Title

Modeling Macromolecular Assemblies

Permalink

https://escholarship.org/uc/item/08b67421

Author

Kim, Michael F.

Publication Date

2008-09-04

Peer reviewed|Thesis/dissertation

Modeling Macromolecular Assemblies

by

Michael F. Kim

DISSERTATION

Submitted in partial satisfaction of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

Biological and Medical Informatics

in the

GRADUATE DIVISION

of the

UNIVERSITY OF CALIFORNIA, SAN FRANCISCO

[redacted]

Acknowledgments

When I reflect on my time at UCSF, I cannot help but feel incredibly blessed. I am filled with gratitude towards everybody who made my experience such an amazing intellectual, physical and spiritual journey.

I begin by thanking my thesis advisor, Andrej Sali. I remember choosing to work for him because I felt like our scientific points of view were so compatible. The elegance that Andrej insists upon finding in all aspects of his research inspired me to think abstractly about the challenges in structure modeling. And it is his unflagging optimism for science that helped carry me through to this point. I will cherish our energetic conversations on the theory of modeling and the amazingly fun and creative atmosphere that he cultivated in his lab.

I would also like to thank the members of my thesis committee: David Agard, for challenging me – I always had to bring my ‘A’ game around him; Ken Dill, for allowing me to rotate in his lab and exposing me to the beauty of simplified models; and Wendell Lim, for showing me that science can be a competitive and fun endeavor at the same time.

Other key faculty members: Patsy Babbitt, from whom I learned as much on how to be a gracious human being as I did on how to be a good scientist; Tom Ferrin, for guiding the BMI Program and being supportive of its students; and Ajay Jain, for aggressively recruiting me to the Bioinformatics program at UCSF and for providing the voice of a computer scientist.

The BMI program could not run without its program coordinators: Barbara Paschke, Denise Chan, Julie Ransom, Rebecca Brown, and Julia Molla. These five women are simply amazing.

I was also blessed to be surrounded by such talented peers and friends, to whom I knew I could always turn for scientific or personal advice, for laughs and smiles in the good times, as well as for hugs and support in the tough times. I would especially like to thank all of the members of Sali Lab (including Heidi Moss Sali), past and present, for being so much more than just my coworkers. I also thank my classmates: Simona Carini, Christina Chaivorapol, Ranyee Chiang, John Chuang, Libusha Kelly, Juanita Li, Mark Peterson, and Tuan Pham. Outside of my classmates and Sali Lab, I am also grateful to all of my friends at UCSF, a woefully incomplete list includes: Shoshana Brown, John Chodera, Kingshuk Ghosh, Courtney Harper, Terry Lang, Dan Mandell, Sunil Ojha, Michael Reese, Alex Schnoes, and Veena Thomas.

On the run, I would like to thank the members of the K-stars running club. In the pews, I would like to acknowledge my friends at KCPC. And at home, I would like to thank my past and present roommates: Scott Pegg, Daniel Thuillier, Shoshana Deutschkron and Jenna Zarate.

Finally, I would like to thank my family for their unwavering support. I would not be here without them, nor would I want to be here without them. In particular, I dedicate this dissertation to my parents – thank you for all of your patience, sacrifices, caring, wisdom, and love.

This dissertation consists of five chapters: an introduction, three chapters based on manuscripts (either published or submitted), and a chapter of work that will likely not be published in its current form. With the exception of the introduction, each of the other four chapters is prefaced by a brief summary of the chapter that places the work in context of the larger theme of modeling macromolecular assemblies and is followed by a section discussing future directions for the work contained in the chapter.

The main body of Chapter 2 takes the form of a manuscript that was published in the journal *Structure* on work with Frank Alber, on which I was the second author providing supporting analysis for this work. The other coauthor, Andrej Sali, listed on this publication directed and supervised the research that forms the basis for this dissertation.

The main body of Chapter 3 takes the form of a manuscript to be submitted on work with Michael Reese, on which we both shared equal amounts of responsibility. The coauthors, Volker Deutsch and Andrej Sali, listed in this publication directed and supervised the research that forms the basis for this chapter.

The main body of Chapter 4 takes the form of a manuscript to be submitted on which I was the primary author, on work completed with the technical assistance of Ben Webb. The other coauthor, Andrej Sali, listed on this publication directed and supervised the research that forms the basis for this dissertation.

Lastly, Chapter 5 represents work that will not likely be published in its current form, as the aim for the work was to be more theoretically descriptive than to produce specific, practical conclusions. However, the collection of models, described in the chapter, addresses various aspects of structure modeling and generalizes ideas presented in the preceding chapters, serving to relate the various themes of this dissertation together.

Abstract

Modeling Macromolecular Assemblies

Michael F. Kim

Macromolecular assemblies are fundamental to most biological processes and here we attempt to improve the structural characterization of assemblies in the hopes that with new and improved models will produce functional insights on assemblies.

Modeling of macromolecular assemblies begins with an analysis of the computational and experimental data available on the entire assembly, subcomplexes, individual subunits and the interactions between subunits. Having collected the data on the assembly, the next challenge is to integrate the disparate data to produce a structural model. Hybrid approaches, which integrate multiple sources of data, provide a way to increase the coverage and accuracy of structure modeling for macromolecular complexes.

Fitting in with the theme of hybrid methods, Chapters 2 and 3 describe methods for modeling macromolecular assemblies, combining overall shape information (*e.g.*, from cryo-electron microscopy) with interaction data (*e.g.*, tandem affinity purification assays); and protein structures (or models) with NMR spectroscopy, respectively.

Chapter 4 proposes an assessment strategy for structure modeling methods that provides a way to measure how much improvement is left to be made, instead of the traditional approach of measuring how much improvement was already made. This assessment strategy provides more information on the specific limitations of the method and provides specific insight into how to best improve the method. The strategy also

presents a more fair method of comparing competing methods that are assessed with different benchmark sets.

Chapter 5 describes the representation of subunits and assemblies by systems of points and restraints, explores the assumptions that underlie using points and restraints to model macromolecular structures, describes the properties of binary and multiple docking, and models the structure modeling framework.

The main contributions of this dissertation are two practical approaches for macromolecular assemblies; an assessment strategy that provides a more explicit description of the accuracy and limitation of assessed methods, improving the confidence with which the resulting models are used; and lastly, a deeper theoretical understanding of modeling macromolecular assemblies, including a path towards a more principled approach for integrating multiple sources of data.

Table of Contents

List of Tables

List of Figures

1. Introduction

While there are still many challenges in protein structure problem, this thesis focuses primarily on modeling macromolecular assemblies. Macromolecular assemblies are fundamental to most biological processes and here we attempt to improve the structural characterization of assemblies in the hopes that with new and improved models we will also provide functional insights on assemblies [1-3].

Modeling of macromolecular assemblies begins with an analysis of the experimental data available on the entire assembly, subcomplexes, individual subunits and the interactions between subunits. At the level of entire assemblies and subcomplexes, a number of large macromolecular structures are being solved at atomic resolution primarily by x-ray crystallography [4-7], and at lower resolution by electron cryo-microscopy [8-10] and tomography [11, 12]. Interaction data can come from experimental methods, such as chemical cross-linking [13-15]; footprinting [16]; immuno-electron microscopy [17]; fluorescence resonance energy transfer (FRET) [18]; site-directed mutagenesis [19]; protein arrays [20]; and yeast two hybrid [21, 22] as well as theoretical and bioinformatics methods [23-26]. Although the atomic structures are more informative, even a low-resolution configuration of subunits in an assembly is useful in biology and provides a starting point for a refinement by higher resolution methods [27-33]. And lower resolution experimental methods, such as small-angle x-ray and neutron scattering [34] can also provide restraints to help refine higher resolution models.

Having collected the data on the assembly, the next challenge is to integrate the disparate data to produce a structural model [1, 35]. Hybrid approaches, which integrate multiple sources of data provide a way to increase the coverage and accuracy of structure modeling for macromolecular complexes [1]. Throughout this work, we represent structures using points and the data on structures as spatial restraints as a unifying framework for representing heterogeneous sources of data. In this framework, subunits are represented by sets of points and possible assembly configurations are represented by configurations of these sets of points. Available data on structural features and relationships are then encoded by spatial restraints. These spatial restraints are then combined into a scoring function, which is then optimized to find the configuration (or ensemble of configurations) that is in least violation of these restraints and therefore in greatest agreement with the input data.

Fitting in with the theme of hybrid methods, Chapter 2 takes a closer look at combining of overall shape information and interaction data to produce assembly models [36]. We use simplified models of assemblies and simulated experiments to describe the properties of the individual contributing experimental sources as well as their combination. We also demonstrate the extraction of spatial restraints from non-spatial data. Specifically, we take simulated tandem affinity purification data and translate the resulting lists of interacting subunits into spatial restraints on the listed subunits.

Continuing on the theme of hybrid methods, Chapter 3 describes the combination of protein structures (or models) with NMR fast-mapping data [37, 38]. As in the work on combining overall shape information with interaction data, we sought to use the available experimental data efficiently. We describe a method for mapping binding sites on to two subunits and then perform a restrained docking on the subunits to produce protein-protein docking results. We also demonstrate the utility of employing orthogonal information gathered by experimental means on the selection of more accurate models. Specifically, we employ the residue count data from the NMR experiments to help score the resulting docking configurations.

Chapter 4 proposes a different way to assess modeling methods. For all structural modeling methods, the current methodology of measuring and reporting the accuracy of a method against a particular benchmark set could be improved to provide more information on the specific limitations of the method and the provide insight into how to best improve the method. Also, when competing methods are assessed with different benchmarks, it becomes difficult to fairly compare the two methods. For users of these modeling methods, having a more explicit description of the accuracy and limitation of the methods available will improve the confidence with which the resulting models are used. This new assessment strategy provides a way to measure how much improvement is left to be made, instead of the traditional approach of measuring how much improvement was already made.

Chapter 5 details the assumptions that underlie using points and restraints to model macromolecular structures, describing the representation of subunits and assemblies by systems of points and restraints. This work also describes the properties of binary and multiple docking systems modeled by this framework. Finally, this work aims to provide a path towards a more principled approach for integrating multiple sources of data. While abstract in nature, this chapter describes the theory behind much of the other work in this thesis and in so doing serves to relate the various themes together.

2. Assembly Structure Characterization from Shape and Subcomplex Data

Electron tomography and cryo-microscopy are indispensable tools for the structural characterization of macromolecular assemblies. However, if the resolution of the assembly density map is lower than ~3 nm or the subunit shapes are unknown, the subunit configuration is difficult to determine. For such situations, we propose an approach that relies on affinity purification data (pulldowns) as well as the assembly density map. While affinity purification is commonly used for identifying protein interactions, we apply it here to extract spatial restraints on protein interactions. To do so, we rely on structure characterization by satisfaction of spatial restraints that (i) represents subunits as spheres, (ii) encodes information about the assembly shape, subunit excluded volume, pulldowns, and symmetry in a scoring function, and (iii) finds subunit configurations that satisfy the input restraints by an optimization of the scoring function. To assess the approach, the accuracy of the optimized configurations was mapped as a function of the variety of simulated restraints for two model assemblies. We find that it is generally possible to determine the packing of subunits in an assembly, given a relatively modest number of pulldowns, the assembly shape, and the subunit excluded volumes. Thus, we suggest that complementing electron cryo-microscopy and tomography with pulldowns provides a way to bridge the resolution gap between the assembly shape and the subunit configuration.

We are grateful to Maya Topf, Dmitry Korkin, Fred Davis, Damien Devos, Min-yi Shen, and Ben Webb for many discussions about structure characterization by satisfaction of spatial restraints. We also acknowledge our collaborators, Brian Chait and Mike Rout, for providing biological context that inspired this study.

2.1. Introduction

The structures of a number of large assemblies are being solved at atomic resolution primarily by x-ray crystallography [4-7] or at lower-resolution by electron cryo-microscopy [8-10] and tomography [11, 12]. Although the atomic structures are more informative, even a low-resolution configuration of subunits in an assembly is useful in biology and provides a starting point for a refinement by higher-resolution methods [27-33].

If the resolution of the assembly density map is lower than ~3 nm or the subunit shapes are unknown, the subunit configuration is difficult to determine without additional experiments. In particular, this problem frequently applies to electron tomography, which is especially suitable for studying macromolecular assemblies in their native cellular context [39] but whose resolution is currently limited to less than ~4 nm. To bridge the resolution gap between the assembly shape and the subunit configuration, the assembly density map can be integrated with several additional types of structural information [40, 41]. This information includes data from experimental methods, such as chemical cross-linking [13-15]; footprinting [16]; immuno-electron microscopy [17]; fluorescence resonance energy transfer (FRET) [18]; small-angle x-ray and neutron scattering [34]; site-directed mutagenesis [19]; protein arrays [20]; and yeast two hybrid [21, 22] as well as theoretical and bioinformatics methods [23-26] .

In this paper, we focus on characterizing the subunit configuration by combining an assembly density map with one particular source of supplementary information, affinity purification assays. These pulldown experiments depend on a tagged protein subunit (the bait) of a complex. The bait and its non-covalently associated partners (the subcomplex) are first purified by affinity chromatography against the tag and then identified by gel electrophoresis and mass spectroscopy [17, 42]. Such affinity purification has been used to identify interacting proteins on large scale in yeast [43, 44]. In contrast to identification of protein interactions, here we exploit the pulldowns for structural characterization. Each affinity purification experiment, in principle, provides some information about spatial relationships among the subunits in the subcomplex. Specifically, all of the proteins identified in a single affinity purification experiment must be located within the expected volume of the subcomplex. Furthermore, each subunit in a subcomplex must interact directly with at least one other subunit in the same subcomplex. For a given assembly, many different subcomplexes can generally be generated by selecting each of the subunits within the assembly as the bait and by varying conditions under which the subcomplexes are purified.

To integrate varied information about the structure of an assembly, we express the structure determination as an optimization approach. In this approach, we need to specify a protein representation, a scoring function, and an optimization method. We use a simplified model with a protein subunit represented by a single sphere. This model can only reveal the configurations of and interactions between subunits, but not their individual conformations nor their relative orientations. Despite these limitations, the proposed representation allows us to encode the affinity purification data and low-resolution assembly density maps as spatial restraints on the subunit configuration, which are then combined into a single scoring function (Figure 2.1). Next, the scoring function is optimized to find all subunit configurations that satisfy the input restraints. To assess the utility of the combination of the affinity purification data and the assembly density map, the accuracy of the optimized configurations was mapped as a function of the variety of simulated restraints for two model assemblies.

Next, we describe in detail an approach to structural characterization by satisfaction of spatial restraints, as well as two model systems and analysis methods used in our calculations (Approach). We then compare the information content of different combinations of spatial restraints by assessing their predictive power for determining the native assembly structure (Results). We end by summarizing the main conclusions (Discussion).

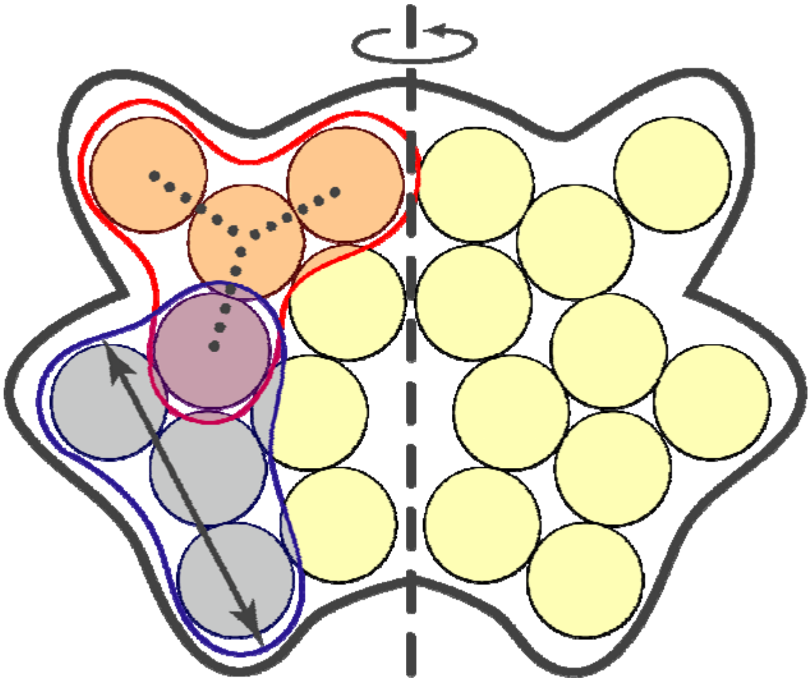

Figure 2.1. Five Information Types

Schematic representation of the five types of information that are assessed with respect to their utility for assembly structure characterization. First, the subunits and their excluded volume are indicated by yellow circles. Second, the assembly shape is indicated by a thick outline. Third, the shapes of two individual subcomplexes, each with four subunits, are shown in red and blue, and the largest diameter of the blue subcomplex is indicated by an arrow (proximity restraint). Fourth, the subunit interactions (connectivity restraint) in the red subcomplex are indicated by dotted lines. Fifth, the symmetry between two parts of the assembly is indicated by a vertical dashed line.

2.2. Approach

2.2.1. Structure Characterization by Satisfaction of Spatial Restraints

We express structure characterization as an optimization problem that calculates 3D models consistent with the input information. The three components of this approach are (i) a representation of the modeled assembly, (ii) a scoring function consisting of the individual spatial restraints, and (iii) an optimization of the scoring function to obtain all possible models that satisfy the input restraints. We describe all three components next.

2.2.2. Representation

Each protein subunit is represented as a point. The subunit excluded volume is encoded as a restraint and is described in the next section. The two specific model assemblies used in this paper are described below.

2.2.3. Scoring Function

The most important aspect of structure characterization by satisfaction of spatial restraint is to accurately capture all available input information about the structure of the assembly. We approach this problem by translating all structural information into spatial restraints. We distinguish restraints on five different spatial features (Figure 2.1): (i) the subunit excluded volume, (ii) the assembly shape, (iii) the subunit proximity in the subcomplex (the proximity restraint), (iv) the subunit connectivity in the subcomplex (the connectivity restraint), and (v) the symmetry. The scoring function is defined as the sum of all individual restraints, described in detail below. In summary, (i) subunit excluded volume restraints are expressed as lower bounds on all pairwise subunit distances; (ii) the proximity and (iii) connectivity restraints are expressed as pairwise upper distance bounds on the subunits within the subcomplex; (iv) the assembly shape restraints are expressed as lower and upper bounds on the absolute subunit coordinates; and (v) the symmetry restraints are expressed as distance restraints on two equivalent parts of the assembly.

In the case of assemblies with multiple copies of the same subunit type (such as the proteasome), there is an ambiguity in the calculation of the proximity and connectivity restraints. For example, there are two copies of each subunit type in the proteasome and four distances between pairs of distinct types. In principle, a restraint on two distinct subunit types could apply to any one of these four pairs. We consider all assignments and only restrain the pair of subunits that leads to the smallest restraint violation.

Subunit excluded volume restraint. The excluded volume restraint imposes a harmonic penalty if the distance between any two subunits is smaller then the sum of their radii (Table 2.1, row 1).

Assembly shape restraint. Subunits can be localized only within a restricted volume in the shape of the target assembly. A harmonic penalty is imposed if the absolute subunit coordinates are below or above the corresponding lower or upper bounds, respectively (Table 2.1, row 2).

Subunit excluded volume restraint	Violated for $f < f_o$,, f is the distance between two subunits, f_o is the sum of the subunit radii, and σ is 0.01 nm.
Assembly shape restraint	Lower bound: violated for $f < f_o$, f is the subunit Cartesian coordinate, f_o is the lower bound on this particular subunit coordinate, and σ is 0.1 nm. Upper bound: violated for $f > f_o$, f is the subunit Cartesian coordinate, f_o is the upper bound on this particular subunit coordinate, and σ is 0.1 nm.
Subcomplex proximity restraint	Violated for $f > f_o$, f is the distance between two subunits in a pulldown complex, f_o is the maximal subcomplex dimension, and σ is 0.1 nm.
Subcomplex connectivity restraint	Violated for $f > f_o$, f is the distance between two subunits, f_o is the sum of their radii, and σ is 0.1 nm.

Table 2.1. Definition of First Four Restraint Types

Each restraint term is equal to $(f - f_o)^2 / \Box^2$, where f is the restrained feature, and □ is the parameter that regulates the strength of the term. For upper feature bounds, the score is 0 for $f > f_o$; for lower feature bounds, the score is 0 for $f < f_o$.

Subcomplex proximity restraint. We impose upper distance bounds on all pairs of subunits in a pulldown subcomplex (Table 2.1, row 3). The upper bound is the largest possible distance between two subunits in a subcomplex and is equal to the maximal diameter of the subcomplex minus the subunit radii. The same subunit pair may appear in multiple subcomplexes and therefore lead to several upper distance bounds. We keep only the smallest of all pairwise upper bounds.

Subcomplex connectivity restraint. Each subunit in a subcomplex must contact at least one other subunit in the subcomplex. For example, in a subcomplex with *n* components, at least *n-1* direct interactions must connect all of its subunits. We refer to this condition as the connectivity restraint of a subcomplex. While the actual subunit contacts are unknown, all valid structural solutions must satisfy this restraint. For a given subcomplex, the restraint is applied with the aid of a minimal spanning tree as follows. We define a fully connected graph with the nodes corresponding to the individual subunits and edges with weights equal to the violation of the hypothetical contact (Table 2.1, row 4). We then find the minimal spanning tree such that the sum of the edge weights is minimal and all subunits are connected to at least one other subunit [45]. For each edge in the minimal spanning tree, we impose harmonic distance restraints enforcing the direct subunit contacts (Table 2.1, row 4). At each step of the optimization, we recalculate the fully connected graph and the minimal spanning tree for each subcomplex.

Symmetry restraints. The similarity between the subunit configurations in each symmetry unit is enforced by imposing a term similar to the distance-root-mean-square (DRMS), $\Sigma_{\mathrm{ij}}\ \omega_{ab}\ (d_{ij}^{a} - d_{ij}^{b})^{2}$, where d_{ij}^{a} and d_{ij}^{b} are the equivalent distances between two subunits i and j in two symmetry related subunit configurations a and b, and ω_{ab} is the restraint weight set to 0.2.

2.2.4. Optimization

We generate subunit configurations by simultaneously minimizing violations of all restraints in Cartesian space. The aim is to obtain as many structures as possible that satisfy all input restraints. The generation of these models is stochastic. For each restraint set, we start from at least 10,000 completely randomized subunit configurations. We use an adapted version of the program MODELLERv7 [46].

An optimized structure is obtained from a single optimization run in a series of steps: The initial Cartesian coordinates of all subunits are randomly distributed from -50 to 50 nm, followed by conjugate gradients minimization of up to 500 steps and subsequent molecular dynamics with simulated annealing. The temperature of the system is increased from 100 to 1000 K within 50 time steps, kept constant for further 100 times steps, and gradually decreased to a temperature of 10 K in 300 time steps. This temperature is kept constant for another 50 time steps, followed by a final optimization by conjugate gradients of up to 1,000 steps.

2.2.5. Model Systems

We use two simple model systems. First, we study a compact assembly consisting of subunits packed in a cube (Figure 2.2). Second, we expand our calculations to a more realistic example, a low-resolution model of the proteasome (Figure 2.4).

Cube model system. The cube assembly consists of 27 different subunits located at the grid points of a 6 nm x 6 nm x 6 nm lattice (Figure 2.2). All subunits are represented as hard spheres with radii of 1 nm. The assembly contains 54 distinct binary contacts shown as a contact map in Figure 2.3. For the assembly shape restraint, the shape is a cube with side lengths of 6 nm. For the subcomplex proximity restraint, the maximal distances between subunit centers in subcomplexes with 3 to 8 subunits are 4, 4.47, 5.66, 6.00, 6.93, and 6.93 nm, respectively.

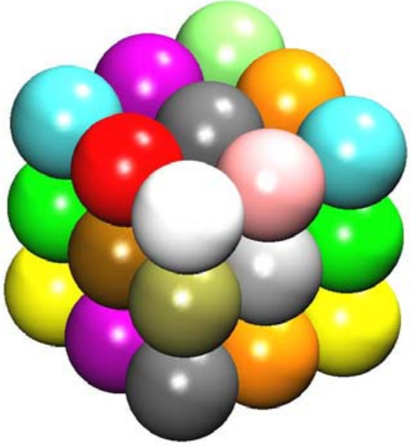

Figure 2.2. Cube Model System Native Structure

The native cube assembly consisting of 27 different subunits each one represented by a single sphere with a radius of 1 nm. The subunits are located at the grid points of a 3x3x3 lattice.

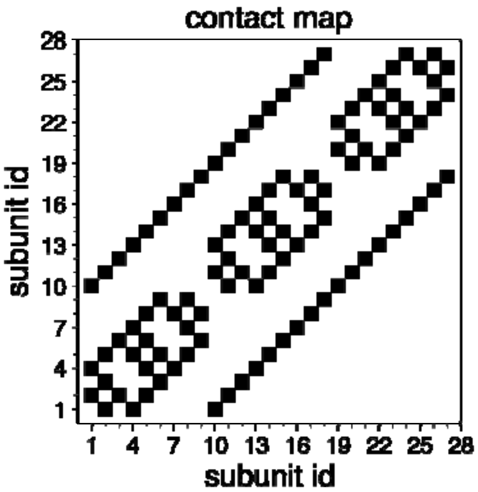

Figure 2.3. Cube Model System Native Contact Map

The corresponding native contact map with 54 binary subunit contacts.

Proteasome model system. The proteasome consists of 28 globular proteins of 14 different types that are arranged in two identical pairs of rings (Figure 2.4). We approximate each protein by a single sphere with its radius (in nm) estimated from the total protein mass: $r = 0.0726\ M^{1/3}$, where M is the protein mass in Da and the coefficient is determined based on masses and sizes of known protein structures. The sphere center is located at the center of mass of the corresponding protein in the x-ray structure of the proteasome [47]. For the assembly shape restraint, the shape is a cylinder with a height of 16.2 nm and a radius of 3.3 nm. For the subcomplex proximity restraint, the upper bound is 1.35 times the estimated maximal subcomplex diameter (in nm) from the empirical relationship between the maximal diameter of a subcomplex and its total number of residues: $D = 0.495\ n^{1/3}$, where n is the total number of residues in the subcomplex. The parameter value of 0.495 was derived by fitting the function to the structurally defined protein assemblies in PIBASE [48], such that 95% of all complexes have predicted maximal diameters that are larger or equal to the actual diameters.

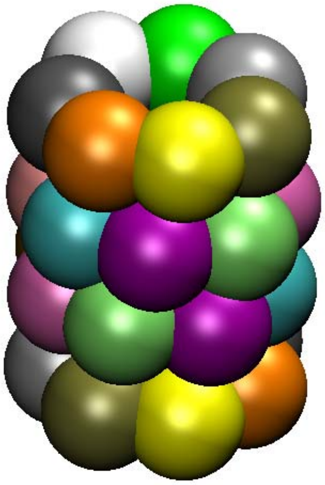

Figure 2.4. Proteasome Model System Native Structure

A low-resolution model of the proteasome with 28 protein subunits. There are 14 different protein types, each occurring twice. Each subunit is represented as a single sphere located at the gravity center of the corresponding protein in the crystal structure of the assembly [47]. The sphere radii are estimated from the number of residues in each protein (Approach).

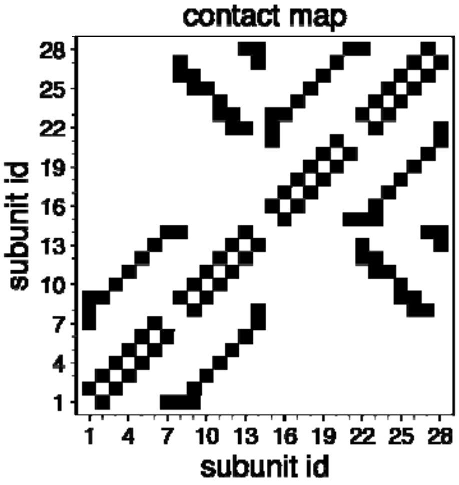

Figure 2.5. Proteasome Model System Native Contact Map

The corresponding native contact map with binary subunit contacts in the low-resolution proteasome structure.

Simulation of pulldown subcomplexes. Subcomplexes are generated by an iterative random selection of subunits that are in direct contact with each other in the native structure. A starting point is a subunit that is selected as the bait of the subcomplex. The acceptance of a newly selected subunit is probabilistic; the probability for accepting a subunit is proportional to the inverse cube of the contact shell number, which is the smallest number of subunits that connect the selected subunit with the bait. A uniform selection probability would lead to artificially elongated subcomplexes, as the number of neighbors in higher contact shells grows rapidly.

Generation of additional models. For some restraint sets (*eg*, derived from subcomplex sets 7 and 8 in Table 2.2c), the optimization protocol was unable to generate a sufficient number of structures that satisfied all the input restraints even in 500,000 independent runs. In such cases, we increased the sample size needed for estimating the utility of various restraint sets for structure characterization as follows. We generated

3,000 additional structures from the native structure by swapping subunits between one and ten randomly selected subunit pairs in the assembly. For the proteasome model, each swap involved two pairs of subunits, one in each symmetry unit. If a structure satisfied all input restraints, it was added to the ensemble of good scoring structures generated in the optimization process.

Subcomplex set	3	4	5	6	7	8
a) Subcomplex proximity restraints						
Models satisfying input restraints [%]	100.0	100.0	100.0	100.0	100.0	100.0
Sensitivity*	22.2	16.6	18.5	20.3	37.0	37.0
False positive rate*	52.2	35.7	41.2	50.0	54.4	71.0
Fraction of correctly predicted models [%]	0.0	0.0	0.0	0.0	0.0	0.0
DRMS [nm]: Smallest, average, largest	1.6, 1.9, 2.2	1.3, 1.6, 1.7	1.7, 1.8, 2.0	1.7, 1.7, 2.2	1.7, 1.9,2.2	1.6, 1.8, 2.0
b) Subcomplex proximity and assembly shape restraints						
Models satisfying input restraints [%]	3.4	1.3	16.0	25.4	36.0	20.7
Sensitivity*^ [%]	48.0	61.0	40.7	57.0	62.9	46.3
False positive rate*[$] [%]	18.8	23.3	46.3	57.5	75.0	77.7
Fraction of correctly predicted models [%]	0.0	1.0	0.0	0.0	0.0	0.0
DRMS [nm]: Smallest, average, largest	0.6, 1.2, 1.5	0.0, 1.1, 1.4	1.1, 1.4, 1.6	1.3, 1.5, 1.7	1.4, 1.6, 1.8	1.4, 1.7, 1.8

c) Subcomplex proximity, subcomplex connectivity, and assembly shape restraints						
Models satisfying input restraints [%]	0.04	0.1	<0.1	<0.1	<0.1	<0.1
Sensitivity*^ [%]	100.0	100.0	100.0	100.0	100.0	100.0
False positive rate*$ [%]	0.0	0.0	0.0	0.0	0.0	0.0
Fraction of correctly predicted models [%]	75.0	50.0	100.0	25	33.0	75.0
DRMS [nm]:Smallest, average, largest	0.0, 0.1, 0.5	0.0, 0.2, 0.5	0.0, 0.0, 0.0	0.0, 0.3, 0.4	0.0, 0.2, 0.7	0.0, 0.0, 0.0

Table 2.2. Properties of Cube Models Satisfying All Input Restraints

Properties of models satisfying all input restraints that are derived from the 6 subcomplex sets 3-8 ("Cube model system" in Results). Models are calculated using subunit excluded volume restraints and (a) subcomplex proximity restraints, (b) subcomplex proximity and the assembly shape restraints, and (c) subcomplex proximity, subcomplex connectivity, and the assembly shape restraints.

* calculated using the reference frequency cutoff (Approach).

^ sensititvity defined as TP/(TP+FN), where TP is the number of true positive contacts and FN is the number of false negative contacts.

# specificity defined as TN/(FN+FP), where TN is the number of true negative contacts and FP is the number of false positive contacts.

$ false positive rate defined as FP/(FP+TP), where FP is the number of false positive contacts and TP is the number of true postive contacts.

