rosetta to yield even better results. Figure 3.8 illustrates how the combination of the first 500 models (out of 1000) by VecFold2 with the first 500 models by Rosetta compare to the full 1000 models by VecFold2. Surprisingly, the changes in the summary statistics are very modest; the total TM-score improves

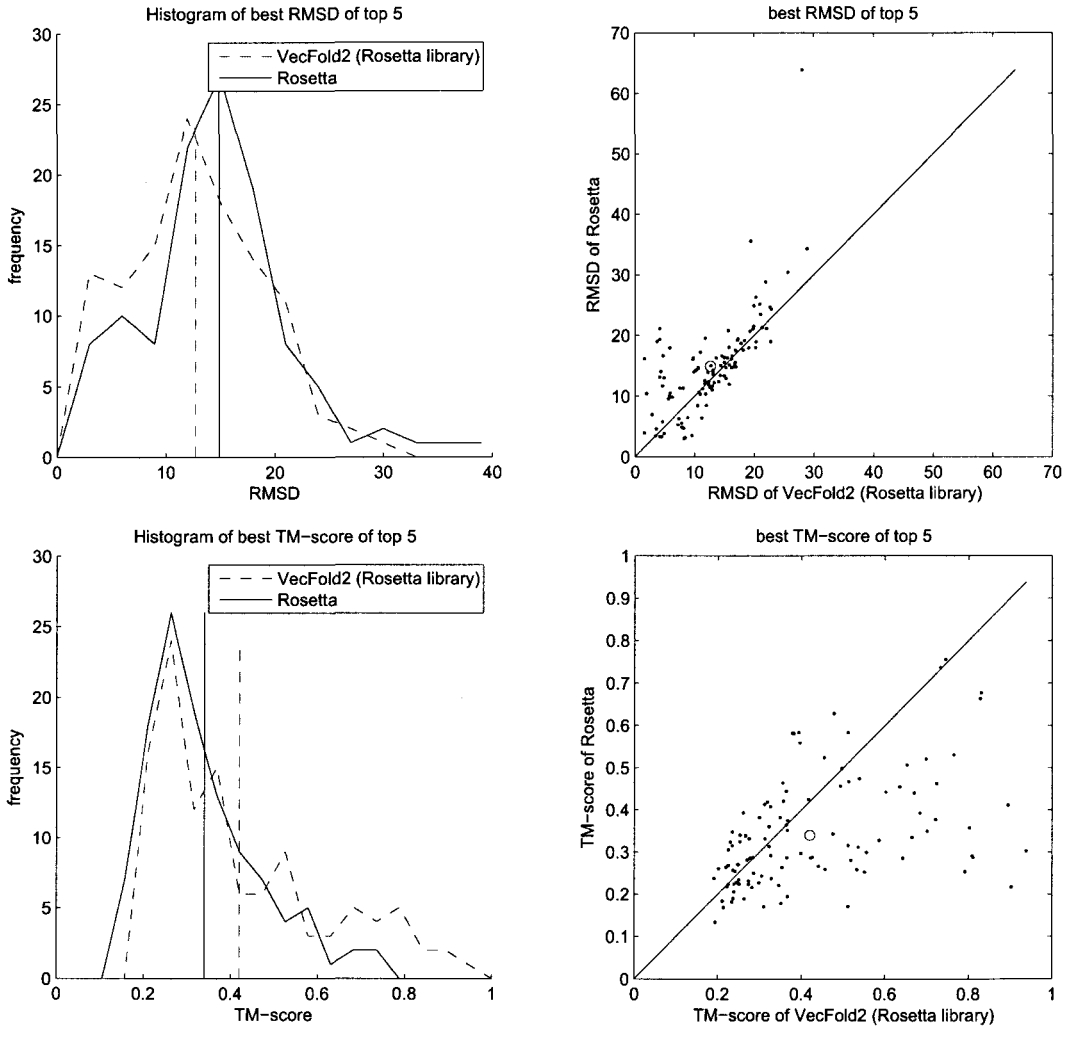

Figure 3.5 : Comparison of best RMSD and TM-score of top 5 structure models for each CASP8 target, generated by VecFold2 and Rosetta, both using the same Rosetta template library. Again, VecFold2 shows a modest improvement in mean RMSD and TM-score versus Rosetta. In contrast to Figure 3.4, the points in the RMSD scatter plot appear to be more tightly clustered around the iso-line.

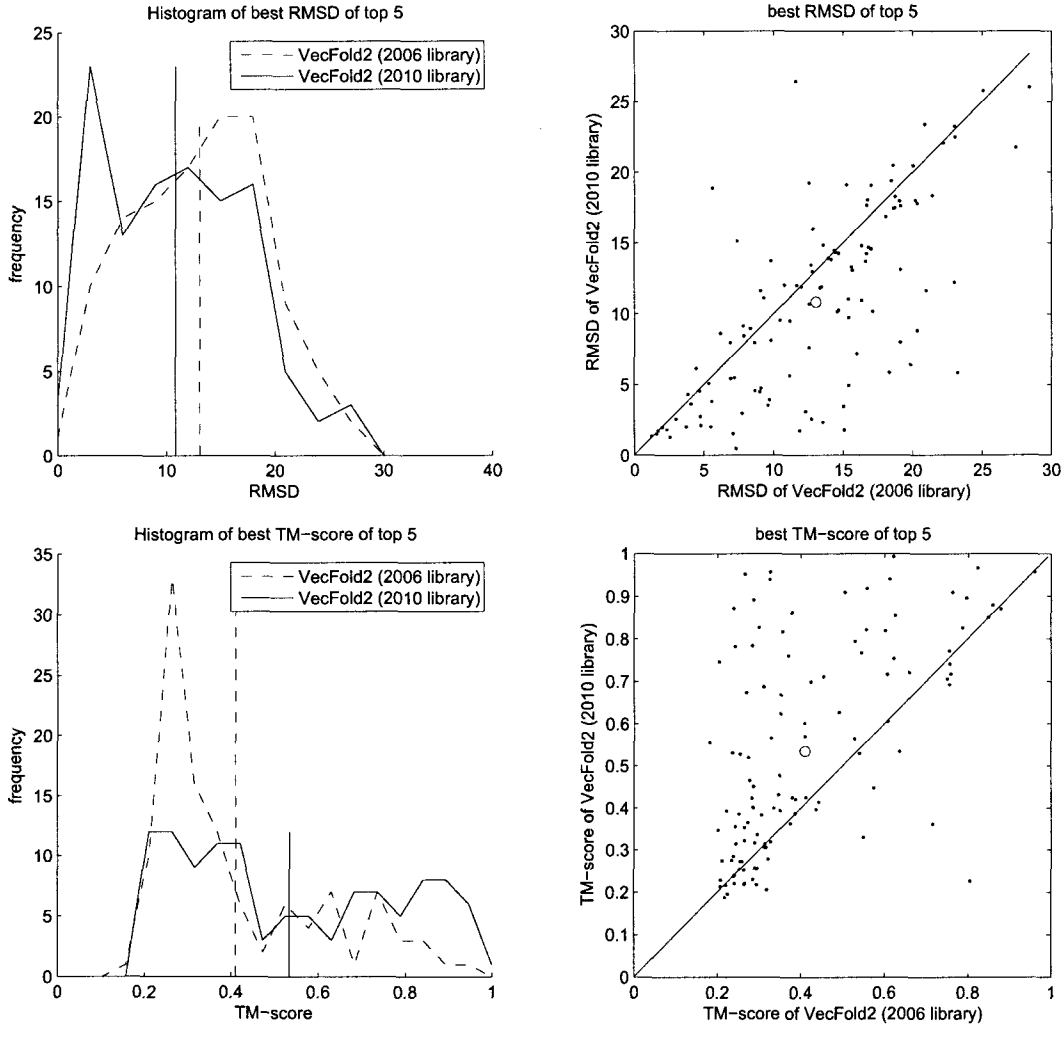

Figure 3.6 : Comparison of best RMSD and TM-score of top 5 structure models for each CASP8 target, generated by VecFold2 using the default 2006 template library versus VecFold2 using the 2010 template library. As expected, VecFold2 with the newer library outperforms VecFold2 with the older library in terms of mean RMSD and TM-score. In addition, the distribution of points in the scatter plots clearly favor the newer library, suggesting that the newer library yields better structure models for nearly every target in the CASP8 set.

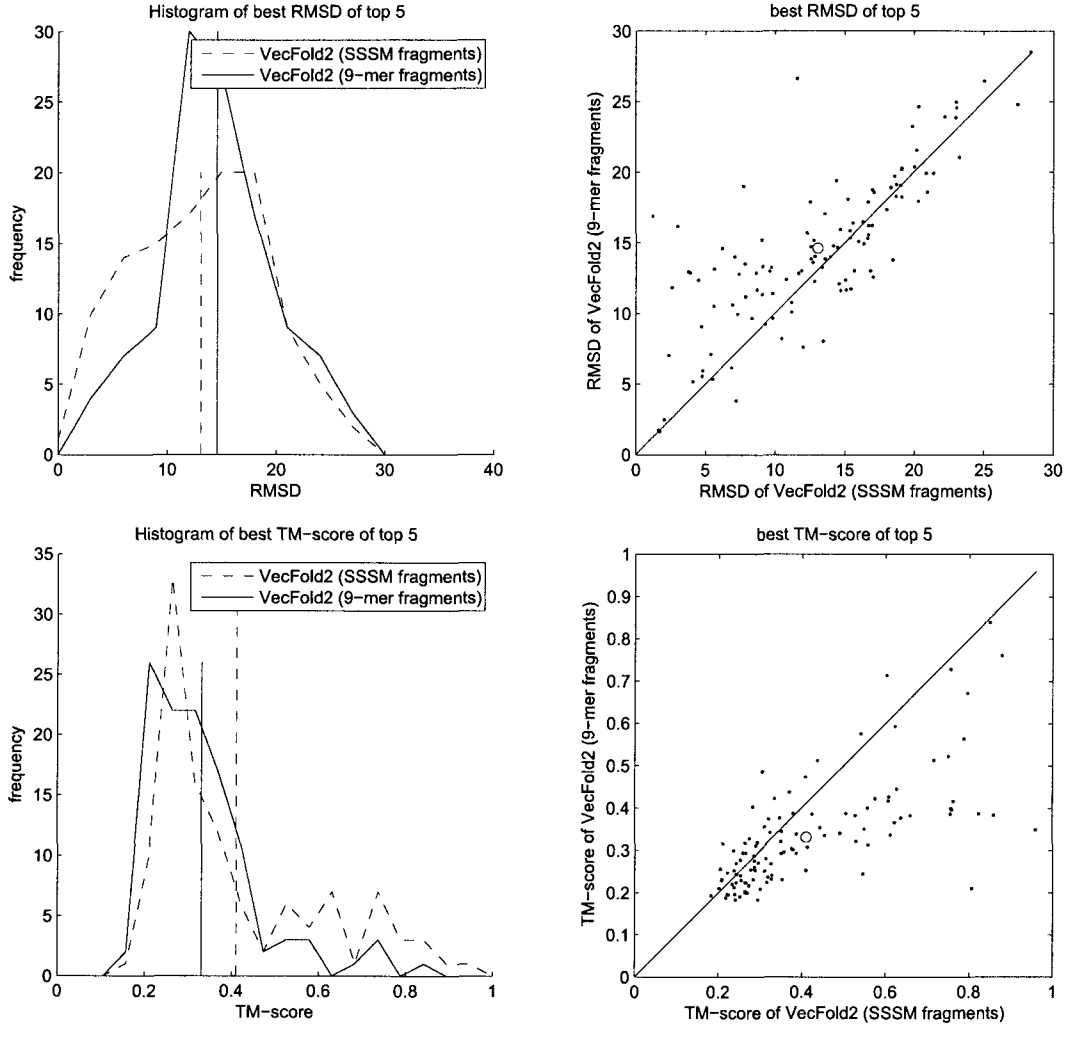

Figure 3.7 : Comparison of best RMSD and TM-score of top 5 structure models for each CASP8 target, generated by VecFold2 using SSSM fragments (default) versus VecFold2 using 9-mer fragments. VecFold2 with SSSM fragments yields better mean RMSD and TM-score than VecFold2 with 9-mer fragments.

by 1-4% and the mean RMSD by 1-3%. Even when including all available (2000) models per target, the TM-score (1-6%) and RMSD improvements (1-8%) are still not large enough to justify the doubling in computational effort.

On a case by case basis, however, the combined approach has relevance. Consider CASP8 target T0512, which is considered a hard comparative modeling target by a group of assessors [127]. As is illustrated in Figure 3.9(a), VecFold2 achieves a best TM-score of 0.642, whereas Rosetta yields a best TM-score of 0.312. Three more challenging examples, fold recognition targets T0460 and T0501 and free modeling target T0496, are shown in Figures 3.9(b)-(d). Rosetta achieves best TM-scores of 0.500, 0.314, and 0.390 versus 0.373, 0.404, and 0.308 with VecFold2 for T0460, T0501, and T0496, respectively.

3.2.4 Sensitivity to secondary structure prediction

One of the design goals for VecFold2 was to make it less sensitive to the initial secondary structure prediction step. In order to test the robustness of VecFold2 to the quality of secondary structure prediction, we compared a standard version of VecFold2 using PSI-PRED and a second version starting with the native SSEs as annotated in the PDB records (note: the PDB entry for 1D3Y contains no SS records for chain B, so the SS definitions for chain A were used for B since A and B are identical in sequence). The scatter plots in Figure 3.10(b) and (d) show that VecFold2 is still sensitive to secondary structure prediction (i.e., there is not a one-to-one correspondence of best models), but the shapes of the TM-score and RMSD histograms are similar and the means nearly identical, suggesting that this sensitivity is small in aggregate.

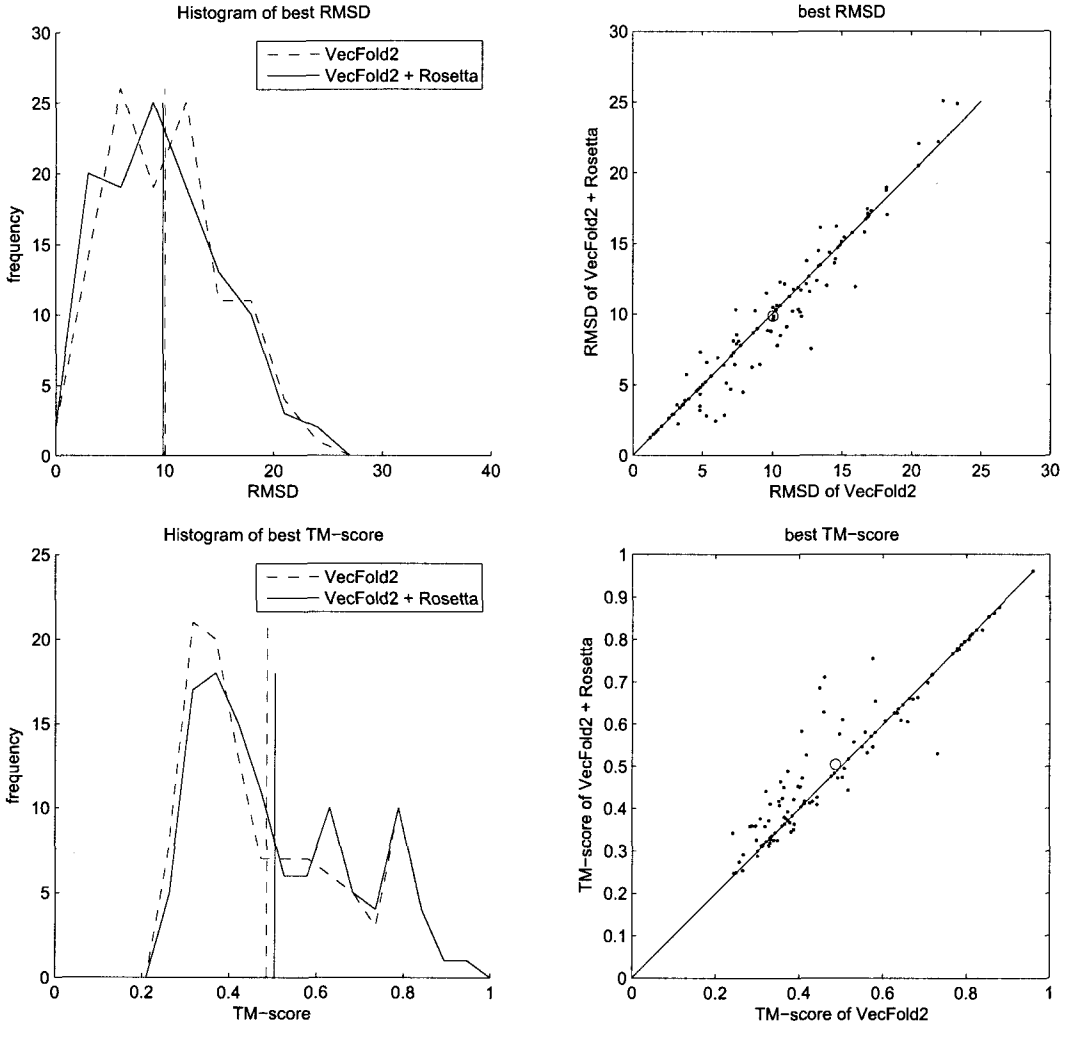

Figure 3.8 : Comparison of best RMSD and TM-score of all 1000 structure models for each CASP8 target, for VecFold2 versus the combination of 500 structure models each from VecFold2 and Rosetta. The scatter plots favor the combined approach by a small amount, such that gains in mean RMSD and TM-score versus VecFold2 alone are very small.

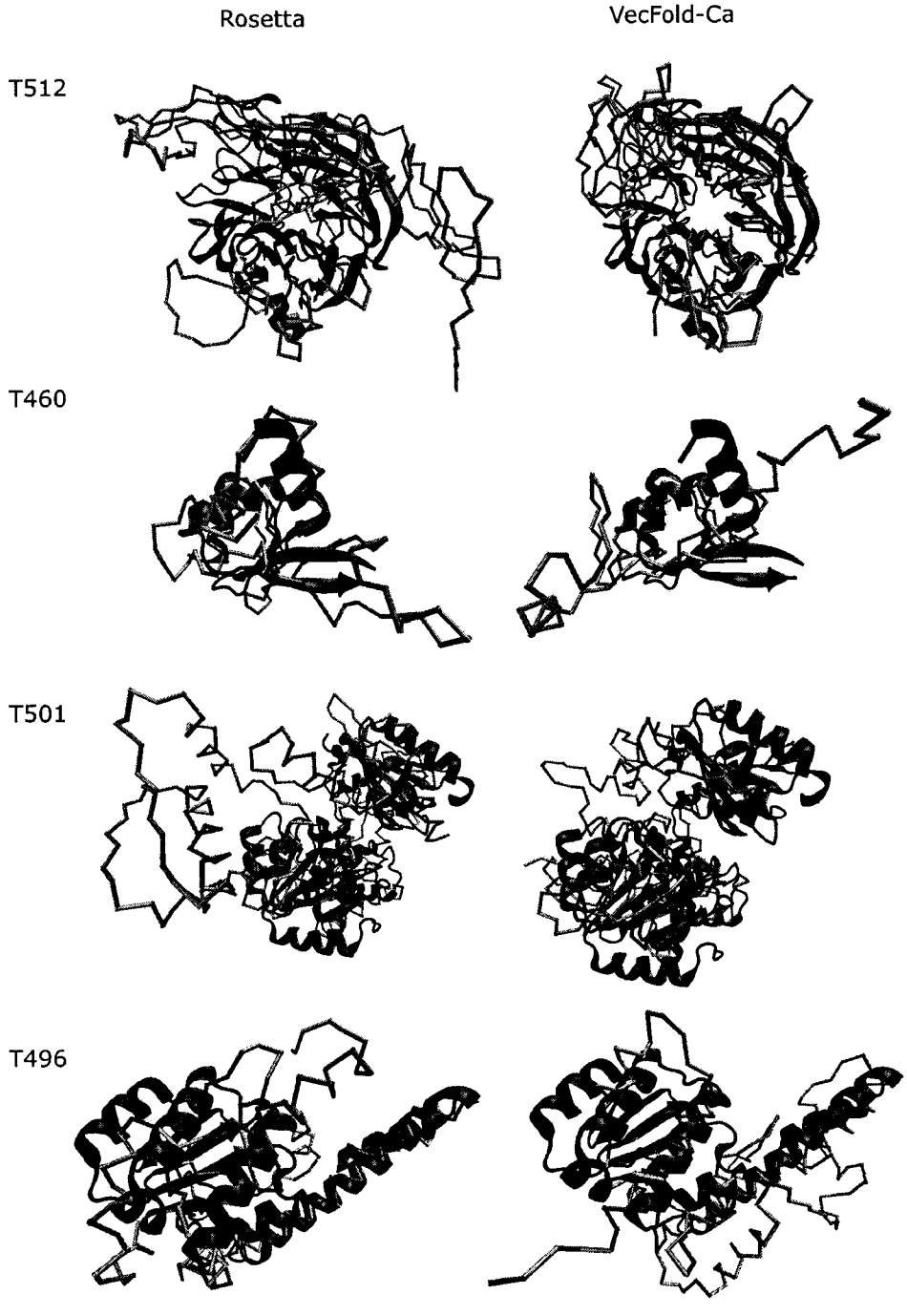

Figure 3.9 : VecFold2 versus Rosetta for four challenging CASP8 targets. In (a), VecFold2 models T0512 with best TM-score of 0.642 versus 0.312 for Rosetta. In (b), VecFold2 models T0460 with best TM-score 0.373 versus 0.500 for Rosetta. With T0501 (c), VecFold2 achieves a best TM-score of 0.404 versus 0.314 for Rosetta. With T0496 (d), VecFold2 and Rosetta achieve best TM-scores of 0.308 and 0.390, respectively.

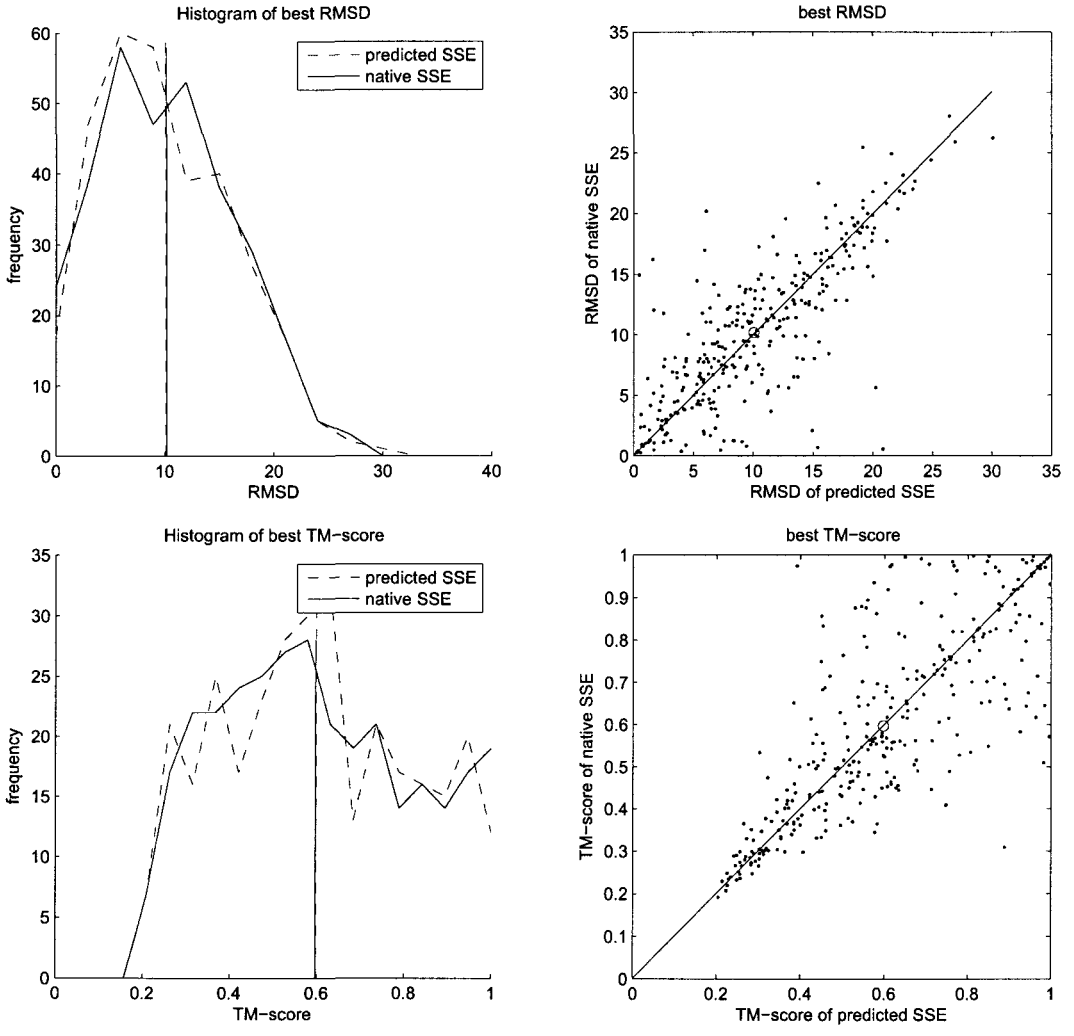

Figure 3.10 : Comparison of best RMSD and TM-score of all 1000 structure models for each target in the combined GM-Miyazaki-MMDB benchmark set (314 targets altogether), for VecFold2 using predicted SSEs versus VecFold2 with knowledge of the native SSEs. Although the scatter plots show some difference in TM-score or RMSD for each target, there is no difference in the mean TM-scores and RMSDs.

Part II

Domain prediction

Protein domains, first proposed by Donald Wetlaufer in 1973 [156], are distinct structural and functional subunits of a larger protein chain. Domains can fold autonomously into self-stable, compact tertiary structures, and thus are of great interest to structural biologists. Identifying protein domains is practically important because it is easier to experimentally solve (by NMR spectroscopy or X-ray crystallography) or computationally fold these subunits rather than whole proteins [115, 48].

Domain prediction is challenging because even human experts can disagree on the domain definitions of well-characterized proteins [55]. Thus, there are a large number of curated domain classification databases such as Class Architecture Topology Homology (CATH) [108], Pfam-A [138], FSSP-Dali Domain Dictionary (DDD) [58, 56], Structural Classification of Proteins (SCOP) [104], Molecular Modeling Database (MMDB) [152, 19], and DIAL-derived Domain Database (DDBASE) [148]. To help them parse domains from three-dimensional structures, human curators often employ computational methods such as the graph-theoretic-based DomainParser [164], DDOMAIN [177], Protein Domain Parser (PDP) [1], and Taylor's domain parsing method [141]. These structure-based predictors are sometimes used in sequence-based domain prediction as well [160, 24, 46, 74].

Like tertiary structure predictors, sequence-based domain prediction methods can be split into template-based and template-free methods. Methods such as CHOP [86] and Ginzu [29] are completely based on sequence homology.

Several methods use statistical methods and information theory, as well as evolutionary information derived from multiple-sequence alignments, to predict domain boundaries and linkers. The Profile Domain Linker propensity Index (PDLI) [36] compares PSI-BLAST profiles from the target sequence to those from known domains. SSEP-Domain [47] predicts domains using secondary structure and profile-profile alignments. Armadillo [37] predicts domain boundaries using an entropy-based residue index [44] and a domain linker propensity index (DLI), which is the frequency that each residue appears in known domain linkers. Several methods predict domain

linkers using hidden Markov models (HMMs) and linker propensity indices [8, 124]. FIEFDom [11] uses a fuzzy mean operator to assign boundaries based on aligned fragments from a reference protein set with known domains.

Machine learning methods include DOMpro [27], DomainDiscovery [130], DLP-SVM ("DOMSERV_H&E" at CASP8) [39], CHOPnet [87], Shandy [149], and KemaDom [21], all of which predict domain linkers using neural networks. DomNet [166] improves on DomainDiscovery with a general regression neural network that predicts domain linkers using secondary structure and solvent accessibility information, as well as an inter-domain linker index derived from DomCut [139].

Hybrid predictors determine domain boundaries using multiple domain predictors, and the most successful methods at CASP often apply different strategies based on the degree of homology detected in the target sequence. In the case of high-homology targets, for example, DOMAC [24] applies Modeller [40] to generate homology-based 3D structures and then PDP to parse each structure into putative domains; otherwise, it invokes DOMpro [27] to predict domains from profile alignments, secondary structure, and solvent accessibility using recursive neural networks. DomPred [16] searches for obvious homologs from which to assign domains by Pfam-A; if no homologous sequences are found, DomPred applies the secondary-structure-alignment scheme DomSSEA [97]. The Baker "DP_Hybrid" server at CASP8 applies either the homology-based iterative strategy Ginzu [29] or the *de novo* method Rosetta-DOM [74] depending on the level of homology of the target to known structures. Meta-predictors, another class of hybrid predictors, yield consensus results from a broad set of methods; for example, Meta-DP [120] accesses results from 10 servers including DomPred, Robetta [73], and DOMpro.

