Computational Protein Interface Design and Prediction with Experimental Constraints

Steven Morgan Lewis

A dissertation submitted to the faculty of the University of North Carolina at Chapel Hill in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Biochemistry and Biophysics (Program in Molecular and Cellular Biophysics).

Chapel Hill
2012

Approved by:

Brian Kuhlman, Ph. D.

Matthew Redinbo, Ph. D.

Nikolay Dokhoylan, Ph. D.

Klaus Hahn, Ph. D.

Jane Richardson

Abstract

STEVEN LEWIS: Computational Protein Interface Design and Prediction with Experimental Constraints
(Under the direction of Brian Kuhlman)

Computational modeling is a powerful companion to direct experimental testing, because it allows researchers to answer questions that are too expensive to test in the lab or insurmountable with existing techniques. The development of algorithms and a computational framework in which to use them form an upfront cost of modeling. The Rosetta software suite is one such framework. Our recent rewrite of Rosetta using modular, reusable, object-oriented code allows Rosetta developers to rapidly and easily create modeling protocols to address new biological questions, reducing the investment needed to use computer modeling and enabling easier biologist-modeler collaborations.

One such Rosetta protocol, AnchoredDesign, performs single-sided flexible-backbone protein-protein interface design. It borrows information from known partners of some protein target to help redesign arbitrary scaffolds into protein affinity reagents against that target. Benchmarking results indicate that the protocol performs well, and we used the protocol in combination with library display selections to generate fibronectin monobody binders to the Keap1 β-propeller domain with dissociation constants in the single nanomolar range. We also modeled the interaction between monobody 1F11 and cSrc-SH3. 1F11 has been functionalized with a solvent-sensitive fluorophore to allow it to report when and where in a cell Src-family kinases are active, and we used AnchoredDesign models to support structural hypotheses explaining this sensing ability.

We developed two other new Rosetta protocols, FloppyTail and UBQ_E2_thioester, to address a biologist's hypotheses about particular protein structures. In the first set of experiments, collaborators suggested that a long, acidic C-terminal tail in a protein Cdc34 bound a basic cleft on its partner Cul1. We created FloppyTail by repurposing code Rosetta normally uses for *ab initio* structure prediction and loop modeling, and were able to use FloppyTail models to predict experimentally-verified protein contacts. In a second set of experiments, mutational data indicated that ubiquitin's interface with Cdc34 is critically important for ubiquitin transfer catalysis, despite the general opinion of the field. UBQ_E2_thioester was written to model the thioester-linked bound complex of ubiquitin and Cdc34, and again its results successfully predicted experimentally-verified protein contacts.

This...is proteins. Proteins compile.

— Kristen S. Lewis

Acknowledgements

This is the scariest part of this thesis to write. Forgetting to cite a paper or mislabeling a graph is fixable, but forgetting to thank someone is irreparable. As this thesis is really a story of collaborations, none of the interesting bits could have happened without the help of others.

I guess my advisor Brian Kuhlman comes first. My main project, AnchoredDesign, is built around the core of a protocol idea he handed to me at the beginning of my work here. All of my side projects, which have led to many publications, have come from him calling me into his office and pitching an idea to me. He quickly learned what sort of protocol development I'm good at and knows how to frame collaborations so that I can make my modeling contributions most efficiently. I'm very grateful!

The AnchoredDesign stories told in this thesis have help from several groups of people. The protocol development itself (this goes for the other protocols as well) received assistance from many members of the Rosetta community. We'll run out of ink if I name names, but generally anyone who's written a useful `Mover` has earned my thanks.

I am grateful to my committee particularly for help with the design of the benchmarking experiments presented in the AnchoredDesign protocol's benchmarking

paper (Chapter 2). I had only the vaguest ideas of how to prove my protocol; the benchmarking we actually performed worked out great.

The SH3 domain story with AnchoredDesign (Chapter 3) is thanks to Akash Gulyani and Klaus Hahn, along with contributions from Brian Kay and his lab. I'm grateful to have my modeling supporting such an interesting biosensor story. Doug Renfrew was instrumental in getting Rosetta ready to handle Mero53, both indirectly with his own thesis work and directly in assisting with generation of parameter files and rotamer libraries.

The Keap1 AnchoredDesign story (Chapter 4) would have gotten nowhere without a lot of help from other members of the Kuhlman lab. Gurkan Guntas performed most of the DNA work and a reasonable fraction of the protein work for that story, and Tom Lane did a big chunk of protein purification and several binding experiments as a rotation project. Pretty much every member of the lab contributed either minor reagents or experimental expertise at some point in the project.

The two ubiquitin stories (Chapters 5 and 6) were mostly due to the hard work of Gary Kleiger and Anjanabha Saha in Ray Deshaies's lab. Their willingness to cleverly design experiments that step around the limitations of Rosetta models really improved those papers. Doug Renfrew and Phil Bradley contributed ideas and code into making the thioester bond in Chapter 6 work.

I want to somewhat pre-emptively thank Jun Zhang and his advisor Drew Lee, along with Rachael Baker and her advisor Sharon Campbell, for two (as yet unpublished, but mentioned in this thesis) collaborations. Their interesting modeling questions have

allowed the originally-single-purpose code written for the ubiquitin chapters to live on in new projects and really proven to me that well written algorithms are eternal.

Many of UNC's administrative crew have been helpful to me over the years. Our department's administrative office is the friendliest and most helpful group of people I've ever met. The Research Computing folks, who administer the supercomputer clusters on which all the modeling results in this thesis were computed, have also been incredibly helpful in debugging cluster issues and letting me override my processor allotment in emergencies.

Last but not least comes my family. My wife Kristen has ensured that I didn't die from hypertension as a complication of eating nothing but salty canned or frozen dinners for the last several years. My family has been very supportive of my entire academic career, and my in-town relatives have been a wonderful resource as well.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Computational modeling of protein structure is a powerful tool for probing the mysteries of life at the smallest scale. Modeling allows us to develop, test, and discard hypotheses much more quickly than bench science. It also enables the testing of hypotheses that are either impossible or extremely challenging to address through conventional means. The ability to test many hypotheses simultaneously allows a researcher to focus expensive bench resources only on the hypotheses best supported by relatively inexpensive modeling data, making the research more efficient and economical. The extra power afforded by computational modeling requires an upfront investment needed to create the modeling tools and verify them against problems with known answers. These costs can be ameliorated by adjustments to the process by which modeling tools are generated. This thesis demonstrates how advances in the Rosetta protein modeling framework created an environment in which new modeling techniques can be rapidly developed, reducing this first investment cost for new techniques in computational modeling.

Rosetta

Rosetta is a large suite of programs supported by dozens of developers at a universities and research institutions in several countries. The suite contains modules for addressing many of the common problems in protein modeling, including *ab initio* structure prediction, protein design, protein-protein and protein-ligand docking, and loop

and homology modeling [1, 2]. Rosetta's successes are numerous, including the first design of a novel protein fold [3] and good showings in protein structure prediction [4] and protein-protein docking prediction [5] competitions.

Early versions of Rosetta were written in FORTRAN77, which was machine-translated into C++ and released as Rosetta++ [6]. The haphazard development of the Rosetta++ codebase greatly complicated generation and dissemination of new protocols written as part of the Rosetta++ suite. In particular, adding new protocols to Rosetta often required adding a few lines of code in the middle of many other existing bits of code, instead of developing new modules as single units. The Rosetta community therefore decided to rewrite the whole body of code, with the major part of the rewrite commencing in 2007. We released this rewritten Rosetta as the Rosetta3 applications suite.

We designed Rosetta3 to remedy the developer-facing flaws of Rosetta++. One of our specific goals was to make it easier to create and test new algorithms for biophysical modeling problems [6]. In other words, Rosetta3 makes it relatively simple to define a search space for a problem of interest, a search function with which to sample that space, and a score function to rate the structures sampled. This is achieved through the use of object-oriented, reusable code, which allows experimenters interested in a particular biophysical problem to develop only the code most pertinent to that problem. Common shared problems like rotamer packing and minimization are available through simple interfaces, and the internal logic of those protocols does not need to be modified to accommodate the biophysical problem in question. This lets a computational modeler

focus on optimizing search and scoring functions at a high level without having to re-implement ideas at every level of minutiae in the whole program.

This thesis will describe the development, purpose, and results of three different executables that have been released as part of the Rosetta3 suite. In each case, the programming flexibility granted by Rosetta3's refactoring was essential to their success. For the AnchoredDesign suite, Rosetta3's modularity allowed the development of a very complicated protocol for the complex problem of flexible protein-protein interface design, which was built with simple calls to Rosetta's packing, minimization, and loop modeling routines [7]. For the FloppyTail and UBQ_E2_thioester protocols, Rosetta's flexibility allowed rapid development of modeling protocols designed to answer a specific biologic question in collaboration with biologists [8, 9].

Rosetta's tools

Protein modeling algorithms are built on two underlying techniques: a scoring function which determines the thermodynamic plausibility of a structure, and a search function which proposes structures for scoring. This thesis is focused on the development of algorithms with novel search techniques, so it relies on Rosetta's previously published and proven scoring functions [3, 10].

Search functions are meant to find the best possible structures in the least amount of time. The set of all possible states they could sample is called their search space. Most Rosetta search functions are built on the Metropolis-Monte Carlo algorithm [10]. Monte Carlo-based algorithms run as many iterations of a common cycle. In each cycle, the algorithm takes the result structure of the previous cycle, makes some random modification(s) to it, and re-scores the structure. If the score improves, the changes are

accepted, which drives the trajectory to ever-improving scores and structures. If the score gets worse, the changes are accepted with a probability inversely related to the size of the score change (the Metropolis criterion) [11]. This occasional increase in score allows escape from local energy minima during the search for the global energy minimum. Metropolis-Monte Carlo does not guarantee finding global energy minima, but has proven to be a fast method for finding plausible protein structures [3].

The new protocols developed in this thesis use Rosetta's score function and use Monte Carlo as a base for their search functions. Monte Carlo alone does not specify how random modifications are selected, just how the modifications accumulate. The novelty of the protein modeling algorithms in this thesis lies in what random modifications are proposed during Monte Carlo modeling, and how those modifications are implemented.

Each individual modification to a protein structure samples some degree of freedom in the model. Rosetta uses internal coordinates to place atoms in an `AtomTree`, also known as an Internal Coordinates Model (ICM) [6, 12]. This iteratively calculates 3-D coordinates for each atom in the model based on a distance (bond length) from the previous atom, a three-body angle calculated with two previous atoms, and a four-body torsion (dihedral) angle calculated with three previous atoms. Rosetta traditionally focuses on modeling the more variable torsions and minimally models the less variable angles and lengths [6]. Nonbonded degrees of freedom—for example, the rigid-body orientation between docking partners—are modeled with a `Jump` degree of freedom which behaves as a pseudobond.

The complexity of an atomic level `AtomTree` is often softened by use of a residue-level `FoldTree` [13]. The `FoldTree` operates on the same principles by organizing an underlying `AtomTree`, but allows the programmer to consider whole residue units instead of individual atoms. This `FoldTree` is very flexible in that it allows the development of many sorts of modeling approaches in Rosetta [13].

A key effect of the internal coordinate representation of proteins is the cascading effect of degree of freedom movement caused by the dependence of atoms late in the chain on those early in the chain. The three-dimensional spatial coordinates for any atom are recalculated from the positions of the atoms it depends on, and the internal coordinate degrees of freedom connecting the atoms. This means that a rotation of one backbone torsion will propagate from parent to child atoms, ultimately causing all downstream dependent atoms to move as well. For some modeling problems, this is a downside of the ICM approach, but for the approaches detailed in this thesis it is critically useful.

Another part of the Rosetta3 rewrite that is important for understanding this thesis is the centrality of the `Mover` idea. In Rosetta, a `Mover` is a C++ class which takes a protein conformation, alters it in some fashion, and returns a new protein conformation [6]. `Mover`s therefore plug into a Monte Carlo search protocol by proposing conformational changes for Monte Carlo's consideration. An algorithm can be built primarily by writing `Mover`s to propose changes, and layering these `Mover`s together in increasingly complex assemblies to produce whole protocols. The modularity of `Mover`s mean that different programmers can readily share `Mover`s of any granularity to re-use already-working code and focus their efforts on truly novel development. Each protocol detailed in this thesis is built of `Mover`s. The outermost `Mover`s represent the new

protocols, and the innermost represent Rosetta's published algorithms for packing, minimization, and structure manipulation.

Tethered docking

Each of the new Rosetta protocols presented here treats a problem in protein-protein interface prediction and docking, with the twist that in each case some wrinkle in the modeling precludes the use of standard rigid-body docking protocols. For every protocol, there is some rigid constraint tethering together the proteins being modeled—either a covalent connection between proteins (UBQ_E2_thioester) or a partially known interface to be maintained (AnchoredDesign and FloppyTail). It should be noted that the applications were not developed with the intention of treating similar problems; rather, each was developed in response to a particular modeling or biological problem of interest. The Rosetta application developed to address each of these problems is unique in that it only searches through solutions that meet the tethering constraint—unproductive conformations that would be rejected for failing the constraint are never searched.

Well-established protein docking protocols have met with good success [5], but all are very general methods and thus can only incorporate system-specific constraints in a limited fashion. For example, Monte Carlo based docking search protocols, such as RosettaDock [14, 15] and HADDOCK [16], can incorporate experimentally derived constraints to bias their score functions, but still propose solutions randomly via Monte Carlo. In other words, the proposal step of Monte Carlo is not aware of the constraint, only the selection step. Other well-supported methods like ZDOCK [17, 18] and ClusPro [19] use an exhaustive grid search at the core of their docking search. Again, structure proposal is not linked to the experimental constraints, and time is wasted on scoring

models that never satisfied the preexisting constraints. These sorts of search functions deal with constraints by filtering or ranking on the constraint post hoc. Figure 1.1 demonstrates how these sorts of docking search protocols propose models for scoring.

For the tethered docking problems addressed in this thesis, the tethering constraint demands that some portion of the interface remain intact. Instead of freely varying the rigid-body degree of freedom between proteins, this tether connection can be remodeled to produce candidate docked structures. Flexing the point of contact between the two rigid bodies produces motion that mimics rigid-body docking. To produce this effect, the folding pathway (`AtomTree`) through which Rosetta converts internal coordinates to three-dimensional coordinates passes through this tether region between the upstream and downstream proteins. This means that modifications to the tether affect all atoms downstream in the folding pathway, allowing the moving side of the interface to swing through space as the tether is sampled. All of the Rosetta modules presented in this thesis take advantage of this tethered atom tree structure to substitute tether remodeling for rigid-body docking, allowing sampling to focus tightly on conformations that respect the tether constraint. Figure 1.1 demonstrates how the Rosetta tethered docking applications in this thesis propose models for scoring in a fashion consistent with the tethering constraint.

AnchoredDesign

The first Rosetta3 application developed during the course of this thesis, AnchoredDesign [7], is a fresh attack on the unsolved problem of *de novo* protein-protein interface design. Chapter 2 introduces the AnchoredDesign algorithm, which is designed to use grafted portions of a known interface between a target and some partner to seed a

new interface between the target and a desired scaffold. This graft, called the anchor, is inserted into a scaffold loop and held rigidly affixed to the target while the scaffold is sampled around it, as demonstrated by Figure 2.1. The purpose of this anchoring is twofold: first, it may allow for some small baseline affinity between the design scaffold and the target, and second, it ensures that the scaffold redesign is aimed at an interface-compatible region of the target surface. Protein surfaces that form transient protein-protein interfaces tend to have more aromatic and fewer charged residues [20], suggesting that targeting preexisting binding interfaces will make it easier to design binding partners.

The algorithm uses both loop modeling and design iteratively to produce flexible backbone interface design. The grafted anchor, in a loop of the design scaffold contacting the target, serves as the docking tether. Docking degrees of freedom are sampled by modifying the loop containing the anchor, which results in a swinging motion about the interface and ultimately altered rigid-body orientation, as demonstrated by Figure 2.2. Other surface loops of the scaffold can also be sampled using Rosetta's implementation of the Cyclic Coordinate Descent [21] or kinematic loop closure [22] algorithms, ensuring good loop-mediated shape complementarity between the target and design scaffold. Rosetta's proven design techniques [3] are used to sample sequence space of the moving loops to select sequence that will best fold into the desired structure. Computational benchmarking of this algorithm demonstrated its utility [7].

New interfaces created with AnchoredDesign

AnchoredDesign is a design technique at heart, written to leverage tethered docking in an interface design context. To develop and validate the algorithm, we have used it to generate fibronectin-based monobodies [23] binding to the Keap1 Kelch-like β-

propeller domain [24]. As reported in Chapter 4, these designed monobodies bind with affinities resembling the wild-type partners of Keap1, and partial designs enhanced with phage display techniques bind much tighter than known partners for Keap1.

Our design scaffold, human fibronectin, domain type 3, repeat 10 (10FNIII), is well known as an adaptable scaffold for generating new protein-protein interfaces [23, 25-27]. Figure 3.2 demonstrates such an interface. The AnchoredDesign protocol was developed with this scaffold in mind: the algorithm designs flexible loops at the surface of the scaffold, and 10FNIII has two surface loops (labeled FG and BC, after the beta strands they occur between) that are very amenable to mutation [23]. Insertion of the anchor in either loop allows AnchoredDesign to use that loop as a flexing point for tethered docking, while remodeling and redesigning both loops to generate a new protein-protein interface. Several crystal structures of 10FNIII are available, including both wild type fibronectin [28, 29] and derived monobodies in complex with partners [27].

Keap1 is an interesting target due to the structure of its naturally-occurring protein-protein interface. It natively binds a short hairpin structure from its partner Nrf2, as revealed in a 1.5 Å resolution crystal structure [24] seen in Figure 4.1. This hairpin has geometrically close end points, which makes it ideal for insertion into a fibronectin loop as an anchor.

AnchoredDesign was used to model structures of 10FNIII with this hairpin anchor inserted into a loop. These models were used to inform directed and random libraries for phage display selection of monobody binders to Keap1. Several resulting sequences were tested for binding to Keap1. All sequences tested had some affinity for the Keap1

target, but experiments designed to elucidate the structure of the monobody/Keap1 complexes have produced either no or ambiguous results.

The success of AnchoredDesign at generating sequences which bind their intended targets showcases how Rosetta3's restructuring makes modeling simpler. The protocol addresses one of the most complex current topics in protein modeling: flexible, one-sided protein-protein interface design. An algorithm for the problem must handle searching through rigid-body, backbone, and sequence space. All of this complexity is manageable due to the ease of use of Rosetta's many modular tools for protein modeling.

AnchoredDesign predicts qualities of monobody-SH3 interfaces

AnchoredDesign has also been proven capable of predicting some qualities of an engineered fluorophore-labeled biosensor [30], reported in Chapter 3. We were interested in generating reagents that used fluorescence to signal the location and activity of SH3 (Src Homology 3) domains. We used an existing fibronectin-based monobody affinity reagent, 1F11, which had been evolved to bind the cSrc SH3 domain [26]. Fluorophores chemically conjugated to different positions on the monobody were found to vary in their response to monobody-cSrc binding. Specifically, at some scaffold positions, the fluorphore's intensity changed on cSrc binding, but at other positions, it did not.

We used AnchoredDesign to generate models of the cSrc SH3/1F11 complex and probe hypotheses about why some fluorophore positions produced an intensity change upon binding and some did not. This biosensor modeling showcases the utility of Rosetta3's flexibility, in that the AnchoredDesign algorithm was rapidly adapted to incorporate the fluorophore as well. Rosetta's non-canonical amino acid capabilities [31] were used to generate a parameter set and rotamer library for the chemically conjugated

fluorophore. This fluorophore could then be dropped into Rosetta modeling with no modifications to the central packing, minimization, or loop closure algorithms, and only minor modifications to AnchoredDesign itself. Ultimately, AnchoredDesign was able to discriminate positions that showed a fluorescence intensity change on binding from those that did not by showing a change in solvent-accessible surface area of the polar groups of the fluorophore between the bound and unbound states [30].

This biosensor modeling project exemplifies the power of computational modeling as a companion to biology, and demonstrates how Rosetta3's structure reduces the upfront investment needed to get modeling data. Here, an application written to perform protein-protein interface design was rapidly adapted for fixed-sequence modeling of a known interface, including the chemically conjugated fluorophore. The modularity of the non-canonical amino acid protocols in Rosetta allowed easy drop-in of the novel fluorophore into an existing protocol. This sort of biology-modeling feedback is a major benefit of the structural improvements in Rosetta 3. Figure 1.2 presents a flowchart model of how these sorts of biology-modeling collaborations function.

FloppyTail modeling of Cdc34-Cul1-Rbx1

A major benefit of Rosetta's new architecture is the ease and rapidity with which whole new applications can be developed to address biological questions. Biologists regularly pose questions about protein structure that can be answered by models, but unless an appropriate modeling tool already exists, getting the question answered may be hard. Chapter 5 follows development of the FloppyTail module of Rosetta3 shows how Rosetta's new flexibility makes it easy to address novel questions with novel modeling techniques designed specifically for the question.

FloppyTail [8] was originally developed in collaboration with a biology-focused lab led by Raymond J. Deshaies in order to answer a specific question about a protein complex of interest. The binding of two proteins, ubiquitin E3 ligase complex SCF (Skp, Cullin, F-box) and Cdc34, was known to be affected by the presence of a long, highly negatively charged C-terminal tail on Cdc34. Sequence homology to solved structures implied that Cdc34 bound SCF via the RING domain in the Rbx1 subunit of SCF, but there were no structures of the complex, nor any Cdc34 structures containing the tail. Experiments demonstrated that the charged tail was necessary for the normal biologic function of these proteins [32]. The SCF complex contains a highly positively charged cleft or patch in its Cul1 subunit (Figure 5.1), and mutations in this region cause defects in ubiquitin transfer [8]. These data lead to the hypothesis that the acidic tail and basic cleft may interacting as a second binding site between SCF and Cdc34.

The available structures for addressing this question were a partial structure of Cdc34 (PDB 2OB4) [33] missing the C-terminal tail, and a partial structure of SCF (PDB 1LDJ) [34], containing the basic cleft and presumed RING domain primary binding site. A homologous bound E2/RING domain complex structure (CBL-UBCH7, PDB 1FBV) [35] existed to guide creation of a docked model for the primary binding site with a straightforward application of Rosetta's docking module. However, this left the tail-cleft binding question open, with no structure for the 57-residue tail (which can stretch to 200 Å) and its hypothesized binding cleft approximately 80 Å from the base of the tail, as detailed in Figure 5.2. Circular dichroism did not suggest strong secondary structure in the tail [36], nor did secondary structure prediction algorithms [8, 37].

No Rosetta application that existed at the time was capable of handling this sort of question. However, Rosetta3's modular structure allowed the rapid development of the FloppyTail application to fit the purpose, detailed in Figure 5.3. Search algorithm tools originally written for *ab initio* structure prediction and loop modeling were combined to allow sampling of the volume available to the flexible tail, even though it does not have significant secondary structure as *ab initio* requires, nor is it a closed loop as loop modeling assumes. FloppyTail keeps the flexible region being modeled tethered to the appropriate attachment points on preexisting structures of the other domains in the system, and modifies backbone and sidechain torsions in the flexible region to determine where and how they might associate. This tethering limits the search space for the interface between the Cdc34 tail and SCF to regions within geometric reach of the tail without breaking its tether connection to Cdc34.

This modeling protocol allowed prediction of certain features of the putative tail-cleft interface which were confirmed by experiment [8]. Specifically, the models were used to predict that Cul1 residue K679 and Cdc34 residue C227 were nearby in the bound state, which was demonstrated by chemical crosslinking. This completes the biology-modeling collaboration in Figure 1.2: the Deshaies lab posed a structural question, a Rosetta protocol was developed to address the question, and the resulting models were used to make verifiable predictions about the interface which deepened understanding of the biology of the system.

UBQ_E2_thioester modeling of the ubiquitin-Cdc34 interface

FloppyTail's success as a biology-modeling collaboration is not singular. Because of the success of that collaboration, the Deshaies lab posed another modeling question.

This time, the nature of the interface between the E2 Cdc34 and ubiquitin was of interest. As with the SCF/E2 tail interface in the FloppyTail project, no structure of the interface existed, but experiments generated data constraining what the interface might look like. Modeling of this interface is discussed in Chapter 6.

The Cdc34-ubiquitin interface is not a normal, labile protein-protein interface. It is instead dominated by the thioester bond that forms between ubiquitin's C-terminus and the catalytic cysteine residue of Cdc34 [9], shown in Figure 6.2. This thioester bond is normal for a ubiquitin-charged E2 that is ferrying activated ubiquitin to an E3 [38], but thioester bonds are rare in proteins in general. While this bond is sufficiently thermodynamically unstable to make structural studies challenging [9, 39], it is a covalent bond, so candidate interface structures must take it into account.

The existence of the thioester bond makes this modeling problem another example of a tethered docking experiment. The most interesting degree of freedom in the system is the rigid-body orientation between ubiquitin and Cdc34. While the possible solution space is enormously limited by the existence of this covalent bond, it is still essentially a docking problem regarding a non-obligate interface.

Besides the thioester bond, mutational data at the interface [9], along with a model structure of a different E2-ubiquitin interface [39], were available for guiding the modeling of the Cdc34-ubiquitin interface. To model the interface between ubiquitin and Cdc34, including specific treatment of the embedded thioester bond, the UBQ_E2_thioester application was developed, as diagrammed in Figure 6.3. Novel code to handle the thioester bond's existence and malleability was combined with the flexible backbone modeling tools used for *ab initio* folding and loop modeling. This was used to

sample the backbone for the ubiquitin C-terminal tail conjugated to Cdc34 and determine what envelope on the surface of Cdc34 it could occupy, as well as identify possible binding modes. This treatment of the tethered docking problem satisfies the covalent thioester constraint by including it in the modeling from the beginning and searching for solutions pivoting around the thioester, instead of performing general docking and filtering for thioester-compatible solutions.

These models were used to predict a rescue mutation, Cdc34 S129L, for a known mutation that disrupted the Cdc34-ubiquitin interface, ubiquitin I44A. The success of this rescue mutation reiterates the power of rapid, collaborative program development in Rosetta3. As with FloppyTail, a biological question was addressed by the development of a specific Rosetta application. Again, the modeling results were verified by experiment in the biological system of interest.

Summary

This thesis aims to demonstrate how the reorganization of the Rosetta codebase for Rosetta3 has enabled rapid development of new protocols to address highly complex modeling problems and address biological questions in collaboration with non-modeling collaborators. In each case study, a new protocol was developed which treats a docking problem with some sort of tethering constraint restricting part of the protein-protein interface. The new protocols use this constraint to focus their search space and propose only constraint-compatible conformations as solutions to the modeling question.

Figure 1.1: Search spaces for different docking methods
This figure demonstrates how different docking search methods propose models for scoring. The correct model for the A and B complex is shown at the top. Note the shape complementarity and the coalignment of the green dot, which represents an experimentally-validated interaction known as a modeling input. For the random-proposal methods like RosettaDock [14, 15] and HADDOCK [16], random orientations and translations of one docking partner against the other are tested. For grid-search methods, such as ZDOCK [17, 18] and ClusPro [19], all orientations and rotations are attempted at some grid granularity; for example 3 Å translations or 5° rotations. The methods presented in this thesis rely on tethered docking, as shown in the bottom panel. Here, the experimental knowledge (represented as the green dot in the interface) is leveraged to propose only solutions compatible with that preexisting knowledge, resulting in richer sampling near the correct answer.

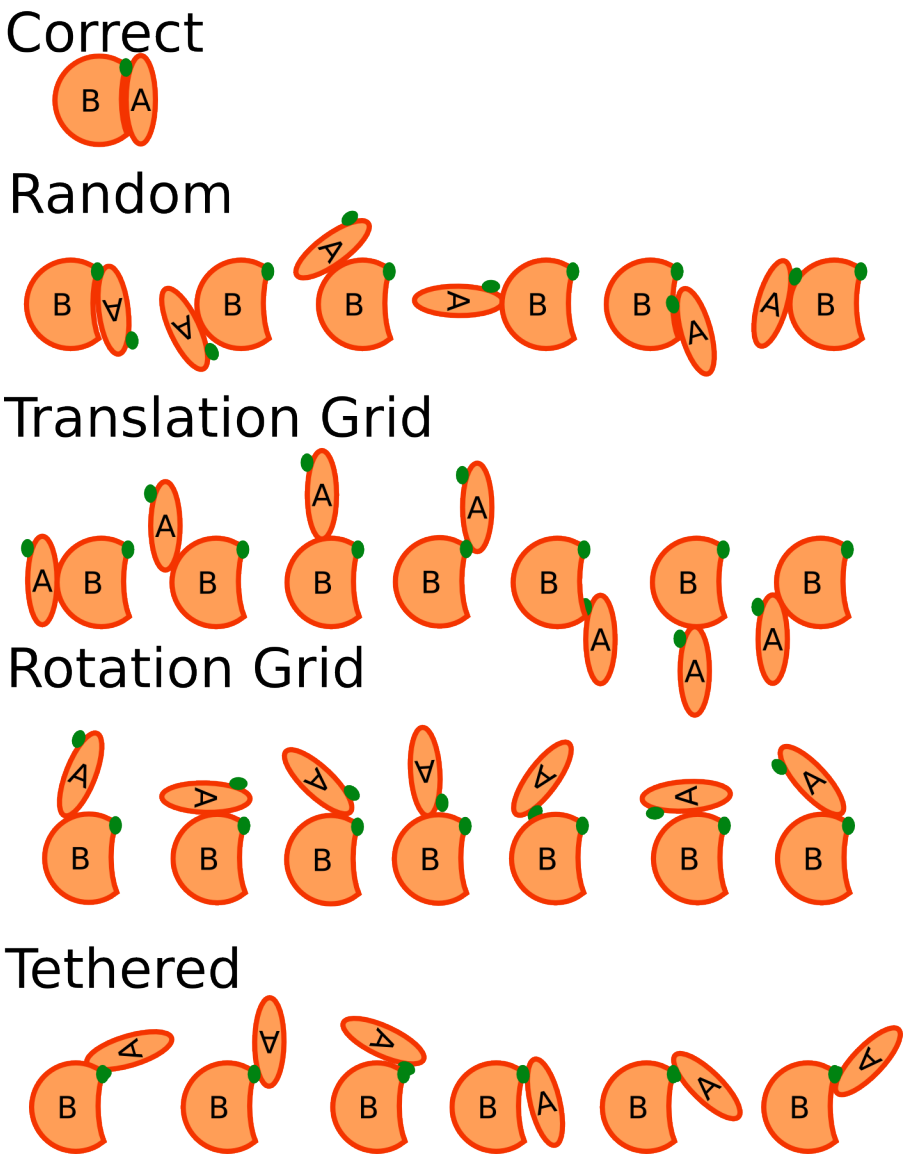
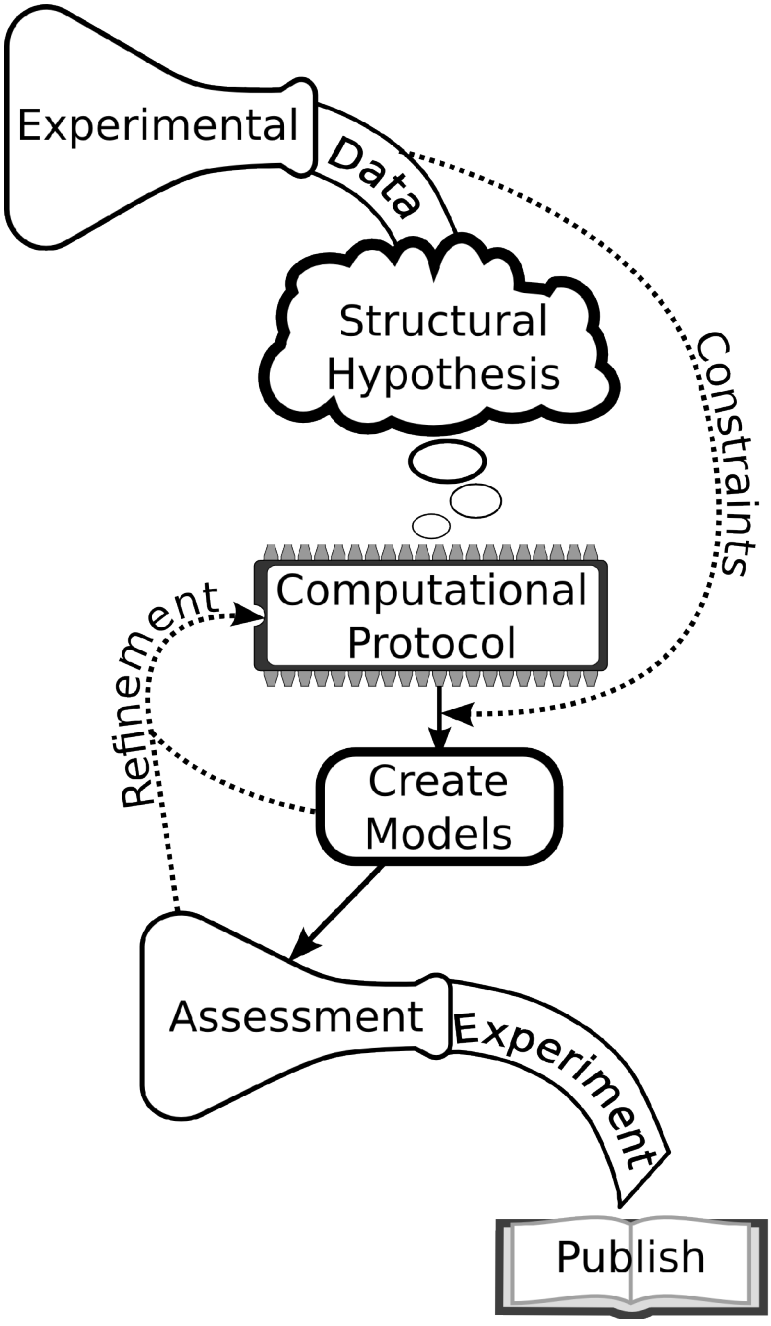


Figure 1.2: A flow diagram representing biology-modeling collaboration
This flowchart summarizes the biology-modeling collaboration seen repeatedly in this thesis. Initial experimental data leads to a structural hypothesis unaddressable by further experiment. This hypothesis is used to generate a computational protocol in Rosetta3, a step which is greatly accelerated by Rosetta3's modularity. Models are created using this protocol, possibly with the help of constraints derived from the original experimental data. These models are assessed, and then new experiments are designed to verify the predictions of the model. Successful models can be published, and failures can be used to refine the modeling protocol.



References

1. Kaufmann KW, Lemmon GH, Deluca SL, Sheehan JH, Meiler J. (2010) Practically useful: What the rosetta protein modeling suite can do for you. Biochemistry 49(14): 2987-2998. 10.1021/bi902153g.

2. Das R, Baker D. (2008) Macromolecular modeling with rosetta. Annu Rev Biochem 77: 363-382. 10.1146/annurev.biochem.77.062906.171838.

3. Kuhlman B, Dantas G, Ireton GC, Varani G, Stoddard BL, et al. (2003) Design of a novel globular protein fold with atomic-level accuracy. Science 302(5649): 1364-1368.

4. Raman S, Vernon R, Thompson J, Tyka M, Sadreyev R, et al. (2009) Structure prediction for CASP8 with all-atom refinement using rosetta. Proteins 77 Suppl 9: 89-99. 10.1002/prot.22540.

5. Fleishman SJ, Corn JE, Strauch EM, Whitehead TA, Andre I, et al. (2010) Rosetta in CAPRI rounds 13-19. Proteins 78(15): 3212-3218. 10.1002/prot.22784.

6. Leaver-Fay A, Tyka M, Lewis SM, Lange OF, Thompson J, et al. (2011) ROSETTA3: An object-oriented software suite for the simulation and design of macromolecules. Methods Enzymol 487: 545-574. 10.1016/B978-0-12-381270-4.00019-6.

7. Lewis SM, Kuhlman BA. (2011) Anchored design of protein-protein interfaces. PLoS One 6(6): e20872. 10.1371/journal.pone.0020872.

8. Kleiger G, Saha A, Lewis S, Kuhlman B, Deshaies RJ. (2009) Rapid E2-E3 assembly and disassembly enable processive ubiquitylation of cullin-RING ubiquitin ligase substrates. Cell 139(5): 957-968. 10.1016/j.cell.2009.10.030.

9. Saha A, Lewis S, Kleiger G, Kuhlman B, Deshaies RJ. (2011) Essential role for ubiquitin-ubiquitin-conjugating enzyme interaction in ubiquitin discharge from Cdc34 to substrate. Mol Cell 42(1): 75-83. 10.1016/j.molcel.2011.03.016.

10. Rohl CA, Strauss CEM, Misura KMS, Baker D. (2004) Protein structure prediction using rosetta. Methods Enzymol 383: 66-+.

11. Metropolis N, Rosenbluth AW, Rosenbluth MN, Teller AH, Teller E. (1953) Equation of state calculations by fast computing machines. J Chem Phys 21(6): 1087-1092.

12. Abagyan R, Totrov M, Kuznetsov D. (1994) ICM-A new method for protein modeling and design: Applications to docking and structure prediction from the distorted native conformation. Journal of Computational Chemistry 15(5): 488-506.

13. Wang C, Bradley P, Baker D. (2007) Protein-protein docking with backbone flexibility. J Mol Biol 373(2): 503-519.

14. Gray JJ, Moughon S, Wang C, Schueler-Furman O, Kuhlman B, et al. (2003) Protein-protein docking with simultaneous optimization of rigid-body displacement and side-chain conformations. J Mol Biol 331(1): 281-299.

15. Chaudhury S, Berrondo M, Weitzner BD, Muthu P, Bergman H, et al. (2011) Benchmarking and analysis of protein docking performance in rosetta v3.2. PLoS One 6(8): e22477. 10.1371/journal.pone.0022477.

16. Dominguez C, Boelens R, Bonvin AM. (2003) HADDOCK: A protein-protein docking approach based on biochemical or biophysical information. J Am Chem Soc 125(7): 1731-1737. 10.1021/ja026939x.

17. Chen R, Weng Z. (2002) Docking unbound proteins using shape complementarity, desolvation, and electrostatics. Proteins: Structure, Function, and Bioinformatics 47(3): 281-294. 10.1002/prot.10092.

18. Chen R, Li L, Weng Z. (2003) ZDOCK: An initial-stage protein-docking algorithm. Proteins 52(1): 80-87. 10.1002/prot.10389.