Subcomplex set	14	28
Sensitivity[^]	96.3	100.0
False positive rate[$]	0.0	0.0
DRMS [nm]: Smallest, average, largest	0.0, 0.8, 1.7	0.0, 0.0, 0.0

Table 2.3. Properties of Proteasome Models Satisfying All Input Restraints

Properties of models satisfying all input restraints that are derived from subcomplex sets containing 14 and 28 subcomplexes ("Proteasome model system" in Results). Subunit excluded volume, assembly shape, subcomplex proximity, subcomplex connectivity, as well as symmetry restraints are applied (Approach). See the legend of Table 2.2 for the definitions of sensitivity and false positive rate.

2.2.6. Analysis

Analysis is performed only on models that completely satisfy all input restraints (good scoring models).

Contact frequencies. A subunit contact is defined if the distance between the two subunits is smaller then the sum of their radii multiplied by a tolerance factor of 1.05. The contact frequency is defined as the ratio of the number of models with the contact and the number of all models.

Receiver Operating Characteristic (ROC) analysis. The ability of different restraint sets to predict the native subunit interactions is ranked with the aid of the ROC curves [49]. For an ensemble of models calculated by a given restraint set, a subunit interaction is predicted if the corresponding contact frequency is sufficiently high (below). The accuracy of the predicted subunit interactions is quantified by calculating the true positive rate (sensitivity) as well as the false positive rate (1-specificity) and plotting them against each other at 16 different cutoff values (the ROC curve). The area under the ROC curve represents the probability of correct classification over the whole range of cutoffs; it can range from 0.5 to 1. An area of 0.5 indicates that the structure calculation could not discriminate between the native and false contacts. If the area under the ROC curve equals 1, the method is able to predict the contact map of the native structure. The closer the ROC curve is to the upper left corner and the closer the integrated area under the curve is to 1, the higher is the overall accuracy of the calculations and the more informative the restraints are about the native contact map of the assembly.

Reference frequency cutoff. This cutoff is defined as 56% of the largest contact frequency value present in a contact frequency map. This value was obtained by maximizing the sum of true positives and true negatives for the restraint set derived from subcomplex set 4 (Table 2.2a) and was adopted as a reference value for the analysis of all the restraint sets. Varying the reference cutoff value in a wide range from 30 to 90% does not change the ranking of the restraint sets by their utility in structure characterization. For convenience, the false positive rates and the number of correctly predicted contacts for each restraint set are determined using the reference frequency cutoff value.

2.3. Results

We rely on two simple model systems with globular protein subunits represented as single spheres (Approach). Our aim is to enumerate all subunit interaction networks and configurations that are consistent with subunit excluded volume, protein affinity purification experiments, mass density maps determined by electron cryo-microscopy or tomography, and potentially symmetry (Approach) (Figure 2.1).

We focus on the utility of affinity chromatography purification for structure characterization. In principle, each pulldown subcomplex contains some information about spatial relationships between its subunits (Approach). One such spatial restraint is the upper distance bound on any two subunits in a subcomplex, which we refer to as the "proximity restraint." The dimension of a subcomplex may be derived from hydrodynamic experiments [50], small angle x-ray scattering [34], and negative-stain or electron cryo-microscopy images [8]. Another spatial restraint, the "connectivity restraint," specifies that every subunit in a subcomplex must interact with at least one other subunit in the subcomplex. While the actual subunit interaction network is unknown, all valid structural solutions must satisfy this connectivity restraint.

2.3.1. Cube Model System

Our first model system is an assembly of 27 different subunits, represented as single hard spheres of identical radii in a cubic close-packed lattice (Figure 2.2). For each number of subunits per subcomplex from 3 to 8, we independently simulate 27 pulldown experiments with each subunit selected as the bait. The corresponding 6 sets of 27 subcomplexes are labeled subcomplex sets 3 to 8 (columns 3 to 8, Table 2.2). These subcomplex sets allow us to investigate which subcomplex size is most informative about the structure of the assembly.

We consider three combinations of restraint types: First, we use a combination of the excluded volume restraints for each subunit and the proximity restraints for each of the 27 subcomplexes as the only information for structure characterization (Table 2.2a); second, we add the assembly shape restraint (Table 2.2b); and third, we also add subcomplex connectivity restraints (Table 2.2c). This sequential buildup of the scoring function allows us to isolate the individual contributions to the structural characterization of assemblies.

Subcomplex proximity restraints

We begin by considering only subunit excluded volume restraints and subcomplex proximity restraints calculated from 6 sets of subcomplexes with 3 to 8 subunits per subcomplex (Approach). For each of the 6 restraint sets, at least 10,000 random subunit configurations were optimized in an attempt to find those configurations that satisfy all input restraints (good scoring models). We then predict a subunit interaction if it occurs frequently in the ensemble of good scoring models. Finally, we rely on the Receiver Operating Characteristic (ROC) curves to rank the different restraint sets by their ability to correctly predict the native contacts.

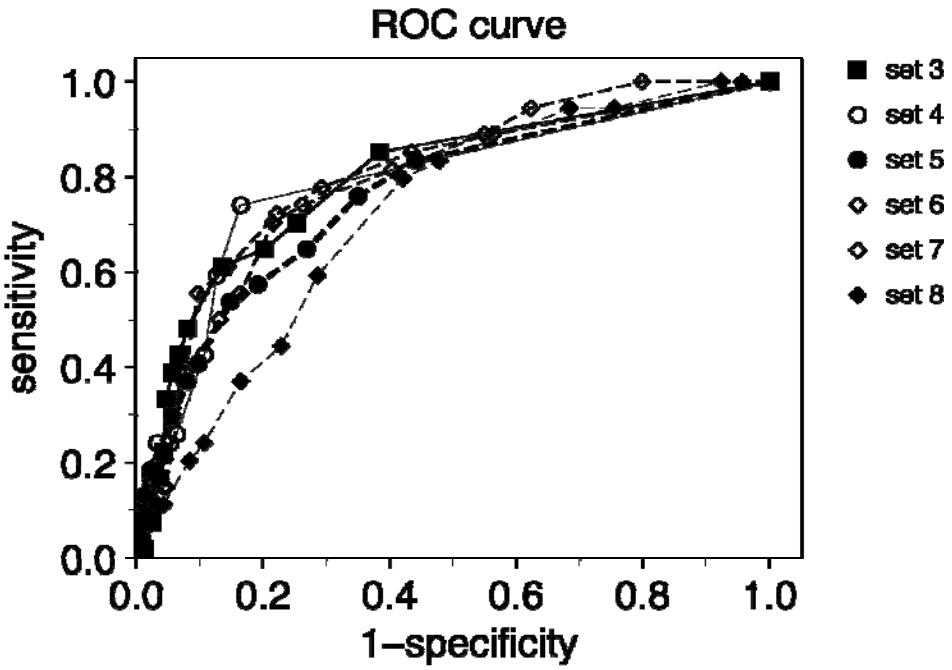

Figure 2.6. ROC Curve for Proximity Restraints

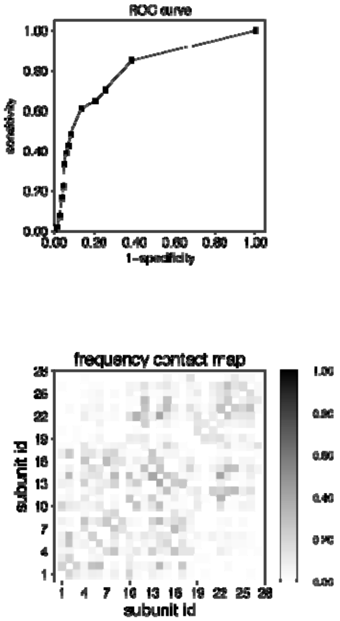

Figure 2.7. Average ROC Curve and Contact Map for Proximity Restraints

The ROC curves for subcomplex sets 3 - 7 are similar to each other (Figure 2.6). The overall performance is poor, as indicated by the small integrated area under the ROC curves that ranges from 0.7 to 0.8 for all subcomplex sets (Figure 2.12). Even for the two best performing subcomplex sets, 3 and 4, only respectively 12 and 14 out of the total of 54 native interactions are predicted correctly (the corresponding false positive rates are ~50% and 36%). This poor performance is also revealed by the 3D structural analysis of the models. The average DRMS deviation between models and the native structure ranges from 1.6 to 1.9 nm (Table 2.2). Therefore, it is not possible to correctly determine the assembly structure only by the subunit excluded volume and subcomplex proximity restraints.

Subcomplex proximity and assembly shape restraints

Next, we investigate the effect of adding the assembly shape restraint on the accuracy of our predictions. We use the same subcomplex sets 3-8, but now restrict the positions of the subunits to be within the assembly shape (a cube with side length of 6 nm) (Approach).

With the addition of the assembly shape restraint, the models are generally more compact. For some of the restraint sets, a substantial fraction of the native contacts can now be predicted correctly. For example, 26 of the 54 native contacts occur in 60% of all models calculated from the restraint set 3, with the false positive rate of 18.8% (Table 2.3b; Figure 2.9). The number of subunits per subcomplex makes a significant difference in the utility of the corresponding restraints, as indicated by the spread of the ROC curves in Figure 2.8.

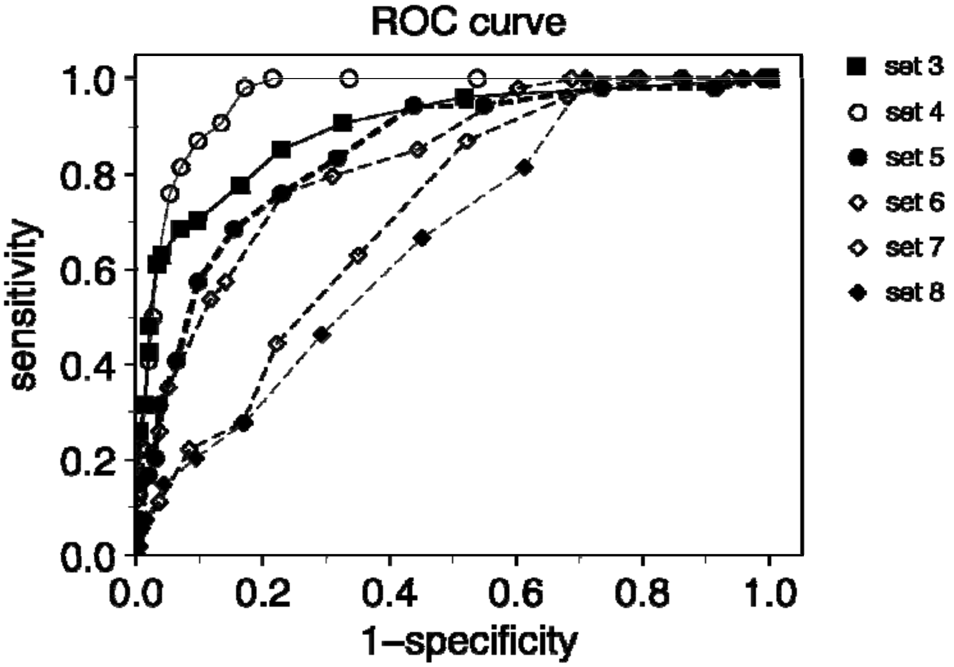

Figure 2.8. ROC Curve for Proximity and Shape Restraints

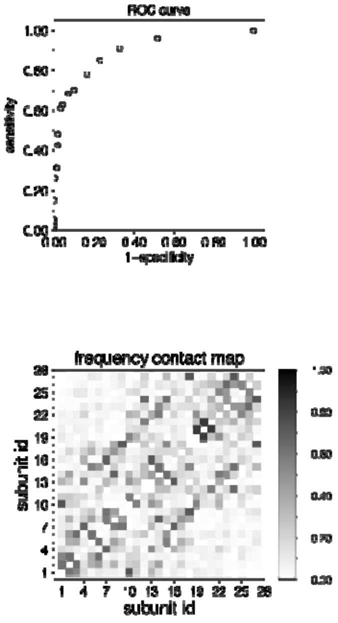

Figure 2.9. Average ROC Curve and Contact Map for Proximity and Shape Restraints

Subcomplex sets with a large number of subunits (*eg*, sets 7 and 8) perform worse with the assembly shape restraint than without it (*cf*, Figure 2.12a and Figure 2.11b). For example, for subcomplex set 7, the integrated ROC area for subcomplex sets 7 and 8 decreases from 0.78 to 0.71 and the false positive rate for subunit interaction prediction rises from 57% to 75% (subcomplex set 7 in Table 2.2). This finding is not surprising as the estimated diameter of subcomplexes with 7 and 8 subunits is similar to the maximum diameter of the assembly. Therefore, subcomplex sets 7 and 8 do not provide any additional structural information if the assembly shape is already specified. However, the increased number of contacts (both native and non-native) resulting from the reduced accessible volume increases the false positive rate and therefore decreases the prediction accuracy as quantified by a measure that depends on the subunit contacts. While it may be surprising that the accuracy of contact prediction from subcomplex sets 7 and 8 is decreased upon addition of the assembly shape restraint, other aspects of the predicted structures are improved; for example, the accuracy of the shape prediction (data not shown).

In contrast, for subcomplex sets with a smaller number of subunits (*eg*, subcomplex sets 3 and 4), the prediction accuracy is strongly improved upon adding the assembly shape restraint. The highest accuracy is found for subcomplex set 4 (Figure 2.11), with 33 out of the 54 native contacts correctly determined, in comparison to the prediction of 12 native contacts without the assembly shape restraints. Also, the false positive rate drops from 36% to 23% and the integrated ROC area increases from 0.8 to 0.96 (subcomplex set 4 in Table 2.2b). Correspondingly, the structural similarity among the models that satisfy the input restraints increases and their average DRMS deviation to the native structure is ~1.1 nm (Table 2.2). Approximately 1% of all models in subcomplex set 4 have all native contacts predicted correctly.

Subcomplex proximity, assembly shape, and subcomplex connectivity restraints

Finally, we investigate the effect of adding the connectivity restraint on the accuracy of our predictions. Using the same subcomplex sets, we now enforce that each subunit in a subcomplex is connected to the rest of the subcomplex subunits via at least one direct contact (subcomplex connectivity restraints in Approach). For the subcomplex sets with a small number of subunits (3 and 4 components), the current optimization scheme provides a sufficient number of models for subsequent analysis. However, for larger subcomplex sets (between 5 and 8 subunits), we supplement the structures provided by the optimization scheme with additional structures (Approach) to improve the reliability of the results.

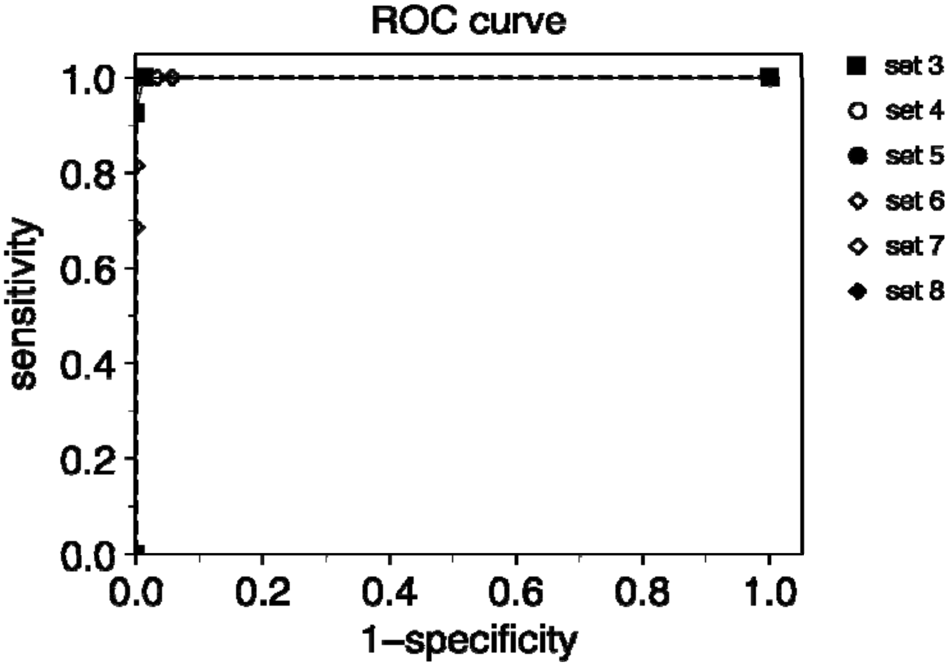

Figure 2.10. ROC Curve for Proximity, Connectivity and Shape Restraints

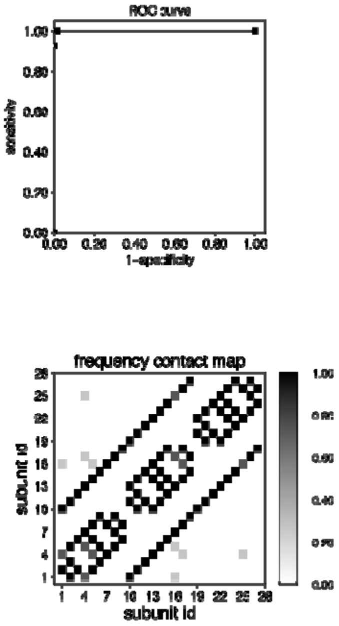

Figure 2.11. Average ROC Curve and Contact Map for All Restraints

Adding subunit connectivity restraints leads to a dramatic improvement in the accuracy of structure determination. The contact frequency maps for all subcomplex sets are almost identical to the contact map of the native structure (*eg*, Figure 2.3 and 6c). Indeed, for subcomplex set 3, all native contacts are reproduced in the good scoring models with a frequency of at least 75% (50 contacts with frequency 100% and 4 contacts with frequency 75%). Hence, the integrated ROC area is ~1 for all subcomplex sets (Figure 2.11). Using the reference frequency cutoff value (Approach), we are able to determine the complete subunit interaction network of the native structure with a false positive rate of 0 (Figure 2.11 and Table 2.2c). Structural comparison between the native structure and all models that satisfied the input restraints revealed an average DRMS deviation ranging from 0.1 to 0.3 nm (Table 2.2c). Indeed, for all of the subcomplex sets, some of the predicted structures differed only by a single interchange of neighboring subunits. Moreover, for the reference frequency cutoff, only models identical to the native structure have the contacts represented in the contact map. Therefore, the native structure can be identified reliably as the most frequently occurring predicted model.

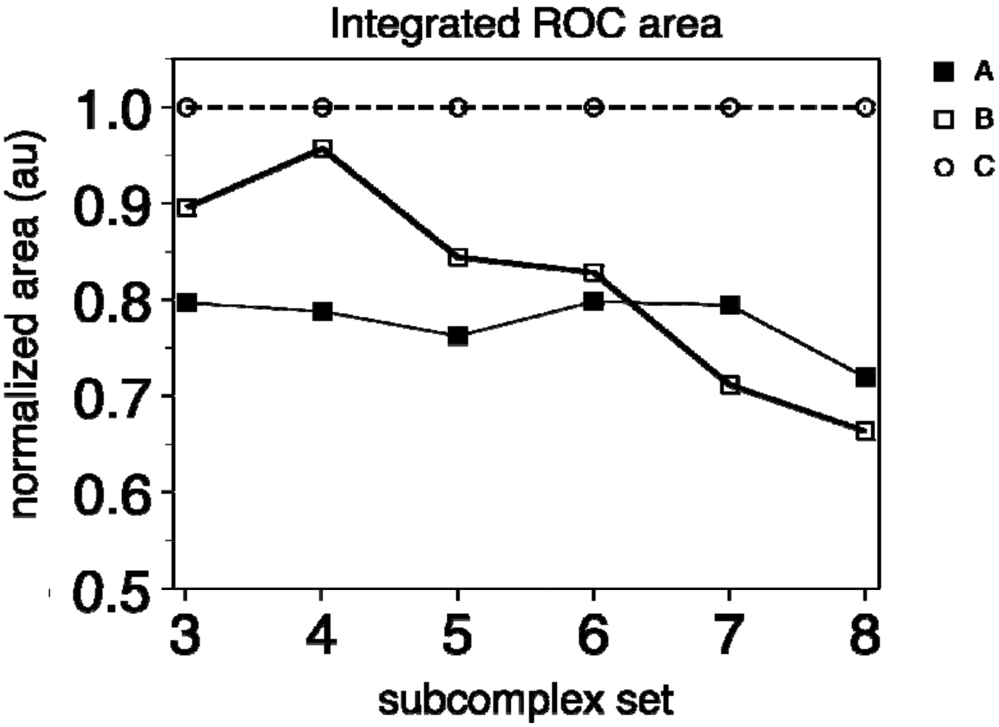

Figure 2.12. Integrated Area Under ROC Curves

Integrated area under ROC curves for calculations with restraints derived from 6 different subcomplex sets 3 to 8 ("Cube model system" in Results). Models are calculated using subunit excluded volume restraints and (A) subcomplex proximity restraints; (B) subcomplex proximity and assembly shape restraints; and (C) subcomplex proximity, subcomplex connectivity, and assembly shape restraints.

2.3.2. Proteasome Model System

Having demonstrated that it is possible to determine the 3D configuration of a simple model assembly, we turn our attention to the more realistic case of the proteasome.

Given the shape of the proteasome, a soft sphere representation of each of the proteins (one sphere per protein), and a new symmetry restraint (Approach), we assessed the information content of a relatively modest set of subcomplexes (with 14 and 28 subcomplexes per subcomplex set) (Table 2.3). Each of these subcomplexes contained between 3 to 5 subunits, with an average of 4 subunits in each subcomplex set. Instead of calculating models by the optimization of the scoring function, we constructed 3000 structures that differed from the native proteasome by a DRMS of 0.0 to 4.1 nm (Figure 2.13) (Approach). These structures were evaluated by the scoring function. All models with scores less than five times the score of the native structure were included in the analysis (Figure 2.13).

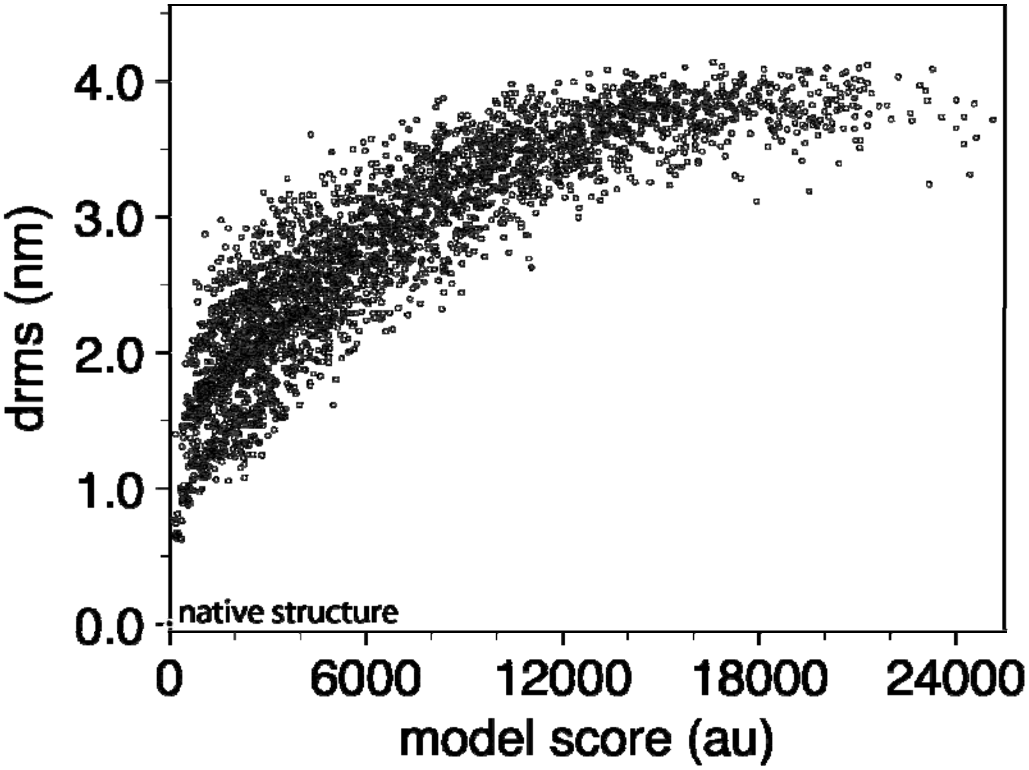

Figure 2.13. Structural Similarity vs. Sampling

The structural similarity between the proteasome models and the native structure plotted against the corresponding model score derived from an input dataset containing 28 subcomplexes. All models with scores (au, arbitrary units) less than five times the score of the native structure were included in the analysis. The native structure is indicated in the lower left corner.

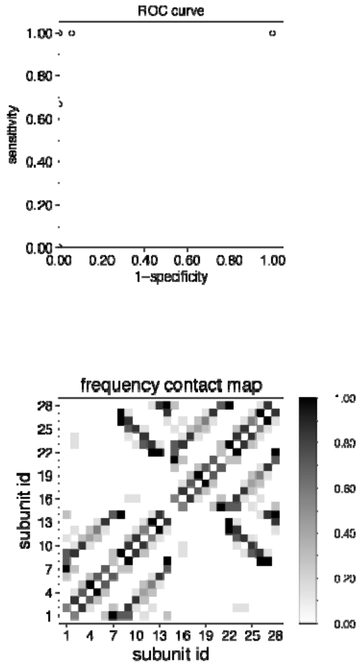

Figure 2.14. ROC Curve and Contact Map for 14 Subunits

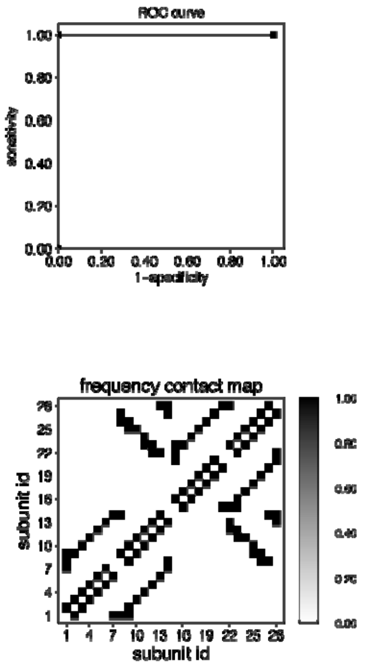

Figure 2.15. ROC Curve and Contact Map for 28 Subunits

With 14 subcomplexes, we were able to predict 55 out of the 57 native contacts with an error rate of 0, using the reference frequency cutoff (Table 2.3, Figure 2.14) (Approach). As expected, the subcomplex set with 28 subcomplexes performed even better, predicting the complete subunit interaction network (Table 2.3, Figure 2.15). For both cases, the integrated ROC area is ~1, indicating the highly discriminative power of the scoring functions (Figure 2.14 and Figure 2.15). The scoring function derived from 14 subcomplexes allowed several models that differed only by a single interchange of neighboring spheres. These models differed on average by a DRMS of 0.8 nm from the native structure. Again, only the native structure contained all predicted direct interactions, which would allow us to determine the native structure without knowing the correct answer in advance.

2.4. Discussion

We showed that it is generally possible to determine the subunit packing in assemblies at low-resolution using as sources of spatial information an appropriate representation of the individual subunits, the assembly shape, and only a modest number of subcomplexes (Table 2.2, Figure 2.6, Figure 2.8, Figure 2.10). This goal is achieved by the satisfaction of spatial restraints that depends on a subunit representation, a scoring function, and an optimization (Approach).

Information about the coarse shape of the individual subunits can be provided by several methods, including hydrodynamic experiments [50], small angle x-ray scattering [34], negative-stain or electron cryo-microscopy images [8], and bioinformatics. If such analyses are unavailable, the upper bound on the size can be estimated from the mass of a subcomplex. The shape of the assembly can be characterized by a variety of imaging techniques, such as electron cryo-microscopy and tomography. However, these imaging methods sometimes lack the resolution to provide the subunit configuration. We suggest that complementing these imaging techniques with protein affinity purification experiments may provide a way to bridge the resolution gap between assembly shape and subunit configuration.

In our calculations, we used restraints on five different spatial features, including subunit excluded volume, assembly shape, subunit proximity in a subcomplex (proximity restraint), subunit connectivity in a subcomplex (connectivity restraint), and symmetry. None of these restraint types are sufficient on their own for the accurate determination of the native assembly structure. However, when all of them are integrated into a single scoring function, the correct subunit configuration can be determined. The subcomplex connectivity restraint in particular is especially useful for accurate structure determination (Table 2.2, Figure 2.2 and Figure 2.3). While the subcomplex proximity restraint is helpful, it is not as informative as the connectivity restraint (Table 2.2, Figure 2.2 and Figure 2.3).

Our analysis depends on sufficiently thorough sampling of the subunit configurations that are consistent with all input restraints. However, once a sufficient sampling is achieved, the analysis is independent of the optimization method. The current optimization protocol provides from hundreds to thousands of configurations that satisfy all the restraints derived from most of the subcomplex sets, which we suggest is sufficient for a coarse ranking of the information content of the different restraint sets (Table 2.2a,b). The exceptions are the restraint sets that include subcomplex connectivity restraints for large subcomplexes (subcomplex sets 5 to 8 in Table 2.2c), which result in a combinatorial explosion in the number of possible minimal spanning trees per subcomplex. In principle, this expansion of the search space requires more sampling to find good-scoring solutions. However, we circumvented this problem by constructing additional good scoring structures based on the native structure (Approach).

In the future, testing of our approach could be expanded in a variety of ways. First, we have not exhaustively explored all combinations of different restraint types. For example, we could assess the information content of various combinations of pulldown sizes. Second, we have not yet mapped the accuracy of the structure determination as a function of the error in the simulated restraint sets. This objective can be achieved by using the same approach as described here, except that some error is introduced in the simulated restraints. Third, we did not study ways to minimize the impact of errors in the input restraints. When the fraction of incorrect restraints is small, we expect that it will be possible to identify incorrect restraints by the inability to find models that are consistent with all of the restraints. We could also employ jack-knifing to identify incorrect restraints. Fifth, we will apply our approach to real assemblies with real data. Large scale tandem affinity purification experiments may provide a way to do so.

This study is part of our effort to develop and apply a computational system for enumerating structures of protein assemblies that are consistent with all available information from experimental methods, physical theories, and statistical preferences extracted from biological databases [40, 41]. We are currently introducing structural representations at multiple levels of resolution. This extension will allow us to use pulldown information together with other sources of spatial information, such as density fitting, computational docking, and cross-linking. The resulting integrated system will maximize efficiency, accuracy, resolution, and completeness of the structural coverage of protein assemblies.

2.5. Future Directions

Two of the future directions for this study would be the application of the ideas in this study to more realistic model systems, and to integrate these spatial restraints into a larger context. The both of these directions are already being pursued in Andrej Sali's lab with continued work on the structure of the nuclear pore complex (NPC) and the development of the Integrated Modeling Platform (IMP).

Another future direction would be to develop a practical, systematic procedure that takes as an input an underdetermined system of restraints and iteratively specifies the next most informative experiment. In principle, this could be done by performing a greedy optimization over the information content of the data from each proposed experiment.

3. Protein-Protein Docking using Residue Content Data from NMR Spectroscopy

Macromolecular complexes are fundamental to most biological processes, but their structures are difficult to determine experimentally or predict computationally. Previously, we reported a methodology for using NMR spectroscopy to identify protein binding sites by the combinatorial labeling of selective amino acid residue types. Here, we extend this approach by developing a computational method for determining heterodimeric configurations, given the amino acid residue content of the binding sites and the known or modeled subunit structures. First, we identify the interacting binding sites based on the labeled amino acid residue types by exhaustively sampling each subunit surface. Second, we dock the subunits, restraining the identified binding sites to form an interface. Third, we score each of the resulting configurations by geometric complementarity as well as the difference between the modeled and experimental residue content. Lastly, we obtain the final configuration by a clustering analysis. Our benchmarking demonstrates that (i) a binding site may be successfully identified with as few as 3 labeled residue types, (ii) certain combinations of residue types yield a more accurate and precise configuration of the complex than others, and (iii) the final configuration is generally within 3.5Å rmsd from the native complex. Therefore, the residue content of binding sites can provide sufficient information to determine an accurate configuration of a heterodimeric complex. In addition to improving the structural coverage of complexes, our method also allows us to incorporate other sources of spatial information to model higher-order complexes, and to form hypotheses for further experimental and computational inquiry.

We are grateful to Vincent Voelz and John Chodera for their many suggestions.