Chapter 4

OPUS-Dom

In this study, we developed a procedure called OPUS-Dom for predicting the domain boundaries of a protein from its amino acid sequence. At the heart of the method is VecFold1, a novel coarse-grained folding method that models a protein structure as a chain of three-dimensional vectors representing the predicted secondary structure elements (SSEs). VecFold1 generates a large number of folded structures, which are then analyzed individually by a protein domain parsing method, and the results are accumulated into a Z-score profile. The domain boundaries are then determined based on the Z-score profile and three empirical filters.

For a systematic evaluation of the predictive power of OPUS-Dom, three multi-domain protein databases were used as the benchmark sets. Additionally, targets from the sixth and seventh Critical Assessment of Techniques for Protein Structure Prediction experiments (CASP6 and CASP7) were also tested. We found that OPUS-Dom generally performs better than all previous methods. For instance, we obtained an overall sensitivity of 55% for CASP6 multi-domain targets, which is substantially better than other published results [74], and 51% for CASP7 targets.

Based on our benchmark results, the consistent and robust behavior of our method suggests that it can locate potential domain boundaries with high confidence. Furthermore, the success of OPUS-Dom provides new insight into the fundamental principles of domain formation.

4.1 Methods

As summarized in Figure 4.1, OPUS-Dom (i) folds 10^4 candidate structures from a query sequence using VecFold1, (ii) generates a structure-based Z-score profile from the distribution of domain boundaries determined for each candidate structure by a domain parsing method, e.g. DOMID, and (iii) predicts domain boundaries using a consensus of three sequence-based domain profiles and the structure-based profile. In this section, we summarize two structure-based domain parsing methods and the process for generating a structure-based domain boundary profile from the distribution of boundaries assigned by either of these methods. The three sequence-based profiles used to improve the domain boundary prediction are conveyed last.

4.1.1 Overall procedure for VecFold-based folding and domain boundary determination

Figure 4.1 is a flowchart of the overall procedure for OPUS-Dom. Given a query sequence, the SSEs are first predicted by PSI-PRED [63]. The SSEs are then grouped into three-SSE windows, each representing a super secondary structure motif (SSSM). The sequence of each target SSSM is then aligned with SSSMs in a structure template library to extract possible structures of the target SSSM. For each SSSM window on the query sequence, a total of ten most-likely structure candidates are saved in vector form after sequence-based alignment.

Guided by a geometric scoring function that describes the packing preference of SSEs, the vector model is folded from an initially extended chain into a compact tertiary structure by simulated annealing Monte Carlo (MC) simulation. In each MC move, a SSSM conformation is replaced by one of the ten saved structure candidates extracted from the template library. At the end of VecFold1, a C^{α} trace is constructed from the vector model in order to carry out further domain boundary analysis.

For each query sequence, 10^4 compact structural models are generated using VecFold1, and then the domain boundaries of each model are analyzed using a structure-

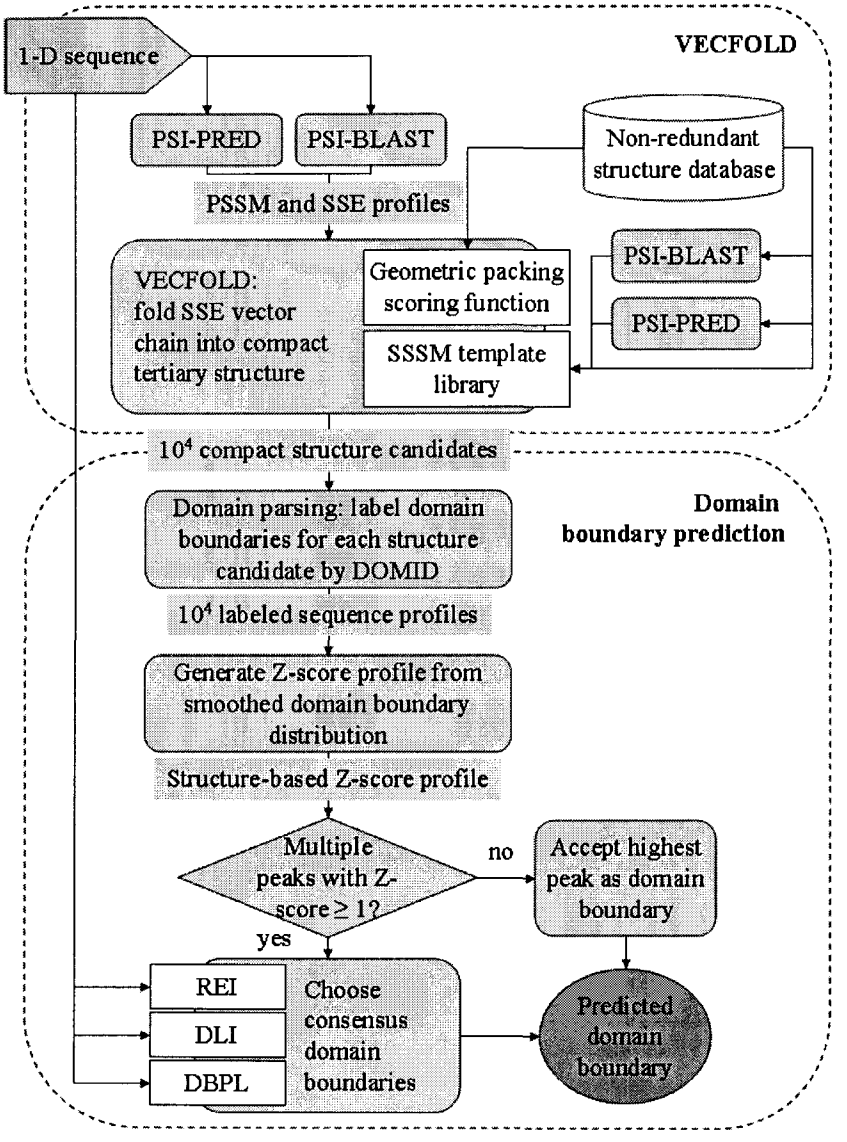

Figure 4.1 : Flowchart of VecFold-based domain boundary prediction method OPUS-Dom. Sequence profiles are generated from the query sequence by PSI-PRED and PSI-BLAST and fed into the VecFold method, which folds the predicted SSEs into a compact tertiary structure by template-based MC guided by a geometric scoring function. Both the scoring function and template library are derived from a non-redundant structure database. DOMID then labels the domains on each of 10^4 candidate structures generated by VecFold. The domain boundaries are counted by residue and a Z-score profile is generated from the smoothed distribution. This structure-based domain boundary profile is then combined with sequence-based profiles generated by REI, DLI, and DBPL from the query sequence. REI, DLI, and DBPL are sequence-based domain boundary predictors that serve to enhance the specificity of the structure-based predictor.[160]

based domain parsing algorithm, e.g., DOMID* or the algorithm of Taylor [141]. Along the sequence, a frequency profile is constructed from the residue-specific domain boundaries identified by the domain parsing algorithm. This profile is then smoothed and converted into a Z-score profile. The residues corresponding to central positions of Z-score peaks above 1.0 are selected as potential candidates for the domain boundaries. Three additional sequence-based profiles are then applied to filter the domain boundary candidates and improve the specificity of OPUS-Dom.

4.1.2 Structure-based domain assignment by DOMID and Taylor's method

For each structure model generated by VecFold1, we assign domain boundaries by the structure-based method DOMID. For this study, we applied, in parallel, the domain parsing method of Taylor [141] to test the sensitivity of our overall VecFold-based domain prediction method to different domain assignment algorithms.

In DOMID, scores for the likelihood of rigid body movement for all the possible domain divisions are derived from the interaction energy of pairs of residues in the target structure. The hypothesis is that the contrast of intra-domain and inter-domain interaction energies is maximized for correct domain partitions, i.e., pairwise residue interactions are strong within the domain and weak between domains. DOMID then assigns domain boundaries to the structural subsets that have considerable tendency for domain movement.

In Taylor's algorithm, each residue in a protein chain is assigned a numerical label (simply, its sequential residue number), and this label changes according to a consensus value of the labels of the residue's physical neighbors. Thus, the labels of nearby residues tend to consolidate, and label boundaries begin to form in regions where the residue density is low. Taylor's method assigns domains to residues that share one of the reduced set of numerical labels.

---

*http://bioinfo1.mbfys.lu.se/Domid/domid.html

4.1.3 Domain boundary Z-score profile

Once the domain boundaries for all 10^4 VecFold1 folded models are obtained by DOMID or Taylor's method, they are accumulated to give a distribution of domain boundaries along the query sequence. This distribution is smoothed using a triangular (biased) sliding window [46] in order to combine nearby peaks. The window slides over the distribution one residue at a time and sums the weighted values within the window, such that the values at more central positions in the window are weighted more than values toward either of the ends. The specific expression for the smoothed score is the convolution sum:

$$S(k) = \sum_{i=1}^{n} \bar{S}\left(k - \frac{n+1}{2} + i\right) \times W_n(n - i + 1) \tag{4.1}$$

where $S(k)$ is the smoothed score at position k, $\bar{S}$ is the raw score before smoothing, and $W_n(i)$ is the window weight function:

$$W_n(i) = \frac{p - |p - i|}{p^2}; \qquad p = \frac{n+1}{2}. \tag{4.2}$$

Here, n is the size of the sliding window, which is set at 5% of the whole sequence length and rounded up to the nearest odd integer. This definition ensures that the sum of the weights is normalized to unity and the weights are biased to the center of the window.

The smoothed domain boundary distribution is then converted into a Z-score profile. The residue positions of all peaks above a Z-score of 1.0 are regarded as potential domain boundary regions. This Z-score threshold is chosen based on empirical data (not shown).

4.1.4 Sequence-based filters

For the potential domain boundaries identified by DOMID or Taylor's method based on VecFold1 models, three sequence-based filters are applied to improve the results: (i) a residue entropy index (REI) filter [37], which is based on the hypothesis that

the domain boundary is dominated by residues with low side-chain entropy and hence small side chain size; (ii) a domain linker index (DLI) filter [44], which is derived from a structure database and is based on the relative frequency of residues found in linker regions compared to compact domains; and (iii) a domain boundary profile library (DBPL) that we developed to extract domain boundary profile information from a non-redundant structure database. All the filtered profiles are in the form of Z-score profiles, and negative values are preferred since the filters are energy-like.

We construct the DBPL by recording the sequence profile information near domain linkage regions of all multi-domain proteins in the non-redundant structure database used to construct the SSSM template library. The profile includes the PSSM and secondary structure information. The DBPL filter score for position i in the query sequence is generated by using an 11-residue window and aligning the sequence profile of the window to all the domain linkage profiles in the DBPL. The filtering profile for the whole query sequence is then generated after recording the lowest alignment score for position i and sliding the window through the whole sequence. The alignment score profile along the sequence is converted into a Z-score profile.

We choose the domain boundary candidates for which at least one filtered Z-score profile falls below -2.0 within ± 5 residues of the candidate boundary position (as determined by the structure-based domain parser, e.g. DOMID). Note that domain boundaries are not assigned to residues that are within 30 residues of the N- or C-termini; furthermore, if two boundaries are less than 30 residues apart, we merge them into a single linkage region. In addition, for cases in which no domain boundary candidate remains after filtering, we consider the target to be a single-domain protein, i.e. having no domain boundaries.

4.1.5 Assessment of domain boundary prediction

For a systematic test of our prediction algorithm, we use the assessment method of Dumontier et al. [37]. First, our predictions are whole linker regions rather than

single-residue domain boundaries in accordance with the literature [37]. As stated earlier in Methods, these linker regions fall within 5 residues of the sequence-based filter predictions and are composed of all residues for which the structure-based Z-score is above the 1.0 cutoff.

We calculate two measures of success, prediction sensitivity and specificity, with respect to the benchmark tests. Prediction sensitivity is the percentage of accurately predicted boundaries out of all official boundaries,

$$\text{Sensitivity} = \frac{TP}{TP + FN} = \frac{\text{correctly predicted domain boundaries}}{\text{total number of official domain boundaries}} \tag{4.3}$$

where TP is the number of true positive boundary predictions and FN is the number of false negatives, i.e. real boundaries that our method failed to predict. To be consistent with the domain prediction literature, we define specificity per Kim *et al.* [74] and Dumontier *et al.* [37] as the percentage of accurate domain boundary predictions out of all predictions,

$$\text{Specificity} = \frac{TP}{TP + FP} = \frac{\text{correctly predicted domain boundaries}}{\text{total number of predictions}} \tag{4.4}$$

where FP is the number of false positives, i.e. predictions that fail to fall near a real boundary. This definition of specificity is sometimes called "accuracy" elsewhere in the literature [144, 41]. If two or more predictions are within the same range of an official boundary, we keep the prediction with the better Z-score in order to prevent biasing the results toward an improved sensitivity and specificity.

In order to assess cases in which no domain boundary exists, we increment the true positive count for each single-domain target for which OPUS-Dom does not predict a domain boundary, and otherwise we increment the false negative count. In other words, we can define $TP' = TP + TP_0$ and $FN' = FN + FN_0$, where TP_0 and FN_0 are the true positive and false negative counts for all single-domain protein targets, respectively, and

$$\text{Sensitivity} = TP'/(TP' + FN')$$

$$\text{Specificity} = TP'/(TP' + FP).$$

4.2 Results

4.2.1 Examples of domain boundary prediction

We follow the criteria of Dumontier *et al.* [37] to assess the accuracy of domain boundary prediction. A predicted domain boundary is regarded as correct when the predicted boundary fully or partially overlaps the true (official) boundary within a margin of tolerance of ± 20 residues. In addition, the 20 residues nearest the N and C termini are excluded from domain boundary prediction. The true boundaries are assigned from the MMDB [152, 19] or the official CASP definitions [145]. In order to generate results comparable to previous methods for CASP6 and CASP7 test proteins [74], the ± 20 residue margin of tolerance is replaced by a ± 10 residue margin for CASP test sets.

Several examples help to illustrate the domain boundary prediction results. The first case is 1QUP, whose VecFold1 results are discussed in a previous chapter and whose native structure is given in Figure 2.6(b) on 27. The 1QUP prediction results are shown in Figure 4.2(a). The black-line-dot profile in the lower part of the figure is the Z-score profile of the statistical distribution of structure-based domain assignments by DOMID on models generated by VecFold1. The solid, dashed and dotted lines in the upper part of the figure represent the three additional filters. The horizontal solid line at Z-score= 1 and the horizontal dash-dotted line at Z-score= -2 are two cutoffs used in this study. Combining the results of DOMID and the additional filters yielded a final predicted domain boundary at residues 73-74 (marked by the solid arrow). The shaded bar below the residue index gives the MMDB domain definition. Comparing the MMDB definition to our results shows that OPUS-Dom successfully predicted the domain boundary for this protein.

The second example is the Ap-2 clathrin adaptor alpha-appendage (PDB code: 1QTS), for which the domain boundary prediction results are shown in Figure 4.2(b). Our method predicted two domain boundaries for this protein. The predicted bound-

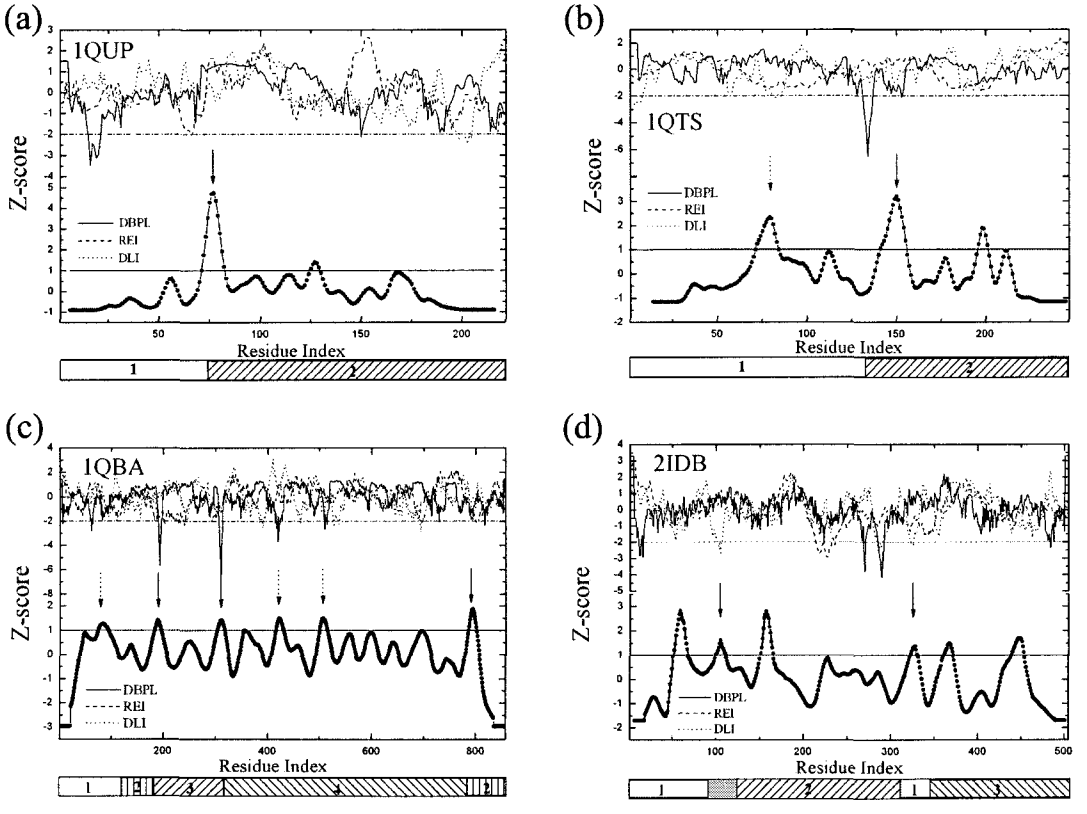

Figure 4.2 : Domain boundary prediction examples for (a) protein 1QUP, (b) protein 1QTS, (c) protein 1QBA, and (d) protein 2IDB. The black line-dot profile in the lower part of the figure is the Z-score of the statistical distribution of structure-based domain assignments by DOMID on models generated by VecFold1. The continuous, dashed, and dotted lines in the upper part of the figure are Z-scores from the three sequence-based filters. The horizontal continuous and dashed-dotted lines are the Z-score cutoffs. The true boundaries are indicated by the shaded bar below each set of Z-score profiles. The continuous arrows indicate correctly predicted boundaries (true positives), whereas the dashed arrows indicate incorrect boundary predictions (false positives).[160]

ary region of residues 141-155 (peak at residue 150, marked by a solid arrow) falls within 20 residues of the true domain boundary spanning residues 130 to 131. The other predicted boundary region at residues 72-84 (peak at residue 79, marked by dashed arrow) is outside the margin of tolerance of the true boundary.

The next example, bacterial chitobiase (PDB code: 1QBA), is a much larger protein with a very complicated domain assignment (Figure 4.2(c)). This protein consists of 858 residues distributed over four domains, with the second domain straddling a long discontinuous region as indicated by the shaded bar. For comparison with previous works, we adopted the domain boundary definition provided in Miyazaki *et al.* [101], where the linker regions span residues 173-222, 335-339 and 780-787. Our method predicted five domain boundary regions, three of which match the predefined domain linker regions. The successful identification of all three predefined linker regions suggests that our method is a sensitive and powerful tool for helping experimentalists determine possible domain regions of large proteins.

The fourth example is the CASP7 prediction target T0356 (PDB code: 2IDB), one of the largest and most difficult targets in the competition. This target consists of 505 residues divided into three domains as illustrated by the shaded bar at the bottom of Figure 4.2(d). The first and third domains of this protein belong to the CASP "free-modeling" category, which is comprised of structures with no homology information. Although the second domain is a template-based modeling target, few groups in the competition found the correct structure analogs. By combining the results from DOMID and the additional filters, our method predicted two boundaries (marked with solid arrows) that are within 20 residues of the official boundaries. The official domain assignment of T0356 is indicated by different shades of gray on the native structure in Figure 4.3(a), and our predicted domains are shaded with the same grayscale index on the native structure in Figure 4.3(b). A comparison of these two figures yields only slight differences near the domain boundaries.

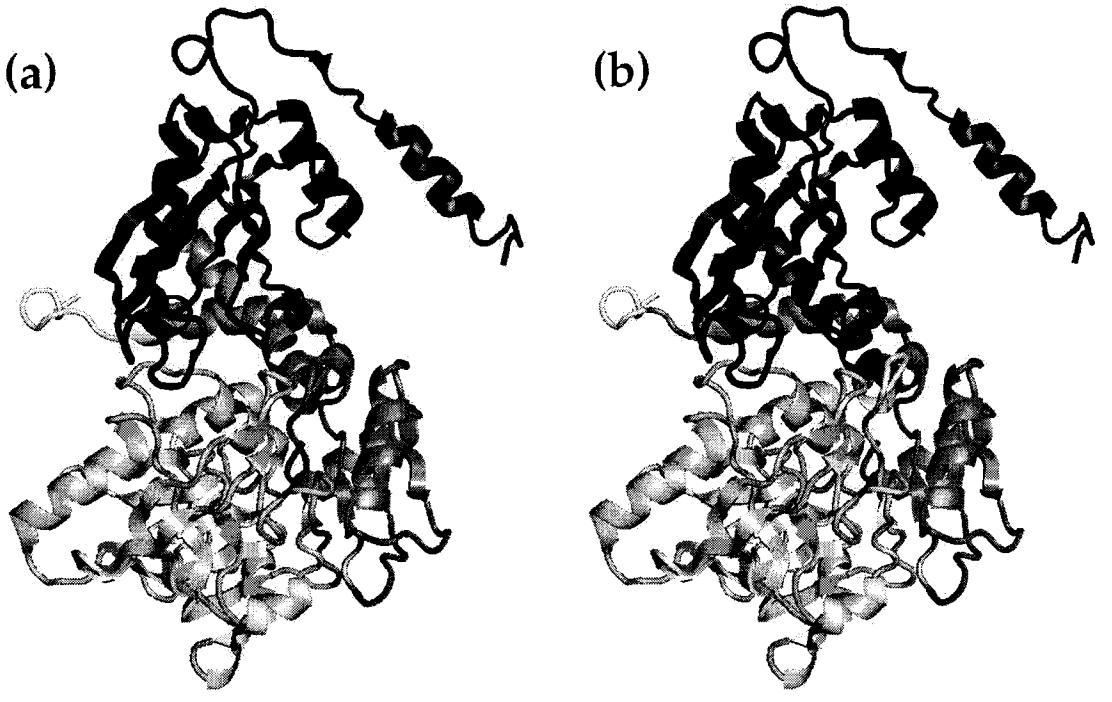

Figure 4.3 : (a) The official domain assignment for protein 2IDB indicated by shades of gray on a cartoon of the native structure and (b) the domains predicted by OPUS-Dom, shaded with the same grayscale index as on the native structure. Note that the domain boundary regions of (a) and (b) differ only slightly.[160] This figure was generated by PyMOL (DeLano Scientific, LLC).

4.2.2 Benchmark evaluations

For the purpose of systematically evaluating the predictive power of our method, three multi-domain protein databases and two sets of CASP domain targets are used as benchmark sets. The first three testing sets were generated for performance evaluation in previous works and could therefore be used for comparative study, and the latter two were created for the CASP6 and CASP7 competitions. We calculated the sensitivity and specificity of our method for all the benchmark testing sets as described in Methods. All specific examples mentioned in the previous sections are drawn from these benchmark testing sets.

For the first three datasets, we compared OPUS-Dom to five other methods: (i) the domain linker propensity index (DLI) of Dumontier *et al.* [37]; (ii) the residue entropy index (REI) method by Galzitskaya & Melnik [44]; (iii) the residue index method by George & Heringa [46] derived from amino acid propensity of all linkers (GHL); (iv) the hydropathy index method by Kyte & Doolittle [82] derived from the propensity of residues to be in the hydrophobic core (KDH); and (v) the DLI+REI consensus domain predictor Armadillo (ARM) by Dumontier *et al.* [37]. In addition to measuring the performance of OPUS-Dom for the entire set, we also analyzed sensitivity and specificity versus protein chain length (Figure 4.4).

The first dataset, called MMDB, is comprised of 211 proteins, each with two contiguous domains and one linker and an average length of 283 residues [37]. The native domain assignments for these proteins follow the MMDB definitions. For this benchmark, our method achieved 50% sensitivity and 39% specificity, both of which are mostly better than previously reported results as shown in Table 4.1. The performance of OPUS-Dom is uncorrelated with protein chain length, with linear regression R-squared values of 0.063 and 0.127 for sensitivity and specificity, respectively (Figure 4.4(a)).

The second benchmark set used in our study is the Galzitskaya & Melnik (GM) representative dataset comprising 29 structures from 44 Structural Classification of

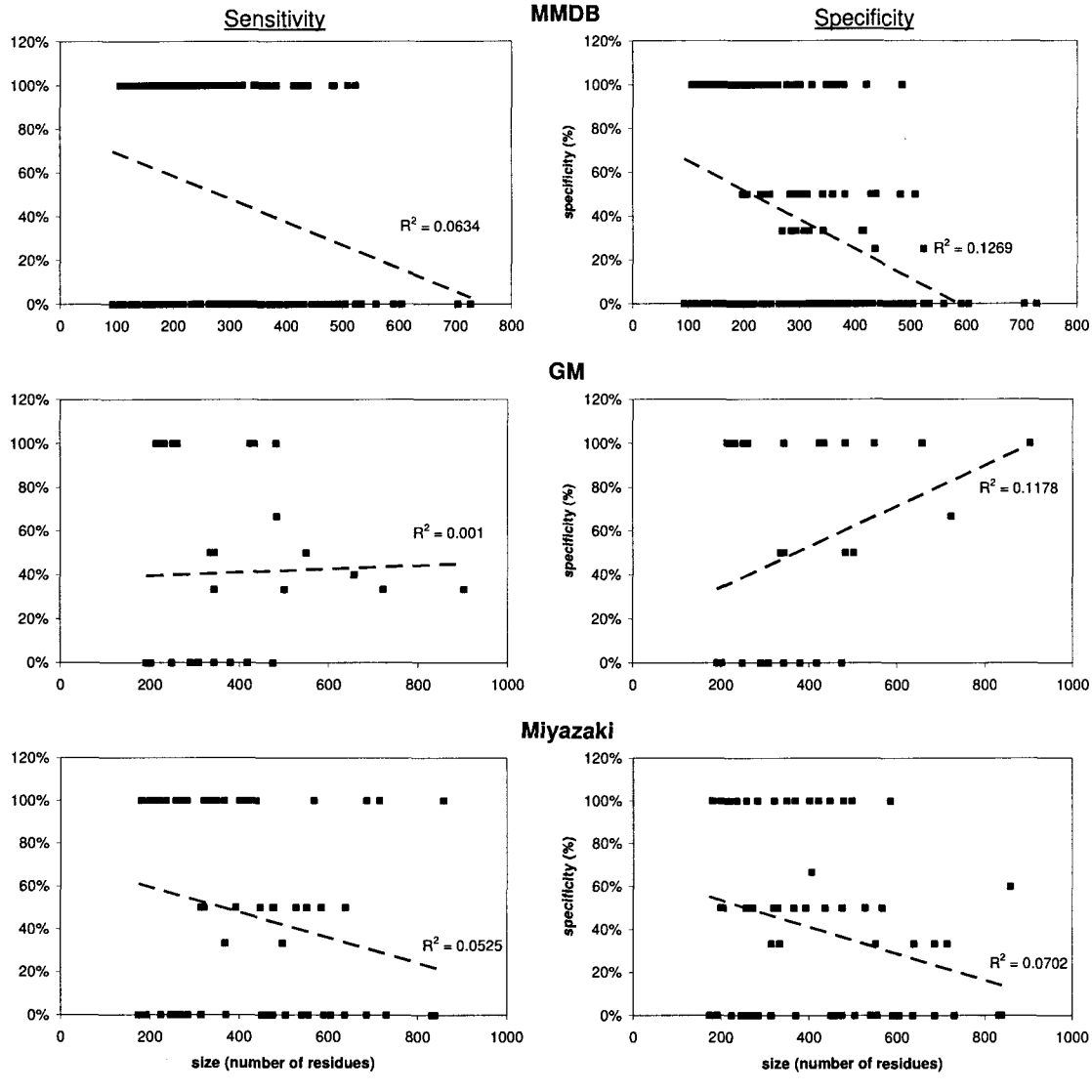

Figure 4.4 : Sensitivity and specificity versus protein size for the (a) MMDB [37], (b) GM [44], and (c) Miyazaki [101] benchmark sets. The correlation coefficients indicate that the sensitivities and specificities are uncorrelated with chain length. Note that the MMDB benchmark set is composed of two-domain proteins with only a single linker, such that the sensitivity is bimodal (0% or 100%).[160]

		DLI	REI	GHL	KDH	ARM	OPUS-Dom
MMDB	Sensitivity (%)	35	26	30	27	56	50
	Specificity (%)	31	23	29	26	32	39
GM	Sensitivity (%)	29	35	29	25	44	42
	Specificity (%)	35	35	42	34	40	59
Miyazaki	Sensitivity (%)	40	41	39	23	45	47
	Specificity (%)	37	38	42	27	46	41

Table 4.1 : Comparisons of OPUS-Dom domain prediction sensitivity and specificity to previous methods (DLI, REI, GHL, KDH, ARM) using three benchmark sets.