19. Comeau SR, Gatchell DW, Vajda S, Camacho CJ. (2004) ClusPro: An automated docking and discrimination method for the prediction of protein complexes. Bioinformatics 20(1): 45-50.

20. Lo Conte L, Chothia C, Janin J. (1999) The atomic structure of protein-protein recognition sites. J Mol Biol 285(5): 2177-2198.

21. Canutescu AA, Dunbrack RL. (2003) Cyclic coordinate descent: A robotics algorithm for protein loop closure. Protein Sci 12(5): 963-972.

22. Mandell DJ, Coutsias EA, Kortemme T. (2009) Sub-angstrom accuracy in protein loop reconstruction by robotics-inspired conformational sampling. Nature Methods 6(8): 551-552. 10.1038/nmeth0809-551.

23. Koide A, Bailey CW, Huang XL, Koide S. (1998) The fibronectin type III domain as a scaffold for novel binding proteins. J Mol Biol 284(4): 1141-1151.

24. Lo SC, Li X, Henzl MT, Beamer LJ, Hannink M. (2006) Structure of the Keap1:Nrf2 interface provides mechanistic insight into Nrf2 signaling. EMBO J 25(15): 3605-3617. 10.1038/sj.emboj.7601243.

25. Koide A, Abbatiello S, Rothgery L, Koide S. (2002) Probing protein conformational changes in living cells by using designer binding proteins: Application to the estrogen receptor. Proc Natl Acad Sci U S A 99(3): 1253-1258.

26. Karatan E, Merguerian M, Han Z, Scholle MD, Koide S, et al. (2004) Molecular recognition properties of FN3 monobodies that bind the src SH3 domain. Chem Biol 11(6): 835-844. 10.1016/j.chembiol.2004.04.009.

27. Koide A, Gilbreth RN, Esaki K, Tereshko V, Koide S. (2007) High-affinity single-domain binding proteins with a binary-code interface. Proc Natl Acad Sci U S A 104(16): 6632-6637.

28. Dickinson CD, Veerapandian B, Dai XP, Hamlin RC, Xuong NH, et al. (1994) Crystal-structure of the 10th type-iii cell-adhesion module of human fibronectin. J Mol Biol 236(4): 1079-1092.

29. Leahy DJ, Aukhil I, Erickson HP. (1996) 2.0 angstrom crystal structure of a four-domain segment of human fibronectin encompassing the RGD loop and synergy region. Cell 84(1): 155-164.

30. Gulyani A, Vitriol E, Allen R, Wu J, Gremyachinskiy D, et al. (2011) A biosensor generated via high-throughput screening quantifies cell edge src dynamics. Nat Chem Biol 7(7): 437-444. 10.1038/nchembio.585; 10.1038/nchembio.585.

31. Renfrew PD, Choi EJ, Bonneau R, Kuhlman B,. (2012) Incorporation of noncanonical amino acids into rosetta and use in computational protein-peptide interface design. PLoS ONE 7(3): e32637. Available: http://dx.doi.org/10.1371%2Fjournal.pone.0032637 via the Internet.

32. Mathias N, Steussy CN, Goebl MG. (1998) An essential domain within Cdc34p is required for binding to a complex containing Cdc4p and Cdc53p in saccharomyces cerevisiae. J Biol Chem 273(7): 4040-4045.

33. Ceccarelli DF, Tang X, Pelletier B, Orlicky S, Xie W, et al. (2011) An allosteric inhibitor of the human Cdc34 ubiquitin-conjugating enzyme. Cell 145(7): 1075-1087. 10.1016/j.cell.2011.05.039.

34. Zheng N, Schulman BA, Song L, Miller JJ, Jeffrey PD, et al. (2002) Structure of the Cul1-Rbx1-Skp1-F boxSkp2 SCF ubiquitin ligase complex. Nature 416(6882): 703-709. 10.1038/416703a.

35. Zheng N, Wang P, Jeffrey PD, Pavletich NP. (2000) Structure of a c-cbl-UbcH7 complex: RING domain function in ubiquitin-protein ligases. Cell 102(4): 533-539.

36. Ptak C, Prendergast JA, Hodgins R, Kay CM, Chau V, et al. (1994) Functional and physical characterization of the cell cycle ubiquitin-conjugating enzyme CDC34 (UBC3). identification of a functional determinant within the tail that facilitates CDC34 self-association. J Biol Chem 269(42): 26539-26545.

37. Cole C, Barber JD, Barton GJ. (2008) The jpred 3 secondary structure prediction server. Nucleic Acids Research 36(suppl 2): W197-W201. 10.1093/nar/gkn238.

38. Nandi D, Tahiliani P, Kumar A, Chandu D. (2006) The ubiquitin-proteasome system. J Biosci 31(1): 137-155.

39. Hamilton KS, Ellison MJ, Barber KR, Williams RS, Huzil JT, et al. (2001) Structure of a conjugating enzyme-ubiquitin thiolester intermediate reveals a novel role for the ubiquitin tail. Structure 9(10): 897-904.

Chapter 2

Anchored Design of Protein-Protein Interfaces

Introduction

Because so many human diseases are caused by dysregulation of proteins or protein-protein interactions, the need to experimentally or therapeutically adjust these systems is great. A powerful tool for probing protein networks is other proteins engineered to bind particular naturally-occurring target proteins and modify or illuminate their behavior. To that end, many authors have introduced computational methods for creating these tool proteins, including both de novo binding partners and redesigns of existing interfaces.[1] One modeling suite used for this purpose, and many others, is Rosetta.[2]

Placing past successes in context, it remains quite challenging to create binding partners with desired functionality, and even minor successes are not routine.[3, 4] This is because interface design combines all the challenges of protein design, itself an incompletely solved problem, with the additional complication of docking orientation between the two proteins.

The protein design problem requires decisions on how to best search the sequence space, how (or even if) to best search the backbone conformational space, and how to combine the two searches efficiently.[5] In Rosetta, flexible design methods are often only iteratively flexible: their design protocol runs on a fixed backbone, and then backbone flexibility is modeled using a fixed sequence. This is because the algorithmic

optimizations necessary to efficiently sample conformations and sequences preclude sampling both simultaneously. Recent flexible-backbone design methods modifying protein interfaces or loops include methods using local backbone minimization [6] and fragment insertion plus loop closure and design.[7, 8]

A second decision must be made when designing new interfaces: which proteins should interact? The two major methods for protein interface design include *de novo* design, which creates an interface between previously non-interacting proteins, and interface redesign, which modifies the properties of existing interactors or homologs thereof. *De novo* designs offer the opportunity to engineer new functions into the interaction, at the great cost of having to create the interface from scratch.[6, 9, 10] Redesigns offer the opposite tradeoff: there is an interaction in place to start from, but the designs are restricted to modifying existing functions by increasing affinity [11-15] or altering specificities [16-18].

Here we propose a new method offering a blend of these strengths which we call AnchoredDesign. The method has been implemented as a protocol in the Rosetta3 software suite.[19] The method creates an interface between an arbitrary (and arbitrarily functional) scaffold and a target, but it also creates the interface along a known interacting surface of the target, using information from a preexisting binding partner. The method accounts for backbone flexibility at the interface by iterating between loop remodeling and design.

This method requires a known structure of the target complexed to some binding partner, as well as a structure of the desired scaffold. The scaffold should have flexible surface loops amenable to design, and otherwise be chosen for desirable experimental

characteristics. For example, the fibronectin domain type 3 repeat 10 (10FNIII or FN3) scaffold used by many researchers is an appropriate design scaffold.[20-25]

The first step of this new method is to create a nascent interface between the target and the scaffold, as described in Figure 2.1. A small sequence-contiguous portion of the target's known partner is extracted, and its sequence identity and coordinates are inserted into a surface loop on the scaffold. This becomes the anchor. Standard Rosetta loop closure techniques can be used to close the scaffold's modified loop.[26] This results in an intermediate structure containing the target, the scaffold, and a small interface between them where the scaffold mimics the original binding partner. It is ripe for flexible redesign to create a real interface between the partners. Note that the use of this anchor guarantees that the new designed binder will bind to a surface area of the target overlapping the original partner's area. This helps control the activity of the new binder by ensuring that experimenters know where it is binding. It also controls for the residue composition of the target surface as suggested by Lo Conte et al.[27]

Grafting interactions into interfaces is not unknown in the literature, suggesting that the grafted anchor is likely to function as hypothesized. Potapov et al. searched for noncontiguous protein fragments (clusters of residues) from the Protein Data Bank (PDB) [28] matching the known backbone structure of an interface, and showed that mutating a new cluster into a pre-existing interface resulted in the maintenance of stability and specificity.[29] Liu et al. performed the opposite experiment: they created a new interface between shape-compatible but nonbinding proteins by grafting three residues from one interactor's normal partner onto the new binding partner.[30]

After creating a nascent interface via grafting, the next step is flexible redesign. Normally one considers the docking problem when thinking about designing protein-protein interfaces. Here, the anchor precludes the use of whole-protein rigid-body motion as in docking, because this would cause the loss of the anchor. Instead, loop remodeling of the loop containing the anchor is used to sample the rigid body space between the proteins, as in Figure 2.2. Holding the anchor in its original binding conformation, rigidly affixed to the target, and remodeling its loop will result in rigid-body transformations between the target and scaffold. This allows us to use loop modeling to generate backbone flexibility at the interface and simultaneously sample possible binding modes of the scaffold. Other surface loops on the scaffold can be concurrently sampled to produce further surface complementarity.

For computational methods like these, benchmarking tests both help develop the protocol and demonstrate its utility. For the AnchoredDesign protocol, we have assembled a set of 16 protein structures from the PDB. These structures were chosen on the basis of having an interfacial loop with an appropriate residue to serve as an anchor. The protocol can then be tested against these structures by deleting the conformation of the anchor-containing loop and using the protocol to predict the proper binding orientation of the two proteins. This serves as a test of the loop modeling and interface predictions of the protocol. Here we present the protocol itself, as used for design or in these benchmarks, and the results of these fixed-sequence structure prediction benchmarks.

Methods

AnchoredDesign protocol

The AnchoredDesign protocol is written as an application in the Rosetta3 software suite, and first released with the 3.3 release. It was designed from the ground up within the Rosetta3 framework and thus takes advantage of all the modularity and ease-of-use offered by that foundation.[19] The protocol uses a multistage Metropolis Monte Carlo search protocol, with large perturbational movements in a reduced centroid representation and smaller refining changes in a higher-resolution fully atomic phase. This sort of multistage centroid/fullatom protocol is common for Rosetta protocols.[26, 31, 32] Conceptually, the centroid phase is meant to sample conformational space widely and jump over relatively high energy barriers between conformations, whereas the fullatom phase is meant to minimize a centroid candidate structure into its local energy minimum. To accomplish this, the protocol iteratively samples loop conformations and sidechain conformations, with interspersed opportunities for design. Figure 2.3 offers a diagram of program flow summarizing the major steps of the protocol.

The first phase of the protocol is the centroid sampling phase, using Rosetta's reduced-sidechain centroid representation and scorefunction.[31] Centroid mode reduces the complexity of the side-chain packing problem while the search function tries to consider larger changes to the protein structure. The centroid sampling phase consists of many Monte Carlo cycles of loop remodeling and minimization. Two types of loop remodeling can be performed here: perturbation followed by cyclic coordinate descent closure (CCD) [26, 33], or “kinematic” (KIC) loop remodeling [34, 35]. Note that for either case, loop modeling proceeds slightly differently than previously published to

account for the anchor; see details below. No sidechain optimization is necessary during centroid-mode perturbation, so after loop closure the algorithm proceeds directly into gradient minimization.[31] Backbone torsions at flexible loop positions are minimized to ensure good loop conformations and to perfect loop closure in the CCD case.

The second phase of the protocol is the refinement phase, which uses a fully atomic representation of the proteins. The use of a fullatom scorefunction, along with smaller-scale changes tested by Monte Carlo, allows this phase to refine the candidate structure produced in the perturbation phase. Here, CCD and KIC loop remodeling are also both available, although the CCD steps are softened to suggest smaller protein changes (described below). After each loop closure step, a quick fixed-sequence rotamer relaxation is performed [36], followed by a gradient minimization. The design portion of AnchoredDesign is incorporated during the fullatom phase by performing a sequence design and/or rotamer repacking on the interface region at user-defined intervals between loop remodeling cycles. Note that to reduce time spent repacking, all rotamer rearrangements used in AnchoredDesign feature automatic detection of the relevant residues: loop residues, their neighbors, and interface residues are automatically included, whereas residues outside those regions (the protein cores and distal surfaces) are not modified during repacking.

Loop modeling for AnchoredDesign

The KIC loop modeling protocol has been modified slightly from its original published implementation to allow for the constancy of the anchor. Normally, KIC solves an equation to determine phi and psi torsions for 3 loop residues, the pivots. The solutions to the equation are those torsions that close the loop. KIC also optionally

selects new values for non-pivot torsions.[34] The modifications used here allow for the anchor positions to be excluded from the list of allowable pivots and modifiable non-pivot torsions; they do not otherwise affect the underlying algorithm at all.

CCD loop closure has also been slightly modified to account for the anchor. Normally, CCD closes a broken loop by iteratively altering phi and psi angles to attempt to bring the broken loop ends together.[33] Here, the anchor's torsions are held fixed during CCD; it has no effect on the algorithm other than introducing inflexible regions which act as a particularly long bond.

Rosetta loop sampling with CCD is normally paired with a perturbation step which breaks the loop and introduces diversity.[26] Here, multiple methods are offered for perturbation before CCD closure. In the perturbation phase, the simplest method offered is randomization of the phi/psi angles (within Ramachandran constraints) of several residues in the loop. Other options include several varieties of fragment-based [31] perturbation: pregenerated fragment sets, automatically generated sequence-specific fragment sets, or automatically generated sequence-nonspecific fragment sets are allowed. The former options are more useful for structure prediction; the latter for design (where the final sequence is not known during the perturbation phase, so fragments of varying sequence are appropriate). In the refinement phase, large loop rearrangements are not desired, so only randomization of phi/psi angles, within a few degrees and Ramachandran constraints, is offered as a method of generating variation before CCD. Fragment insertion is not performed during the fullatom refinement.

Because AnchoredDesign is an interface design tool, but not quite a docking tool, it is necessary to explain how loop remodeling can effect rigid-body sampling without

modifying the anchor or core of either protein. Figure 2.4 is a Rosetta fold tree diagram representing an AnchoredDesign fold tree, modeled after those in Wang et al.[26] Rosetta regularly updates atomic coordinates from internal coordinates (bond lengths, angles, and torsions) held by the atom tree and fold tree data structures.[19] The group of atoms moved by the rotation of any one bond is controlled by the connectivity of the atom tree, which is in turn set by the more general fold tree. In AnchoredDesign, the fold tree is built in such a way that the anchor residues are dependent only on the target protein, the anchor's loop depends on the anchor, and the scaffold (which is rigid around the loop) is dependent on the loop. This setup ensures that conformational changes to the anchor loop result in relative motion of the two proteins' cores: rigid-body sampling. Other surface loops are treated with a standard loop fold tree as in Wang et al.[26]

Methods for target selection

Because appropriate interfaces for testing the AnchoredDesign approach are only a small fraction of the available interfaces in the PDB, an automated method was created to find interfaces with loops resembling an anchored loop. This method has been released alongside AnchoredDesign as AnchorFinder within the 3.3 release. The AnchorFinder algorithm was written to help find appropriate benchmarking structures, but it can also suggest useful anchors against targets of biological interest.

AnchorFinder searches any number of input structures for the qualities that define an anchored interface. In particular, it searches for protein regions that are dominated by loop secondary structure (as determined by Rosetta's internal implementation of the DSSP secondary structure algorithm [37]) and contain large numbers of protein contacts involving two chains (which are therefore across an interface). AnchorFinder will output

a listing of the DSSP assignment and cross-interface neighbors for each residue in each structure studied, plus summaries for contiguous regions that meet user-specified thresholds for length, secondary structure, and number of cross-interface neighbors. Regions with many cross-interface neighbors represent candidate anchors. When using AnchoredDesign to create new interfaces, AnchorFinder can help identify plausible anchors, but for small numbers candidate target/partner structures, manual examination is sufficient.

To choose our benchmarking set, the highest-ranking results from AnchorFinder were examined individually. AnchorFinder was run against the entire PDB (snapshot May 2009). The top several hundred structures returned by AnchorFinder were filtered to ensure that the hits were biological dimers and had identifiable anchors. The remaining hits contained redundant copies of many biological interactions due to multiple structures of some interactions, and multiple copies of one interaction within an asymmetric unit. Single representatives of each biological interaction were chosen. In general benchmarking systems were chosen to have a variety of biological sources, structures, and functions. One benchmarking structure, 2obg [24], was chosen for its identity as a fibronectin monobody structure without it appearing in the top fraction of AnchorFinder results: it represents the sort of structure AnchoredDesign is intended to create.

Choosing anchors, loops, and designable positions

AnchorFinder's results strongly suggest candidate anchors for use with AnchoredDesign. In general, the anchors used for benchmarking in this work were chosen by examining the loop residues suggested by AnchorFinder and picking one that

either buried large amounts of surface area across the interface or choosing a residue with a cross-interface hydrogen bond. For the purposes of this benchmarking, only single-residue anchors were allowed, although the algorithm is compatible with longer contiguous anchors.

For the design case, anchors will be grafted into a different protein. It is therefore important to choose an anchor with some internal structure and/or a very well-defined interaction with the protein partner; examples might be 4 residues of a hairpin turn binding into a cleft or a phosphotyrosine binding an SH2 domain, respectively. Another possibility is the use of hot-spot residues [38, 39], including those determined by fast computational tools [40, 41]. Ultimately, anchor choice is a dimension of conformational space that must be searched by testing different anchors. Anchors can be evaluated computationally by examining the scores assigned by Rosetta to models using different anchors.

For the benchmarking presented here, the length for the remodeled loop containing the anchor was chosen by simply accumulating residues out from the anchor in both directions until non-loop secondary structure was encountered. Only this single loop was varied, although the code is compatible with multiple (non-anchored) surface loops on both sides of the interface.

In the design case, choice of flexible loops will be dependent on knowledge of the scaffold. Loops must have an absolute minimum of three mobile positions for KIC modeling to work.[34] Which loops and residues should be considered flexible, which scaffold loop should accept the anchor insert and at what position, and what length the

loops should be must be determined manually by feeding different inputs to AnchoredDesign and comparing the quality of the resulting models.

Similarly, the choice of designable positions is dependent on knowledge of the scaffold. Scaffolds are presumably chosen on the basis of experimental experience with their tolerance to mutation (for example, fibronectin monobodies [22] or diverse other scaffolds [25]). The protocol assumes, but does not require, that the designable positions are all on flexible loops on one side of the interface (one-sided design). It will nevertheless accept two-sided design problems or non-loop design positions. Designable positions that are near neither flexible loops nor the interface may fail to be designed as desired, because the protocol automatically freezes those portions of the protein.

Creating starting structures

For the benchmarking in this paper, inputs for AnchoredDesign were generated from the crystal structure interaction with little modification. Nonprotein atoms (waters, cryoprotectants, and in some cases ligands) were deleted. These were passed through a simple structure minimizer to relax out any clashes with the Rosetta scorefunction. This protocol, InterfaceStructMaker (Peter Benjamin Stranges, unpublished protocol) performs a full-protein minimization and packing. It was determined that this preparatory step had no effect on the RMSD of the best scoring models (data not shown); its purpose was to remove data artifacts due to clashes in the crystal structures. These minimized structures were then fed directly to AnchoredDesign. AnchoredDesign internally deletes unwanted starting structure information (loop conformation, sidechains) when performing the benchmarks described in this paper.

In the design case, preparation of AnchoredDesign starting structures is much more complicated, because the anchor must be grafted from one structure into another. AnchoredDesign has a companion protocol also released with Rosetta3.3, AnchoredPDBCreator, designed to take care of this process. Two structures embodying three protein regions are necessary: a structure of the target protein with the protein containing the anchor bound, and a structure of the scaffold. Coordinates for the anchor, target, and scaffold are extracted into separate PDB files and offered as inputs to AnchoredPDBCreator, along with a file specifying what scaffold positions form the anchor loop and which positions the anchor should occupy. AnchoredPDBCreator inserts the anchor into the scaffold loop, closes the scaffold loop using CCD, and aligns the anchor (still rigid within the scaffold) with its binding site on the target. This resulting structure has the anchor and target correctly oriented (although the scaffold might interact poorly or eclipse the target), and is suitable as input to AnchoredDesign. The process is described in the first three subpanels of Figure 2.1, panel B.

Performing modeling

Workflow for AnchoredDesign is much like other Rosetta protocols: create and tweak input files, feed them to a cluster supercomputer to run tens of thousands of trajectories, then sift through the results. AnchoredDesign requires starting structures (outlined above), anchor and loop specifications (also outlined above), optionally a fragments file, and a resfile when performing design. These file formats and AnchoredDesign command line options are described in the Rosetta3.3 documentation. Briefly, options can be used to tweak the intensity of packing, control scorefunction and

minimization settings, and control the length and temperature of the two Monte Carlo sampling phases.

For the benchmarking case, sufficient results to generate a score vs. RMSD metric plot are all that is required; this tends to be several thousand structures.

For the design case, the search space is much larger and the correct answer is not known, so generating many tens of thousands of structures for a particular design problem is appropriate. The protocol cannot perform insertions or deletions, or slide the anchor's position within the loops, so testing scaffold variants in this vein is highly recommended. It is also a good idea to use the results of one round of modeling to inform the next: if one round of modeling shows that a particular loop length never results in a tight interface, throw that series of structures out.

The starting structure produced by AnchoredPDBCreator is very rough and does not consider scaffold-target interactions. It is always necessary to run AnchoredDesign on these structures with sufficient perturbation-phase cycles to get a reasonable alignment of the two partners. Later modeling beginning from better structures can run through only the refinement phase (option AnchoredDesign::refine_only) to find the lowest energy sequences possible.

The optimum settings for the length of the perturbation and refinement phases of AnchoredDesign are system-specific. A good starting point would be 500-1000 perturbation cycles, followed by twice that many refinement cycles. The option AnchoredDesign::refine_repack_cycles controls how often a full repacking/design step is performed during the refinement phase; this option should not be less than 50 (more than

that is designing needlessly frequently) and should not be more than 1/4 of the total refine cycles (or design is too infrequent).

Analyzing results

AnchoredDesign results are analyzed similarly to other Rosetta protocols'. The resulting structures and scorefile will contain the summed and individual scores, per-residue, for each term in the Rosetta scorefunction. Choosing the most likely models means choosing the lowest-scoring structures. AnchoredDesign also features a series of extra analysis tools to help highlight the better structures. These tools are implemented as Movers [19] which allows their analysis to be easily added to other protocols. Table 2.1 annotates the scorefile, and Figure 2.S1 annotates the extra analysis output appended to result PDB files.

InterfaceAnalyzerMover examines the quality of the interface in the final model. Included considerations are the burial of solvent-accessible surface area (SASA), the energy of binding, and the number and location of unsatisfied hydrogen bonds in the interface. These are important because AnchoredDesign optimizes stability of the complex (total energy), not binding energy.

LoopAnalyzerMover examines the quality of the flexible loops. It emphasizes the scorefunction terms relevant to loop closure (standard terms rama, dunbrack, and omega, [31] along with the chainbreak [26] term. It also prints the torsion angles of loop residues and the peptide bond distances. These data make it easy to spot poorly closed loops underpenalized by the Score12 scorefunction. These data are included at the end of the PDB output, as shown in Figure 2.S1.

The benchmarking presented here also triggers an extra suite of RMSD analyses which examine the similarity of the result structure to the correct complex. These include the RMSD fields in Table 2.1, and are further discussed in the Results.

Results

Selected models

In order to test the AnchoredDesign protocol, we used the AnchorFinder protocol to search for protein dimers with naturally-occurring anchor sequences where a residue of one partner is deeply buried into the other partner and part of an interfacial loop. Table 2.2 lists the structures' identities along with the anchors and loops chosen for benchmarking. All anchors are single residues. Loop length varies from 8 to 16 residues. Represented structures include homodimers of various functions (1fc4, 1qni, 2qpv, 3dxv, 1u6e, 2bwn, 2hp2, 2wya, 3cgc, 3ean, 1fec), two antibody/antigen complexes (1jtp, 2i25), one enzyme/inhibitor complex (1zr0), one engineered binder/target complex (2obg), and one nonbiological crystal dimer (1dle) chosen as a test of weak interactions. The crystal structures' resolution ranges from 1.7-2.75 Å, and the SASA buried in the interface ranges from 1400 to 11,800 $Å^2$.

Overall quality of predictions

The AnchoredDesign protocol was challenged with a benchmark where it was given a dimer structure with the rigid-body orientation, interface side chains, and an interfacial loop's conformation deleted. With knowledge of the position and conformation within the interface of one residue (the anchor) of the deleted loop, AnchoredDesign was asked to predict the correct loop structure and rigid-body orientation of the two proteins. Due to the fixation of the anchor, the backbone degrees

of freedom and rigid-body orientation are treated simultaneously by loop closure (Figures 2.2 and 2.4). Beyond this prediction experiment, two further experiments were performed to diagnose the source of failed predictions and provide performance comparisons. In one, the starting structure's loop and side chain information is not deleted: the simulation starts at the correct answer; the test is whether and how far the result drifts from the correct starting structure. These are often called "relaxed natives". In the second, the same information is not deleted, plus the AnchoredDesign protocol is instructed to skip the broad-sampling centroid perturbation step, and perform only the high-resolution refinement step. This is a more conservative calculation of the relaxed native population. If differences between these two forms of relaxed natives occur, it indicates that data are being lost during the centroid phase due to the low resolution of that protein representation.

AnchoredDesign's prediction of the interface was measured with three root-mean-square deviation (RMSD) metrics. The first metric was Cα RMSD of loop residues after superimposition, which gives a measure of how well the crystal loop was recapitulated. The second was backbone atom RMSD (after superimposition) of residues found at the interface, IRMSD, which measures how well the shape of the interface was recovered. The third metric was Cα RMSD for all residues on the moving side of the interface, LRMSD. This was calculated with superimposition on the nonmoving side of the interface, and thus gives a measure of whether the moving side of the interface is placed in its proper rigid-body position and orientation by AnchoredDesign. IRMSD and LRMSD are approximately equivalent to the metrics used in the communitywide CAPRI docking prediction experiment for gauging the quality of docked interfaces.[42] These

three RMSD calculations are labeled loop_CA_sup_RMSD, I_sup_bb_RMSD, and ch2_CA_RMSD in AnchoredDesign output (Table 2.1, Figure 2.S1).

Table 2.3 lists the RMSD metrics for the lowest-score structure predicted by AnchoredDesign for each input (compared to the minimized crystal structure used as input). In most cases, AnchoredDesign produces extremely accurate models for which each metric is below 1 Å RMSD; exceptions are further discussed below. Figures 2.5 and 2.S2 show each of the 16 complexes, including the minimized crystal structure and lowest-scoring result from AnchoredDesign. For most cases, the prediction is indistinguishable from the correct structure. Note that lowest-scoring is defined purely by Rosetta's standard Score12 scorefunction, plus a chainbreak term used in CCD loop modeling; these weights are listed in Supporting Table 2.S1.

Figures 2.6, 2.S3 and 2.S4 show score versus RMSD plots for each structure for each of the three metrics. These plots demonstrate that AnchoredDesign produces "funnels" for most of these experiments: all low-energy points are also low-RMSD, and RMSD rises with energy. This indicates the scorefunction grades these structures accurately and AnchoredDesign samples possible structures effectively. To visualize the space which AnchoredDesign samples, Supporting Figure 2.S5 shows the result of 100 trajectories for two PDBs (2obg and 1fc4). The protocol clearly samples many possible interfaces and rigid body orientations, but is nevertheless able to determine which is correct.

For most experiments, these predictions required 1 day on 128 2.33 GHz processors per experiment, which produces thousands to tens of thousands of trajectories depending on the size of the input structure. Some experiments required extra computer

time to accommodate larger proteins. Similar quantities of sampling were used for the relaxed native experiments; only 512 models per structure were produced for the conservative, fullatom-only relaxed natives.

Sampling errors

The most significant failure in this benchmark is the inability of AnchoredDesign to predict a correct interface for structure 3cgc. This structure is of a bacterial Coenzyme A disulfide reductase.[43] Figures 2.6, 2.S3 and 2.S4, panel 3cgc, show no low-RMSD points and no real score discrimination for the prediction experiment (black points). When the loop is not deleted prior to prediction, lower score, low-RMSD conformations are created (red and blue points). This demonstrates that it is not a flaw in the fullatom scorefunction but rather in sampling: the protocol never examines a loop resembling the correct loop, but it does give low scores to correct loops for relaxed natives. The relaxation experiment (red points), which runs AnchoredDesign as normal on an intact input loop, can be seen to hop out of the score well for correct structures and produces a smear of isoenergetic high-RMSD points. The fact that relaxed natives can lose their correct conformation implies that the problem may be a combination of errors. It could be that the low-resolution centroid scorefunction is unable to recognize the correct structure, and the fullatom phase's sampling is insufficient or ineffective in recovering low-RMSD structures for 3cgc.

Loop conformation errors

Structures 1qni and 2hp2 represent a pair of partial failures. These two structures score well on IRMSD and LRMSD metrics, but have relatively poor predicted loop RMSDs. In these two cases, AnchoredDesign finds and recognizes the correct interface

between proteins without folding the anchor loop correctly. Figure 2.7 shows the ten lowest-energy predictions for these two structures. In each case, all structures have the correct interface, but the loop itself is not predicted correctly, and does not converge onto a single prediction. Apparently, the energy well containing the correctly-bound interface is deep enough that the scorefunction can find it through minimization of loop degrees of freedom without accurately sampling the loop itself. Figure 2.S4 indicates that AnchoredDesign is probably failing to sample the correct loop in both cases, because the conservative relaxed natives (which maintain the correct loop conformation) are lowest RMSD and lowest scoring. This is probably related to the fact that these loops are longer than most loops in this benchmark (see Table 2.2, Discussion). AnchoredDesign's assignment of varied incorrect loops as isoenergetic is probably due to the lack of nearby steric restraints. The 1qni loop is mostly solvent-exposed, meaning that many solutions are possible. The 2hp2 loop borders a ligand absent during the modeling, freeing volume which then allows for many solutions.

Rigid body placement errors

Table 2.3 shows that the three metrics are slightly inconsistent for structure 2i25, a shark antibody bound to lysozyme.[44] Specifically, the loop RMSD and IRMSD metrics indicate a correct solution, while the LRMSD metric indicates a deviation. Examination of this structure (Figure 2.5, panel 2i25) shows that the interface and the loop are nearly the same set of residues: the CDR3 loop of the antibody. The relatively elongated antibody fold and the presence of a C-terminal tail pointing away from the interface amplify tiny errors in loop structure between the interface and antibody core to produce a relatively large displacement of the opposite side of the antibody. The

prediction shows clearly that AnchoredDesign is correct despite the slightly high LRMSD.

Comparison of loop closure methods

To test whether CCD or KIC loop closure was more appropriate for AnchoredDesign, all experiments were repeated using only CCD or KIC loop remodeling. Three general trends were found. First, AnchoredDesign with only CCD sampling is usually slower than AnchoredDesign with only KIC sampling; the default protocol using both falls in the middle (data not shown). Second, for a few structures (2obg and 2i25), more trajectories were required with KIC sampling to get results equivalent to CCD or combined sampling. Finally, all three methods produce results of equivalent qualities, as shown in Table 2.4. We also found that structure 2bwn, which can be seen in Figures 2.6, 2.S3 and 2.S4 to rarely sample the correct conformation, samples the correct conformation even less efficiently with only one style of loop remodeling. Taken together, these results imply that the default protocol, using both methods, is most appropriate for the design case where the correct structure is not known. KIC-only closure offers a speed benefit but may not work as well on all structures.

Effects of anchor displacement

To test how sensitive our results were to the exact position of the anchor, we performed an experiment where the anchor was randomly displaced from its correct position. The protocol was modified to allow the anchor to move freely, to allow relaxation of clashes introduced by the random displacement. The anchor was gently constrained to its original position to prevent these initial clashes causing a total ejection of the anchor. Supporting Table 2.S2 shows that AnchoredDesign is able to correctly

predict the interface in most cases in this modified experiment. This demonstrates that AnchoredDesign is not hypersensitive to the exact starting conformation at the interface; small errors and flexibility in anchor placement do not pose a problem.

Summary of results

Overall, these results are very encouraging for AnchoredDesign. Most structures tested are predicted with a very high level of accuracy, as seen in Figure 2.6 and Table 2.3. Note that none of the IRMSD versus score plots in Figure 2.6 demonstrate false funnels (there are no populations of low-energy, high-RMSD points). This indicates a lack of scoring failures, where incorrect structures are scored better than correct structures. Figure 2.6 also indicates that sampling failures are rare: only one case (3cgc) has no sub 2.5 Å RMSD points, and most cases have many sub-1 Å interface RMSD predictions. The few failures of the protocol can be attributed with some confidence to issues in the input (missing ligands, loop placement and length) rather than problems with the protocol itself. Additionally, the protocol is robust against small errors in anchor placement.

Discussion

The novel AnchoredDesign protocol described in this paper is capable of predicting the proper conformation of loop-mediated interfaces, as demonstrated by its benchmarking performance against 15 of 16 structures. In these predictions, AnchoredDesign is able to assemble one fixed, correct backbone with another mostly fixed, mostly correct backbone by knowing one point of contact and performing loop modeling to search rigid-body and loop conformational space. AnchoredDesign's success should not come as a surprise: Rosetta and many other docking protocols have been

proven to perform very well in fixed-backbone docking from bound backbones.[32, 42, 45] Rosetta has also succeeded at docking with loop remodeling.[26] That protocol was similar in its degrees of freedom to the one presented here, but different in its treatment of those freedoms. Recently, Rosetta-based docking algorithms were shown to correctly predict a trypsin/inhibitor complex like 1zr0 [46]; CAPRI Target 40, which Rosetta predicted at the highest level of accuracy.[47] We recently used AnchoredDesign to model an unknown interface between a fibronectin monobody and SH3 domain target, using a canonical polyproline binding interaction as an anchor.[48] We found that AnchoredDesign's models of fluorophore-tagged protein produced results consistent with experimental fluorescence.

The total (3cgc) and partial (1qni, 2hp2) failures of AnchoredDesign in this benchmark all share a common thread: these tests have loops longer than other, better-predicted complexes. These three tests have the longest loops at 14, 15, and 16 residues, respectively. Structure 2bwn, with a 14-residue loop, represents a borderline success. The lowest-energy structures are low in RMSD, but they are quite rare: careful examination of Figures 2.6, 2.S3 and 2.S4 (panel 2bwn) reveals only two low-energy points (both also low-RMSD) for structure 2bwn. For the other 12 structures that are clearer successes, the loops are 8, 10, or 12 residues. Loops were chosen not based on length but rather local structure characteristics: the loop length is the length of the loop in the native structure, chosen by extending from the anchor out to the closest sheet or helix. This dependence of result quality on loop length is not surprising; other authors have found similar dependencies with Rosetta's loop modeling protocols. The KIC protocol

was proven to work well on loops of length less than 13 residues [34] and Rosetta's other loop prediction techniques also work better on loops shorter than 13.[49]

Benchmarking successes and failures aside, the purpose of AnchoredDesign is not to provide another docking tool to work on known interfaces; it is to provide an interface design tool for the creation of new interfaces. This work demonstrates the ability of AnchoredDesign to address the flexible-backbone interface prediction aspect of the interface design problem. A necessary ingredient not tested here is the design aspect of AnchoredDesign, present in the protocol but not showcased by this benchmark. Known-structure benchmarking of this variety is not capable of testing both backbone flexibility and design quality at the same time: there is no way to know that the natural protein is the best possible structure and sequence at the interface, and so there is no reason to believe flexible-backbone design will converge to nature's solution. Fortunately, Rosetta has also proven to be very effective at the design problem.[6, 8, 36, 50] The anchor displacement experiment suggests that in the design case, small displacements and rotations at the anchor position could be searched to increase the probability that designable interfaces are sampled.

A complete test for the AnchoredDesign protocol will be a full pass from separated target and scaffold starting structures, through computational prediction of a binding sequence, to experimental verification that the model is correct.

Acknowledgements

The authors would like to acknowledge University of North Carolina at Chapel Hill Research Computing for their assistance in obtaining computer time. This work was supported by the US National Institutes of Health (GM073960), the Defense Advanced

Research Projects Agency (DARPA) and the W.M. Keck foundation. SML was also supported by a University of North Carolina Pogue Fellowship.

Figure 2.1: Anchor insertion
Panel A demonstrates the AnchoredDesign process with a simple cartoon. At left, we start with a known interaction between a target (cyan) and a natural partner (orange) with a characteristic interaction (the anchor, yellow). In the middle, we graft the anchor into the scaffold (magenta) to create a rough starting structure. At right, we fill out the scaffold-target interface with the AnchoredDesign protocol. Panel B demonstrates the process using protein structures for greater clarity (using the same color scheme).