3.1. Introduction

Macromolecular complexes perform many important biological functions and provide layers of complexity in biological systems. Elucidating the structure of these complexes might provide a mechanistic picture of their function and a more holistic view of their relationships within a larger systems context. To date, approximately 15% of the protein structures deposited in the Protein Data Bank (PDB) [51] are heteromultimers, and, of these, a large portion are proteins that are not transient in their associations (*e.g.*, antibodies, multi-chain enzymes). This highlights an opportunity for the determination of protein complexes starting from the solved structures of individual domains involved in protein complexes or those that can be accurately modeled from closely related structures [52-54]. It would therefore be useful to have a method that can quickly combine available knowledge of subunit structures with new, easily collectible, experimental data to provide models of macromolecular complexes that can be used to assist in the design of further biological experiments.

Existing approaches for the structural characterization of macromolecular complexes include experimental and computational approaches [55]. While high resolution experimental techniques (*i.e.*, X-ray crystallography, NMR spectroscopy) provide the most accurate characterization of complexes and are being used to solve increasingly larger complexes, they can be very difficult and expensive and are still limited in the sizes of complexes that they can address (especially for NMR). Approaches such as electron microscopy and electron tomography can provide data on larger complexes, but often lack the atomic level of detail of the higher resolution approaches [11, 56]. While purely computational approaches for protein-protein docking [57, 58] do provide atomic level models, they generally lack the accuracy to be reliably used to direct further inquiry.

Hybrid approaches, which integrate multiple sources of data provide a way to increase the coverage and accuracy of structure determination for macromolecular complexes [1]. These approaches range from combining two techniques (*e.g.*, fitting structures or models into density maps from cryo-electron microscopy [59, 60]) to combining many different sources of data (*e.g.*, as used to determine the arrangement of subunits in the nuclear pore complex [35, 61]).

The hybrid method described in this study efficiently determines heterodimeric complexes, using data from NMR spectroscopy and the known or modeled structures of subunits. Previously, we developed a methodology that combines specific isotopic labeling strategies with heteronuclear NMR chemical shift changes to quickly map the binding sites of a protein-protein complex [38]. This NMR method enables the efficient identification of binding sites without prior backbone assignment of the protein of interest.

This goal is achieved by comparing spectra of samples selectively labeled by amino acid residue type in free and complexed forms. Our hybrid method combines the residue content data resulting from the NMR method with the solved or modeled structures of the subunits to inform a docking procedure for heterodimeric configurations. The configurations produced by our method for our benchmark set are within 3.5Ǻ rmsd of the native structures, which is sufficient to inform future experimental studies, for example, to generate further biophysical data that may further improve the structural model, or to predict mutations that may have a specific affect on function.

3.2. Results

Our method consists of four stages (Figure 3.1): first, identifying the interacting binding sites; second, docking the subunits by restraining the identified binding sites; third, scoring each of the resulting configurations; and lastly, selecting the final configuration by a clustering analysis.

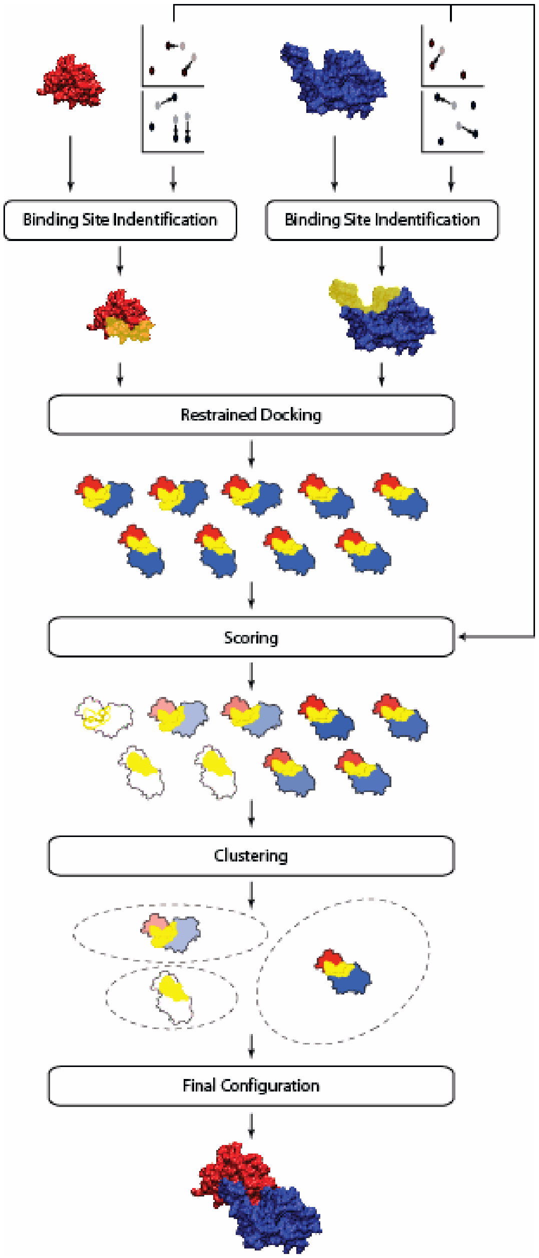

Figure 3.1. Method Flowchart

The inputs to the method are structures (or models) of the two subunits (in red and blue) and chemical shift data for each subunit. After identifying the binding sites on each subunit (region in yellow), we perform restrained docking, generating an ensemble of models. Then, we score each of the models (here using color intensity to represent the hypothetical scores) by their consistency with the geometric restraints and their ability to predict the chemical shift data. Lastly, we cluster the solutions and select the final configuration.

3.2.1. Binding Site Identification

The first goal of our method is to determine as quickly and unambiguously as possible which sites on the protein surfaces are the binding sites. Because the acquisition of NMR experimental data is the most time-consuming part of such work, we sought to maximize the information contained in a given number of labeling experiments, allowing the experimental design to be streamlined. Specifically, we sought the set of residue types that minimizes the uncertainty about the binding site locations. This aim is achieved by searching over the space of possible sets of labeling experiments.

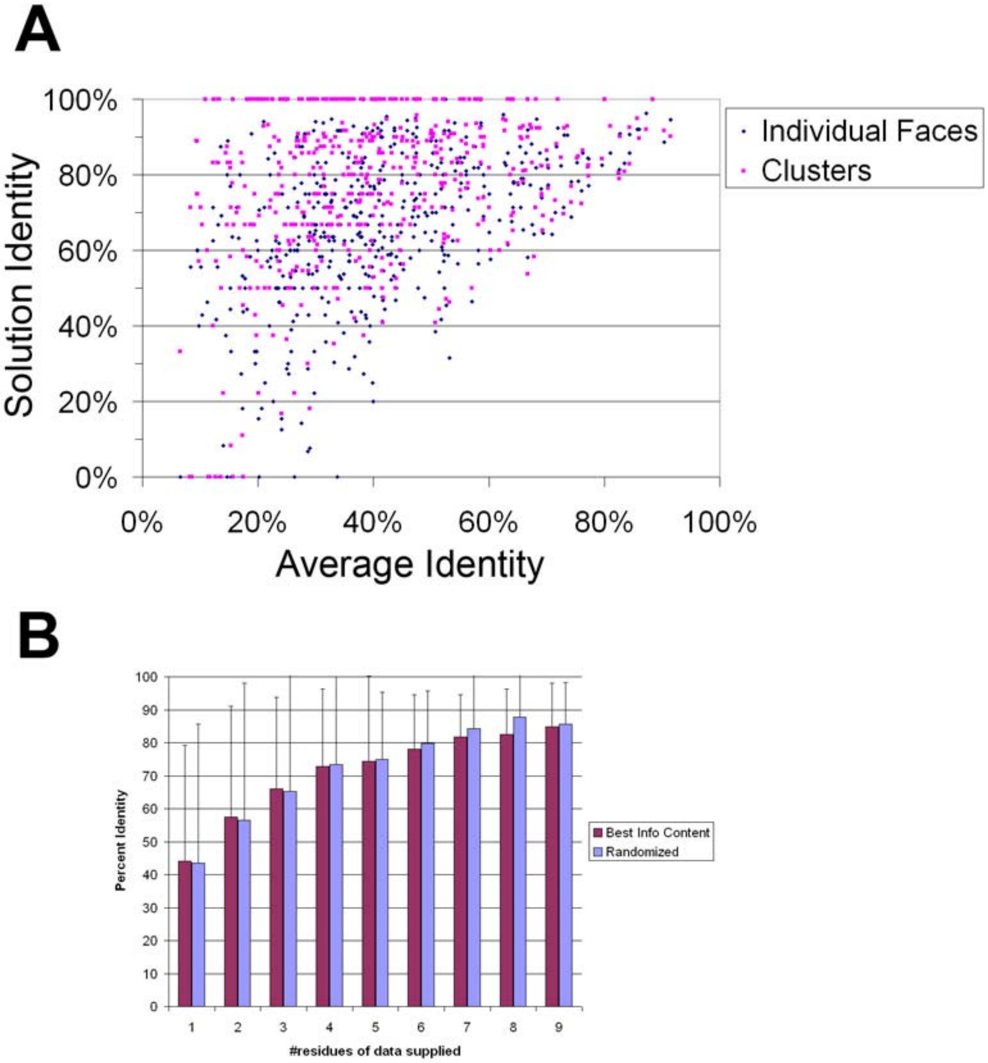

Figure 3.2. Automated Identification of Interfaces

A) The vast majority of the highest-scoring interfaces as determined by our method share >50% identity with the actual interface, and 46% sharing >70% identity. In part, this is due to the robustness of the input information, as the average identity (plotted on the x-axis) of an interface composed of a random combination of residues is often greater than 50%. However, our method usually exceeds what could be expected from simple chance.

B) In many cases, data from three residue-types can yield sufficient restraints to correctly identify an interface. In such cases, optimizing the labeling strategy for potential information content is important, however, the data of a combination of any four or more residues yields remarkably similar results.

Using a database of 500 heterodimeric structures deposited in the PDB, we tested the information content of hypothetical labeling experiments for each of the 9 residue types we previously used in our fast-mapping method (Arg, His, Ile, Leu, Lys, Met, Phe, Tyr, and Val). Once residue content data for 4-5 residue types has been collected, additional experiments yield almost no non-redundant information. Furthermore, any combination of 4-5 residue types generally provides an equivalent reduction in uncertainty, which indicates the robustness of this type of information.

Having theoretically demonstrated that the binding site residue content data should be able to identify binding sites, we sought to develop a method for localizing a binding site using this data. Our binding site localization method exhaustively searches a protein surface for regions that contain the exact number of each residue type given as input and outputs a scored clustering of possible binding sites (Figure 3.1, Figure 3.12). While a simple conceptual problem, identifying clusters of given residue types on an irregular protein surface represents a complex topological problem. We overcame this problem by projecting a given protein's surface to a continuous two-dimensional coordinate system (*i.e.*, a fixed-radius spherical coordinate system). All combinations of residues in a given structure corresponding to the input residue content are searched exhaustively and scored as described in Materials and Methods.

To test our binding site identification method, we created a non-redundant database of the surface information of heterodimeric complexes whose structures have been deposited into the PDB. We then used our binding site localization method on each of 500 interfaces of heterodimeric complexes from our database, and checked the identities of the top-scored binding sites by clustering the solutions from our method and comparing the clusters to the correct binding site (Figure 3.2A). In 75% of the test cases, the top scored cluster of binding sites shared a minimum of 50% identity with the correct binding site, which we took as the minimum value to identify the correct binding site. The average accuracy was 63%, and 46% of the clusters had an accuracy greater than 70%. If the top three scored clusters were examined, the success rate jumped to 90%.

In examining the performance of our binding site-searching algorithm, two special cases are of interest. First, it is possible that the majority of the potential binding sites with a given residue content will share significant identity with the actual binding site. Under these conditions, it becomes impossible to pick an incorrect solution; *i.e.,* the location of the site is completely determined by the residue content. This case is more likely to occur in smaller proteins (<200 residues) which tend to have few (or none) of the more rare residues. However, we see such cases even in larger proteins. For instance, in the interaction of β-lactamase with a protein inhibitor (262 and 273 residues, respectively; PDB: 1JTD), the residue content of the inhibitor's binding site limited the potential sites to $5.5x10^5$, 100% of which shared a minimum 50% identity with the correct site. Not surprisingly, the top predicted cluster shared 90% identity with the correct site. However, the residue content of the β-lactamase chain was much more limiting, with $4.7x10^9$ potential sites, only 1.9% of which shared a minimum 50% identity with the correct site. The top scored cluster predicted by our method shared 63% identity with the correct site (the second scored cluster shared 70% identity).

The second case is the failure of our method to identify a binding site with any overlap with the correct site. While our method successfully identified the correct site in 90% of the tested cases, rarely the residue content did not allow the differentiation of a decoy site from the correct site. For instance, in the structure of cell-cycle dependent kinase 6 with cell-cycle inhibitor p19INK4d (PDB: 1BLX), the top-scored clusters identified a surface of the kinase uninvolved in its interaction with p19INK4d (Figure 3.3). This decoy region of the surface is similar in area and residue content to the correct binding site and thus cannot be distinguished from the correct site by our scoring metric. To differentiate the two sites, we need additional experimental data. Here, our method can guide the design of these additional experiments (*e.g.*, the assignment of one of the residues in the putative site).

To measure the information an individual residue type is providing to our method, we repeated our search on 120 high-scoring test cases, testing the accuracy of the sites predicted with incremental amounts of data, adding the content of each residue-type incrementally, in the order predicted to yield the best solution by our information theoretic model. Consistent with our theoretical model little marginal improvement is realized after the addition of data for 5 or more residue types (Figure 3.2B). Together, these data gave us confidence in our method's ability to predict binary protein interfaces.

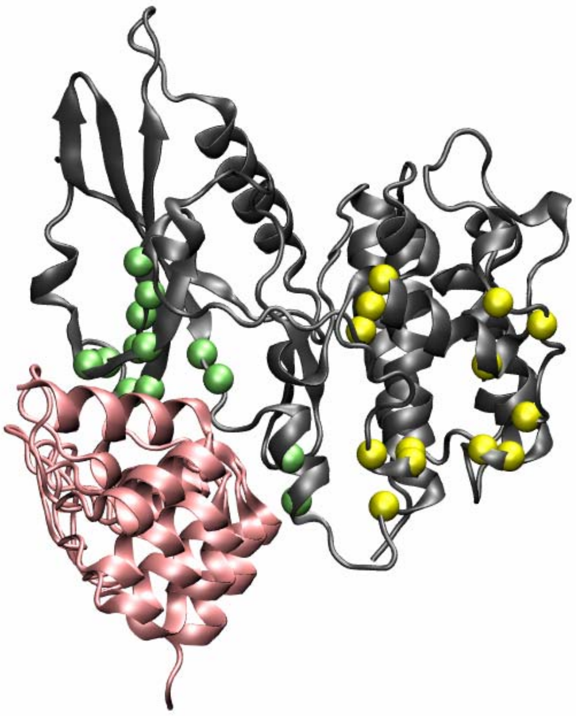

Figure 3.3. cyclin-dependent kinase 6 and p19INK4d

The residue content and surface area of the correct interface (green spheres) between cyclin dependent kinase 6 (grey) and p19INK4d (pink) is similar to the decoy (yellow spheres), making it difficult for our method to determine between the two without additional data.

3.2.2. Restrained Docking

We wished to test whether the top scoring binding sites generated by our surface search could be used as restraints in the docking of two subunit structures. While labeling only a few residue types is an efficient way of identifying the site of an interface, docking two subunits using restraints derived from such sparse data can result in biased configurations, depending on the specific distribution of labeled residues in the binding sites. To solve the problem of docking subunits using restraints derived from incompletely defined binding sites, we restrained models using the geometric centroid of a predicted interface (Figure 3.4). This centroid-driven approach creates a set of restraints evenly distributed over a predicted binding site, minimizing biases that might be introduced by sparse data.

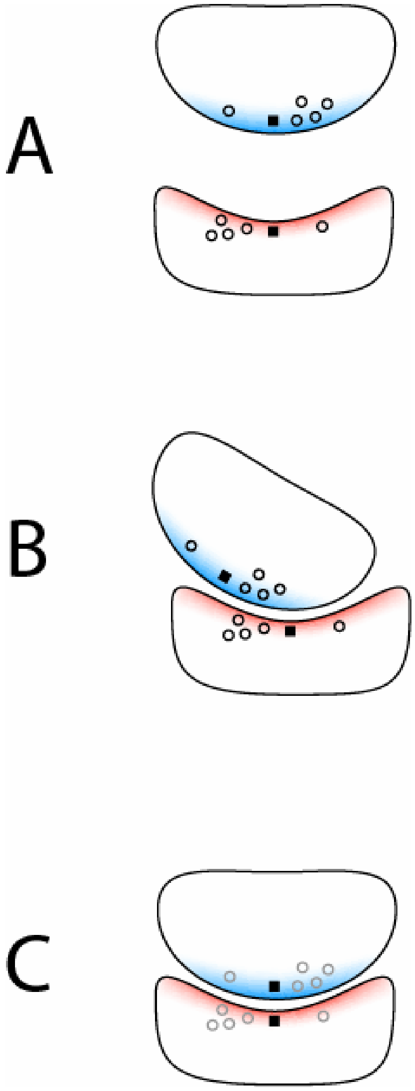

Figure 3.4. Ambiguous Binding Sites and Restrained Docking

A) Because of the sparse labeling data (circles), we necessarily define ambiguous binding sites on each of the subunits (shapes, binding sites in blue and red shading). Here we also label the unweighted geometric centroid of the labeled residues with a filled square.

B) Were we to use only the labeled residues to generate restraints for restrained docking, we would bias the resulting configurations.

C) By including restraints on the centroid of the interface and restraining all of the residues around that centroid, we avoid biasing the final configuration.

Our restraint-based docking method consists of three steps: defining spatial restraints on the two subunits and their binding sites, randomizing the initial positions of the two subunits, and optimizing the relative orientation of the two subunits using the defined spatial restraints (Materials and Methods). Our restraint-based docking method is independent of the particular restraint definition, optimization and scoring program. We implemented our method as a part of IMP (http://salilab.org/imp/).

To avoid compounding errors due to our initial ambiguous identification of the binding sites, for each binding site, rather than using the calculated centroid directly, we use it to search for an optimal centroid. For each subunit, we create an initial set of seed centroids based on the calculated centroid by translating it 1-6Å along the protein surface in various directions, which covers an approximate 36 π Å^2 surface area. For each combination of two initial seed centroids (one for each subunit), we generate 50 independent configurations of the complex and score them by the difference between the modeled and experimental residue content. From the top 500 scoring models, we select the seed centroid with the highest number of top scores and generate 2000 independent configurations for further analysis.

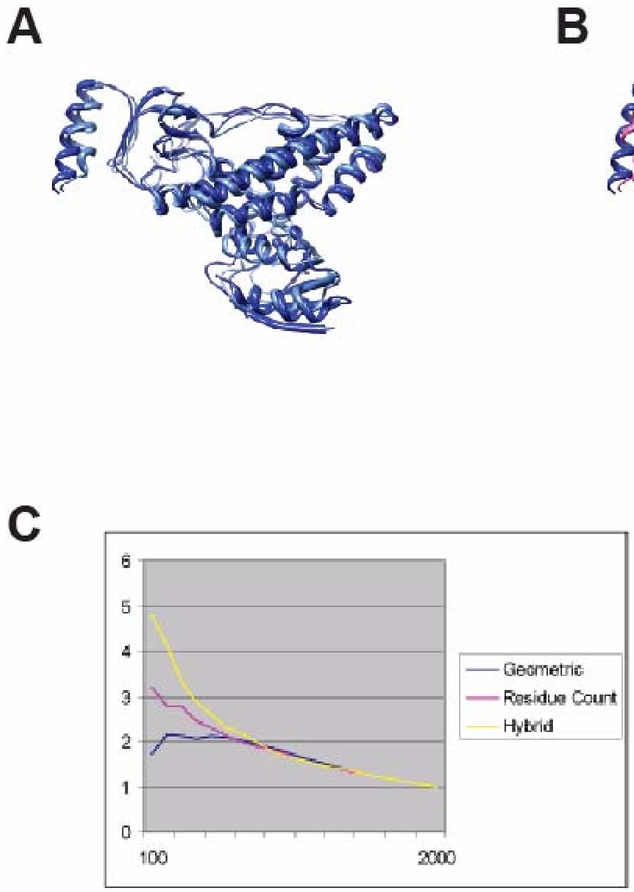

Figure 3.5. 3EZA Centroid-Driven Docking Summary

A) Superposition of the representative native-like solution (light blue) and the native structure (dark blue). The final solution is less than 3.5Å rmsd from the native structure.

B) Superposition of the representative decoy solution (pink) and the native structure (dark blue). The decoy solution is rotated almost 180º from the native structure.

C) Enrichment plot of the three scoring functions.

3.2.3. Model Scoring

To evaluate the configurations generated by docking, we define three scoring functions. The geometric score is determined by the violations of the spatial restraints used to optimize the model. The residue-content score was calculated as the sum of the differences between the experimental and modeled numbers of residues in the complex interfaces (defined here as within 5Å of the opposite subunit, as a conservative estimate of those residues that would undergo chemical shift changes). We combine these two scores into a hybrid score by linearly scaling each score between zero and one, and taking the arithmetic average of the two scores.

PDB	3EZA	1GGR	1BRS	1A0O	1UGH
Yield <3.5Å rmsd	13%	19%	3%	4%	37%
Geometric Score (top 200)	28.5% (2.2)	58.5% (3.0)	0.0% (0.0)	1.5% (0.4)	73.5% (2.0)
Residue Content Score (top 200)	36.5% (2.8)	38.5% (2.0)	17.5% (5.9)	16.0% (4.1)	46.5% (1.2)
Hybrid Score (top 200)	54.5% (4.1)	42.0% (2.2)	14.0% (4.7)	14.5% (3.7)	58.0% (1.6)

Table 3.1. Summary of Test Cases

For each of the five test cases, we summarize the efficiency of the restrained docking method by the yield of configurations <3.5Å rmsd. For each scoring function, we list the percentage of native-like configurations found and the enrichment for native-like configurations, in parentheses.

To validate our approach, we tested our restrained docking on the complex of Enzyme I with HPr (PDB: 3EZA), which could not be accurately modeled directly using the sparse interface restraints (Figure 3.6). Using the structures of the unbound subunits (PDBs: 1ZYM, 1POH), we simulated the collection of fast-mapping data by filtering available chemical shift perturbation data (refs) for the above nine residue types to define the residue content. We used this simulated residue content data with our method as previously described. Then, we scored the configurations with the restraint-based geometric score, and calculated the enrichment of configurations with main chain atoms <3.5Å rmsd of the native structure (Figure 3.5C). Approximately 13% of the configurations produced were within this cutoff, but we achieved only a 2.2-fold enrichment, resulting in 57 native-like configurations (28.5%) among the top 200 configurations using the geometric scoring (Table 3.1). Having demonstrated that the residue content provides sufficient information to localize a binding site, we asked whether a simple scoring function that compared the residue content of a given configuration's interface to the experimental data could differentiate between native and decoy configurations.

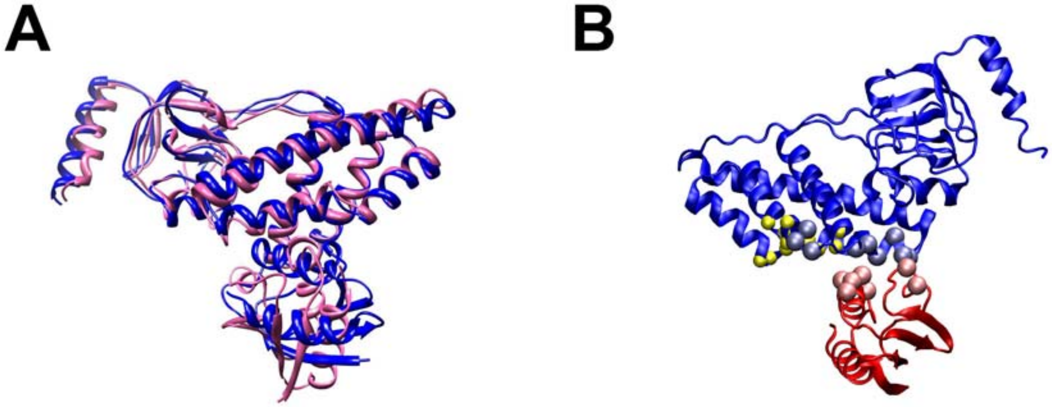

Figure 3.6. 3EZA HADDOCK Docking Summary

A) Running HADDOCK with restraints generated by “ambiguous interfaces” produces only a decoy model (pink), rotated ~90° from the native structure (blue). B) The same decoy, (view rotated 180°) with the residues used as ambiguous restraints highlighted in pink and blue-grey. Residues that were ignored as ambiguous restraints due to a theoretical lack of data (i.e. that could not have been specifically isotopically labeled and measured by fast-mapping) are highlighted in yellow.

Using the residue content score, we were able to achieve a 2.8-fold enrichment, resulting in 73 native-like configurations (36.5%) among the top 200 configurations in the 3EZA case, which was higher than the enrichment found by the geometric score. We then measured the effect of combining the two orthogonal scoring methods. For the 3EZA case, using the hybrid score, we achieved synergistic results, yielding a 4.1-fold enrichment, resulting in 109 (54.5%) among the top 200 configurations. We repeated our method with four other cases (Table 3.1, Figure 3.7, Figure 3.8, Figure 3.9, Figure 3.10). In each case, our centroid-driven docking method was able to produce configurations that closely resembled the native structure.

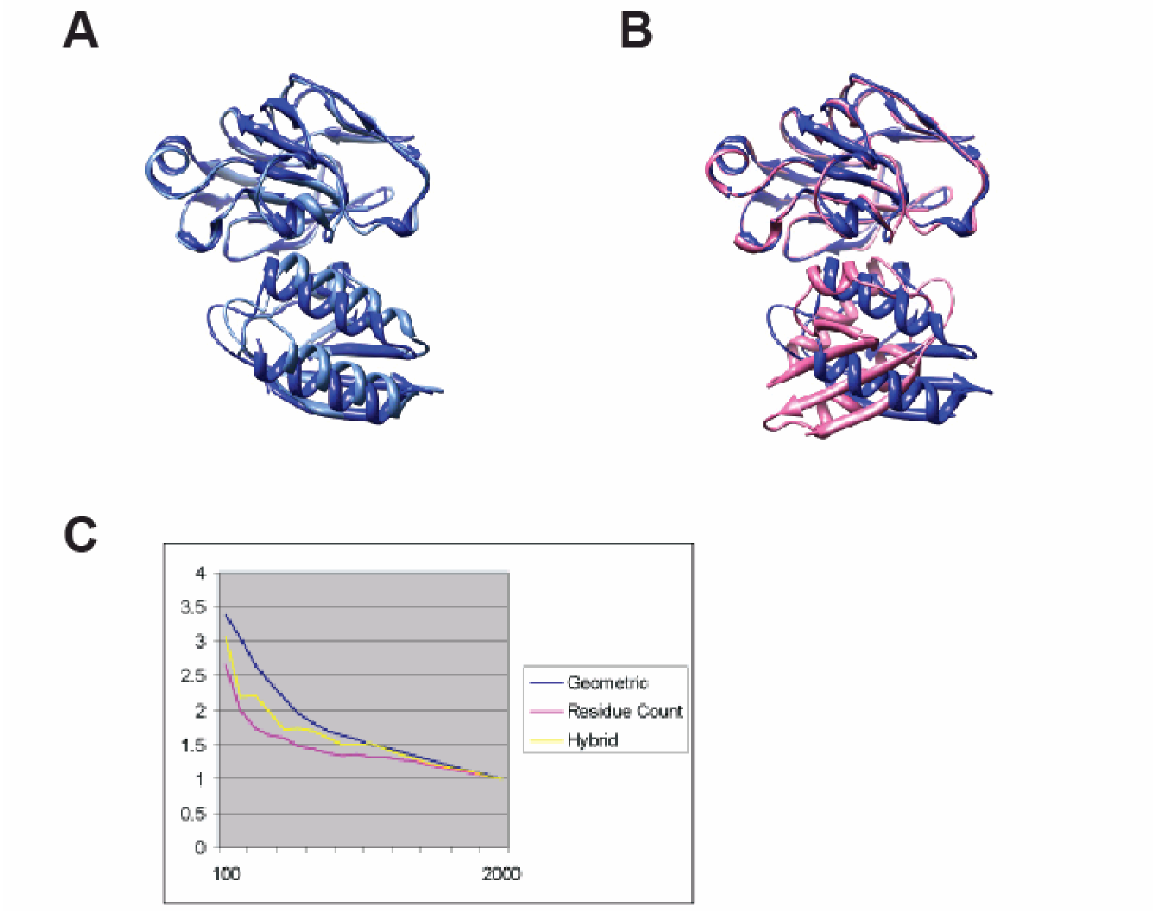

Figure 3.7. 1GGR Centrod-Driven Docking Summary

A) Superposition of the representative native-like solution (light blue) and the native structure (dark blue). The final solution is less than 3.5Å rmsd from the native structure.

B) Superposition of the representative decoy solution (pink) and the native structure (dark blue). The decoy solution is rotated almost 180º from the native structure.

C) Enrichment plot of the three scoring functions.

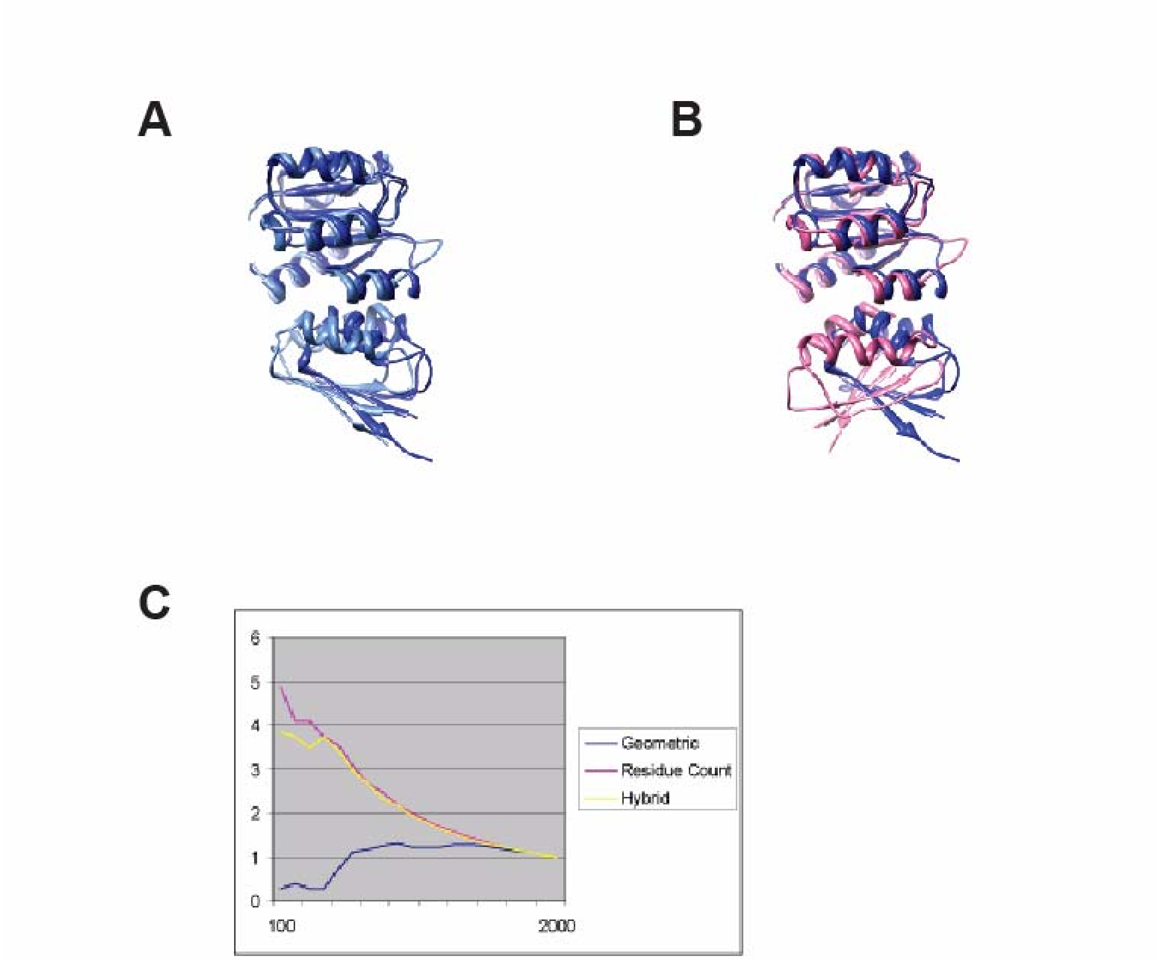

Figure 3.8. 1A0O Centroid-Driven Docking Summary

A) Superposition of the representative native-like solution (light blue) and the native structure (dark blue). The final solution is less than 3.5Å rmsd from the native structure.