Proteins (SCOP) super-families [44]. The GM dataset is one of the most challenging benchmarks because it includes some very large proteins and it also contains multi-domain proteins with multiple domain linkers. Our method achieved 42% sensitivity and 59% specificity. As before, the sensitivity and specificity are uncorrelated with chain length (Figure 4.4(b)). The PDB codes and individual predictions for all 29 proteins are given in Table 4.2.

The third benchmark set was originally presented by Miyazaki *et al.* [101], which consists of 74 protein structures with 99 SCOP-derived domain linkers. Compared with previous methods, our method achieved comparable specificity and sensitivity as shown in Table 4.1. The sensitivity and specificity are uncorrelated with chain length (Figure 4.4(c)).

We also used OPUS-Dom to predict the domain boundaries for CASP6 and CASP7 protein targets. To provide comparable results with previous methods, a 10-residue margin of tolerance was used instead of the typical 20-residue margin. For all multi-domain protein targets in CASP6 whose domain boundaries were officially defined by the CASP assessor [145] (the set will be called CASP6-Multi), we achieved almost twice the sensitivity of one of the most successful domain prediction methods, Rosetta-DOM [74] (the results are shown in Table 4.3).

In addition, OPUS-Dom was applied to a set of 95 CASP7 targets with domain

PDB (Chain ID)	Linker	TP	FN	FP	TP_0	FN_0
1CD9 (B)	1	1	0	0	0	0
1D9X (A)	5	2	3	0	0	0
1DR9 (A)	1	0	1	2	0	0
1DXH (A)	2	1	1	1	0	0
1E43 (A)	3	2	1	0	0	0
1E4E (A)	3	0	3	0	0	0
1EK4 (A)	0	0	0	1	0	1
1EP3 (B)	1	1	0	0	0	0
1F5Q (B)	1	1	0	0	0	0
1FNG (B)	1	1	0	0	0	0
1GE8 (A)	0	0	0	1	0	1
1HBN (A)	2	1	1	0	0	0
1HR6 (A)	0	0	0	3	0	1
1HWX (A)	3	1	2	1	0	0
1HYH (A)	0	0	0	2	0	1
1I4G (A)	1	1	0	0	0	0
1IH7 (A)	6	2	4	0	0	0
1ISA (A)	1	0	1	1	0	0
1JCF (A)	3	1	2	0	0	0
1JR3 (D)	2	1	1	1	0	0
1KMM (A)	2	2	0	0	0	0
1MPY (A)	1	0	1	1	0	0
1PDZ (-)	3	3	0	0	0	0
1PMT (-)	0	0	0	1	0	1
1QH4 (A)	1	0	1	0	0	0
1QM9 (A)	1	0	1	1	0	0
1X01 (A)	1	0	1	1	0	0
2NAP (A)	6	2	4	1	0	0
2PGD (-)	1	1	0	1	0	0
Total	52	24	28	19	0	5

Table 4.2 : Number of actual and correctly predicted linkers for 29 protein structures from the GM dataset (TP: true positive; FN: false negative; FP: false positive; TP_0: true positive for zero domain linkers; FN_0: false negative for zero domain linkers).

		Rosetta-DOM	OPUS-Dom
CASP6	Sensitivity (%)	29	55
	Specificity (%)	55	56

Table 4.3 : Comparisons of OPUS-Dom domain prediction sensitivity and specificity to that of Rosetta-DOM for CASP6 multi-domain benchmark set.

	OPUS-Dom	1-domain	2-domains	3-domains	Multi	All
CASP7	Sensitivity (%)	54	47	40	44	51
	Specificity (%)	52	41	100	47	50

Table 4.4 : OPUS-Dom domain prediction sensitivity and specificity for the CASP7 benchmark set.

definitions, split into single-domain (68 targets), two-domain (25 targets), and three-domain (2 targets) subsets. For single-domain proteins, we counted the predictions as correct when OPUS-Dom did not identify any domain boundaries. As shown in Table 4.4, OPUS-Dom achieved sensitivities of 54%, 47%, and 40% on the single-, double-, and triple-domain sets, respectively. Overall sensitivity and specificity were 51% and 50%, respectively, for the full 95-target set. We also tested OPUS-Dom on a subset of multi-domain proteins that lack any significant sequence homology information. This subset, which we call CASP7-Multi, is a useful benchmark for de novo methods like OPUS-Dom, and it consists of 14 proteins with more than 150 residues and for which PSI-BLAST 35 and the fold recognition method FFAS03 [62] cannot find any proper templates. For these 14 targets, OPUS-Dom achieved 44% sensitivity and 47% specificity.

Finally, the overall domain prediction results from CASP7 are presented in Figure 4.5. OPUS-Dom, entered as "Ma-OPUS" and "Ma-OPUS-DOM", yields the top 7-8 methods among 26 entries. This is an impressive achievement for a first-time entry in the CASP experiments.

In summary, the performance of OPUS-Dom on several different benchmark test-

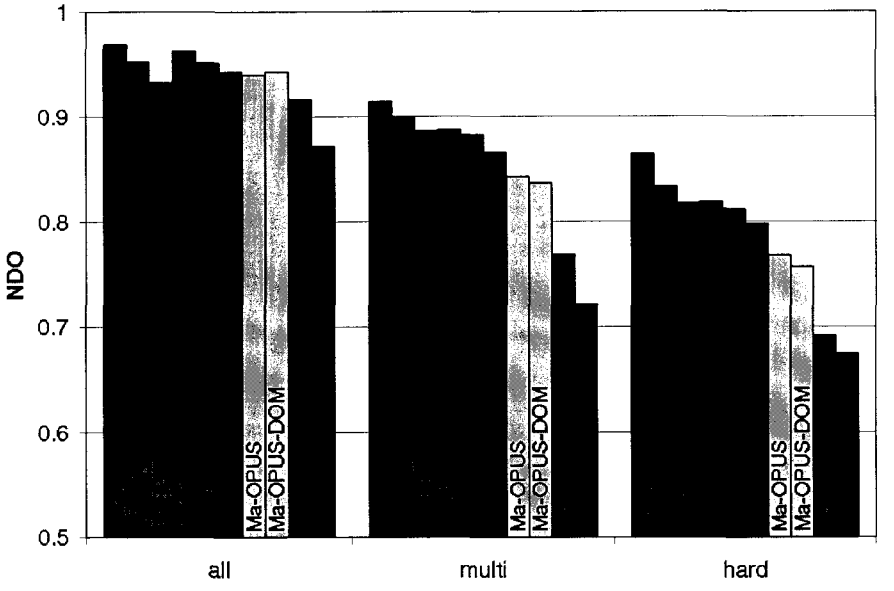

Figure 4.5 : Normalized domain overlap (NDO) for top 10 of 26 CASP7 entrants, split into all (95 targets), multi-domain (31 targets), and hard (24 targets) categories. The two entries of OPUS-Dom, "Ma-OPUS" and "Ma-OPUS-DOM", are highlighted in yellow.

ing sets suggests that the method can locate potential domain boundaries with relatively high confidence.

4.2.3 Other domain prediction methods not included in benchmark

Several other domain prediction methods are discussed below, but due to differences in testing sets or metrics, we did not make a direct performance comparison with OPUS-Dom. SnapDRAGON [46] is a folding-based prediction method like OPUS-Dom and Rosetta-DOM. SnapDRAGON uses DRAGON to generate a large number of atomic structure models, and then it assigns domain boundaries using Taylor's method. SnapDRAGON achieved 42% sensitivity and 40% specificity on a custom multi-domain testing set. DomNet [166] is a very recent machine-learning algorithm that uses a new compact domain profile and outperforms many other machine learning methods on the Benchmark_2 [55] and CASP7 datasets. CHOPnet [87] is a neural-

network-based method that takes into account several parameters such as amino acid composition and flexibility, predicted secondary structure, and solvent accessibility. CHOPnet achieved 51% sensitivity on a custom two-domain testing set.

4.2.4 Domain prediction results: Taylor's method versus DOMID

There is no standard domain definition for a given protein structure, and different structure-based domain parsing algorithms typically give different results. We used DOMID to assign domain boundaries for the VecFold1-generated structure models in the benchmarks described above. For comparison, we also predicted the domain boundaries using the structure-based domain parsing method developed by Taylor [141]. In most cases, e.g., Figures 4.6(a) and (b), the Z-score profiles from DOMID and Taylor's method had similar shapes and were highly correlated, with only slight differences in peak shift and relative peak height (Figure 4.6(c) is an example of less-correlated results). Furthermore, the specificities and sensitivities on all benchmark testing sets (Table 4.5) are comparable, with Taylor's method performing slightly better on the GM, Miyazaki, and MMDB benchmark sets, and DOMID performing better on the CASP6 and CASP7 sets. The success of the OPUS-Dom method using DOMID or Taylor's algorithm suggests that OPUS-Dom is mostly insensitive to the choice of domain parser, and it underscores the power of VecFold1 to generate structures that are topologically correct. Even so, a future direction for our work is to combine different structure-based domain parsing methods to give better results by consensus.

4.2.5 Using sequence-based filters to enhance structure-based domain prediction

Three sequence-based filters were added to OPUS-Dom to improve the specificity of the structure-based domain prediction algorithm. Table 4.6 shows that the specificity improves by as much as 19 percentage points for the five benchmark test sets (MMDB,

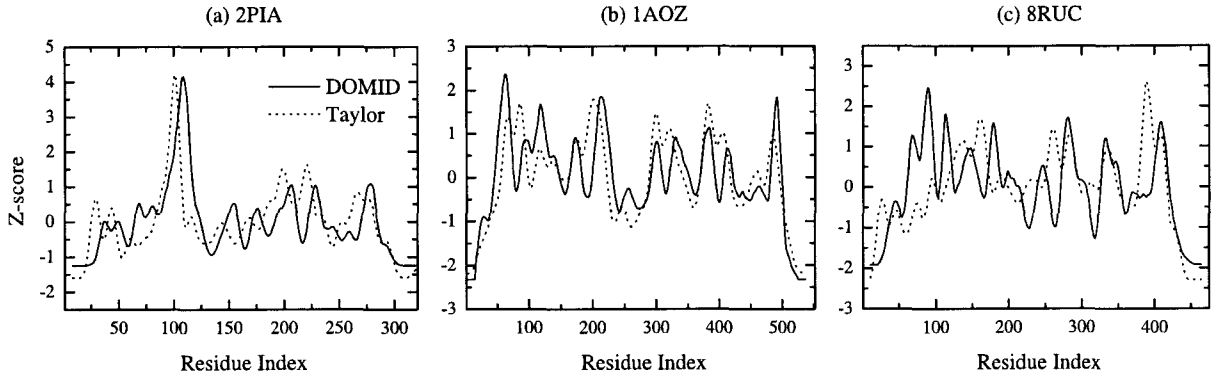

Figure 4.6 : Comparison of the domain boundary predictions from DOMID and Taylor's method. (a and b) Examples of globally similar results with highly correlated Z-score profiles. Here we found slight differences in peak shift and relative peak height. (c) An example of a structure with less correlated Z-score profiles.[160]

		DOMID	Taylor
MMDB	Sensitivity (%)	50	59
	Specificity (%)	39	43
GM	Sensitivity (%)	42	49
	Specificity (%)	59	68
Miyazaki	Sensitivity (%)	47	58
	Specificity (%)	41	45
CASP6-Multi	Sensitivity (%)	55	42
	Specificity (%)	56	46
CASP7-Multi	Sensitivity (%)	44	44
	Specificity (%)	47	47

Table 4.5 : Sensitivity and specificity of domain prediction using DOMID and Taylor's method

		OPUS-Dom	OPUS-Dom without filters
MMDB	Sensitivity (%)	50	74
	Specificity (%)	39	32
GM	Sensitivity (%)	42	60
	Specificity (%)	59	41
Miyazaki	Sensitivity (%)	47	69
	Specificity (%)	41	33
CASP6-Multi	Sensitivity (%)	55	58
	Specificity (%)	56	41
CASP7-Multi	Sensitivity (%)	44	50
	Specificity (%)	47	26

Table 4.6 : Sensitivity and specificity of domain prediction for OPUS-Dom with and without sequence-based domain filters.

GM, Miyazaki, CASP6-Multi, and CASP7-Multi) when the sequence-based filters are applied, which comes at the cost of decreased sensitivity. This specificity-sensitivity trade-off must be considered when fine-tuning OPUS-Dom for different applications, i.e., the structure-based method alone may be sufficient for cases in which sensitivity is most important.

Chapter 5

OPUS-Dom 2

Both the original and latest version of OPUS-Dom are inspired by SnapDRAGON [46], which generates a large ensemble of structure candidates by the *ab initio* method DRAGON[7], and then applies the domain parsing method of Taylor [141] to assign consensus domain boundaries. Rosetta-DOM [74] employs a similar strategy as SnapDRAGON, using Taylor's method to parse domains from structures generated by Rosetta [131]. OPUS-Dom generates structure ensembles using VecFold1 or VecFold2 and then labels domains by Taylor's method as well. OPUS-Dom 2 is summarized in Figure 5.1.

OPUS-Dom 2 improves upon the original OPUS-Dom in three fundamental ways:

- OPUS-Dom 2 uses the more accurate VecFold2 to generate ensembles of candidate structures.
- OPUS-Dom 2 uses Taylor's method instead of DOMID for domain parsing of the candidate structures.
- OPUS-Dom 2 uses a more sensitive scheme for identifying domain boundaries, such that the use of sequence-based filters is no longer necessary.

5.1 Methods

5.1.1 Domain definitions

Domain definition is still a challenging exercise, as expert human assessors may disagree on the boundaries of domains and linkers. Domain definitions are assigned based on the MMDB definitions [19] or those by Ezkurdia *et al.* [41] for CASP8.

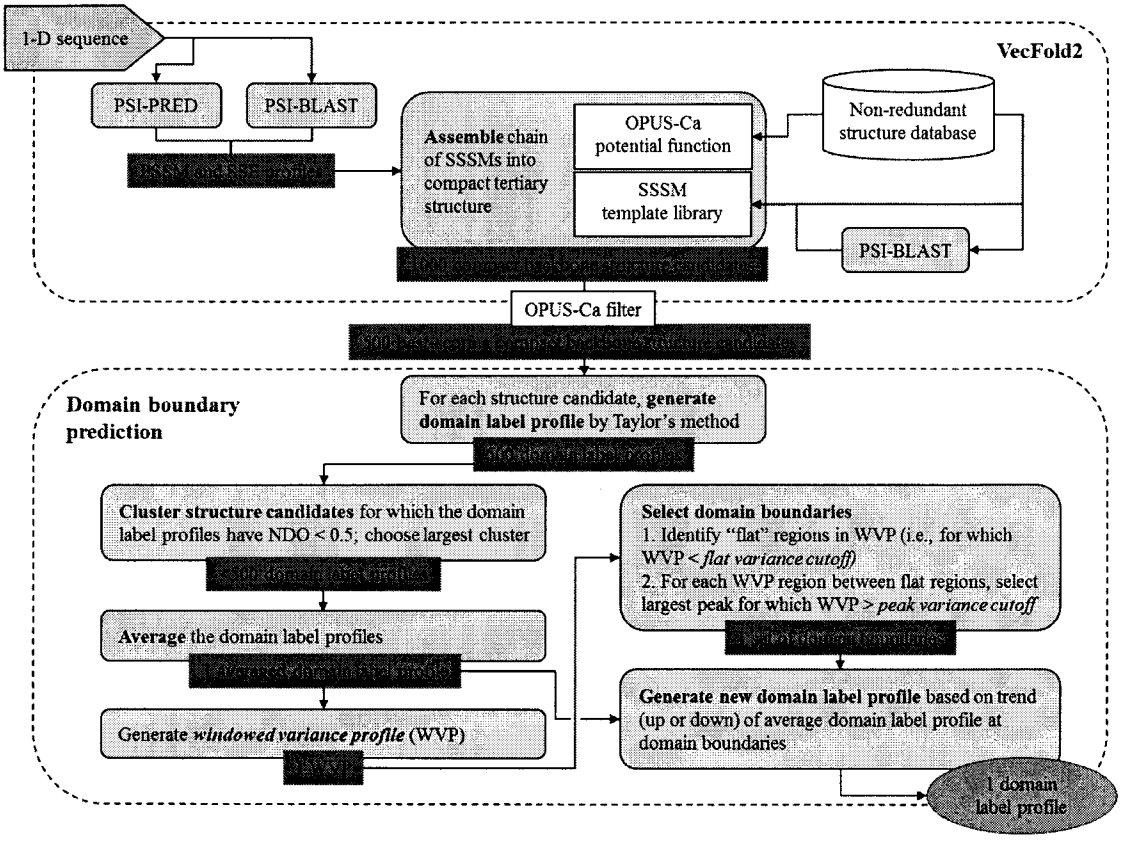

Figure 5.1 : OPUS-Dom 2 flowchart. The target sequence is fed into PSI-BLAST and PSI-PRED to generate PSSM and secondary structure profiles. These profiles are then used to extract SSSM-based fragments from a template library, which also includes PSSM and secondary structure information for each template structure. Next, guided by the potential function OPUS-Ca [161] the SSSM fragments are assembled by simulated annealing Monte Carlo (MC) into compact tertiary structures. 1000 such structures are generated by independent MC trajectories; these structures are ranked by OPUS-Ca energy score and the top 500 are retained for the domain prediction phase of OPUS-Dom 2. Taylor's structure-based domain parsing method [141] then generates a domain label profile for each of the top 500 structures. The normalized domain overlap (NDO) [140] between each pair of these 500 profiles is calculated, and all pairs with NDO ¡ 0.5 are clustered together. A windowed variance profile (WVP) is generated based on the average domain label profile (ADLP) of the largest cluster, and the WVP and ADLP are then used to generate a new consensus domain label profile.

5.1.2 Labeling domains of 3D structures by Taylor's method

We apply Taylor's domain parsing method [141] to identify domain regions from the C^{α} coordinates. Taylor's method is patterned after an Ising model in which the numerical domain label of each residue is influenced by the labels of its neighbors. Furthermore, it is a fast, simple, and fairly accurate domain labeling scheme that requires no putative hydrogen bonding or solvation information.

Taylor's method was used in Rosetta-DOM [74] as well as in the original OPUS-Dom [160]. We used a neighborhood cutoff radius of 17Å, with domain label smoothing and no beta-sheet bias.

5.1.3 Identifying consensus domain boundaries

Clustering domain labels by normalized domain overlap.

Normalized domain overlap (NDO) is a measure for assessing the quality of domain predictions that simultaneously penalizes over- and underprediction. It was first introduced by Tai *et al.* [140] for the CASP6 experiment and is described in Tai *et al.* and Tress *et al.* [144]. It consists of 4 steps:

1. Generate label matrix: Let L be the sequence length of a protein chain and $\mathbf{x} = \{x_1, x_2, \ldots, x_L\}$ be a vector of domain labels, where $x_i \in \{1, \ldots, D\}$ is an integer label for the ith residue, and D is the number of domain labels. Then let $\mathbf{M}$ be a $(D+1) \times L$ label matrix for which the entry m_{di} in each of the first D rows is unity if the row index $d = x_i$ (i.e., the row index equals the domain label at residue i), and zero otherwise. The last row $(D+1)$ is reserved for the linkers, i.e., $m_{(D+1)i} = 1$ if there is no domain label for residue i and zero otherwise.

2. Calculate the score matrix: Let $\mathbf{M}$ and $\hat{\mathbf{M}}$ be the label matrices for the prediction and official target, respectively, and D and $\hat{D}$ be the number of domains in the prediction and official target. Then each entry in the $(D+1) \times (\hat{D}+1)$

score matrix $\mathbf{S}$ is $s_{d\hat{d}} = \sum_{i=1}^{L} m_{di} m\hat{d}i$, i.e. the inner product of the dth and $\hat{d}$th rows of $\mathbf{M}$ and $\hat{\mathbf{M}}$, respectively.

3. Calculate the overlap score: Let $\mathbf{s}_d = \{s_{d1}, s_{d2}, \ldots, s_{d(\hat{D}+1)}\}$ be the dth row of score matrix $\mathbf{S}$, and $\hat{\mathbf{s}}_{\hat{d}} = \{s_{1\hat{d}}, s_{2\hat{d}}, \ldots, s^T_{(D+1)\hat{d}}$ be the $\hat{d}$th column of $\mathbf{S}$. The overlap score V is simply

$$V = \sum_{d=1}^{D} \left(\max(\mathbf{s}_d) - \frac{1}{2} \sum_{\hat{d}=1}^{\hat{D}+1} s_{d\hat{d}} \right) + \sum_{\hat{d}=1}^{\hat{D}} \left(\max(\hat{\mathbf{s}}_{\hat{d}}) - \frac{1}{2} \sum_{d=1}^{D+1} s_{d\hat{d}} \right)$$
$$= \sum_{d=1}^{D} \max(\mathbf{s}_d) + \sum_{\hat{d}=1}^{\hat{D}} \max(\hat{\mathbf{s}}_{\hat{d}}) - \sum_{d=1}^{D} \sum_{\hat{d}=1}^{\hat{D}} s_{d\hat{d}} - \frac{1}{2} \left(\sum_{d=1}^{D} s_{d(\hat{D}+1)} + \sum_{\hat{d}=1}^{\hat{D}} s_{(D+1)\hat{d}} \right)$$

4. Normalize by the perfect score: The perfect score is the sequence length L minus the number of linker residues in the official/target structure. NDO is the ratio of the overlap score and the perfect score.

The domain labels are then clustered by NDO in the following way. The first structure model is assigned to cluster 1. Then starting with the second structure model and iterating through all previous model indices $n \in \{1, \ldots, m-1\}$, the NDO for each pair (m, n) is evaluated. If no pairwise NDO is less than a predefined NDO cutoff, model m is assigned to a new cluster. Otherwise, model m is assigned to the cluster corresponding to the model n for which the NDO is smallest.

For each target, OPUS-Dom 2 first selects the top 500 models ranked by OPUS-Ca score, and then clusters those 500 models with an NDO cutoff of 0.5.

Identifying domain boundaries by windowed variance.

In order to identify changes in domain labels that signify domain boundaries, we apply a simple scheme inspired by methods for piecewise linear approximation of planar curves [110, 20, 38]. These methods often converge on lines that minimize the mean-square error to a targeted contiguous subset of points. Our case is simpler in that we only need to detect changes in numerical labels (i.e., domain 1, domain 2, etc.), as we

assume that the domains will have flat label profiles. Thus, for a window centered at residue i of sequence length $2L+1$, with window average $\bar{x}_{i,L} = \frac{1}{2L+1}\sum_{k=i-L}^{i+L} x_k$,

$$R_i = \frac{1}{2L+1}\sum_{k=i-L}^{i+L}(x_k - \bar{x}_{i,L})^2 \tag{5.1}$$

which is simply a measure of variance over the window length.

To identify domain boundaries, we first average the domain labels at each residue for all candidate structures in the largest NDO cluster (again, the candidate structures are the top 500 models by OPUS-Ca score, and the NDO cutoff is 0.5) to obtain a single domain label profile $\mathbf{x} = \{x_1, x_2, \ldots, x_L\}$, where L is the sequence length of a protein chain. With Equation 5.1, we then calculate a windowed variance profile (WVP) that indicates the transition regions in the averaged domain label profile.

We next look for the peaks in the WVP that exceed a threshold, which we call the *peak variance cutoff*. We also assume that a "flat" domain label profile region must exist between domain boundaries, so we add an additional condition that domain boundaries must be separated by regions in the WVP that fall below another threshold, the *flat variance cutoff*. Finally, we assign domain labels based on the trend of the domain label profile at the domain boundaries, i.e., if the average domain label increases between domain region indexed by G and region $G+1$, and domain region has domain label D, then region $G+1$ is assigned label $D+1$; otherwise, $G+1$ is assigned label $D-1$.

5.1.4 Raw CASP8 domain assessment data

Raw assessment data for CASP8 predictors was downloaded from the CASP data archives* in March 2010 and used in sensitivity and specificity calculations.

*http://predictioncenter.org/download_area/CASP8/

5.1.5 Assessment of domain boundary prediction

For all benchmarks, we calculate two measures of success, the sensitivity and specificity of prediction, defined previously in Equations 4.3 and 4.4 but summarized below:

$$\text{Sensitivity} = TP/(TP + FN)$$

$$\text{Specificity} = TP/(TP + FP)$$

where TP is the number of true positive boundary predictions, FN is the number of false negatives, and FP is the number of false positives.

5.2 Results

5.2.1 Tuning OPUS-Dom for desired sensitivity and specificity

Figure 5.2 illustrates how the domain prediction sensitivity (a) and specificity (b) can be easily tuned by varying two parameters, the variance peak cutoff and the variance window size (see Section 5.1.3). These figures were generated by averaging the sensitivity and specificity for each parameter pair over a range of other parameters and the CASP8 benchmark set. Note that sensitivity generally increases and specificity decreases as the variance peak cutoff is lowered. Increasing the variance window size has the effect of reducing the change in sensitivity or specificity for changes in the variance peak cutoff. For optimal sensitivity, a small variance window and small variance peak cutoff is necessary. For optimally balanced sensitivity and specificity, both the variance window and peak cutoff must be increased.

5.2.2 Domain prediction with GM, Miyazaki, and MMDB benchmarks

First, we compare the new OPUS-Dom 2 (OD2) with the original OPUS-Dom (OD1) and a group of five domain predictors from our preceding domain prediction study, using the GM, Miyazaki, and MMDB benchmark sets. Figure 5.3(a) illustrates the substantial improvements in sensitivity for OD2 versus OD1 and the other predictors

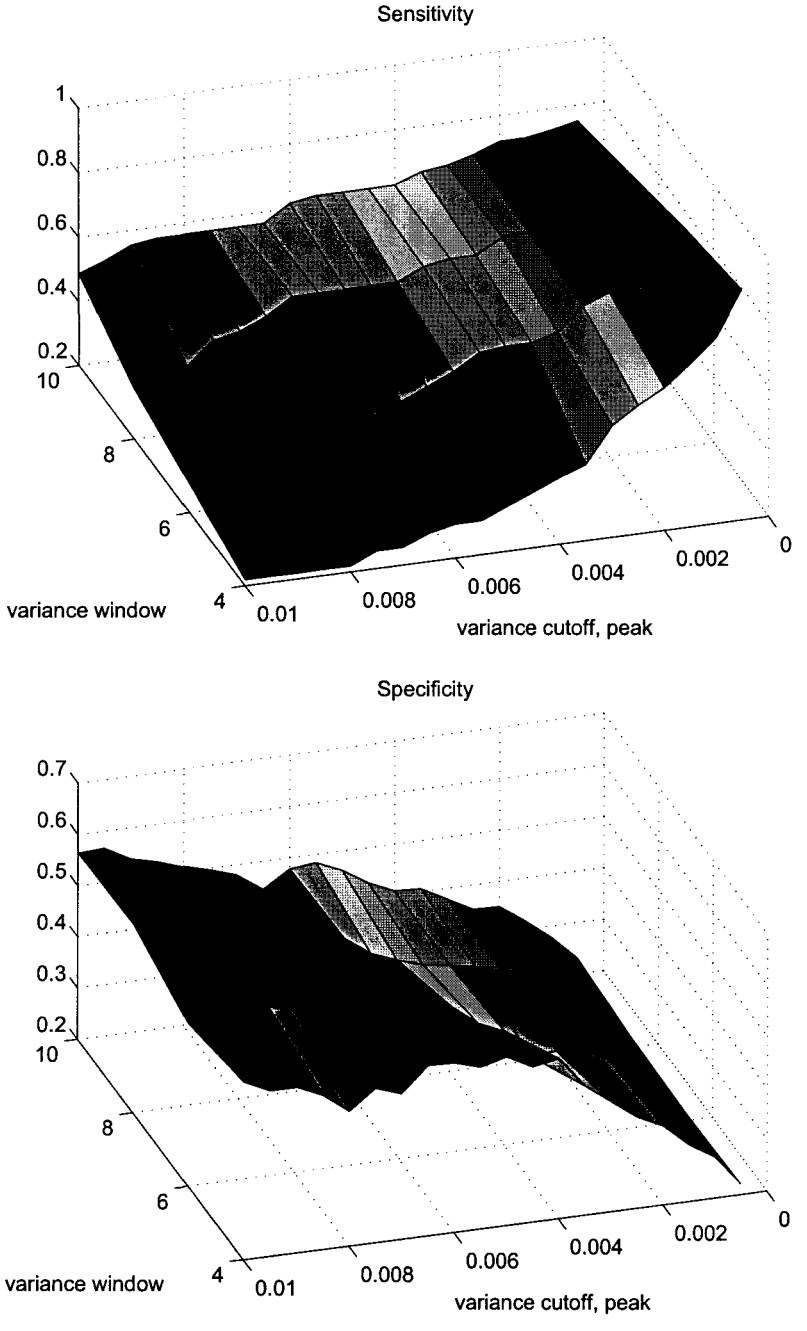

Figure 5.2 : Tuning OPUS-Dom 2 for sensitivity and specificity. The surface plots from top to bottom show how sensitivity and specificity, respectively, can be tuned by varying two parameters, the variance window size the and peak variance cutoff. In general, simultaneously decreasing the variance window size and the peak variance cutoff (lower right on the x-y-plane of each plot) increases sensitivity at the cost of specificity. For more balanced sensitivity and specificity, a larger window size and cutoff are required (upper left on the x-y-plane).

in the benchmarks, while (b) shows that OD2 has somewhat better specificity than the rest. As in the original study, we count a prediction as a true positive if it is within 20 residues of the official domain boundary or linker.