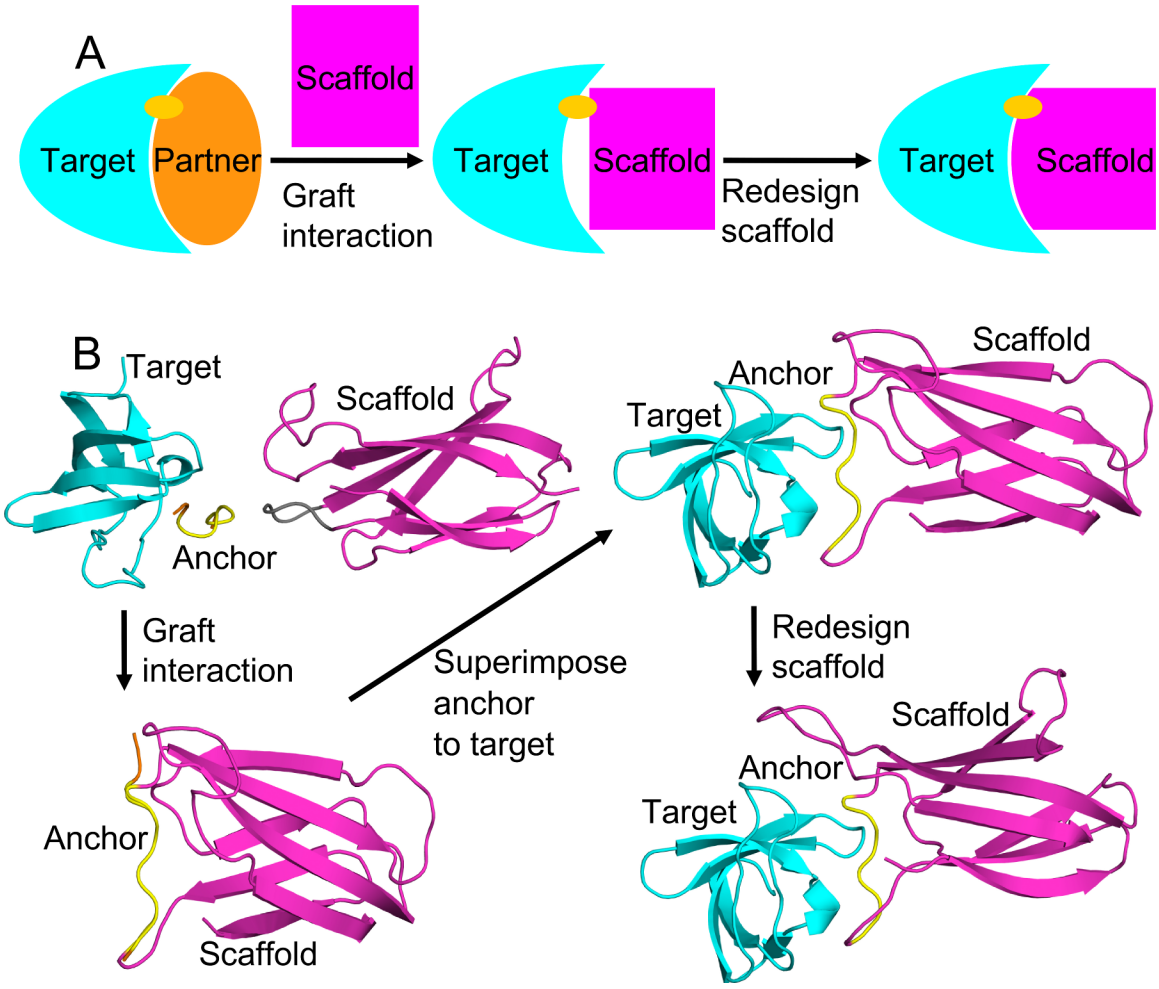


Figure 2.2: AnchoredDesign treatment of rigid-body and loop degrees of freedom
These complexes demonstrate how the AnchoredDesign protocol samples the rigid-body degree of freedom via loop sampling. These complexes are colored as in Figure 2.1: cyan for the target, yellow for the anchor, and magenta for the scaffold. The flexible residues of the loop containing the anchor are colored red. The two complexes are related only by the alteration of the backbone torsion angles of the red positions; the overall viewpoint has not been rotated (notice the targets are identical). Remodeling of this loop (and no other changes) produces a large rigid-body like change between the two partners, while leaving the anchor/target interface and both protein cores intact. This allows sampling of the target/scaffold interface without losing the anchor information.

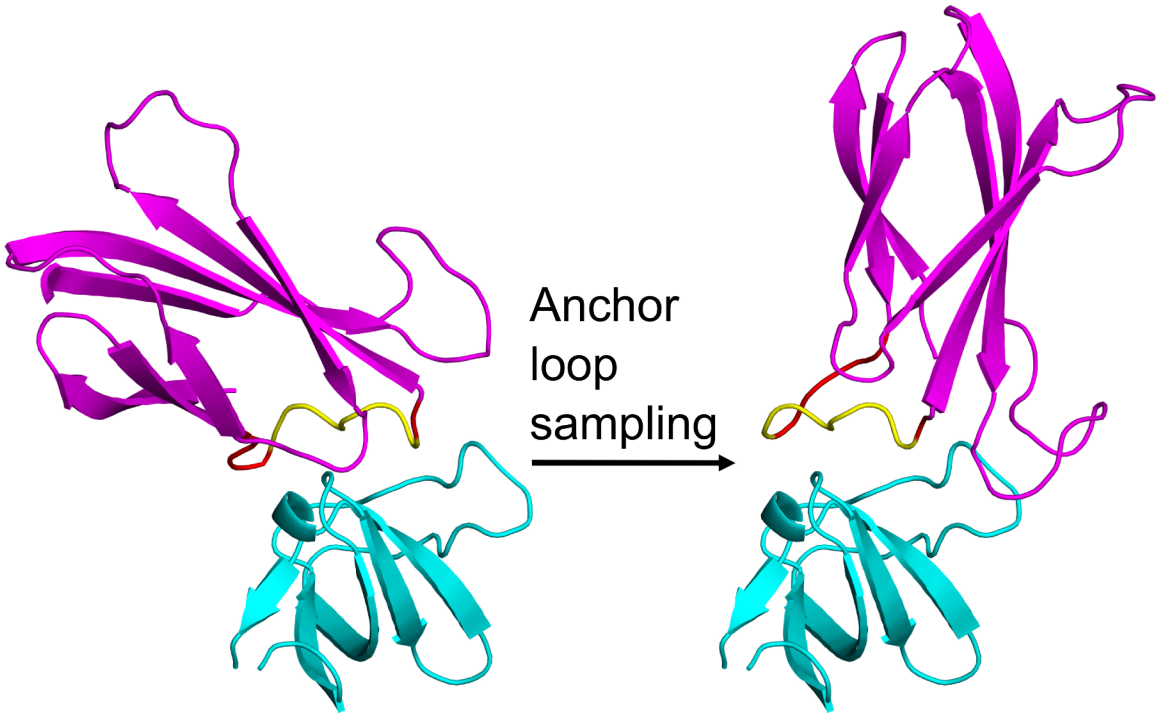


Figure 2.3: Protocol flowchart
This flowchart summarizes the AnchoredDesign protocol and process. In the top part, preliminary steps are marked in green. These steps are primarily manual but can be assisted with AnchorFinder and AnchoredPDBCreator. Initial steps vary depending on whether the benchmarking case for this paper, or the more general design case, is being addressed. The steps of the AnchoredDesign protocol are shown in blue and pink. The perturbation steps, performed in centroid mode, are in blue. The refinement steps, performed with a fully atomic representation, are in pink. In both portions of the protocol, many Monte Carlo cycles are performed; the results here used 500 perturb and 1000 refine cycles, but optimal cycle counts are best determined on a per-experiment basis.

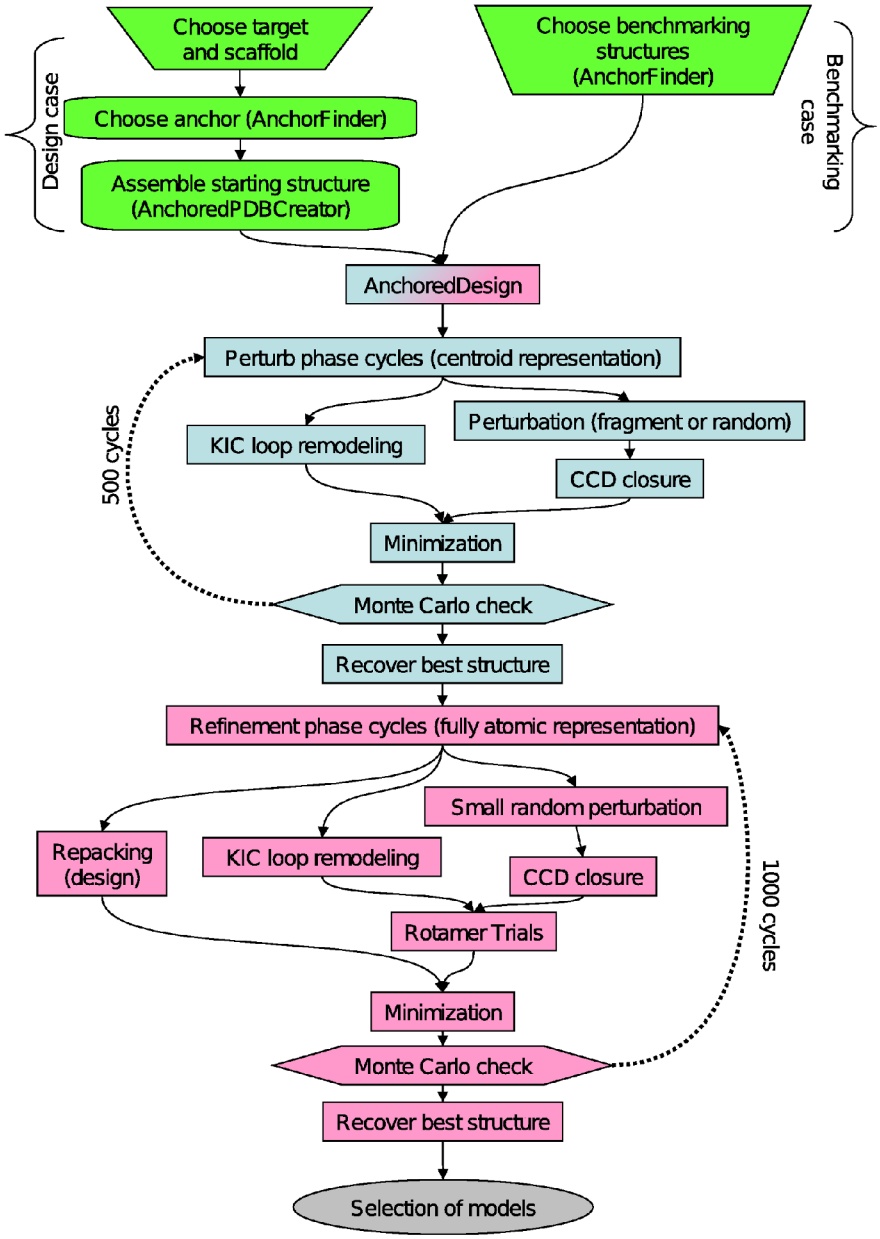


Figure 2.4: Fold tree diagram
This figure, modeled after the fold tree diagrams in Figure 1 of Wang et al. [26], demonstrates the kinematic connectivity that makes AnchoredDesign work. The arrows trace the direction of folding as Rosetta recalculates 3D coordinates from internal coordinates, starting at the green root residue. The upper and lower sections represent the target and scaffold respectively. Shaded regions represent rigid torsions (including the entire target and the core of the scaffold, in this case). Unshaded regions represent mobile torsions: the loops. All jumps between noncontiguous residues (dotted lines) are held rigid. AnchoredDesign embeds rigid torsions (the anchor, red) inside a loop, and affixes the anchor to the target by having the anchor's coordinates depend on the target instead of the scaffold in which the anchor is embedded. The scaffold is then dependent on the anchor via the mobile loop containing the anchor. Also allowed are arbitrarily placed other loops, handled with the standard loop fold tree.

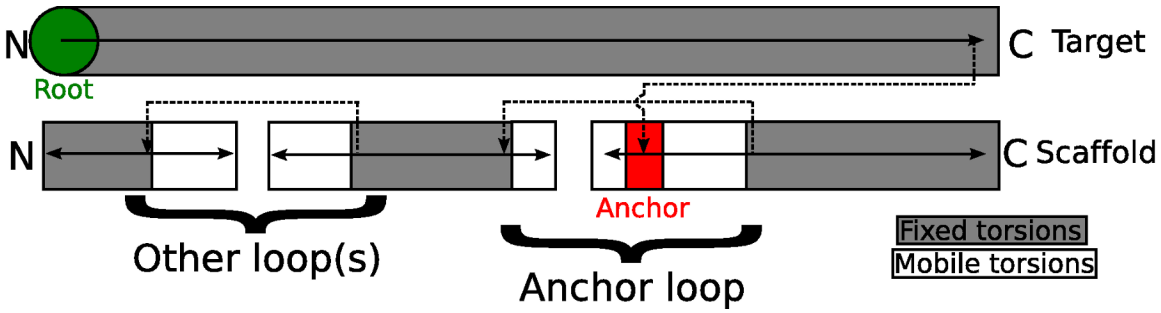


Figure 2.5: Best scoring prediction for 8 complexes
This figure demonstrates the relaxed crystal structure input and AnchoredDesign's lowest-score prediction for 8 of the 16 structures. The first and third rows show whole structures, and the second and fourth zoom in on the predicted loops. The nonmoving side of the interface is in green, the actual partner in cyan and the prediction in yellow. The predicted loop is red and the anchor is white. Structures are labeled with their PDB code in the lower right of each cell. For most structures, the predicted rigid-body placement and loop is indistinguishable from the relaxed crystal structure; 3cgc (lower left) is the exception. The other 8 structures are shown similarly in Figure 2.S2.

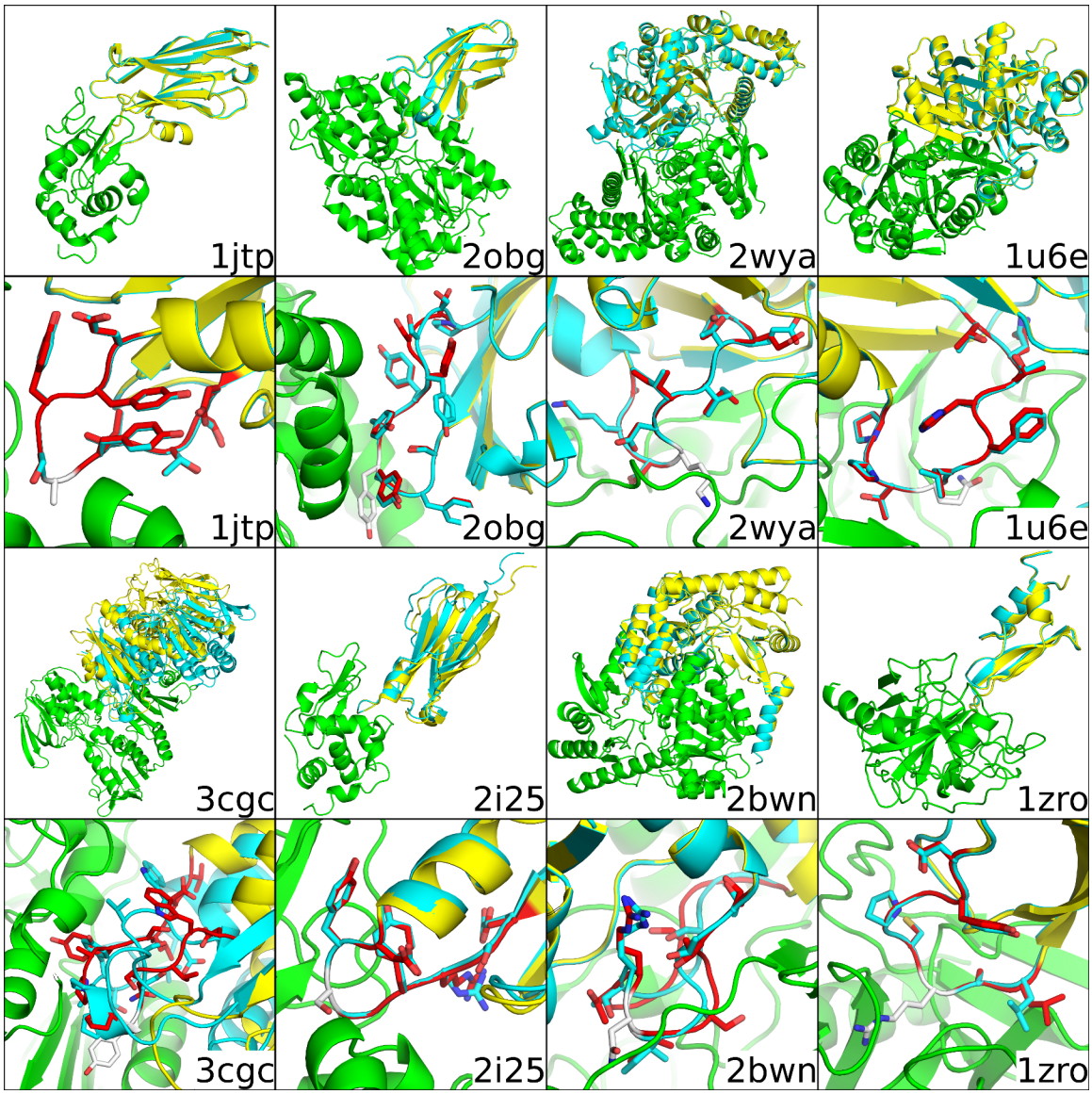


Figure 2.6: IRMSD versus score plots
This figure shows score versus IRMSD plots for each of the 16 structures. RMSD was calculated between the relaxed crystal structure (input) and AnchoredDesign's output. Plots are labeled with their PDB ID in the lower right of each cell. Black points are predictions, red points are relaxed native trajectories, and blue points are conservative relaxed native trajectories which skipped the centroid phase (see main text). Blue points may lie under red points, and red points may lie under black points. Some high-score points are out of view on all plots; all low-score points are present. On the RMSD (X) axis, the first, second, and third tic marks represent 1, 2.5, and 5 Ångstroms. Some plots are zoomed in beyond 5 or 2.5 Ångstroms and fewer tics appear.

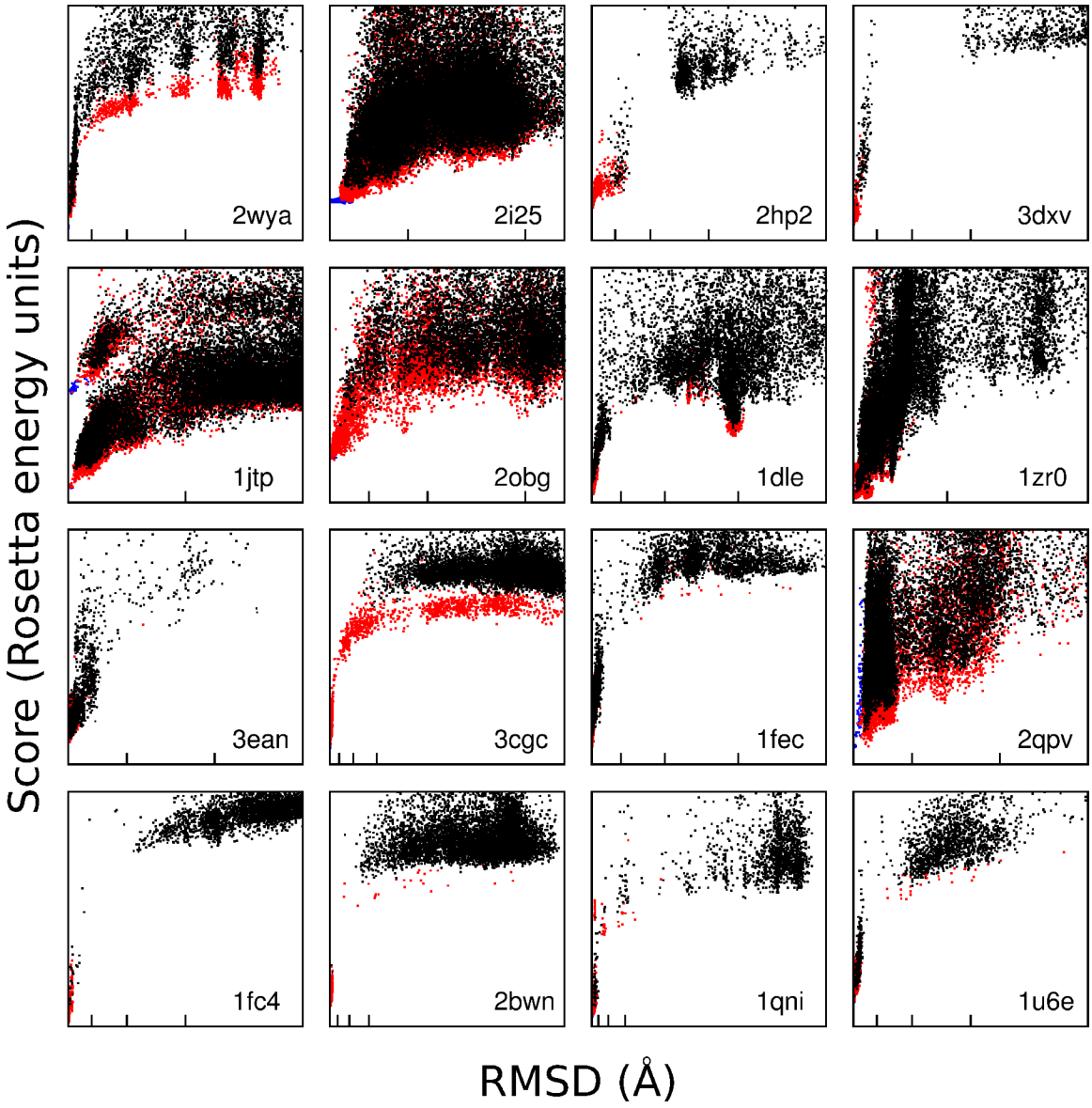


Figure 2.7: Poorly predicted 1qni and 2hp2 loops
This figure shows the insufficiency of AnchoredDesign's loop predictions for 1qni and 2hp2. In gray is the nonmoving side of the interface and in black is the correct structure. Each of the ten colors represents the loops of one of the top ten predictions by lowest energy. The predictions' protein cores are green. Notice that the loops themselves do not converge, but that the rigid-body placement and interface as a whole is correct (the green and black portions are overlaid towards the top of each panel).

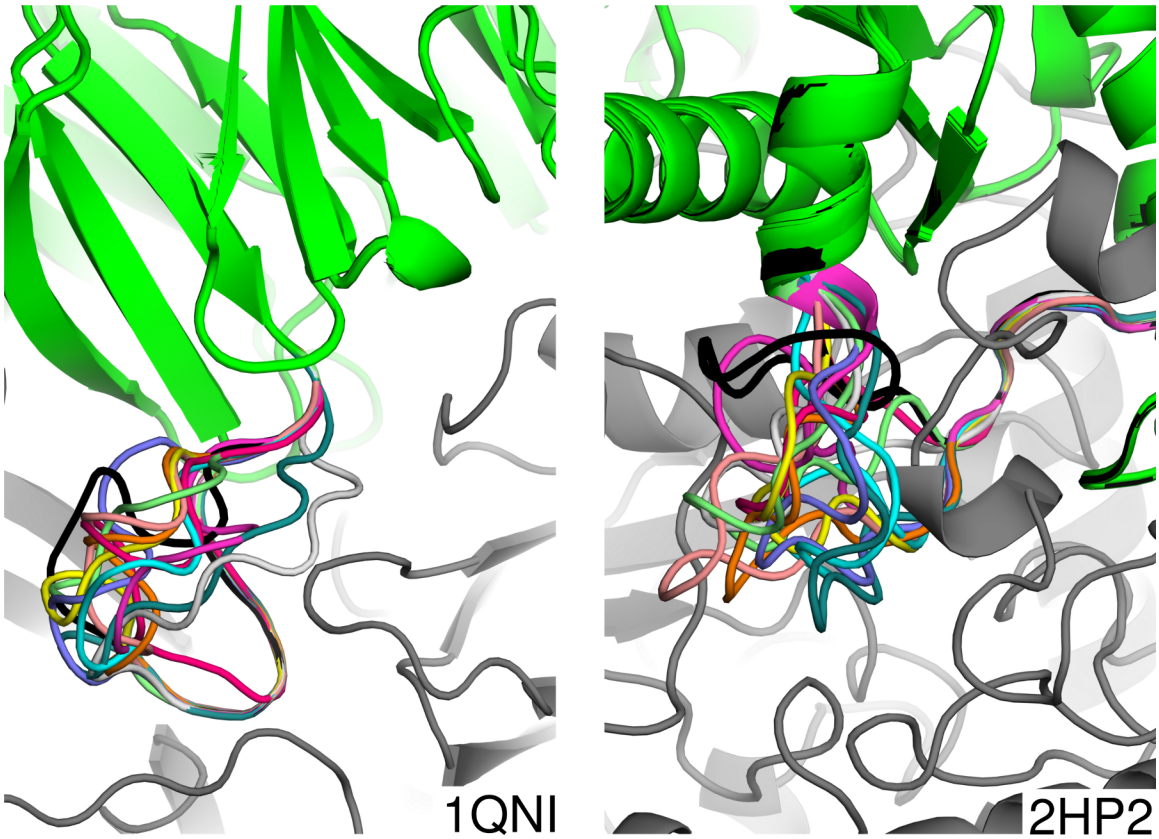


Figure 2.S1: Annotated AnchoredDesign results
This text represents excerpts from a PDB file result from an AnchoredDesign run. Vertical or horizontal ellipses (…) indicate where text has been excised for space or section boundaries. As in most PDB files, there are many thousands of ATOM records that compose the bulk of the file (A). The next section is the Rosetta-standard score section, listing the whole-structure and per-residue scores for all scorefunction terms. After B comes the output from LoopAnalyzerMover, including per-residue listings for several statistics (along with an annotation); C demarcates skipped residue lines. The utility of LoopAnalyzerMover is in finding small loop errors; for example residue 322 has a slightly longer peptide bond (1.339 Å, pbnd_dst) than the Rosetta standard 1.329 Å. Section D includes the protein sequence (useful in design mode) followed by output from InterfaceAnalzyerMover. This includes a listing of residues with unsatisfied, buried hydrogen bonds near the interface (clipped by E) and PyMOL selections of interface residues (clipped by F). The file ends with a long list of added statistics, which also appear in the scorefile (clipped by G) (see also Table 2.1).

```
ATOM   8184 3HZ  LYS B 520       7.595  15.887  51.686  1.00  0.00
```

A

```
label fa_atr fa_rep fa_sol fa_intra_rep pro_close fa_pair ...
weights 0.8 0.44 0.65 0.004 1 0.49 0.585 1.17 1.17 1.1 0.5 ...
pose -1824.66 224.591 887.884 4.10957 4.47253 -50.5034 ...
ALA_p:NtermProteinFull_1 -0.589162 0.0369979 0.330737 ...
ASP_2 -2.63173 0.568146 1.90627 0.00722233 0.00358695 ...
```

B

```
LoopAnalyzerMover: unweighted bonded terms and angles (in degrees)
position phi_angle psi_angle omega_angle peptide_bond_C-N_distance ...
 pos phi_ang psi_ang omega_ang pbnd_dst    rama  omega_sc dbrack pbnd_sc   cbreak
 314  -101.7   114.9    -171.6    1.329  -0.863     0.712 0.0098   -3.44    0.664
```

C

```
 321    -67   135.6       174    1.329   -1.37     0.359   2.88   -3.41    0.336
 322  -94.25   157.3       179    1.339  -0.402   0.00959   2.68   -3.18   0.0418
 323  -137.4   115.3     175.4    1.329    1.13     0.216  0.226   -3.19    0.201
total_rama 0.507822
total_omega 3.63856
total_peptide_bond -33.7835
total_chainbreak 3.67884
total rama+omega+peptide bond+chainbreak -25.9583
```

D

```
SEQUENCE: ADPDESQSLSLCGMVKGTDYHKQPWQAKISVIRPSKGHESCMGAVVSEYFVLTAAHCFTVDDK...
Residues missing H bonds:
Residue      Chain       Atom
38     A     NE2
101    A     OE1
248    A     O
```

E

```
pymol-style selection for unstat hbond res
select start_5411_unsat, /start_5411//A/38-101+248+250+ + /start_5411//B/344...

pymol-style selection for interface res
select start_5411_interface,
/start_5411//A/31+32+33+34+35+36+37+38+39+40+41+54+56+57+59-60+61+62+64+65+66+...
```

F

```
LAM_total -25.9583
dSASA_int 2396.33
dG_separated -35.3379
dG_separated/dSASAx100 -1.47467
```

G

Figure 2.S2: Best scoring prediction for 8 complexes
This figure demonstrates the relaxed crystal structure input and AnchoredDesign's lowest-score prediction for 8 of the 16 structures. The first and third rows show whole structures, and the second and fourth zoom in on the predicted loops. The nonmoving side of the interface is in green, the actual partner in cyan and the prediction in yellow. The predicted loop is red and the anchor is white. Structures are labeled with their PDB code in the lower right of each cell. For most structures, the predicted rigid-body placement and loop is indistinguishable from the relaxed crystal structure. The other 8 structures are shown similarly in Figure 2.5.

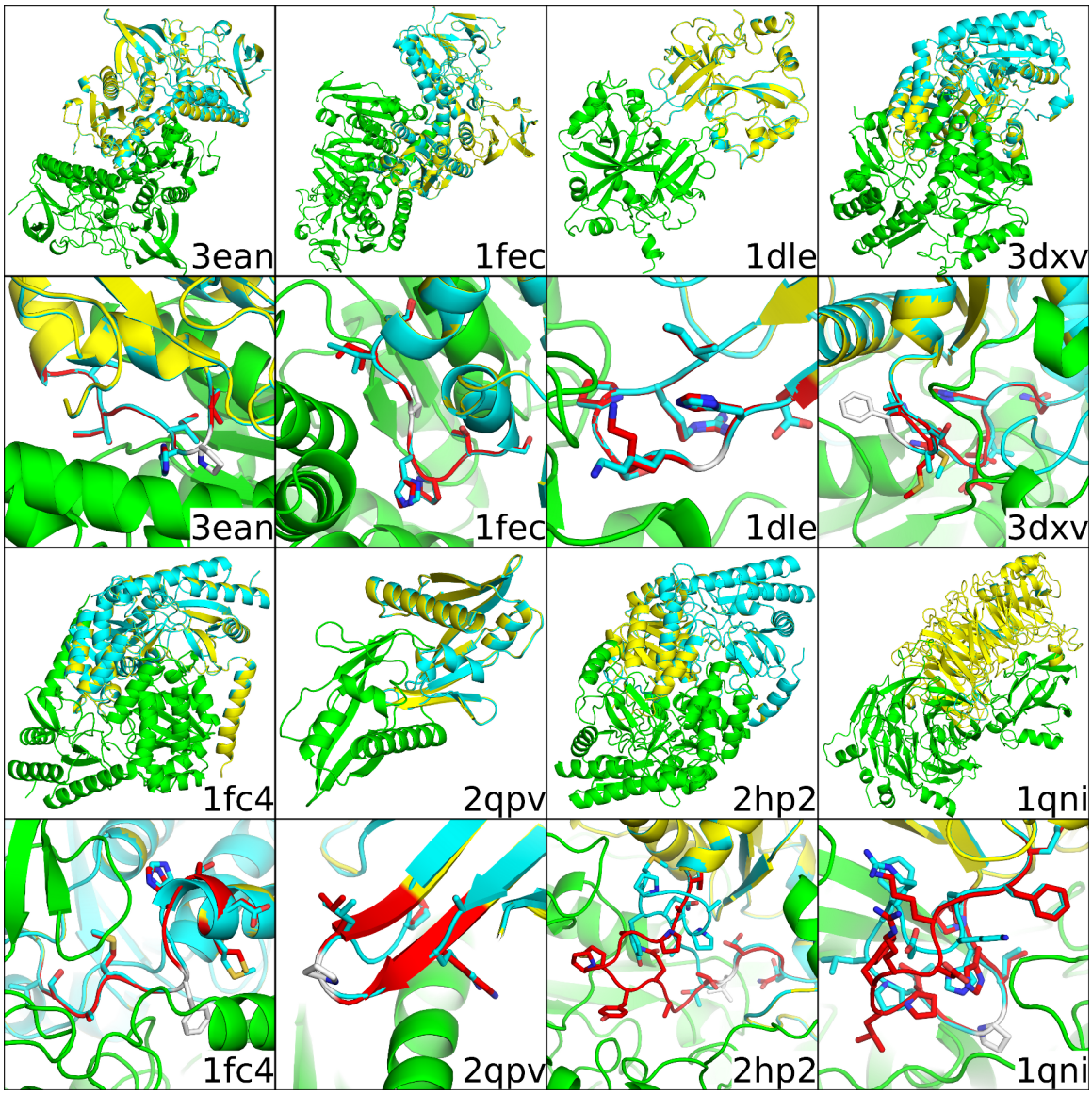


Figure 2.S3: LRMSD versus score plots
This figure shows score versus LRMSD plots for each of the 16 structures. RMSD was calculated between the relaxed crystal structure (input) and AnchoredDesign's output. Plots are labeled with their PDB ID in the lower right of each cell. Black points are predictions, red points are relaxed native trajectories, and blue points are conservative relaxed native trajectories which skipped the centroid phase (see main text). Blue points may lie under red points, and red points may lie under black points. Some high-score points are out of view on all plots; all low-score points are present. On the RMSD (X) axis, the first, second, and third tic marks represent 1, 2.5, and 5 Ångstroms.

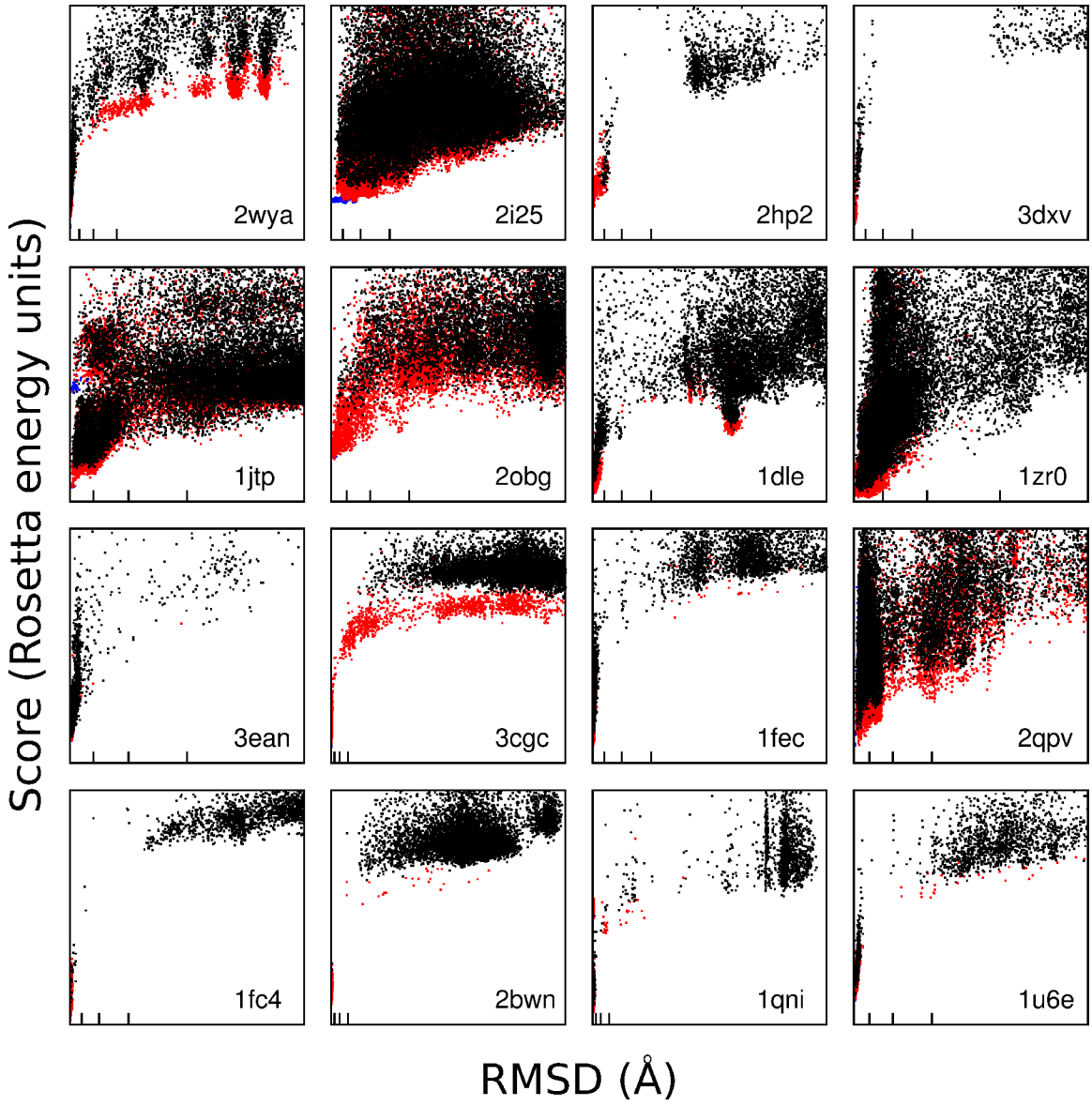


Figure 2.S4: Loop RMSD versus score plots
This figure shows score versus loop RMSD plots for each of the 16 structures. RMSD was calculated between the relaxed crystal structure (input) and AnchoredDesign's output. Plots are labeled with their PDB ID in the lower right of each cell. Black points are predictions, red points are relaxed native trajectories, and blue points are conservative relaxed native trajectories which skipped the centroid phase (see main text). Blue points may lie under red points, and red points may lie under black points. Some high-score points are out of view on all plots; all low-score points are present. On the RMSD (X) axis, the first, second, and third tic marks represent 1, 2.5, and 5 Ångstroms. Some plots are zoomed in beyond 5 or 2.5 Ångstroms and fewer tics appear.