B) Superposition of the representative decoy solution (pink) and the native structure (dark blue). The decoy solution is rotated almost 180º from the native structure.

C) Enrichment plot of the three scoring functions.

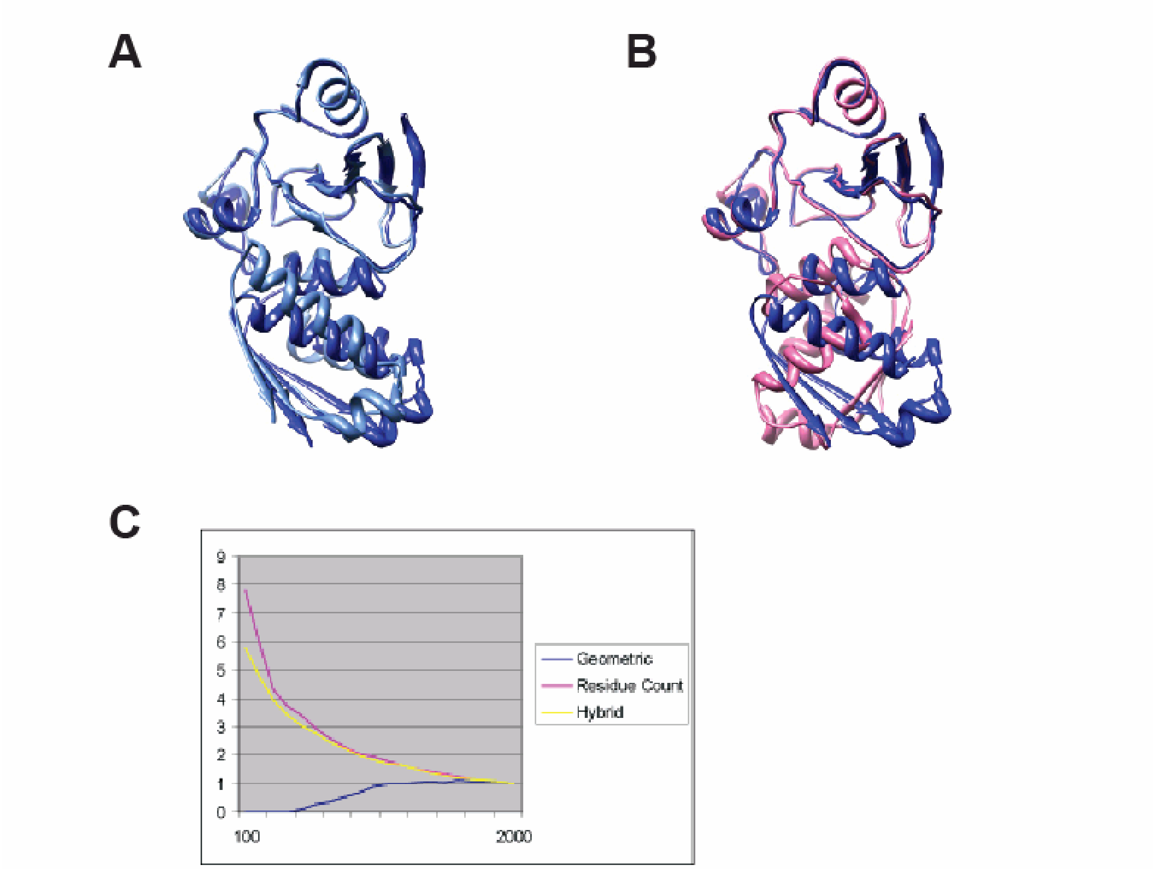

Figure 3.9. 1BRS Centroid-Driven Docking Summary

A) Superposition of the representative native-like solution (light blue) and the native structure (dark blue). The final solution is less than 3.5Å rmsd from the native structure.

B) Superposition of the representative decoy solution (pink) and the native structure (dark blue). The decoy solution is rotated almost 180º from the native structure.

C) Enrichment plot of the three scoring functions.

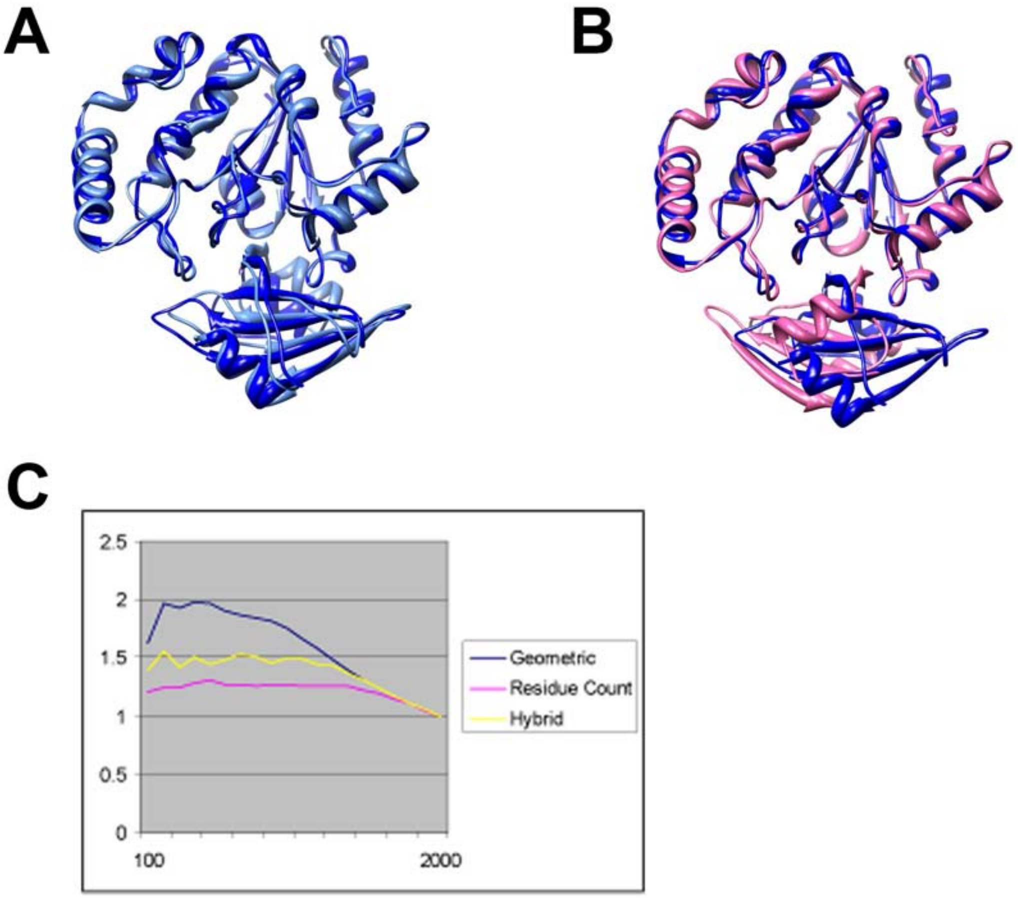

Figure 3.10. 1UGH Centroid-Driven Docking Summary

A) Superposition of the representative native-like solution (light blue) and the native structure (dark blue). The final solution is less than 3.5Å rmsd from the native structure.

B) Superposition of the representative decoy solution (pink) and the native structure (dark blue). The decoy solution is rotated almost 180º from the native structure.

C) Enrichment plot of the three scoring functions.

3.2.4. Clustering Analysis

We also verified that our method produces a sufficiently high concentration of native-like solutions by a clustering analysis (detailed in Materials and Methods). For each test case, we reported the number of native-like configurations and the number of significant clusters (i.e., clusters containing at least 10% of the models) found by clustering the models at four levels: the top-scoring 200, 500, 1000 and all 2000 models (Table 3.2). For the 3EZA case, we see that at each of the levels we have no more than four significant clusters of solutions, and for the top 200 scoring configurations, 124 of them (62.0%) are native-like.

PDB	**top 200**	**top 500**	**top 1000**	**top 2000**
3EZA	124 (2)	219 (4)	381 (4)	812 (3)
1GGR	135 (2)	281 (2)	506 (2)	878 (2)
1BRS	32 (3)	73 (4)	152 (4)	379 (6)
1A0O	36 (3)	106 (3)	182 (3)	356 (4)
1UGH (raw)	70 (3)	197 (4)	422 (4)	501 (4)
1UGH (corrected)	123(2)	297 (3)	605 (3)	790 (4)

Table 3.2. Summary of Clustering Analysis

For each of the five test cases, we summarize the clustering analysis by reporting the number of native-like models and the number of clusters with at least 10% of models, in parentheses.

3.2.5. Integrating Additional Experimental Data

While subsequent refinement of the rigidly docked configurations in explicit solvent allows physics-based methods to better differentiate between native and decoy configurations, even without this more expensive computational step, the set of configurations can suggest future experimental steps for conclusively identifying the correct model. For example, in the 3EZA test case, clustering the top 10% of the models reveals 4 clusters, of which only the top two are significantly populated (the native-like cluster and a decoy cluster with 62% and 25% of the models, respectively). By measuring the distance between the side-chain hydroxyl atoms of S210 of chain A and T30 of chain B, we could decisively distinguish which of the two clusters was the correct solution (Figure 3.11A). Measuring this distance in the native structure yielded a distance of 51Å. We repeated our docking with this single additional distance restraint, together with a 5% error, to simulate its measurement by a technique such as fluorescent resonance energy transfer. The resulting solutions find that only the first, native-like, cluster remains significantly populated (Figure 3.11B). This result is consistent with our experience in modeling the interface between PSD-95 and microtubule associated protein 1a, in which our fast-mapping data, combined with 2 upper-bound distance restraints derived from paramagnetic relaxation enhancement data provided sufficient restraints to generate models populating only a single structure [37].

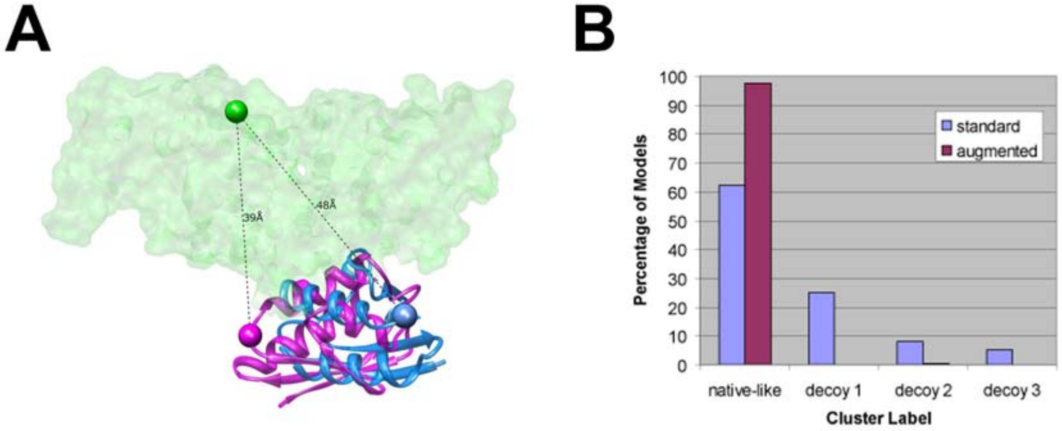

Figure 3.11. Adding One Unambiguous Distance Restraint

A) By adding a single long-range distance restraint between two atoms to our centroid-driven docking we were able to test whether this could discern between the decoy (pink) and native-like (blue) solutions. Here we show the value of the native-like distance (between green and blue spheres) and the decoy distance (between the green and pink spheres).

B) Such a docking strategy yields a single dominant cluster of native-like solutions. Where the standard (in blue) docking procedure has four significantly populated clusters and the augmented (in red) docking procedure only has the native-like cluster remaining.

3.3. Discussion

Our benchmarking of binding site identification demonstrated that our method can successfully identify a binding site with as few as 3 labeled residue types, and that certain combinations of residue types yield a more accurate and precise configuration of the complex than others. As expected, the predicted information content of the labeling of a specific amino acid correlates with its surface distribution on the molecule. Those residues that are more common but unevenly distributed yield greater potential information about the location of an interface on the surface of the protein. This observation suggests that residues that are common, likely to be surface exposed, but less likely to pack in protein-protein interfaces would be the ideal candidates for our mapping experiments. However, while the interface-content of such residues as serine, threonine, asparagine, and glutamine potentially may provide the most information, these residues are difficult to isotopically label using conventional methods due to metabolic "scrambling" of the label. However, such difficulties are circumvented by cell-free protein synthesis methods [62], greatly increasing the number of potential residue-specific isotopic labeling strategies. Still, as we have demonstrated that any three to six experiments can positively identify an interface (Figure 3.2B), choosing a labeling strategy that yields the greatest amount of information becomes less important.

Our benchmarking of the restrained docking demonstrated that our method can efficiently produce a final configuration within 3.5Å rmsd from the native complex. We also demonstrated that the residue content of an interface not only may be used to identify binding sites, but provides a useful scoring metric in the evaluation of computational docking of protein-protein complexes. Also, as this residue content data is available in most experimentally restrained docking, but is currently discarded, incorporating residue content data can provide an additional potential crosscheck between a model and the experimental data. Finally, we demonstrated that the results from our method are of sufficient accuracy to serve as the starting point for future experiments that can completely differentiate between native and decoy configurations.

As the experimental fast-mapping data may be collected very quickly, and require very little analysis (*i.e.,* no assignments), our method provides an attractive path to structure-based experimental design. While most methods are applied to stable complexes, our method can also be used to address the more difficult case of transient associations of macromolecules. Also with recent advances in NMR, such as methyl-TROSY [63], the molecular weight limit of the models that may be studied has drastically increased, further extending the coverage of our method. In practice, the major issue often becomes spectral dispersion, which our specific labeling strategy avoids. We estimate that our methodology may be useful in characterizing complexes whose subunits are under 100kDa. Examining of PIBASE [64], a database of the complexes represented in the PDB, we find that 897 complexes, or 98.7% of the non-redundant protein-protein complexes catalogued, meet these criteria.

As structural biology proceeds further into the post-genomic era, approaches to quickly provide structural details of macromolecular complexes are becoming increasingly necessary in the study of complex biological systems. While crystallography yields a picture of a structure in an all-or-nothing manner, NMR can provide incremental information that can be integrated with other structural data. It is this hybrid approach that allows us to efficiently and accurately combine available knowledge of subunit structures with new, easily collectible, experimental data to provide models of macromolecular complexes that allows us to improve the structural coverage of complexes, to model higher-order complexes, and to form hypotheses for further experimental and computational inquiry.

3.4. Materials and Methods

3.4.1. Binding Site Localization from Residue Content Data

The initial input is a file to our binding site localization method contains the coordinates of an unbound protein structure in PDB format. Using XPLOR [65] to parse the structure, we created a list of those residues that are both in close proximity to one another (<3.5Å) and surface exposed. The graph of contacts is then searched breadth-wise to create a table of pairwise "paths" along the surface between residue combinations, with the length of each path considered as the integral number of separating neighbor residues.

The surface exposed residues in the structure are projected onto a sphere of radius $p_{max}/2\pi$ (where p_{max} is the maximum path length in the above table), using rays from the protein's center of mass. The residues' positions on the sphere are then minimized, using the lengths of the paths in the pairwise table as restraints. This procedure creates a simplified representation of the surface while preserving the relative positions of residues on the protein surface.

A list of potential interfaces is created by merging single-residue combinations based on the experimental data. For instance, given an initial dataset indicating the presence of 1 of 4 Lys, 2 of 4 Tyr, 2 of 6 Phe, and 4 of 12 Leu in the interface, every possible combination of surface-exposed residues containing the correct residue content will be created. That is, merging the potential 4 Lys, 6 Tyr, 15 Phe, and 495 Leu interfaces results in 178,200 potential interfaces. Each of these potential interfaces is scored by the likelihood that it contains the binding surface that yielded the initial chemical shift perturbation data. We consider the most likely binding surface to contain the correct residue content in the smallest area on the surface of the protein. In our spherical representation of the surface, this smallest found area corresponds to the “interface” circumscribed by a polygon with the smallest surface area. The score is further refined by penalizing those potential interfaces whose circumscribed polygon contains additional residues (*i.e.,* areas of the surface whose residue content does not match the experimental input). The potential interfaces are then hierarchically clustered so that a given cluster’s members all share a minimum of 75% identity of residues. Lastly, the clusters are scored as the average of their members.

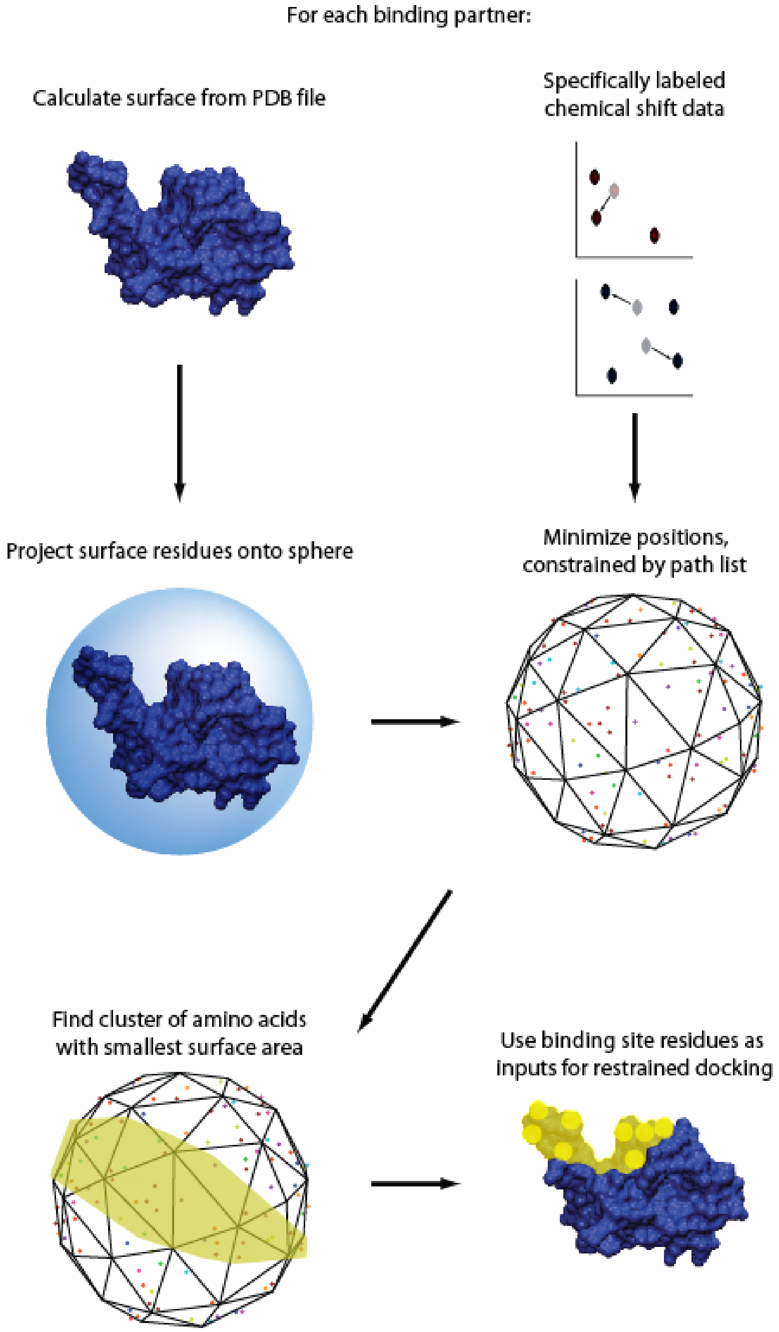
For each binding partner:
Calculate surface from PDB file
Specifically labeled chemical shift data
Project surface residues onto sphere
Minimize positions, constrained by path list
Find cluster of amino acids with smallest surface area
Use binding site residues as inputs for restrained docking

Figure 3.12. Binding Site Identification

For each subunit (in blue), we project the surface residues onto a sphere and then use the chemical shift data along with this spherical projection to identify the smallest cluster of amino acids on the surface (shaded region) which allows us to identify the binding site residues.

3.4.2. Restrained Docking

We employ the restraint definition and optimization capabilities of IMP and MODELLER to perform the generation and geometric scoring of docking solutions. As inputs we take the atomic coordinates for all of the heavy atoms of both subunits, and the list of residues types used for the scan. Then for each interface centroid seed, we determine the set of residues surrounding the centroid seed to define as interface residues for use in our docking calculations. In the modeling step, we determine the surface accessible residues using the PSA method with a cutoff of 8.0 [66, 67]. We define r_{min} and r_{max} as the respective distances between the center of mass and the interface centroid of the smaller and larger proteins.

To determine the interface residues we take all of the residues within a sphere of radius r around the centroid coordinate that are defined as surface accessible. We determine r by this formula: $r = r_{\min}\left(1+\left(1-\left(\frac{r_{\min}^2}{r_{ave}^2}\right)\right)\Big/2\right)$, where r_{ave} is the average of r_{min} and r_{max}.

Then we define our restraints. First, we define each protein to be a rigid body. Second, we restrain the mass centers to have an upper bound at r_{min}+r_{max}+10.0Å with a standard deviation of 1/10 of this value. We also restrain the centroids with a Gaussian upper distance restraint of 10.0Å with a standard deviation of 1 Å. To restrain the putative interfaces together, we employ a minimal distance restraint between the two interfaces using a Gaussian distance restraint with a mean of 5Å, and a standard deviation of 0.1Å. To enforce excluded volume, we impose a harmonic penalty if the distance between any two atoms is smaller than the sum of their radii scaled by 90%.

For each independently generated model, we allow the two proteins to relax from the solved position by performing 50 steps of minimization by conjugate gradients, a 50 K molecular dynamics run for 50 iterations and then 50 more steps of minimization by conjugate gradients. After this initial relaxation, we fix the backbone of the two subunits and allow the sidechains to relax further by performing 100 steps of minimization by conjugate gradients, a 100 K molecular dynamics run for 100 iterations and then 100 more steps of minimization by conjugate gradients. Then, we randomize the positions and orientations of the two proteins within a cubic volume of $200x200x200Å^3$. For the initial optimization, we employ only the restraints above using 50 steps of minimization by conjugate gradients, followed by 50 iterations of molecular dynamics at 100 K, followed by 50 more steps of minimization by conjugate gradients. This initial optimization brings the two subunits together and roughly places them into the correct orientation (*i.e.,* interfaces facing each other). Then, we do three rounds of simulated annealing. Each annealing round consists of a 10000 K molecular dynamics run for 200 iterations, a 6000 K molecular dynamics run for 200 iterations, and a 2000 K molecular dynamics run for 200 iterations, with 100 steps of minimization by conjugate gradients following each molecular dynamics run. We finish the annealing step with the same minimization protocol used to bring the subunits together. At the conclusion of this step, we have the final relative orientation for the two subunits.

3.4.3. Model Scoring

For the final scoring step, we redefine the set of residues defined as interface residues as follows: we use a less stringent surface accessibility criterion (cutoff = 2.0) for the PSA method, and include the same radius around the centroid coordinates. Using this modified definition of the interface, we calculate the violations to the minimal distance restraint and the excluded volume penalty and report this as the geometric score of the model. We also calculate the residue content of the interface, and define a residue content score as the difference between the observed residue content and the actual residue content.

$$residueContentScore = \sum_{i}^{residueTypes} \left| observed_i - actual_i \right|$$

Finally we output the model with these two scores.

Using this procedure, we first perform a broad sampling over all of the interface centroid seeds from the seeding step. We then identify the best scoring seeds based on the normalized residue content score, and perform a more thorough sampling over this smaller set of seeds.

After generating this final ensemble of models, we calculate the enrichment of native-like hits (as defined by 3.5Å rmsd after least-squares superposition) for three scoring functions: the geometric score alone (soft-sphere overlap and minimal distance restraints), the residue content score alone (an orthogonal experimentally derived measure of interface quality), and the hybrid score, defined as the average of the two normalized scores. Enrichment, for a given level *k*, is defined by the number of true hits within the top *k* scoring models divided by the number of hits expected at random within *k* models.

$$Enrichment(k) = \frac{TrueHits(k) * TotalModels}{k * TotalHits}$$

For each scoring function, we can generate an enrichment curve by calculating the enrichment for native-like hits at multiple levels.

3.4.4. Clustering Analysis

From the final ensemble of models generated by independent optimizations, we perform a clustering analysis on the top scoring 10%, 25%, 50% and the entire set of models (top 200, top 500, top 1000, and all 2000) as scored by the hybrid score. We employ an adaptive distance cutoff covering algorithm for clustering for its efficiency and because we are only interested in the identifying the dominant populations of models within the ensemble and do not know *a priori* the number of clusters to expect.

As inputs to the covering algorithm, we start with a list of models sorted by hybrid score and an all-by-all rmsd table. We initialize the procedure by placing the first model (top-scoring model) into its own cluster. We calculate the threshold distance for that cluster as the one standard deviation less than the mean distance from the first model and all the other models. For the all of the remaining models, we determine the closest model which already-considered model the current model is closest to. If the distance from the current model to the nearest already-considered model is within the distance cutoff for the cluster containing that model, we add the current model to that cluster and update the distance cluster for that cluster by calculating the average distance cutoff for the models in the cluster. If the current model does not fall within the distance cutoff for the cluster containing the model it is closest to, we form a new cluster, using the current model to calculate the distance cutoff for that new cluster.

3.4.5. Information Content of Residue Content of a Binding Site

The first goal of our method is to determine as quickly and unambiguously as possible which of the possible binding sites on the protein surface is the true binding site. As the acquisition of experimental data is the most time-consuming part of such work, we sought to define the amount of information a series of labeling experiments gives about a binding site, allowing the experimental design to be streamlined. This can be posed as an information theoretic question: We are seeking the set of experiments that minimizes the uncertainty over the possible binding sites, by selecting the set of labeling experiments that has the lowest mutual information between them, *i.e.*, each experiment should give us as much non-redundant information as possible.
Let *n* be the total number of possible sites on the surface. The total uncertainty on the surface, *H(N)*, is a function of the number of possible sites:

$$H(N) = \log_2 n \qquad (3.1)$$

For a given set of labeling experiments, *R*, the uncertainty over the surface is a function of the distribution of residue count signatures.

$$H(N \mid R) = -\sum_i p_i \log_2 p_i \qquad (3.2)$$

Where we can determine the probability of a particular signature, p_i, by scanning over all possible sites, counting the number of times that particular signature occurs and dividing by the total number of sites.

The mutual information between the total uncertainty and the uncertainty given a particular set of labeling experiments is given by:

$$I(N : R) = H(N) - H(N \mid R) \qquad (3.3)$$

And this is the quantity that we seek to minimize, with the selection of an optimal set of labeling experiments. The size of the search space is given by 2^r, where r is the number of residues that might be labeled by experiment.

Given the structure of a protein, we perform the information theoretic calculation as follows. First for each residue on the surface of the subunit, we define a binding site centered on that residue by including the first four contact shells of residues on the surface. This yields a set of possible binding sites, from which we extract the possible labeling experiment data. The result is a set of vectors representing the possible binding sites on the surface. We determine the order of labeling experiments by a greedy optimization, selecting the experiment that minimizes the uncertainty by the greatest amount at each step. We also verified that this greedy optimization corresponds well to the global optimal solution.

3.5. Future Directions

The future directions for this study would be the continued application of the ideas in this study to other systems, and to integrate these spatial restraints into a larger context. The latter direction is already being pursued in Andrej Sali's lab with the continued development of the Integrated Modeling Platform (IMP).

4. Modular Assessment of Protein Structure Modeling Methods

Assessing a protein structure modeling method by reporting its accuracy over a benchmark set has at least two limitations: (i) the reported accuracy does not, by itself, provide insight into how a method could be improved and (ii) the reported accuracy often depends on the specific benchmark set, making a fair comparison between methods that do not share a benchmark set difficult. To address these two limitations, we propose a three step framework for assessing structure modeling methods. First, we decompose the method into modules corresponding to possible improvement areas. Second, for each module, we build a set of enhanced modules (*i.e.*, oracles) that can provide as much accuracy as we desire. Third, we test the method by substituting oracles for the various modules, and measuring the differences in accuracy between the current method and the various oracle-assisted methods. We demonstrate this framework on a method for hierarchical docking (*i.e.*, modeling protein assemblies by combining interacting, pairwise dockings). Where a traditional assessment might only report the current accuracy of the method, our framework also revealed that the method's primary bottleneck is the accuracy of the pairwise dockings, and not other factors (*e.g.*, the amount of sampling). Moreover, our framework also shows how the accuracy of the pairwise dockings is related to the overall accuracy of the method, demonstrating potential synergy among the pairwise dockings. Thus, by providing a more detailed description of a method's limitations, our framework facilitates further development and improves the confidence with which we use the generated models.

We are grateful to David Eramian, Libusha Kelly, Daniel Russel, Keren Lasker, Eswar Narayanan, Javier Velazquez-Muriel, Min-yi Shen, and Jeremy Phillips for many discussions about hierarchical docking and assessment methods for computational structural biology.

4.1. Introduction

Structural characterization of proteins and their assemblies provides insights into their function. As a result, increasing numbers of protein structures are being determined experimentally [51, 68]. In addition, there is a proliferation of computational protein structure prediction methods, aiming to be more general, faster, and cheaper than experimental methods. These computational methods include *ab initio* [69, 70] and comparative approaches [52, 54] as well as hybrid methods that integrate multiple types of data for modeling proteins and their assemblies [2, 35]. Several of these hybrid methods rely on experimentally observed data from X-ray crystallography, NMR spectroscopy, and electron microscopy [35, 55, 56, 71]. In particular, hybrid methods increase the coverage and accuracy of structure determination for macromolecular complexes [1]. While these methods span different resolutions, sizes, and types of targets, all of them require a quantitative assessment of modeling accuracy and an unbiased basis for comparison against competing methods.

The methods for modeling of protein structures and protein-protein interaction modes are prominently assessed at the Critical Assessment of Techniques for Protein Structure Prediction (CASP) [72] and the Critical Assessment of Prediction of Interactions (CAPRI) [58] meetings, respectively. These meetings highlight the improvements made and challenges remaining, They have also have provided detailed comparative assessments of various approaches and motivated the development of methods for predicting errors in models [73-75]. Automated protein structure prediction servers are also assessed both biennially as a part of the CASP effort [76] and in an automatic and continuous fashion by assessment servers such as EVA [77].

While these approaches provide a quantitative measure of the prediction accuracy and the relative strengths of competing methods, they also reveal two opportunities for providing more detailed and informative assessments. First, the accuracy reported by these techniques does not, by itself, provide insight into how a method could be improved nor does it specifically identify why the performance of competing methods differ. Second, by using the same benchmark set of newly determined structures, these assessment techniques avoid the problem of using different benchmarks for different methods, which is frequently the case in publications. In the case of differing benchmark sets, it becomes difficult to provide an unbiased view of the relative strengths of competing methods. While reporting the accuracy of a method against a favorable benchmark set will provide an optimistic view of the improvements made, it will also obscure opportunities for improvements and any underlying reasons for poorer accuracy against another method's benchmark set.

By addressing these two opportunities, we will gain a better appreciation of a method's limitations, in turn facilitating further development and improving the confidence with which we use the resulting models [78, 79]. To do so, we propose a framework for assessing modeling methods that we call modular assessment. Next, we provide an overview of the framework, describe an example modeling method, and demonstrate modular assessment on the example method (Results). Then, we discuss our findings, expanding on the implications of and possible extensions to our framework (Discussion). Lastly, we present the specific details of the example modeling method and assessment framework (Materials and Methods).