5.2.3 Domain prediction with CASP8 targets versus other automated methods

We also assess OPUS-Dom 2 sensitivity and specificity relative to the other server-only predictors from the CASP8 competition. We count a prediction as a true positive if it is within 10 residues of the official domain boundary or linker. Figure 5.4 shows OPUS-Dom 2 with two choices of sensitivity-specificity parameters, for multi-domain CASP8 targets (35 total) and hard targets (22 total), as designated in Ezkurdia *et al.*[41]. In the yellow, OPUS-Dom 2 exhibits superior sensitivity versus the other methods, achieving nearly 95% sensitivity with some specificity penalty. For optimal sensitivity, we apply a peak variance cutoff of 0.0005, a flat variance cutoff of 0.002, and a window half-length of 5 residues.

In the green, the sensitivity and specificity of OPUS-Dom 2 are more balanced. For this balanced variant, we apply the default parameters: peak variance cutoff of 0.0088, a flat variance cutoff of 0.002, and a window half-length of 10 residues. Figure 5.5 compares OPUS-Dom 2 (with default parameters) with the CASP8 predictors in terms of NDO.

The calculations of sensitivity and specificity depend on the definition of a true positive prediction, which is based on a predicted boundary falling within some range of the official boundary. Figure 5.6 illustrates how the sensitivity and specificity vary with this range or distance from the official domain boundary, for multi-domain CASP8 targets. At zero distance from the official boundary, MUProt [153] is best, but beyond 2 residues from the official boundary, OPUS-Dom 2 has the highest sensitivity.

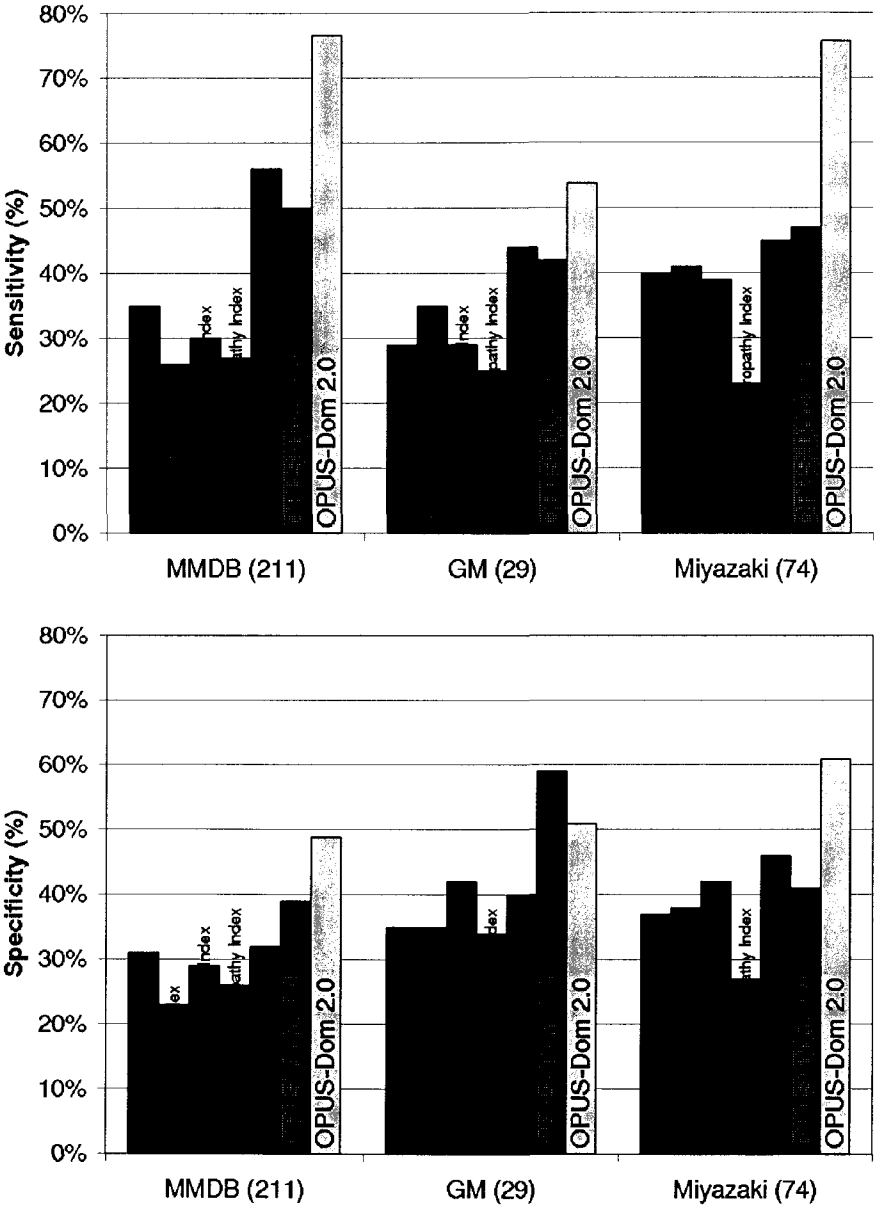

Figure 5.3 : Comparison of sensitivity and specificity for OPUS-Dom 2 (in yellow) and the original OPUS-Dom (in orange) versus several other domain predictors, over three benchmark sets used in our previous study [160]: the 29-target GM set [44], the 74-target Miyazaki set [101], and the 211-target MMDB set [37].

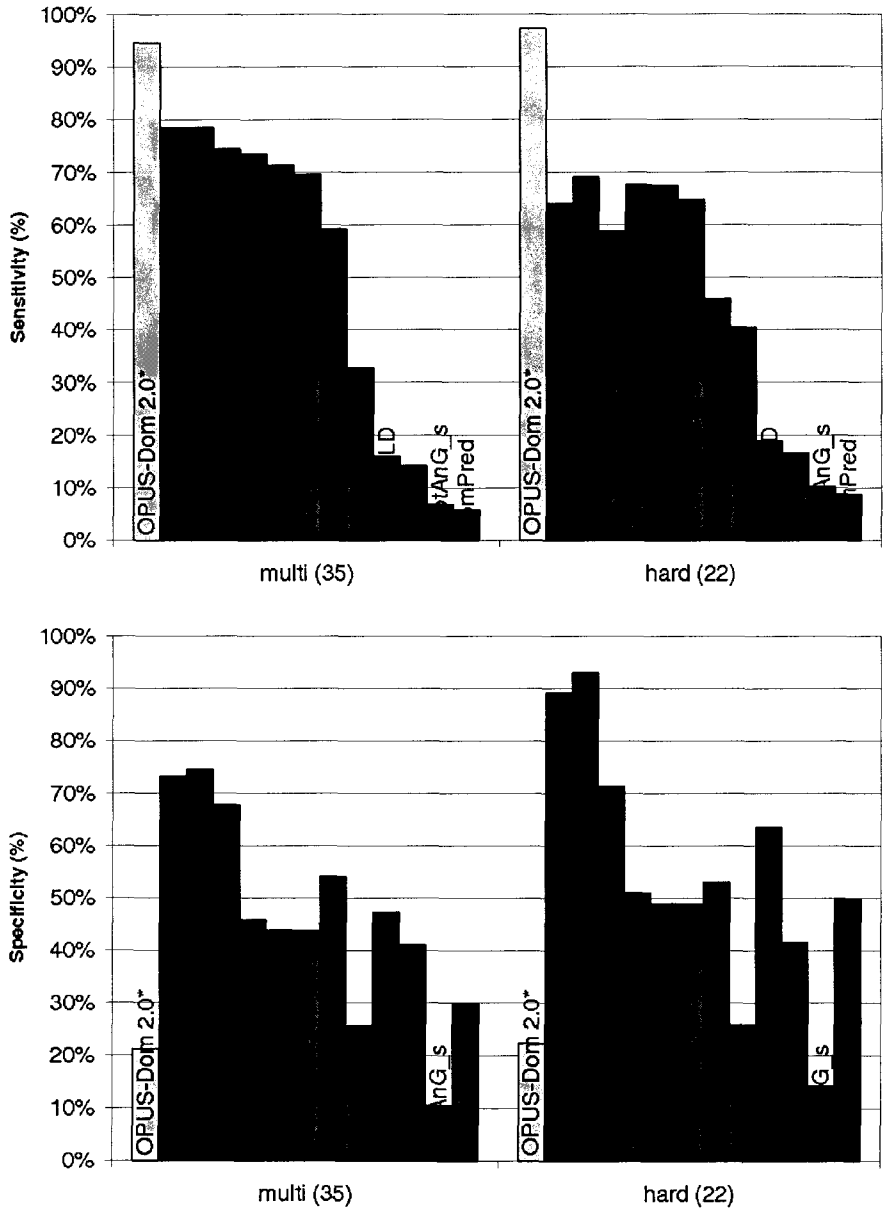

Figure 5.4 : Comparison of sensitivity and specificity for two variants of OPUS-Dom 2, optimal-sensitivity (yellow) and balanced-sensitivity-specificity (green), versus the top CASP8 domain predictors. The two CASP8 subsets used are the 35-target multi-domain set and 22-target hard set defined by Ezkurdia *et al.*[41].

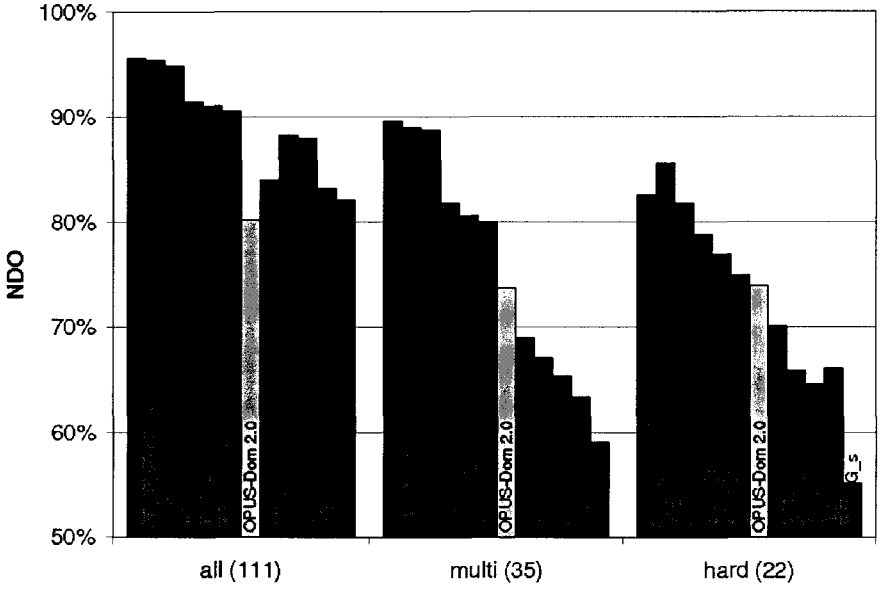

Figure 5.5 : Comparison of normalized domain overlap for OPUS-Dom 2 (yellow) versus the top CASP8 domain predictors, over the 111-target "all" CASP8 set, the 35-target multi-domain set and 22-target hard set defined by Ezkurdia *et al.*[41]. Note that the "all" set excludes targets T0471 and T0492 because VecFold1 could not generate sufficient structure models.

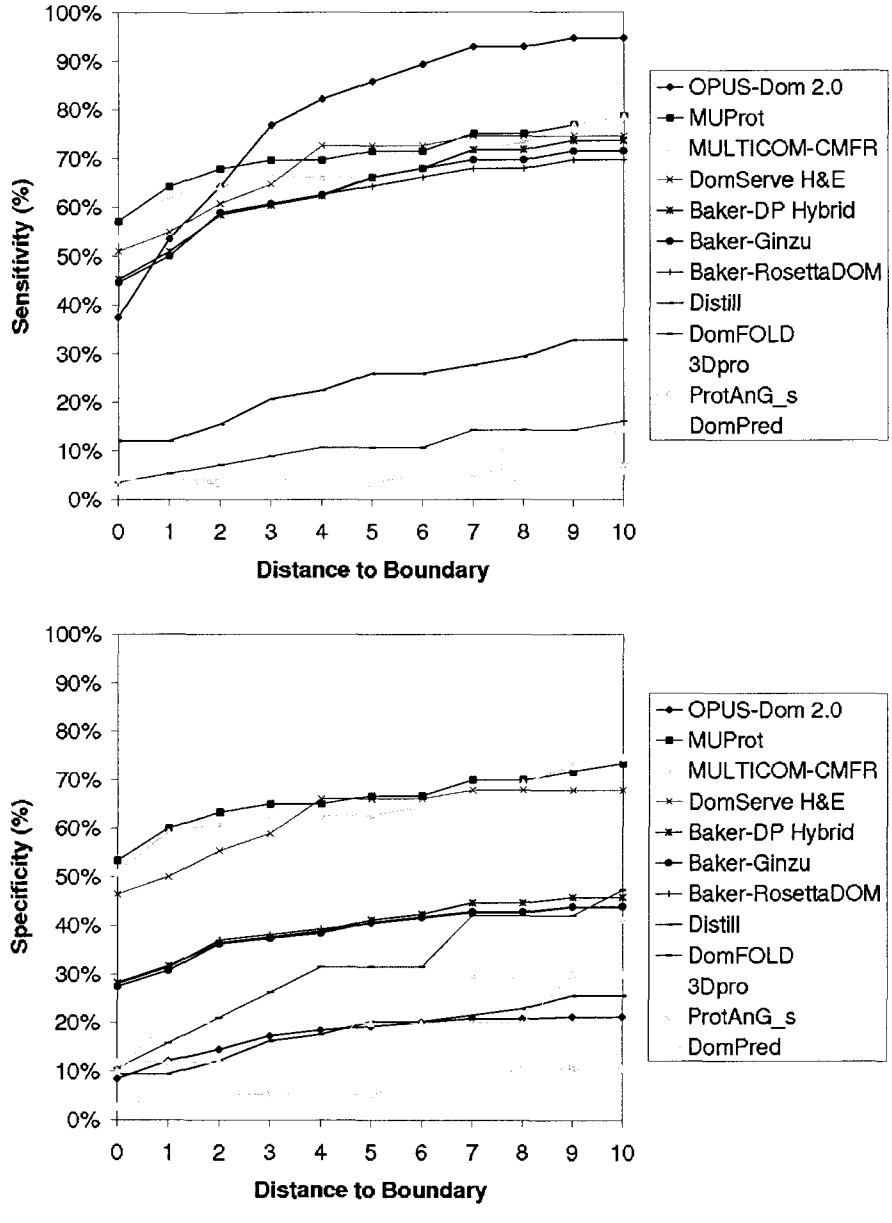

Figure 5.6 : Comparison of sensitivity and specificity measured at distance tolerance intervals of 1 to 10 residues from the official domain boundary, for OPUS-Dom 2 versus the top CASP8 domain predictors.

Part III

Protein folding core prediction

Understanding the mechanisms by which proteins fold is one of the grand challenges of molecular biology. Theoretical studies suggest a funnel-like free energy landscape for protein folding, which helps to explain how an extended polypeptide chain consistently folds into its stable native three-dimensional conformation in a speedy fashion [121, 106, 107, 84].

Theoretical and in vitro experiments suggest that protein folding nuclei, or cores, form early in the folding process [23, 111, 112, 95, 31, 32, 30, 18, 85]. This finding, in turn, supports Hammond's postulate [51] that thermodynamics and kinetics are closely correlated in proteins and that proteins may have evolved to optimize both folding rate and native-state stability [114]. Our earlier combined experimental-theoretical study on *Pseudomonas aeruginosa* apo-azurin and another beta-sandwich protein demonstrated this correlation, in which the stable folding cores predicted by our energetic method also harbored the key residues involved in the folding transition [23].

Among the experimental methods to probe the protein folding process, protein hydrogen-deuterium exchange (HX) helps identify protein regions that are shielded from solvent and thus "protected" from deuterium exchange (i.e., resulting in a slower rate of exchange). Based on HX experiments, the hydrogen-bonded amide protons (NHs) that are most protected from deuterium exchange in the protein native state are often found in the same protein regions as the NHs protected earliest during the protein folding reaction, as well as those NHs that are most protected in partially-folded intermediate states of the protein [85, 75, 157]. In contrast, NHs in turns and loops are rarely among the very slowest protons to exchange. Therefore, HX is useful in identifying the slow-exchanging NHs that make up the protein folding core.

Several computational models have been developed that try to connect folding theory with experimental data on protein unfolding/folding kinetics. Examples are graph-theoretical approaches based on effective contact order [128, 154], several variants of a motion planning method [3, 4, 142, 143], molecular dynamics simulations of

unfolding fluctuations around the native state [70, 35], an unfolding approach using a secondary-structure contact network and minimum cuts [167], a simplified lattice-protein model of native-state HX [68], and a method that exploits a correlation between slowest exchanging cores and low conformational entropy [59]. The two most relevant examples of computational models, with respect to this study, are the Floppy Inclusions and Rigid Substructure Topography (FIRST) method [53] and the Gaussian Network Model (GNM) [114, 34]. In the FIRST method, inter-atomic covalent and hydrogen bonds and hydrophobic interactions are replaced by rigid bars whose lengths and bond angles are constrained-only bond rotations are allowed. FIRST then identifies the rigid and flexible parts of the all-atomic protein model by selectively breaking hydrogen bonds in order of weakest to strongest. The GNM method coarse-grains a protein into an elastic network of residues, whereby pairs of residues within a cut-off distance are connected by virtual elastic springs, and it predicts the stable folding cores by studying the collective motions of the elastic network. In GNM, slow mode minima imply hinge sites, whereas high frequency mode peaks indicate stable "kinetically hot" residues.

Despite some success with these computational methods, there remains room for improvement. Empirical potential functions have been used previously to study changes in protein stability [25, 50, 15]. In our former work [23], we developed an empirically-weighted set of statistical potential functions and used them to analyze interaction energies among secondary-structure elements in two β-sandwich proteins. In the current study, we test the power of our empirical potential functions by applying them to the prediction of protein-folding cores as revealed by HX experiments, using a large set of proteins with different structures.

Here, and in earlier studies [85, 114], the experimental folding cores are defined as those that make up the folding core elements, which are the secondary structure elements (SSEs) containing the slowest exchanging residues (those with the greatest protection factors) identified in HX experiments. Using a set of 29 unrelated proteins

that were extensively studied in the literature, we show that, on average, our predictions correlate better with the experimentally-identified folding cores than those of two GNM methods and a third method using the FIRST software. We believe that our prediction method may be useful to facilitate a better understanding of the factors that dictate protein folding and native-state stability.

Chapter 6

OPUS-Core

6.1 Methods

6.1.1 Choice of experimental data and protein folding core prediction targets

HX experiments are typically subdivided into three types based on their detection purposes and experimental settings [85]: slow exchange core experiments (NHs most protected in the native state), pulsed exchange experiments (NHs first protected during folding), and folding competition experiments (NHs most protected in partially folded species). The folding core secondary structure elements (SSEs) revealed by these three methods are often identical or very similar. Thus, we follow Rader & Bahar [114] in using experimental data from slow exchange core experiments, the most abundant experimental folding core data in the literature, as our prediction targets. In addition, the secondary structure definitions are based on the Protein Data Bank SHEET and HELIX records.

To train our empirical potential function and then compare our computational predictions with experimental results, we used a set of 29 proteins (listed in Table 6.1) that were extensively studied in the literature [85, 114, 118, 28, 165, 6, 49, 77, 83].

6.1.2 Prediction of folding cores based on an empirical potential function OPUS-Core

The computational prediction method using our all-atom empirical potential function OPUS-Core is described in detail in our previous work [25]. The stability cores are

(a) Testing set

	Name	Abbrev.	PDB ID	Residues	SSEs
1	Apo-myoglobin	apoMb	1mbo	151	8
2	Barnase	Bnase	1a2p	108	8
3	Cytochrome c	Cytc	1hrc	104	5
4	T4 lysozyme	T4lzm	2lzm	164	14
5	Ribonuclease T1	RnaseT1	9rnt	104	8
6	a-Lactalbumin	ha-LA	1hml	123	12
7	Chymotrypsin inhibitor 2	CI2	2ci2	64	5
8	Ubiquitin	Ubq	1ubi	76	7
9	Bovine pancreatic trypsin inhibitor	BPTI	5pti	58	5
10	Interleukin-1b	IL-1b	1il1b	151	14
11	Hen egg-white lysozyme	HEWL	1hel	129	10
12	Equine lysozyme	Eqlzm	2eql	129	10
13	Protein A, B-domain	pAB	1bdd	60	3
14	Ribonuclease A	RnaseA	1rbx	124	10
15	Guinea pig a-lactalbumin	gpa-LA	1hfx	123	9
16	B1 Ig-binding domain protein G	GB1	1pga	56	5
17	B1 Ig-binding domain protein L	LB1	2ptl	78	5
18	Cardiotoxin analog III	CTX-3	2crt	60	5
19	Tendamistat	Tnds	2ait	74	6
20	Single chain antibody fragment[a]	scFv	2mcp	237	23
21	Human acidic fibroblast GF-1	hFGF-1	2afg	127	14
22	Cytochrome c551	pacc551	351c	82	5
23	Outer surface protein A	ospA	1osp:O	251	24
24	Ovomucoid third domain	OMTKY3	1iy5	54	4
25	Chicken src SH3 domain	cSH3	1srm	56	6
26	CheY	CheY	3chy	128	10
27	Human carbonic anhydrase I	HCA-1	1hcb	258	22

(b) Training set

28	Staphyloccoccal nuclease	SNase	1stn	136	9
29	Ribonuclease H	RnaseH	2rn2	155	10

[a] For consistency with previous studies [85, 114], we consider only the residues 1-115 in chain L and 1-122 in chain H.

Table 6.1 : Proteins used in study

ranked by the interaction energy between multiple SSEs (two, three, or four) using a scoring function:

$$S_{\text{core}} = 3.45\bar{E}_{\text{Packing}} + 5.0\bar{E}_{\text{AS}} + 1.9\bar{E}_{\text{HB}} \tag{6.1}$$

Here, the three terms in the scoring function represent the effects of side-chain packing ($\bar{E}_{\text{Packing}}$), solvent accessible surface area ($\bar{E}_{\text{AS}}$), and hydrogen bonding interactions ($\bar{E}_{\text{HB}}$), respectively. The parameters for these three terms are statistically derived from a non-redundant structure database of 2701 non-homologous soluble proteins [150], and the weight for each term is chosen by fitting to the folding core results of two proteins with the most consistent HX data [85], listed in Table 6.1(b). These two proteins, staphyloccoccal nuclease [78, 61] and ribonuclease H [17, 116], both have α-helix and β-sheet SSEs, and they are excluded from the set of 27 proteins used for cross-validation.

For comparison with the experimental HX results by Li & Woodward [85], we define the folding core as the group of SSEs with the lowest interaction energy. The interaction energies are calculated for groups of two, three, and four SSEs, and each grouping type is considered a separate but related method for predicting the folding core.

6.1.3 Evaluation of overlap between predictions and experiments

To compare our approach to previous methods and experimental results, we adopted the method for evaluating overlap employed by Rader & Bahar [114]. There are two related measures for the overlap between methods A and B (A and B may be experimental or computational prediction methods):

$$s(A,B) = \frac{o(A,B)}{\frac{N_A \cdot N_B}{N}} \tag{6.2}$$

$$z(A,B) = o(A,B) - \frac{N_A \cdot N_B}{N} \tag{6.3}$$

Here, N is the total number of residues in the target protein, N_A and N_B are the numbers of folding core residues revealed by methods A and B, respectively, and

$o(A, B)$ is the overlap in the number of residues revealed by methods A and B. These two quantities $s(A, B)$ and $z(A, B)$ measure the extent of difference between the observed overlap, $o(A, B)$, and the expected overlap for random matches, $N_A \cdot N_B/N$. Thus, $s = 1$ and $z = 0$ correspond to random matches and larger values of s and z indicate greater correlation between methods A and B.

6.2 Results

Figure 6.1 illustrates the folding cores predicted by HX experiments and the empirical potential function for a few examples within the 27-protein test set. Folding core elements are mapped as dark ribbons on the light gray 3D cartoon backbone of the protein structure. Each column represents one of the four methods (HX experiments; two-, three-, and four-SSE interaction groups).

Figure 6.2 summarizes the comparisons of the four methods for all 27 test proteins using the reduced representation from Rader & Bahar [114]. The x-axis corresponds to the residue index, and the stacked bars represent the experimentally-determined or predicted folding core elements. With the exceptions of ha-LA, CTX-3, and Eqlzm, the predictions yielded by the empirical potential function have substantial overlap with the experimental results. Figure 6.3 overlaps experimental phi-values with the folding core elements determined by the four methods for 10 of the 27 test proteins.

Tables 6.2 and 6.3 list the two measures of overlap (i.e., s and z in Equation 6.2 and Equation 6.3) for each of the 27 proteins in the test set in Table 6.1(a). The columns of Tables 6.2 and 6.3 compare the overlap between HX (X) and predictions based on the interaction energies (Equation 6.1) for groups of two, three, and four SSEs, as well as the prediction results of other computational methods. These other methods are the fast mode peak residues (H) [34], FIRST (F) [53], and GNM global modes (G) [9] methods. The results show that our method consistently outperforms the three previous studies in terms of the mean values of s and z. The lowest mean value $\langle s \rangle = 2.254$ by our method is better than that of H, F, and G. For z, the

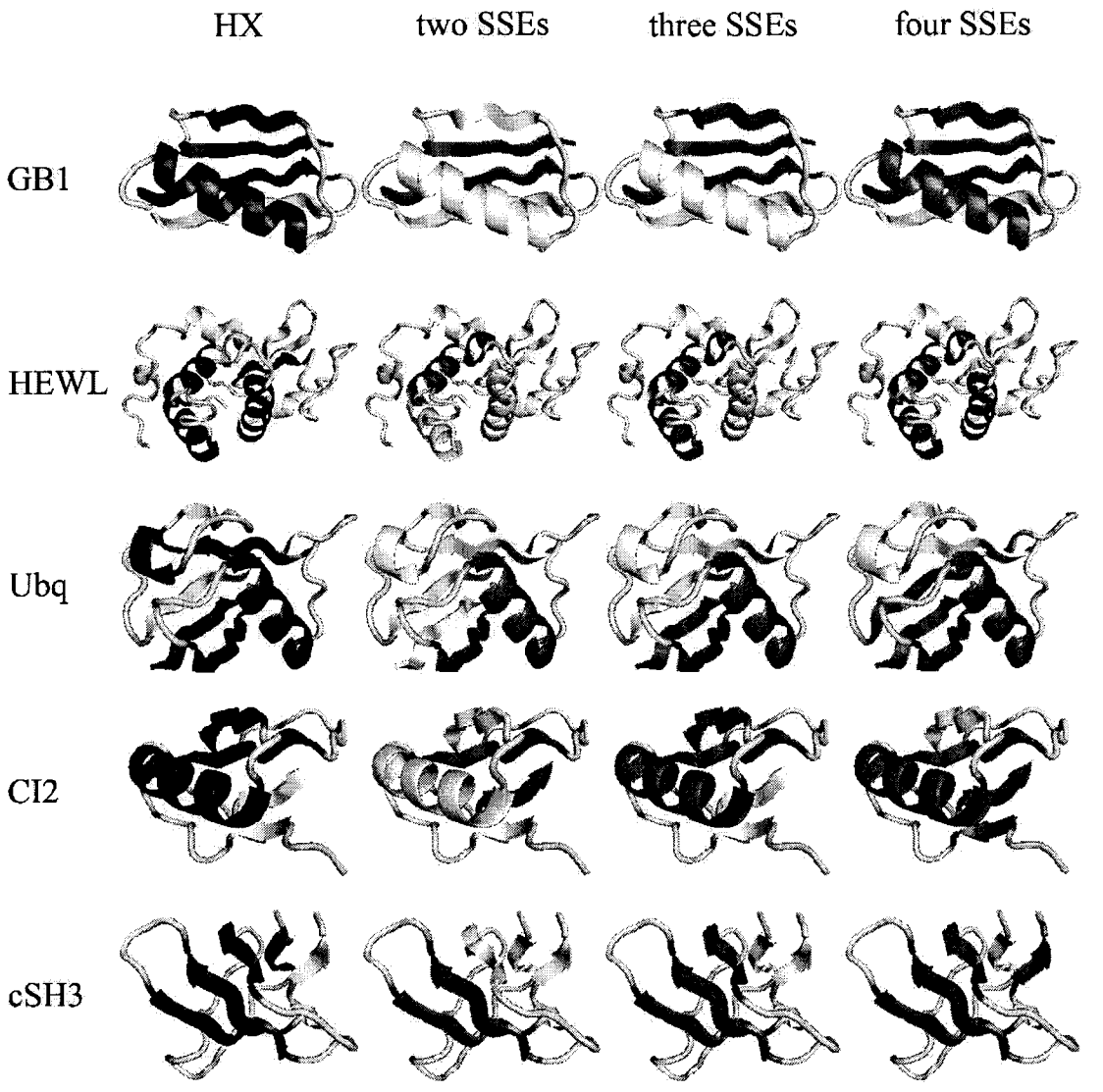

Figure 6.1 : Folding cores predicted by HX experiments and the empirical potential function for a few examples (GB1, HEWL, Ubiquitin, CI-2 and cSH3) within the 27-protein test set. Folding core elements are mapped as dark ribbons on the light gray 3D cartoon backbone of the protein structure. Each column represents one of the four methods (HX experiments; two-, three- and four-SSE interaction groups).[22] The cartoons were generated using PyMOL (DeLano Scientific, LLC).