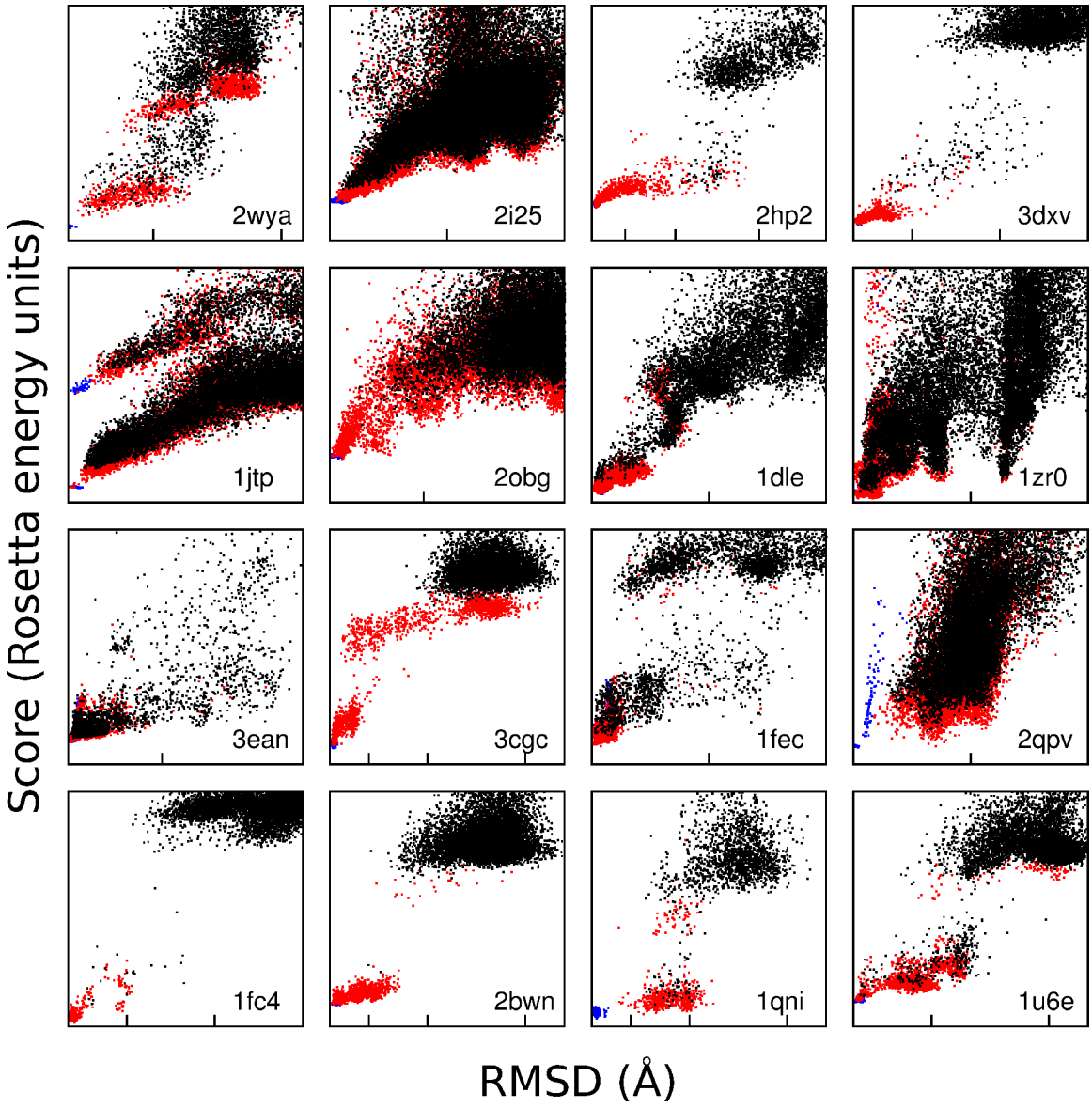


Figure 2.S5: 2obg and 1fc4 sampling
This image shows the range of sampling in 100 convenience-sample result structures from the standard protocol. Panel A contains 100 predictions for PDB 2obg, and panel B contains the correct 2obg structure for comparison. Panels C and D contain the same for the 1fc4 system. In each panel, the fixed side of the interface is at the bottom in green and the spread of possible docking orientations are across the top in a rainbow of colors. It can be seen that many possible interfaces and rigid body degrees of freedom are sampled. Furthermore, this sample represents result structures; many more orientations are sampled within a trajectory but rejected by Monte Carlo or superseded by better structures.

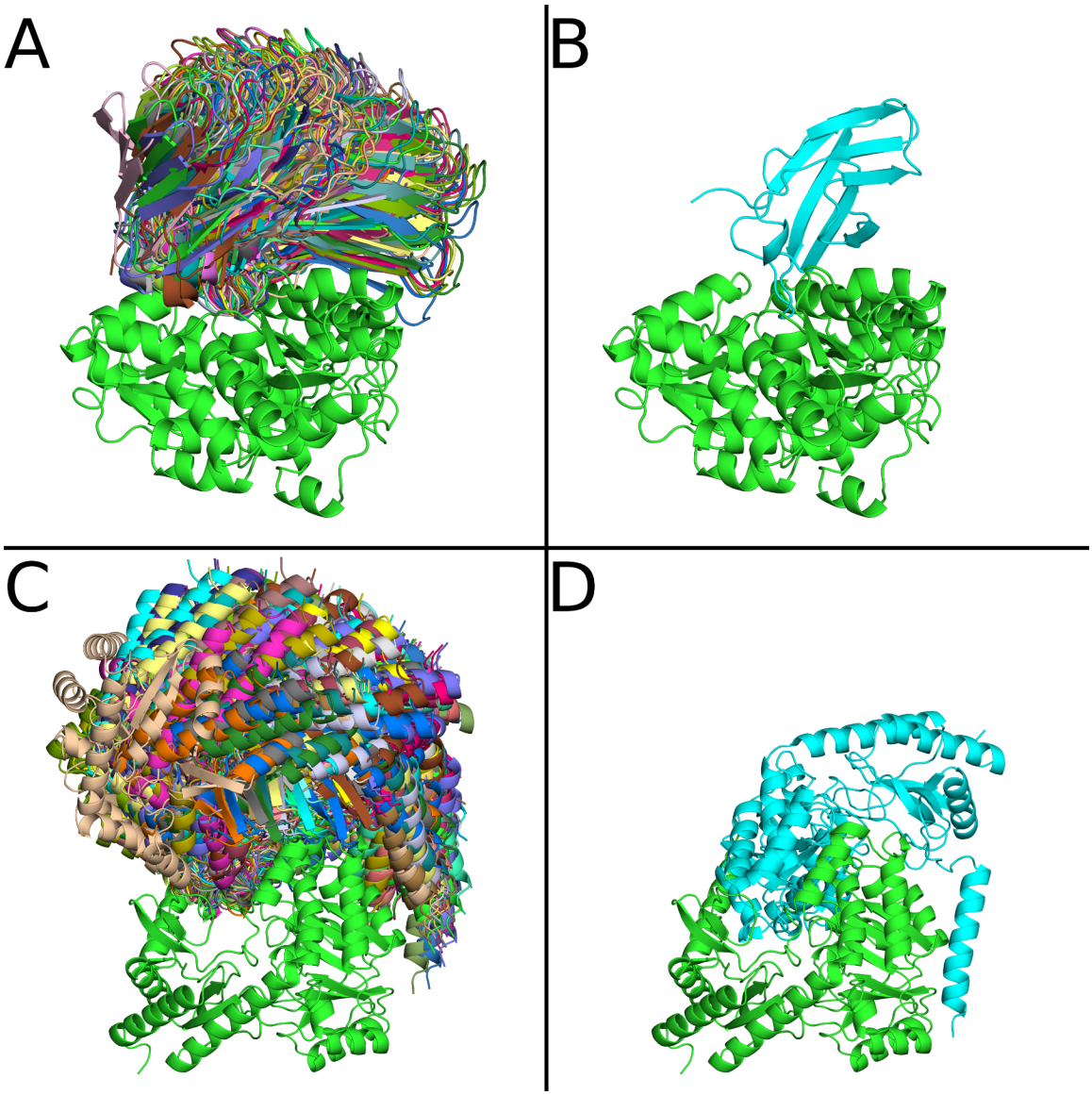


Table 2.1: Annotated scorefile headers
This table annotates the regions of the scorefile produced by AnchoredDesign. The first column lists metrics useful for analyzing benchmark or designed structures, and the second lists the meanings of those metrics. Of particular interest are total_score, loop_CA_sup_RMSD (loop RMSD), I_sup_bb_RMSD (IRMSD), and ch2_CA_RMSD (LRMSD), which provide the metrics used for the other plots and tables in this paper. Scorefile columns not listed here are either scorefunction terms [31, 36] or InterfaceAnalyzerMover metrics not useful for AnchoredDesign.

Metric	Purpose
CA_sup_RMSD	Whole complex Cα RMSD after superimposition
I_sup_bb_RMSD	Interface main chain atom RMSD, after superimposition
ch1_CA_RMSD	Chain 1 Cα RMSD without superimposition
ch1_CA_sup_RMSD	Chain 1 Cα RMSD with superimposition
ch2_CA_RMSD	Chain 2 Cα RMSD without superimposition
ch2_CA_sup_RMSD	Chain 2 Cα RMSD with superimposition
loop_CA_sup_RMSD	Loop residues' Cα RMSD with superimposition
dSASA_int	SASA buried by the interface
dG_cross	Interface binding energy, calculated from residue interactions between chains
dG_cross/dSASAx100	dG_cross, scaled by dSASA_int and a constant factor
dG_separated	Interface binding energy, calculated by separating components
dG_separated/dSASAx100	dG_separated, scaled by dSASA_int and a constant factor
delta_unsatHbonds	Number of unsatisfied hydrogen bonds in the interface
total_score	Weighted, summed score of the scorefunction
LAM_total	A sensitive descriptor of loop closure quality
description	The trajectory label (e.g., 2OBG_0001)

Table 2.2: Input structures and accessory data

This table collects structural parameters for the complexes used in this work, along with chosen parameters like anchor placement. The PDB column identifies the structure. The chains column identifies which complex within the PDB file was used (several have many complexes in the asymmetric unit). *: 2obg was not crystallized as a heterodimer; it was expressed as a fusion protein and crystallized as an infinitely domain-swapped polymer.[24] The resolution column contains the reported crystal structure resolution; all are reasonable. The dimer type column identifies the type of dimer. **: 1dle represents a crystal dimer rather than a biological one, making it a good test of weak interactions. The SASA column notes the area buried by the interface. The anchor and chain columns together identify the residue used as an anchor. The loop column identifies which residues (on the same chain as the anchor) were flexible. The loop length column collects the lengths of these loops. ***: 1dle has residues with insertion codes in the loops, leading to a longer loop than is obvious. The name column identifies the name and function of the protein as listed in the PDB.

PDB	Chains	Resolution (Å)	Dimer type	SASA (Å²)	Anchor	Chain	Loop	Loop length	Protein name
1dle	A/B	2.1	crystal dimer**	2,800	38	B	36-40***	8	Complement factor b; serine protease domain
1fc4	A/B	2	homodimer	10,000	74	B	69-79	11	2-amino-3-ketobutyrate coa ligase
1qni	A/B	2.4	homodimer	11,800	408	B	397-411	15	Nitrous oxide reductase
2qpv	A/B	2.35	homodimer	2,900	55	B	50-57	8	Uncharacterized protein atu1531
3dxv	A/B	2.21	homodimer	7,900	291	B	288-299	12	Alpha-amino-epsilon-caprolactam racemase
1u6e	A/B	1.85	homodimer	6,700	86	B	80-89	10	3-oxoacyl-[acyl-carrier-protein] synthase iii
2bwn	A/B	2.1	homodimer	8,500	85	B	77-90	14	5-aminolevulinate synthase
1jtp	M/B	1.9	heterodimer	1,600	104	B	99-108	10	Camelid antibody cab-lys3/lysozyme c
2hp2	A/B	2.7	homodimer	9,500	2306	B	2294-2309	16	Glutamate-1-semialdehyde 2,1-aminomutase
2wya	B/C	1.7	homodimer	5,900	1009	C	1003-1012	10	Hydroxymethylglutaryl-coa synthase, mitochondrial
2obg	A*	2.35	heterodimer	1,600	1080	A	1077-1086	10	Maltose binding protein/mbp74 monobody
2i25	M/O	1.8	heterodimer	1,400	91	O	86-93	8	Shark IgNAR antibody/lysozyme c
3cgc	A/B	2.3	homodimer	5,400	429	B	421-434	14	Pyridine nucleotide-disulfide oxidoreductase, class i
3ean	A/B	2.75	homodimer	7,900	473	B	467-474	8	Thioredoxin reductase 1
1fec	A/B	1.7	homodimer	6,400	459	B	456-463	8	Trypanothione reductase
1zr0	A/B	1.8	heterodimer	1,400	15	B	10-17	8	Trypsin/tissue factor pathway inhibitor-2 kunitz domain 1

Table 2.3: RMSD of lowest-scoring models

This table summarizes predictions of the AnchoredDesign benchmark. Each value represents the RMSD of the lowest-scoring model produced by AnchoredDesign for that structure using its standard protocol. The PDB column identifies the structure. The loop RMSD, IRMSD, and LRMSD columns describe the RMSD of the lowest-scoring prediction against the relaxed input structure. The input columns compare AnchoredDesign's output to the scorefunction-minimized crystal structures used as input. The crystal columns compare the same output to the unrelaxed crystal structures. The low values throughout indicate that AnchoredDesign does a good job recovering native interfaces starting from extended loops. The similarity of the input and crystal columns indicates that the relaxed starting structures are not far from the crystal structures.

PDB	Input loop RMSD (Å)	Crystal loop RMSD (Å)	Input IRMSD (Å)	Crystal IRMSD (Å)	Input LRMSD (Å)	Crystal LRMSD (Å)
1dle	0.09	0.42	0.04	2.24	0.16	1.69
1fc4	0.40	0.43	0.08	0.57	0.08	0.70
1qni	1.99	1.98	0.53	0.94	0.47	1.00
2qpv	0.63	0.68	0.19	0.78	0.23	0.97
3dxv	0.62	0.65	0.14	0.55	0.13	0.60
1u6e	0.13	0.35	0.06	0.59	0.12	0.62
2bwn	1.30	1.35	0.30	0.75	0.29	0.86
1jtp	0.08	0.29	0.07	0.73	0.30	0.97
2hp2	3.43	3.43	1.37	1.52	1.24	1.45
2wya	0.24	0.28	0.06	0.54	0.06	0.88
2obg	0.34	0.36	0.26	0.61	0.58	0.74
2i25	0.11	0.28	0.28	0.61	2.18	2.28
3cgc	2.23	2.26	4.74	4.90	14.16	14.41
3ean	0.06	0.38	0.03	0.91	0.06	0.91
1fec	0.14	0.39	0.03	0.60	0.04	0.72
1zr0	0.17	0.40	0.14	0.47	0.47	1.63

Table 2.4: Comparison of loop closure methods

This table demonstrates the rough equality of results from KIC, CCD, and combined loop sampling. The default protocol columns are the same as in Table 2.3; the CCD and KIC columns were generated using either loop modeling type alone. Each column represents one of the RMSD metrics described in the Results. Each of the loop modeling choices is broadly equivalent. Notice that structure 3cgc has in lower RMSDs under the combined protocol than either loop sampling type alone; this is a falsely optimistic interpretation because AnchoredDesign is never correctly predicting 3cgc under any of these protocols. The structures produced for 3cgc are isoenergetic and it is coincidental that the combined protocol happens to have lower RMSDs.

	RMSD (Å) for default protocol			RMSD (Å) for CCD protocol			RMSD (Å) for KIC protocol		
PDB	loop	IRMSD	LRMSD	loop	IRMSD	LRMSD	loop	IRMSD	LRMSD
1dle	0.09	0.04	0.16	0.25	0.10	0.17	0.11	0.10	0.39
1fc4	0.40	0.08	0.08	0.34	0.09	0.12	0.50	0.11	0.10
1qni	1.99	0.53	0.47	2.85	0.60	0.56	2.21	0.48	0.43
2qpv	0.63	0.19	0.23	0.66	0.20	0.30	0.34	0.11	0.24
3dxv	0.62	0.14	0.13	0.51	0.13	0.12	0.78	0.20	0.18
1u6e	0.13	0.06	0.12	0.17	0.05	0.09	0.09	0.05	0.09
2bwn	1.30	0.30	0.29	1.77	0.40	0.58	1.80	0.36	0.39
1jtp	0.08	0.07	0.30	0.20	0.11	0.21	0.19	0.12	0.42
2hp2	3.43	1.37	1.24	4.56	1.60	1.44	4.68	1.43	1.28
2wya	0.24	0.06	0.06	0.71	0.14	0.12	0.12	0.02	0.03
2obg	0.34	0.26	0.58	0.45	0.33	1.05	0.47	0.36	0.70
2i25	0.11	0.28	2.18	0.22	0.40	3.17	0.11	0.24	2.00
3cgc	2.23	4.74	14.16	3.08	19.38	56.97	2.66	6.63	21.68
3ean	0.06	0.03	0.06	0.06	0.02	0.04	0.07	0.03	0.06
1fec	0.14	0.03	0.04	0.14	0.03	0.02	0.11	0.02	0.03
1zr0	0.17	0.14	0.47	0.10	0.07	0.70	0.06	0.04	0.20

Table 2.S1: AnchoredDesign scorefunction

This table lists the Rosetta scorefunction terms used in the benchmarking experiments for AnchoredDesign. All terms and weights except chainbreak are the standard Score12 terms used for many Rosetta experiments. Chainbreak is used with CCD loop modeling; the weight of 2 was determined empirically and can be modified by an AnchoredDesign command line flag.

Scorefunction term	Weight
fa_atr	0.8
fa_rep	0.44
fa_sol	0.65
fa_intra_rep	0.004
pro_close	1
fa_pair	0.49
hbond_sr_bb	0.585
hbond_lr_bb	1.17
hbond_bb_sc	1.17
hbond_sc	1.1
dslf_ss_dst	0.5
dslf_cs_ang	2
dslf_ss_dih	5
dslf_ca_dih	5
rama	0.2
omega	0.5
fa_dun	0.56
p_aa_pp	0.32
ref	1
chainbreak	2

Table 2.S2: Effects of anchor displacement

This table lists the RMSD metrics (see results) for both the standard and anchor-displaced AnchoredDesign benchmarking experiment. The "standard" columns duplicate Table 2.3 for ease of reading; the "displaced" columns list the values from this new experiment. To produce displacement, the alpha carbon of the anchor residue (and accordingly, all other residues in the moving side of the interface) was translated to a random position within one Ångstrom in x, y, and z (an 8 square-Ångstrom cube) of its original position. To relax possible clashes caused by this displacement, the anchor was allowed to move instead of being held in its original position. It was gently constrained to its correct position using constraints were automatically generated between the anchor position's alpha carbon and the closest four alpha carbons across the interface. Each constraint was scored by a harmonic potential weighted to produce a score penalty of half a unit at 1 Ångstrom deviation. For comparison, total protein scores for these systems are in the range of hundreds to thousands, so half a score unit is weak.

PDB	Input loop RMSD (Å) standard	Input loop RMSD (Å) displaced	Input IRMSD (Å) standard	Input IRMSD (Å) displaced	Input LRMSD (Å) standard	Input LRMSD (Å) displaced
1dle	0.09	0.18	0.04	0.07	0.16	0.16
1qni	1.99	2.40	0.53	0.69	0.47	0.61
1fc4	0.40	0.40	0.08	0.09	0.08	0.11
2qpv	0.63	1.13	0.19	0.39	0.23	0.52
2wya	0.24	0.38	0.06	0.10	0.06	0.09
3dxv	0.62	1.55	0.14	0.35	0.13	0.32
1u6e	0.13	0.62	0.06	0.12	0.12	0.14
1jtp	0.08	0.20	0.07	0.11	0.30	0.24
2hp2	3.43	3.29	1.37	1.06	1.24	0.97
2obg	0.34	1.04	0.26	2.69	0.58	9.10
3cgc	2.23	4.54	4.74	6.71	14.16	17.06
2i25	0.11	0.23	0.28	0.32	2.18	2.49
3ean	0.06	0.28	0.03	0.05	0.06	0.06
1fec	0.14	0.22	0.03	0.04	0.04	0.05
1zr0	0.17	0.08	0.14	0.04	0.47	0.15
2bwn	1.30	4.50	0.30	4.99	0.29	11.52

References

1. Mandell DJ, Kortemme T. (2009) Computer-aided design of functional protein interactions. Nature Chemical Biology 5(11): 797-807. 10.1038/nchembio.251.

2. Kaufmann KW, Lemmon GH, Deluca SL, Sheehan JH, Meiler J. (2010) Practically useful: What the rosetta protein modeling suite can do for you. Biochemistry 49(14): 2987-2998. 10.1021/bi902153g.

3. Kortemme T, Baker D. (2004) Computational design of protein-protein interactions. Curr Opin Chem Biol 8(1): 91-97.

4. Karanicolas J, Kuhlman B. (2009) Computational design of affinity and specificity at protein-protein interfaces. Curr Opin Struct Biol 19(4): 458-463. 10.1016/j.sbi.2009.07.005.

5. Mandell DJ, Kortemme T. (2009) Backbone flexibility in computational protein design. Curr Opin Biotechnol 20(4): 420-428. 10.1016/j.copbio.2009.07.006.

6. Jha RK, Leaver-Fay A, Yin S, Wu Y, Butterfoss GL, et al. (2010) Computational design of a PAK1 binding protein. J Mol Biol 400(2): 257-270. 10.1016/j.jmb.2010.05.006.

7. Hu X, Wang H, Ke H, Kuhlman B. (2007) High-resolution design of a protein loop. Proc Natl Acad Sci U S A 104(45): 17668-17673. 10.1073/pnas.0707977104.

8. Murphy PM, Bolduc JM, Gallaher JL, Stoddard BL, Baker D. (2009) Alteration of enzyme specificity by computational loop remodeling and design. Proc Natl Acad Sci U S A 106(23): 9215-9220. 10.1073/pnas.0811070106.

9. Chevalier BS, Kortemme T, Chadsey MS, Baker D, Monnat RJ, et al. (2002) Design, activity, and structure of a highly specific artificial endonuclease. Mol Cell 10(4): 895-905.

10. Huang PS, Love JJ, Mayo SL. (2007) A de novo designed protein-protein interface. Protein Sci 16(12): 2770-2774.

11. Haidar JN, Pierce B, Yu Y, Tong W, Li M, et al. (2009) Structure-based design of a T-cell receptor leads to nearly 100-fold improvement in binding affinity for pepMHC. Proteins-Structure Function and Bioinformatics 74(4): 948-960. 10.1002/prot.22203.

12. Reynolds KA, Hanes MS, Thomson JM, Antczak AJ, Berger JM, et al. (2008) Computational redesign of the SHV-1 beta-Lactamase/beta-lactamase inhibitor protein interface. J Mol Biol 382(5): 1265-1275. 10.1016/j.jmb.2008.05.051.

13. Lippow SM, Wittrup KD, Tidor B. (2007) Computational design of antibody-affinity improvement beyond in vivo maturation. Nat Biotechnol 25(10): 1171-1176. 10.1038/nbt1336.

14. Sammond DW, Eletr ZM, Purbeck C, Kimple RJ, Siderovski DP, et al. (2007) Structure-based protocol for identifying mutations that enhance protein-protein binding affinities. J Mol Biol 371(5): 1392-1404. 10.1016/j.jmb.2007.05.096.

15. Song G, Lazar GA, Kortemme T, Shimaoka M, Desjarlais JR, et al. (2006) Rational design of intercellular adhesion molecule-1 (ICAM-1) variants for antagonizing integrin lymphocyte function-associated antigen-1-dependent adhesion. J Biol Chem 281(8): 5042-5049. 10.1074/jbc.M51045200.

16. Kortemme T, Joachimiak LA, Bullock AN, Schuler AD, Stoddard BL, et al. (2004) Computational redesign of protein-protein interaction specificity. Nat Struct Mol Biol 11(4): 371-379.

17. Shifman JM, Mayo SL. (2003) Exploring the origins of binding specificity through the computational redesign of calmodulin. Proc Natl Acad Sci U S A 100(23): 13274-13279.

18. Shifman JM, Mayo SL. (2002) Modulating calmodulin binding specificity through computational protein design. J Mol Biol 323(3): 417-423.

19. Leaver-Fay A, Tyka M, Lewis SM, Lange OF, Thompson J, et al. (2011) ROSETTA3: An object-oriented software suite for the simulation and design of macromolecules. Methods Enzymol 487: 545-574. 10.1016/B978-0-12-381270-4.00019-6.

20. Batori V, Koide A, Koide S. (2002) Exploring the potential of the monobody scaffold: Effects of loop elongation on the stability of a fibronectin type III domain. Protein Eng 15(12): 1015-1020.

21. Karatan E, Merguerian M, Han ZH, Scholle MD, Koide S, et al. (2004) Molecular recognition properties of FN3 monobodies that bind the src SH3 domain. Chem Biol 11(6): 835-844.

22. Koide A, Bailey CW, Huang XL, Koide S. (1998) The fibronectin type III domain as a scaffold for novel binding proteins. J Mol Biol 284(4): 1141-1151.

23. Koide A, Abbatiello S, Rothgery L, Koide S. (2002) Probing protein conformational changes in living cells by using designer binding proteins: Application to the estrogen receptor. Proc Natl Acad Sci U S A 99(3): 1253-1258.

24. Koide A, Gilbreth RN, Esaki K, Tereshko V, Koide S. (2007) High-affinity single-domain binding proteins with a binary-code interface. Proc Natl Acad Sci U S A 104(16): 6632-6637.

25. Skerra A. (2007) Alternative non-antibody scaffolds for molecular recognition. Curr Opin Biotechnol 18(4): 295-304.

26. Wang C, Bradley P, Baker D. (2007) Protein-protein docking with backbone flexibility. J Mol Biol 373(2): 503-519.

27. Lo Conte L, Chothia C, Janin J. (1999) The atomic structure of protein-protein recognition sites. J Mol Biol 285(5): 2177-2198.

28. Berman HM, Westbrook J, Feng Z, Gilliland G, Bhat TN, et al. (2000) The protein data bank. Nucleic Acids Res 28(1): 235-242.

29. Potapov V, Reichmann D, Abramovich R, Filchtinski D, Zohar N, et al. (2008) Computational redesign of a protein-protein interface for high affinity and binding specificity using modular architecture and naturally occurring template fragments. J Mol Biol 384(1): 109-119. 10.1016/j.jmb.2008.08.078.

30. Liu S, Liu SY, Zhu XL, Liang HH, Cao AN, et al. (2007) Nonnatural protein-protein interaction-pair design by key residues grafting. Proc Natl Acad Sci U S A 104(13): 5330-5335.

31. Rohl CA, Strauss CEM, Misura KMS, Baker D. (2004) Protein structure prediction using rosetta. Methods Enzymol 383: 66-+.

32. Gray JJ, Moughon S, Wang C, Schueler-Furman O, Kuhlman B, et al. (2003) Protein-protein docking with simultaneous optimization of rigid-body displacement and side-chain conformations. J Mol Biol 331(1): 281-299.

33. Canutescu AA, Dunbrack RL. (2003) Cyclic coordinate descent: A robotics algorithm for protein loop closure. Protein Sci 12(5): 963-972.

34. Mandell DJ, Coutsias EA, Kortemme T. (2009) Sub-angstrom accuracy in protein loop reconstruction by robotics-inspired conformational sampling. Nature Methods 6(8): 551-552. 10.1038/nmeth0809-551.

35. Coutsias EA, Seok C, Wester MJ, Dill KA. (2006) Resultants and loop closure. International Journal of Quantum Chemistry 106(1): 176-189. - 10.1002/qua.20751.

36. Kuhlman B, Dantas G, Ireton GC, Varani G, Stoddard BL, et al. (2003) Design of a novel globular protein fold with atomic-level accuracy. Science 302(5649): 1364-1368.

37. Kabsch W, Sander C. (1983) Dictionary of protein secondary structure: Pattern recognition of hydrogen-bonded and geometrical features. Biopolymers 22(12): 2577-2637. 10.1002/bip.360221211.

38. Clackson T, Wells JA. (1995) A hot-spot of binding-energy in a hormone-receptor interface. Science 267(5196): 383-386.

39. Bogan AA, Thorn KS. (1998) Anatomy of hot spots in protein interfaces. J Mol Biol 280(1): 1-9.

40. Kortemme T, Baker D. (2002) A simple physical model for binding energy hot spots in protein-protein complexes. Proc Natl Acad Sci U S A 99(22): 14116-14121.

41. Meireles LMC, Dömling AS, Camacho CJ. (2010) ANCHOR: A web server and database for analysis of protein–protein interaction binding pockets for drug discovery. Nucleic Acids Research 38(suppl 2): W407-W411. 10.1093/nar/gkq502.

42. Janin J. (2010) Protein-protein docking tested in blind predictions: The CAPRI experiment. Mol Biosyst 6(12): 2351-2362. 10.1039/c005060c.

43. Wallen JR, Paige C, Mallett TC, Karplus PA, Claiborne A. (2008) Pyridine nucleotide complexes with bacillus anthracis coenzyme A-disulfide reductase: A structural analysis of dual NAD(P)H specificity. Biochemistry 47(18): 5182-5193. 10.1021/bi8002204.

44. Stanfield RL, Dooley H, Verdino P, Flajnik MF, Wilson IA. (2007) Maturation of shark single-domain (IgNAR) antibodies: Evidence for induced-fit binding. J Mol Biol 367(2): 358-372. 10.1016/j.jmb.2006.12.045.

45. Daily MD, Masica D, Sivasubramanian A, Somarouthu S, Gray JJ. (2005) CAPRI rounds 3-5 reveal promising successes and future challenges for RosettaDock. Proteins 60(2): 181-186.

46. Schmidt AE, Chand HS, Cascio D, Kisiel W, Bajaj SP. (2005) Crystal structure of kunitz domain 1 (KD1) of tissue factor pathway inhibitor-2 in complex with trypsin. implications for KD1 specificity of inhibition. J Biol Chem 280(30): 27832-27838. 10.1074/jbc.M504105200.

47. Fleishman SJ, Corn JE, Strauch EM, Whitehead TA, Andre I, et al. (2010) Rosetta in CAPRI rounds 13-19. Proteins 78(15): 3212-3218. 10.1002/prot.22784.

48. Gulyani A, Vitriol E, Allen R, Wu J, Gremyachinskiy D, et al. (2011) A biosensor generated via high-throughput screening quantifies cell edge src dynamics. Nat Chem Biol 7(7): 437-444. 10.1038/nchembio.585.

49. Rohl CA, Strauss CEM, Chivian D, Baker D. (2004) Modeling structurally variable regions in homologous proteins with rosetta. Proteins 55(3): 656-677.

50. Siegel JB, Zanghellini A, Lovick HM, Kiss G, Lambert AR, et al. (2010) Computational design of an enzyme catalyst for a stereoselective bimolecular diels-alder reaction. Science 329(5989): 309-313. 10.1126/science.1190239.

Chapter 3

AnchoredDesign predicts qualities of monobody-SH3 interfaces

Introduction

Most activities in a cell are modulated by protein-protein interactions occurring in cell-spanning networks. One method to deeply probe these networks requires the development of highly specialized tools: affinity reagent biosensors useful for detecting the presence, intracellular localization, or activity of interesting cellular proteins. The idea of fluorescently tagged biomarkers is not recent [2], but it remains a powerful technique for learning how cellular proteins perform their functions. Here we discuss the development of such a biosensor for Src family kinases. The biosensor is able to report on the activation state of its target kinase by changing fluorescence intensity upon binding, and this chapter focuses on biophysical modeling investigating how this sensing works.

Src kinases and the SH3 domain

Src family kinases are involved in many cell signaling networks, and their dysregulation can lead to cancer [3, 4]. In fact, the family was originally discovered via the relevance of Src to tumors induced by Rous Sarcoma Virus. These kinases' importance both for normal cell function and cancer makes them primary targets for biosenors designed to further elucidate their function in cellular signaling networks.

A conserved mechanism for regulation of Src family kinases is their characteristic autoinhibition [5]. These kinases have a three domain structure: an SH3 (Src Homology

3) domain, an SH2 (Src Homology 2) domain, and a kinase domain. The domains occur in that order in the peptide chain, as demonstrated in Figure 3.1. In the inactive state, the SH3 domain binds intramolecularly to a linker between the SH2 and kinase domains, and the SH2 domain binds intramolecularly to a phosphotyrosine on a C-terminal tail. In this conformation, the SH2 and SH3 domains prevent the kinase domain from achieving a catalytically active conformation, and signaling is suppressed. When inhibition is relieved by some other protein binding to the SH3 and SH2 domains, signaling can become active. From the point of view of biosensing, the significance of this autoinhibitory arrangement is that if the SH3 domain is exposed, the kinase is not autoinhibited, which is correlated with activity. Therefore, biosensor binding to Src SH3 domain is an indicator of Src activity [1].

SH3 domain binding is well-studied and most SH3 domains bind their ligands in a certain canonical fashion [6]. SH3 ligands usually have a leucine- and proline-rich sequence with two critical prolines spaced two residues apart (PxxP), along with a basic residue just before or just after this motif. The ligand adopts a particular secondary structure called the polyproline II (PPII) helix for binding the SH3 domain. The PPII structure is observed both in the autoinhibitory SH3 domain ligand in Src family kinases, and their intermolecular regulators.

Monobodies to Src family kinases

Human fibronectin, domain type 3, repeat 10 (10FNIII), is well known as an adaptable scaffold for generating new protein-protein interfaces [7]. 10FNIII has a small β-sandwich fold with flexible loops. It has proven to be an attractive scaffold for evolutionary techniques aimed at deriving binders to many targets [7-10]. This utility

comes from 10FNIII's mix of antibodies' useful qualities without some of antibodies' downsides [7]. 10FNIII has the loop mutability and accompanying wide binding compatibility of the antibody variable domains, but does not have antibodies' large size, difficulty in expressing in simple cultures like *Escherichia coli*, or disulfide bonds which limit use as intracellular biosensors. 10FNIII based binders are often called monobodies, due to their single (mono) domain structure and their similarities to antibodies. Typically monobodies have been generated by randomizing the sequence of the FG and BC loops of 10FNIII and screening large numbers of candidate sequences with display techniques. Figure 3.2 demonstrates the 10FNIII fold, highlighting the varied FG and BC loops, and shows a crystal structure of an evolutionarily-derived monobody binding its target.

Karatan et al. generated an SH3-binding monobody, 1F11, using phage display against Src SH3 [9]. Sequencing of this monobody revealed a sequence in the FG loop compatible with canonical SH3-binding: RPLPSKP. This led to the hypothesis that 1F11 was binding to SH3 domains via its polyproline sequence, which was supported by a variety of experimental data. First, 1F11 competes for binding to Src SH3 with a canonically-binding peptide. Second, NMR experiments indicated similar chemical shift differences caused by binding this peptide or 1F11 to labeled Src SH3 domain. Third, reversion of the 1F11 FG loop containing the polyproline sequence abolished binding.

Monobody-based biosensors for Src family kinases

Gulyani et al. sought to extend this Src-SH3-binding 1F11 monobody into a useful intracellular biosensor [1]. They experimented with many methods of fluorescently functionalizing 1F11 with a series of merocyanine dyes, which fluoresce well in intracellular environments. The fluorescence quantum yield of these fluorophores

—effectively their brightness—is sensitive to the polarity of the environment around the fluorophore, especially the environment around the fluorophore's polar groups [1, 11]. Gulyani et al. intended to leverage this solvent sensitivity to generate a 1F11-based biosensor. After attaching a fluorophore to 1F11 such that the fluorophore's environment is altered by Src SH3 domain binding, they determined that fluorescence intensity change is detectable as a signal that SH3 binding had occurred. Src SH3 availability implies that the Src kinase domain is not autoinhibited and thus active, so a biosensor that indicates SH3 domain binding serves as a proxy for an intracellular, time-and-space localizable biosensor for Src kinase activity.

A biosensor of this form was successfully generated by mutating 1F11 position 24 to cysteine and attaching the dye Mero53 [1]. Mero53, as it exists attached to protein, is shown in Figure 3.3. Position 24 is in the BC loop, which contains phage-display-derived mutations in 1F11. The loop is expected to be involved in SH3 binding [9]. Sequence positions 53 and 55 in the fibronectin DE loop, adjacent to the BC loop, were found not to experience a fluorescence intensity change upon binding to Src SH3. To learn why position 24 did show an intensity change on binding and positions 53 and 55 did not, we modeled the complexes between Src SH3 and Mero53-functionalized 1F11 [1].

Methods

Modeling 1F11-SH3 complexes via AnchoredDesign

The AnchoredDesign module of Rosetta3, described extensively in Chapter 2, is designed to model protein-protein interfaces between known targets and scaffolds with flexible loops. In fact, the protocol was written with 10FNIII in mind as an ideal

scaffold. A second requirement of the protocol is its need for an anchoring interaction between the two proteins about which to perform tethered docking. The presence of a canonical polyproline motif in the FG loop of 1F11 [9] offers a plausible anchor for protein-protein docking via AnchoredDesign.

Before attempting to model the the complex between Src SH3 and Mero53-functionalized 1F11 and search for a source of the fluorescence intensity shift, we first modeled the complex in the absence of the fluorophore. We chose the 10FNIII domain out of a 2.0 Å crystal structure [12] (PDB 1FNF) as a starting model for 1F11. PDB 1QWF represents an NMR structure of a complex between Src SH3 domain and a peptide VSLARRPLPPLP [13], which partially matches the 1F11 FG loop sequence fragment RPLPSKP. The 1QWF structure therefore represents an ideal target (SH3) and anchor (RPLP) structure for AnchoredDesign.

The AnchoredDesign suite's AnchoredPDBCreator executable was used to assemble the 1FNF-derived 10FNIII scaffold with the 1QWF-derived RPLP anchor and Src SH3 target. The 1F11 FG loop is four residues shorter than 10FNIII's FG loop, due to both intended deletions to select for flatter interfaces and an unintentional deletion that was selected by phage display [7, 9]. Accordingly, 8 residues of 1FNF 10FNIII were deleted from the PDB file. The gap was replaced via AnchoredPDBCreator with a 4 residue anchor (RPLP) derived from 1QWF and aligned to the 1QWF SH3 domain. AnchoredPDBCreator uses Cyclic Coordinate Descent [14] to close the loop accepting the anchor, thus closing the extra four-residue gap. See Figure 2.1 for a graphical depiction of this technique. The AnchoredDesign demo included with Rosetta3.3 and further releases shows example command lines for this functionality.

To fully convert 10FNIII into 1F11, Rosetta's fixed backbone design protocol, fixbb [15], was used to computationally mutate the remaining mismatched 10FNIII residues in the BC and FG loops into their 1F11 counterparts. This was performed with the straightforward use of a resfile to pick the mutations needed.

AnchoredPDBCreator focuses on creating an acceptable conformation for the anchor loop relative to the scaffold (1F11), but it only aligns against the target (SH3) after loop modeling. This possibly poor conformation at this early stage is not problematic because the goal of the AnchoredDesign executable is to model this interface. The AnchoredPDBCreator-derived starting model was therefore handed off to AnchoredDesign for a brief round of modeling to eliminate any gross errors in the 1F11-SH3 interface. After ensuring a structure with no major clashes, the entire structure was repacked (again using fixbb) to ensure that all rotamers were considered acceptable by Rosetta's scorefunction. These steps are important because AnchoredDesign repacks positions based on their proximity to the target-scaffold interface, which shifts as a trajectory proceeds. If all positions in the structure are not pre-packed in this fashion, the resulting scores will contain cryptic contributions from uninformative repacking of residues transiently near the protein-protein interface, which moves as modeling proceeds.

These steps produced a medium-quality starting structure with which to search for plausible 1F11-SH3 complex conformations. Up to this point, the described computational procedures were aimed at creating a starting structure with certain qualities rather than an actual model of the interaction. AnchoredDesign was used to extensively sample the 1F11-SH3 interface based on this starting model, using the same

techniques described in Chapter 2. The models produced by this procedure, one of which is shown in Figure 3.4, are consistent with the known details of the interaction between Src SH3 and 1F11. The polyproline-anchor-containing FG loop is in contact with the SH3 domain due to an assumption of the model rather than a result, but the models are also consistent with the mutagenesis-derived suggestion that the BC loop is relevant for binding [9], as the BC loop is in contact with the SH3 domain.