4.2. Results

While our assessment framework is generally applicable to any modeling problem, for clarity and as a demonstration of feasibility, we describe the three steps of modular assessment by applying them to a method for hierarchical protein docking (*i.e.,* the modeling of protein assemblies by combining pairwise docking results [55, 80, 81]).

4.3. Hierarchical Docking

With the increasing number of experimentally determined single domain structures and with the improvements in comparative modeling, many more protein subunit structures are available for use as starting points for protein-protein docking. Protein-protein docking methods aim to predict the configuration (*i.e.*, the relative orientation) of two protein subunit structures. While accurate computational protein-protein docking of two subunits is still an unsolved problem, we can, in principle, model the assembly of more than two protein subunits by hierarchical docking. Hierarchical docking methods combine docking results for interacting pairs of proteins into progressively larger “subcomplexes”, and eventually, into configurations of the entire assembly. By combining the pairwise docking results in a hierarchical fashion, the space of possible configurations is reduced. This hierarchical approach may also create a synergistic effect between the pairwise docking results, where the selection of a particular configuration of one subcomplex influences the configuration of other subcomplexes within the assembly.

For hierarchical docking methods, we must describe the representation of the subunits, the inputs to the method, the sampling or optimization procedure used to produce models, and the scoring process used to rank the resulting models. In general, we choose a representation of the subunits that corresponds to the resolution of the results of the underlying pairwise docking methods. We summarize the inputs to any hierarchical docking method (*i.e.*, the outputs from the underlying pairwise dockings) as a ranked and scored set of relative orientations for each of the interacting pairs of subunits. With these sets of relative orientations, there are many possible strategies for combining them into higher order complexes, but all of these strategies share the common goal of selecting the most native-like pairwise relative orientation for each interacting pair. Finally, the scoring of models is usually the combination of terms for individual pairwise interactions and higher-order terms for complexes.

From this description of hierarchical docking, we reduce methods for hierarchical docking into the following three modules (Figure 1). First, a data module encapsulates the inputs to the hierarchical docking method. Within this data module, we can represent pairwise dockings consisting of varying degrees of completeness (*i.e.* whether the pairwise dockings are sufficient to determine a unique best scoring solution), accuracy (*i.e.* whether the unique solution is the native structure), and precision (*i.e.* whether the accurate pairwise dockings are selected with sufficient probability to determine the native structure). Second, an integration module specifies how the inputs from the data module are combined to produce an ensemble of scored models. Third, a ranking module ranks the resulting models from the integration module. Paired with an integration module that generates models by the optimization of a scoring function, the ranking module sorts the ensemble of models by their score.

Hierarchical docking is performed by the satisfaction of spatial restraints derived from pairwise dockings (Materials and Methods). The data module consists of the top 20 scoring pairwise dockings from PATCHDOCK [82, 83], for each pair of interacting subunits within the complex, which are transformed into a uniformly weighted list of spatial restraints. The integration module, implemented in MODELLER [46, 84], generates an ensemble of assembly models by optimizing the spatial restraints from the data module. Finally, the ranking module ranks the models by a sum of the pairwise scores.

Having decomposed hierarchical docking methods into three modules, we introduce the key idea of our assessment approach: substituting enhanced modules (*i.e.*, oracles) for each of the modules to improve the accuracy of the method using perfect information. This perfect information is derived directly from the native structure of the targets being modeled. By swapping in oracles for each module and comparing the oracle-assisted performances against the unassisted (*i.e.*, baseline) performance, we can measure the specific contribution each module makes towards the overall accuracy of the method. Also, each oracle-assisted method determines the accuracy ceiling (*i.e.*, the best possible accuracy) of the method if improvements were made to the particular module being studied. Next, we describe the various oracles for hierarchical docking and the associated tests we can perform.

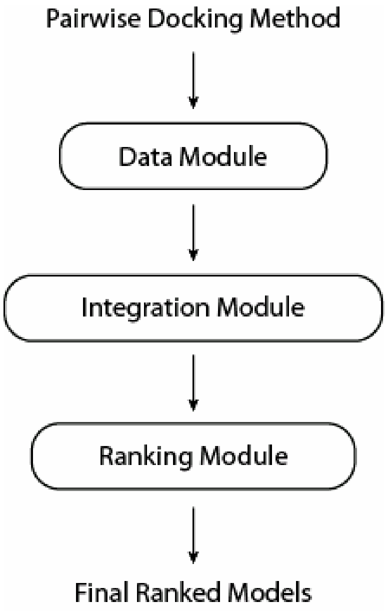

Figure 4.1. Modules for Hierarchical Docking Methods

Overview of modules for hierarchical docking methods. The inputs to the data module are the outputs from the pairwise docking methods. The lists of pairwise configurations are then integrated together by the integration module to produce ensembles of models. This ensemble of models is than ranked by the ranking module, usually by some scoring function, to produce the final list of ranked models.

4.3.1. Data Oracles

Many methods are hampered by incomplete, erroneous or imprecise inputs. A data oracle could provide, in the limit, a complete set of error-free, precise input data. We can also vary the power of the data oracle by limiting the amount of error-free data it provides, or by limiting the amount of noise it reduces. By varying the power of the data oracle, we can determine if the method has sufficient information in the underlying data (*i.e.* the completeness of the data), and the robustness of the method with respect to erroneous or noisy data.

For the assessed method, we created the following three data oracles that we can apply per pairwise interaction. The "native-added" data oracle increases the accuracy of the input data without improving the precision by adding the native pairwise configuration to the list of 20 top-scoring configurations generated by PATCHDOCK. The "perfectly-ranked" data oracle increases the accuracy and precision of the input data by sorting all of the configurations generated by PATCHDOCK by their distance to the native pairwise configuration and choosing the 20 closest configurations. The "perfectly-weighted" data oracle increases the precision even further by replacing the uniform weighting of the pairwise configurations by a weighting based on the distance from the native pairwise configuration.

4.3.2. Integration Oracle

Depending on the quality of the underlying data, it may not be possible to uniquely determine the correct native structure (*i.e.* recapitulate the structure as determined by x-ray crystallography). However, any method should generate as near-to-native structures as possible with the data at hand. Given any level of completeness and noisiness of the underlying data, an integration oracle employs the data as optimally as possible, generating models that are closest to native. By employing an integration oracle, with and without a data oracle, we can determine the accuracy ceiling and separation for the method with respect to any level of data completeness, accuracy and precision. For the assessed method, the integration oracle generates models by performing an exhaustive combinatorial search over the possible pairwise configurations.

4.3.3. Ranking Oracle

Given an ensemble of models generated by the method, a ranking oracle will correctly rank the models. The use of a ranking oracle alone provides a ceiling on a method's unassisted accuracy. The separation between the unassisted method and the ranking oracle-assisted method can be reported by enrichment of top hits, or more stringently by the sum-square difference between the outputted rankings. For the assessed method, the ranking oracle sorts all of the models by their distance to the target.

4.4. Modular Assessment

We performed our assessment on a test benchmark set of 6 assemblies (Table 4.1). For each of the test cases, we started by reporting the accuracy of the unassisted method against the benchmark set. We determined the unassisted accuracy of the method by generating 10,000 models from independent initial conditions, reporting the average percent-overlap (%-overlap) of mainchain atoms at 3.5Å, best-found %-overlap, and top-scoring %-overlap. We report %-overlap, because other measures (*e.g.* root-mean-square-deviation or distance-root-mean-square) are more affected by the different sizes of the benchmark complexes, making comparisons across the benchmark set difficult. While exhaustive sampling of these systems of restraints is computational infeasible, we demonstrate that the level of sampling that we used here is sufficient, as the accuracy of the best sampled model does not significantly increase with more sampling (Figure 2). For the unassisted method, we generated 50,000 models and found an asymptotic relationship between the accuracy of the best sampled model and the number of independent optimizations. In particular, 80% of the final improvement was achieved using ~2,000 independently generated models and 93.5% of the final improvement was achieved by 10,000 models.

		Number of	Number of	Number of
PDB code	Name	subunits	interactions	residues
1bz0	HEMOGLOBIN A	4	4	574
1h4l	CDK5-P25 (NCK5A) COMPLEX	4	3	850
1ikn	I-KAPPA-B-ALPHA / NF-KAPPA-B COMPLEX	3	2	612
1ivo	HUMAN EFG / EFGR COMPLEX	4	3	1116
1ss8	GROEL	7	7	3668
1vcb	VHL-ELONGIN C-ELONGIN B	3	2	328

Table 4.1. List of Target Complexes

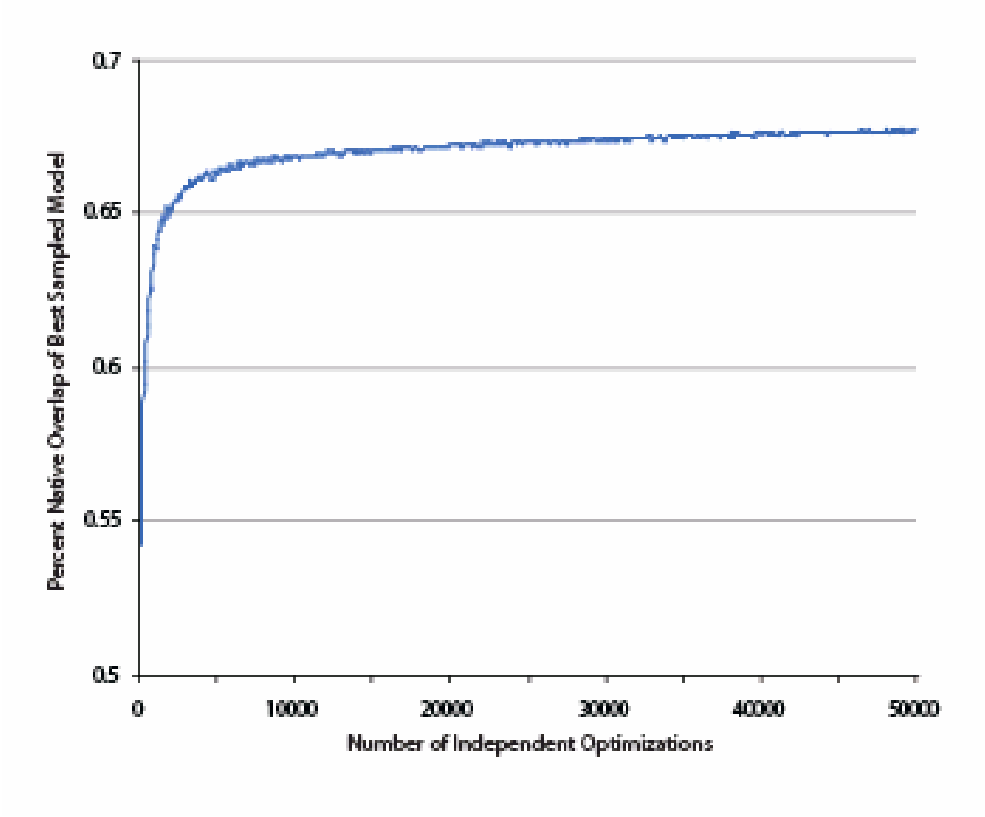

Figure 4.2. Sampling Sufficiency

We plot the quality of best-sampled model, as measured by percent-overlap with the native structure, with respect to the number of independent optimizations.

For hierarchical docking by the satisfaction of spatial restraints generated by a pairwise docking, we found an unassisted average accuracy across our benchmark set of 37.0% average %-overlap and 52.2% best-found %-overlap. Because of the high degree of degeneracy in the scoring function, we report the average %-overlap for all of the models with the same top score. For the unassisted case, we found 39.4% top-scoring %-overlap (Table 4.2). From this baseline, we used the above described oracles, alone and in combinations, and compared the results with the unassisted method.

Test Case	Average	Best-Found	Best-Scoring
unassisted	37.0	52.2	39.4
native-added (partial)	37.0	52.5	39.1
native-added (complete)	37.4	80.2	47.6
perfectly-ranked	36.8	66.9	48.5
perfectly-ranked + native-added (partial)	36.8	66.3	46.2
perfectly-ranked + native-added (complete)	37.1	85.5	43.7
perfectly-ranked + perfectly-weighted (partial)	36.8	67.1	46.9
perfectly-ranked + perfectly-weighted (complete)	52.6	71.1	54.5
perfectly-ranked + perfectly-weighted (complete) + native-added (partial)	52.6	74.3	54.4
perfectly-ranked + perfectly-weighted (partial) + native-added (complete)	37.2	87.0	43.9
perfectly-ranked + perfectly-weighted (complete) + native-added (complete)	99.7	100.0	100.0

Table 4.2. Data Oracle Results

Summary of assessment by data oracles (alone and in combinations). For each test case, we report the average percent overlap between the target structure and the average, best-found and best-scoring models over the benchmark set.

4.4.1. Data Oracles

To improve the accuracy of the data module, we used the “native-added” data oracle in two ways: partial assistance (up to 50% of the interactions) and complete assistance (100% of the interactions). For the partial assistance cases, we performed multiple random selections for the interactions chosen for use with the more powerful oracle. Under partial assistance, we found no improvement from the unassisted case with an average accuracy across our benchmark set of 37.0% average %-overlap, 52.5% best-found %-overlap, and 39.1% top-scoring %-overlap (Table 4.2). With complete assistance, the best-found and top-scoring accuracy improved: we found an average accuracy across our benchmark set of 37.4% average %-overlap, 80.2% best-found %-overlap, and 47.6% top-scoring %-overlap (Table 4.2).

To improve the precision of the data module, we used the “perfectly-ranked” data oracle for all of the pairwise interactions and found an average accuracy across our benchmark set of 36.8% average %-overlap, 66.9% best-found %-overlap, and 48.5% top-scoring %-overlap (Table 4.2). Without the native interaction, the assistance of the “perfectly-ranked” data oracle failed to improve the average accuracy over the unassisted method, but did improve the best-found and top-scoring accuracy of the method.

To improve the accuracy of the “perfectly-ranked” data oracle, we tested the combination of the “perfectly-ranked” and “native-added” data oracles. As before, we used the “native-added” data oracle under partial and complete assistance. Under partial assistance, we did not find any improvement accuracy over the “perfectly-ranked” data oracle, specifically we found across our benchmark set a 36.8% average %-overlap, 66.3% best-found %-overlap, and 46.2% top-scoring %-overlap (Table 4.2). Under complete assistance, the average accuracy across our benchmark set still did not improve at 37.1% average %-overlap, nor did the top-scoring accuracy improve at 43.7% top-scoring %-overlap, but we did find an improvement for the best-found %-overlap at 85.5% (Table 4.2).

To further increase the precision of the input data, we used the "perfectly-weighted" data oracle in combination with the "perfectly-ranked" data oracle under partial and complete assistance. Without the native orientation, with partial assistance from the "perfectly-weighted" data oracle, we found no improvement over the "perfectly-ranked" data oracle alone, with an average accuracy across our benchmark set of 36.8% average %-overlap, 67.1% best-found %-overlap, and 46.9% top-scoring %-overlap (Table 4.2). With complete assistance, the accuracy improved: we found an average accuracy across our benchmark set of 52.6% average %-overlap, 71.1% best-found %-overlap, and 54.5% top-scoring %-overlap (Table 4.2).

To improve the accuracy of the "perfectly-weighted" data oracle, we tested the combination of the "perfectly-weighted" and "native-added" data oracles under partial and complete assistance. When we included the native orientation for half of the interactions (selected at random), we found an average accuracy across our benchmark set of 52.6% average %-overlap, 74.3% best-found %-overlap, and 54.4% top-scoring %-overlap (Table 4.2). With complete assistance, we found an average accuracy across our benchmark set of 99.7% average %-overlap, 100.0% best-found %-overlap, and 100.0% top-scoring %-overlap (Table 4.2).

To compare the relative strengths of the "perfectly-weighted" and "native-added" oracles, we tested the combination of using the "native-added" oracle under complete assistance and the "perfectly-weighted" oracle under partial assistance and found an average accuracy across our benchmark set of 37.2% average %-overlap, 87.0% best-found %-overlap, and 43.9% top-scoring %-overlap (Table 4.2).

4.4.2. Integration Oracle

We tested the integration oracle with the unassisted data module and the "perfectly-ranked" data oracle (Table 4.3). Unfortunately, because of the exponential nature of exhaustive combinatorial search, we were only able to test the top 10 hits for both data modules for 5 of the test cases, and only the top 3 hits for 1ss8. The best possible model found with the unassisted data module had an average %-overlap of 52.0%. With the "perfectly-ranked" data oracle the best possible model found had an average %-overlap of 71.7%. In order to make an appropriate comparison, we generated 10,000 independent models for the test cases using the same amount of input data (top 3 hits for 1ss8, top 10 hits for the other five test cases) with the unassisted and the "perfectly-ranked" data oracles. We found an average unassisted %-overlap of 51.9%, and with the "perfectly-ranked", an average %-overlap of 70.4% (Table 4.3). Using the same protocol, we tested the combination of "perfectly-ranked" and "perfectly-weighted" data oracles and found an average %-overlap of 71.6% (Table 4.3).

4.4.3. Ranking Oracle

Finally, the best found percent-overlap, reported above, also corresponds to the accuracy of the method (assisted or unassisted) with the use of the ranking oracle (Table 4.2).

Test Cases	Using Integration Oracle	Best-found model
unassisted data module	Yes	52.0
unassisted data module	No	51.9
perfectly-ranked data oracle	Yes	71.7
perfectly-ranked data oracle	No	70.4
perfectly-ranked + perfectly-weighted data oracle	No	71.6

Table 4.3. Integration Oracle Results

For each test case, with and without the integration oracle, we report the average percent overlap between the target structure and the best-found models over the benchmark set.

4.5. Assessment Interpretation

Having measured the accuracy of the method assisted by oracles, alone and in combinations, we determined the limiting aspects to hierarchical docking by the satisfaction of spatial restraints generated by a pairwise docking.

4.5.1. Evaluation of the Data Module by the Data Oracles

For the average model, we failed to see any improvement without the complete assistance of both the "perfectly-ranked" and the "perfectly-weighted" data oracles. By adding both of these oracles, we found an improvement of 42.7%. By adding the complete assistance of the "native-added" data oracle to the other two data oracles, we found an average %-overlap of 99.7%.

For the best-found model, we found an average improvement of 37.0% with the addition of the complete assistance of the "native-added" data oracle. For the "perfectly-ranked" data oracle, we found an average improvement of 18.3%. We found an additional

12.2% improvement by the addition of the complete assistance of the "perfectly-weighted" oracle.

For the best-scoring model, we did not find a distinct pattern of improvement. The addition of the complete assistance of the "native-added" data oracle alone, or any other combination of oracles without the complete assistance of the "perfectly-weighted" data oracle, improved the accuracy by an average of 20.2%. With the complete assistance of the "perfectly-weighted" data oracle, without the complete assistance of the "native-added" data oracle, we found an additional 23.0% improvement. With the complete assistance of all of the data oracles, we found an average %-overlap of 100%.

As the power of the data oracles increased, the separation between the average model and the best-scoring model decreased, and the separation between the best-scoring and the best-found model also decreased. Also, the increased power of the data oracles decreased the degeneracy of the scoring function. With just the addition of the "native-added" data oracle for just over half of the interactions, we found best-found %-overlap of 80.2%, but we were unable to identify these models with our scoring function.

4.5.2. Evaluation of the Optimization Method by the Integration Oracle

The integration oracle established an upper bound on the accuracy of any optimizer. In our experiments, we found for the unassisted scoring function that the optimization method found solutions that were on average within 0.19% of the best %-overlap found by the integration oracle. For the scoring function assisted by the "perfectly-ranked" data oracle, the optimization method found solutions that were on average within 1.8% of the best %-overlap found by the integration oracle. If we used the "perfectly-ranked" data oracle with the "perfectly-weighted" data oracle, the optimization method found solutions that were on average within 0.14% of the best %-overlap found by the integration oracle.

4.5.3. Evaluation of the Scoring Function by the Ranking Oracle

As the best-found model represents the performance of a method assisted by the ranking oracle, the difference in accuracy between the top-scoring model and the best-found model represents the improvement to be gained by improving the ranking module.

4.5.4. Implications and Recommendations

We found that the accuracy of this method is almost totally data driven. The limiting factor was the presence of a sufficiently native-like orientation for a sufficiently high percentage of the interactions. Even the degeneracy of the scoring function was dependent on the quality of the data module.

While more sophisticated optimization techniques might yield more efficient usage of computing resources, this method is computationally tractable (on ~300 2.3 GHz 64-bit cores it takes ~6 hours to generate 10,000 models for all six targets in the benchmark set). And while the degeneracy of the scoring function is problematic, the improvement in the input data quality drives improvements in both the quality of the output models as well as the scoring function. We found that by increasing the amount of assistance provided by each of the oracles to at least 50% of the interactions (*c.f.*, up to 50% of the interactions) we achieved the same accuracy found under complete assistance (see Supporting Information (SI) Table T1). Therefore, further resources should be devoted to improving the input data accuracy.

4.6. Discussion

First, to provide a specific example for the modular assessment framework, we described the hierarchical docking problem and a method for addressing the problem. Second, we decomposed the problem into three modules and described oracles for each of the modules. Third, we performed modular assessment by applying the oracles (alone and in combinations) to our method and reported the results. Specifically, we showed that the accuracy bottleneck for this particular approach lies in the quality of the underlying input data. Further, we determined the improvement needed to achieve varying levels of accuracy. Finally, based on these findings, we made recommendations for future development.

The standard assessment strategy of reporting the accuracy of a method over a benchmark set has at least two limitations: the reported accuracy (i) does not, by itself, provide insight into how a method could be improved and (ii) often depends on the specific benchmark set, making a fair comparison between methods that do not share a benchmark set difficult. Modular assessment is a completely general strategy for assessing methods for structure modeling that addresses both of the limitations of the standard assessment strategy.

First, modular assessment provides specific data on how various modules of the method impact the overall accuracy of the method and measures the accuracy ceilings for each of module. These data on the limitations of the method can guide future development along more efficient avenues.

Second, modular assessment can be used to compare two different methods in much greater detail. By using modular assessment on a second method with a different benchmark, we could compare the accuracy of each aspect of the two methods separately. Then by using modular assessment on both methods using the other method's benchmark set, we could further highlighting the reason for each method's comparative advantage over its respective benchmark set. Finally, by construct a new benchmark set by combining the two benchmark sets, we could provide a summary of the accuracy differences between the two methods.

The cost for this additional data comes in additional time and effort. The modular assessment framework requires methods developers to decompose their methods into modules, construct oracles, and assess their methods multiple times over. For example, the assessed method for hierarchical docking was run over the benchmark set 16 times: once unassisted, 10 times for analysis using data oracles, and 5 more times for analysis using integration oracles.

But with this additional data comes a better understanding of the errors and limitations in our methods, and this improves the confidence with which we use the models generated by these methods. While we demonstrated our method on a particular structure modeling problem, this approach can be adapted to any problem that can be broken into modules representing possible areas of improvement. Any structure modeling problem that can be expressed as an optimization of a scoring function can benefit from our assessment strategy. Ultimately, this framework provides a higher resolution of assessment whether used to assess a single method or compare multiple methods.

4.7. Materials and Methods

4.7.1. Pairwise Docking Results to Spatial Restraints

To demonstrate the modular assessment framework for hierarchical docking on a specific method, we describe a method for hierarchical docking that generates models by the satisfaction of spatial restraints generated by pairwise docking. In particular, we use lists of PATCHDOCK hits, ranked by the provided score, as our input data. For each test case, we take the top 20 hits as ranked by the PATCHDOCK scoring function and represent them using a specialized spatial restraint. To express relative orientation data, we use a three-tiered combination of restraint types (see SI Figure S1).

For each possible configuration, we extract the representative distances, translate these distances into distance restraints, and merge these distance restraints into a single multidimensional Gaussian (MDG) restraint. We further combine each of these multidimensional Gaussian restraints into a weighted exclusive-OR (XOR) restraint. This weighted XOR restraint determines which of the possible configurations is most satisfied and current configuration of the model, and then enforces the corresponding underlying MDG restraint.

For each configuration of a pair of subunits, we extract all sixteen distances, between the references points for the two subunits, *i.e.* the complete bipartite graph of distances between the two sets of reference points. The sixteen representative distances are expressed as Gaussian distance restraints with a standard deviation of $s_i = d_i / 10$.

For each relative orientation, the sixteen distance restraints are then combined into a single MDG restraint, which has a violation of: $MDG = \left(\frac{1}{16}\right)\left(\frac{rt}{2}\right)\sum_{i=1}^{16}\frac{(d_i - \mu_i)^2}{\sigma_i^2}$

This allows us to express each configuration with a single restraint.

For each pair of subunits, every configuration under consideration is expressed in this way, and then combined using an XOR restraint, with a violation of:

$$XOR = \underset{j}{Min}\left(\frac{MDG_j(m)}{w_j}\right)$$

This XOR restraint is a runtime restraint that evaluates the current state of the model, and selects the least-violated alternative of its constituent restraints. In this case, the XOR restraint evaluates which configuration of a pair of subunits is most consistent (creates the least violation) and enforces the weighted multiple Gaussian restraints representing that configuration. We can assign the weights for each relative orientation to represent a selection probability for a particular underlying configuration of a pair of subunits. For this experiment, we assign a uniform weight and selection probability to each relative orientation from the underlying pairwise docking method.

The integration module (*i.e.* the satisfaction of these spatial restraints by optimization) will be done using MODELLER. To assign acceptance probabilities, we use a delta-function accepting and denying configurations based solely on excluded volume, defined by 0.8*atomic radii for main-chain, C-alpha atoms only.

As in the generalized solution, the scoring function is a combination of the underlying scores for the pairwise relative orientations and optionally, a physics-based score (*e.g.* DOPE score). For this experiment, we complete the ranking module by using this scoring function, without a physics-based score contribution, to sort the models generated by the independent optimization from 10,000 randomized initial configurations.

4.7.2. Oracle Descriptions

Here we describe the specific oracles for the assessment of hierarchical docking by the satisfaction of spatial restraints generated by a pairwise docking. We define the correct native structure in these trials as the X-ray crystallographic structures of the complexes.

We tested our data module using four different data oracles, described here in increasing power. The "native-added" data oracle takes the raw output from the underlying docking method and adds the native orientation to the top 20 hits. The "perfect-ranking" data oracle, takes the raw output from the underlying docking method, ranks each of the relative orientations by the distance from the relative orientation in the native structure, and then returns the true top 20 hits. The "perfect-ranking" data oracle can then be further augmented by applying the "native-augmenting" data oracle, which adds the native orientation to the true top 20 hits. In each of these cases, the list of relative orientations, nodes in the partitions of *G'*) are weighted uniformly (*i.e.* selection probabilities are equal). The "perfect-weighting" data oracle modifies the data module by weighting the selection probabilities by distance from the native orientation.

The integration oracle exhaustively searches all possible combinations of pairwise interactions to find the complex configuration with highest similarity to the native complex, guaranteeing the best possible configuration from the underlying data. The ranking oracle sorts the ensemble of resulting models by the distance from the native structure.

We used the %-overlap of mainchain atoms at 3.5Å as the distance function, after a least-square superposition, for each of the oracles.

4.8. Future Directions

The future directions for this study would be to push for continued adoption of this assessment strategy, and to use the framework to assess more methods. One possible avenue to lower the barrier for adoption could be by developing software tools to help automate the process of developing, managing and running modular assessment and to help perform the associated comparisons and analysis.

5. Theoretical Models

Given an assembly, we can perform many analyses, both experimental and computational to elucidate structural features and relationships of the subunits within an assembly. With multiple sources of structural data, the challenge of assembly structure modeling becomes one of data integration. Throughout this work, we have explicitly and implicitly presented the use of points and spatial restraints as a unifying framework for representing heterogeneous sources of data. In this framework, subunits are represented by sets of points and possible assembly configurations are represented by configurations of these sets of points. Available data on structural features and relationships are then encoded by spatial restraints. These spatial restraints are then combined into a scoring function, which is then optimized to find the configuration (or ensemble of configurations) that is in least violation of these restraints and therefore in greatest agreement with the input data.

This chapter details the assumptions that underlie using points and restraints to model macromolecular structures, describes properties of systems modeled by this framework, and aims to provide a path towards a more principled approach for integrating multiple sources of data. Graph-theoretic and geometric models are used to represent structures of subunits and assemblies. Graph theory and Bayesian networks are used to describe modeling methods. Information theory is used throughout this chapter to describe the properties of assemblies and modeling methods.

5.1. Representing Assemblies by Sets of Points

To represent assemblies as sets of subunits, we begin by describing the sets of points we use to describe subunits. Let $C = (p_1, \dots, p_n)$, $p_i \in \mathbb{R}^d$ be a configuration of *n* points in *d*-space, which can be collectively represented as a point in $\mathbb{R}^{dn}$, where for structure modeling *d* is most commonly taken to be three with notable exceptions (*e.g.*, one-dimensional spin glass models and two-dimensional lattice models). Describing the representation of subunits is now equivalent to choosing *n* and how to partition these *n* points into subsets representing subunits. In particular, for an assembly consisting of *m* subunits, we begin the modeling process by deciding how many points in space we will need to represent each subunit. Assuming that each of the *m* subunits are spatially distinct, we need a minimum of one point per subunit, yielding *m* points. This case arises when subunits are modeled as spheres without respect to any specific orientations; each subunit can be represented by the center of the sphere and a radius *r* for the size of the sphere.

However, in cases when the specific orientation of the subunit is important to the modeling problem, we need more points per subunit to describe each subunit. Specifically, to represent subunits with orientations in *d*-space, we need to represent each subunit as a *d*-simplex. By labeling each of the *d*+1 points of the *d*-simplex, we can describe the relative orientation of the subunit.

5.1.1. Internal Distance Restraints for Subunit Representations

We now examine the specific case of using distances to represent a configuration C in three-dimensions. Let $f(n)$, for $n > 3$ represent the number of distances between points in C that are necessary and sufficient to completely specify the configuration of points. Because we are using distance to represent configurations, a complete specification of a configuration of points can only be complete with respect to translation and rotation, but not to symmetry – a set of chirally related points are indistinguishable by internal distances. The total number of distances is $\binom{n}{2}$ representing an upper bound on *f (n)*. For $n > 5$, we will show that $\binom{n-4}{2}$ of the $\binom{n}{2}$ distances are redundant.

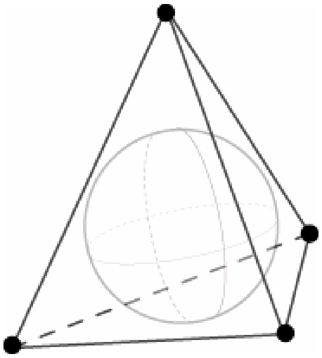

Figure 5.1. Four points forming a 3-simplex

For three dimensions, a 3-simplex, which consisting of 4 points, will be selected as a reference object (Figure 5.1). For this reference object, every internal distance is required to specify the configuration of points, $\binom{4}{2} = 6$ distances. All additional points, must be specified by 4 distances, by construction, we select them to be distances to the vertices of our reference 3-simplex. Intuitively, this is because each additional point must also form a 3-simplex (requiring 3 more distances) and additionally requires one more distance to disambiguate the two possible positions relative to the plane specified by the other three points in the 3-simplex. The fifth point is a special case as it adds every possible distance to the reference 3-simplex, requiring a set of $\binom{5}{2} = 10 = \binom{4}{2} + 4$ specifying distances. But for each subsequent point, there are redundant distances. From this, we obtain: $f(n) = 6 + 4(n-4) = 4n - 10$ for $n > 3$. This is equivalent to:

$$f(n) = \begin{cases} \binom{n}{2}, 2 \le n \le 5 \\ \binom{n}{2} - \binom{n-4}{2}, n > 5 \end{cases}$$

Therefore, for more than five points, $\binom{n-4}{2}$ of the $\binom{n}{2}$ distances are redundant.

It is worth noting that in two dimensions, $f(n) = 3n - 6$ for $n > 2$ by a similar argument and that by subtracting the two expressions, we find that *n* – 4 distances encode for the additional dimension. These arguments can, by analogy, be applied to higher dimensions, but we omit this discussion for lack of relevance to structures or simplified models of structures.