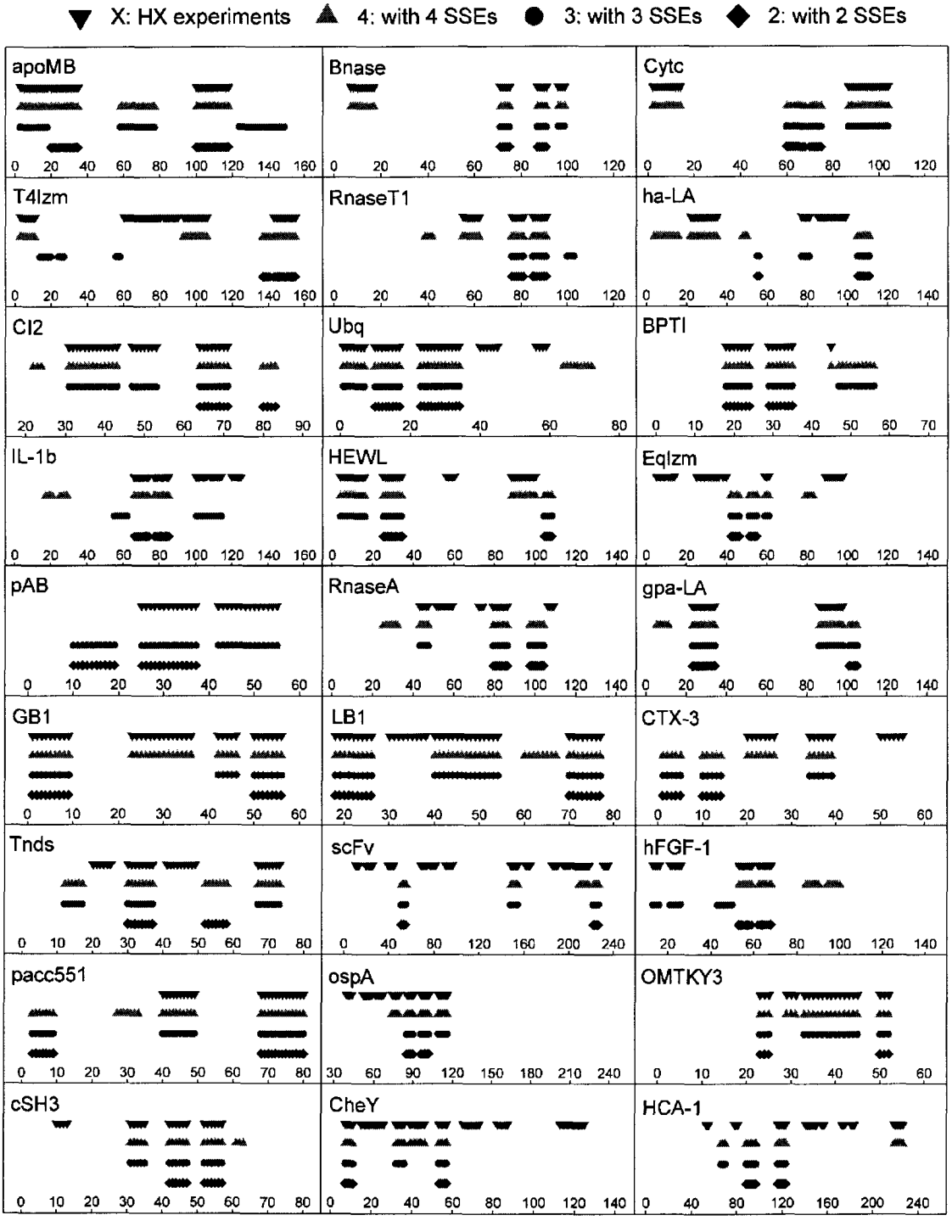

Figure 6.2 : Comparison of folding cores predicted by HX experiments and the empirical potential function (for four-, three- and two-SSE interaction groups) for all 27 test proteins using the reduced representation from Rader and Bahar [114]. The x-axis corresponds to the residue index, and the stacked bars represent the experimentally-determined or predicted folding core elements.[22]

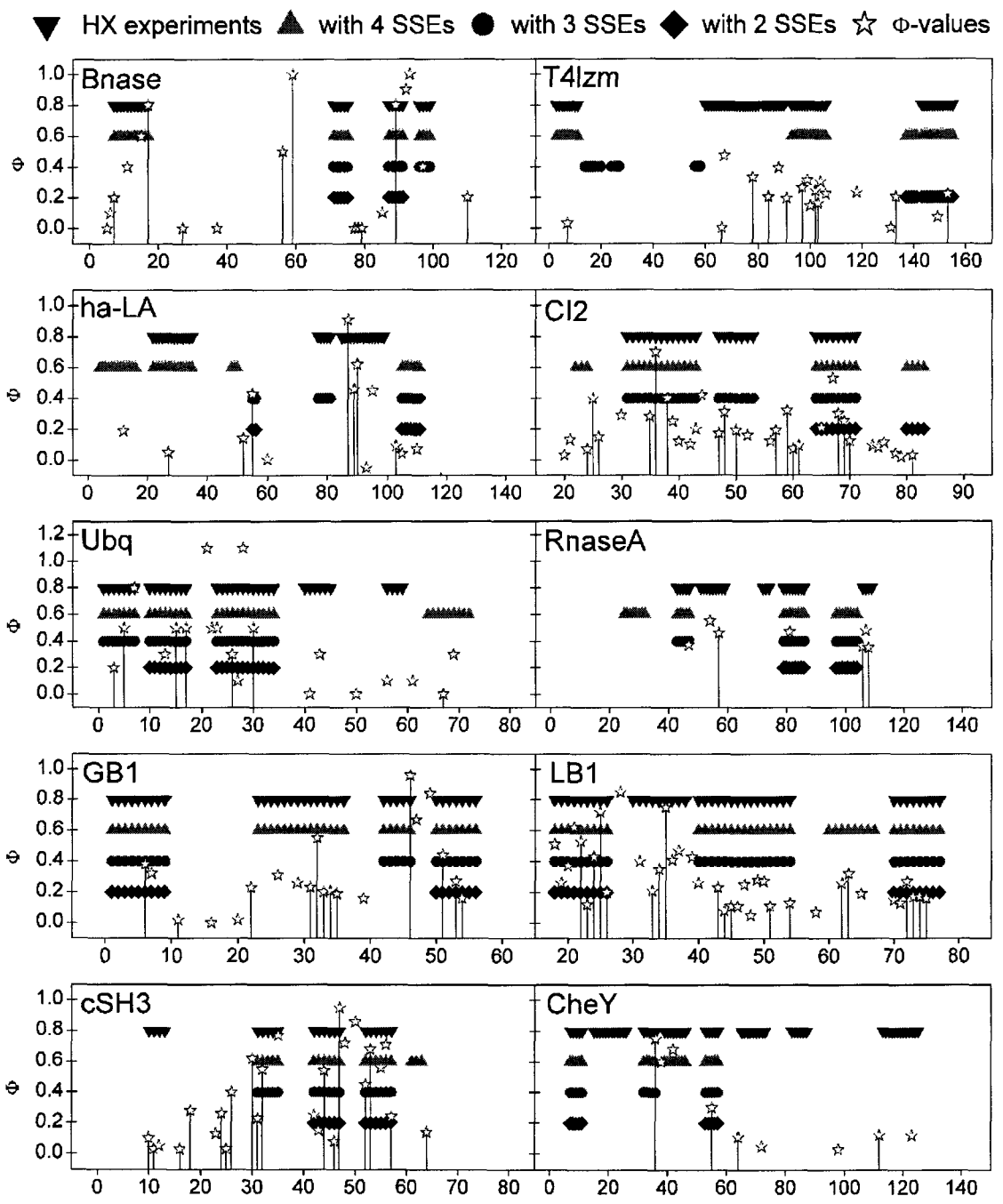

Figure 6.3 : Experimental phi-values for ten of the 27 test proteins, plotted as functions of residue index. The corresponding protein folding core elements determined by HX experiments and the empirical potential function (from Fig. 6.2) are provided for reference. The phi-values for GB1, CheY, Bnase, CI-2, cSH3 and LB1 were sourced from Garbuzynskiy *et al.* [45]. The phi-values for RnaseA, Ubiquitin, ha-LA and T4 lysozyme were drawn from Font *et al.* [43], Went and Jackson [155], Saeki *et al.* [119] and Kato *et al.* [67], respectively.[22]

smallest mean value by our method is for the two-SSE case ($\langle z \rangle = 5.718$), which is better than the mean values by H, F, and G.

For proteins HCA-1, CI-2, and cSH3, all versions of our method are better at matching the HX-detected folding cores than the other methods. However, for ha-LA and Eqlzm, the H, F, and G methods are generally better than our method in predicting the HX-detected folding cores. For nearly half of the test proteins (13 of 27), all versions of our method match the HX results with greater than 100% improvement over random agreement ($s > 2.0$), whereas G can claim only 6 of 27, H can claim 10 of 27, and F can claim 11 of 27 with $s > 2.0$. In addition, for Bnase and RnaseT1, all methods but G match the HX results with roughly 200% or better improvement over random agreement ($s > 3.0$). The success of our method in predicting the folding cores of Bnase and RnaseT1 may be due to the use of nucleases RnaseH and Snase in our training set. Interestingly, all methods perform poorly for pAB, which is a small three-helix protein. It is possible that for such a small and symmetrical protein, all elements have rather similar contributions to overall stability.

In addition, we tested our method on a few proteins (Cytc, ha-LA, scFv, and IL-1b) whose secondary structure definitions, namely the number of SSEs, were modified in the PDB header within the past three years. For ha-LA and scFv, the folding core predictions changed with the increase in the number of SSEs, whereas the predictions remained the same for IL-1b. Furthermore, although the overlap measures s and z declined for ha-LA and scFv with the increase in SSEs, we found no overall correlation between the number of SSEs and our performance in terms of s and z. In fact, we found little correlation between the number of SSEs and overlap performance for all the proteins in the test set (see Figure 6.4).

For ten of the proteins in our data set, the transient folding-transition states have been assessed by the phi-value approach [128, 89, 60, 119, 45, 67, 155, 122, 43]. This is an experimental approach to indirectly obtain residue-specific structural information about interactions in the transition state pioneered by Fersht [98]. It is

	s(X,two)	s(X,three)	s(X,four)	s(X,H)	s(X,F)	s(X,G)
apoMb	3.000	0.774	2.155	1.377	3.084	2.329
Bnase	4.320	4.320	4.320	3.333	2.971	1.909
Cytc	0.000	1.830	2.261	3.200	3.200	1.231
T4lzm	1.795	0.000	2.182	1.268	0.702	2.161
RnaseT1	4.952	3.787	4.160	4.370	3.294	1.359
ha-LA	0.000	1.292	1.490	1.491	1.435	2.681
CI2	1.548	2.321	1.741	1.103	0.940	0.768
Ubq	2.054	2.054	1.541	1.827	1.070	1.247
BPTI	3.867	2.256	2.320	1.726	3.255	1.184
IL-1b	3.974	2.529	2.555	1.348	2.555	1.648
HEWL	2.098	2.584	2.763	1.929	0.896	0.860
Eqlzm	0.000	0.783	0.566	2.092	1.743	1.550
pAB	1.256	1.622	N/A	1.607	1.382	0.964
RnaseA	2.138	2.647	1.917	3.000	1.444	1.091
gpa-LA	3.473	4.100	3.324	0.447	2.811	2.916
GB1	1.600	1.600	1.600	1.667	1.355	0.602
LB1	1.902	1.902	1.522	2.182	2.086	1.773
CTX-3	0.000	1.184	1.785	3.195	1.846	2.517
Tnds	1.316	1.762	1.321	2.921	0.974	1.263
scFv	0.000	1.138	1.776	1.484	1.467	1.240
hFGF-1	5.375	2.986	2.688	4.233	2.540	0.977
pacc551	2.317	2.733	2.216	1.621	0.000	2.228
ospA	5.457	5.457	5.457	3.508	1.960	1.364
OMTKY3	2.455	2.455	2.455	1.142	2.077	1.350
cSH3	2.667	2.667	2.667	1.778	2.545	1.167
CheY	2.032	2.032	2.032	1.173	1.365	1.102
HCA-1	2.606	2.040	3.115	0.701	1.020	0.623
mean	2.304	2.254	2.382	2.004	1.991	1.509
stdev	1.618	1.174	1.040	1.044	1.265	0.729

Table 6.2 : The correlation measure of overlap s between predictions and experiments. The folding cores are determined by HX slow exchange (X), the empirical potential function for two, three, and four SSEs (two, three, four), and the methods of fast mode peak residues (H), FIRST (F), GNM global modes (G).

	z(X,two)	z(X,three)	z(X,four)	z(X,H)	z(X,F)	z(X,G)
apoMb	23.333	-4.667	27.333	1.642	16.219	16.550
Bnase	7.685	10.759	19.213	7.000	17.250	10.000
Cytc	-4.471	8.163	17.288	5.500	16.500	0.750
T4lzm	5.756	-5.634	19.500	1.689	-5.512	10.744
RnaseT1	10.375	9.567	15.952	7.712	4.875	2.115
ha-LA	-2.488	1.130	4.602	1.317	4.244	10.659
CI2	2.831	15.938	8.938	0.563	-1.469	-4.844
Ubq	10.263	13.855	9.474	4.526	2.105	4.158
BPTI	10.379	7.793	8.534	2.103	7.621	0.621
IL-1b	13.470	8.464	10.954	0.775	16.430	4.325
HEWL	5.233	14.101	22.969	3.372	-1.395	-0.977
Eqlzm	-2.946	-0.829	-2.302	3.132	3.411	4.256
pAB	2.650	10.350	N/A	3.400	5.533	-0.333
RnaseA	4.258	8.089	6.218	6.000	4.000	0.500
gpa-LA	8.545	18.902	17.480	-1.236	7.732	10.512
GB1	6.000	7.875	13.125	4.804	1.571	-1.321
LB1	8.064	15.179	10.974	6.500	7.808	6.538
CTX-3	-3.167	0.933	5.717	6.183	4.583	3.617
Tnds	1.919	6.486	3.649	5.919	-0.135	2.500
scFv	-3.759	0.848	7.430	1.304	9.228	7.544
hFGF-1	11.395	6.651	8.791	4.583	2.425	-0.094
pacc551	7.390	14.585	12.622	1.915	-6.451	8.268
ospA	10.618	17.151	22.052	6.434	18.124	3.470
OMTKY3	3.556	11.259	13.037	5.185	1.370	2.593
cSH3	7.500	10.625	9.500	5.464	7.000	0.429
CheY	5.078	7.617	11.680	2.672	5.164	2.781
HCA-1	4.930	4.078	11.543	0.039	-2.558	-7.256
mean	5.718	8.121	12.164	3.391	5.285	3.703
stdev	6.106	6.377	6.663	2.542	6.649	5.334

Table 6.3 : The correlation measure of overlap z between predictions and experiments. The methods assessed (by column) are similar to those in Table 6.2.

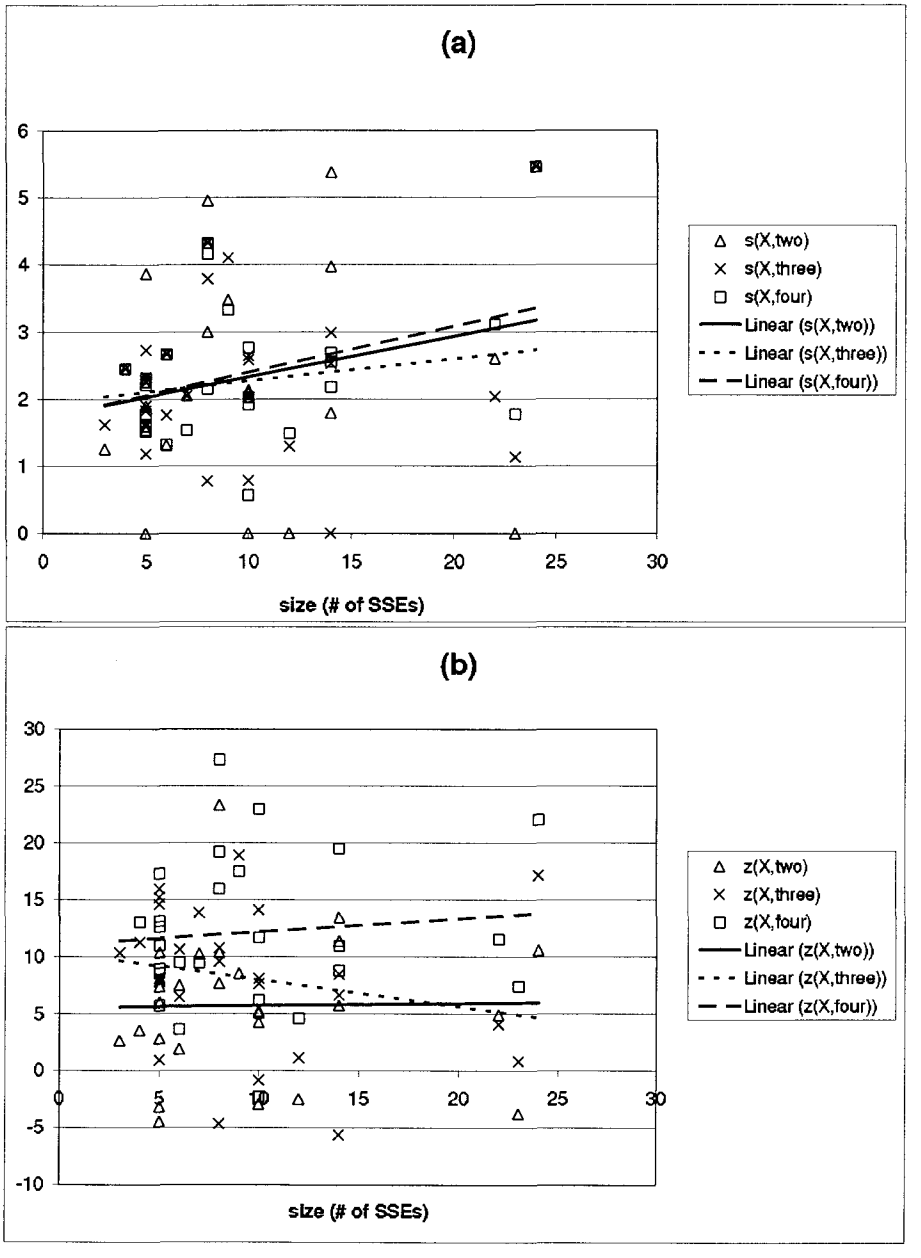

Figure 6.4 : Number of SSEs versus the correlation measures of overlap between predictions and experiments. The folding cores are determined by HX slow exchange (X) and the empirical potential function for two, three, and four SSEs (two, three, four). (a) shows results for the measure of overlap s, and (b) shows results for the measure of overlap z.[22]

often assumed that the folding core found in HX experiments corresponds to the region adopting native-like structure in the kinetic folding-transition state [85]. For some of the proteins having polarized, highly-organized transition-state structures (e.g., cSH3, Bnase, Ubiquitin and ha-LA), as identified by phi-values, our method selects the same structural elements as those harboring residues with high phi-values (see Figure 6.3). In contrast, for proteins with diffuse folding-transition states (i.e., GB1, CI-2, RNase A, T4 lysozyme), there is less correlation between phi-values and our predicted folding cores (or between HX data as well). Taken together, we conclude that the stable folding cores, as identified by our empirical method or by HX data, often match the kinetic folding-transition states although these sometimes differ; for proteins folding via diffuse transition states involving many partially-formed interactions, the stable folding cores must be assessed by methods other than phi-values.

Part IV

Conclusion

Chapter 7

Concluding discussions

7.1 VecFold1

VecFold1 is our unique coarse-grained, vector-based sampling method, and it is very fast. It can easily handle targets with more than 600 residues and achieve thousands of trajectories in a short time on single processor. For example, in our study, VecFold1 generated 10^4 trajectories in roughly 0.016 hours per residue (on average for the Miyazaki 12 test set) on a single 900MHz Intel Itanium2 processor, i.e., VecFold1 can generate 10^4 compact tertiary structure models of a 250-residue protein in roughly 4 hours. Because VecFold1 is very efficient in searching protein conformational space, it might sample the protein native topology with far fewer steps than other residue-based or atom-based folding algorithms. Furthermore, as evidenced by the benchmark results, VecFold1 seems to generate structures with global features that are roughly consistent with the native structure. Two weaknesses of VecFold1 are that it (a) models loops very crudely and (b) is sensitive to the initial secondary structure prediction by PSI-PRED.

7.2 VecFold2

VecFold2 is the newest version of our SSSM-based assembly method for tertiary structure prediction. Compared to VecFold1, VecFold2 generates more accurate C^{α} structure models using a more sophisticated sampling method and a more accurate scoring function, at the cost of increased computational complexity. In addition, in terms of summary statistics, VecFold2 is fairly insensitive to the initial step of secondary struc-

ture prediction, correcting a weakness in VecFold1.

With VecFold2, we demonstrate that our SSSM-based assembly method, combined with our OPUS-Ca scoring function, can modestly outperform one of the top tertiary structure predictors, Rosetta. We tested several variants of VecFold2 with different fragment assembly strategies and template libraries to identify the source of the improved performance, and it appears that the sampling method is the key to the gains, although it is left to future studies to determine how these gains break down between profile alignment and SSSM-based assembly.

Even so, the template-free tertiary structure prediction problem is far from solved, and neither VecFold2 nor Rosetta is sufficient alone. In combination, however, the two methods can yield better results, even if the aggregate gains in performance may be small.

7.3 OPUS-Dom

Based on our encouraging benchmark results on several different testing sets, OPUS-Dom is a potentially useful tool to help experimentalists determine possible domain regions for large proteins. OPUS-Dom generally outperformed the other methods in our benchmark comparison (DLI, REI, GHL, KDH, Armadillo, Rosetta-DOM) using the MMDB, GM, Miyazaki, and CASP6 sets, as indicated in Table 4.1 and Table 4.3. Furthermore, OPUS-Dom performed especially well relative to the state-of-the-art in predicting domain boundaries of structures with no available sequence homologs. In the case of the very difficult CASP7 prediction target T0356 (Figure 4.2(d) and Figure 4.3), for which few teams found the correct structure analogs, OPUS-Dom predicted two of the three boundaries. In addition, OPUS-Dom obtained the highest score for the comparably difficult CASP7 target T0301, which was highlighted in Fig.7d of Tress et al. [144] (our group ID was DP229). Furthermore, OPUS-Dom is several times faster than other folding-based prediction methods such as Rosetta-DOM because of our unique coarse-grained, vector-based sampling method VecFold1.

A distinct advantage of folding-based domain boundary prediction methods is that they rely less on sequence homology information than purely-sequence-based methods. In our case, evolutionary information is used only to establish the library of structure fragment candidates, whereas the overall tertiary structure is determined by VecFold1. Although each folded structure carries relatively large errors, the statistical consensus derived from an ensemble of such structures reveals a much more robust delineation of domain boundaries. This suggests that the domain arrangement in protein structures is less a consequence of sequence-specific constraints. Instead, tertiary structure packing of secondary structure segments yields a limited set of coordination patterns per domain. An interesting analogy is the short-range ordering in simple liquids [129]. Although the instantaneous configuration of liquid molecules may be constantly changing, the position of the peak of the first coordination shell, as manifested by the radial distribution function, is very stable for a given set of macroscopic environmental parameters. Such a static pattern of thickness in coordination shells resembles the domain volume arrangement in proteins, i.e., if the vectorial arrangement of secondary structure segments are given, tertiary packing of a single domain would result in a relatively stable domain volume distribution due to topological constraints.

It is worth mentioning that decomposing protein structures into domains is challenging even for human experts [147] because of the diverse nature of domains. Furthermore, structure-based domain parsing methods such as Domain-Parser [164], NCBI [94], PDP [1], and PUU [57] cannot yet match the level of accuracy of human experts [147]. Thus, the inconsistency of domain assignments makes the unbiased assessment of prediction results even more difficult. For example, in the CASP domain prediction competition, assessors use human inspection and several automatic domain assignment methods to define the domains, often assigning multiple domain definitions to a protein target [145].

Finally, OPUS-Dom successfully defines the domain boundaries, but it is less

reliable in specifying the domain identity of a polypeptide segment in between two contiguous domain boundaries. For example, consider a two-domain protein in which the sequence of domain A is continuous, and domain B is formed by two separate sequence fragments (B1 and B2), yielding a sandwich-like B1-A-B2 sequence motif. In this case, it is harder to firmly specify which fragment belongs to which domain, even though the domain boundaries are determined via statistical analysis. This weakness is addressed in OPUS-Dom 2.

7.4 OPUS-Dom2

Version 2 of OPUS-Dom is a major improvement from the first version of our method, partly due to the more accurate structure models by VecFold2 and partly to a more advanced approach to identifying domains from aggregate label profiles generated by a structure-based domain parser such as Taylor's method. A simple way to further improve OPUS-Dom is to replace Taylor's method with a scheme that combines results from multiple domain parsers.

In addition, OPUS-Dom 2 is able to identify split domain regions, e.g., it can identify the B1-A-B2 domain sandwich motif as described above in Section 7.3. This is illustrated with CASP8 target T0418 in Figure 7.1.

Like the original OPUS-Dom, OPUS-Dom 2 requires no sequence homology. Another major strength of OPUS-Dom 2 is that its prediction sensitivity may be tuned to substantially outperform the best domain prediction methods with a modest penalty in prediction specificity. We demonstrate this performance gain using the CASP8 benchmark set, despite the fact that the most successful CASP8 domain predictors can switch between homology-based and *de novo* components to take advantage of the fact that most CASP8 targets comprised of at least one domain for which a template could be found [146, 41].

We focus our domain prediction assessment on prediction sensitivity because this is a very important domain prediction feature to experimental structural biologists.

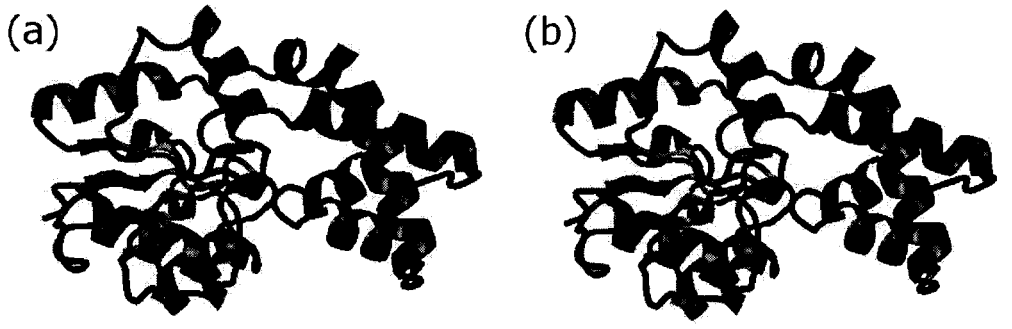

Figure 7.1 : Split domains in CASP8 target T0418 identified by OPUS-Dom 2. T0418 is a 222-residue two-domain protein. (a) Officially, domain B spans residues 1-16 and 86-211, and domain A spans residues 17-85. (b) OPUS-Dom 2 identifies domain B as residues 1-18 and 81-222, and domain A as residues 19-80.