Incorporation of Mero53 into Rosetta

With a reasonable model for the 1F11-SH3 interface in hand, the next step was to incorporate the Mero53 dye into Rosetta and then into the model. Rosetta3 uses a system of human-and-computer readable parameter files that convert into the `ResidueType` class created for Rosetta3's reboot [16]. A streamlined system for generating parameter files for arbitrary small-molecule chemical entities for use as Rosetta residue types was written for use with ligand docking [17]. This was later extended to allow for polymeric residue types that incorporate fully into proteins as non-canonical amino acids [18]. Collectively, these Rosetta3 tools make it very simple to treat the cysteine-linked merocyanine dye Mero53 as a large non-canonical amino acid, which Rosetta can handle readily. The modular organization of Rosetta3 was paramount to this operation: code written by one author to handle varied small-molecule druglike ligands was extended by a different author to allow generation of non-canonical residues incorporable into protein backbones, which was then used to allow modeling of a chemically conjugated fluorophore in a tethered docking simulation.

The generation of this cysteine-Mero53 parameter file allows Rosetta to understand the chemical connectivity of the residue and protein. Most of

AnchoredDesign's search protocol is designed to sample loop backbones, which will perform normally in the cysteine-mero53 case as the backbone itself and immediate neighbor atoms are unchanged. Rosetta's score functions are partially knowledge-based [19], which poses problems for novel residue types without known structures from which to parameterize the scoring terms [18]. Rotamer libraries used in packing are also knowledge-based, as they are drawn from samples of thousands of high-resolution crystal structures in the PDB database [20, 21].

To get around these problems, we used a Rosetta-based rotamer library generation protocol [18] to create rotamers for cysteine-Mero53, and made minor modifications directly to AnchoredDesign. Specifically, we excluded the Mero53-labeled position from minimization, ensuring that the Mero53 conformation will not stray from the predicted rotamers. With this alteration, AnchoredDesign can pack the Mero53 dye's 8 rotatable chi angles using a Rosetta-generated rotamer library and remodel the loop containing the Mero53 position normally. The only difference is that knowledge-based scorefunction terms will ignore the Mero53 position during scoring and minimization. The small scale of the modifications to AnchoredDesign to account for this new functionality is indicative of the wonderful reusability of Rosetta3 code: almost all of the code handling this non-canonical residue resides at lower levels in the codebase where it can be freely re-used.

Creation of Mero53-labeled models of 1F11-SH3

Once Rosetta was capable of handling cysteine-Mero53, we used the fixbb module to individually mutate 1F11 positions 24, 53, and 55 to cysteine-Mero53 in our starting model produced from unlabeled 1F11-SH3 models. Each of these models was fed back through the AnchoredDesign protocol to allow the protocol to test if the

presence of a large dye molecule would alter 1F11-SH3 binding, and determine the most likely binding orientation of the two domains in the labeled state. Furthermore, each complex was also modeled in an unbound state where all the same atoms are present, but a large (> 100 Å) gap exists between the two proteins. Here, the anchor is not held rigidly but is instead sampled as a normal loop. This unbound modeling allows the system to relax into the unbound state, allowing direct comparisons between bound and unbound models to elucidate the source of the fluorescence intensity change seen at position 24 but not at positions 53 and 55.

Results

The top-scoring dye-functionalized models for each dye position (24, 53, and 55) of the 1F11-SH3 complex are similar to each other and the original 1F11-SH3 model, as shown in Figure 3.5. This is consistent with the experimental result that dye functionalization does not have a large effect on 1F11-SH3 binding [1]. A perhaps surprising result of the modeling is that there is no contact between the Mero53 dye and the SH3 domain. Instead, Mero53 largely interacts with 1F11 or solvent, as seen in Figure 3.5.

To search for an explanation of why functionalization at position 24 allowed for dye intensity change upon binding, and functionalization at positions 53 and 55 did not, we compared bound and unbound models of the three positions. Qualitatively, we found that for top-scoring models of position 24, different loop conformations preferred in the bound and unbound state cause different packing interactions between Mero53 and 1F11 (Figure 3.6, panel A). Positions 53 and 55 are further from the interface on a loop that is

not predicted to directly interact with SH3 and do not show this effect (Figures 3.7 and 3.8, panels A and B).

While it was encouraging that Rosetta qualitatively detects a difference between position 24 and positions 53 and 55, a more physical explanation than altered packing was desired. Merocyanine dyes are known to be sensitive to their solvation environment, especially the polarity of the environment around their polar substituents [11]. The altered packing of Mero53 in the bound and unbound states at position 24 suggests that solvent exposure could be used to quantify the differences between models. We generated distributions of solvent-accessible surface area (SASA) of the entire Mero53 residue, fluorophore moiety, and each polar group (sulfone, sulfonate, and carbonyl). Figure 3.3 highlights each chemical group in the Mero53 dye.

Distributions of the whole Mero53 residue were uninterpretable due to noise from different packing of fluorescence-irrelevant linker atoms, and the carbonyl and sulfonate moieties were uninformative (data not shown). However, both the fluorophore and sulfone SASA distributions showed interesting differences between the bound and unbound state for position 24 (Figure 3.6, panels B and C), and no such differences for positions 53 or 55 (Figure 3.7 and 3.8, panels C and D). At position 24, the population of top scoring models has a bimodal sulfone SASA distribution centered at 90 and 120 $Å^2$. The bulk of the population shifts from the high-SASA to low-SASA state between the unbound and bound models. The discovery that modeled sulfone SASA anticipates fluorescence behavior is consistent with the theory that the sulfone group is most responsible for solvent sensitivity of this fluorophore [11]. For positions 53 and 55, small differences in fluorophore and sulfone SASA are seen (Figure 3.7 and 3.8, panels C and

D), consistent with the small fluorescence intensity differences seen at those positions [1].

These SASA distributions support a hypothesis that Mero53 sulfone SASA changes between the bound and unbound states of 1F11-SH3 complexes functionalized by Mero53 at position 24. Furthermore, these SASA differences are due to different packing between Mero53 and 1F11 due to loop conformational changes, and not direct Mero53-SH3 interactions. Other tested positions, 53 and 55, do not show any of these changes, consistent with their experimentally demonstrated lack of fluorescence response to binding.

Conclusion

This biosensor modeling showcases the utility of Rosetta3's flexibility, in that the AnchoredDesign algorithm was rapidly adapted to incorporate a novel fluorophore. Rosetta's non-canonical amino acid capabilities [18] were used to generate a parameter set and rotamer library for the chemically conjugated fluorophore. This fluorophore could then be dropped into Rosetta modeling with no modifications to the central packing, minimization, or loop closure algorithms, and only minimal modifications to AnchoredDesign itself.

AnchoredDesign was able to discriminate positions that showed a fluorescence intensity change on binding from those that did not. These models support a hypothesis that the Mero53 sulfone group differential burial is responsible for the altered fluorescence intensity in the bound state, consistent with dye photophysics [11]. Successes like these demonstrate the power of biology-modeling partnerships.

Figure 3.1: Src-family kinase autoinhibition
This diagram outlines the basic signaling states of a Src family kinase [5]. The diagram coloring follows the color spectrum from blue (N-terminal) to red (C-terminal). The SH3, SH2, and kinase domains (KD) are labeled, as are the internal PPII and variably phosphorylated tyrosine (pT; T) signaling regions. In the autoinhibited state (panel A), the N-terminal SH3 domain binds the internal polyproline II (PPII) linker sequence, and the SH2 domain binds a C-terminal phosphotyrosine motif (pT). These binding events prevent the kinase domain (KD) from adopting an active conformation. When not inhibited (panel B), the SH2 domain is freed by a lack of phosphorylation on the tyrosine, and the SH3 domain no longer binds the internal linker either. Generally, SH3 and SH2 may be binding other proteins with PPII or pT motifs of their own (not pictured). For the purposes of the Src biosensor presented here, the SH3 domain is available when the kinase domain is active, although competition with other cellular PPII sequences is possible.

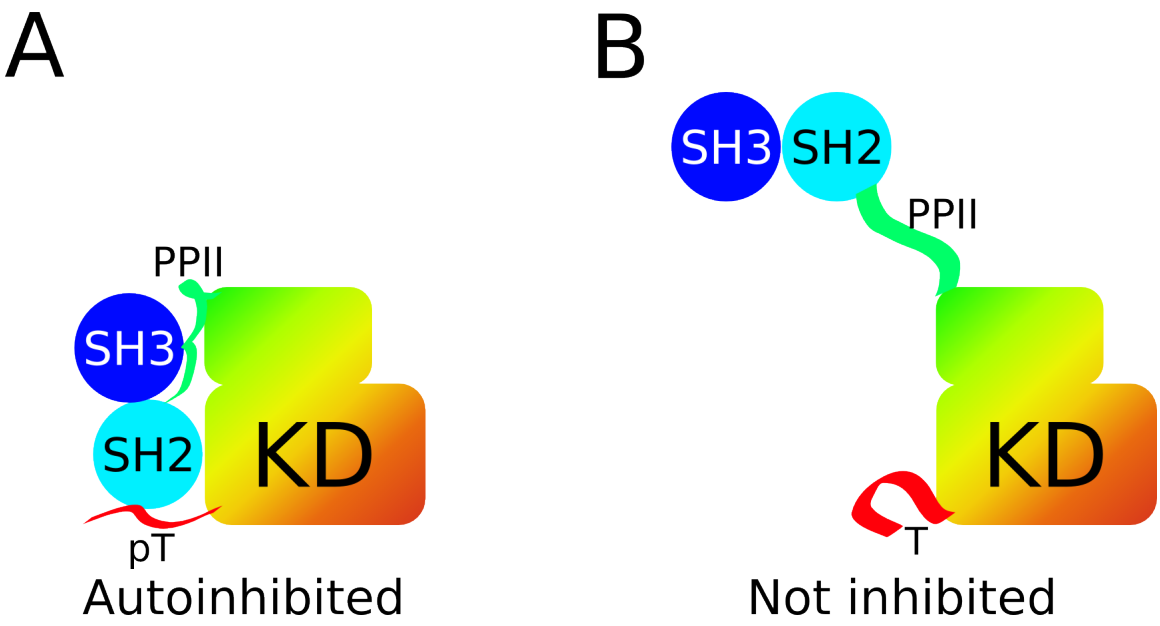


Figure 3.2: A fibronectin monobody bound to its target
This figure, rendered in PyMOL [22], was created from PDB 2OBG [10]. On the right, shown as an electrostatic potential surface (blue is positive, red is negative), is maltose binding protein (MBP). On the left, in green, purple, and yellow, is a phage-display-derived fibronectin monobody, MBP74, created to bind MBP. The figure also highlights the BC (purple) and FG (yellow) loops of the fibronectin fold, as these loops are often used for generating protein-protein interfaces [7-10].

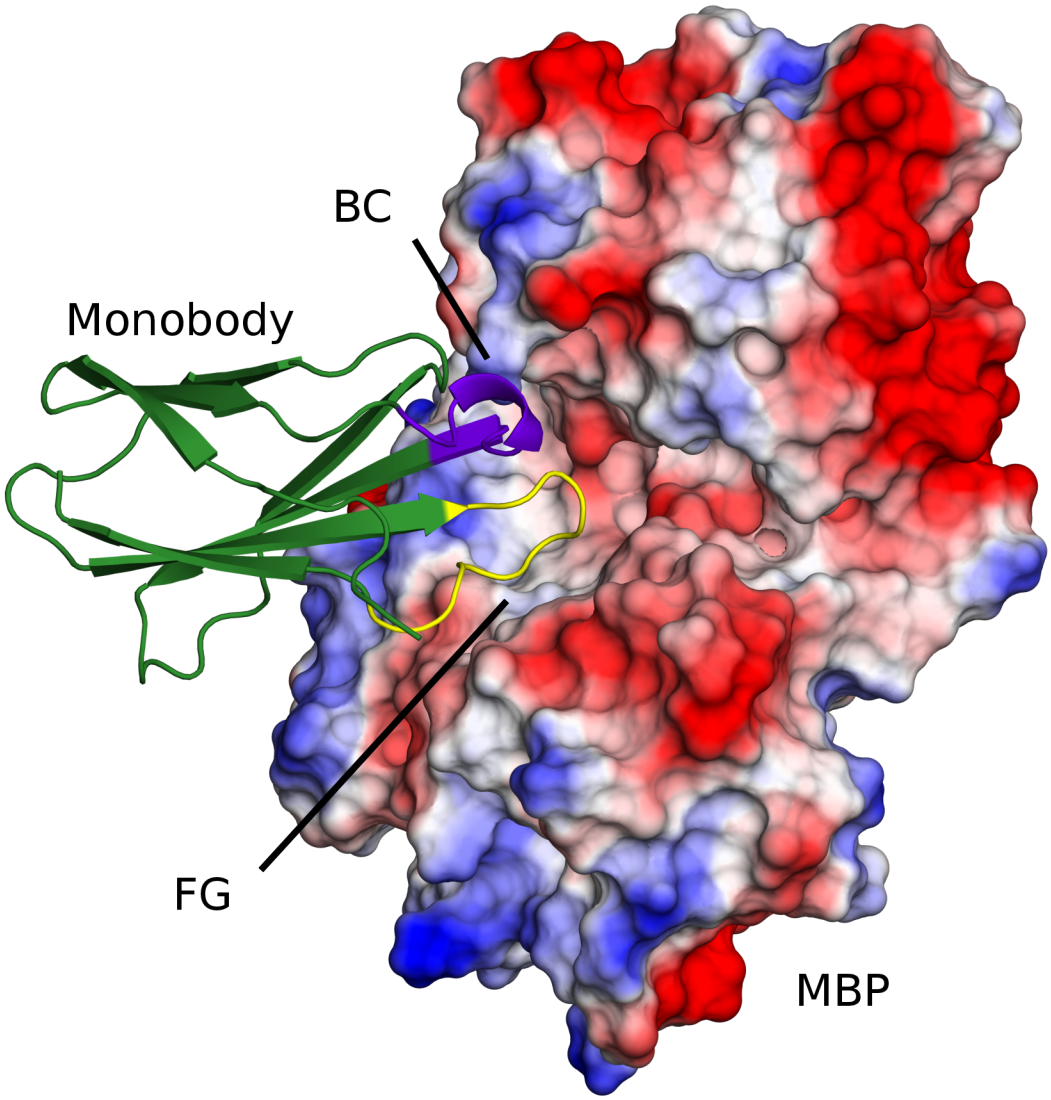


Figure 3.3: Mero53 dye structure and chemical moieties
This figure highlights the structure of the Mero53 merocyanine dye. The upper blue-circled moiety is the fluorophore, whereas the lower blue oval marks the normal cysteine residue to which the dye is attached. The green box represents the atoms in the entire residue for the new residue type created for Rosetta in this work. Polar groups in the fluorophore are also marked.

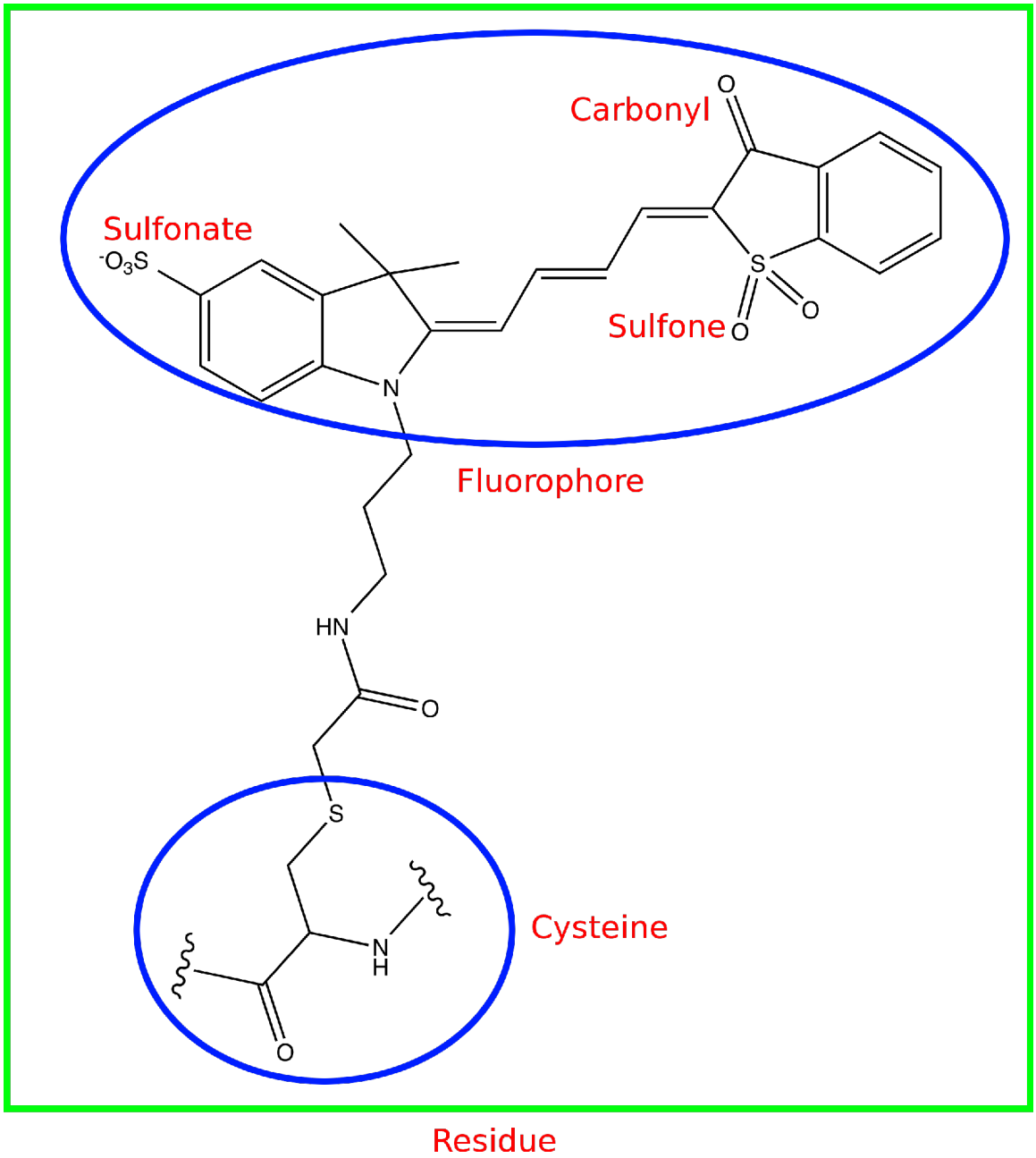


Figure 3.4: 1F11-SH3 complex model
This figure shows a representative 1F11 and Src SH3 model complex. In gray at the bottom is the SH3 domain. In yellow above is 1F11. The FG loop contains the RPLP anchor sequence. The BC loop and DE loop, both labeled, contain positions tested with Mero53 dyes in this work. The N-terminus of 1F11 (unlabeled) also makes contact with SH3 in this model in the background.

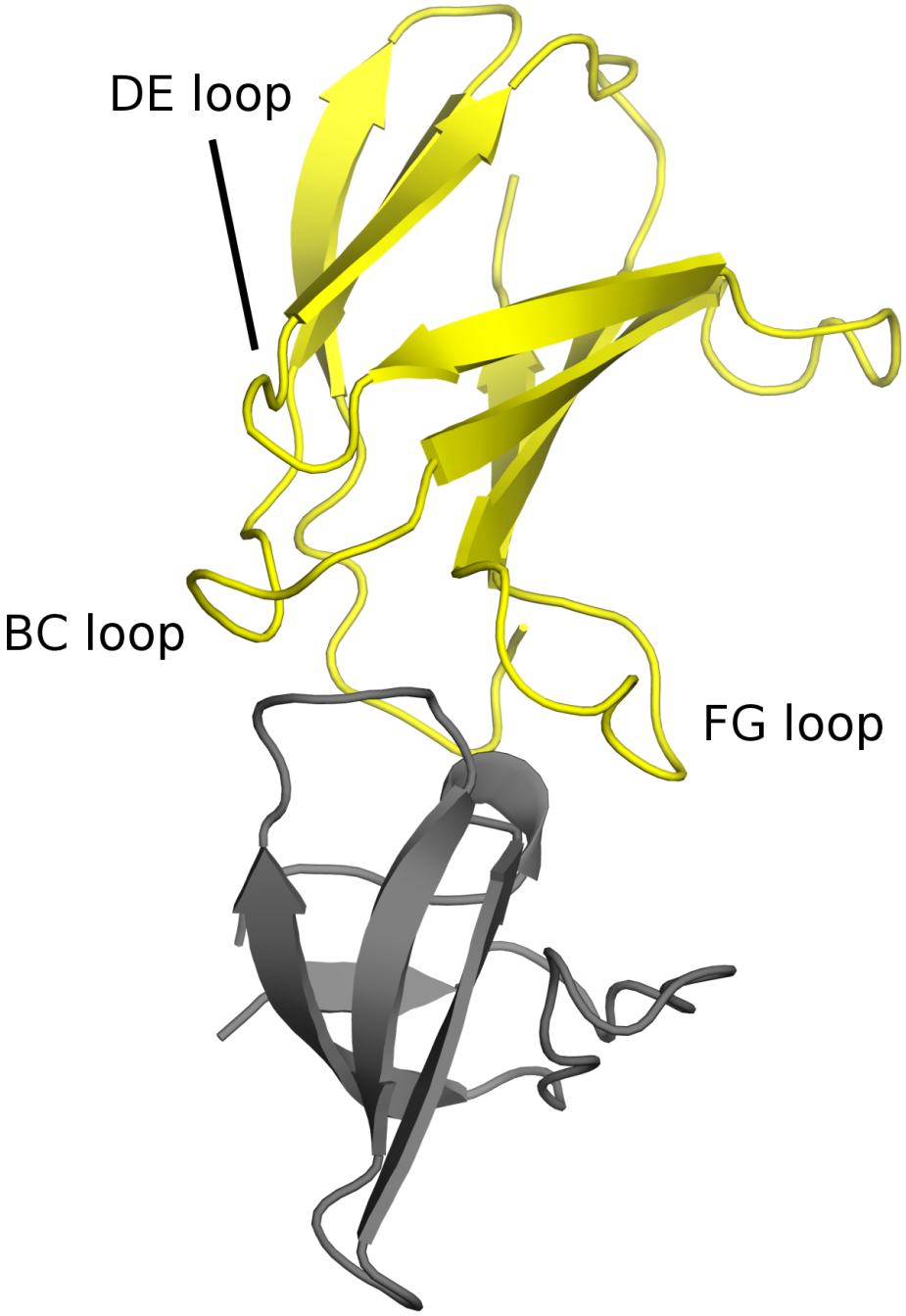


Figure 3.5: 1F11-SH3 models for the 3 dye positions
This figure shows four model 1F11 and Src SH3 complexes. As seen in Figure 3.4, an unfunctionalized 1F11 is shown in yellow and Src SH3 in gray. To the left is the region occupied by the Mero53 dye, which is roughly similar for the three positions. 1F11 functionalized at position 24 (BC loop) is in green, position 53 (DE loop) is in cyan, and position 55 (DE loop) is in magenta.

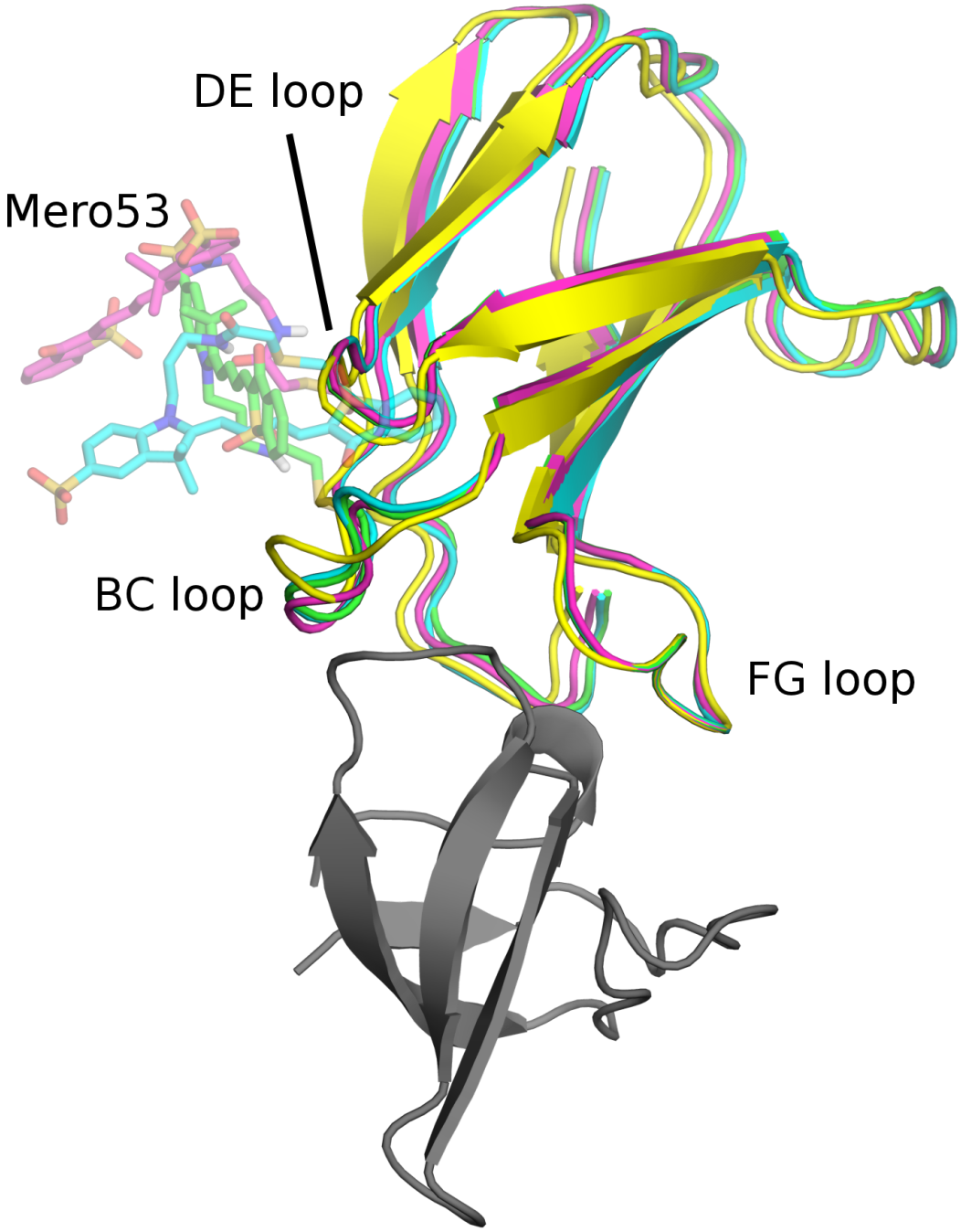


Figure 3.6: Modeling of the 1F11 dye–SH3 interface
(a) Computer models of either unbound 1F11-mero53 conjugate alone (i and ii) or 1F11-mero53 in complex with cSrc-SH3 (iii and iv). Dye is attached to residue 24, as in the final 'merobody' biosensor. c-Src SH3 is green, 1F11 is blue and dye is salmon. In (ii) and (iv) (magnified versions), the sulfone group of the dye is circled. The model of unbound biosensor shown here is part of the subpopulation in which the dye has higher solvent-accessible surface area (SASA). The model of bound biosensor is the highest-scoring model and a member of the low-SASA cluster. (b,c) SASA distribution for the top 0.5% of models of the bound and unbound states, either for the whole fluorophore (b) or the sulfone group (c).

This figure and legend are reproduced directly from Figure 6 of the source paper of this chapter [1].

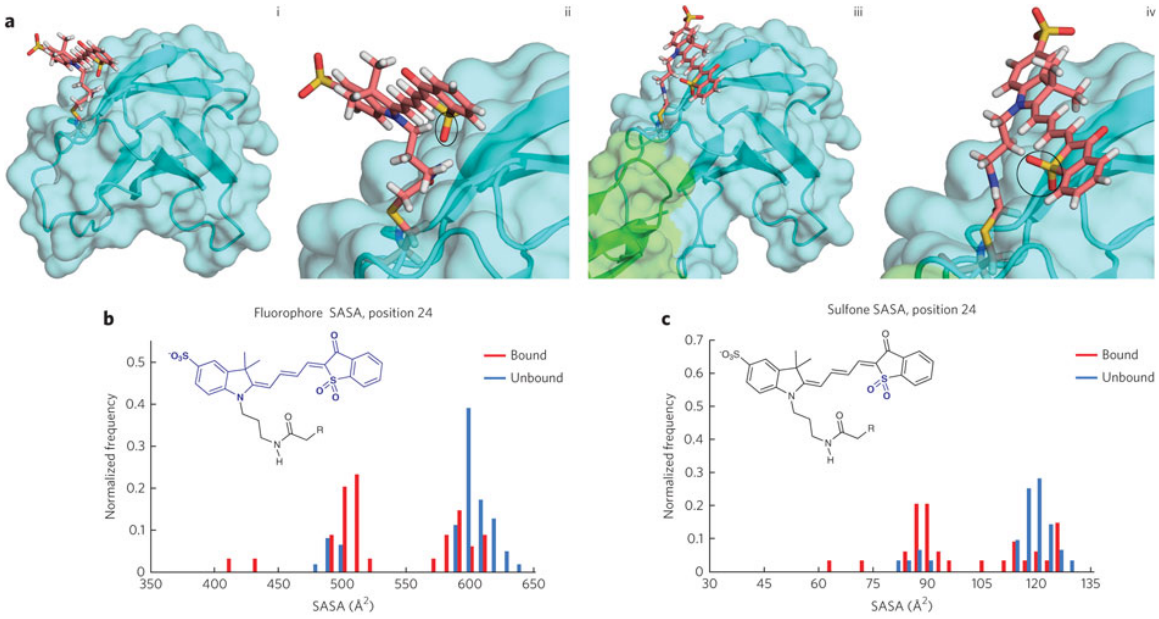


Figure 3.7: Modeling 1F11-SH3 functionalized at position 53
Panel A shows an unbound model of 1F11 (cyan cartoon and surface) functionalized at position 53 with Mero53 (salmon and CPK sticks). Panel B shows 1F11 (as before) bound to Src SH3 (green cartoon and surface). Notice that the dye does not change environment or conformation between A and B, although the loops on the 1F11 surface do. Panel C shows a SASA distribution histogram for the fluorophore moiety (refer also to Figure 3.3), with the bound and unbound states in red and blue respectively. Panel D shows the same distribution for the sulfone group. Both distributions show little change between bound and unbound state, especially compared to Figure 3.6.

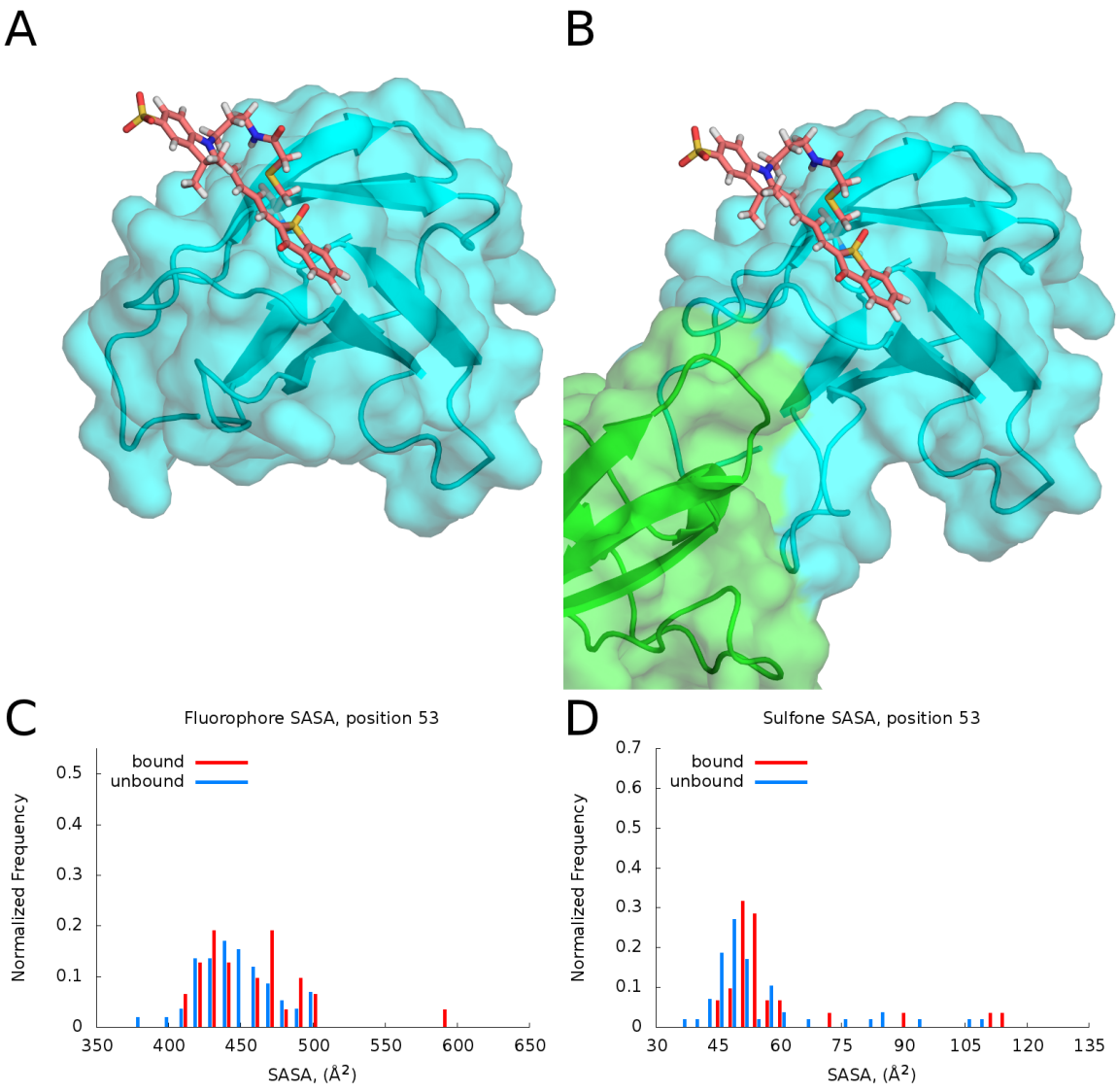
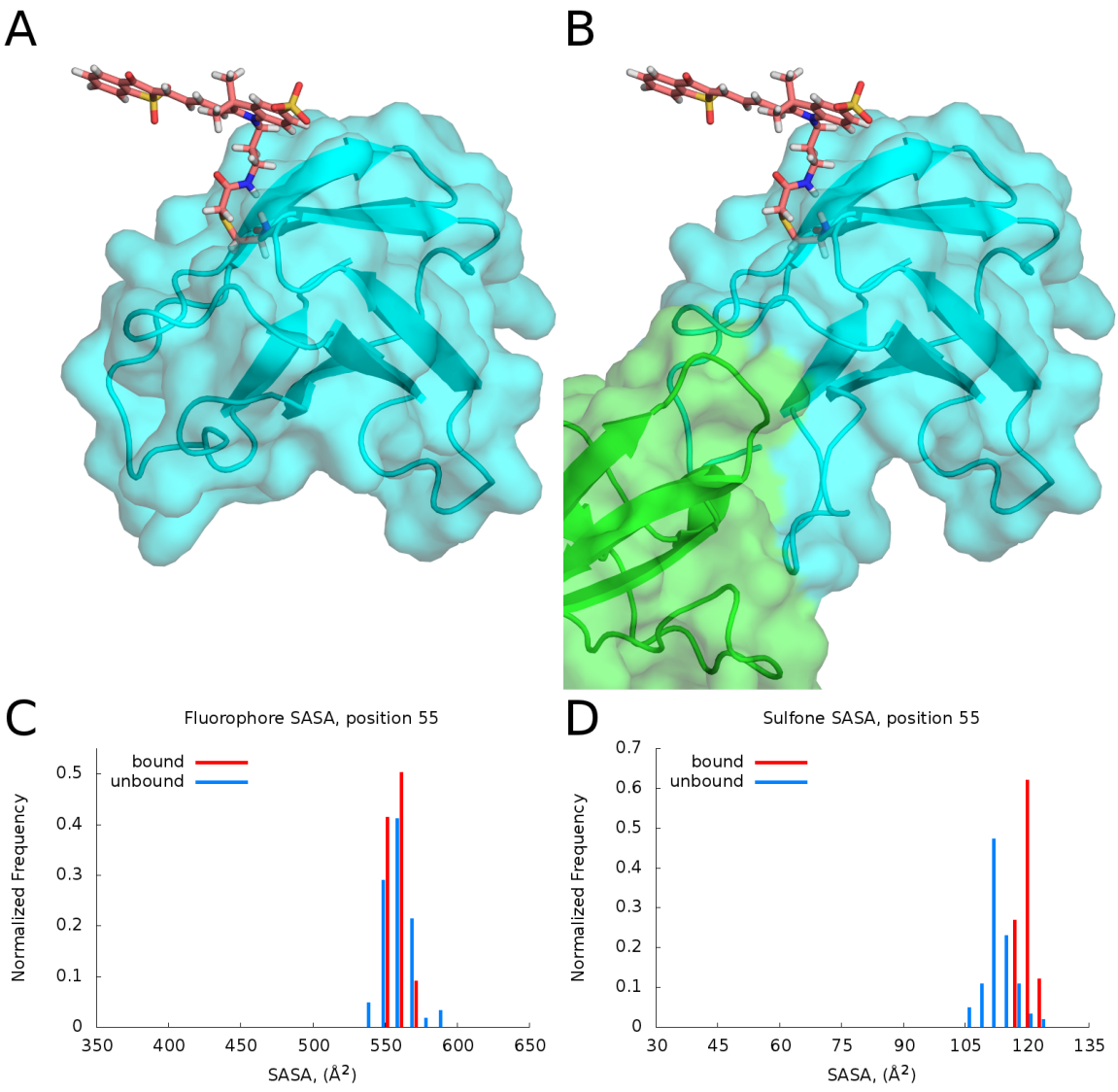


Figure 3.8: Modeling 1F11-SH3 functionalized at position 55
Panel A shows an unbound model of 1F11 (cyan cartoon and surface) functionalized at position 55 with Mero53 (salmon and CPK sticks). Panel B shows 1F11 (as before) bound to Src SH3 (green cartoon and surface). Notice that the dye does not change environment or conformation between A and B, although the loops on the 1F11 surface do. Panel C shows a SASA distribution histogram for the fluorophore moiety (refer also to Figure 3.3), with the bound and unbound states in red and blue respectively. Panel D shows the same distribution for the sulfone group. Both distributions show little change between bound and unbound state, especially compared to Figure 3.6.