5.1.2. Distance Restraints for Assembly Modeling

When, in three dimensions, we model each subunit of an assembly using a 3-simplex, the function above also specifies the minimum number of additional distance restraints between the two subunits required to specify the relative orientations of the two subunits. In an assembly with two subunits, there are 8 points (two 3-simplexes of 4 points) and 12 internal distances (6 internal distances for each 3-simplex). To completely specify a configuration of 8 points requires $f(8) = 22$ points, yielding the requirement of 10 additional non-redundant distances between the two subunits (Figure 5.2). By extension, to specify any assembly of m subunits, where the subunits are represented by 3-simplexes, requires $10(m-1)$ additional non-redundant distances between subunits. The selection of non-redundant distances is guided by rigidity theory and can be summarized as adding the distances to prevent any deformation or ambiguity of the resulting structure.

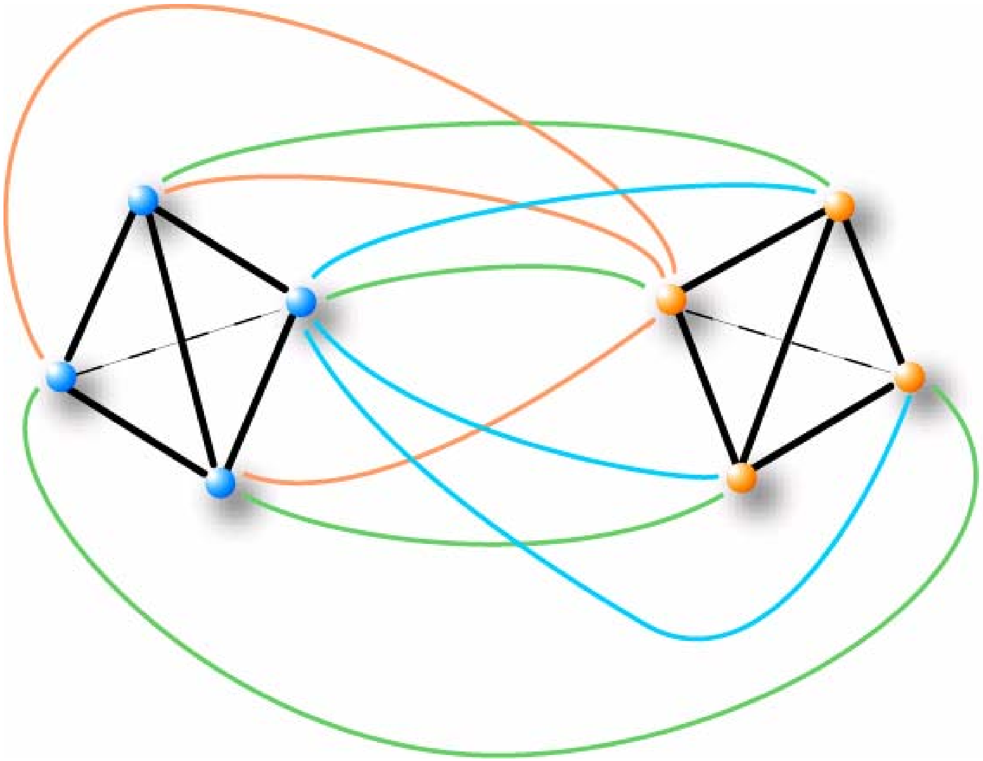

Figure 5.2. Ten Non-Redundant Distance Restraints

For models with atomic level of detail, we can use a point for every atom (or some subset of atoms, such as all heavy atoms, main-chain atoms, alpha-carbons, *etc.*). Here, we note that distance restraints are defined at the level of chemical bonds and physical forces, and that the requirements for rigidity (*i.e.*, complete specification of a configuration of points) are the same and that large-scale flexibility in a structure corresponds either to having an insufficient number of distance restraints or having distance restraints with higher uncertainty.

It is also worth noting that this work on distance restraints is closely related to ideas in distance geometry [85-87] and rigidity theory [88].

5.2. Configurational Entropy of Binary Docking

Binary docking is the pairwise docking of two subunits, also known as protein-protein docking. Here, we present an information theoretic analysis of binary docking and sample calculations.

5.2.1. Center-Center Distance Restraints

We begin by examining the information content of a single distance restraint between the centers of two subunits for different possible combination of geometries for the subunits.

Sphere-Sphere Docking

Given an assembly of two subunits represented as spheres using one point per subunit, we assert that docking results cannot be additionally informative. Let Z be data representative of the subunit descriptions (*e.g.,* the excluded volume of each subunit). Given Z and the contact criterion (*i.e.,* the subunits in question are in contact) which we represent by C, the docking results X, encoded by a single distance restraint r_d, yields no reduction of uncertainty over the model, represented by θ. Let *d* be the distance between the centers of the two spheres restrained by the distance restraint r_d.

$$H(\theta \mid Z, C) = H(\theta \mid Z, C, r_d) \tag{5.1}$$

Let $\Psi = (\theta \mid Z, C)$ be the distribution of models consistent with *Z* (subunit descriptions) and *C* (contact criterion). Equation (5.1) simplifies to:

$$H(\Psi) = H(\Psi \mid r_d) \tag{5.2}$$

We can express the information, or reduction of uncertainty, introduced by r_d:

$$I(\Psi : r_d) = H(\Psi) - H(\Psi \mid r_d) \tag{5.3}$$

Rearranging:

$$H(\Psi \mid r_d) = H(\Psi) - I(\Psi : r_d) \tag{5.4}$$

But for the case that *Z* defines spherical subunits, with radii r_1 and r_2, and *C* is enforced, *d* is a function of Ψ.

$$d = f(\Psi) = r_1 + r_2 \tag{5.5}$$

Since we can express *d* deterministically and completely as a function of Ψ, $I(\Psi : r_d) = 0$, verifying the assertion.

Sphere-Rod Docking

If either of the two subunits is not spherical in representation, then docking results can be informative. Using the same formalism from the previous section:

$$H(\Psi) \geq H(\Psi \mid r_d) \tag{5.6}$$

$$\therefore I(\Psi : r_d) \geq 0 \tag{5.7}$$

As *d* is no longer a function of Ψ, this suggests that docking information could reduce the uncertainty over possible models. In this situation, given Ψ, a distribution of possible values for *d* arises. The degree to which *X*, as encoded by a single distance restraint reduces the uncertainty over the center of mass of distances is the information content of *X*.

The range of the distributions for *d* is bounded by the minimum and maximum center of mass distances. The anisotrophism present in the representation of the subunits is the only determinant of this distance. This relates to the first assertion, the case where there was only one possible value for *d*. In this simplified model, we will consider two possible forms of anisotrophism (Figure of sphere, rod, disc). In the limit, these two forms can be summarized as stretching a sphere along one (resulting in a rod) or two (resulting in a disc) of the three dimensions. However, if represent subunits as a collection of spheres of various sizes connected together we only need to consider the "rod" case of stretching a sphere along one dimension (Figure of spheres to rod).

We begin by considering the interaction between a sphere of radius r_1 and a rod of radius r_2 and length *l*.

Without any additional data, other than the contact criterion, we can determine a prior distribution for *d* that we will denote S_0. To get an intuition for S_0, we examine an assembly on a two-dimensional lattice for $r_1 = r_2 = r$ and $l = 3r$ (Figure 5.3).

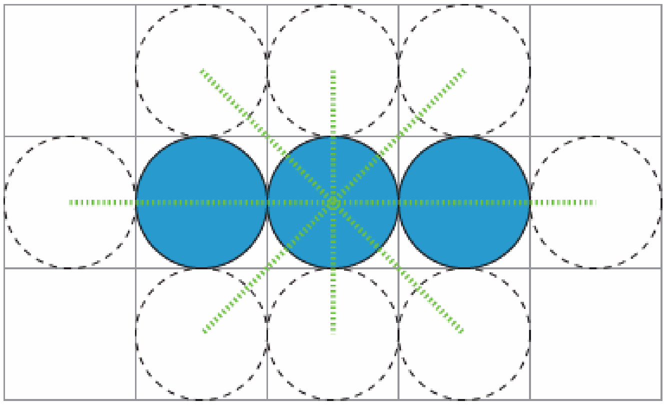

Figure 5.3. Rod and Possible Sphere Positions

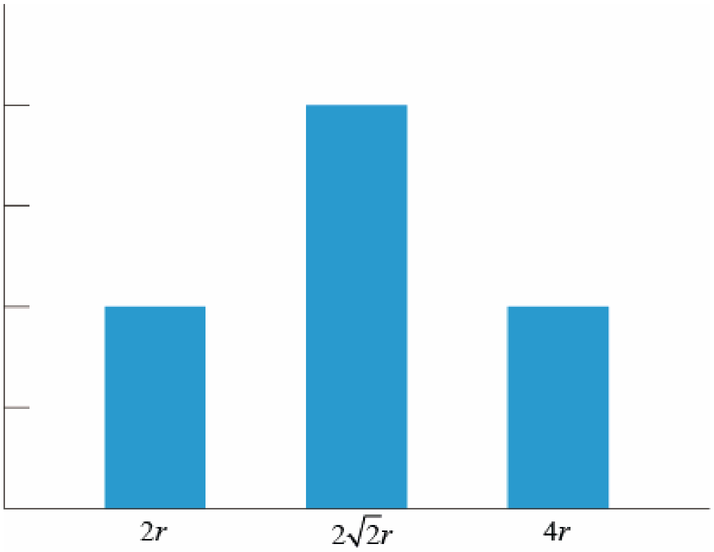

Figure 5.4. Histogram of Rod-Sphere Center-Center Distances

From the histogram (Figure 5.4), we can normalize the distribution and calculate the entropy of S_0 for $r_1 = r_2 = r$ and $l = 3r$ on a two-dimensional lattice. In general, we can determine the entropy of S_0 computationally by running many independent simulations of such a system, calculating the center-center distances from the resulting models, discretizing the distances, normalizing the resulting distribution of distances and calculating $H(S_0)$.

As a general note, the degree to which the spherical representation is an over-simplification may play a role in the amount of information that is actually present. But this effect is likely to be cancelled by excluded volume effects (for observed distances less than *r*) and by the contact criterion (for observed distances greater than *r*). At a higher resolution (*i.e.*, using more points per subunit) a more realistic center of mass definition is likely to make the center-center distance more informative. This is in agreement with the assertion that the degree of anisotrophism in a representation is proportional to the total amount of potential information.

For binary docking between a sphere of radius r_1 and a rod of radius r_2 and length l the relationship between r_1 / r_2, l and $H(r_d)$ can be summarized by these two statements: first, for a given r_1 / r_2 as l increases, $H(r_d)$ increases; and second, for a given l as r_1 / r_2 diverges from 1 (towards 0 and ∞), $H(r_d)$ increases. Below, we analytically demonstrate both of these relationships.

In the first statement, for a fixed r_1 / r_2 as l increases the total surface area of potential contact for the sphere increases. Now, without any additional data, we solve for $H(S_0)$, beginning with the surface area of the rod:

$$SA_{rod} = 2\pi r_2^2 + 2\pi r_2 l = 2\pi r_2 (r_2 + l) \qquad (5.8)$$

Because we are calculating a distribution of center-center distances, we can consider only a quarter of a maximal longitudinal slice, without loss of generality (Figure 5.5). ${}^{l_2}\!/_{2}$

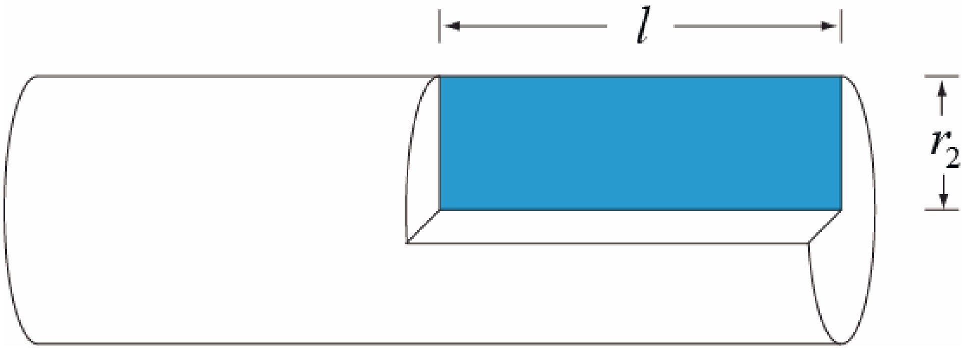

Figure 5.5. Quarter of a Maximal Longitudnal Slice of a Rod

Independent of r_1, on this quarter of a slice, the number of potential points of contact is proportional to $k(r_2 + l)$. As the number of potential points of contacts increases, the uncertainty over the possible center-center distances also rises. Applying the continuous form of the entropy equation yields:

$$H(S_0) = -k_l \int_0^l p_c(x) log_2 p_c(x) dx - k_{r_2} \int_0^{r_2} p_c(y) log_2 p_c(y) dy \qquad (5.9)$$

Here p_c is the probability of contact. Equation (5.9) can also be expressed as a surface integral:

$$H(S_0) = -k \int_0^l \int_0^{r_2} p_c(x,y) \log_2 p_c(x,y) dxdy \qquad (5.10)$$

In the second statement, for a given *l* as r_1 / r_2 diverges from 1 (towards 0 and ∞), $H(r_d)$ increases, Equation (5.9) is a better fit than Equation (5.10). The entropy of the background distance distribution can be expressed by:

$$H(S_0) = -k_m (k_l \int_0^l p_c(x) \log_2 p_c(x) dx + k_{r_2} \int_0^{r_2} p_c(y) \log_2 p_c(y) dy) \qquad (5.11)$$

Here k_m represents the symmetry multiplier and is equal to $4\pi r$, and $k_l \cong k_{r_2} \cong 1$. This can be thought of as a path.

The interaction of the rod and sphere at the corner of the rod is not account for in Equations (5.9), (5.10) and (5.11). For completeness, we can address this by adding a term:

$$k_a \int_0^{\pi r/2} p_c(z) \log_2 p_c(z) dz \tag{5.12}$$

However, as we are interested in the difference in entropy under a fixed r_1 / r_2 this term drops out.

Returning to the first case, the total possible information gain would then be the value of $H(S_0)$ as calculated by Equation (5.11) with the additional corner term, Equation (5.12). The relative information gain provided by some data would be expressed by the difference in information theoretic entropy. In either case, it is clear from Equation (5.11) that $H(S_0)$ is dominated by *l* as r_1 / r_2 is fixed and the corner terms are also fixed. Therefore, as *l* increases, so does $H(S_0)$.

Finishing the second case, from Equation (5.11), having a fixed *l* allows the r_2 term and corner term in Equation (5.12) to dominate. Note that r_1 does not factor into $H(S_0)$, therefore as r_1 / r_2 diverges from 1 (towards 0 and ∞), $H(S_0)$ increases.

Rod-Rod Docking

For two anisotropic subunits (here, two rods) there are four variables, two for each subunit: (r_1, l_1) and (r_2, l_2) defining the radii and lengths of the two rods. This allows us to observe two ratios: r_1 / r_2 and l_1 / l_2. In this system, the center-center distance is bounded below by $d = r_1 + r_2$ for the two rods aligned by centers with primary axes parallel (Figure 5.6A) and bounded above by $d = {}^{l_1}\!/_2 + {}^{l_2}\!/_2$ for the two rods aligned end to end (Figure 5.6B).

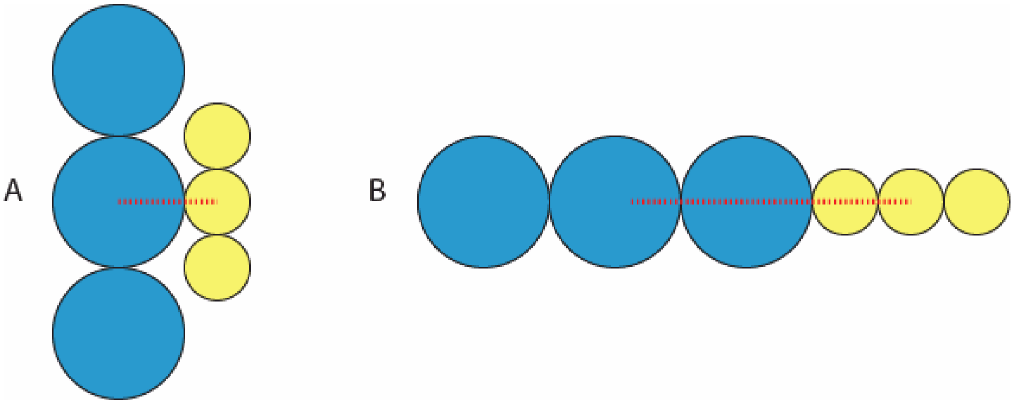

Figure 5.6. Bounding Cases for Rod-Rod, Center-Center Distance

Using an analogous derivation as the previous section, we calculate $H(S_0)$ for this system as follows: without loss of generality, we fix the position of rod 1 and allow rod 2 to move around the surface of rod 1 changing its point of contact. Again, we consider only a quarter of a maximal longitudinal slice for both rods. With this simplification, we derive the possible space for valid center-center distances (Figure 5.7).

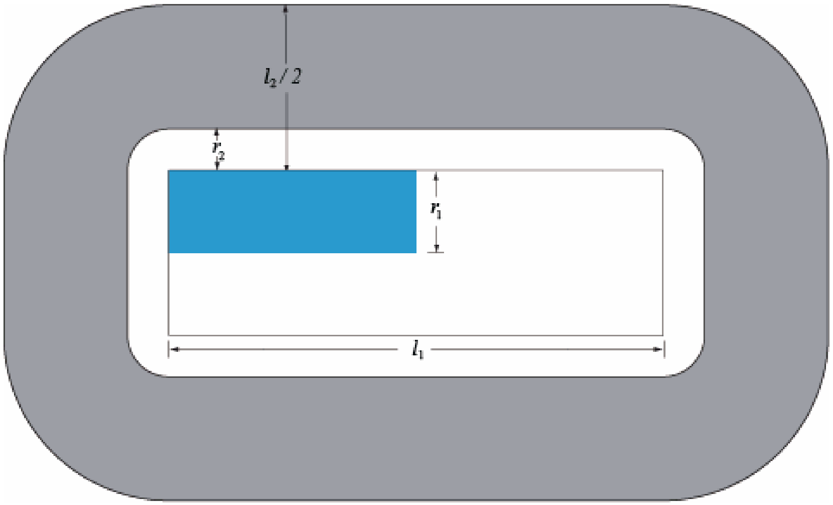

Figure 5.7. Valid Rod-Rod, Center-Center Distances

This area can be calculated piecewise, and using scaling factors, we determine $H(S_0)$ as follows:

$$H(S_0) = -k_m \begin{pmatrix} k_l \int_0^{l_1} \int_0^{l_2 r_2} p(x,u) \log_2 p(x,u) dxdu + \\ k_r \int_0^{r_1} \int_0^{l_2 r_2} p(y,v) \log_2 p(y,v) dydv + \\ k_a \int_0^{\pi/2} \int_0^{l_2 r_2} p(z,w) \log_2 p(z,w) dzdw \end{pmatrix} \tag{5.13}$$

$H(S_0)$ increases as the integrated area increases. As $l_1 + l_2$ increases and as $r_1 + r_2$ increases, $H(S_0)$ increases.

Another consideration we need to make for this case involves the density of solutions for a given point of contact along rod 1 with respect to the position of the center of rod 2. Aside from the extremes, there are multiple solutions for any give point of contact with rod 1 and the center of rod 2. Fortunately, this behavior is the same for all such pairs of points and can be factored out as a constant. Using this formulation, we can revisit the result for one rod and one sphere and state that anything increasing the length of the path of contact along the quarter of a maximal longitudinal slice increases $H(S_0)$. For emphasis: the reason that the contact path dominates in the case of one rod and one sphere is because a sphere only as a single point of contact which uniquely determines the position of the center of the sphere, changing the radius of the sphere will increase the center-center distance but it does not change the distribution of distances. This is different in the case of two rods, where the anisotrophism of both subunits changes the distribution of center-center distances.

5.2.2. Orientation-Dependent, Center-Center Distance Restraints

For the previous section, we assumed that we had no data to orient or even specifically identify the point of contact. Here, we return to spherical representations for both subunits and we address orientation dependence and contact points (*i.e.*, patches) in reduced models. As determined in the previous section, the minimum number of points required to be able to specify the orientation of an object in three dimensions is a 3-simplex, defined by 4 points and 6 internal distances. If we include the center of mass as an additional reference point, this yields five points. For computational ease, we consider six points for orientation-dependence and a seventh point to specify the center of mass (a point in the center, and two points in each of three orthogonal directions).

Patch-Patch Contact Restraints for Sphere-Sphere Docking

For patch-patch contacts, we take the representation for subunits and add a point per patch defined relative to the seven points that define the subunit. The excluded volume per subunit is defined by a convex hull around the points.

So *Z*, the description of the subunits, is now defined by the seven points and the excluded volume. We enforce *C*, the contact criterion, and *Q* the patch-patch location. Now the degrees of freedom in a docking result are around the axis of contact (defined by the mass centers and patches).

$$H(\theta \mid Z, C, Q) = H(\Psi \mid Q) \tag{5.14}$$

$$I(\Psi : Q) = H(\Psi) - H(\Psi \mid Q) \tag{5.15}$$

Here $\Psi = (\theta \mid Z, C)$ is the distribution of models consistent with *Z* (subunit descriptions) and *C* (contact criterion). The background distribution entropy $H(\Psi)$ can be calculated analytically or estimated from simulation using a similar construction as in the center-center distance restraints without orientation dependence.

Using the expression from the previous section, we find that specifying the seven points requires 18 internal distances and four additional points per patch. Minimally, using the 3-simplex representation, we require many more points to define the subunits in question two points (one point per subunit) for orientation-independent subunits *vs.* 20 points (ten points per subunit) for orientation-dependent subunits.

In the orientation-independent case, the two distances that we extract are the center-center distance and the zero-length distance that is enforced by the contact criterion (Figure 5.8).

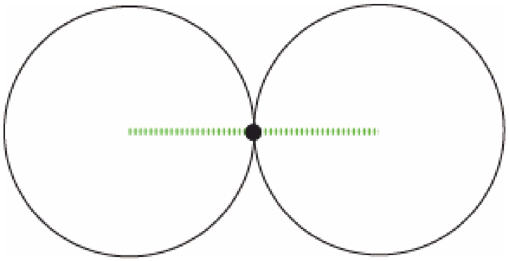

Figure 5.8. Center-Center Distance and Zero-Length Distance

Without loss of generality, we fix the first subunit and allow the second to rotate around the axis defined by the distance between the mass centers. The contact point (*i.e.*, patch) is collinear to this axis. Let S_1 be the prior distribution of distances for this system. The uncertainty is then given by:

$$H(S_1) = -\int_0^{2\pi} p(r)\log_2 p(r)dr \tag{5.16}$$

A priori, we cannot define a greater probability to any particular relative orientation, thus the uniform distribution formulation of entropy applies, giving:

$$H(S_1) = \log_2(2\pi) \approx 2.65 \text{ bits} \tag{5.17}$$

Ignoring the non-uniform distribution of information over the distance restraints, we can consider the average information content per distance restraint.

$$I(S_0 : S_1) = H(S_0) - H(S_1) \tag{5.18}$$

$$H(S_0) = H(\theta \mid Z_0, C) \tag{5.19}$$

$$H(S_1) = H(\theta \mid Z_1, C, Q) \tag{5.20}$$

Substituting Equations (5.19) and (5.20) into Equation (5.18) yields:

$$I(S_0 : S_1) = H(\theta \mid Z_0, C) - H(\theta \mid Z_1, C, Q) \tag{5.21}$$

By factoring Z_1 into components: Z_0 and points added for orientation dependence (Z') and substituting $Z' = Z_1 - Z_0$ into Equation (5.21) gives:

$$H(\theta \mid Z_0, C) - H(\theta \mid Z_0, C, Z') \tag{5.22}$$

Let $\Psi = (\theta \mid Z_0, C)$ be the distribution of models consistent with Z_0 (subunit descriptions) and *C* (contact criterion). Substituting Ψ into Equation (5.22) yields:

$$I(\Psi : Q, Z') = H(\Psi) - H(\Psi \mid Q, Z') \tag{5.23}$$

Specifically, the contribution from orientation dependence can be expressed by (Q, Z') which we call *T*. Substituting *T* into Equation (5.23) gives:

$$I(\Psi : T) = H(\Psi) - H(\Psi \mid T) \tag{5.24}$$

The distance restraint from the center-center result is contained within the *Q* term. We can factor the *Q* term into r_d and the zero length distance r_0, yielding: $Q = (r_d, r_0)$.

$$I(\Psi : r_d, T') = H(\Psi \mid r_d) - H(\Psi \mid r_d, r_0, Z') \tag{5.25}$$

$$\Psi' = (\Psi \mid r_d) \tag{5.26}$$

$$I(\Psi' : T') = H(\Psi') - H(\Psi' \mid r_0, Z') \tag{5.27}$$

Here T' is the marginal contribution from the patch-patch restraint adjusted for the information content from the center-center distance restraint. So returning to the average contribution per additional restraint, we take this information content and divide by the number of additional restraints. For the system represented by 3-simplexes, we go from three distance restraint to 20 distance restraints and two additional distances (center-center distance and zero-length distance). Thus Equation (5.27) must be divided by 18 (20+2-3) to obtain the average contribution per additional distance restraint.

Sample Calculations for Sphere-Sphere Docking

Here, we present a sample calculation for sphere-sphere docking based on the derivations above.

$$N_0 = (4\pi r_1^2)(4\pi r_2^2)\ ,\quad H(S_0) = \log_2 N_0 \tag{5.28}$$

$$N_1 = 2\pi\ ,\quad H(S_1) = \log_2(2\pi) \tag{5.29}$$

$$I(\Psi : T') = \log_2(16\pi^2 r_1^2 r_2^2) - \log_2(2\pi) \tag{5.30}$$

$$= \log_2\left(\frac{16\pi^2 r_1^2 r_2^2)}{2\pi}\right) = \log_2(8\pi r_1^2 r_2^2) \tag{5.31}$$

For $r_1 = r_2 = 1$, Equation (5.31) yields $\log_2(8\pi) \approx 4.65 \text{ bits}$ for the 18 additional distance restraints. We can separate the contribution of the distance restraints into three classes: (1) simplex enforcing, (2) center-center, and (3) patch-patch.

It should also be noted that if a method for binary docking is producing data with greater information theoretic entropy than the prior background distribution for a given representation, then conditioning the background with this data is not informative.

Patch-Patch Contact Restraints for Anisotropic Subunit Docking

For rod-sphere docking, the orientation-dependent rod will be considered fixed. Note that in this case, unlike the sphere-sphere docking case, the patch-patch contact point is not necessarily collinear. In this type of docking, the only degrees of freedom in the system are around the axis defined by the sphere center and the patch-patch contact point.

For rod-rod docking, we fix the first rod and consider the second rod free. If possible, we choose the larger of the two rods to fix. Similarly to the rod-sphere docking case, we can define the free axis of rotation from the center of the second rod to the patch-patch contact point.

The differences between the anisotropic docking cases and the sphere-sphere docking come in the form of different surface area calculations based on potential steric clashes (*i.e.*, violating excluded volume). In the rod-sphere docking case, we find that we do not need to make any special considerations as no rotation of the sphere along the free axis can cause a steric clash. However, in the rod-rod docking case, we must define the free axis of rotation to be normal to both surfaces. If we modify the rod-shaped subunits such that the ends of the rods become half-spheres that cap a cylinder, this simplifies the calculation for co-normal axes. This resolves the issue of excluded volume by preventing steric clashes. This formulation for the anisotropic cases simplifies the calculations for $H(S_1)$ such that all of the cases have the same for that follows from Equation (5.16), and yielding the same a priori value as in Equation (5.17).

The resolution that should be applied scales the factors for the integrals. For the same geometric object, as the discretization of space becomes finer, the uncertainty increases. Said another way, to resolve a structure to a higher resolution, more information is required.

Patch-Patch Contact Restraints with Relative Orientations

We now add the relative orientation information to the orientation-dependent subunits, requiring $4n-10$ total distance restraints to completely specify the relative orientation of the subunits (*i.e.*, to reduce the configurational entropy to zero). Using a 3-simplex, a mass-center and a patch contact point, we have six points that we use to specify a subunit. To internally restrain an individual subunit requires 14 distance restraints. To restrain two subunits and their relative orientations requires 28 distance restraints (14 internal restraints per subunit), and the number required to enforce patch-patch contact and fully specify the specific relative orientation. From the previous section, we find that 38 total distance restraints are required, leaving 10 distance restraints for the relative orientation of the two subunits.

Now we calculate the average information content of the 38 total distance restraints, $I(\Psi : r_d, Q, E)$, where Ψ is the distribution of models consistent with our definition of the subunits and the contact criterion, r_d is the center-center distance restraint, *Q* is the patch-patch contact restraint, and *E* is the relative orientation restraint. The total *a priori* uncertainty for the distribution, S_2, for the states of the system is given by:

$$H(S_2) = \log_2(k_p(SA_1SA_2)k_{r_0}(2\pi)) \tag{5.1}$$

Here, k_p is the patch-patch contact resolution scaling factor and k_{r_0} is the relative orientation scaling factor. For example, for two spheres $(r_1 = r_2 = 1)$, and all scaling factors equal to 1, we find the *a priori* uncertainty to be:

$$H(S_2) = \log_2((4\pi 4\pi)2\pi) = \log_2(32\pi^3) \approx 9.95 \text{ bits} \tag{5.2}$$

Then for 38 distance restraints, for this system, with the specified scaling factors, we find the average information content per distance restraint to be ~0.26 bits. Further dissection could be carried out by systematically adding degrees of freedom (*i.e.*, removing distance restraints) and comparing the resulting increase of entropy.

5.2.3. Generalized Binary Docking

Here, we generalize the results above to cover arbitrary shapes, any number of patches, any shape of patches, and over any resolution scale. Without knowing where the patches are, the uncertainty over some granularity of surface mesh is determined by the resolution of the representation. In the all-atom case, the molecular surface binned at some resolution might be appropriate.

In the event that several possible patches are known and are distinguishable, the support set for the uncertainty calculation becomes the product of both sets of patches on the two subunits. This notion of distinguishable patches needs to encapsulate the possibility of overlapping, disjoint or otherwise irregular patches. Finally, given a pair of patches, the uncertainty is similar to some resolution dependent scaling of 2π, but must now also account for gapped fits, tilting or skewing in three dimensions and imperfect overlaps.

The methodology set forth above for the simplified subunit descriptions still applies to this generalized definition of subunits. Here, as before, subunits are defined by a set of points and a set of distance restraints fixing the relative positions of these points. For docking, we are now concerned with the surfaces of the subunits. To study the interaction patches, we further specify as many points as necessary to define the patches and add as many distance restraints as required to fix the relative position of each patch-defining point with respect the points that define the respective subunit. In the most general case, we distribute a set of surface mesh points evenly distributed at a density sufficient to capture differences in configurations at the desired resolution. Patches are then defined as subsets of points on this surface mesh. For all-atom models, surface residues (or atoms there of) can be thought of as generating such a surface mesh.

Now, we consider the total configuration entropy of a system with two such subunits. The number of patch-patch interactions is governed by the surface area of the two subunits and their respective shapes and excluded volumes. In general, for each surface point on one subunit there are some fraction of all the surface points on the other that are accessible to it and the sum over all possible sets of mutually accessible points between the two surfaces represents the total *a priori* patch-patch configurational entropy, H_{pp}. Assuming a uniform probability distribution over the support set, we find:

$$H_{pp} = \log_2 \left(\sum_i \sum_j \delta_a(i, j) \right) \tag{5.3}$$

Here, *i* and *j* are the support set representing the surface points of the two subunits, and $\delta_a(i, j)$ is a delta function that has value 1 if *i* and *j* are mutually accessible and 0 if not.

Then, for all possible patch-patch interactions, we consider the configurational entropy from the relative orientation between the two subunits. Again, we apply a resolution dependent discretization. For the case of strict point-point contact (*i.e.*, neglecting any skewing, tilting), the relative orientation can be expressed as a rotation about the co-normal axis. Generalizing the definition of contact beyond a point-point contact by allowing for fuzzy contacts will also account for "off-axis" skewing or tilting along different axes of rotation. So for a given patch-patch contact, we represent this generalized off-axis configurational entropy, H_{oa}, as proportional to the surface area of a sphere, less the excluded volume arising from steric clashes.

$$H_{oa} = \log_2\left(\sum_s \delta_s(s)\right) \tag{5.4}$$

Here, s is the support set representing the discretization of a sphere around a contact, and δ_s is a delta function that has value 1 if allowed by steric considerations and 0 otherwise.