Crystallographers, for example, want to know roughly where to cut the protein chain, and a few more false positive domain boundaries is acceptable if the domains exist in the fragments with high certainty, as the total number of possible fragments is $N = (B + 1)(B + 2)/2$ (i.e., the sum of a simple finite arithmetic series), where B is the number of domain boundaries. One direction for improving domain prediction is to computationally split a protein target into all possible domain regions, generate an ensemble of tertiary structures for each putative domain region using VecFold2, and then select the most energetically-favorable combination of domains.

7.5 OPUS-Core

OPUS-Core is an empirical potential function that can detect protein-stability cores revealed by HX experiments. The average prediction results of our method are better than those of previous computational attempts. Although there is still room for improvement in the model, we believe the method reported here provides a more accurate way of estimating stability cores of proteins that can be useful in elucidating the mechanisms of protein folding.

Bibliography

[1] N. Alexandrov and I. Shindyalov. Pdp: protein domain parser. *Bioinformatics*, 19(3):429–30, 2003.

[2] S. F. Altschul, T. L. Madden, A. A. Schaffer, 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):3389–402, 1997.

[3] N. M. Amato, K. A. Dill, and G. Song. Using motion planning to map protein folding landscapes and analyze folding kinetics of known native structures. *J Comput Biol*, 10(3-4):239–55, 2003.

[4] N. M. Amato and G. Song. Using motion planning to study protein folding pathways. *J Comput Biol*, 9(2):149–68, 2002.

[5] C. B. Anfinsen. Principles that govern the folding of protein chains. *Science*, 181(96):223–30, 1973.

[6] C. B. Arrington, L. M. Teesch, and A. D. Robertson. Defining protein ensembles with native-state nh exchange: kinetics of interconversion and cooperative units from combined nmr and ms analysis. *J Mol Biol*, 285(3):1265–75, 1999.

[7] A. Aszoadi and W. R. Taylor. Folding polypeptide alpha-carbon backbones by distance geometry methods. *Biopolymers*, 34(4):489–505, 1994.

[8] K. Bae, B. K. Mallick, and C. G. Elsik. Prediction of protein interdomain linker regions by a hidden markov model. *Bioinformatics*, 21(10):2264–70, 2005.

[9] I. Bahar, A. Wallqvist, D. G. Covell, and R. L. Jernigan. Correlation between native-state hydrogen exchange and cooperative residue fluctuations from a simple model. *Biochemistry*, 37(4):1067–75, 1998.

[10] 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):235–42, 2000.

[11] R. Bondugula, M. S. Lee, and A. Wallqvist. Fiefdom: a transparent domain boundary recognition system using a fuzzy mean operator. *Nucleic Acids Res*, 37(2):452–62, 2009.

[12] R. Bonneau and D. Baker. Ab initio protein structure prediction: progress and prospects. *Annu Rev Biophys Biomol Struct*, 30:173–89, 2001.

[13] R. Bonneau, C. E. Strauss, C. A. Rohl, D. Chivian, P. Bradley, L. Malmstrom, T. Robertson, and D. Baker. De novo prediction of three-dimensional structures for major protein families. *J Mol Biol*, 322(1):65–78, 2002.

[14] W. Boomsma and T. Hamelryck. Full cyclic coordinate descent: solving the protein loop closure problem in calpha space. *BMC Bioinformatics*, 6:159, 2005.

[15] A. J. Bordner and R. A. Abagyan. Large-scale prediction of protein geometry and stability changes for arbitrary single point mutations. *Proteins*, 57(2):400–13, 2004.

[16] K. Bryson, D. Cozzetto, and D. T. Jones. Computer-assisted protein domain boundary prediction using the dompred server. *Curr Protein Pept Sci*, 8(2):181–8, 2007.

[17] A. K. Chamberlain, T. M. Handel, and S. Marqusee. Detection of rare partially folded molecules in equilibrium with the native conformation of rnaseh. *Nat Struct Biol*, 3(9):782–7, 1996.

[18] L. L. Chavez, J. N. Onuchic, and C. Clementi. Quantifying the roughness on the free energy landscape: entropic bottlenecks and protein folding rates. *J Am Chem Soc*, 126(27):8426–32, 2004.

[19] J. Chen, J. B. Anderson, C. DeWeese-Scott, N. D. Fedorova, L. Y. Geer, S. He, D. I. Hurwitz, J. D. Jackson, A. R. Jacobs, C. J. Lanczycki, C. A. Liebert, C. Liu, T. Madej, A. Marchler-Bauer, G. H. Marchler, R. Mazumder, A. N. Nikolskaya, B. S. Rao, A. R. Panchenko, B. A. Shoemaker, V. Simonyan, J. S. Song, P. A. Thiessen, S. Vasudevan, Y. Wang, R. A. Yamashita, J. J. Yin, and S. H. Bryant. Mmdb: Entrez's 3d-structure database. *Nucleic Acids Res*, 31(1):474–7, 2003.

[20] Jen-Ming Chen, Jose A. Ventura, and Chih-Hang Wu. Segmentation of planar curves into circular arcs and line segments. *Image and Vision Computing*, 14(1):71–83, 1996.

[21] L. Chen, W. Wang, S. Ling, C. Jia, and F. Wang. Kemadom: a web server for domain prediction using kernel machine with local context. *Nucleic Acids Res*, 34(Web Server issue):W158–63, 2006.

[22] M. Chen, A. D. Dousis, Y. Wu, P. Wittung-Stafshede, and J. Ma. Predicting protein folding cores by empirical potential functions. *Arch Biochem Biophys*, 483(1):16–22, 2009.

[23] M. Chen, C. J. Wilson, Y. Wu, P. Wittung-Stafshede, and J. Ma. Correlation between protein stability cores and protein folding kinetics: a case study on pseudomonas aeruginosa apo-azurin. *Structure*, 14(9):1401–10, 2006.

[24] J. Cheng. Domac: an accurate, hybrid protein domain prediction server. *Nucleic Acids Res*, 35(Web Server issue):W354–6, 2007.

[25] J. Cheng, A. Randall, and P. Baldi. Prediction of protein stability changes for

single-site mutations using support vector machines. *Proteins*, 62(4):1125–32, 2006.

[26] Jianlin Cheng and Pierre Baldi. A machine learning information retrieval approach to protein fold recognition. *Bioinformatics*, 22(12):1456–1463, 2006.

[27] Jianlin Cheng, J. Sweredoski Michael, and Pierre Baldi. Dompro: Protein domain prediction using profiles, secondary structure, relative solvent accessibility, and recursive neural networks. *Data Mining and Knowledge Discovery*, 13(1):1–10, 2006.

[28] Y. H. Chi, T. K. Kumar, K. M. Kathir, D. H. Lin, G. Zhu, I. M. Chiu, and C. Yu. Investigation of the structural stability of the human acidic fibroblast growth factor by hydrogen-deuterium exchange. *Biochemistry*, 41(51):15350–9, 2002.

[29] D. Chivian, D. E. Kim, L. Malmstrom, P. Bradley, T. Robertson, P. Murphy, C. E. Strauss, R. Bonneau, C. A. Rohl, and D. Baker. Automated prediction of casp-5 structures using the robetta server. *Proteins*, 53 Suppl 6:524–33, 2003.

[30] C. Clementi, P. A. Jennings, and J. N. Onuchic. How native-state topology affects the folding of dihydrofolate reductase and interleukin-1beta. *Proc Natl Acad Sci U S A*, 97(11):5871–6, 2000.

[31] C. Clementi, P. A. Jennings, and J. N. Onuchic. Prediction of folding mechanism for circular-permuted proteins. *J Mol Biol*, 311(4):879–90, 2001.

[32] C. Clementi, H. Nymeyer, and J. N. Onuchic. Topological and energetic factors: what determines the structural details of the transition state ensemble and "en-route" intermediates for protein folding? an investigation for small globular proteins. *J Mol Biol*, 298(5):937–53, 2000.

[33] Wendy D. Cornell, Piotr Cieplak, Christopher I. Bayly, Ian R. Gould, Kenneth M. Merz, David M. Ferguson, David C. Spellmeyer, Thomas Fox, James W. Caldwell, and Peter A. Kollman. A second generation force field for the simulation of proteins, nucleic acids, and organic molecules. *Journal of the American Chemical Society*, 117(19):5179–5197, 1995.

[34] M. C. Demirel, A. R. Atilgan, R. L. Jernigan, B. Erman, and I. Bahar. Identification of kinetically hot residues in proteins. *Protein Sci*, 7(12):2522–32, 1998.

[35] N. V. Dokholyan, S. V. Buldyrev, H. E. Stanley, and E. I. Shakhnovich. Identifying the protein folding nucleus using molecular dynamics. *J Mol Biol*, 296(5):1183–8, 2000.

[36] Q. Dong, X. Wang, L. Lin, and Z. Xu. Domain boundary prediction based on profile domain linker propensity index. *Comput Biol Chem*, 30(2):127–33, 2006.

[37] M. Dumontier, R. Yao, H. J. Feldman, and C. W. Hogue. Armadillo: domain boundary prediction by amino acid composition. *J Mol Biol*, 350(5):1061–73, 2005.

[38] J. G. Dunham. Optimum uniform piecewise linear approximation of planar curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(1):67–75, 1986.

[39] T. Ebina, H. Toh, and Y. Kuroda. Loop-length-dependent svm prediction of domain linkers for high-throughput structural proteomics. *Biopolymers*, 92(1):1–8, 2009.

[40] N. Eswar, D. Eramian, B. Webb, M. Y. Shen, and A. Sali. Protein structure modeling with modeller. *Methods Mol Biol*, 426:145–59, 2008.

[41] I. Ezkurdia, O. Grana, J. M. Izarzugaza, and M. L. Tress. Assessment of domain boundary predictions and the prediction of intramolecular contacts in casp8. *Proteins*, 77 Suppl 9:196–209, 2009.

[42] C. A. Floudas, H. K. Fung, S. R. McAllister, M. Monnigmann, and R. Rajgaria. Advances in protein structure prediction and de novo protein design: A review. *Chemical Engineering Science*, 61(3):966–988, 2006.

[43] J. Font, A. Benito, R. Lange, M. Ribo, and M. Vilanova. The contribution of the residues from the main hydrophobic core of ribonuclease a to its pressure-folding transition state. *Protein Sci*, 15(5):1000–9, 2006.

[44] O. V. Galzitskaya and B. S. Melnik. Prediction of protein domain boundaries from sequence alone. *Protein Sci*, 12(4):696–701, 2003.

[45] S. O. Garbuzynskiy, A. V. Finkelstein, and O. V. Galzitskaya. Outlining folding nuclei in globular proteins. *J Mol Biol*, 336(2):509–25, 2004.

[46] R. A. George and J. Heringa. Snapdragon: a method to delineate protein structural domains from sequence data. *J Mol Biol*, 316(3):839–51, 2002.

[47] J. E. Gewehr and R. Zimmer. Ssep-domain: protein domain prediction by alignment of secondary structure elements and profiles. *Bioinformatics*, 22(2):181–7, 2006.

[48] K. Ginalski, N. V. Grishin, A. Godzik, and L. Rychlewski. Practical lessons from protein structure prediction. *Nucleic Acids Res*, 33(6):1874–91, 2005.

[49] V. P. Grantcharova and D. Baker. Folding dynamics of the src sh3 domain. *Biochemistry*, 36(50):15685–92, 1997.

[50] R. Guerois, J. E. Nielsen, and L. Serrano. Predicting changes in the stability of proteins and protein complexes: a study of more than 1000 mutations. *J Mol Biol*, 320(2):369–87, 2002.

[51] George S. Hammond. A correlation of reaction rates. *Journal of the American Chemical Society*, 77(2):334–338, 1955.

[52] C. Hardin, T. V. Pogorelov, and Z. Luthey-Schulten. Ab initio protein structure prediction. *Curr Opin Struct Biol*, 12(2):176–81, 2002.

[53] B. M. Hespenheide, A. J. Rader, M. F. Thorpe, and L. A. Kuhn. Identifying protein folding cores from the evolution of flexible regions during unfolding. *J Mol Graph Model*, 21(3):195–207, 2002.

[54] D. Hoffmann and E. W. Knapp. Protein dynamics with off-lattice monte carlo moves. *Phys Rev E Stat Phys Plasmas Fluids Relat Interdiscip Topics*, 53(4):4221–4224, 1996.

[55] T. A. Holland, S. Veretnik, I. N. Shindyalov, and P. E. Bourne. Partitioning protein structures into domains: Why is it so difficult? *J. Mol. Biol.*, 361:562–590, 2006.

[56] L. Holm, S. Kaariainen, P. Rosenstrom, and A. Schenkel. Searching protein structure databases with dalilite v.3. *Bioinformatics*, 24(23):2780–1, 2008.

[57] L. Holm and C. Sander. Parser for protein folding units. *Proteins*, 19(3):256–68, 1994.

[58] L. Holm and C. Sander. Dali/fssp classification of three-dimensional protein folds. *Nucleic Acids Res*, 25(1):231–4, 1997.

[59] S. W. Huang and J. K. Hwang. Computation of conformational entropy from protein sequences using the machine-learning method–application to the study of the relationship between structural conservation and local structural stability. *Proteins*, 59(4):802–9, 2005.

[60] L. S. Itzhaki, D. E. Otzen, and A. R. Fersht. The structure of the transition state for folding of chymotrypsin inhibitor 2 analysed by protein engineering

methods: evidence for a nucleation-condensation mechanism for protein folding. *J Mol Biol*, 254(2):260–88, 1995.

[61] M. D. Jacobs and R. O. Fox. Staphylococcal nuclease folding intermediate characterized by hydrogen exchange and nmr spectroscopy. *Proc Natl Acad Sci U S A*, 91(2):449–53, 1994.

[62] L. Jaroszewski, L. Rychlewski, Z. Li, W. Li, and A. Godzik. Ffas03: a server for profile–profile sequence alignments. *Nucleic Acids Res*, 33(Web Server issue):W284–8, 2005.

[63] D. T. Jones. Protein secondary structure prediction based on position-specific scoring matrices. *J Mol Biol*, 292(2):195–202, 1999.

[64] D. T. Jones. Predicting novel protein folds by using fragfold. *Proteins*, Suppl 5:127–32, 2001.

[65] D. T. Jones, W. R. Taylor, and J. M. Thornton. A new approach to protein fold recognition. *Nature*, 358(6381):86–9, 1992.

[66] K. Karplus, C. Barrett, M. Cline, M. Diekhans, L. Grate, and R. Hughey. Predicting protein structure using only sequence information. *Proteins*, Suppl 3:121–5, 1999.

[67] H. Kato, H. Feng, and Y. Bai. The folding pathway of t4 lysozyme: the high-resolution structure and folding of a hidden intermediate. *J Mol Biol*, 365(3):870–80, 2007.

[68] H. Kaya and H. S. Chan. Explicit-chain model of native-state hydrogen exchange: implications for event ordering and cooperativity in protein folding. *Proteins*, 58(1):31–44, 2005.

[78] M. A. Knauf, F. Lohr, G. P. Curley, P. O'Farrell, S. G. Mayhew, F. Muller, and H. Ruterjans. Homonuclear and heteronuclear nmr studies of oxidized desulfovibrio vulgaris flavodoxin. sequential assignments and identification of secondary structure elements. *Eur J Biochem*, 213(1):167–84, 1993.

[79] A. Kolinski and J. M. Bujnicki. Generalized protein structure prediction based on combination of fold-recognition with de novo folding and evaluation of models. *Proteins*, 61 Suppl 7:84–90, 2005.

[80] A. Kolinski and J. Skolnick. Assembly of protein structure from sparse experimental data: an efficient monte carlo model. *Proteins*, 32(4):475–94, 1998.

[81] A. Kryshtafovych, O. Krysko, P. Daniluk, Z. Dmytriv, and K. Fidelis. Protein structure prediction center in casp8. *Proteins*, 77 Suppl 9:5–9, 2009.

[82] J. Kyte and R. F. Doolittle. A simple method for displaying the hydropathic character of a protein. *J Mol Biol*, 157(1):105–132, 1982.

[83] E. Lacroix, M. Bruix, E. Lopez-Hernandez, L. Serrano, and M. Rico. Amide hydrogen exchange and internal dynamics in the chemotactic protein chey from escherichia coli. *J Mol Biol*, 271(3):472–87, 1997.

[84] Y. Levy, S. S. Cho, J. N. Onuchic, and P. G. Wolynes. A survey of flexible protein binding mechanisms and their transition states using native topology based energy landscapes. *J Mol Biol*, 346(4):1121–45, 2005.

[85] R. Li and C. Woodward. The hydrogen exchange core and protein folding. *Protein Sci*, 8(8):1571–90, 1999.

[86] J. Liu and B. Rost. Chop proteins into structural domain-like fragments. *Proteins*, 55(3):678–88, 2004.

[87] J. Liu and B. Rost. Sequence-based prediction of protein domains. *Nucleic Acids Res*, 32(12):3522–30, 2004.

[88] A. Liwo, S. Oldziej, M. R. Pincus, R. J. Wawak, S. Rackovsky, and H. A. Scheraga. A united-residue force field for off-lattice protein-structure simulations. i. functional forms and parameters of long-range side-chain interaction potentials from protein crystal data. *J. Comp. Chem.*, 18(7):849–873, 1997.

[89] E. Lopez-Hernandez and L. Serrano. Structure of the transition state for folding of the 129 aa protein chey resembles that of a smaller protein, ci-2. *Fold Des*, 1(1):43–55, 1996.

[90] M. Lu, A. D. Dousis, and J. Ma. Opus-psp: an orientation-dependent statistical all-atom potential derived from side-chain packing. *J Mol Biol*, 376(1):288–301, 2008.

[91] M. Lu, A. D. Dousis, and J. Ma. Opus-rota: a fast and accurate method for side-chain modeling. *Protein Sci*, 17(9):1576–85, 2008.

[92] M. Lu, B. Poon, and J. Ma. A new method for coarse-grained elastic normal-mode analysis. *J. Chem. Theor. Comp*, 2:464–471, 2006.

[93] A. D. MacKerell, D. Bashford Jr., M. Bellott, R. L. Dunbrack Jr., J. D. Evanseck, M. J. Field, S. Fischer, J. Gao, H. Guo, S. Ha, D. Joseph-McCarthy, L. Kuchnir, K. Kuczera, F. T. K. Lau, C. Mattos, S. Michnick, T. Ngo, D. T. Nguyen, B. Prodhom, W. E. Reiher, III, B. Roux, M. Schlenkrich, J. C. Smith, R. Stote, J. Straub, M. Watanabe, J. Wiorkiewicz-Kuczera, D. Yin, and M. Karplus. All-atom empirical potential for molecular modeling and dynamics studies of proteins. *J. Phys. Chem.*, B102:3586–3616, 1998.

[94] T. Madej, J. F. Gibrat, and S. H. Bryant. Threading a database of protein cores. *Proteins*, 23(3):356–69, 1995.

[95] D. E. Makarov and K. W. Plaxco. The topomer search model: A simple, quantitative theory of two-state protein folding kinetics. *Protein Sci*, 12(1):17–26, 2003.

[69] L. A. Kelley, R. M. MacCallum, and M. J. Sternberg. Enhanced genome annotation using structural profiles in the program 3d-pssm. *J Mol Biol*, 299(2):499–520, 2000.

[70] G. Kieseritzky, G. Morra, and E. W. Knapp. Stability and fluctuations of amide hydrogen bonds in a bacterial cytochrome c: a molecular dynamics study. *J Biol Inorg Chem*, 11(1):26–40, 2006.

[71] D. Kihara, H. Lu, A. Kolinski, and J. Skolnick. Touchstone: an ab initio protein structure prediction method that uses threading-based tertiary restraints. *Proc Natl Acad Sci U S A*, 98(18):10125–10130, 2001.

[72] D. Kihara and J. Skolnick. The pdb is a covering set of small protein structures. *J Mol Biol*, 334(4):793–802, 2003.

[73] D. E. Kim, D. Chivian, and D. Baker. Protein structure prediction and analysis using the robetta server. *Nucleic Acids Res*, 32(Web Server issue):W526–31, 2004.

[74] D. E. Kim, D. Chivian, L. Malmstrom, and D. Baker. Automated prediction of domain boundaries in casp6 targets using ginzu and rosettadom. *Proteins*, 61 Suppl 7:193–200, 2005.

[75] K. S. Kim, J. A. Fuchs, and C. K. Woodward. Hydrogen exchange identifies native-state motional domains important in protein folding. *Biochemistry*, 32(37):9600–8, 1993.

[76] S. Kirkpatrick, Jr. Gelatt, C. D., and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–80, 1983.

[77] A. Kjellsson, I. Sethson, and B. H. Jonsson. Hydrogen exchange in a large 29 kd protein and characterization of molten globule aggregation by nmr. *Biochemistry*, 42(2):363–74, 2003.

[96] E. R. Mardis. Anticipating the 1,000 dollar genome. *Genome Biol*, 7(7):112, 2006.

[97] R. L. Marsden, L. J. McGuffin, and D. T. Jones. Rapid protein domain assignment from amino acid sequence using predicted secondary structure. *Protein Sci*, 11(12):2814–24, 2002.

[98] A. Matouschek, Jr. Kellis, J. T., L. Serrano, and A. R. Fersht. Mapping the transition state and pathway of protein folding by protein engineering. *Nature*, 340(6229):122–6, 1989.

[99] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *J. Chem. Phys.*, 21:1087–1092, 1953.

[100] N. Metropolis and S. Ulam. The monte carlo method. *J Am Stat Assoc*, 44(247):335–41, 1949.

[101] S. Miyazaki, Y. Kuroda, and S. Yokoyama. Characterization and prediction of linker sequences of multi-domain proteins by a neural network. *J Struct Funct Genomics*, 2(1):37–51, 2002.

[102] F. A. Momany, R. F. McGuire, A. W. Burgess, and Harold A. Scheraga. Energy parameters in polypeptides. vii. geometric parameters, partial atomic charges, nonbonded interactions, hydrogen bond interactions, and intrinsic torsional potentials for the naturally occurring amino acids. *The Journal of Physical Chemistry*, 79(22):2361–2381, 1975.

[103] J. Moult, J. T. Pedersen, R. Judson, and K. Fidelis. A large-scale experiment to assess protein structure prediction methods. *Proteins*, 23(3):ii–v, 1995.

[104] A. G. Murzin, S. E. Brenner, T. Hubbard, and C. Chothia. Scop: a struc-

tural classification of proteins database for the investigation of sequences and structures. *J. Mol. Biol.*, 247:536–540, 1995.

[105] S. Oldziej, C. Czaplewski, A. Liwo, M. Chinchio, M. Nanias, J. A. Vila, M. Khalili, Y. A. Arnautova, A. Jagielska, M. Makowski, H. D. Schafroth, R. Kazmierkiewicz, D. R. Ripoll, J. Pillardy, J. A. Saunders, Y. K. Kang, K. D. Gibson, and H. A. Scheraga. Physics-based protein-structure prediction using a hierarchical protocol based on the unres force field: assessment in two blind tests. *Proc Natl Acad Sci U S A*, 102(21):7547–52, 2005.

[106] J. N. Onuchic and P. G. Wolynes. Theory of protein folding. *Curr Opin Struct Biol*, 14(1):70–5, 2004.

[107] J. N. Onuchic, Luthey-Schulten Z., and P. G. Wolynes. Theory of protein folding: the energy landscape perspective. *Annu Rev Phys Chem*, 48:545–600, 1997.

[108] 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):1093–1109, 1997.

[109] A. R. Panchenko, A. Marchler-Bauer, and S. H. Bryant. Combination of threading potentials and sequence profiles improves fold recognition. *J Mol Biol*, 296(5):1319–31, 2000.

[110] A. Pikaz and I Dinstein. An algorithm for polygonal approximation based on iterative point elimination. *Pattern Recognition Letters*, 16:557–563, 1995.

[111] K. W. Plaxco, K. T. Simons, and D. Baker. Contact order, transition state placement and the refolding rates of single domain proteins. *J Mol Biol*, 277(4):985–94, 1998.

[112] K. W. Plaxco, K. T. Simons, I. Ruczinski, and D. Baker. Topology, stability, sequence, and length: defining the determinants of two-state protein folding kinetics. *Biochemistry*, 39(37):11177–83, 2000.

[113] D. Przybylski and B. Rost. Improving fold recognition without folds. *J Mol Biol*, 341(1):255–69, 2004.

[114] A. J. Rader and Ivet Bahar. Folding core predictions from network models of proteins. *Polymer*, 45(2):659–668, 2004.

[115] I. Radhakrishnan, G. C. Perez-Alvarado, D. Parker, H. J. Dyson, M. R. Montminy, and P. E. Wright. Structural analyses of creb-cbp transcriptional activator-coactivator complexes by nmr spectroscopy: implications for mapping the boundaries of structural domains. *J Mol Biol*, 287(5):859–65, 1999.

[116] T. M. Raschke and S. Marqusee. The kinetic folding intermediate of ribonuclease h resembles the acid molten globule and partially unfolded molecules detected under native conditions. *Nat Struct Biol*, 4(4):298–304, 1997.

[117] C. A. Rohl, C. E. Strauss, K. M. Misura, and D. Baker. Protein structure prediction using rosetta. *Methods Enzymol*, 383:66–93, 2004.

[118] B. S. Russell, L. Zhong, M. G. Bigotti, F. Cutruzzola, and K. L. Bren. Backbone dynamics and hydrogen exchange of pseudomonas aeruginosa ferricytochrome c(551). *J Biol Inorg Chem*, 8(1-2):156–66, 2003.

[119] K. Saeki, M. Arai, T. Yoda, M. Nakao, and K. Kuwajima. Localized nature of the transition-state structure in goat alpha-lactalbumin folding. *J Mol Biol*, 341(2):589–604, 2004.

[120] H. K. Saini and D. Fischer. Meta-dp: domain prediction meta-server. *Bioinformatics*, 21(12):2917–20, 2005.

[121] A. Sali, E. Shakhnovich, and M. Karplus. How does a protein fold? *Nature*, 369(6477):248–51, 1994.

[122] X. Salvatella, C. M. Dobson, A. R. Fersht, and M. Vendruscolo. Determination of the folding transition states of barnase by using phii-value-restrained simulations validated by double mutant phiij-values. *Proc Natl Acad Sci U S A*, 102(35):12389–94, 2005.

[123] R. Sanchez and A. Sali. Evaluation of comparative protein structure modeling by modeller-3. *Proteins*, Suppl 1:50–8, 1997.

[124] E. D. Scheeff and P. E. Bourne. Application of protein structure alignments to iterated hidden markov model protocols for structure prediction. *BMC Bioinformatics*, 7:410, 2006.

[125] Walter R. P. Scott, Philippe H. Hunenberger, Ilario G. Tironi, Alan E. Mark, Salomon R. Billeter, Jens Fennen, Andrew E. Torda, Thomas Huber, Peter Kruger, and Wilfred F. van Gunsteren. The gromos biomolecular simulation program package. *The Journal of Physical Chemistry A*, 103(19):3596–3607, 1999.

[126] M. Y. Shen and A. Sali. Statistical potential for assessment and prediction of protein structures. *Protein Sci*, 15(11):2507–24, 2006.

[127] S. Shi, J. Pei, R. I. Sadreyev, L. N. Kinch, I. Majumdar, J. Tong, H. Cheng, B. H. Kim, and N. V. Grishin. Analysis of casp8 targets, predictions and assessment methods. *Database (Oxford)*, 2009:bap003, 2009.

[128] A. Shmygelska. Search for folding nuclei in native protein structures. *Bioinformatics*, 21 Suppl 1:i394–402, 2005.

[129] B. A. Shoemaker, J. Wang, and P. G. Wolynes. Structural correlations in protein folding funnels. *Proc Natl Acad Sci U S A*, 94(3):777–82, 1997.

[130] A. R. Sikder and A. Y. Zomaya. Improving the performance of domaindiscovery of protein domain boundary assignment using inter-domain linker index. *BMC Bioinformatics*, 7 Suppl 5:S6, 2006.

[131] K. T. Simons, C. Kooperberg, E. Huang, and D. Baker. Assembly of protein tertiary structures from fragments with similar local sequences using simulated annealing and bayesian scoring functions. *J Mol Biol*, 268(1):209–25, 1997.

[132] M. J. Sippl. Calculation of conformational ensembles from potentials of mean force. an approach to the knowledge-based prediction of local structures in globular proteins. *J Mol Biol*, 213(4):859–83, 1990.

[133] J. Skolnick, A. K. Arakaki, S. Y. Lee, and M. Brylinski. The continuity of protein structure space is an intrinsic property of proteins. *Proc Natl Acad Sci U S A*, 106(37):15690–5, 2009.

[134] J. Skolnick and D. Kihara. Defrosting the frozen approximation: Prospector–a new approach to threading. *Proteins*, 42(3):319–31, 2001.

[135] J. Skolnick, D. Kihara, and Y. Zhang. Development and large scale benchmark testing of the prospector3 threading algorithm. *Proteins*, 56(3):502–18, 2004.