References

1. Gulyani A, Vitriol E, Allen R, Wu J, Gremyachinskiy D, et al. (2011) A biosensor generated via high-throughput screening quantifies cell edge src dynamics. Nat Chem Biol 7(7): 437-444. 10.1038/nchembio.585; 10.1038/nchembio.585.

2. Giuliano KA, Post PL, Hahn KM, Taylor DL. (1995) Fluorescent protein biosensors: Measurement of molecular dynamics in living cells. Annu Rev Biophys Biomol Struct 24: 405-434. 10.1146/annurev.bb.24.060195.002201.

3. Thomas SM, Brugge JS. (1997) Cellular functions regulated by src family kinases. Annu Rev Cell Dev Biol 13: 513-609. 10.1146/annurev.cellbio.13.1.513.

4. Parsons SJ, Parsons JT. (2004) Src family kinases, key regulators of signal transduction. Oncogene 23(48): 7906-7909. 10.1038/sj.onc.1208160.

5. Boggon TJ, Eck MJ. (2004) Structure and regulation of src family kinases. Oncogene 23(48): 7918-7927. 10.1038/sj.onc.1208081.

6. Feng SB, Chen JK, Yu HT, Simon JA, Schreiber SL. (1994) 2 binding orientations for peptides to the src Sh3 domain - development of a general-model for Sh3-ligand interactions. Science 266(5188): 1241-1247.

7. Koide A, Bailey CW, Huang XL, Koide S. (1998) The fibronectin type III domain as a scaffold for novel binding proteins. J Mol Biol 284(4): 1141-1151.

8. Koide A, Abbatiello S, Rothgery L, Koide S. (2002) Probing protein conformational changes in living cells by using designer binding proteins: Application to the estrogen receptor. Proc Natl Acad Sci U S A 99(3): 1253-1258.

9. Karatan E, Merguerian M, Han Z, Scholle MD, Koide S, et al. (2004) Molecular recognition properties of FN3 monobodies that bind the src SH3 domain. Chem Biol 11(6): 835-844. 10.1016/j.chembiol.2004.04.009.

10. Koide A, Gilbreth RN, Esaki K, Tereshko V, Koide S. (2007) High-affinity single-domain binding proteins with a binary-code interface. Proc Natl Acad Sci U S A 104(16): 6632-6637.

11. Toutchkine A, Kraynov V, Hahn K. (2003) Solvent-sensitive dyes to report protein conformational changes in living cells. J Am Chem Soc 125(14): 4132-4145. Available: http://dx.doi.org/10.1021/ja0290882 via the Internet.

12. Leahy DJ, Aukhil I, Erickson HP. (1996) 2.0 angstrom crystal structure of a four-domain segment of human fibronectin encompassing the RGD loop and synergy region. Cell 84(1): 155-164.

13. Feng SB, Kasahara C, Rickles RJ, Schreiber SL. (1995) Specific interactions outside the proline-rich core of two classes of src homology 3 ligands. Proc Natl Acad Sci U S A 92(26): 12408-12415.

14. Canutescu AA, Dunbrack RL. (2003) Cyclic coordinate descent: A robotics algorithm for protein loop closure. Protein Sci 12(5): 963-972.

15. Kuhlman B, Dantas G, Ireton GC, Varani G, Stoddard BL, et al. (2003) Design of a novel globular protein fold with atomic-level accuracy. Science 302(5649): 1364-1368.

16. Leaver-Fay A, Tyka M, Lewis SM, Lange OF, Thompson J, et al. (2011) ROSETTA3: An object-oriented software suite for the simulation and design of macromolecules. Methods Enzymol 487: 545-574. 10.1016/B978-0-12-381270-4.00019-6.

17. Davis IW, Baker D. (2009) RosettaLigand docking with full ligand and receptor flexibility. J Mol Biol 385(2): 381-392. 10.1016/j.jmb.2008.11.010.

18. Renfrew PD, Choi EJ, Bonneau R, Kuhlman B,. (2012) Incorporation of noncanonical amino acids into rosetta and use in computational protein-peptide interface design. PLoS ONE 7(3): e32637. Available: http://dx.doi.org/10.1371%2Fjournal.pone.0032637 via the Internet.

19. Rohl CA, Strauss CEM, Misura KMS, Baker D. (2004) Protein structure prediction using rosetta. Methods Enzymol 383: 66-+.

20. Dunbrack RL. (2002) Rotamer libraries in the 21st century. Current Opinion in Structural Biology, 12(4): 431-440.

21. Berman HM, Westbrook J, Feng Z, Gilliland G, Bhat TN, et al. (2000) The protein data bank. Nucleic Acids Res 28(1): 235-242.

22. Schrödinger L. (2002) The PyMOL molecular graphics system. 1.4.1. Available: http://www.pymol.org; via the Internet.

Chapter 4

New interfaces created with AnchoredDesign

Introduction

The purpose of the AnchoredDesign protocol introduced in Chapter 2 [1] and expanded upon in Chapter 3 [2] is for the design of protein-protein interfaces. Published work includes a computational benchmark which indicates it should be proficient at interface design [1] and a modeling experiment in a fixed sequence context [2]. This chapter details a protein-protein interface design experiment with AnchoredDesign.

Design scaffold

The design scaffold used for this experiment is the monobody scaffold, human fibronectin, domain type 3, repeat 10 (10FNIII) [3]. This scaffold was extensively introduced in Chapter 3. Briefly, it is useful for protein interface engineering because it contains multiple flexible surface loops which are broadly permissive of mutations. This gives great freedom for rational and computational design, as well as evolutionary techniques, to find 10FNIII loop sequences which result in conformations that bind desired targets. In this study, we will be using the FG and BC loops for flexible-backbone interface design.

Selection of Keap1 as the design target

Keap1 was chosen as the target for this interface design experiment. Keap1 is a negative regulator of the Nrf2 transcription factor [4, 5]. Nrf2 upregulates the expression of cytoprotective proteins when the cell is under oxidative chemical stress. Keap1

prevents Nrf2 activity when the cell is in a healthy oxidative state. When the cell experiences oxidative stress, reactive sensor cysteines in Keap1 become chemically modified and prevent Keap1 repression of Nrf2, thus upregulating Nrf2's cytoprotective targets. The precise details of this regulation are not known, but it appears Keap1 modulates the lifetime of Nrf2 through interactions with ubiquitin E3 ligases, possibly by directing Nrf2 to the proteasome via polyubiquitination [5].

The C-terminal β-propeller domain of Keap1 is the domain of interest for this work, and future references to Keap1 in this work refer to this domain in isolation. This domain binds an ETGE motif present in Nrf2 [6]. Crystal structures indicate that the ETGE motif adopts a hairpin structure which inserts into the pocket in the center of the β-propeller for both human [7] and murine [8] forms of the protein. The human protein structure used in these experiments, PDB 2FLU [7], can be seen in Figure 4.1. This hairpin buries itself quite deeply into the β-propeller pocket, which led to its selection as an AnchoredDesign target.

The Nrf2 ETGE motif, whose immediate sequence neighbors expand to DEETGE, is a highly negatively charged sequence. Accordingly, the binding site on Keap1 is positively charged, as seen in Figure 4.1. The Keap1 β-propeller, like all Kelch β-propellers, is a repeat protein of six units with four β-strands in each unit [9]. The positive charge of Keap1's β-propeller DEETGE-binding interface is due to a conserved arginine residue that occurs in each blade.

This charged-peptide binding format is shared with other binding targets of Keap1. A different Nrf2 sequence, the DLG motif, contains two nearby aspartates that bind in a similar fashion [10]. Keap1 has also been found to bind a peptide from

prothymosin α containing a nEEnGE sequence (echoing DEETGE) with a highly similar charged ligand structure [11]. Finally, a structure of a DpsTGE peptide from sequestosome-1/p62 also adopts an equally similar hairpin structure [12]. The wide spread of this charged-hairpin binding pattern indicates that the Keap1 interface is ripe for binding other charged-hairpin-containing sequences and is thus a good target for AnchoredDesign.

Display, selection, and screening strategies

Past successes in computational protein interface design have benefited from use of post-design display techniques to either accelerate testing of many computational designs at once, use evolutionary strategies to enhance affinity of binders, or both [13, 14]. It has also been demonstrated that computational design techniques can be used to bias the selection of residues in display libraries to improve the resulting binders and enrich the library with useful hits [15-17]. For this reason, instead of selecting single sequences to test experimentally, we converted AnchoredDesign's predicted sequences directly into a computationally-designed library of sequences. We then used standard selection schemes to find the best binders in this directed library and experimentally validated their binding.

A computationally-designed population of proposed interfaces is composed of distinct sequences, and each position within that sequence was designed in the context of all the other positions in the sequence. The population therefore contains information not just in the amino acid propensities at each position, but also in correlated mutations that occur at different positions. A library that naïvely compiles sequence propensities will discard this correlation information and combinatorially include many undesigned

possibilities, including even poor-scoring sequences rejected by computational design. Lippow et al. tested this idea by generating a library in four parts, maintaining computationally-directed correlation within each part [18]. However, they recognized that the difficulty of constructing DNA libraries that properly remember correlation even at sequentially distant positions is challenging and potentially expensive.

We use a similar library design strategy in this study that exploits the sequence proximity of mutations within a loop to maintain AnchoredDesign's predicted mutation correlations in the display library. Our focus on 10FNIII's BC and FG loops means that all mutations occur within two small, highly sequentially local groups of residues. Oligonucleotides long enough to cover a single loop and thus maintain the designed correlations within that loop are inexpensive. Maintenance of correlation between loops is still problematic and not addressed directly by this study.

Methods

Selection of the anchor for AnchoredDesign

The Keap1-Nrf2 peptide interface was identified by the AnchorFinder algorithm as a possible AnchoredDesign target during a wide-scale search for AnchoredDesign compatible interfaces for its published computational benchmark [1]. The system is inappropriate for benchmarking, as the crystal structures show the ETGE hairpin binding as a peptide instead of as a loop of a folded protein [7, 8], leaving AnchoredDesign with no protein-protein interface to predict. However, this hairpin is ideal for use as an anchor by AnchoredDesign in generating new interfaces because the hairpin buries deeply into its target, is made of residues that are immediate sequence neighbors, and has geometrically proximal endpoints so that it will insert neatly into a 10FNIII loop. Most

of the AnchorFinder hits used for benchmarking [1] were homodimers and thus inappropriate for interface design, as the unmodified target would make itself unavailable for binding; the remainder were unusable for design purposes for other reasons.

After identifying Keap1 as a good target, the many peptide-Keap1 structures were considered individually as source structures for modeling. The 1.5 Å resolution crystal structure (PDB 2FLU) of the human Keap1-ETGE interaction [7] was chosen over the murine Keap1-ETGE interaction [8] because it is of more-medically-relevant human protein, over the structure of the murine Keap1-DLG motif [10] for the same reason and because ETGE binding is tighter [10, 19], and over and the Keap1-prothymosin α [11] structure because much more biological data exist for the Keap1-ETGE (Nrf2) interaction.

Choosing 2FLU as the source structure for modeling leaves open the question of precisely what residues to use as the anchor sequence for AnchoredDesign. The ETGE-containing peptide in structure 2FLU is 16 residues long: AFFAQLQLDE<u>ETGE</u>FL. Figure 4.1 demonstrates that the N-terminal half of the peptide (before aspartate) does not interact with Keap1 in the crystal structure [7], although those residues do influence in vitro binding affinity [10, 19, 20]. The subsequent DEETGE forms a symmetric hairpin which accounts for almost all peptide-Keap1 interactions. Both ETGE and DEETGE were tested as anchors for AnchoredDesign.

As was described in Chapter 2, AnchoredDesign searches conformational and sequence space for a protein-protein interaction, but it cannot search through possible loop lengths and anchor placements within those loops [1]. The 10FNIII domain has two neighboring loops, FG and BC, useful for this sort of protein engineering [3]. However,

AnchoredDesign is not restricted to using the original lengths of those loops, nor is there an *a priori* reason to prefer one anchor location within the loops to another. For this reason, we searched many possible loop lengths and anchor placements, as shown in Table 4.1.

Creation of starting structures for AnchoredDesign

Construction of starting structures for AnchoredDesign is performed with the companion application AnchoredPDBCreator [1]. The function of this code is shown schematically in Figure 2.1. Briefly, AnchoredPDBCreator replaces a gap between loop residues in the scaffold (10FNIII) with the anchor (the DEETGE or ETGE peptide), closes the insert-accepting loop without modifying the conformation of the anchor, and then aligns the anchor to its binding location against the target (Keap1 β-propeller).

AnchoredPDBCreator performs best when the scaffold residues that will be replaced are not present in its input structures. To try different loop lengths and anchor placements (Table 4.1) a variety of 10FNIII structures with gaps in different places in their loops were prepared. Deletion of residues to accommodate the anchor is straightforward: the PDB file can be manually edited in a text editor to delete the unneeded residues and used immediately for AnchoredPDBCreator. Addition of residues is best handled with Rosetta3's loop modeling executable, as is described in the demo documentation for AnchoredDesign released with Rosetta3.3 and newer releases. Briefly, new residues are created in a text edtior by copying an existing residue at the insertion point, changing the numbering to force Rosetta to treat the copy as a different residue despite identical coordinates, followed by using of the -build_initial flag to the loop modeling executable to create new loop coordinates from scratch. Additionally, the size

of the gap prepared for AnchoredPDBCreator need not equal the size of the insert: replacing a 2-residue deletion with a 6-residue anchor leads to a loop that is four residues longer. AnchoredPDBCreator does not strictly require a gap if the entire anchor is desired as an insert rather than a replacement for the existing loop residues.

AnchoredPDBCreator closes the anchor-containing loop via Rosetta's implementation of Cyclic Coordinate Descent [1, 21, 22], and thus requires a moderate amount of Monte Carlo sampling to assure a good solution. Between 2000 and 3500 AnchoredPDBCreator results were generated for each computational construct reported here. Generally, the most important quality of these results is the bond lengths and angles at each edge of the anchor. These degrees of freedom are not generally sampled by AnchoredDesign, and errors may not be repaired later. Clashes and poor torsions are not particularly important because they will be relaxed out by the primary AnchoredDesign protocol. There is a code module, LoopAnalyzer, which runs after AnchoredPDBCreator (and AnchoredDesign) and reports these qualities into the output PDBs for later analysis and filtering of the best results [1].

Designing 10FNIII monobodies targeting the Keap1 Kelch domain

AnchoredDesign was used to create and refine models of possible 10FNIII-Keap1 interactions. Modeling proceeded using the protocol reported for AnchoredDesign benchmarking [1]. The addition of design into the protocol requires only a small change: the use of a resfile to specify which positions can mutate replaces the design-precluding command line flag -packing:repack_only used in benchmarking. The modeling proceeded in three stages. First, a very few preliminary models were created to search

for errors in the inputs and verify that the loop, anchor, and resfile specifications were correct; these models were not used further.

Next, an initial batch of models for each loop length and anchor placement in Table 4.1 was created. Each of these experiments produced approximately 1000-1500 models of varying sequences and conformations, which collectively required 192 processor-days on 2.33 GHz processors. Figure 4.2 shows the command line options used for this experiment. These models were of widely varying quality by the three metric sets used in AnchoredDesign (total score, loop quality, and interface quality) [1], so further experiments were desired.

The best-scoring and structurally interesting or distinct models from the first round of models were used as inputs for another round of modeling. This modeling used a slightly different set of options (Figure 4.3) as in the earlier round: the perturbation phase was skipped entirely to instead focus on refinement of the already-reasonable interface models. These models were also produced with fewer rounds of refinement per model. Each of 74 starting models, distributed fairly evenly across 8 of the 9 constructs in Table 4.1, were used to seed independent collections of approximately 700-1200 models from each starting model. This required 42 processor-days on 2.33 GHz processors per group. This round of modeling resulted in a collection of about 73,000 10FNIII-Keap1 models distributed in 74 structure/sequence clusters from which to select models to test with bench experiments.

Design of the directed library

To convert our computational sequence population into a display library, we first focused on variation in the anchor loop. The anchor loop is largely fixed in sequence and

conformation because of the anchor, and thus contains less variation than the non-anchor loop. We searched for anchor loop sequences which were recurrent in the 73,000 member population and particularly for those that were attached to structures that scored well by the overall score, interface, and loop metrics [1]. After selecting the best anchor loop sequence, we examined the subpopulation of models with that sequence to determine the best and most recurrent sequences in the other loop, using the same metrics. The library was then constructed to capture the computational variation in the second loop, with the anchor loop held to a single designed sequence. The library itself is detailed in the Results.

This two-phase selection offers a variety of benefits. First, the short primary sequence length of the second loop ensures that all computational correlated mutations can be maintained in the library, as a one-oligo-per-designed-loop strategy is not unduly expensive. Second, the focus on a single anchor loop obviates the need to consider maintenance of cross-loop correlations, as all designs and all library sequences contain a fixed anchor loop sequence. This strategy allows for use of the library as both a screening method to rapidly select the best binder from many computational designs, and simultaneously prepares for rational, noncomputational expansion of the library to increase diversity to improve the chance of finding a good binder, or reduce cost by allowing more degenerate oligonucleotides.

Construction of the library

Dr. Gurkan Guntas selected oligonucleotides, assembled them into the directed design library, and performed all selection experiments. Experimental details will be published with a forthcoming paper.

We constructed a random library as a companion to the design library which used the designed anchor loop sequence, but random residues for a seven-residue portion of the BC loop. This library had a theoretical complexity of $20^7 = 1.28x10^9$, which was not covered completely by the actual library.

Protein expression and purification

Expression and purification conditions were designed by Dr. Gurkan Guntas, and full procedures will be reported in a forthcoming paper. The expression and purification work was performed by this author, by Dr. Guntas, or by Thomas Lane.

A plasmid for Keap1's β-propeller domain was generously provided by the Hannink lab. Keap1 was expressed using conditions similar to those published [23], except that after induction, the incubation temperature was lowered to 16-20 °C and cells were harvested after expression overnight (approximately 16 hours). 10FNIII monobody sequences were expressed under identical conditions.

Keap1 was expressed with a 6xHis tag. It was purified using methods similar to those used in earlier purifications of the protein [23] and also similar to purifications reported in this lab [24]. The three-step purification uses a Ni-NTA nickel affinity column (HisTrap, HP, GE Healthcare), followed by an anion exchange column (Source 15Q, GE Healthcare), followed by a polishing size exclusion step (Superdex 75, GE Healthcare). The 6xHix tag was not removed.

Fibronectin monobodies were expressed with a glutathione-S-transferase tag, and thus were initially purified with a glutathione affinity column. After incubation (generally 40-64 hours) with thrombin protease to cleave the tag, purification was

completed via size exclusion (Superdex 75, GE Healthcare). All designs tested purified in sufficient quantity for ITC, FP, or crystallographic analysis from 1.5 L cultures.

Analysis of protein-protein binding by isothermal titration calorimetry

Isothermal titration calorimetry (ITC) was used to measure the binding affinity of monobody-Keap1 interactions. Some experiments were performed by Dr. Gurkan Guntas or Thomas Lane, and all experiments were performed with the assistance of Dr. Ashutosh Tripathy. A Microcal AutoITC200 calorimeter (GE) was used for all experiments. Keap1 and various monobody sequences were dialyzed into identical buffers containing 20 mM KH_2PO_4, 150 mM NaCl, and 5 mM 2-mercaptoethanol at pH 7. Most experiments were performed with 20 injections of 100-200 μM monobody into 10-20 μM Keap1. The data were analyzed for K_d and other thermodynamic parameters by fitting to a one-site binding model with Origin 7.0 (OriginLab Corporation).

Analysis of protein-protein binding by fluorescence polarization

Fluorescence polarization was used in two ways. First, it was used to make a repeat measurement of the binding affinity (as K_d) between Keap1 and an Nrf2-derived ETGE-containing fluorophore-functionalized peptide, LDEETGEFL-K-(5FAM) (Dr. Krzysztof Krajewski). With this binding affinity in hand, we then used fluorescence polarization to measure competition between the labeled peptide and unlabeled monobody sequences. The presence of the DEETGE sequence in the two competitors implies both are binding to the crystallographically indicated binding site. Some sequences tested were hits from the display or random library, and some were point mutants designed to probe the validity of the design models.

Experiments were performed on a Jobin Yvon Horiba Spex FluoroLog-3 (Jobin Yvon Inc.). Fluorescent excitation and emission were set at the peaks for the 5FAM fluorophore, 494 and 521 nm respectively. Experiments were carried out in a 4.5 mL, 1 cm path length cuvette. The 5FAM-labeled peptide is the only fluorescent moiety in the system of significance at these wavelengths. Peptide concentrations sufficient for $>10^5$ photon counts per second (20 nM or greater) were used in all experiments. Standardized settings included 1 s integration times, 6 measurement replicates per data point, and 6 nm monochromator widths.

Peptide-Keap1 binding was analyzed using the same equations and techniques reported previously in this lab [24].

Monobody-Keap1 binding was measured via a competition assay which measures polarization decrease as monobody binding frees peptide from Keap1, increasing its tumbling rate. Generally, 20 nM peptide and 350 nM Keap1 were in the starting solution; this represents approximately 80% bound peptide as determined in the earlier peptide-Keap1 experiments. Monobody was then titrated into the cuvette, and polarization measured after each addition. These data were fit for IC_{50} using a procedure reported by Lungu et al. [25], and refined into K_i with an Internet-based calculator from Nikolovska-Coleska et al. [26, 27]. Certain experiments violate the assumptions of Lungu et al., and fits were generated from an iterative numeric fitting program adapted from Purbeck et al. [28].

Attempts at crystallization

To verify the design models of successful binders, we attempted to crystallize multiple Keap1-monobody complexes. In each case, Keap1 and monobody were mixed

in a 1:2 ratio (at 100-1000 μM concentrations) and transferred into a buffer consisting of 100 mM ammonium acetate and 5 mM 2-mercaptoethanol at pH 7. Excess monobody was purified away by running the mixture through a Superdex 75 column (GE Healthcare). In each case, the mixture eluted as one peak corresponding to a heterodimer molecular weight and one corresponding to pure monobody. These samples were screened against the JCSG I-IV core suite (Qiagen) at 25 °C and the JCSG+ suite (Qiagen) at 4, 12, and 25 °C. Protein concentrations of 3-20 mg/mL were tested. No crystals were obtained.

Results

Library construction

We produced 73,000 protein structures in the final round of modeling, and approximately 16,000 in the round preceding that. The earlier round of modeling demonstrated that not all anchor and loop combinations were compatible with the intended binding. In particular, design setup 6.77.84 (refer to Table 4.1) had consistently poor scores in early modeling and was discarded from later consideration.

To select sequences for use in a library, we first sorted all models by the sequence of their anchor loop, and searched for an anchor loop sequence that scored well by total, loop, and interface energies [1]. We also wanted an anchor loop sequence that was highly repeated in the dataset, both to indicate design convergence and provide enough sequences to make a library. The selected anchor loop sequence was yavRDEETGEFHWPis, which occurs in the FG loop of 10FNIII. Lowercase indicates constant 10FNIII sequence for framing, and the underline indicates the anchor. The post-anchor phenylalanine residue happens to be the Nrf2 residue immediately following the

ETGE sequence as well; this represents Rosetta's convergence onto biology but was indeed a designed position. The final proline residue is native to the scaffold, but again was allowed to design. For reference, this anchor sequence is compatible with label 6_77.83 (any BC loop length) in Table 4.1.

After selecting the computationally preferred anchor sequence, we compiled a list of all BC loop sequences associated with that FG anchor loop. As in choosing the anchor loop sequence, we searched for BC loop sequences in structures that scored well by the AnchoredDesign metrics [1] and recurred in the data set. 71 BC loop sequences were chosen as occurring in structures that scored well by total energy or binding energy, and an additional 9 were chosen as a convenience sample of members of the general model population containing the correct FG anchor loop. Each of these sequences belonged to the BC plus 1 6_77.83 construct. Other BC loop lengths (plus 0 or 2) were considered, but did not score sufficiently well. In general, the sequences contained many acidic residues, probably because of the highly basic nature of the Keap1 binding pocket targeted by these monobodies [9]. These 80 sequences were covered by a set of 28 oligonucleotides, some degenerate, as shown in Table 4.2. The degeneracy introduced a total of 587 sequences into the library over the original 80. This represents a reduction in cost, as 28 oligonucleotides is far fewer than 80 and 587 BC loop sequences is many orders of magnitude smaller than typical phage display library sizes [17] and can be sampled completely.

Library screening

The library was screened with 8 rounds of panning against Keap1. Experiments were performed by Dr. Gurkan Guntas and will be fully detailed in a future publication.

Clones from the library were extracted after the 5th and 8th rounds of panning for individual testing; this is indicated by a prefix 5_ or 8_ in the sequence name. The random library was selected in a similar manner. Selected sequences, along with Keap1 binding affinities, are reported in Table 4.3.

Binding analysis of selected library sequences

Previous publications have measured the binding affinity for the Nrf2-derived LDEETGEFL 9mer to be 352 nM by surface plasmon resonance [20] and 182 nM by isothermal titration calorimetry [19]. We measured the affinity of LDEETGEFL-K-5FAM by fluorescence polarization to be 141 nM. This 9mer peptide (ignoring the fluorophore linker lysine) is the shortest Nrf2-derived peptide with strong binding to Keap1 [20], so we considered this binding to be a useful benchmark with which to gauge our selected sequences.

Two monobody-based controls also serve as benchmarks for binding. One is the native 10FNIII sequence, which did not detectably bind Keap1 in an ITC experiment (Table 4.3). Another is a 10FNIII sequence with the designed anchor FG loop, but an unmodified wild-type BC loop. This sequence, wtBC, represents the affinity offered by only the designed anchor loop. The binding affinity of this was found to be 1.5 μM by ITC. It should be noted that this wtBC sequence, while useful as a comparison, is not a purely rational design. The insertion of the 6-residue DEETGE anchor into the FG loop represents noncomputational rational design, but the surrounding sequence is computationally designed by Rosetta, so some of the increase in binding between the wild-type 10FNIII sequence and the wtBC sequence may be due to the effects of computational design. Rosetta's recovery of the wild-type phenylalanine C-terminal to

the anchor DEETGE sequence represents either computational design, or recovery of wild-type preferences in the system. These nonanchor, but anchor loop, positions are highlighted in Figure 4.4, panel A.

We individually measured the binding of several sequences drawn from the directed library after 5 and 8 rounds of panning against Keap1, as well as sequences drawn from the random library after panning against Keap1. These binding experiments are summarized in Table 4.3. For sequences which directly map to models from the second round, the best-scoring models are presented in Figure 4.4. Many hits, including specifically designed sequences and sequences with library-derived mutations, showed binding with K_d in the hundreds of nanomolar range, consistent with the peptide binding. Hits from the random library tens of nM to sub-nM K_ds, considerably better than the biological peptide from which the anchor derives [19, 20].

Structural characterization of monobody-Keap1 interfaces

To assess the quality of the design models, we attempted to obtain structural data for the monobody-Keap1 complexes. Crystallization trials repeatedly failed. We instead turned to mutagenesis of putative interface residues to determine the effects of mutation on binding. We used AnchoredDesign to predict binding affinities for a series of alanine mutations to sequences 5_2 and 8_1, as shown in Table 4.4. Rosetta predicted binding affinity deficits for all alanine mutations tested. However, FP determining of binding affinity shows that for all mutations, the alanine mutation was of a similar binding affinity to the original sequence. This is inconsistent with the design structure for these sequences, as seen in Figure 4.5. The most parsimonious explanation is that the design models are incorrect and BC loop binding occurs in a fashion not anticipated by the

design structures. The designed BC loop sequences do have an effect on binding, because the library-selected sequences bind more tightly than a control with a wild-type BC loop, as seen in Table 4.3.

Discussion

Partial success of selected designs

Mutation of designed BC loop residues, predicted by AnchoredDesign to be important for binding, indicated that the design models were unlikely to accurately depict the conformation of the 10FNIII BC loop (Table 4.4 and Figure 4.5). Nevertheless, the experimental system described here produced multiple sub-micromolar binders to the Keap1 target with a variety of BC loop sequences selected from the design and random libraries, even though a control experiment with the wtBC sequence indicated 1.5 μM binding for that unmodified BC loop. Hits from the random library, which includes a computationally-designed anchor-containing FG loop, bound tighter than the 9mer minimal Nrf2 peptide that the anchor loop mimics. The designed FG loop positions surrounding the anchor may be important for supporting the anchor sequence in its Keap1-binding hairpin conformation, as is true for residues immediately surrounding the ETGE sequence when presented as a peptide [20, 29].

Keap1's native partner Nrf2 binds the ETGE sequence with a K_d of 5 nM [19], and the isolated peptide binds with affinities in the tens to hundreds of nM range, depending on the number of residues present [20]. Our best monobody, R1, binds better than Nrf2 with single or sub-nanomolar affinity (Table 4.3), making it the tightest Keap1 β-propeller binder known. Overall, we have succeeded at creating tight binders to Keap1

through combination of a native partner's sequence, computational design, and display techniques.

Our results indicate that either the search or score functions used by AnchoredDesign are not quite sufficient for the enormous task of flexible-backbone protein-protein interface design. Part of the problem may have been the choice of a highly charged anchor loop and consequently a highly charged and polar target interface. As seen in Table 4.2, many sequences in the design library contain many negative charges in the BC loop, on top of the fixed four negative charges in the anchor and multiple positive charges on the target surface (Figure 4.1). This large collection of charged residues poses significant challenges for Rosetta's score function, which does not include a directly physical treatment of charge-charged interaction [30, 31], instead using a coarse-grained knowledge-based residue pair term. There is some evidence that this treatment is not precise enough for the design of highly polar interfaces like those used in this study [32-34]. It also seems likely that the AnchoredDesign algorithm is simply sampling insufficiently, given the tremendous number of degrees of freedom in the system. Despite the incomplete success of the AnchoredDesign algorithm, its complexity should not be underestimated. The use and reuse of other Rosetta3 code modules via the `Mover` interface was crucial to the development of this code [35].

A major advantage of our method over display technique alone is that we have directed the formation of the interface to a particular region of the target. Pure display techniques offer no control over where and how a new binder will attach to the target protein [17], but our method has produced binders that bind along the intended interface. The existence of binding competition between the Nrf2-derived peptide and our

monobodies, as measured by the decrease in fluorescence polarization as monobody competes off peptide for Keap1 binding, indicates that the peptide and monobody are binding in the same place. As this location on the Keap1 β-propeller is known [7], we can be reasonably confident that our monobodies are at least binding the correct face of Keap1, if not in the designed orientation.

The monobodies in this paper are not the only reported experimentally-generated reagents that tightly bind Keap1. A recently published study by Hancock et al. [29] revealed rational and display-derived peptides based on Nrf2 ETGE motif and the sequestosome-1/p62 motif [12] that tightly bind Keap1. The best monobody sequences reported here are two orders of magnitude tighter than those reported in that paper. Additionally, the monobodies here are in a scaffold that has already been shown to be useful for biosensor development [2], as seen in Chapter 3.

Figure 4.1: Keap1 has a positively charged binding pocket for DEETGE
This figure, rendered in PyMOL [36] from PDB 2FLU [7, 37], shows the Keap1 Kelch-like β-propeller domain, cocrystallized with a 16mer peptide derived from its native partner Nrf2. In both panels, the electrostatically colored surface (red is negative, blue is positive) shows Keap1, and the rainbow ribbon shows the peptide (blue for the N-terminus, shading through to red for the C-terminus). The binding sequence DEETGE, responsible for much of the binding affinity [6, 19] and used as an anchor in this study, is highlighted as sticks. In A, a top-down view onto the binding pocket shows the β-propeller repeat structure as a cartoon, along with showing how the peptide sits in the pocket. The entire central electrostatic-blue region represents the binding pocket; a pink highlight shows the unfilled portions of the pocket targeted with the non-anchor loop by AnchoredDesign in this study. In panel B, the hairpin structure of the DEETGE sequence, along with the depth of the hairpin binding, is shown. Notice that much of the 16mer peptide, outside of the 6-residue sequence shown as sticks, does not interact directly with Keap1.

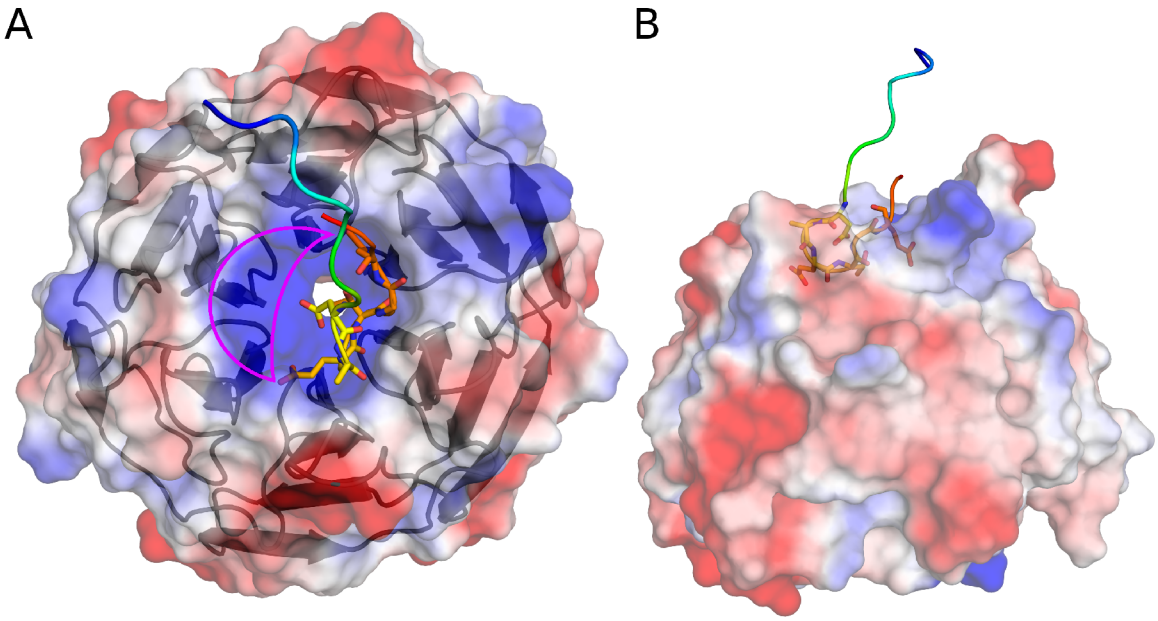


Figure 4.2: Round 1 modeling options
This set of options reproduces the first round of modeling, which is useful for determining which anchor and loop length constructs are useful and seeding round 2. Options are further discussed in the original AnchoredDesign publication [1] (Chapter 2). linmem_ig refers to the linear memory interaction graph [38].

```
-database /path/to/rosetta_database

#adds extra output for debugging & reproducibility
-unmute protocols.loops.CcdLoopClosureMover core.util.prof
-run::version
-options::user

#actual contents of these files vary per construct
-s start.pdb
-loops::loop_file loopsfile
-resfile resfile

#packing options: use extra rotamers for chi1 and chi1 if 8 or more neighbors are present;
use the linmem_ig for better memory/speed, and allow minimized input sidechain to stay
during packing
-ex1
-ex2
-use_input_sc
-extrachi_cutoff 8
-linmem_ig 42

#minimization: use the dfpmin_armijo minimizer and update the neighbor list as needed
-run::min_type dfpmin_armijo
-nblist_autoupdate

#MPI control; no science relevance
-mpi_work_partition_job_distributor
-mpi_tracer_to_file proc

#see the AnchoredDesign documentation for a detailed description of each option
-AnchoredDesign
        -anchor anchor
        -allow_anchor_repack false
        -vary_cutpoints true
        -debug false
        -show_extended false
        -refine_only false
        -perturb_show false
        -perturb_temp 0.8
        -refine_temp 0.8
        -refine_repack_cycles 500
        -rmsd false
        -unbound_mode false
        -no_frags false
        -perturb_CCD_off false
        -perturb_KIC_off false
        -refine_CCD_off false
        -refine_KIC_off false
        -chainbreak_weight 10.0

#number of cycles to run in Monte Carlo.  Note this nstruct was never satisfied; the jobs
instead ran for fixed numbers of CPU hours.
-AnchoredDesign::perturb_cycles 5000
-AnchoredDesign::refine_cycles 10000
-nstruct 9999
```

Figure 4.3: Round 2 modeling options
Rosetta options used in modeling round 2 were the same as those used in round 1, with these listed exceptions. Round 2 used the best results of round 1 as starting models; most of these altered command line flags represent that. vary_cutpoints was deactivated because most models were re-sampled using the cutpoint that produced the start model. refine_only is used to prevent the unwanted low-resolution perturbation stage of AnchoredDesign. rmsd allows calculation of how far models drift from their starting model. no_frags speeds the modeling up; fragments are not used in refinement anyway so there is no need to generate them. The chainbreak weight has been increased to improve loop quality. The number of refinement cycles has decreased because the input models are already of reasonable quality.

```
-AnchoredDesign::vary_cutpoints false
-AnchoredDesign::refine_only true
-AnchoredDesign::rmsd true
-AnchoredDesign::no_frags true
-AnchoredDesign::chainbreak_weight 100.0
-AnchoredDesign::refine_cycles 2500
```

Figure 4.4: Design models selected by phage display
This figure shows Rosetta models for three sequences, 5_11, 5_2, and 5_AWS, which were part of the model population used to generate the design library. Panel A shows the orientation between the monobodies (green, cyan, magenta) and Keap1 (yellow). Most of the good-scoring monobodies adopt this shared orientation. Panels B, C, and D share colors: Keap1 as an electrostatic surface as in Figure 4.1, 10FNIII scaffold in green, BC loop in cyan, FG loop in magenta, and relevant hydrogen bonds as yellow dashes. Panel B demonstrates the designed positions of the anchor loop FG, which makes similar contact in all three models. Highlighted are the hydrogen bonds between these design positions (RdeetgeFHWP) and Keap1. Panel C shows the BC loop for sequences 5_2 and 5AWS (AWSYYEV and AWSYDEV) and panel D shows 5_11 (GDLGTNT). Notice that the BC loops do not insert into the binding pocket as deeply as the native-derived anchor loop, nor is there an extensive network of hydrogen bonds despite the polar nature of the Keap1 surface.