Now, for each co-normal or off-axis contact, we consider the rotational degree of freedom in the system. Again, we apply a resolution dependent discretization, arbitrarily fix one of the subunits, and rotate the other subunit one full revolution. The total *a priori* rotational configurational entropy, H_{rot} can be expressed by:

$$H_{rot} = \log_2\left(\sum_r \delta_r(r)\right) \tag{5.5}$$

Here, r is the support set representing the discretization of a sphere around a contact, and δ_r is a delta function that has value 1 if allowed by steric considerations and 0 otherwise.

Finally, we combine all of these terms to express the total *a priori* configurational entropy in the system.

$$H_c = \log_2\left(\sum_{i\in A}\sum_{j\in B}\left(\delta_a(i,j)\left(\sum_{s_{ij}\in S_{ij}}\delta_s(s_{ij}\left(\sum_{r\in R(S_{ij})}\delta_r(r)\right)\right)\right)\right) \tag{5.6}$$

The multiplicity of solutions, *W*, is inside of the log expression and expresses all of the possible valid binary docking results for the subunits A and B. Now abstracting, we can think of H_{pp}, from Equation (5.3), as proportional to the product of the surface areas of the two subunits, giving:

$$H_{pp} = \log_2(SA_A SA_B d_{pp} v_{pp}) \tag{5.7}$$

Here d_{pp} is the discretization constant for patch-patch contact and v_{pp} is the fraction of the total number of possible contacts that is valid.

We can think of H_{oa}, from Equation (5.4), as being proportional to the surface area of possible off-axis contacts with the two subunits, yielding:

$$H_{oa} = \log_2(4\pi d_{oa} v_{oa}) \tag{5.8}$$

Here d_{oa} is the discretization constant for off-axis contacts and encapsulates the fuzzy area of contact, and v_{oa} is the fraction of the total number of possible contacts that is valid.

We can think of H_{rot}, from Equation (5.5), as being proportional to the circumference of a circle represent possible rotations, giving:

$$H_{rot} = \log_2(2\pi d_{rot} v_{rot}) \tag{5.9}$$

Here d_{rot} is the discretization constant for rotation and encapsulates the radius of rotation, and v_{pp} is the fraction of the total number of possible rotation that is valid.

Finally, this allows us to express H_c, from Equation (5.6) as:

$$H_c = \log_2(SA_A SA_B d_{pp} v_{pp} 4\pi d_{oa} v_{oa} 2\pi d_{rot} v_{rot}) \tag{5.10}$$

Folding all of the constants together into, *K*, yields:

$$H_c = \log_2(SA_A SA_B K) = \log_2(SA_A) + \log_2(SA_B) + \log_2(K) \tag{5.11}$$

Now, we could attempt to estimate *K* and its constituent constants, but all we really need is an estimate of the relative size in bits of $\log_2(K)$ and $\log_2(SA_A SA_B)$. We can determine an upper-bound for *K* as follows:

$$K = d_{pp} v_{pp} 4\pi d_{oa} v_{oa} 2\pi d_{rot} v_{rot} \tag{5.12}$$

We make the simplifying assumption, for the purposes of calculating an upper-bound that all possible degrees of freedom are valid, (*i.e.*, $v_{pp} = v_{oa} = v_{rot} = 1$). This simplifies Equation (5.12) to:

$$K = d_{pp} 4\pi d_{oa} 2\pi d_{rot} = 8\pi^2 d_{pp} d_{oa} d_{rot} \tag{5.13}$$

Now, all of the remaining factors in *K* are resolution dependent scaling factors. One approach to dealing with these scaling factors is to express them in terms of the surface area of the two subunits. If we decompose the three remaining constants in terms of the contributions from each subunit, we can calculate scaled surface area terms, by some function which scales the multiplicity based on the surface area of each subunit. Using this scaling function, denoted by the hat symbol above the term, upper-bound for the total configurational entropy, H_c^*, can be expressed as:

$$H_c^* = \log_2(\widehat{SA}_A) + \log_2(\widehat{SA}_B) \tag{5.14}$$

Extending this idea of a scaling function, we also define a scaled radius of off-axis contact and scaled radius of rotation, also denoted by a hat above the term. From this, we can give the upper-bound expressions for the patch-patch, off-axis, rotation and total configurational entropies.

$$H_{pp}^* = \log_2(\widehat{SA}_A \widehat{SA}_B) \tag{5.15}$$

$$H_{oa}^* = \log_2(4\pi \hat{r}_{oa}^2) \tag{5.16}$$

$$H_{rot}^* = \log_2(2\pi \hat{r}_{rot}) \tag{5.17}$$

$$H_c^* = H_{pp}^* + H_{oa}^* + H_{rot}^* \tag{5.18}$$

Here we apply these expressions for generalized binary docking to the case of spheres. We can determine the scaled surface area of a sphere by:

$$\widehat{SA}_A = 4\pi \hat{r}_A^2 \tag{5.19}$$

We can also derive a radius scaling expression that is a function of both of the subunit radii:

$$\hat{r} = \frac{r_A r_B}{r_A + r_B} \tag{5.20}$$

So with two spheres, with $r_A \geq r_B$, we have an upper bound on the total configurational entropy of:

$$H_c^* = 7\log_2(\hat{r}) + 4\log_2(\pi) + 7 \tag{5.21}$$

This reveals an inherent, constant uncertainty of $4\log_2(\pi) + 7 \approx 13.6$ bits and leaves only the first term that is dependent on the size of the spheres.

Now we briefly address the scaling factor. For two spheres, with $r_A \geq r_B$, we calculate an effective tiling factor, *x*, required to achieve an error in measurement of *y*. We can also use this to account for thermal noise or any other source of imprecision.

$$\sin\alpha = \left(\frac{y}{2}\right)\left(\frac{1}{r_A + r_B}\right) \tag{5.22}$$

$$x = \left(\frac{y}{2}\right)\left(\frac{r_A}{r_A + r_B}\right) \tag{5.23}$$

Then the effective radius (the scaled radius) of r_A becomes r_A / x. For a sample calculation, taking $r_A = r_B = r$, *y* = 1Å, and *x* = *y/4* = ¼ Å, yields $\hat{r} = 4r / 1$Å. If we assume that globular proteins are approximately sphere, then for an average radius of a globular protein of 16Å, with 1Å tolerance, we can calculate an estimate of the total configuration entropy for two average interacting proteins. With this scaling factor, we find that $\hat{r} = 64$. Substituting this result into Equation (5.21) yields ~55.6 bits of uncertainty.

5.3. Analysis of Restraints for Multiple Docking

In the previous section, we performed an information theoretic analysis of the configurational entropy of binary docking based on the geometry of the subunits. Here, we will analyze the restraints used for binary docking. Specifically, we will be analyzing the Multidimensional Gaussian (MDG) / exclusive-OR (XOR) restraints, described in the Materials and Methods of the work on modular assessment.

5.3.1. Restraint Definition

This spatial restraint is used to encode several possible configurations for two subunits. For each possible configuration, we extract the representative distances, translate these distances into distance restraints, and merge these distance restraints into a single MDG restraint. We further combine each of these MDG restraints into a weighted XOR restraint. This weighted XOR restraint determines which of the possible configurations is most satisfied and current configuration of the model, and then enforces the corresponding underlying MDG restraint (Figure 5.9).

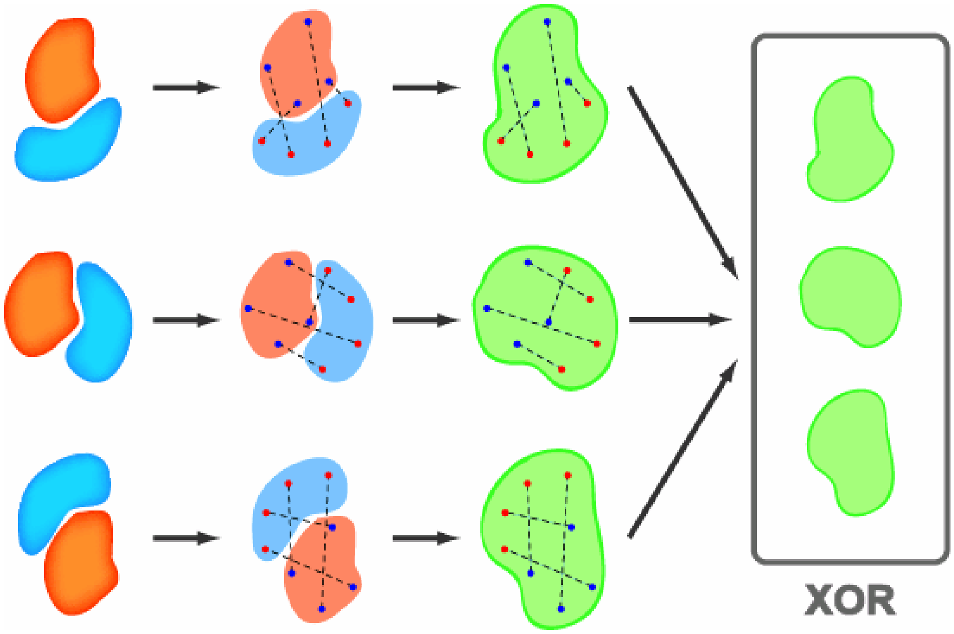

Figure 5.9. Multidimensional Gaussian, Exclusive OR Restraint

5.3.2. Analysis of Single MDG/XOR Restraint

For a single MDG/XOR restraint *x*, let *n* be the total number of possible configurations, let *w* be the relative weight of the target configuration and let all other configurations have weight 1, and let $t = w + (n-1)$. The entropy contained in the restraint is given by:

$$H(x_w) = -\left(\frac{n-1}{t}\right)\log_2\left(\frac{1}{t}\right) - \left(\frac{w}{t}\right)\log_2\left(\frac{w}{t}\right) \tag{5.24}$$

This expression simplifies to:

$$H(x_w) = \log_2 t - \left(\frac{w}{t}\right)\log_2 w \tag{5.25}$$

Let Δw be some positive increment to the target weight. Let $w^* = w + \Delta w$ and $t^* = t + \Delta w$. Substituting w^* and t^* into Equation (5.24) yields:

$$H(x_w^*) = -\left(\frac{n-1}{t^*}\right)\log_2\left(\frac{1}{t^*}\right) - \left(\frac{w^*}{t^*}\right)\log_2\left(\frac{w^*}{t^*}\right) \tag{5.26}$$

This expression simplifies to:

$$H(x_w^*) = \log_2 t^* - \left(\frac{w^*}{t^*}\right)\log_2 w^* \tag{5.27}$$

Subtracting Equation (5.24) from Equation (5.26) gives us the information content of increasing the weight of the target configuration:

$$I(x_w : x_w^*) = H(x_w) - H(x_w^*) \tag{5.28}$$

$$I(x_w : x_w^*) = \log_2(t / t^*) + \left(\frac{w^*}{t^*}\right)\log_2 w^* - \left(\frac{w}{t}\right)\log_2 w \tag{5.29}$$

From Equation (5.28) we can show that improving the weight of the target configuration always has a strictly positive affect on $I(x_w : x_w^*)$.

5.3.3. Analysis of Multiple MDG/XOR Restraint

For two independent MDG/XOR restraints with $w_1 = w_2 = w$, $n_1 = n_2 = n$, and therefore, $t_1 = t_2 = t$. We can write the total information theoretic entropy by:

$$H(x_{w^2}) = -\left(\frac{(n-1)^2}{t^2}\right)\log_2\left(\frac{1}{t^2}\right) - \left(\frac{2w(n-1)}{t^2}\right)\log_2\left(\frac{w}{t^2}\right) - \left(\frac{w^2}{t^2}\right)\log_2\left(\frac{w^2}{t^2}\right) \quad (5.30)$$

This can be rewritten as:

$$H(x_{w^2}) = \log_2 t^2 - \left({}^{w}\!/_{t}\right)\log_2 w^2 = 2\left(\log_2 t - \left({}^{w}\!/_{t}\right)\log_2 w\right) \quad (5.31)$$

$$(5.32)$$

For m independent MDG/XOR restraints, under the same conditions, we can write:

$$H(x_{w^m}) = m\left(\log_2 t - \left({}^{w}\!/_{t}\right)\log_2 w\right) \quad (5.33)$$

Now, we consider two independent MDG/XOR restraints with different number of choices (n, m), weights (w, v). Let $t = w + n - 1$ as before, and let $u = v + m - 1$. Now the configurational entropy of this system is given by:

$$H(x_{vw}) = \log_2 tu - \left({}^{w}\!/_{t}\right)\log_2 w - \left({}^{v}\!/_{u}\right)\log_2 v \quad (5.34)$$

For m independent MDG/XOR restraints with arbitrary number of choices and weights, we can write:

$$H(x_{\vec{w}}) = \log_2\left(\prod_{i=1}^{m} t_i\right) - \sum_{i=1}^{m}\left({}^{w_i}\!/_{t_i}\right)\log_2 w_i \quad (5.35)$$

5.3.4. Correlated Restraints

Thus far, we have only considered independent restraints and independent choices within restraints. However, any correlations either within a restraint or between restraints can only decrease the entropy. Within a restraint, any correlations between choices can not change the probability distribution over the choices to increase the entropy. Likewise, between restraints, any correlations will deviate from the sum of the underlying, more uniform, entropies, always towards decreased overall entropy. Thus, the calculations in this section provide an upper bound on the total entropy of this restraint type.

5.4. Graph-theoretic Analysis of Multiple Docking

We use a simplified model of assemblies and binary docking to study the effect of the accuracy of the underlying restraints on multiple docking. We assume the knowledge of the assembly subunit stoichiometry and connectivity (*ie* interaction topology).

5.4.1. Graph Representation

We define a graph *G*, where each of the *n* subunits is represented as nodes, and each of the contact interactions define the *m* edges (Figure 5.10b). Then we define a complete, *m*-partite graph *G'*, where each partition represents an edge (*eg*, edge *e*, spanning nodes *u* and *v*) in *G* and contains nodes representing possible relative orientations of the subunits connected by that edge (*eg*, relative orientations between subunits *u* and *v*) (Figure 5.10c). Each edge in *G'* now represents a choice of two relative orientations between three connected subunits. On each edge in *G'*, we assign edge weights as the acceptance probability of that particular choice of relative orientations. Within each partition in *G'*, we can assign a selection probability of a particular relative orientation. We further augment each partition with an additional "failure" node, representing situations where the presumed contact is either not made or the excluded volume of the two subunits is violated. This failure node has a selection probability of zero, and any edges to or from a failure node has an acceptance probability of zero.

To fully specify the interaction topology, we need to capture all of the interactions presumed to exist, which is done in *G'*, but we also need to capture all of the interactions which are thereby forbidden. The inverse contact interaction topology, *H*, is given by the complement of *G* (Figure 5.10d). Using the edges in *H*, we can now create a set of partitions that represent interactions that should not exist. Each of these partitions contains only two nodes: valid (with selection probability of one) and invalid (with selection probability of zero). In this case, the valid nodes represent a non-contact, and the invalid nodes represent either a contact or a collision and have the same properties as the failure nodes in *G'*. We define a new graph *G''*, which is a complete $\binom{n}{2}$-partite graph over both contacting interactions and non-contacting interactions (Figure 5.10e). Edge weights are defined the same way as in *G'*, with the addition that edge weights between valid nodes and all other nodes that are not failure nodes or invalid nodes are defined as one.

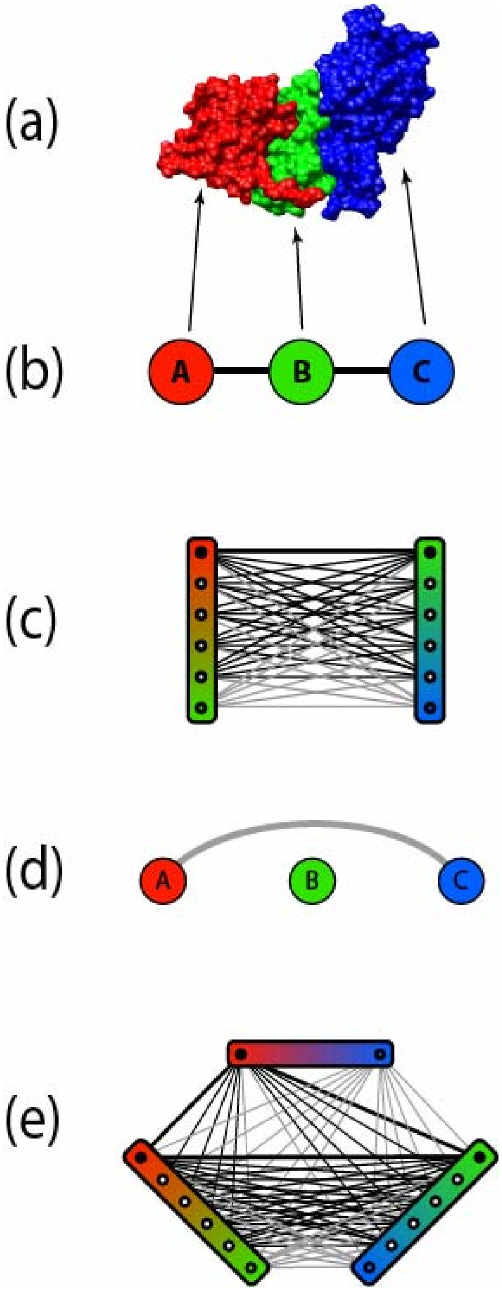

Figure 5.10. Graph-theoretic Model for Multiple Docking

5.4.2. Calculation of Configuration Probabilities

After defining the specific parameters of our model, we can write an expression for the probability of finding any structure. In either *G'* or *G''*, we calculate the raw probability of a configuration, by taking the product of the selected selection probabilities and the product of the associated acceptance probabilities. The normalized probability is given by the raw probability divided by the sum over all possible raw probabilities.

5.4.3. Analysis of Contributions to Configurational Entropy Loss

With *G'* and *G''*, we can calculate the contributions that excluded volume, topological enforcement and the various weights of the selection and acceptance probabilities have on the difficulty of finding the native structure by comparing the normalized probabilities of the native structure, p_{native}, under various regimes. The difference in p_{native} between *G'* and *G''*, represents the added information gained from negative interaction data. And by varying the selection and acceptance probabilities, we can survey regimes for synergistic multiple docking. Finally, we can vary the underlying interaction topology, *G*, by adding or removing edges and the resulting differences in p_{native} represents the information from a particular topology.

For example, if we take a complex formed of three subunits (A, B and C) that are arranged in a linear chain, such that A and B are in contact and B and C are in contact, we can use this graph theoretic model to describe the relationship between the various selection probabilities and their effect on the overall probability of finding the native structure. For the binary interaction between A and B, let *j* represent the total number possibilities and *a* represent the probability of selecting the native configuration. For the binary interaction between B and C, let *k* represent the total number possibilities and *b* represent the probability of selecting the native configuration. The probabilities for selecting a non-native configuration are: ${(1-a)}/{(j-1)}$ and ${(1-b)}/{(k-1)}$ for the AB and

BC binary interactions, respectively. We can express the probability of selecting the native structure, p_{native}, in this graph theoretic model by the following expression:

$$p_{native} = \frac{ab}{ab + \frac{na(1-b)}{k-1} + \frac{mb(1-a)}{j-1} + \frac{c(1-a)(1-b)}{(j-1)(k-1)}} \tag{5.36}$$

Here *n* is the number of interactions between the native configuration for AB and a non-native configuration for BC, *m* is the number of interactions between the native configuration for BC and a non-native configuration for AB, and *c* is the number of interactions between non-native interactions for both AB and BC. To simplify our notation, let $\beta = j-1$ and let $\gamma = k-1$. Solving for $p_{native} > {}^{1}\!/_{2}$:

$$p_{native} = \frac{ab}{ab + \frac{na(1-b)}{\beta} + \frac{mb(1-a)}{\gamma} + \frac{c(1-a)(1-b)}{\beta\gamma}} > {}^{1}\!/_{2} \tag{5.37}$$

If we consider the case where $a = b = w$ and $j = k$, $\beta = k-1$, we can solve a particular instance of the problem for *w*. After substituting for the redundant variables, we rewrite Equation (5.37) as:

$$p_{native} = \frac{w^2}{w^2 + \left(\frac{n+m}{\beta}\right)\left(w(1-w)\right) + \frac{c(1-w)^2}{\beta^2}} > {}^{1}\!/_{2} \tag{5.38}$$

Solving for *w*, the native probability required to have $p_{native} > {}^{1}\!/_{2}$, we arrive at the expression:

$$w > \frac{(\beta(n+m)+2c) \pm \sqrt{(\beta(n+m)+2c)^2 - 4c(\beta^2 + \beta(n+m)+c)}}{2(\beta^2 + \beta(n+m)+c)} \tag{5.39}$$

If we allow $X = \beta(n+m)+2c$ and $Y = \beta^2\beta(n+m)+c$, Equation (5.39) simplifies to:

$$w > \frac{X \pm \sqrt{X^2 - 4cY}}{2Y} \tag{5.40}$$

Solving for *w*, we only keep the positive root. This allows us, for this simplified model, to analytically determine the relative weight need on the native configuration to be able to definitely select the native structure of the complex using multiple docking. It also describes the relationship between the number of choices, the required accuracy of the underlying restraints and the overall accuracy of the method.

5.4.4. Assessment Metrics

In assessing the accuracy of underlying binary docking methods for multiple docking, the metric of choice may not be the configuration entropy of the resulting distributions, but rather a quantity related to p_{native} or expected rank of the native (which in unweighted context are equivalent).

Given underlying binary docking data represented in partitions with selection probabilities (and negative interactions also represented exclusion probabilities), we can generate all possible valid solution states (those with non-zero probabilities). Here, the raw and normalized values of p_{native} can be calculated and also the entropy of the distribution can be calculated. If we rank order the valid configuration by probability, we can also find thresholds for the expected rank of the native solution.

This third measure is of greater practical value as the expected number of follow-up experiments is the applicable metric of interest. Here, the entropy of the distribution conditioned by the knowledge of p_{native} or of the rank ordering might be of theoretical value, but from a practical standpoint, since every valid solution is an alternative that needs to be eliminated, an unweighted "count" based measure is entirely sufficient. Another potential metric might be the entropy over the support set of candidate solutions or candidate experiments (*i.e.*, the number of actual experiments needed to determine the correct structure).

5.5. Revealing Synergistic Regimes for Multiple Docking

Here we use simplified models for multiple docking to describe the relationship between the overall accuracy of multiple docking and the synergistic combinations of binary docking results. By describing the effect of one binary docking prediction on another, we will be able to describe a lower limit on the accuracy and precision required from binary docking methods to be useful for multiple docking.

5.5.1. Representation of Assemblies

We use a simplified model of assemblies and binary docking to study the effect of the accuracy and precision of the underlying restraints on multiple docking. We represent the *n* subunits of the assembly as hard spheres of the same size.

5.5.2. Representation of Binary Docking

Docking results are expressed as probability distributions over the possible adjacencies (*i.e.*, contacts or dockings) for two subunits. We can vary *k*, the number of alternative positions for each docking result, as well as p_{native}, the probability of the native position. We define a uniform probability distribution over the non-native, ie decoy, positions. The probability for a decoy, p_{decoy}, is given by $p_{decoy} = \frac{1 - p_{native}}{k-1}$.

5.5.3. Calculation of Native Structure Probability

After defining the specific parameters of our model, we can write an expression for the probability of finding any structure, accounting for exclude volume. For *n=3* subunits, with subunits A and B, and B and C interacting, we can define *p(AB)* and *p(BC)* as either p_{native} or p_{decoy} for the respective interactions. Then the probability of the structure is given by:

$$p_{structure} = \frac{p(AB)\,p(BC)}{p_{native}{}^{2} + 2(k-2)\,p_{native}\,p_{decoy} + (k^{2} - 3k + 3)\,p_{decoy}{}^{2}} \tag{5.41}$$

5.5.4. One-Dimensional Model System

As a proof of concept, we describe a simple model system of three subunits in a one-dimensional system. We describe a native structure as having the structure ABC and the decoy structure as having the structure CBA. We can encode the rules for this one-dimensional system using probabilities.

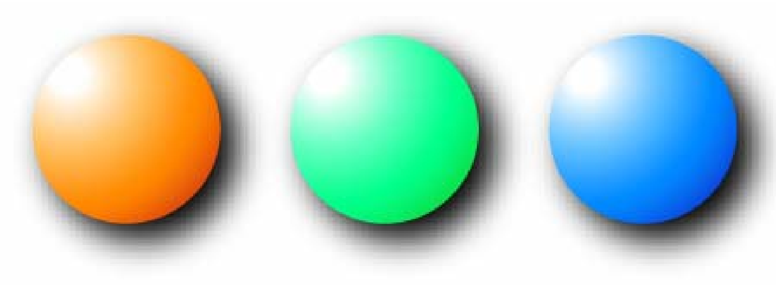

Figure 5.11. One-Dimensional Model, Native Structure

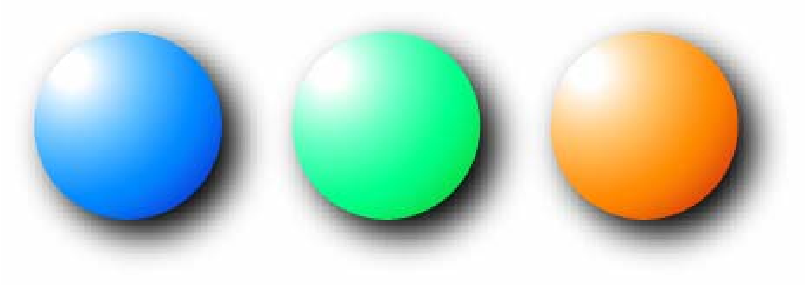

Figure 5.12. One-Dimensional Model, Decoy Structure

$$\begin{aligned} p(AB) &= 1 - p(BA) \\ p(BC) &= 1 - p(CB) \\ p(AC) &= p(CA) = 0 \end{aligned} \tag{5.42}$$

Building on these rules, we can write a simplified version of Equation (5.41) for the p_{native} of this particular system (Figure 5.13):

$$p_{native} = p(ABC) = \frac{p(AB)\,p(BC)}{p(AB)\,p(BC) + p(CB)\,p(BA)} \tag{5.43}$$

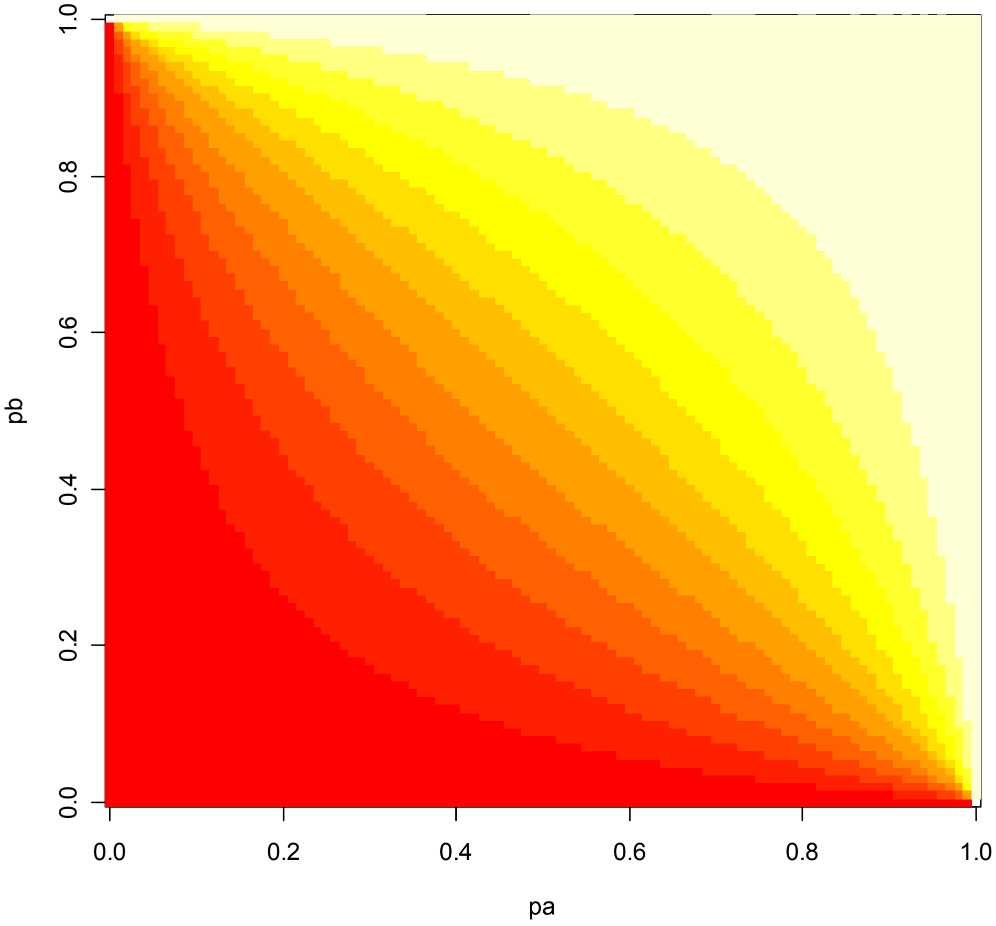

Figure 5.13. Synergistic Behavior

We can compare this to the probability of finding a native solution if we considered the two underlying probabilities independent (Figure 5.14):

$$p_{independent}(ABC) = p(AB)\,p(BC) \tag{5.44}$$

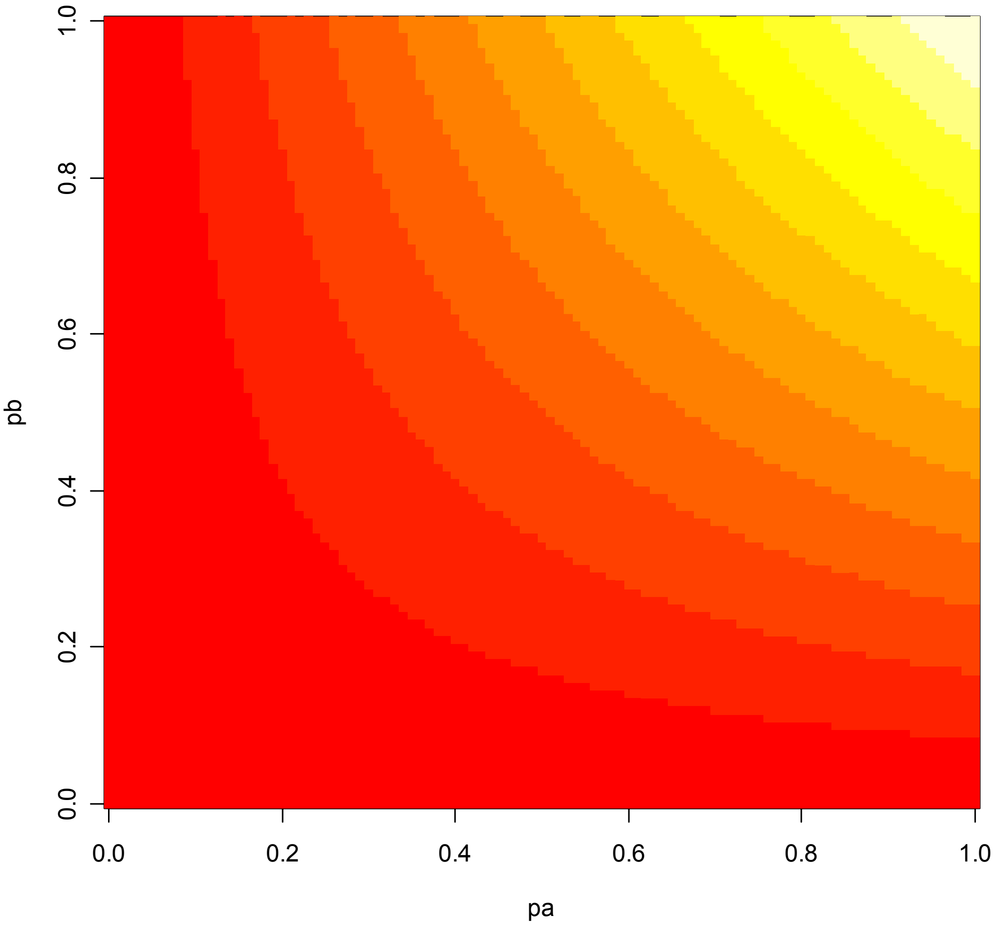

Figure 5.14. Independent Behavior

Now, we plot p_{native} with respect to the underlying binary probabilities (Figure 5.15). As we can see, the model displays synergistic properties.