[136] J. Skolnick, Y. Zhang, A. K. Arakaki, A. Kolinski, M. Boniecki, A. Szilagyi, and D. Kihara. Touchstone: a unified approach to protein structure prediction. *Proteins*, 53 Suppl 6:469–79, 2003.

[137] T. F. Smith and M. S. Waterman. Identification of common molecular subsequences. *J Mol Biol*, 147(1):195–7, 1981.

[138] E. L. Sonnhammer, S. R. Eddy, and R. Durbin. Pfam: a comprehensive database of protein domain families based on seed alignments. *Proteins*, 28(3):405–20, 1997.

[139] M. Suyama and O. Ohara. Domcut: prediction of inter-domain linker regions in amino acid sequences. *Bioinformatics*, 19(5):673–4, 2003.

[140] C. H. Tai, W. J. Lee, J. J. Vincent, and B. Lee. Evaluation of domain prediction in casp6. *Proteins*, 61 Suppl 7:183–92, 2005.

[141] W. R. Taylor. Protein structural domain identification. *Protein Eng*, 12(3):203–16, 1999.

[142] S. Thomas, G. Song, and N. M. Amato. Protein folding by motion planning. *Phys Biol*, 2(4):S148–55, 2005.

[143] S. Thomas, X. Tang, L. Tapia, and N. M. Amato. Simulating protein motions with rigidity analysis. *J Comput Biol*, 14(6):839–55, 2007.

[144] M. Tress, J. Cheng, P. Baldi, K. Joo, J. Lee, J. H. Seo, J. Lee, D. Baker, D. Chivian, D. Kim, and I. Ezkurdia. Assessment of predictions submitted for the casp7 domain prediction category. *Proteins*, 69 Suppl 8:137–51, 2007.

[145] M. Tress, C. H. Tai, G. Wang, I. Ezkurdia, G. Lopez, A. Valencia, B. Lee, and Jr. Dunbrack, R. L. Domain definition and target classification for casp6. *Proteins*, 61 Suppl 7:8–18, 2005.

[146] M. L. Tress, I. Ezkurdia, and J. S. Richardson. Target domain definition and classification in casp8. *Proteins*, 77 Suppl 9:10–7, 2009.

[147] S. Veretnik, P. E. Bourne, N. N. Alexandrov, and I. N. Shindyalov. Toward consistent assignment of structural domains in proteins. *J Mol Biol*, 339(3):647–78, 2004.

[148] A. Vinayagam, J. Shi, G. Pugalenthi, B. Meenakshi, T. L. Blundell, and R. Sowdhamini. Ddbase2.0: updated domain database with improved identification of structural domains. *Bioinformatics*, 19(14):1760–4, 2003.

[149] I. Walsh, A. J. Martin, C. Mooney, E. Rubagotti, A. Vullo, and G. Pollastri. Ab initio and homology based prediction of protein domains by recursive neural networks. *BMC Bioinformatics*, 10:195, 2009.

[150] G. Wang and Jr. Dunbrack, R. L. Pisces: a protein sequence culling server. *Bioinformatics*, 19(12):1589–91, 2003.

[151] G. Wang and Jr. Dunbrack, R. L. Pisces: recent improvements to a pdb sequence culling server. *Nucleic Acids Res*, 33(Web Server issue):W94–8, 2005.

[152] Y. Wang, J. B. Anderson, J. Chen, L. Y. Geer, S. He, D. I. Hurwitz, C. A. Liebert, T. Madej, G. H. Marchler, A. Marchler-Bauer, A. R. Panchenko, B. A. Shoemaker, J. S. Song, P. A. Thiessen, R. A. Yamashita, and S. H. Bryant. Mmdb: Entrez's 3d-structure database. *Nucleic Acids Res*, 30(1):249–52, 2002.

[153] Z. Wang, J. Eickholt, and J. Cheng. Multicom: a multi-level combination approach to protein structure prediction and its assessments in casp8. *Bioinformatics*, 26(7):882–8, 2010.

[154] T. R. Weikl and K. A. Dill. Folding rates and low-entropy-loss routes of two-state proteins. *J Mol Biol*, 329(3):585–98, 2003.

[155] H. M. Went and S. E. Jackson. Ubiquitin folds through a highly polarized transition state. *Protein Eng Des Sel*, 18(5):229–37, 2005.

[156] D. B. Wetlaufer. Nucleation, rapid folding, and globular intrachain regions in proteins. *Proc Natl Acad Sci U S A*, 70(3):697–701, 1973.

[157] C. Woodward. Is the slow exchange core the protein folding core? *Trends Biochem Sci*, 18(10):359–60, 1993.

[158] S. Wu, J. Skolnick, and Y. Zhang. Ab initio modeling of small proteins by iterative tasser simulations. *BMC Biol*, 5:17, 2007.

[159] Y. Wu, M. Chen, M. Lu, Q. Wang, and J. Ma. Determining protein topology from skeletons of secondary structures. *J. Mol. Biol.*, 350:571–86, 2005.

[160] Y. Wu, A. D. Dousis, M. Chen, J. Li, and J. Ma. Opus-dom: applying the folding-based method vecfold to determine protein domain boundaries. *J Mol Biol*, 385(4):1314–29, 2009.

[161] Y. Wu, M. Lu, M. Chen, J. Li, and J. Ma. Opus-ca: a knowledge-based potential function requiring only calpha positions. *Protein Sci*, 16(7):1449–63, 2007.

[162] Y. Wu, X. Tian, M. Lu, M. Chen, Q. Wang, and J. Ma. Folding of small helical proteins assisted by small-angle x-ray scattering profiles. *Structure (Camb)*, 13(11):1587–97, 2005.

[163] Y. Xia, E. S. Huang, M. Levitt, and R. Samudrala. Ab initio construction of protein tertiary structures using a hierarchical approach. *J Mol Biol*, 300(1):171–85, 2000.

[164] Y. Xu, D. Xu, and H. N. Gabow. Protein domain decomposition using a graph-theoretic approach. *Bioinformatics*, 16(12):1091–104, 2000.

[165] S. Yan, S. D. Kennedy, and S. Koide. Thermodynamic and kinetic exploration of the energy landscape of borrelia burgdorferi ospa by native-state hydrogen exchange. *J Mol Biol*, 323(2):363–75, 2002.

[166] P. D. Yoo, A. R. Sikder, J. Taheri, B. B. Zhou, and A. Y. Zomaya. Domnet: protein domain boundary prediction using enhanced general regression network and new profiles. *IEEE Trans Nanobioscience*, 7(2):172–81, 2008.

[167] M. J. Zaki, V. Nadimpally, D. Bardhan, and C. Bystroff. Predicting protein folding pathways. *Bioinformatics*, 20 Suppl 1:i386–93, 2004.

[168] J. Zhang, Q. Wang, B. Barz, Z. He, I. Kosztin, Y. Shang, and D. Xu. Mufold: A new solution for protein 3d structure prediction. *Proteins*, 78(5):1137–52, 2010.

[169] Y. Zhang, D. Kihara, and J. Skolnick. Local energy landscape flattening: parallel hyperbolic monte carlo sampling of protein folding. *Proteins*, 48(2):192–201, 2002.

[170] Y. Zhang, A. Kolinski, and J. Skolnick. Touchstone ii: a new approach to ab initio protein structure prediction. *Biophys J*, 85(2):1145–64, 2003.

[171] Y. Zhang and J. Skolnick. Scoring function for automated assessment of protein structure template quality. *Proteins*, 57(4):702–10, 2004.

[172] Y. Zhang and J. Skolnick. The protein structure prediction problem could be solved using the current pdb library. *Proc Natl Acad Sci U S A*, 102(4):1029–34, 2005.

[173] W. Zheng and S. Doniach. Protein structure prediction constrained by solution x-ray scattering data and structural homology identification. *J Mol Biol*, 316(1):173–187, 2002.

[174] W. Zheng and S. Doniach. Fold recognition aided by constraints from small angle x-ray scattering data. *Protein Eng Des Sel*, 18(5):209–19, 2005.

[175] H. Zhou, S. B. Pandit, S. Y. Lee, J. Borreguero, H. Chen, L. Wroblewska, and J. Skolnick. Analysis of tasser-based casp7 protein structure prediction results. *Proteins*, 69 Suppl 8:90–7, 2007.

[176] H. Zhou and J. Skolnick. Protein structure prediction by pro-sp3-tasser. *Biophys J*, 96(6):2119–27, 2009.

[177] H. Zhou, B. Xue, and Y. Zhou. Ddomain: Dividing structures into domains using a normalized domain-domain interaction profile. *Protein Sci*, 16(5):947–55, 2007.

[178] Hongyi Zhou and Yaoqi Zhou. Fold recognition by combining sequence profiles derived from evolution and from depth-dependent structural alignment of fragments. *Proteins: Structure, Function, and Bioinformatics*, 58(2):321–328, 2005.

Appendix A

Glossary

BLAST Basic Local Alignment Search Tool

C$^{\alpha}$ the alpha carbon on the protein backbone or main-chain

CABS C$^{\alpha}$, C$^{\beta}$, and side-group

CASP Critical Assessment of Techniques for Protein Structure Prediction

CATH Class, Architecture, Topology, Homologous superfamily

DBPL domain boundary profile library

DLI domain linker index

FIRST Floppy Inclusions and Rigid Substructure Topography

FN false negative

FP false positive

GM Galzitskaya & Melnik

GNM Gaussian Network Model

HX hydrogen-deuterium exchange

LJ Lennard-Jones

MMDB Molecular Modeling Database

NDO normalized domain overlap

NH hydrogen-bonded amide proton

NMR nuclear magnetic resonance

REI residue entropy index

RMSD root mean square distance

PDB Protein Data Bank

PDLI Profile Domain Linker propensity Index

PDP Protein Domain Parser

PSSM position-specific substitution matrix

SAS solvent accessible surface

SAXS small-angle X-ray scattering

SCOP structural classification of proteins

SICHO side chain only

SSE secondary structure element

SSSM super-secondary structure motif

TP true positive

WVP windowed variance profile

Appendix B

Additional background

B.1 Techniques for protein structure prediction

B.1.1 Fold recognition

Fold recognition is dominated by three general techniques: (1) advanced sequence comparison; (2) profile-profile comparison; and (3) threading. In addition, modern fold recognition methods are increasingly incorporating a blend of elements of the above techniques.

Advanced sequence comparison (sequence-profile, profile-sequence) Advanced sequence comparison methods are very similar to homology modeling methods, but instead of making direct sequence-sequence comparisons, they look for a profile of similar sequences within a database of sequence-structure families. Such methods include hidden Markov model methods [66] and position-specific iterated BLAST (PSI-BLAST) searches [2, 42].

Profile-profile comparison These fold recognition methods align the target sequence and the template sequences by profiles like those developed for advanced sequence comparison. This may involve the use of position-specific substitution matrices or some other one-dimensional descriptor mapping. For example, the method 3D-PSSM/PHYRE [69] uses a structural classification of proteins (SCOP) database, while AGAPE [113] generates secondary structure and solvent accessibility descriptors of a target and compares these to descriptors of known secondary structures.

Threading (sequence-structure) In threading, "sequences are fitted directly onto the backbone coordinates of known protein structures" [65]. A library of known protein folds or motifs is derived from the Protein Data Bank or another database of structures, and the test sequence is threaded through each motif and scored for goodness of fit, often employing the same potential functions used in *ab initio* methods. These methods are computationally expensive in general, but can be simplified considerably by ignoring certain interactions in the scoring process [42]. Among the most successful fold recognition algorithms are the threading methods of Skolnick *et al* [136, 135]. Their latest approach [135] applies three different scoring functions to triangulate the most accurate regions of structure prediction.

B.1.2 Alignment and optimization methods

A large variety of optimization methods have been applied to fold recognition, including simulated annealing, branch and bound searching, genetic algorithms, and neural networks.

B.2 Mathematical conventions and definitions

This section describes several of the notations that will be used in this thesis.

B.2.1 Definitions

Matrices

Matrices will use boldfaced capital letters: $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}, \mathbf{E}$, etc. A matrix $\mathbf{A}$ will be defined as $\mathbf{A} \in \mathbb{A}^{M \times N}$, where $\mathbb{A}$ is a set of numbers, such as real numbers ($\mathbb{R}$), complex numbers ($\mathbb{C}$), or integers ($\mathbb{Z}$). The size of matrix $\mathbf{A}$ is denoted by the exponent of $\mathbb{A}$, such that $\mathbb{A}^{M \times N}$ is the set of matrices with M rows and N columns. The element of matrix $\mathbf{A}$ in the mth row and nth column will be enoted as $a_{m,n}$.

Transpose For all matrices (and vectors), a superscripted "T" will be used to denote *transpose.* For a matrix $\mathbf{A} \in \mathbb{R}^{M \times N}$, the transpose of $\mathbf{A}$ is defined as $\mathbf{A}^{\mathrm{T}} \in \mathbb{C}^{N \times M}$ with elements $a^{\mathrm{T}}_{m,n} = a_{n,m}$.

Vectors

Vectors are a subset of matrices and assume all of the same properties. Real vectors are denoted using boldfaced lowercase letters: $\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}, \mathbf{e}$, etc. For this thesis, it is implied that vectors are column vectors, unless specified otherwise. We define an N-length column vector $\mathbf{a} = (a_0, a_1, a_2, ..., a_{N-1})^{\mathrm{T}} \in \mathbb{A}^N$, where a_n is the nth element of $\mathbf{a}$ and "T" denotes transpose. An N-length row vector is defined as $\mathbf{b} = (b_0, b_1, b_2, ..., b_{N-1}) \in \mathbb{A}^{1 \times N}$.

Orthogonality For two vectors $\mathbf{a}, \mathbf{b} \in \mathbb{C}^N$, $\mathbf{a}$ is orthogonal to $\mathbf{b}$ if the inner product of the two vectors results in the zero vector

$$\mathbf{x} \perp \mathbf{y} \Leftrightarrow \mathbf{x}^{\mathrm{T}}\mathbf{y} = \mathbf{0}$$

where "$\perp$" denotes *orthogonality.*

The delta function The delta function, $\boldsymbol{\delta}$, is defined as

$$\delta_n = \begin{cases} 1 & n = 0 \\ 0 & n \neq 0 \end{cases}$$

Vector operations

For the following operations, all vectors take on the form of an N-length column vector $\mathbf{a} = (a_0, a_1, a_2, ..., a_{N-1})^{\mathrm{T}}$.

Element-by-element multiplication The "$\tilde{\times}$" operator will denote array, or element-by-element, multiplication. Given two vectors $\mathbf{a}, \mathbf{b} \in \mathbb{C}^N$,

$$\mathbf{a} \tilde{\times} \mathbf{b} = (a_0 b_0, a_1 b_1, a_2 b_2, ..., a_{N-1} b_{N-1})^{\mathrm{T}}$$

Convolution sum The "$*$" operator will denote convolution sum. Given $\mathbf{x} \in \mathbb{C}^N$, $\mathbf{h} \in \mathbb{C}^L$, and $\mathbf{y} = \mathbf{h} * \mathbf{x}$, then the elements of $\mathbf{y} \in \mathbb{C}^{N+2L-1}$ are

$$y_n = h_n * x_n = \sum_{k=0}^{N-1} x_k h_{n-k}$$

We can also describe convolution by using the Toeplitz matrix $\mathbf{H}$ of $\mathbf{h}$, where $\mathbf{H} \in \mathbb{C}^{(N+2L-1)\times N}$

$$\mathbf{H} = \begin{bmatrix} h_0 & 0 & 0 & \cdots & 0 \\ h_1 & h_0 & 0 & \cdots & 0 \\ h_2 & h_1 & h_0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ h_{N-1} & h_{N-2} & h_{N-3} & \cdots & h_0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & h_{N-1} \end{bmatrix}$$

Thus, $\mathbf{y} = \mathbf{H}\mathbf{x}$.

Vector absolute value The absolute value of a vector $\mathbf{x} \in \mathbb{R}^N$ is defined as

$$|\mathbf{x}| = (|x_0|, |x_1|, |x_2|, \cdots, |x_{N-1}|)^{\mathrm{T}}$$

Vector norms The two vector norms used in this thesis, the L-2 and L-∞ norms, are defined for all $\mathbf{x} \in \mathbb{R}^N$ as

$$L\text{-}2: \qquad \|\mathbf{x}\|_2 = (\mathbf{x}^{\mathrm{T}}\mathbf{x})^{1/2} = \left[\sum_{n=0}^{N-1} |x_n|^2\right]^{1/2}$$

$$L\text{-}\infty: \qquad \|\mathbf{x}\|_\infty = \max_n |x_n|$$

Special definitions and functions

The signum function The signum function $sgn(x)$ is defined as

$$\mathrm{sgn}(x) \equiv \begin{cases} 1 & x \geq 0 \\ -1 & x < 0 \end{cases}$$

Root mean square distance (RMSD) Root mean square distance (RMSD) between two vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^N$ is defined as

$$\text{RMSD} = \left[\frac{1}{N} \sum_{n=1}^{N} (x_n - y_n)^2 \right]^{1/2}$$

B.2.2 Model geometry

Backbone internal coordinate system

The backbone Cartesian coordinates are converted to internal coordinates by the following procedure used in Lu *et al* [92]:

$$\mathbf{e}_i = \cos\theta_i \frac{\mathbf{u}_i}{\|\mathbf{u}_i\|_2} + \sin\theta_i \left(\cos\phi_i \frac{\mathbf{v}_i}{\|\mathbf{v}_i\|_2} + \sin\phi_i \frac{\mathbf{w}_i}{\|\mathbf{w}_i\|_2} \right)$$
$$\mathbf{r}_i = \mathbf{r}_{i-1} + \ell_i \mathbf{e}_i$$

where

$$\mathbf{u}_i = \mathbf{e}_{i-1}, \qquad \mathbf{v}_i = \mathbf{w}_i \times \mathbf{u}_i, \qquad \mathbf{w}_i = \mathbf{u}_{i-1} \times \mathbf{u}_i$$

given some initial $\mathbf{r}_0$, $\mathbf{u}_0$, $\mathbf{v}_0$, $\mathbf{w}_0$. The reverse conversion (i.e., internal to Cartesian coordinates) is accomplished by:

$$\ell_i = \|\mathbf{r}_i - \mathbf{r}_{i-1}\|_2, \qquad \mathbf{e}_i = \frac{\mathbf{r}_i - \mathbf{r}_{i-1}}{\ell_i}$$
$$\theta_i = \cos^{-1}(\mathbf{u}_i \cdot \mathbf{e}_i)$$
$$\phi_i = \text{sign}(\mathbf{w}_i \cdot \mathbf{e}_i) \cdot \cos^{-1}\left(\frac{\mathbf{w}_i}{\|\mathbf{w}_i\|_2} \cdot \frac{\mathbf{w}_{i+1}}{\|\mathbf{w}_{i+1}\|_2} \right)$$

where $\text{sign}(\cdot)$ is the signum function

$$\text{sign}(x) = \begin{cases} 1, & \text{if } x \geq 0 \\ -1, & \text{if } x < 0 \end{cases}$$

Side chain internal coordinate system

The side chain Cartesian coordinates are converted to internal coordinates by the following formula:

$$\mathbf{e}_i^{sc} = \ell_i^{sc}\left[\sin\theta_i^{sc}\frac{\mathbf{u}_i^{sc}}{\|\mathbf{u}_i^{sc}\|_2} + \cos\theta_i^{sc}\left(\cos\phi_i^{sc}\frac{\mathbf{v}_i^{sc}}{\|\mathbf{v}_i^{sc}\|_2} + \sin\phi_i^{sc}\frac{\mathbf{w}_i^{sc}}{\|\mathbf{w}_i^{sc}\|_2}\right)\right]$$

$$\mathbf{r}_i^{sc} = \mathbf{r}_i + \mathbf{e}_i^{sc}$$

where

$$\mathbf{u}_i^{sc} = \begin{cases} \mathbf{e}_i + \mathbf{e}_{i+1}, & \text{if } i < N \\ \mathbf{e}_N, & \text{if } i = N \end{cases}, \qquad \mathbf{w}_i^{sc} = \begin{cases} \mathbf{e}_i \times \mathbf{e}_{i+1}, & \text{if } i < N \\ \mathbf{e}_{N-1} \times \mathbf{e}_N, & \text{if } i = N \end{cases}$$

$$\mathbf{v}_i^{sc} = \mathbf{w}_i^{sc} \times \mathbf{u}_i^{sc}$$

Conversely, side chain internal coordinates are converted back to Cartesian coordinates by:

$$\ell_i^{sc} = \|\mathbf{e}_i^{sc}\|_2, \qquad \mathbf{e}_i^{sc} = \mathbf{r}_i^{sc} - \mathbf{r}_i$$

$$\theta_i^{sc} = \sin^{-1}\left(\frac{\mathbf{u}_i^{sc}}{\|\mathbf{u}_i^{sc}\|_2} \cdot \frac{\mathbf{e}_i^{sc}}{\|\mathbf{e}_i^{sc}\|_2}\right)$$

$$\phi_i^{sc} = \text{sign}\left(\mathbf{w}_i^{sc} \cdot \mathbf{e}_i^{sc}\right) \cdot \cos^{-1}\left(\frac{\mathbf{w}_i^{sc}}{\|\mathbf{w}_i^{sc}\|_2} \cdot \frac{\mathbf{u}_i^{sc} \times \mathbf{e}_i^{sc}}{\|\mathbf{u}_i^{sc} \times \mathbf{e}_i^{sc}\|_2}\right)$$

Reconstructing the positions of main-chain atoms from the C^{α} trace

$$\mathbf{v}_x = (\mathbf{r}_{i+1} - \mathbf{r}_i)/|\mathbf{r}_{i+1} - \mathbf{r}_i|$$

$$\mathbf{v}_b = (\mathbf{r}_i - \mathbf{r}_{i-1})/|\mathbf{r}_i - \mathbf{r}_{i-1}|$$

$$\mathbf{v}_y = \mathbf{v}_x \times \mathbf{v}_b$$

$$\mathbf{v}_z = \mathbf{v}_x \times \mathbf{v}_y$$

$$\mathbf{v}'_x = (\mathbf{r}_{i+1} - \mathbf{r}_i)/|\mathbf{r}_{i+1} - \mathbf{r}_i|$$
$$\mathbf{v}'_b = (\mathbf{r}_i - \mathbf{r}_{i-1})/|\mathbf{r}_i - \mathbf{r}_{i-1}|$$
$$\mathbf{v}'_y = \mathbf{v}'_x \times \mathbf{v}'_b$$
$$\mathbf{v}'_z = \mathbf{v}'_x \times \mathbf{v}'_y$$

B.2.3 Mean square displacement

Rotation and translation operations using matrices

$$T_x = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

$$R_x = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{pmatrix}$$

$$R_z = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Homogeneous matrices

Let a 4×4 homogeneous matrix H_i be defined using a 3×3 rotation matrix R_i and a translation 3-vector t_i:

$$H_i = \begin{pmatrix} \begin{bmatrix} & & \\ & R_i & \\ & & \end{bmatrix} & \begin{bmatrix} \\ t_i \\ \\ \end{bmatrix} \\ & 1 \end{pmatrix}$$

Then

$$H_0H_1 = \begin{pmatrix} \begin{bmatrix} & R_0R_1 & \end{bmatrix} & \begin{bmatrix} R_0t_1 + t_0 \end{bmatrix} \\ & & 1 \end{pmatrix}$$

Homogeneous matrices allow the simultaneous application of rotation and translation operations, yielding the new coordinate x_1 in the upper right 3-vector given by $R_0t_1 + t_0$. It follows that

$$x_i = R^c_{i-1}t_i + x_{i-1}$$

where $R^c_i = R_0R_1 \cdots R_i$, i.e.,

$$H_0H_1 = \begin{pmatrix} \begin{bmatrix} & R^c_1 = R_0R_1 & \end{bmatrix} & \begin{bmatrix} x_1 = R_0t_1 + t_0 \end{bmatrix} \\ & & 1 \end{pmatrix}$$

Forward and reverse rotation matrices after fragment insertion

A reverse rotation matrix V is calculated starting from the end point of the chain and working backwards. This is achieved by successively multiplying the end-chain cumulative rotation matrix, i.e., $R^c_n = R_0R_1 \cdots R_n$, by transpose rotation matrices starting from the end, i.e., $V_i = R^c_nR'^T_nR'^T_{n-1} \cdots R'^T_i$, where $i < n$. Note that if $R'_i = R_i$ for all i, then $V_i = R^c_i$.

Let h and t represent the head and tail indices of the target sequence. Then let

$$F = R_hR_{h+1} \cdots R_{t-1}R_t$$
$$F' = R'_hR'_{h+1} \cdots R'_{t-1}R'_t,$$

i.e. the initial and new cumulative rotation matrices, respectively, of the insertion

region. Also, let

$$R_n^c = R_0 R_1 \cdots R_{h-1} \underline{F} R_{t+1} \cdots R_{n-1} R_n$$
$$R_n'^c = R_0 R_1 \cdots R_{h-1} \underline{F'} R_{t+1} \cdots R_{n-1} R_n$$

where an underline is placed underneath the F and F' for emphasis.

$$\begin{aligned} V_h' = R_n^c (R_n'^c)^T R_{h-1}'^c &= R_0 R_1 \cdots F \cdots R_n R_n^T \cdots F'^T \cdots R_1^T R_0^T R_{h-1}'^c \\ &= R_{h-1}^c F F'^T (R_{h-1}^c)^T R_{h-1}'^c \\ &= R_{h-1}^c F F'^T \end{aligned}$$

since $R_{h-1}^c = R_{h-1}'^c$.

To calculate the reverse-case pivot point x_h^{pr}:

$$x_h^{pr} = -R_t^c (F')^T t_f' + x_t$$

where t_f' is the end coordinate of the fragment and x_t is the target chain coordinate at the tail index t.

B.2.4 Geometric packing potential

Minimum distance between two skew lines

[This is based largely on the webpage http://www.netcomuk.co.uk/ jenolive/skew.html] Let the equations for two lines $L1$ and $L2$ be:

$$\begin{aligned} L1 : \mathbf{x}_{12} &= \mathbf{x}_1 + (\mathbf{x}_2 - \mathbf{x}_1) s_1 \\ &= \mathbf{x}_1 + \mathbf{v}_1 s_1 \\ L2 : \mathbf{x}_{34} &= \mathbf{x}_3 + (\mathbf{x}_4 - \mathbf{x}_3) s_2 \\ &= \mathbf{x}_3 + \mathbf{v}_2 s_2 \end{aligned}$$

where $\mathbf{v}_1 = \mathbf{x}_2 - \mathbf{x}_1$ and $\mathbf{v}_2 = \mathbf{x}_4 - \mathbf{x}_3$. First, we need to find a vector that will enable us to shift $L2$ such that it intersects with $L1$, forming a plane $P1$ with normal

$$\mathbf{n} = \mathbf{v}_1 \times \mathbf{v}_2$$

The perpendicular distance of plane $P1$ to the origin d_1 is given by the projection of L1 on $\mathbf{n}$, i.e.,

$$d_1 = \mathbf{x}_{12} \cdot \mathbf{n} = \mathbf{x}_1 \cdot \mathbf{n}$$

since $\mathbf{v}_1 \cdot \mathbf{n} = 0$. If we shift $L1$ to intersect with $L2$, we form plane $P2$ with the same normal $\mathbf{n}$ as before, and the distance of $P2$ to the origin is $d_2 = \mathbf{x}_3 \cdot \mathbf{n}$. The minimum distance between planes $P1$ and $P2$ is therefore

$$d = |d_1 - d_2| = |(\mathbf{x}_3 - \mathbf{x}_1) \cdot \mathbf{n}|$$

and the vector that connects $P1$ and $P2$ is $\mathbf{v}_{12} = d\hat{\mathbf{n}}$, where $\hat{\mathbf{n}} = \mathbf{n}/\|\mathbf{n}\|$. To find the points on lines $L1$ and $L2$ which are at the minimum distance d from each other, we need to shift L2 to L1 via $d\hat{\mathbf{n}}$ and solve the equation

$$\mathbf{x}_1 + \mathbf{v}_1 s_1 + d\hat{\mathbf{n}} = \mathbf{x}_3 + \mathbf{v}_2 s_2$$

After some rearranging of terms, we can rewrite this set of linear equations in matrix form

$$\begin{bmatrix} & \\ \mathbf{v}_1 & -\mathbf{v}_2 \\ & \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} = \mathbf{x}_3 - \mathbf{x}_1 - d\hat{\mathbf{n}}$$

Let

$$A = \begin{bmatrix} \mathbf{v}_1 & -\mathbf{v}_2 \end{bmatrix}$$
$$s = \begin{bmatrix} s_1 & s_2 \end{bmatrix}^T$$
$$b = \mathbf{x}_3 - \mathbf{x}_1 - d\hat{\mathbf{n}}$$

We can solve for s in $As = b$ as

$$s = (A^T A)^{-1} A^T b$$

The solution to the inverse of the 2×2 $A^T A$ is very simple.