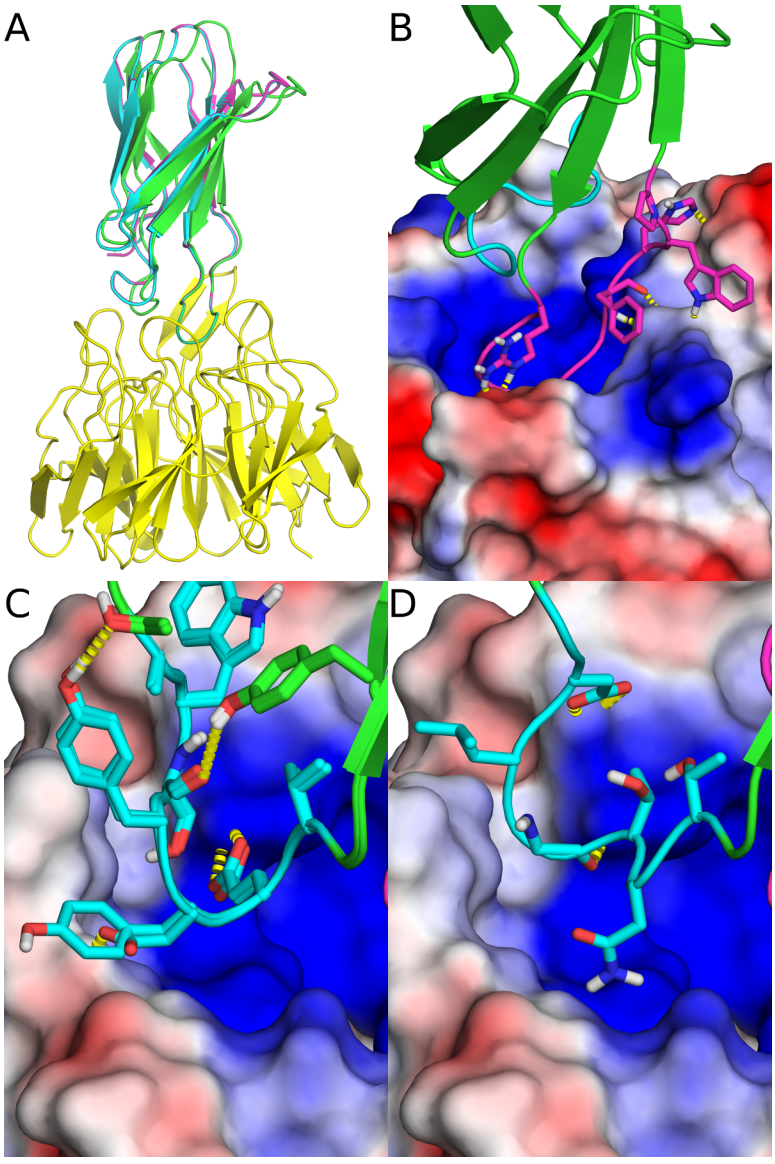


Figure 4.5: Alanized positions in 5_2 and 8_1
This figure, with protein colored as in Figure 4.4, demonstrates the positions at which alanine mutations were used to probe the validity of the design models. 5_2 (AWSYDEV) is in panel A, and 8_1 (GDGHEEI) in panel B. 8_1 is not a design model; this model was created from the known 8_1 sequence using AnchoredDesign. For each model, the orange oval indicates the position of a single alanine reversion in 5_2SM and 8_1SM (S to A and H to A, respectively) and the green ovals indicate the double alanine mutation in 5_2DM and 8_1DM (DE to AA and EE to AA, respectively). Each of these mutations deletes at least one hydrogen bond either between the monobody and Keap1, or within the BC loop. The lack of experimentally-observed deficit in binding affinity for alanization at these positions implies that these design models are incorrect.

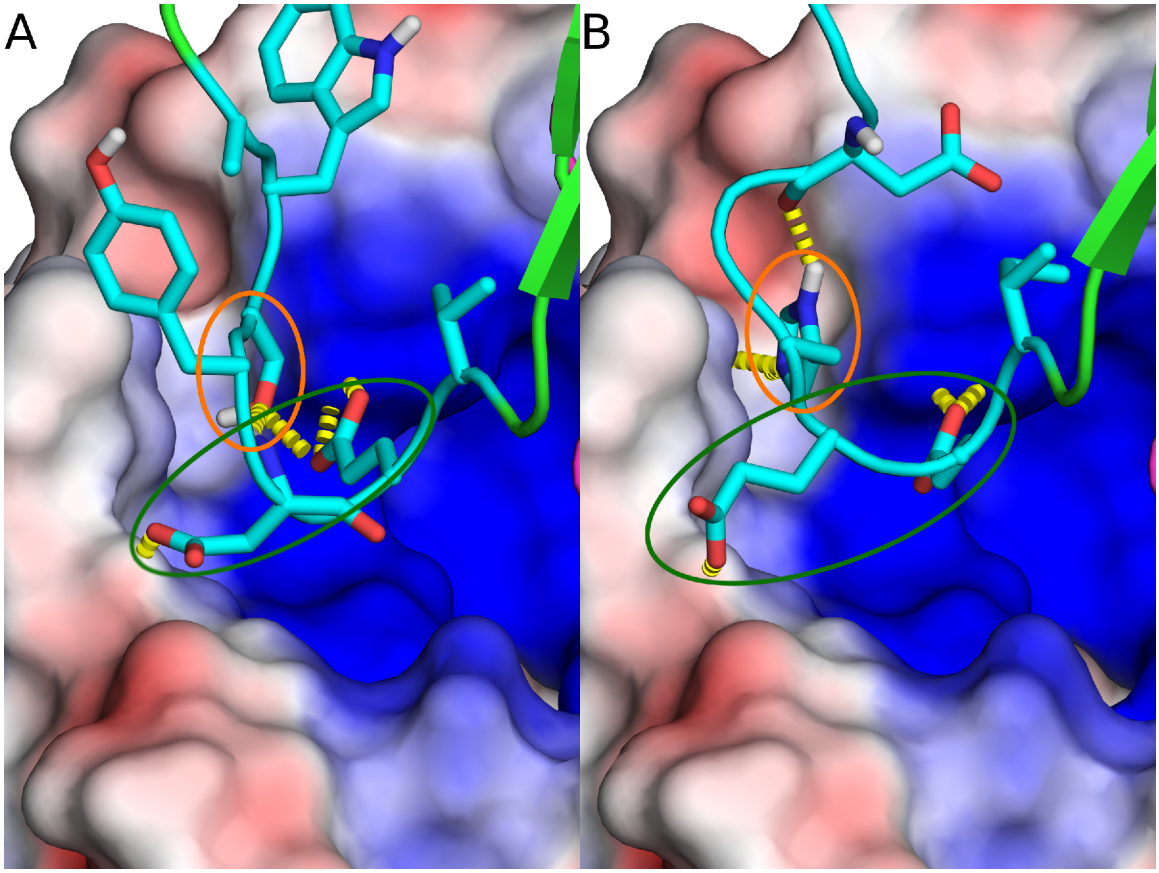


Table 4.1: Loop length and anchor placement constructs

This table lists the loop lengths and anchor positions on the 10FNIII scaffold tested by AnchoredDesign in this work. Position numbers are marked with their equivalents from PDB 1FNA [39], which is offset by -1415 from the PDB 1FNF [40] numbering scheme. Each row represents one computational construct. Lowercase letters indicate fixed scaffold protein sequence, and uppercase letters DEETGE or ETGE indicate fixed mutations to the anchor sequence. Uppercase X indicates a designable position that AnchoredDesign can mutate to any of the standard amino acids except cysteine. A dash indicates that a position in not present in that version of the sequence—some sequences have additions to the BC loop and/or deletions in the FG loop. All designable positions (X) were backbone-flexible during modeling. Each model also has four flexible N-terminal residues EVRD, with EV left over from cloning/expression and RD being native sequence. Each row's nomenclature indicates the construction of the loop length and anchor placement. The # BC plus marker indicates the number of residues added to the BC loop for FG-anchored constructs. For the #_##.## nomenclature, the first digit is the anchor length (4 for ETGE into BC, and 6 for DEETGE into FG), and the two-digit pair represents which scaffold residues the anchor replaces.

	BC loop														FG loop														
Position	21	22	23	24	25	26	27	28	-	-	29	30	31	32	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89
10FNIII sequence	s	w	d	a	p	a	v	t	-	-	v	r	y	y	a	v	t	g	r	g	d	s	p	a	s	s	k	p	i	s
BC anchor																														
4_25.28	s	w	X	X	E	T	G	E	-	-	X	X	y	y	a	v	X	X	X	X	X	X	X	X	X	X	X	X	i	s
4_26.28	s	w	X	X	X	E	T	G	E	-	X	X	y	y	a	v	X	X	X	X	X	X	X	X	X	X	X	X	i	s
BC plus 0																														
6_77.82	s	w	d	X	X	X	X	X	-	-	X	r	y	y	a	v	X	D	E	E	T	G	E	X	X	X	X	X	i	s
6_77.83	s	w	d	X	X	X	X	X	-	-	X	r	y	y	a	v	X	D	E	E	T	G	E	-	X	X	X	X	i	s
6_77.84	s	w	d	X	X	X	X	X	-	-	X	r	y	y	a	v	X	D	E	E	T	G	E	-	-	X	X	X	i	s
BC plus 1																														
6_77.82	s	w	d	X	X	X	X	X	X	-	X	r	y	y	a	v	X	D	E	E	T	G	E	X	X	X	X	X	i	s
6_77.83	s	w	d	X	X	X	X	X	X	-	X	r	y	y	a	v	X	D	E	E	T	G	E	-	X	X	X	X	i	s
6_77.84	s	w	d	X	X	X	X	X	X	-	X	r	y	y	a	v	X	D	E	E	T	G	E	-	-	X	X	X	i	s
BC plus 2																														
6_77.82	s	w	d	X	X	X	X	X	X	X	X	r	y	y	a	v	X	D	E	E	T	G	E	X	X	X	X	X	i	s
6_77.83	s	w	d	X	X	X	X	X	X	X	X	r	y	y	a	v	X	D	E	E	T	G	E	-	X	X	X	X	i	s
6_77.84	s	w	d	X	X	X	X	X	X	X	X	r	y	y	a	v	X	D	E	E	T	G	E	-	-	X	X	X	i	s

Table 4.2: Contents of directed library

This table, graciously provided by Dr. Gurkan Guntas, lists the protein sequences chosen and ordered as part of the directed display library. The listed sequence is a 7-residue BC loop sequence (replacing the X residues in row BC plus 1, 6_77.83 in Table 4.1). Underlined residues represent residue types not found in the batch of 80 designs that slip in due to codon degeneracy. A dash stands for a stop codon.

Oligo number	Protein sequence covered by variable region of oligonucleotide	# of Sequences	# of Models covered	Model IDs covered
1	(A/S)D(L/W/S/-)(G/S)(T/S)(D/N)(V/I)	128	8	4,5,6,9,10,70,72,76
2	(A/S)D(A/K/T/E)GTLV	8	3	3,52,59
3	A(D/F/Y/V)SYYEV	4	2	7,11
4	(A/G)WS(H/L/Y/F)(D/Y)EV	16	5	12,13,14,46,47
5	FWSLYEV	1	1	19
6	EDRGTDV	1	1	18
7	GD(D/N/R/S/H/G)H(D/E/Q/H)(D/E/H/Q)I	96	14	20,21,22,26-31,34-38
8	GDNS(E/R/Q/G)EI	4	2	32,33
9	G(D/E/H/Q)SQDEV	4	3	39,40,44
10	G(E/H/Q/D)S(S/T)DEV	8	2	41,45
11	GHSQDEH	1	1	43
12	SD(A/H/D/P)G(T/S)DV	8	4	51,54,56,78
13	SD(H/Q/N/K/D/E/R/S/G)D(D/N/R/S/G/H)(H/N)V	108	11	53,55,57,60,61,62,64-67,69
14	(A/S/Q/P/Y)D(K/R)GTD(V/I)	20	5	1,2,58,68,74
15	ADWGSNH	1	1	8
16	DHN(R/H)(E/H/R/K/S/N/G)(D/E/H/Q)V	56	3	15,16,17
17	GD(F/L/D/E/V/Y/-)(H/S/N/R)(D/N)H(V/I)	112	3	23,24,25
18	GFSTDEI	1	1	42
19	HDYGTDV	1	1	48
20	MDFGTLV	1	1	49
21	RHNHSDV	1	1	50
22	SDNSFEV	1	1	63
23	SWLGTDV	1	1	71
24	AWFGSDI	1	1	73
25	SDVGTHV	1	1	75
26	SDSGSNV	1	1	77
27	GDLGTNT	1	1	79
28	SDDSDHV	1	1	80
Totals		587	80	

Table 4.3: Affinities of Keap1-binding monobodies by ITC and FP
This table showcases the binding affinities between the Keap1 β-propeller and various binding partners. Most sequences are monobodies based on 10FNIII; one is a 10mer peptide based on Keap1's native target Nrf2. Unmodified 10FNIII was not found to bind Keap1, but a mixed sequence with a wild-type BC loop and an anchor-containing FG loop (wtBC) bound with low micromolar affinity. Sequence 5_2 is marked with a star to indicate that it existed in the sequence database from which the library was generated (although not in the 80 sequences used to generate the library). 5_11 and 5_AWS are marked with two stars to indicate their membership in the population used to generate the library. 8_1 and 8_6 contain sequences allowed by codon degeneracy but not deliberately placed in the library. This table includes all sequences tested from these two libraries. All measurements are of at least two repeat experiments, except R1 by ITC and 8_1 SM by FP.

Sequence	BC loop	FG loop		
ITC measurements				
Controls			K_d (nM)	n
wt	APAVTV-	TGRGDSPASSK	undetectable	
wtBC	APAVTV-	RDEETGEFHWP	1555	1.01
Directed library			K_d (nM)	n
5-2*	AWSYDEV	RDEETGEFHWP	104	0.82
5_11**	GDLGTNT	RDEETGEFHWP	116	0.84
5_AWS**	AWSYYEV	RDEETGEFHWP	202	0.83
8_1	GDGHEEI	RDEETGEFHWP	849	0.87
8_6	GDHHEDI	RDEETGEFHWP	328	0.90
Random library			K_d (nM)	n
R1	RAYGYPS	RDEETGEFHWP	3	1.08
R6	LRRFGRQ	RDEETGEFHWP	37	0.94
FP measurements				
Directed library and alanine mutants			K_i (nM)	
5_2	AWSYDEV	RDEETGEFHWP	201	
5_2 SM	AWAYDEV	RDEETGEFHWP	260	
5_2 DM	AWSYAAV	RDEETGEFHWP	185	
8_1	GDGHEEI	RDEETGEFHWP	690	
8_1 SM	GDGAEEI	RDEETGEFHWP	906	
8_1 DM	GDGHAAI	RDEETGEFHWP	663	
Random library			K_i (nM)	
R1	RAYGYPS	RDEETGEFHWP	0.3	
Controls			K_d (nM)	
peptide	-	LDEETGEFLK-5FAM	141	

Table 4.4: Rosetta-predicted energies of alanized interfaces
This table lists Rosetta-calculated binding energies for a variety of alanine mutants to selected sequences. Rosetta energy calculation experiments were performed by Thomas Lane using AnchoredDesign, with settings similar to those elsewhere in this work. It should be noted that these are not fixed-backbone mutant energy predictions; these are for AnchoredDesign repredictions of the mutant sequence, starting at good-scoring models of the 5_2 or 8_1 sequence.

Sequence	BC loop	ΔΔTotal Energy (REU)	ΔΔG Binding
5_2	AWSYDEV	-	-
5_2 SM	AWAYDEV	1.816	1.024
5_2 DM	AWSYAAV	5.008	2.47
8_1	GDGHEEI	-	-
8_1 SM	GDGAEEI	3.323	4.183
8_1 DM	GDGHAAI	2.289	2.642

References

1. Lewis SM, Kuhlman BA. (2011) Anchored design of protein-protein interfaces. PLoS One 6(6): e20872. 10.1371/journal.pone.0020872.

2. Gulyani A, Vitriol E, Allen R, Wu J, Gremyachinskiy D, et al. (2011) A biosensor generated via high-throughput screening quantifies cell edge src dynamics. Nat Chem Biol 7(7): 437-444. 10.1038/nchembio.585; 10.1038/nchembio.585.

3. Koide A, Bailey CW, Huang XL, Koide S. (1998) The fibronectin type III domain as a scaffold for novel binding proteins. J Mol Biol 284(4): 1141-1151.

4. Itoh K, Wakabayashi N, Katoh Y, Ishii T, Igarashi K, et al. (1999) Keap1 represses nuclear activation of antioxidant responsive elements by Nrf2 through binding to the amino-terminal Neh2 domain. Genes Dev 13(1): 76-86.

5. Baird L, Dinkova-Kostova AT. (2011) The cytoprotective role of the Keap1-Nrf2 pathway. Arch Toxicol 85(4): 241-272. 10.1007/s00204-011-0674-5.

6. Kobayashi M, Itoh K, Suzuki T, Osanai H, Nishikawa K, et al. (2002) Identification of the interactive interface and phylogenic conservation of the Nrf2-Keap1 system. Genes Cells 7(8): 807-820.

7. Lo SC, Li X, Henzl MT, Beamer LJ, Hannink M. (2006) Structure of the Keap1:Nrf2 interface provides mechanistic insight into Nrf2 signaling. EMBO J 25(15): 3605-3617. 10.1038/sj.emboj.7601243.

8. Padmanabhan B, Tong KI, Ohta T, Nakamura Y, Scharlock M, et al. (2006) Structural basis for defects of Keap1 activity provoked by its point mutations in lung cancer. Mol Cell 21(5): 689-700. 10.1016/j.molcel.2006.01.013.

9. Li X, Zhang D, Hannink M, Beamer LJ. (2004) Crystal structure of the kelch domain of human Keap1. J Biol Chem 279(52): 54750-54758. 10.1074/jbc.M410073200.

10. Tong KI, Padmanabhan B, Kobayashi A, Shang C, Hirotsu Y, et al. (2007) Different electrostatic potentials define ETGE and DLG motifs as hinge and latch in oxidative stress response. Mol Cell Biol 27(21): 7511-7521. 10.1128/MCB.00753-07.

11. Padmanabhan B, Nakamura Y, Yokoyama S. (2008) Structural analysis of the complex of Keap1 with a prothymosin alpha peptide. Acta Crystallogr Sect F Struct Biol Cryst Commun 64(Pt 4): 233-238. 10.1107/S1744309108004995.

12. Komatsu M, Kurokawa H, Waguri S, Taguchi K, Kobayashi A, et al. (2010) The selective autophagy substrate p62 activates the stress responsive transcription factor Nrf2 through inactivation of Keap1. Nat Cell Biol 12(3): 213-223. 10.1038/ncb2021.

13. Karanicolas J, Corn JE, Chen I, Joachimiak LA, Dym O, et al. (2011) A de novo protein binding pair by computational design and directed evolution. Mol Cell 42(2): 250-260. 10.1016/j.molcel.2011.03.010.

14. Fleishman SJ, Whitehead TA, Ekiert DC, Dreyfus C, Corn JE, et al. (2011) Computational design of proteins targeting the conserved stem region of influenza hemagglutinin. Science 332(6031): 816-821. 10.1126/science.1202617.

15. Hayes RJ, Bentzien J, Ary ML, Hwang MY, Jacinto JM, et al. (2002) Combining computational and experimental screening for rapid optimization of protein properties. Proc Natl Acad Sci U S A 99(25): 15926-15931. 10.1073/pnas.212627499.

16. Treynor TP, Vizcarra CL, Nedelcu D, Mayo SL. (2007) Computationally designed libraries of fluorescent proteins evaluated by preservation and diversity of function. Proc Natl Acad Sci U S A 104(1): 48-53. 10.1073/pnas.0609647103.

17. Guntas G, Purbeck C, Kuhlman B. (2010) Engineering a protein-protein interface using a computationally designed library. Proc Natl Acad Sci U S A 107(45): 19296-19301. 10.1073/pnas.1006528107.

18. Lippow SM, Moon TS, Basu S, Yoon SH, Li X, et al. (2010) Engineering enzyme specificity using computational design of a defined-sequence library. Chem Biol 17(12): 1306-1315. 10.1016/j.chembiol.2010.10.012.

19. Tong KI, Katoh Y, Kusunoki H, Itoh K, Tanaka T, et al. (2006) Keap1 recruits Neh2 through binding to ETGE and DLG motifs: Characterization of the two-site molecular recognition model. Mol Cell Biol 26(8): 2887-2900. 10.1128/MCB.26.8.2887-2900.2006.

20. Chen Y, Inoyama D, Kong AN, Beamer LJ, Hu L. (2011) Kinetic analyses of Keap1-Nrf2 interaction and determination of the minimal Nrf2 peptide sequence required for Keap1 binding using surface plasmon resonance. Chem Biol Drug Des 78(6): 1014-1021. 10.1111/j.1747-0285.2011.01240.x; 10.1111/j.1747-0285.2011.01240.x.

21. Canutescu AA, Dunbrack RL. (2003) Cyclic coordinate descent: A robotics algorithm for protein loop closure. Protein Sci 12(5): 963-972.

22. Wang C, Bradley P, Baker D. (2007) Protein-protein docking with backbone flexibility. J Mol Biol 373(2): 503-519.

23. Li X, Zhang D, Hannink M, Beamer LJ. (2004) Crystallization and initial crystallographic analysis of the kelch domain from human Keap1. Acta Crystallogr D Biol Crystallogr 60(Pt 12 Pt 2): 2346-2348. 10.1107/S0907444904024825.

24. Jha RK, Leaver-Fay A, Yin S, Wu Y, Butterfoss GL, et al. (2010) Computational design of a PAK1 binding protein. J Mol Biol 400(2): 257-270. 10.1016/j.jmb.2010.05.006.

25. Lungu OI, Hallett RA, Choi EJ, Aiken MJ, Hahn KM, et al. (2012) Designing photoswitchable peptides using the AsLOV2 domain. Chemistry & Biology In press.

26. Fang X, Wang R. (2004) The ki calculator for fluoresence-based competitve binding assays. 2012(04/12).

27. Nikolovska-Coleska Z, Wang R, Fang X, Pan H, Tomita Y, et al. (2004) Development and optimization of a binding assay for the XIAP BIR3 domain using fluorescence polarization. Anal Biochem 332(2): 261-273. 10.1016/j.ab.2004.05.055.

28. Purbeck C, Eletr ZM, Kuhlman B. (2010) Kinetics of the transfer of ubiquitin from UbcH7 to E6AP. Biochemistry 49(7): 1361-1363. 10.1021/bi9014693.

29. Hancock R, Bertrand HC, Tsujita T, Naz S, El-Bakry A, et al. (2012) Peptide inhibitors of the Keap1-Nrf2 protein-protein interaction. Free Radic Biol Med 52(2): 444-451. 10.1016/j.freeradbiomed.2011.10.486.

30. Rohl CA, Strauss CEM, Misura KMS, Baker D. (2004) Protein structure prediction using rosetta. Methods Enzymol 383: 66-+.

31. Kuhlman B, Dantas G, Ireton GC, Varani G, Stoddard BL, et al. (2003) Design of a novel globular protein fold with atomic-level accuracy. Science 302(5649): 1364-1368.

32. Kortemme T, Joachimiak LA, Bullock AN, Schuler AD, Stoddard BL, et al. (2004) Computational redesign of protein-protein interaction specificity. Nat Struct Mol Biol 11(4): 371-379.

33. Stranges PB, Machius M, Miley MJ, Tripathy A, Kuhlman B. (2011) Computational design of a symmetric homodimer using beta-strand assembly. Proc Natl Acad Sci U S A 108(51): 20562-20567. 10.1073/pnas.1115124108.

34. Stranges PB, Kuhlman BA. (In preparation.) A comparison of successful and failed computational protein interfaces. .

35. Leaver-Fay A, Tyka M, Lewis SM, Lange OF, Thompson J, et al. (2011) ROSETTA3: An object-oriented software suite for the simulation and design of macromolecules. Methods Enzymol 487: 545-574. 10.1016/B978-0-12-381270-4.00019-6.

36. Schrödinger L. (2002) The PyMOL molecular graphics system. 1.4.1. Available: http://www.pymol.org; via the Internet.

37. Berman HM, Westbrook J, Feng Z, Gilliland G, Bhat TN, et al. (2000) The protein data bank. Nucleic Acids Res 28(1): 235-242.

38. Leaver-Fay A, Snoeyink JS, Kuhlman B. (2008) On-the-fly rotamer pair energy evaluation in protein design. The 4th International Symposium on Bioinformatics Reasearch and Applications (ISBRA 2008) : 343-354.

39. Dickinson CD, Veerapandian B, Dai XP, Hamlin RC, Xuong NH, et al. (1994) Crystal-structure of the 10th type-iii cell-adhesion module of human fibronectin. J Mol Biol 236(4): 1079-1092.

40. Leahy DJ, Aukhil I, Erickson HP. (1996) 2.0 angstrom crystal structure of a four-domain segment of human fibronectin encompassing the RGD loop and synergy region. Cell 84(1): 155-164.

Chapter 5:

FloppyTail modeling of Cdc34-Cul1-Rbx1

Introduction

Ubiquitin is the most important regulator of protein lifetime in the cell because it is the posttranslational signal used by the cell to direct unwanted, damaged, or end-of-lifetime proteins to the proteasome for degradation and recycling [2, 3]. Ubiquitin serves other signaling purposes in the cell [4], so the formation of proteasome-directing lysine-48-linked chains of four or more ubiquitin molecules [3] must be carefully regulated to prevent accidental formation of misinterpretable, incomplete degradation signals. One way nature has accomplished this goal is via processivity: once polyubiquitination begins, the polyubiquitinating complex is unlikely to release its target substrate until a sufficient number of ubiquitins have been attached. One such polyubiquitinating complex is the Cdc34-SCF complex [5]. In this chapter, we present a computational model which supports a kinetic and structural hypothesis describing the source of processivity in ubiquitin conjugation by Cdc34-SCF [1].

Cdc34 is a ubiquitin-conjugating enzyme (E2), and SCF denotes a family of polysubunit ubiquitin ligases (E3). The particular SCF studied in this chapter contains a RING domain protein Rbx1, a scaffolding cullin Cul1, adaptor Skp1, and F-box containing recruiter β-TrCP. The SCF nomenclature recognizes Skp, Cullin, and F-box. During ubiquitin conjugation, the E2 Cdc34 delivers an activated ubiquitin molecule to a substrate bound onto the E3 SCF complex. The purpose of the SCF and its subunits are

to provide a scaffold of the correct geometry and specificity to bring E2-carried activated ubiquitin to the target substrates. For the purposes of the modeling presented here, only Cdc34 binding to the Rbx1+Cul1 subcomplex is significant.

Cdc34-SCF is known to be processive [5]. Processivity ensures that one substrate capture by SCF usually leads to polyubiquitination, reducing the occurrence of ambiguous mono-, di-, and triubiquitinations intended as degradation signals. However, it is also known that the Cdc34-SCF complex (ignoring the substrate) cannot be maintained during polyubiquitination, because the ubiquitin activating enzyme E1 and SCF's RING domain bind the same surface of Cdc34 [6]. This means that the E2-E3 complex must dissociate to allow E1 to charge E2 with a fresh ubiquitin, without the E3-substrate complex coming apart.

One possible mechanism for this processivity would be if E2-E3 association and dissociation kinetics are much faster than E3-SCF turnover kinetics. This turns out to be the case: the association rate constant k_{on} for Cdc34 and a partial SCF complex (Rbx1+Cul1) was measured to be 4.7×10^8 $M^{-1}s^{-1}$ [1], more than the diffusion-controlled upper limit rate constant of $\sim 10^6$ $M^{-1}s^{-1}$ [7]. The best understood violation of the diffusion limit for protein-protein association occurs when proteins interact via electrostatically charged interactions, whose long range allows for more and better pre-binding transient complexes and thus more binding events per unit time[6, 8]. The hypothesis of a charge-based interaction between Cdc34 and SCF is supported by the fact that high salt concentrations interfere with both Cdc34-SCF association and that of charged complexes in general [1].

The fast, plausibly charge-based association kinetics of Cdc34-SCF draw attention to a peculiar, incompletely understood aspect of Cdc34: its highly negatively charged C-terminal tail. Eleven of the last 20 positions are acidic, and overall 20 of the 57 residues that define the tail are aspartate or glutamate. This tail is necessary for Cdc34's biological activity [9], and its grafting onto another E2 can partially convert that E2 to behave like Cdc34 [10, 11]. Truncation or partial deletion of the tail severely impairs ubquitin throughput [1].

To rationalize a charge-based interaction between SCF and Cdc34, a positively charged region on SCF must exist to interact with the negatively charged Cdc34 tail. Such a patch does exist on Cul1 (Figure 5.1). Furthermore, base-to-acid mutations in this basic cleft proved to produce a kinetic defect similar to deletion of the Cdc34 acidic tail, implying a tail-cleft interaction [1].

The hypothesis that the Cdc34 tail and the Cul1 basic patch interact is a structural explanation for kinetic data. To test the validity of this vague structural assertion, we modeled the putative tail-patch interface with Rosetta. The tail itself has neither predicted nor detectable secondary structure [1, 12], nor does it appear in crystal structures of Cdc34 [13]. Given its large net charge this is unsurprising. For this reason, we modeled whether it is geometrically possible and energetically reasonable for the tail to reach the basic patch, rather than focusing on one high-precision binding model.

Methods

This methods section is reproduced with minor changes directly from the (computational) Supplemental Experimental Procedures of the source paper for this chapter: Kleiger G, Saha A, Lewis S, Kuhlman B, Deshaies RJ. Rapid E2-E3 assembly

and disassembly enable processive ubiquitylation of cullin-RING ubiquitin ligase substrates. Cell. 2009 Nov 25;139(5):957-68.

Generation of starting models

No crystal structure exists of the Cul1-Rbx1-Cdc34 complex. Additionally, no structure exists of the flexible Cdc34 tail we were interested in modeling. To create the requisite starting models, we used a combination of structural alignments to homologous structures and computational docking. Cul1-Rbx1-Cd34 was pieced together from PDB files 2OB4 (human Cdc34) [13] and 1LDJ (Cul1-Rbx1) [14]. Cul1 was trimmed to residues 411-776. To approximate the E2–RING interaction, these two PDB files were aligned against the E2–RING complex CBL-UBCH7, 1FBV [15]. This docking alignment was then refined using Rosetta's docking tools [16].

The final step of generating starting models was attaching the tail sequence to Cdc34 (not seen in the crystal structure). We used the simple expedient of extending the tail straight out into space from the C-terminal residue in the Cdc34 crystal structure, as demonstrated in Figure 5.2.

Modeling the flexible tail

To explore how the flexible tail might interact with the E3 in this system, we built a new protocol within the Rosetta framework. Figure 5.3 demonstrates the flow of this protocol, which was published with Rosetta3.1 as FloppyTail. The protocol has a reduced-representation “centroid” phase and a second refinement phase with all atoms present.

The centroid phase is designed to collapse the long tail, which initially points straight into space, into some reasonable conformation. The reduced representation has

only single large “centroid” atoms replacing the side-chains [17], obviating the need to consider packing details. The advantage of using this representation in the early part of the trajectory is that we no longer need to repack side-chains to smooth out minor side-chain/side-chain clashes, thus speeding up the computation. This phase consisted of 5,000 Metropolis Monte Carlo cycles. Most cycles were used to perturb the tail structure with one of three Rosetta tools [17].

The first tool, marked as Small_180 on Figure 5.3, randomly perturbed a phi and psi angle of a tail residue, weighted to choose good Ramachandran values. The second, Shear_180 on Figure 5.3, modifies a phi angle at position n and psi at position n-1 in opposite directions. This has the effect of tweaking local structure without longer-distance effect. Again, this was a random perturbation subject to Ramachandran constraints. Shear movements are designed to minimize perturbation of the backbone far from the move, so they were disabled for the first third of the protocol to give the tail a chance to develop interactions before their use.

The third type of perturbation was a 3-residue fragment insertion. Fragment insertion consists of replacing the backbone torsional parameters with parameters from a small protein fragment derived from the PDB. Fragments were generated via the Robetta server [18]. This allowed for somewhat faster sampling of local conformational and secondary structure preferences.

After 19 cycles, the protocol performed a gradient-based energy minimization of the structure. The available degrees of freedom were the backbone torsion angles phi, psi and omega of tail residues (the same freedoms perturbed in the other steps). The score function used in this phase contained terms for van der Waals repulsion, hydrogen

bonding, electrostatics, Ramachandran, and residue-pair potentials (which incorporate van der Waals attraction and solvation) [17].

After 5,000 perturbing cycles, the lowest-scoring structure seen in the trajectory was recovered. Side-chains were returned to this structure. The crystallographic side-chains were used outside the tail. One of Rosetta's default full atomic representation energy functions, score12, was used for scoring in this phase [19, 20]. The tail region and any side-chains within 10 Å of the tail were subjected to a combinatorial repacking wherein a Monte Carlo procedure chose low-energy side-chain conformations from the full set of available rotamers. The side-chains and the tail backbone torsions were then energy minimized. The refinement phase ran for 3,000 cycles of Metropolis Monte Carlo optimization. Most cycles consisted of a random small perturbation. These perturbations were small or shear movements as before, except the perturbation was constrained to within 4 degrees, instead of anywhere in Ramachandran space. This sort of perturbation was immediately followed by a single-pass random-order rotamer optimization of the tail and its neighbors (rotamer trials). After 14 cycles, a minimization of the tail side-chains, tail backbone torsions, and side-chains neighboring the tail was performed. Another 14 cycles after that, a full combinatorial repacking operation was performed before the minimization. At the end of each trajectory, the best structure recorded was reported as output.

The protocol as a whole was run for approximately 30,000 trajectories, which is slightly more than 1,000 processor-days on a 2.3 GHz chip. It is customary to run extremely large numbers of trajectories when doing structure prediction with Rosetta. The random nature of Monte Carlo sampling means that trajectories will sometimes get

trapped in poor minima, so many trajectories are needed to sufficiently sample the structure space.

Sorting models

To search through our dataset for models that addressed the question of how the E2 might bind the E3, we sorted models based on E2-E3 binding energy. This was calculated as the complex score minus the scores of the E2 and E3-RING in isolation. It should be noted that there are two components to this binding energy. There is a constant component represented by the direct, canonical E2-RING interaction. This interaction was optimized by docking as a precursor step but not modified during the main part of the protocol. The second interaction represents binding between the E2 tail and the body of the E3.

Results

Even before using FloppyTail, the simple Cdc34-Rbx1-Cul1 model with the Cdc34 tail extended into space was sufficient to confirm the hypothesis that the tail could reach the putative binding cleft. As seen in Figure 5.2, the fully extended length of the tail is 200 Å, and the distance from the base of the tail to the basic cleft is less than half that. Of course, such a mental model says nothing about energetically plausible states of the system, so we continued with the FloppyTail protocol to explore possible tail conformations.

Because of the tail's length, lack of predicted secondary structure [1], lack of secondary structure detectable by circular dichroism [12], and inhibition of crystal growth [13], we did not expect the tail to adopt a single binding conformation, nor did we expect FloppyTail to converge on a single candidate binding structure. We instead generated

thousands of structures and analyzed them as a group to tease out structural implications. These models treat the base of the tail as a tether attaching the peptide-like tail to Cdc34 and samples the 3-D spatial envelope the tail might occupy.

We immediately found that FloppyTail preferred models with the tail in contact with the cleft. Figure 5.4 demonstrates the top 20 structures by Cdc34-SCF binding energy, of which 16 have interaction between the tail and cleft. Figure 5.5 demonstrates one such interaction in detail. It should be noted that the model population in general, consisting of 28,574 models, does not show much bias towards the cleft unless the structure energies are taken into account.

Qualitative support for the tail-binding structural hypothesis was encouraging, but we wanted a quantifiable measure of tail binding, and preferably a specifically verifiable prediction from the model. The native tail sequence contains a free cysteine residue at position 227, suggesting the possibility of a chemical cross-linking experiment. Modeling data suggests this pre-existing cysteine may neighbor Cul1 lysine 679, as the two residues' Cα atoms are within 18 Å in 9 of the 20 best models sorted by binding energy. This tail-to-cleft interatomic distance correlates strongly with binding energy and therefore model quality, as is seen in Figure 5.6, which shows the distribution of this distance for increasingly selective model populations. The model-identified putative Cdc34 C227 to Cul1 K679 interaction was verified by in-vitro chemical cross-linking between C227 and a K679C mutant [1].

Conclusions

The FloppyTail protocol was developed within Rosetta3's framework to address a specific biological problem: the structural hypothesis that Cdc34's acidic tail and Cul1's

basic cleft interacted. Models demonstrating possible modes of interaction were supported by a chemical cross-linking experiment, which demonstrated that a model-predicted interaction was supportable *in vitro*. Collectively, these structural data support the idea that the Cdc34 tail is responsible for the faster-than-diffusion association kinetics of Cdc34 and SCF, which are important for the polyubiquitination function of this E2-E3 system. These data also support this thesis's assertion that Rosetta3's framework makes it straightforward to develop new modeling protocols in response to interesting biological questions, and that the biology and modeling can feed back into each other for useful results.

The FloppyTail protocol was developed for a rather singular purpose, but it has seen further life in similar but distinct biological problems. For example, FloppyTail was used by an independent group of researchers, with documentation-only involvement by the application's author, to model the autoactivation of myosin II heavy chain kinase A via a C-terminal flexible region [21].

Since its original publication, FloppyTail has undergone updates allowing better treatment of N-terminal and fully internal flexible regions. With this extension, it has recently been used to model the flexible internal connection between a PDZ and SH3 domain in PSD95 (Jun Zhang, Steven Lewis, Brian Kuhlman, Andrew Lee; unpublished data). We used FloppyTail in conjunction with extensive experimentally-derived constraints to assemble the domains into docked models, as demonstrated in Figure 5.7. These models were used to predict mutations that might disrupt the interface, and this disruption was confirmed experimentally by NMR. These developments, especially Crawley et al.'s independent use of FloppyTail, demonstrate the unexpected utility and

flexibility of the FloppyTail code. Even code written for a particular purpose, if well written and modular due to the Rosetta3 framework, may find future life in similar but unrelated modeling experiments.