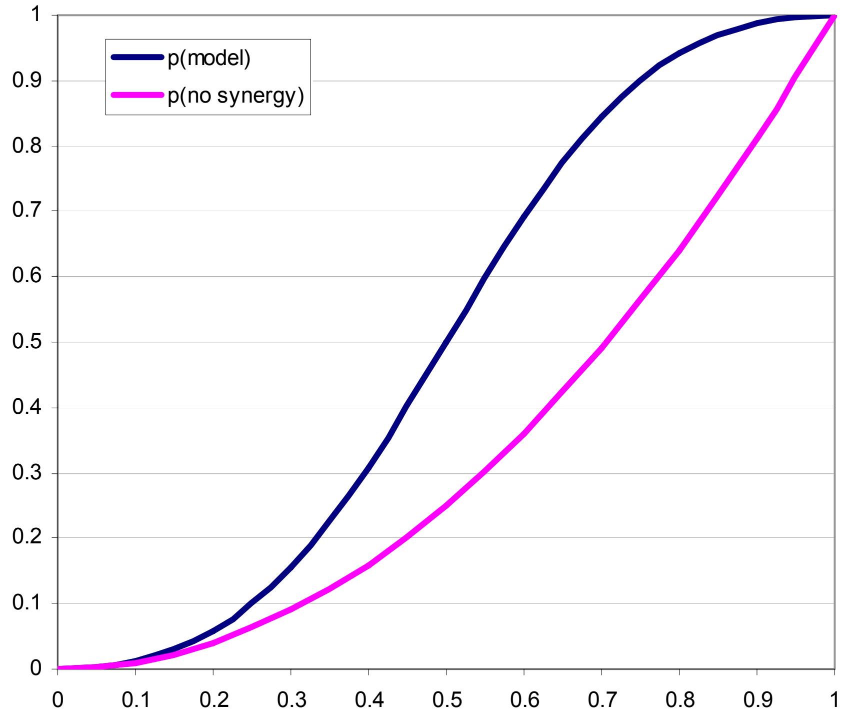

Figure 5.15. Synergistic and Independent Model Comparison

5.5.5. Dimensions and Adjacencies are Equivalent

To extend the basic idea of this model to higher dimensions, only requires increasing the possible number of adjacencies. Here we extended the one-dimensional model to a two-dimensional square lattice by fixing the position of subunit B to the center of a 3x3 lattice. Now, we can write the p_{native} of this particular system:

$$p_{native} = p(ABC) = \frac{p(A = A_{native})\, p(B = B_{native})}{\sum_{i=1}^{n} \sum_{j=1}^{n} (1 - \delta_{ij})\, p(A = i)\, p(B = j)} \qquad (5.45)$$

Note that this expression has no dimensional dependence; the only parameter is the number of possible adjacencies. In general, the maximum possible number of adjacencies for any number of subunits is governed by the closest possible packing of those subunits. For hard spheres of identical size: in one dimension, the maximum possible number of adjacencies is 2 subunits; in two dimensions, the maximum possible number of adjacencies is 6 subunits; and in three dimensions, the maximum possible number of adjacencies is 12 subunits.

Using such an expression, we can describe the relationship between number of possible adjacencies and p_{native} for a particular topology and probability for the underlying binary docking results to find the native positions or configurations.

5.5.6. Excluded Volume and Synergy in Multiple Docking

Using the extension of the model described above, we described the effect of increasing the number of adjacencies on the synergy in multiple docking. As the number of possible adjacencies rises, the synergy found in combining multiple binary docking results decreases. This can also be interpreted in terms of excluded volume: as the number of possible positions for each subunit rises, there are fewer possible ways for excluded volume to be violated.

5.6. Sampling and Scoring in Underdetermined Systems

In structure modeling by representing systems as points and restraints, and optimizing the configurations with respect to input data, it can be observed that in underdetermined systems, multiple global optima exist. Selecting the largest clustering set of optima may be representative of the native state, but this is not always certain. This is observed, when dealing with restraints that are not physical in nature or without sufficient sampling. Furthermore there is the complication of having correlated restraints and biased sampling. Optimization methods generally perform biased sampling. Here using simple model systems, we describe the relationship between underdetermined systems, and sampling and scoring. In particular, we focus on correlated restraints and optimization bias.

5.6.1. Preliminary Demonstration of Underdetermined Systems

The multiplicity of solutions in an underdetermined system is a consequence of multi-body interactions where some solutions (not necessarily native) are favored in the subsequent optimization. For example, in a system with n points and less than $4n - 10$ non-redundant distance restraints, there are multiple satisfying solutions to the configuration of points.

We can also construct an underdetermined system with a subset of points that is fully specified in such a way as to make particular classes of solutions easier to find and others more difficult.

5.6.2. Scoring and Information Sufficiency

Given a particular representation of a system, we translate the available data as restraints which are then encapsulated into a scoring function. The support set for this scoring function is the space of all possible configurations of the system. In a completely, uniquely determined system, over this support set, the scoring function defines a single optimal solution. However, in an underdetermined system, multiple solutions satisfy the input restraints and receive the same optimum score by the scoring function. Therefore, the scoring function is incomplete in the sense that there was insufficient information encoded in the restraints to completely determine a unique solution.

5.6.3. Ranking Solutions in Underdetermined Systems

For underdetermined systems, with incomplete scoring functions, we have multiple solutions and a need to be able to distinguish a representative solution from the ensemble of models with optimum score.

For the case of satisfiable restraints, without additional qualification, all of the models of optimum score (*i.e.*, without violation) are equally likely and satisfy all of the input restraints. We have no guarantee of completeness of sampling over model space, nor can we declare a particular model or representative model to be correct. The largest basin of models represents the solutions that satisfy all input restraints and are most often found by the optimization protocol used.

5.7. Bayesian Network Framework for Structure Modeling

Here we seek to describe the integration of multiple sources of structural data for modeling macromolecular structures by the use of a Bayesian network. This model of data integration is general enough to be applied to any problem in structure modeling, and aims to express in a unified probabilistic framework all the information that we can collect about macromolecular structure. For this work, we use the protein structure prediction problem as an example.

5.7.1. Joint Probability Density Function

Prediction of the native structure of a protein would be enabled by expressing our knowledge of any kind as a scoring function whose global optimum corresponds to the native structure. One such function is a joint probability density function of the coordinates of the *n* protein atoms, *R*, given available information *I* about the system.

$$p(\vec{R} \mid I) = p(R_1, R_2, R_3, \ldots, R_n \mid I) \tag{5.46}$$

We have chosen to approach the problem of building the joint probability density using a Bayesian network. Here we describe the construction of a Bayesian network representing the structure modeling problem, and describe the conditional independences that arise from the network. This description will allow us to calculate the joint probability density by multiplying (or adding in log space) the conditionally independent features together.

First, we enumerate all of the variables, which are represented as nodes in the network. As described above, we have the variable *R*, representing the coordinates of the atoms in the model. We represent the sequence of the protein being modeled by the variable *S*, which contains the sequence of residues forming the protein, information about the atom and residue types. We represent the environment of the protein by the variable *W*, which encapsulates quantities like temperature, pressure, pH, *etc.* We use the variable *C* to encapsulate the data from a set of sample structures that are already known. This set of structures might provide us with homology information or serve as the basis for a statistical potential. A typical source for *C* might be the PDB. We represent experimental evidence with the variable *X*. Each independent experiment gives us direct or indirect data on the coordinates. We represent all scoring functions on the positions of the model, such as molecular dynamics force fields, using the variable *E*. For this reason, *E* also encapsulates the value of the "energy" of the system (*i.e.*, the value that we observe after applying the force field function to the coordinates). For any given force field, in *E*, we represent the parameters to that force field with the variable *P*. Finally, we have the variable *Q*, representing the acceptance or rejection criteria for the models (*e.g.*, only accepting models without any steric clashes).

5.7.2. Bayesian Network Construction

With the nodes described above, we construct a Bayesian network to describe the protein structure modeling problem (Figure 5.16).

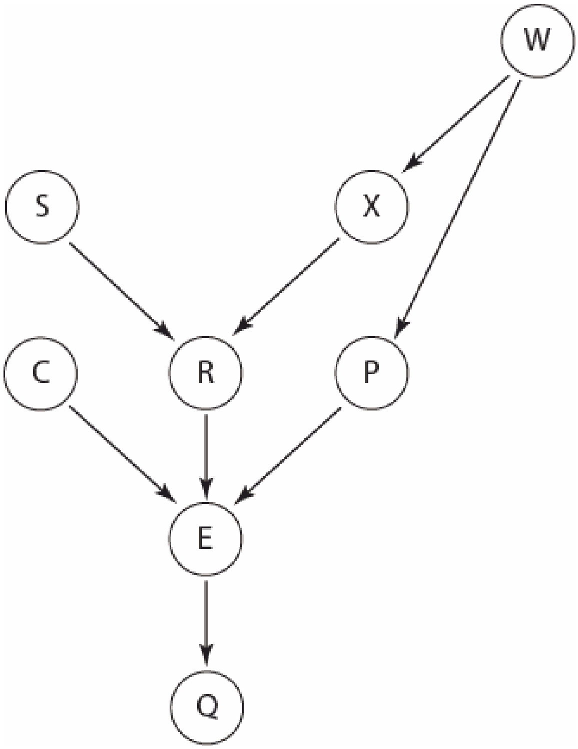

Figure 5.16. Bayesian Network for Protein Structure Modeling

The nodes S, W, C, X, P, and Q are observed variables and hence, evidence nodes. The coordinates, R, and consequently the value of the scoring function, E, are not observed. Now we list and justify the arcs connecting three nodes.

$W \rightarrow X \rightarrow R$. We have a chain. If we observe the results of an experiment, then the environment of the experiment and the resulting coordinates are independent.

$W \rightarrow P \rightarrow E$. We have a chain. If we observe the parameters to an energy function, then the environmental factors influencing the choice of parameters and the resulting energy function are independent.

$S \rightarrow R \leftarrow X$. We have a collider. If we observe the coordinates of the model, then the sequence and the experiments are dependent. Both the sequence and the experiments have effects on our model coordinates.

$S \rightarrow R \rightarrow E$. We have a chain. If we observe the coordinates of the model, then the sequence and the resulting value of the energy function are independent.

$X \rightarrow R \rightarrow E$. We have a chain. If we observe the coordinates of the model, then the experimental evidence and the resulting value of the energy function are independent.

$C \rightarrow E \leftarrow R$. We have a collider. If we observe the value of the energy function, then homology information and the coordinates of the model are dependent. Both the homology information and the coordinates of the model have effects on the value of the energy function (assuming it uses homology derived terms).

$C \rightarrow E \leftarrow P$. We have a collider. If we observe the value of the energy function, then homology information and the parameters to the energy function are dependent. Both the homology information and the parameters to the energy function have effects on the value of the energy function (assuming it uses homology derived terms).

$C \rightarrow E \rightarrow Q$. We have a chain. If we observe the value of the energy function, then the homology information and the value acceptance criteria are independent.

$R \rightarrow E \leftarrow P$. We have a collider. If we observe the value of the energy function, then the coordinates of the model and the parameters to the energy function are dependent. Both the coordinates of the model and the parameters to the energy function have effects on the value of the energy function.

$R \rightarrow E \rightarrow Q$. We have a chain. If we observe the value of the energy function, then the coordinates of the model and the acceptance criteria are independent.

$P \rightarrow E \rightarrow Q$. We have a chain. If we observe the value of the energy function, then the parameters to the energy function and the acceptance criteria are independent.

5.7.3. Calculating Joint Probability Density Functions

The joint probability density function of a Bayesian network is given by:

$$p(X_1, X_2, \ldots, X_n) = \prod_{i=1}^{n} p(X_i / parents(X_i)) \tag{5.47}$$

According to the graph (Figure 5.16) we constructed above, the joint probability density function is:

$$p_{joint} = p(Q \mid E)\, p(E \mid CRP)\, p(C)\, p(R \mid SX)\, p(P \mid W)\, p(S)\, p(X \mid W)\, p(W) \tag{5.48}$$

Now if we apply the chain rule to solve for $p(R \mid WSXCPEQ)$, we get:

$$p(R \mid WSXCPEQ) = \frac{p_{joint}}{p(WSXCPEQ)} \tag{5.49}$$

Removing R from the original Bayesian network and redirecting the arc directly from $S \rightarrow E$ and $X \rightarrow E$, we can calculate $p(WSXCPEQ)$:

$$p(WSXCPEQ) = p(Q \mid E)\, p(E \mid CPSX)\, p(C)\, p(P \mid W)\, p(S)\, p(X \mid W)\, p(W) \quad (5.50)$$

Substituting Equation (5.50) into Equation (5.49) and simplifying, we arrive at:

$$p(R \mid WSXCPEQ) = \frac{p(E \mid CRP)\, p(R \mid SX)}{p(E \mid CPSX)} \quad (5.51)$$

The formula seems to be in agreement with our intuition of the problem, as the energy depends on the information derived from homology, the parameters, and the coordinates of the model, and the coordinates of the model depend on sequence and experimental data. The denominator only includes evidence nodes, which are observed, and becomes a normalization term.

5.7.4. Implications of the Bayesian Network Analysis

Above we explicitly stated the conditional dependencies and independencies of the various factors in protein structure modeling by representing them in a Bayesian network. Now we extend this analysis to describe the correlations between coordinates in the model. If the coordinates of the model were conditionally independent of each other, given the observed variables, then all of the probabilities for the individual coordinates could be multiplied together to solve for the probability of the entire model.

Unfortunately, because we observe the acceptance criteria, represented by the variable *Q*, if we were to separate the coordinates of the model into individual nodes, they would all be conditionally dependent on each other. Given the conditional dependence of the coordinates of the models, the most principled integration of data would need to account for the correlations between positions.

5.8. Future Directions

This chapter represents work of various levels of completeness and abstractness. The future directions for this work are to address both of these issues: first, to improve the completeness and accuracy of the models; and second, to adapt and apply these models to different applied to different systems with more concrete applications.

While many different simplified representations were presented in this chapter, the practical question of choosing the optimal representation with respect to the amount data (and information contained in the data) and the completeness sampling was not address. Also along this line of inquiry is the use of multiple granularities simultaneously.

This chapter also presented derivations and calculations for several upper bounds, notably on the configurational entropy of binary and multiple docking. As with any work establishing upper bounds, further work can be done to tighten this bound. Also making increasingly realistic assumptions and recalculating the bounds might be able to explain why some assemblies are more difficult than others to solve. Another interesting question would be to explore the possible thermodynamic implications of such entropy bounds. Along these lines, having described the information theoretic entropy, we could also define a corresponding “free energy” quantity, perhaps by translating the defined entropies into relative entropies with priors weighted by a scoring function.

The work on multiple docking presumes the existence of a unique solution rather than perhaps a more general inferential approach. Expanding on this theme: the output of the analytical model for subunit assembly given topological restraints, respecting excluded volume and interaction data generates a probability distribution over the space of allowed configurations (*i.e.*, the support set). Evaluating the underlying method from an inferential framework would yield p_{native} as the only metric of relevance. Another line of inquiry could describe the relationship between adding more data (decreasing precision) and accuracy. Also, the models in this work assumed independences between the various restraints, examining the possible correlations within these restraints could also lower the bounds on the entropy of multiple docking. Using more realistic models, we could also demonstrate that multiple docking indeed follows the same behavior as the simplified models.

Another limitation in this chapter is that the simplified models used were generally of a fixed topology and composition. Describing the effect of different topologies on the models presented in this work would provide another variable with which we could calculate entropies and information contents.

For underdetermined systems, describing the specific conditions under which the largest cluster is also the native cluster would be of practical use. Another approach to pursue would be to employ active learning strategies; attempting to determine which restraint or experiment would be most informative.

Also, while underdetermined systems were addressed, the opposite case of over-constrained systems was not. These frustrated systems are important because with multiple sources of structural data, it becomes more and more likely that there will be conflicting restraints. Describing the properties of over-constrained systems and creating principled strategies for resolving conflicting data could also prove useful.

Finally, with the work started on the Bayesian network description of protein structure modeling, we could build more-principled statistical potentials and scoring functions, perhaps ultimately improving the accuracy of protein structure modeling.

References

1. Alber, F., et al., *Integrating Diverse Data for Structure Determination of Macromolecular Assemblies.* Annual Review of Biochemistry, 2008. **77**(1): p. 443-477.
2. Aloy, P., M. Pichaud, and R.B. Russell, *Protein complexes: structure prediction challenges for the 21st century.* Current Opinion in Structural Biology, 2005. **15**(1): p. 15-22.
3. Aloy, P. and R.B. Russell, *The third dimension for protein interactions and complexes.* Trends Biochem Sci, 2002. **27**(12): p. 633--638.
4. Carter, A.P., et al., *Functional insights from the structure of the 30S ribosomal subunit and its interactions with antibiotics.* Nature, 2000. **407**(6802): p. 340-8.
5. Harms, J., et al., *High resolution structure of the large ribosomal subunit from a mesophilic eubacterium.* Cell, 2001. **107**(5): p. 679-88.
6. Ban, N., et al., *The complete atomic structure of the large ribosomal subunit at 2.4 A resolution.* Science, 2000. **289**(5481): p. 905-20.
7. Zhang, G., et al., *Crystal structure of Thermus aquaticus core RNA polymerase at 3.3 A resolution.* Cell, 1999. **98**(6): p. 811-24.
8. Frank, J., *Single-particle imaging of macromolecules by cryo-electron microscopy.* Annu Rev Biophys Biomol Struct, 2002. **31**: p. 303-19.
9. Schmid, M.F., et al., *Structure of the acrosomal bundle.* Nature, 2004. **431**(7004): p. 104-7.
10. Zhang, W., et al., *Visualization of membrane protein domains by cryo-electron microscopy of dengue virus.* Nat Struct Biol, 2003. **10**(11): p. 907-12.
11. Baumeister, W., R. Grimm, and J. Walz, *Electron tomography of molecules and cells.* Trends Cell Biol, 1999. **9**(2): p. 81-5.
12. Beck, M., et al., *Nuclear pore complex structure and dynamics revealed by cryoelectron tomography.* Science, 2004. **306**(5700): p. 1387-90.
13. Young, M.M., et al., *High throughput protein fold identification by using experimental constraints derived from intramolecular cross-links and mass spectrometry.* Proc Natl Acad Sci U S A, 2000. **97**(11): p. 5802-6.
14. Chu, F., et al., *Unraveling the interface of signal recognition particle and its receptor by using chemical cross-linking and tandem mass spectrometry.* Proc Natl Acad Sci U S A, 2004. **101**(47): p. 16454-16459.

15. Malhotra, A. and S.C. Harvey, *A quantitative model of the Escherichia coli 16 S RNA in the 30 S ribosomal subunit.* J Mol Biol, 1994. **240**(4): p. 308-40.
16. Li, F., et al., *Evidence for an internal entropy contribution to phosphoryl transfer: a study of domain closure, backbone flexibility, and the catalytic cycle of cAMP-dependent protein kinase.* J Mol Biol, 2002. **315**(3): p. 459-69.
17. Rout, M.P., et al., *The yeast nuclear pore complex: composition, architecture, and transport mechanism.* J Cell Biol, 2000. **148**(4): p. 635-51.
18. Truong, K. and M. Ikura, *The use of FRET imaging microscopy to detect protein-protein interactions and protein conformational changes in vivo.* Curr Opin Struct Biol, 2001. **11**(5): p. 573-8.
19. Wells, J.A., *Systematic mutational analyses of protein-protein interfaces.* Methods Enzymol, 1991. **202**: p. 390-411.
20. Phizicky, E., et al., *Protein analysis on a proteomic scale.* Nature, 2003. **422**(6928): p. 208-15.
21. Uetz, P., et al., *A comprehensive analysis of protein-protein interactions in Saccharomyces cerevisiae.* Nature, 2000. **403**(6770): p. 623-7.
22. Ito, T., et al., *A comprehensive two-hybrid analysis to explore the yeast protein interactome.* Proc Natl Acad Sci U S A, 2001. **98**(8): p. 4569-74.
23. Valencia, A. and F. Pazos, *Computational methods for the prediction of protein interactions.* Curr Opin Struct Biol, 2002. **12**(3): p. 368-73.
24. Aloy, P., et al., *Structure-based assembly of protein complexes in yeast.* Science, 2004. **303**(5666): p. 2026-9.
25. Russell, R.B., et al., *A structural perspective on protein-protein interactions.* Curr Opin Struct Biol, 2004. **14**(3): p. 313-24.
26. Gray, J.J., et al., *Protein-protein docking predictions for the {CAPRI} experiment.* Proteins, 2003. **52**(1): p. 118--122.
27. Fokine, A., et al., *Molecular architecture of the prolate head of bacteriophage T4.* Proc Natl Acad Sci U S A, 2004. **101**(16): p. 6003-8.
28. Yonekura, K., S. Maki-Yonekura, and K. Namba, *Complete atomic model of the bacterial flagellar filament by electron cryomicroscopy.* Nature, 2003. **424**(6949): p. 643--650.
29. Holmes, K.C., et al., *Electron cryo-microscopy shows how strong binding of myosin to actin releases nucleotide.* Nature, 2003. **425**(6956): p. 423--427.
30. Gao, H., et al., *Study of the structural dynamics of the {E} coli 70{S} ribosome using real-space refinement.* Cell, 2003. **113**(6): p. 789--801.
31. Chacon, P. and W. Wriggers, *Multi-resolution contour-based fitting of macromolecular structures.* J Mol Biol, 2002. **317**(3): p. 375-84.
32. Volkmann, N. and D. Hanein, *Quantitative fitting of atomic models into observed densities derived by electron microscopy.* J Struct Biol, 1999. **125**(2-3): p. 176-84.
33. Topf, M., et al., *Structural characterization of components of protein assemblies by comparative modeling and electron cry-microscopy.* J.Struc.Biol., 2004.
34. Koch, M.H., P. Vachette, and D.I. Svergun, *Small-angle scattering: a view on the properties, structures and structural changes of biological macromolecules in solution.* Q Rev Biophys, 2003. **36**(2): p. 147-227.
35. Alber, F., et al., *Determining the architectures of macromolecular assemblies.* Nature, 2007. **450**(7170): p. 683-694.

36. Alber, F., M.F. Kim, and A. Sali, *Structural characterization of assemblies from overall shape and subcomplex compositions.* Structure (Camb), 2005. **13**(3): p. 435-45.

37. Reese, M.L., et al., *The guanylate kinase domain of the MAGUK PSD-95 binds dynamically to a conserved motif in MAP1a.* Nat Struct Mol Biol, 2007. **14**(2): p. 155-63.

38. Reese, M.L. and V. Dötsch, *Fast mapping of protein-protein interfaces by NMR spectroscopy.* J Am Chem Soc, 2003. **125**(47): p. 14250-1.

39. Medalia, O., et al., *Macromolecular architecture in eukaryotic cells visualized by cryoelectron tomography.* Science, 2002. **298**(5596): p. 1209-13.

40. Sali, A., et al., *From words to literature in structural proteomics.* Nature, 2003. **422**(6928): p. 216-25.

41. Alber, F., N. Eswar, and A. Sali, *Structure determination of macromolecular complexes by experiment and computation*, in *Nucleic Acids and Molecular Biology, Vol. 15, Practical Bioinformatics*, J.M. Bujnicki, Editor. 2004, Springer-Verlag: Berlin, Heidelberg. p. 73-96.

42. Aebersold, R. and M. Mann, *Mass spectrometry-based proteomics.* Nature, 2003. **422**(6928): p. 198-207.

43. Gavin, A.C., et al., *Functional organization of the yeast proteome by systematic analysis of protein complexes.* Nature, 2002. **415**(6868): p. 141-7.

44. Huh, W.K., et al., *Global analysis of protein localization in budding yeast.* Nature, 2003. **425**(6959): p. 686-91.

45. Corman, T.H., et al., *Introduction to algorithms*. second edition ed. 2001, Cambridge, Massachusettts: MIT press.

46. Sali, A. and T.L. Blundell, *Comparative protein modelling by satisfaction of spatial restraints.* J.Mol.Biol., 1993. **234**(3): p. 779-815.

47. Groll, M., et al., *Structure of 20S proteasome from yeast at 2.4 A resolution.* Nature, 1997. **386**(6624): p. 463-71.

48. Davis, F.P.S., A., *PIBASE: A comprehensive database of structurally defined protein domain interfaces.* Bioinformatics, 2004.

49. Theodoridis, S. and K. Koutroumbas, *Pattern recognition*. 1999: Academic press.

50. de la Torre, J.G. and V.A. Bloomfield, *Hydrodynamic theory of swimming of flagellated microorganisms.* Biophys J, 1977. **20**(1): p. 49-67.

51. Berman, H.M., et al., *The Protein Data Bank.* Nucleic Acids Research, 2000. **28**: p. 235-242.

52. Baker, D. and A. Sali, *Protein structure prediction and structural genomics.* Science, 2001. **294**: p. 93-96.

53. Marti-Renom, M.A., et al., *Comparative protein structure modeling of genes and genomes.* Annu Rev Biophys Biomol Struct, 2000. **29**: p. 291-325.

54. Ginalski, K., *Comparative modeling for protein structure prediction.* Current Opinion in Structural Biology, 2006. **16**(2): p. 172-177.

55. Russell, R.B., et al., *A structural perspective on protein-protein interactions.* Current Opinion in Structural Biology, 2004. **14**(3): p. 313-324.

56. Zhou, Z.H., *Towards atomic resolution structural determination by single-particle cryo-electron microscopy.* Current Opinion in Structural Biology, 2008. **18**(2): p. 218-228.

57. Gray, J.J., *High-resolution protein-protein docking.* Current Opinion in Structural Biology, 2006. **16**(2): p. 183-193.

58. Lensink, M.F., R. Méndez, and S.J. Wodak, *Docking and scoring protein complexes: CAPRI 3rd Edition.* Proteins: Structure, Function, and Bioinformatics, 2007. **69**(4): p. 704-718.

59. Fabiola, F. and M.S. Chapman, *Fitting of high-resolution structures into electron microscopy reconstruction images.* Structure (Camb), 2005. **13**(3): p. 389-400.

60. Topf, M. and A. Sali, *Combining electron microscopy and comparative protein structure modeling.* Current Opinion in Structural Biology, 2005. **15**: p. 1-8.

61. Alber, F., et al., *The molecular architecture of the nuclear pore complex.* Nature, 2007. **450**(7170): p. 695-701.

62. Schwarz, D., et al., *Preparative scale expression of membrane proteins in Escherichia coli-based continuous exchange cell-free systems.* Nat Protoc, 2007. **2**(11): p. 2945-57.

63. Sprangers, R., A. Velyvis, and L.E. Kay, *Solution NMR of supramolecular complexes: providing new insights into function.* Nat Methods, 2007. **4**(9): p. 697-703.

64. Davis, F.P. and A. Sali, *PIBASE: a comprehensive database of structurally defined protein interfaces.* Bioinformatics, 2005. **21**(9): p. 1901-1907.

65. Brunger, A.T., *X-PLOR Version 3.1. A System for X-ray Crystallography and NMR*. 1992, New Haven, CT: Yale University.

66. Richmond, T. and F. Richards, *Packing of a-helices: Geometrical constraints and contact areas.* . J Mol Biol, 1978. **119**: p. 537-555.

67. Sali, A. and J.P. Overington, *Derivation of rules for comparative protein modeling from a database of protein structure alignments.* Protein Sci, 1994. **3**(9): p. 1582-96.

68. Levitt, M., *Growth of novel protein structural data.* Proceedings of the National Academy of Sciences, 2007. **104**(9): p. 3183-3188.

69. Bradley, P., K.M.S. Misura, and D. Baker, *Toward High-Resolution de Novo Structure Prediction for Small Proteins.* Science, 2005. **309**(5742): p. 1868-1871.

70. Dill, K.A., et al., *The Protein Folding Problem.* Annual Review of Biophysics, 2008. **37**(1): p. 289-316.

71. Cavalli, A., et al., *Protein structure determination from NMR chemical shifts.* Proceedings of the National Academy of Sciences, 2007. **104**(23): p. 9615-9620.

72. Moult, J., et al., *Critical assessment of methods of protein structure prediction - Round VII.* Proteins: Structure, Function, and Bioinformatics, 2007. **69**(S8): p. 3-9.

73. Kopp, J., et al., *Assessment of CASP7 predictions for template-based modeling targets.* Proteins: Structure, Function, and Bioinformatics, 2007. **69**(S8): p. 38-56.

74. Cozzetto, D., et al., *Assessment of predictions in the model quality assessment category.* Proteins: Structure, Function, and Bioinformatics, 2007. **69**(S8): p. 175-183.

75. Méndez, R., et al., *Assessment of CAPRI predictions in rounds 3-5 shows progress in docking procedures.* Proteins: Structure, Function, and Bioinformatics, 2005. **60**(2): p. 150-169.

76. Battey, J.N.D., et al., *Automated server predictions in CASP7.* Proteins: Structure, Function, and Bioinformatics, 2007. **69**(S8): p. 68-82.

77. Koh, I.Y., et al., *EVA: Evaluation of protein structure prediction servers.* Nucleic Acids Res, 2003. **31**(13): p. 3311-5.

78. Sanchez, R., et al., *Protein structure modeling for structural genomics.* Nat Struct Biol, 2000. **7 Suppl**: p. 986-90.

79. Peitsch, M.C., *About the use of protein models.* Bioinformatics, 2002. **18**(7): p. 934-938.

80. Inbar, Y., et al., *Protein structure prediction via combinatorial assembly of sub-structural units.* Bioinformatics, 2003. **19**(suppl_1): p. i158-168.

81. Inbar, Y., et al., *Prediction of Multimolecular Assemblies by Multiple Docking.* Journal of Molecular Biology, 2005. **349**(2): p. 435-447.

82. Duhovny, D., R. Nussinov, and H. Wolfson. *Efficient unbound docking of rigid molecules*. in *Proceedings of the 2nd Workshop on Algorithms in Bioinformatics (WABI)*. 2002. Rome, Italy: Springer Verlag.

83. Schneidman-Duhovny, D., et al., *Taking geometry to its edge: Fast unbound rigid (and hinge-bent) docking.* Proteins: Structure, Function, and Genetics, 2003. **52**(1): p. 107-112.

84. Fiser, A. and A. Sali, *Modeller: generation and refinement of homology-based protein structure models.* Methods Enzymol, 2003. **374**: p. 461-91.

85. Havel, T.F., I.D. Kuntz, and G.M. Crippen, *The Theory and Practice of Distance Geometry.* Bulletin of Mathematical Biology, 1983. **45**(5): p. 665-720.

86. Oshiro, C.M., J. Thomason, and I.D. Kuntz, *Effects of Limited Input Distance Constraints Upon the Distance Geometry Algorithm.* Biopolymers, 1991. **31**: p. 1049-1064.

87. Sullivan, D.C., et al., *Information Content of Molecular Structures.* Biophys. J., 2003. **85**(1): p. 174-190.

88. Thorpe, M.F. and P.M. Duxbury, eds. *Rigidity Theory and Applications*. Fundamental Materials Research, ed. M.F. Thorpe. 1999, Springer: New York, NY. 428.

Publishing Agreement

It is the policy of the University to encourage the distribution of all theses and dissertations. Copies of all UCSF theses and dissertations will be routed to the library via the Graduate Division. The library will make all theses and dissertations accessible to the public and will preserve these to the best of their abilities, in perpetuity.

Please sign the following statement:

I hereby grant permission to the Graduate Division of the University of California, San Francisco to release copies of my thesis or dissertation to the Campus Library to provide access and preservation, in whole or in part, in perpetuity.

3 September 2008

Author Signature Date