B.2.5 Loop perturbation

Using the singular value decomposition (SVD) to find optimal alignment by RMSD

Following summarizes an excellent explanation by Kavraki [http://cnx.org/content/m11608].

Let $X, Y \in \mathbb{R}^{3\times N}$ be two sets of paired points in Cartesian coordinates, i.e. $X = \{x_1, x_2, \cdots, x_N\}$, where $x_i \in \mathbb{R}^3 \forall i$. We seek to minimize

$$E = \sum_{i=1}^{N} \|Ux_i - y_i\|_2^2 = Tr((UX - Y)^{\mathrm{T}}(UX - Y))$$

The right hand side can be expanded as

$$\begin{aligned} Tr\left((UX - Y)^{\mathrm{T}}(UX - Y)\right) &= Tr\left((UX)^{\mathrm{T}}UX - Y^{\mathrm{T}}UX - Y(UX)^{\mathrm{T}} + Y^{\mathrm{T}}Y\right) \\ &= Tr\left(X^{\mathrm{T}}U^{\mathrm{T}}UX\right) - 2Tr(Y^{\mathrm{T}}UX) + Tr(Y^{\mathrm{T}}Y) \end{aligned}$$

Since U is a rotation matrix, $U^{\mathrm{T}}U = I$ and

$$E = Tr(X^{\mathrm{T}}X) - 2Tr(Y^{\mathrm{T}}UX) + Tr(Y^{\mathrm{T}}Y)$$

Note that only the middle term is dependent on U, so E is minimized when $Tr(Y^{\mathrm{T}}UX)$ is maximized.

$$\vdots$$

This is also known as the Kabsch algorithm.

Some useful linear algebra

We wish to convert

$$X = \begin{pmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ x_{31} & x_{32} & x_{33} \end{pmatrix}$$

into

$$X' = \begin{pmatrix} -x_{11} & -x_{12} & x_{13} \\ -x_{21} & -x_{22} & x_{23} \\ x_{31} & x_{32} & x_{33} \end{pmatrix}$$

Let

$$Y = \begin{pmatrix} 1 & & \\ & 1 & \\ & & \end{pmatrix}$$

Then

$$\begin{aligned} X' &= -YXY + X(I-Y) + (I-Y)X - (I-Y)X(I-Y) \\ &= -YXY + X - XY + X - YX - (X - YX)(I-Y) \\ &= -YXY + 2X - XY - YX - (X - XY - YX + YXY) \\ &= -2YXY + X \end{aligned}$$

Protein loop closure by full cyclic coordinate descent (FCCD)

Full cyclic coordinate descent (FCCD) is a simple procedure for closing the gaps in protein loops by a series of rotations. The details are described by Boomsma & Hamelryck [14], so here we will provide only a quick summary. For the case of C-alpha trace internal coordinate geometry consisting of fixed pseudo-bond lengths and variable pseudo-bond angles θ and pseudo-dihedrals ϕ, the goal of FCCD is to compute, for a given pivot residue (chosen randomly), the rotation matrix that places the last three points of the moving loop, $m_{N-3}, m_{N-2}, m_{N-1}$, as close as possible (in the

L2-norm sense) to the corresponding three points of the fixed loop, $f_{N-3}, f_{N-2}, f_{N-1}$, while keeping the N-terminal overlaps m_1, m_2, m_3 and f_1, f_2, f_3 unchanged.

Appendix C

Additional data

C.1 VecFold2

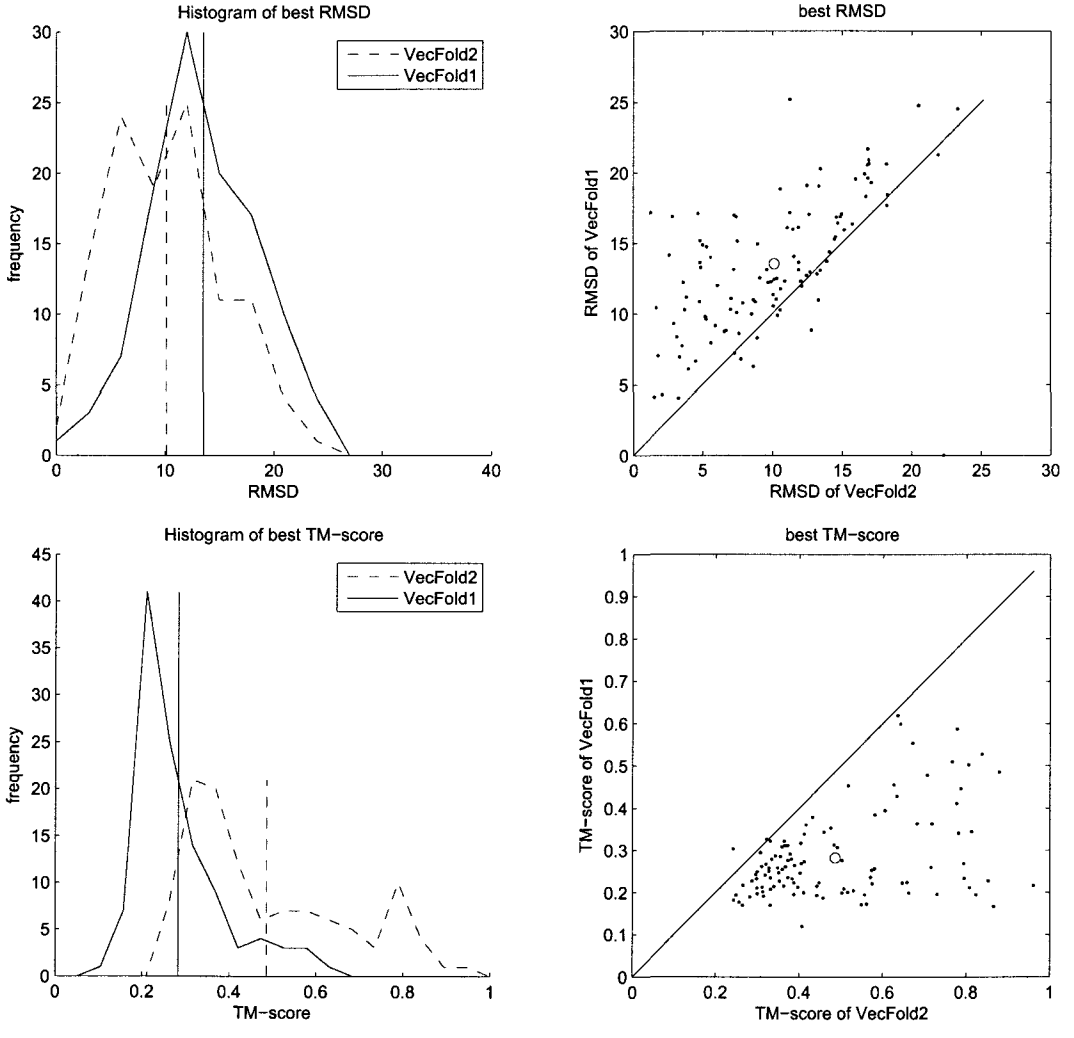

Figure C.1 : Best RMSD and TM-score, VecFold2 vs. VecFold1, CASP8

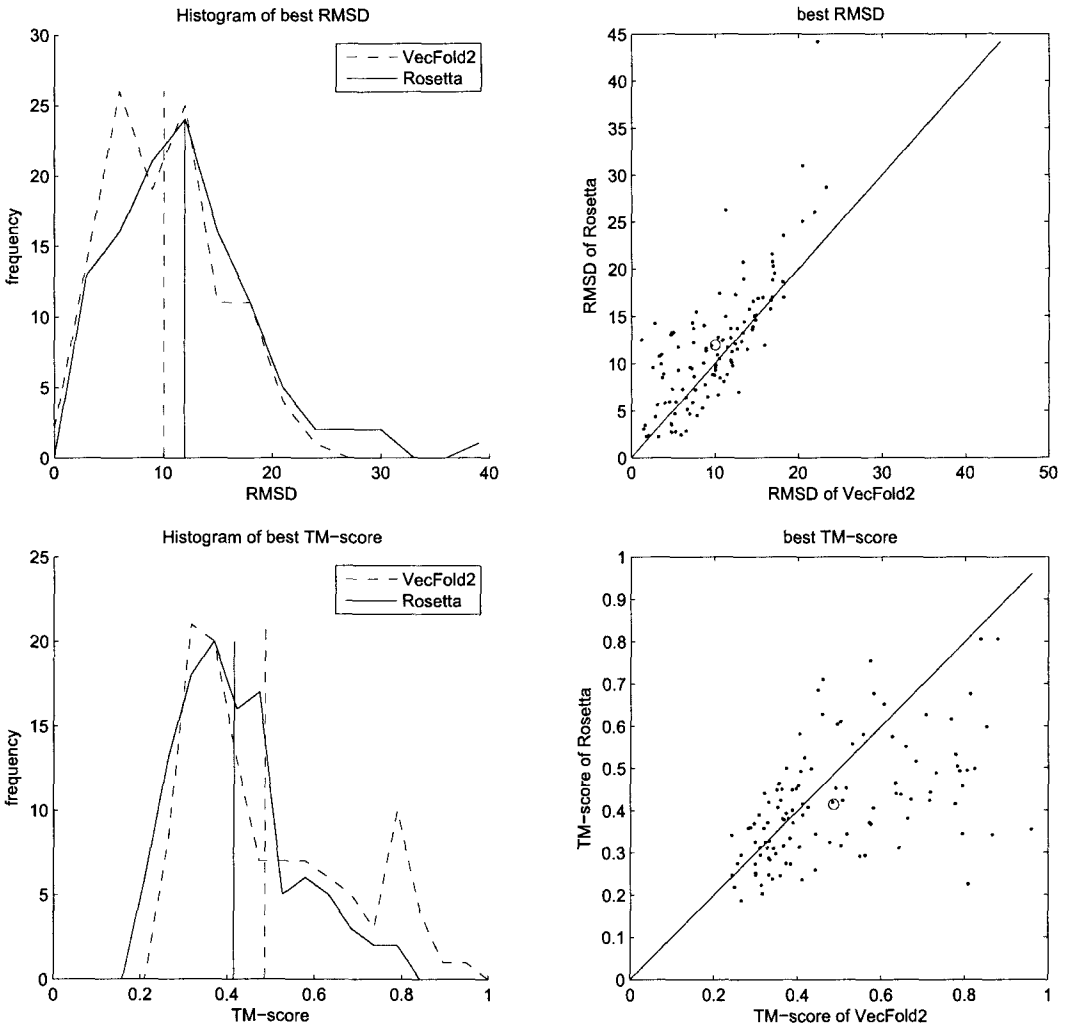

Figure C.2 : Best RMSD and TM-score, VecFold2 vs. Rosetta, CASP8

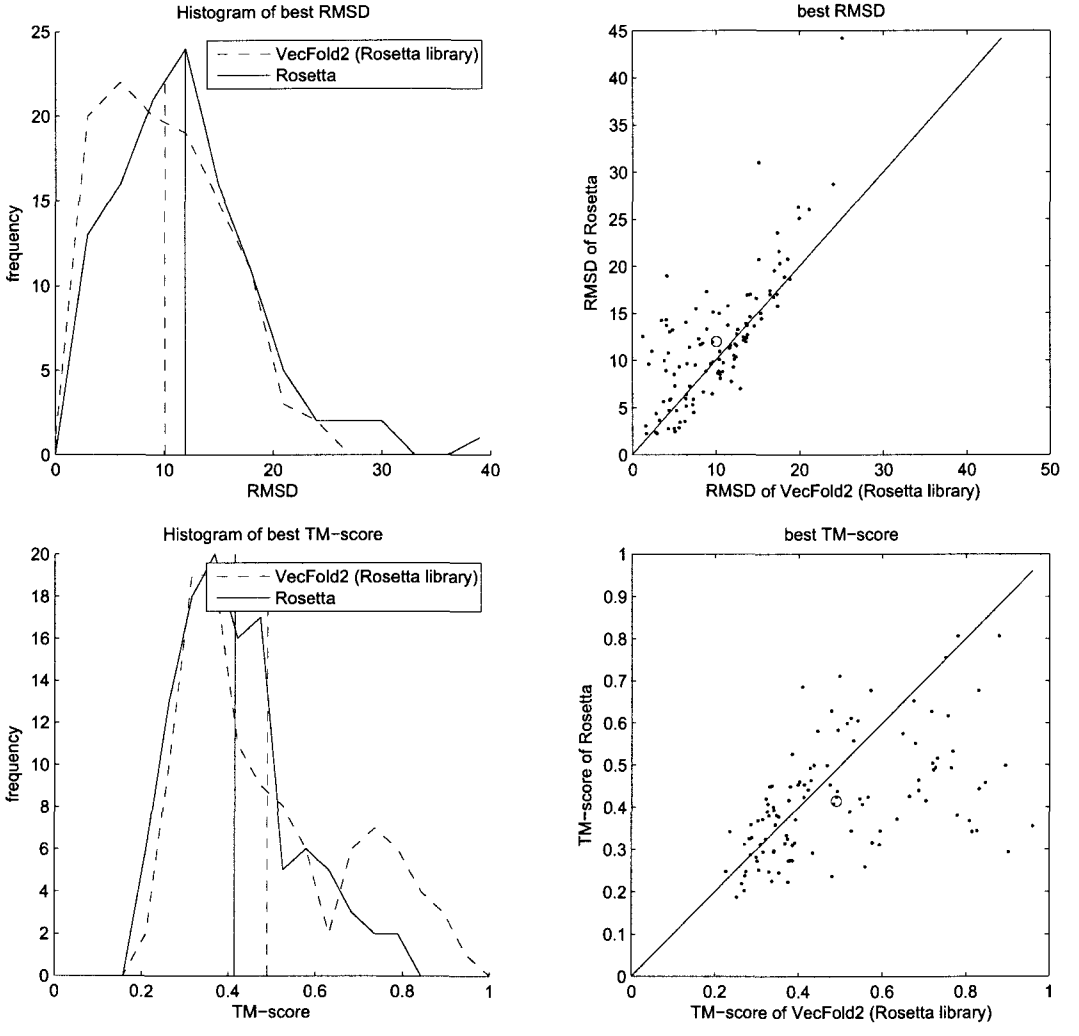

Figure C.3 : Best RMSD and TM-score, VecFold2 (with Rosetta template library) vs. Rosetta, CASP8

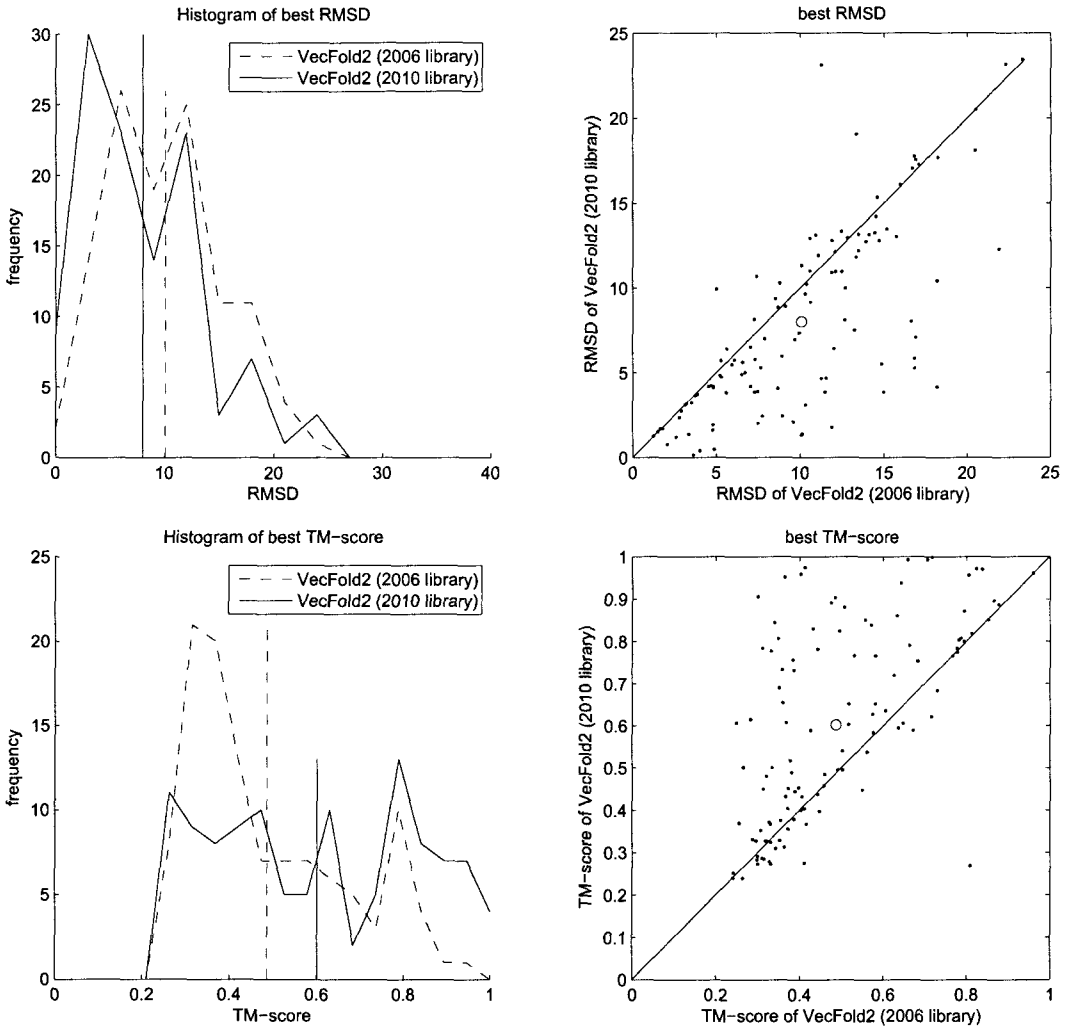

Figure C.4 : Best RMSD and TM-score, VecFold2 (2006 template library) vs. VecFold2 (2010 template library), CASP8

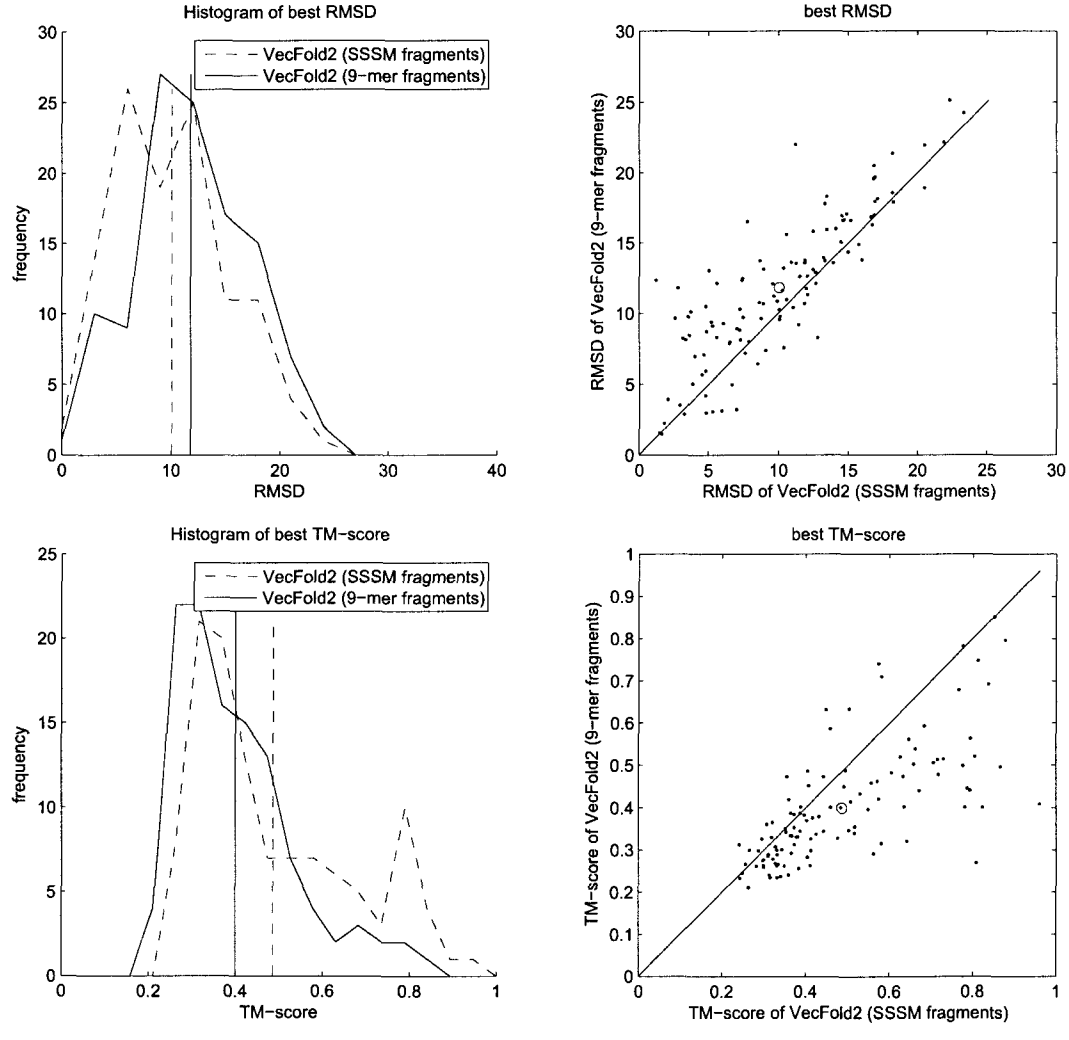

Figure C.5 : Best RMSD and TM-score, VecFold2 (SSSM fragments) vs. VecFold2 (9-mer fragments), CASP8

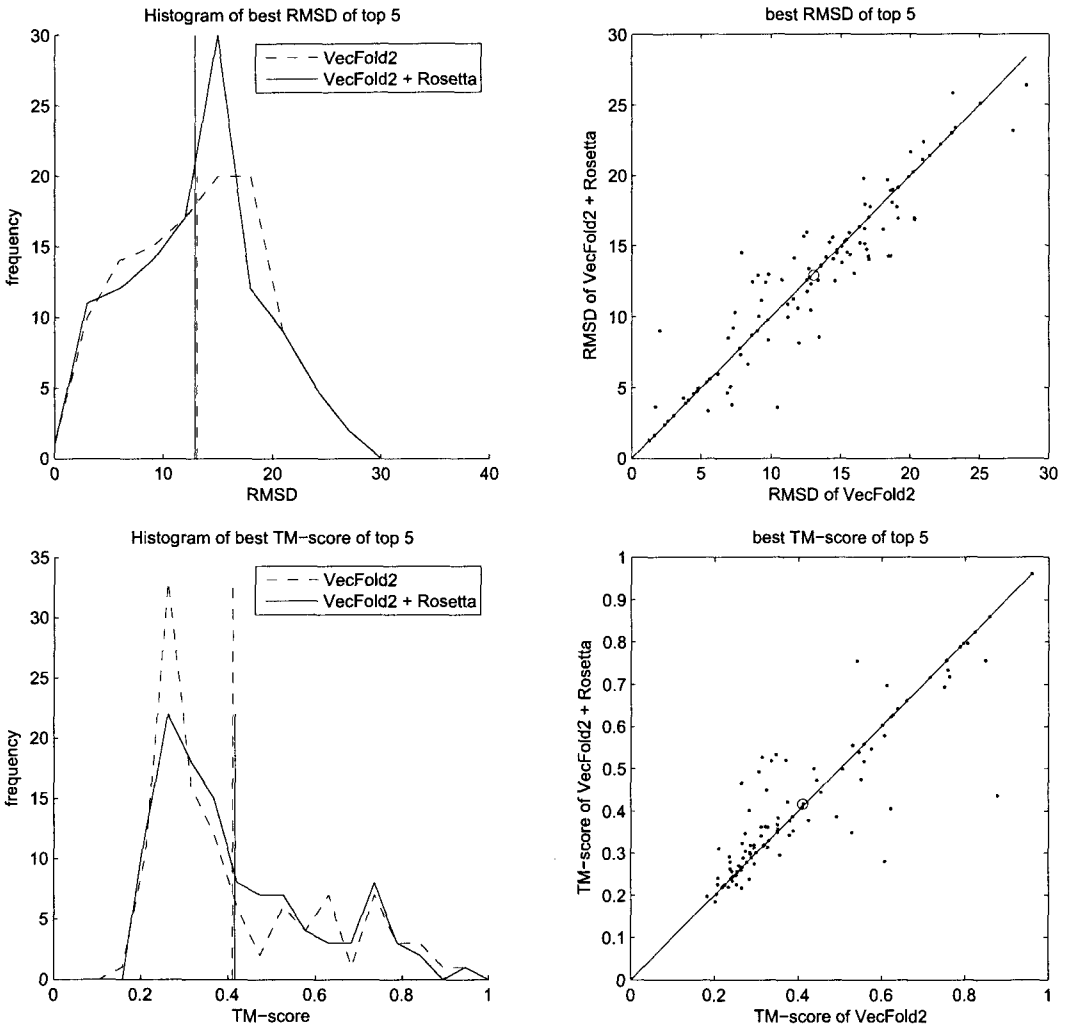

Figure C.6 : Best RMSD, TM-score of top 5 by energy-score, VecFold2 + Rosetta, CASP8

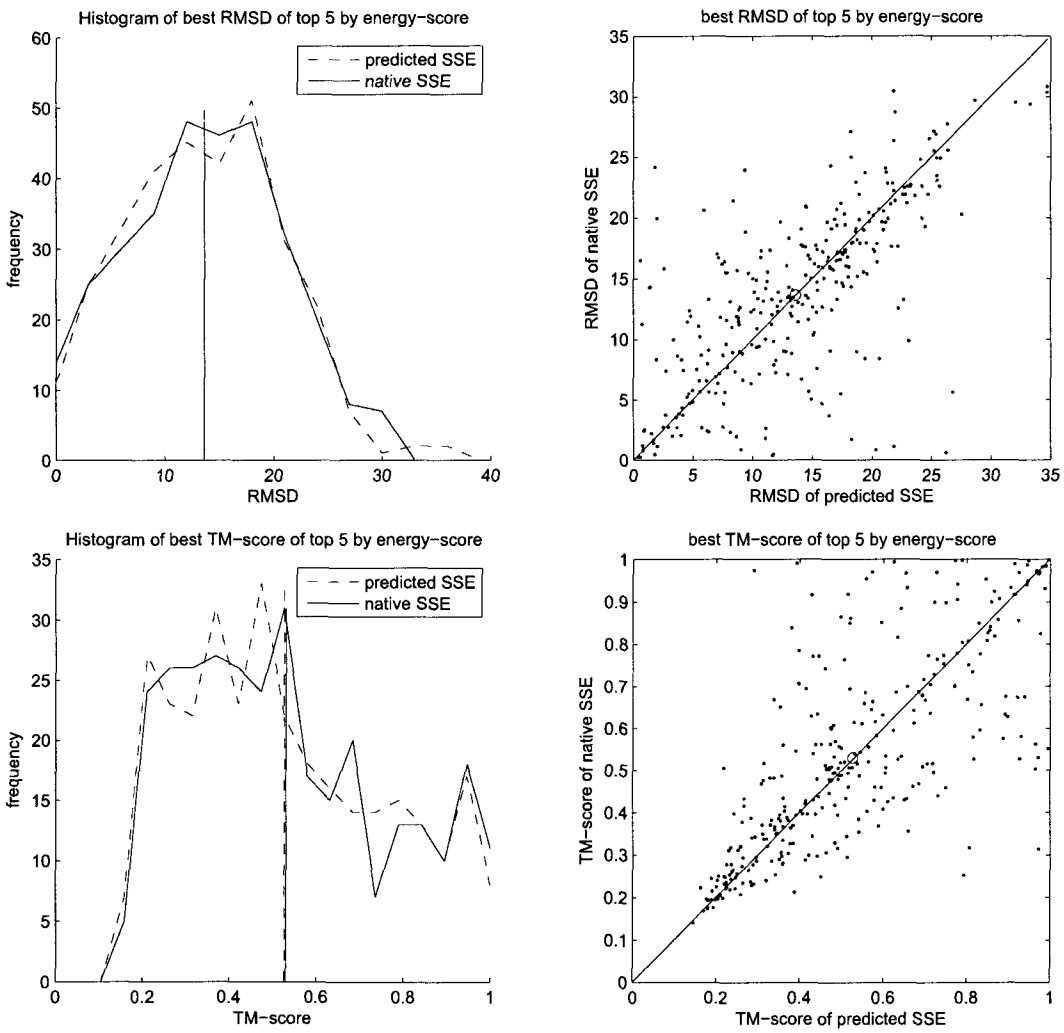

Figure C.7 : Best RMSD, TM-score of top 5 by energy-score, VecFold2 with predicted SSEs vs. VecFold2 with native SSEs, combined GM/Miyazaki/MMDB set