Figure 5.1: The basic patch on Cul1
Presented with as an electrostatic surface (blue is positive, red is negative) are the components of SCF used in these modeling experiments. Present are Rbx1 (upper right) and residues 411-776 of Cul1 (remainder), both from PDB 1LDJ [14]. The basic patch is the large blue cleft in the bottom center of the figure. The camera orientation is shared with Figure 5.4 for comparison.

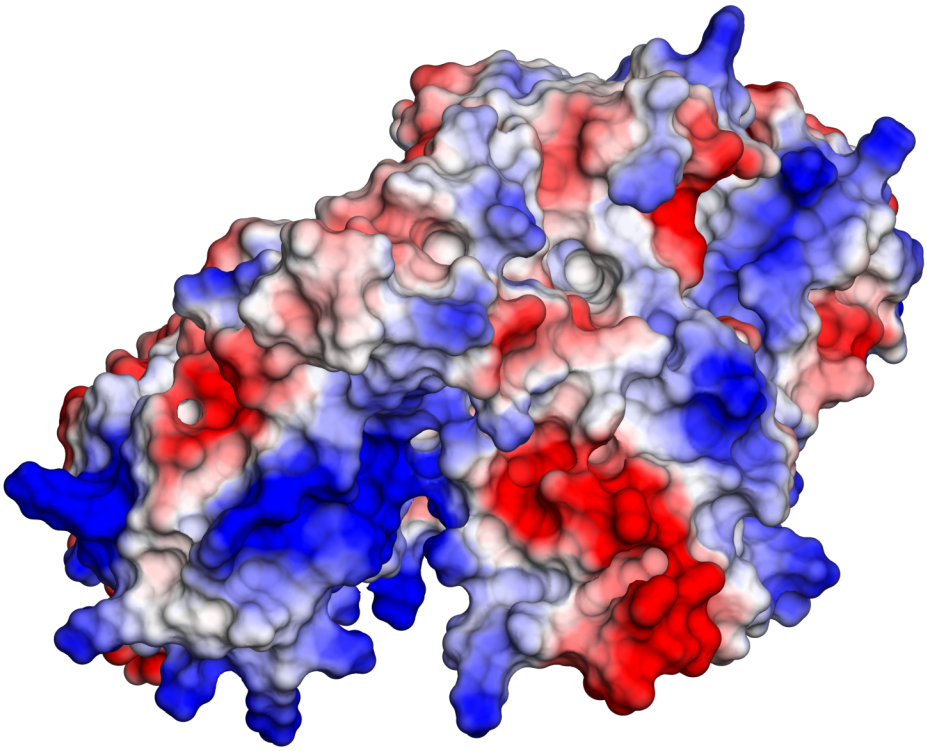


Figure 5.2: The Cdc34 tail's extended length and acidic residues
This figure demonstrates the length of Cdc34's flexible tail, to scale with its folded domain and a portion of SCF. Cdc34 is the magenta cartoon at the top right and Cul1 and Rbx1 are present as an electrostatic surface (blue is positive, red is negative). The tail is yellow, with the acidic residues (aspartate and glutamate) highlighted as salmon and red atomic spheres. The binding cleft is at 7 o'clock in the electrostatic surface Cul1 model, highlighted with an oval. The tail is clearly long enough to reach the binding cleft in terms of linear space, but that is no guarantee that low-energy conformations are available.

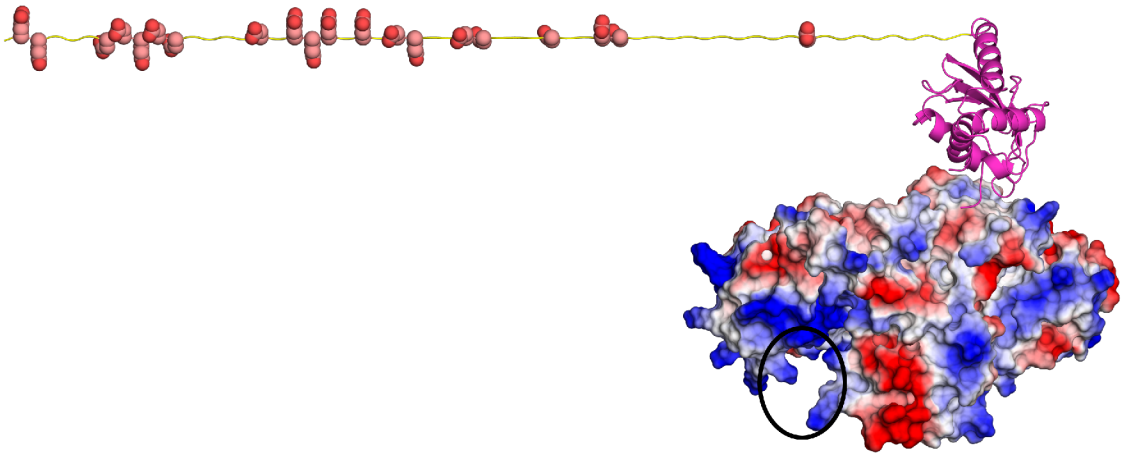


Figure 5.3: FloppyTail control flow
This figure demonstrates the control flow for the FloppyTail protocol. The perturbation stage is on the left and the refinement stage on the right. Trapezoids represent the beginning and end of the protocol. Rectangles represent miscellaneous single-occurrence operations and the loop end within a Monte Carlo trajectory. Rounded rectangles represent Movers modifying the structure inside the Monte Carlo loop. Parallelograms represent a split between the common random operations in Monte Carlo (left) and occasional repacking/minimization operations that occur at fixed intervals (right).

This figure is modified from Figure S8 of the source paper of this chapter [1].

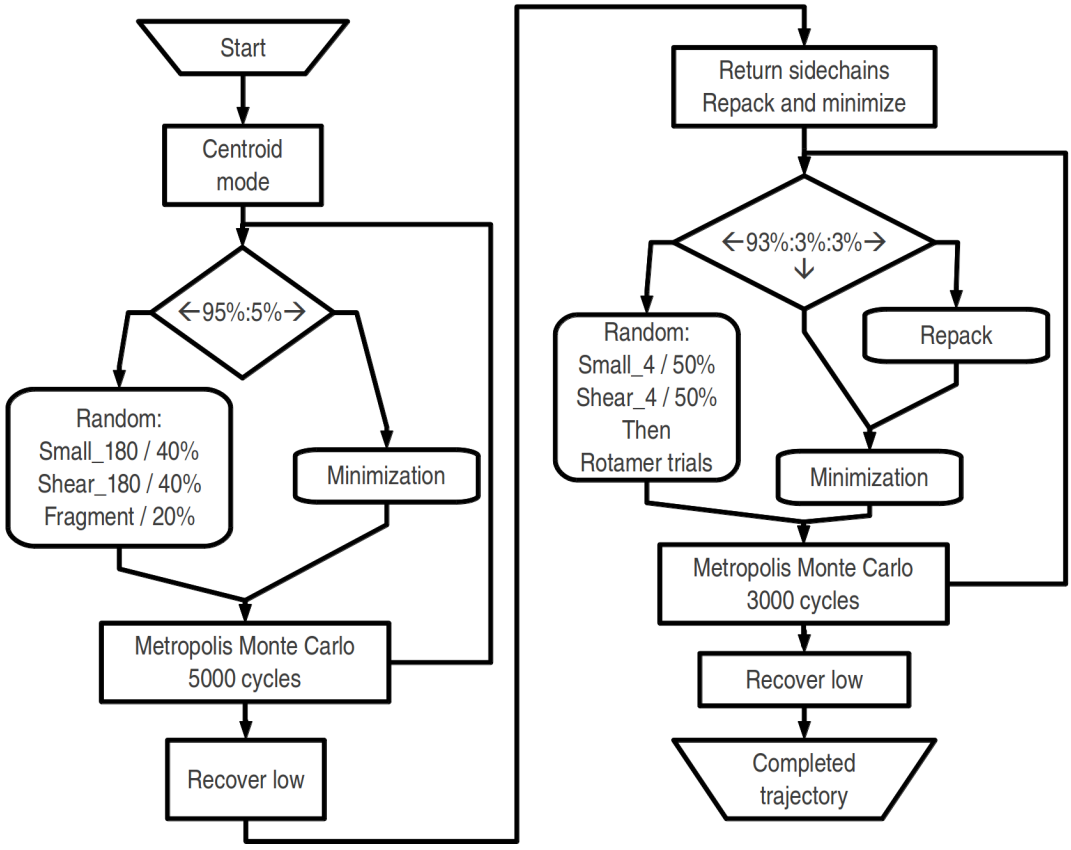


Figure 5.4: The top 20 Cdc34-Cul1-Rbx1 models by binding energy
This figure demonstrates a selection of good-scoring Cdc34 tail models for the Cdc34-Cul1-Rbx1 complex. Cdc34 is the magenta surface at the top, and Cul1 and Rbx1 are present as an electrostatic surface (blue is positive, red is negative). The basic patch is the large blue cleft in the bottom center of the figure. The 20 different tail models are presented in a rainbow of colors. Many tail models weave their way into the pocket, but there is little overall structural similarity in the models. The camera orientation is shared with Figure 5.1 for comparison.

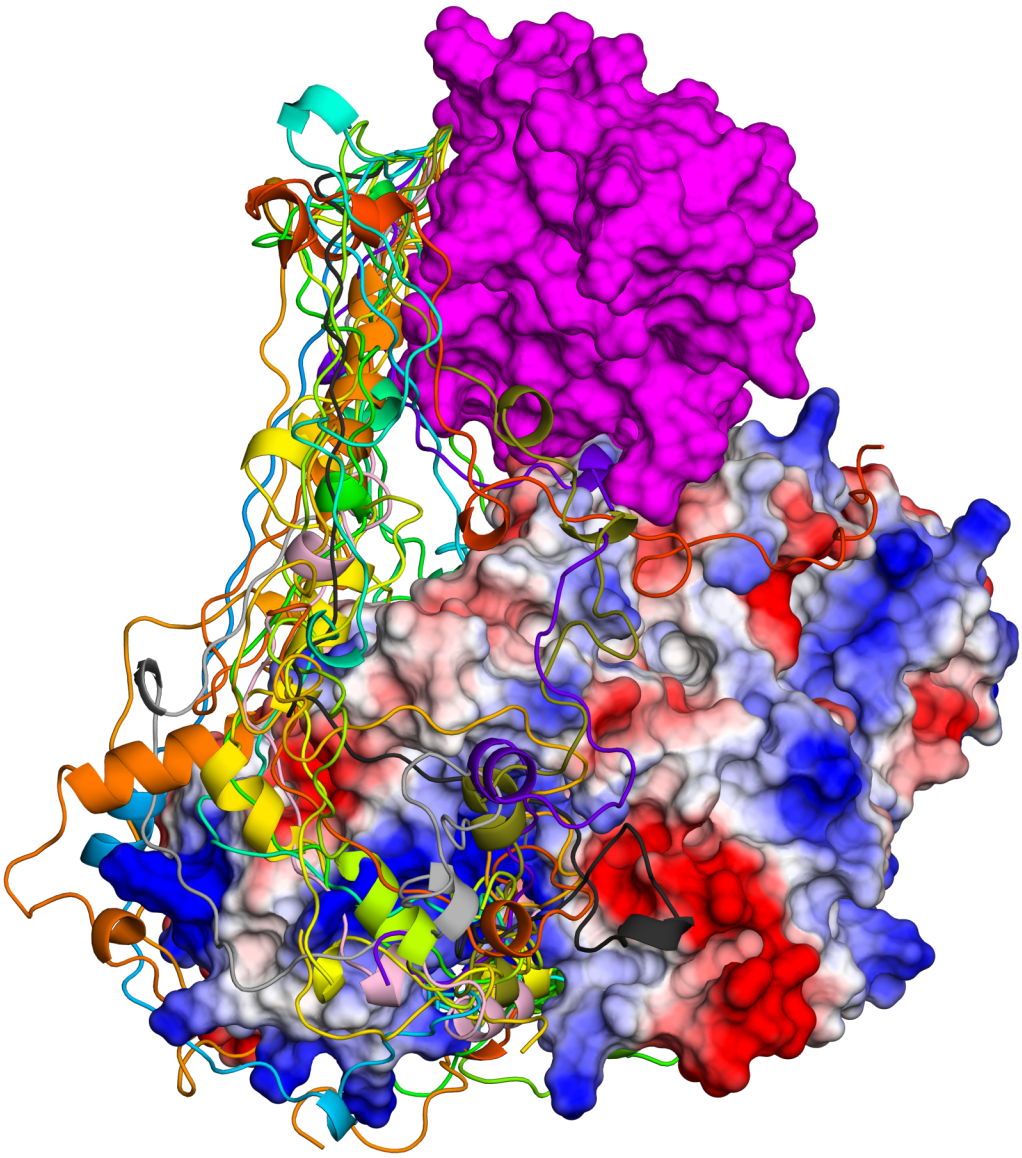


Figure 5.5: A close-up view of the Cdc34 tail in the best-scoring model
This figure zooms in on the tail-cleft interface in the best-scoring (by binding energy) Cdc34-Cul1-Rbx1 complex. Cdc34 is the magenta cartoon at the top, and Cul1 and Rbx1 are present as an electrostatic surface (blue is positive, red is negative). The tail is gold, and acidic sidechains (glutamate and aspartate) are shown as sticks.

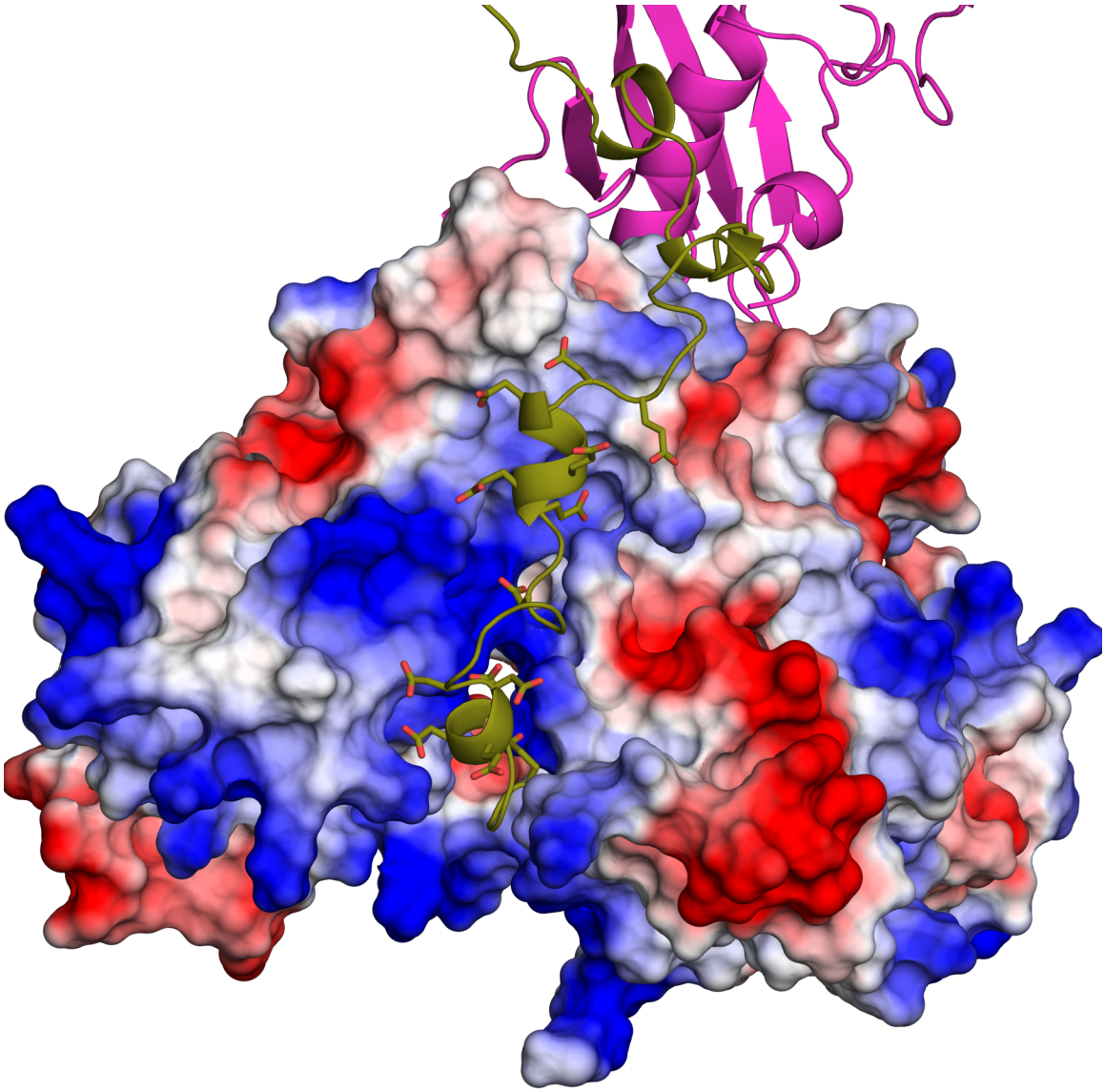


Figure 5.6: Cdc34 C227-Cul1 K679 distance distribution by model score
This histogram shows the distance distribution between the Cα atoms of Cdc34 C227 and Cul1 K679 in the modeling experiment's population of 28,574 models. In red is the distribution for the entire population, in blue is the distribution for the top 20 models by binding energy, and in green is the distribution for the top 20 models by binding energy (those seen in Figure 5.4). Notice that the low-energy distributions are extremely left shifted (shorter distances) as compared to the population as a whole. This indicates that FloppyTail is not sampling the tail-bound conformation efficiently, but Rosetta's score function can detect those rare tail-cleft models as energetically favorable. This modeled distance was experimentally probed with the cross-linking experiment described in the main text.

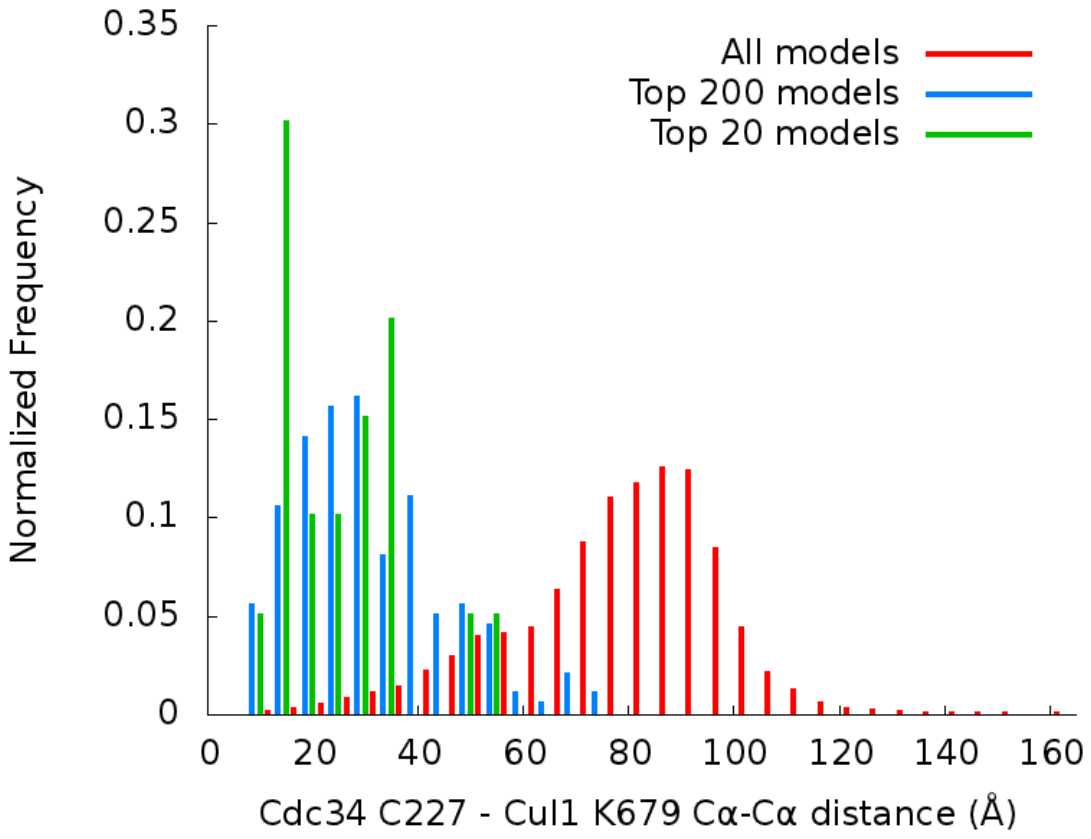
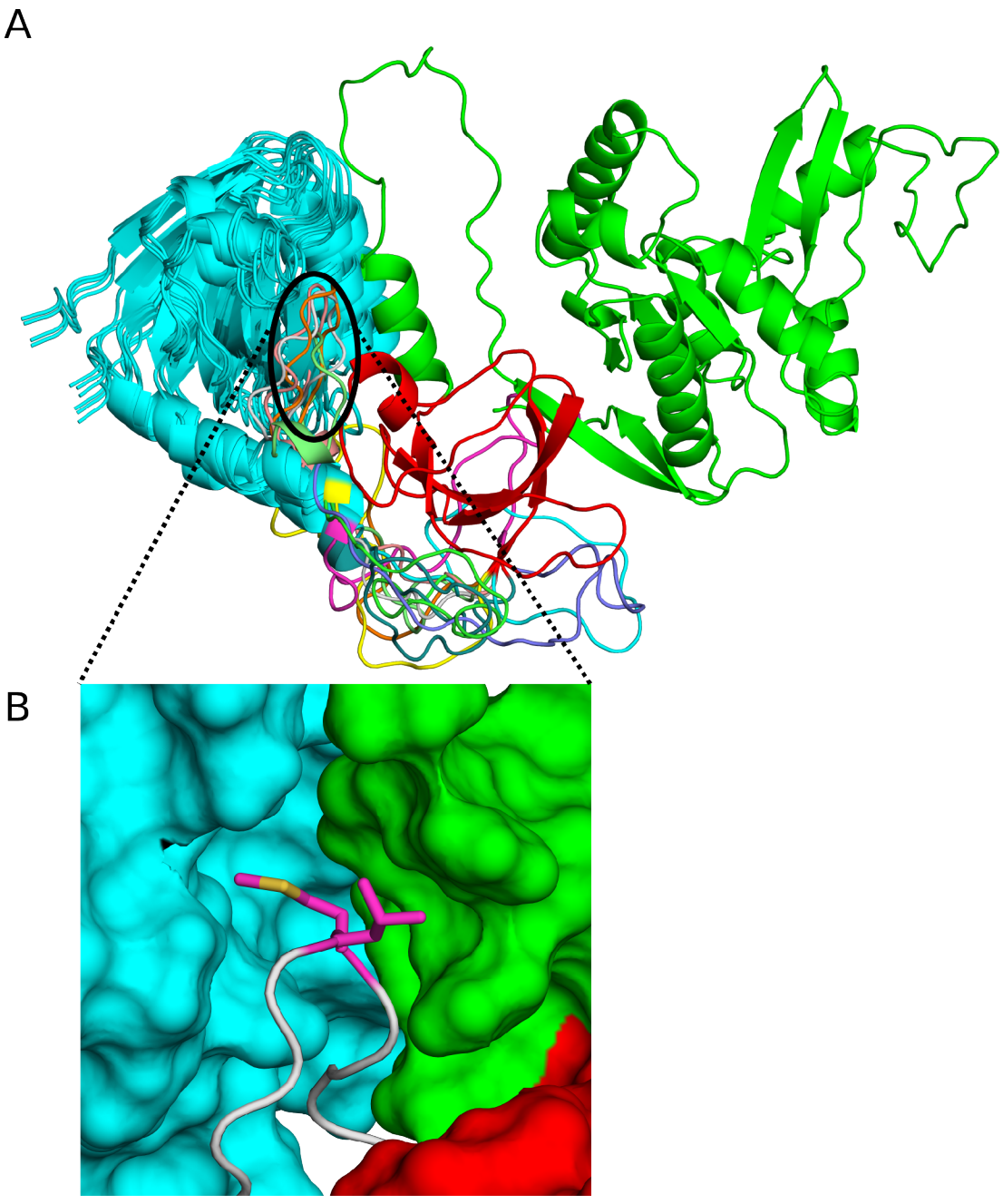


Figure 5.7: FloppyTail in new systems
Panel A shows ten models of protein PSD95. The GK (green) and SH3 (red) domains have been solved as single structures, and the PDZ(3) domain (cyan) has been solved separately. These ten models were assembled via FloppyTail with the help of extensive experimentally-derived constraints, and represent the top ten models by total score (inclusive of the constraint scores). The oval region expands to show one model in panel B. Colors are as before but proteins are now shown as surfaces. Mutations at the leucine and methionine residues, highlighted as sticks, have been experimentally demonstrated to disrupt the interface between the cyan PDZ domain and the rest of the protein. This figure is based on unpublished data from collaboration with Jun Zhang and Andrew Lee.



References

1. Kleiger G, Saha A, Lewis S, Kuhlman B, Deshaies RJ. (2009) Rapid E2-E3 assembly and disassembly enable processive ubiquitylation of cullin-RING ubiquitin ligase substrates. Cell 139(5): 957-968. 10.1016/j.cell.2009.10.030.

2. Chau V, Tobias JW, Bachmair A, Marriott D, Ecker DJ, et al. (1989) A multiubiquitin chain is confined to specific lysine in a targeted short-lived protein. Science 243(4898): 1576-1583.

3. Thrower JS, Hoffman L, Rechsteiner M, Pickart CM. (2000) Recognition of the polyubiquitin proteolytic signal. EMBO J 19(1): 94-102. 10.1093/emboj/19.1.94.

4. Behrends C, Harper JW. (2011) Constructing and decoding unconventional ubiquitin chains. Nat Struct Mol Biol 18(5): 520-528. 10.1038/nsmb.2066.

5. Saha A, Deshaies RJ. (2008) Multimodal activation of the ubiquitin ligase SCF by Nedd8 conjugation. Mol Cell 32(1): 21-31. 10.1016/j.molcel.2008.08.021.

6. Eletr ZM, Huang DT, Duda DM, Schulman BA, Kuhlman B. (2005) E2 conjugating enzymes must disengage from their E1 enzymes before E3-dependent ubiquitin and ubiquitin-like transfer. Nat Struct Mol Biol 12(10): 933-934. 10.1038/nsmb984.

7. Schreiber G, Haran G, Zhou H. (2009) Fundamental aspects of protein-protein association kinetics. Chem Rev 109(3): 839-860. 10.1021/cr800373w.

8. Alsallaq R, Zhou HX. (2008) Electrostatic rate enhancement and transient complex of protein-protein association. Proteins 71(1): 320-335. 10.1002/prot.21679.

9. Mathias N, Steussy CN, Goebl MG. (1998) An essential domain within Cdc34p is required for binding to a complex containing Cdc4p and Cdc53p in saccharomyces cerevisiae. J Biol Chem 273(7): 4040-4045.

10. Kolman CJ, Toth J, Gonda DK. (1992) Identification of a portable determinant of cell cycle function within the carboxyl-terminal domain of the yeast CDC34 (UBC3) ubiquitin conjugating (E2) enzyme. EMBO J 11(8): 3081-3090.

11. Silver ET, Gwozd TJ, Ptak C, Goebl M, Ellison MJ. (1992) A chimeric ubiquitin conjugating enzyme that combines the cell cycle properties of CDC34 (UBC3) and the DNA repair properties of RAD6 (UBC2): Implications for the structure, function and evolution of the E2s. EMBO J 11(8): 3091-3098.

12. Ptak C, Prendergast JA, Hodgins R, Kay CM, Chau V, et al. (1994) Functional and physical characterization of the cell cycle ubiquitin-conjugating enzyme CDC34 (UBC3). identification of a functional determinant within the tail that facilitates CDC34 self-association. J Biol Chem 269(42): 26539-26545.

13. Ceccarelli DF, Tang X, Pelletier B, Orlicky S, Xie W, et al. (2011) An allosteric inhibitor of the human Cdc34 ubiquitin-conjugating enzyme. Cell 145(7): 1075-1087. 10.1016/j.cell.2011.05.039.

14. Zheng N, Schulman BA, Song L, Miller JJ, Jeffrey PD, et al. (2002) Structure of the Cul1-Rbx1-Skp1-F boxSkp2 SCF ubiquitin ligase complex. Nature 416(6882): 703-709. 10.1038/416703a.

15. Zheng N, Wang P, Jeffrey PD, Pavletich NP. (2000) Structure of a c-cbl-UbcH7 complex: RING domain function in ubiquitin-protein ligases. Cell 102(4): 533-539.

16. Gray JJ, Moughon S, Wang C, Schueler-Furman O, Kuhlman B, et al. (2003) Protein-protein docking with simultaneous optimization of rigid-body displacement and side-chain conformations. J Mol Biol 331(1): 281-299.

17. Rohl CA, Strauss CEM, Misura KMS, Baker D. (2004) Protein structure prediction using rosetta. Methods Enzymol 383: 66-+.

18. Kim DE, Chivian D, Baker D. (2004) Protein structure prediction and analysis using the robetta server. Nucleic Acids Res 32: W526-W531.

19. Kuhlman B, Dantas G, Ireton GC, Varani G, Stoddard BL, et al. (2003) Design of a novel globular protein fold with atomic-level accuracy. Science 302(5649): 1364-1368.

20. Smith CA, Kortemme T. (2008) Backrub-like backbone simulation recapitulates natural protein conformational variability and improves mutant side-chain prediction. J Mol Biol 380(4): 742-756.

21. Crawley SW, Gharaei MS, Ye Q, Yang Y, Raveh B, et al. (2011) Autophosphorylation activates dictyostelium myosin II heavy chain kinase A by providing a ligand for an allosteric binding site in the alpha-kinase domain. J Biol Chem 286(4): 2607-2616. 10.1074/jbc.M110.177014.

Chapter 6

UBQ_E2_thioester modeling of the ubiquitin-Cdc34 interface

Introduction

Chapter 5 of this thesis presented a computational model of the interaction between a ubiquitin-conjugating E2 enzyme Cdc34 and a ubiquitin E3 ligase SCF (represented by Cul1 and Rbx1). Not present in that model, but of importance to ubiquitin transfer biology, is the ubiquitin molecule itself. In this chapter, we present a model of the interactions between E2 Cdc34 and a chemically attached, or charged, ubiquitin [1]. This model represents the transfer-competent state that immediately precedes ubiquitin attachment to a target substrate.

As introduced in Chapter 5, ubiquitin is a small, ubiquitous protein that the cell uses as a posttranslational tag. Chains of four or more ubiquitin molecules linked by their lysine 48 residue signal that the ubiquitins' target should go to the proteasome for recycling [2, 3]. Shorter ubiquitin chains and those linked together via other ubiquitin lysines send varying messages [4]. It is therefore important that the ubiquitin-conjugating machinery, E2 and E3 enzymes, maintain specificity in the ubiquitin linkages they form.

Existing models of the thioester-bound E2-ubiquitin complex

Protein function is determined by protein form, and the ultimate results of ubiquitin conjugation depend on the conformational forms sampled while ubiquitin, E2/E3, and substrate are bound together during catalysis. To understand the nature of

these interactions, many authors have reported structures and models of various E2 enzymes chemically charged with ubiquitin [5-10]. Unfortunately, these structures and models have generated no consensus as to what the E2-ubiquitin charged state actually looks like [1]. Figure 6.1 demonstrates the variety in these published E2-ubiquitin models and structures. These conformations represent a variety of different E2 enzymes studied under different conditions, so some variation is to be expected. Nevertheless, this lack of agreement resulted in a vague belief in the field that ubiquitin's orientation relative to the E2 carrying it is unimportant; ubiquitin's precise conformation was not thought to be important for ubiquitin transfer.

The interaction between an E2 and ubiquitin is dominated and complicated by their chemical attachment. Biologically, this is a thioester (also thiolester and thiol ester) bond between the backbone carbonyl of the C-terminal glycine of ubiquitin and the sidechain sulfur of the active site cysteine of the E2, as seen in Figure 6.2. This bond is stable on the biological timescale, but is not stable on the timescale of structural determination experiments like X-ray crystallography or NMR [5]. In fact, existing structures of the thioester state are all models, either computationally assembled [5] as in this work, or created through replacement of the labile thioester with a stable oxyester [7, 8, 10] or a stable disulfide bond [9].

Mutational data suggests a specific interaction

The idea that the ubiquitin-E2 interaction is conformationally nonspecific, especially for Cdc34-ubiquitin, is contradicted by a crucial bit of mutational data. The ubiquitin mutation isoleucine 44 to alanine results in defective discharge of ubiquitin from Cdc34 in both chain-initiating and polyubiquitin-chain-forming reactions [1]. This

defect is due to altered interaction between ubiquitin and E2, largely independent of the target substrate and E3 enzyme present during catalysis. The powerful effect of this single point mutation is inconsistent with the idea that ubiquitin-E2 interactions are unimportant for ubiquitination.

To further explore the I44A mutation, Saha et al. interpreted the Cdc34-ubiquitin interaction by homology to a published Ubc1-ubiquitin model [5]. They rationally predicted an interaction between ubiquitin arginine 42 and Cdc34 glutamate 133. This prediction was borne out by a charge-swap mutation pair ubiquitin R42E and Cdc34 E133R. Ubiquitin R42E is defective in ubiquitin discharge from Cdc34 in a fashion similar to the defect induced by ubiquitin I44A. However, mutation Cdc34 E133R can rescue the activity of ubiquitin R42E. The defect in ubiquitin R42E gives further credence to the hypothesis that the conformation of the ubiquitin-Cdc34 interaction is significant for ubiquitination.

Because ubiquitin R42E can be rescued by a mutation on Cdc34, we wished to test if ubiquitin I44A could be rescued in a similar fashion. We constructed a novel protocol in Rosetta3, UBQ_E2_thioester, to create models of the ubiquitin-Cdc34 charged state, and used this protocol to predict a Cdc34 mutation which rescues ubiquitin I44A activity.

Methods

This methods section is reproduced with some additions directly from the (computational) Supplemental Experimental Procedures of the source paper for this chapter: Saha A, Lewis S, Kleiger G, Kuhlman B, Deshaies RJ. Essential role for

ubiquitin-ubiquitin-conjugating enzyme interaction in ubiquitin discharge from Cdc34 to substrate. Mol Cell. 2011 Apr 8;42(1):75-83. [1]

Rosetta docking model

To model the Cdc34–ubiquitin interaction, a new protocol, UBQ_E2_thioester, was written as a part of the Rosetta3 suite [11]. It was designed to search the conformational space available to ubiquitin chemically conjugated via thioester to Cdc34 by sampling rotation about torsion angles near the thioester bond. This thioester-tethered docking behaves somewhat like rigid-body docking, in that the two protein cores move relative to one another, yet the thioester linkage is maintained.

Selection of input structures

For ubiquitin, PDB 1UBQ [12] was used structurally unmodified except for the removal of waters. For Cdc34, structure 2OB4 [13] was used with minor modifications. 2OB4 has no density for loop residues 102-117, and high b-factors (greater than 40 for all atoms) for residues 99-101. The loop's stem residues 99-101 have few interactions with the rest of the protein. These three residues, along with all water molecules, were deleted from 2OB4 before use. Other modeling (results not shown) demonstrated that computationally modeling this loop along with the interface did not improve the predictive power of the protocol.

Modeling the chemically conjugated complex

As a preliminary step, the protocol chemically bonds the C-terminal Gly76 of ubiquitin to the Cdc34 catalytic cysteine residue. The thioester bond length and 3-body bond angles are set appropriately at this step and not modified throughout the protocol. Next, the protocol varies nearby bond torsion angles to sample the possible interfaces

between ubiquitin and Cdc34. Torsion angles allowed to vary included: both chi angles of the conjugated cysteine 93, the thioester bond, both phi and psi for the ubiquitin terminal Gly76, and phi and psi for the two penultimate ubiquitin residues, Arg74 and Gly75. Figure 6.2 highlights these torsions. Positions Arg74 and Gly75 are chemically unmodified and were treated with Rosetta's standard small and shear moves and score function (including the Ramachandran term) [14]. The other torsions impinging on the thioester cannot be handled by Rosetta's standard score function and standard torsion sampling options, because its knowledge-based terms have no knowledge of thioester. Therefore, they were separately handled with a molecular-mechanics based score term that individually evaluates each bond torsion [15]. Torsional parameters for the thioester bond were based on a simpler methyl thioacetate model published by Yang and Drueckhammer in 2001 [16].

Sampling proceeded with the standard Rosetta Metropolis-Monte Carlo search protocol, detailed in Figure 6.3. At the beginning of each Monte Carlo cycle, a torsion change is proposed. For the nonstandard thioester torsions, they were sampled tightly within acceptable local minima instead of sampled widely, so that any proposed torsion is reasonable for at least the atoms involved in the torsion. If the new torsion produced no major clashes between Cdc34 and ubiquitin, a fast rotamer relaxation protocol followed by minimization of ubiquitin residues Arg74's and Gly75's backbone dihedrals was performed. This structure was subjected to the Metropolis criterion, followed by the next cycle. Interspersed with these changes every hundred cycles was a complete repack of interface residues in lieu of backbone movements. 20,000 cycles were used for these experiments.

To bias sampling towards conformations relevant to the experimental data, constraints were introduced. One constraint encouraged the formation of one or two hydrogen bonds between residues Arg42 of ubiquitin and Glu133 of Cdc34, matching the experimental data. A second constraint favored burial of Ile44, again to favor models supported by experimental data. Possible interactors for Ile44 were drawn from comparison with structure 1FXT [5], a model of ubiquitin bound to a different E2. A third constraint was meant to encourage interaction between ubiquitin Arg42 and the Cdc34 helix containing Ser129, mimicking the bifurcated hydrogen bond between ubiquitin Arg42 and E2 Ala111 in 1FXT. This constraint was never satisfied due to the competing interaction of Arg42 and Glu133, along with the fact that in 2OB4 the helix containing S129 is not kinked to allow a bifurcated hydrogen bond. Figure 6.4 demonstrates the similarity between our models and 1FXT and the impact of constraints on the modeling.

Creation and evaluation of models

Results were automatically filtered according to the solvent-accessible surface area (SASA) buried by the interface and the total score of the complex. SASA was required to be greater than 1000 $Å^2$ and total score less than 0. Note that these filters do not explicitly evaluate the constraints mentioned above. For the final round of modeling, approximately 1693 models were generated, of which 98 passed the filters. This represents approximately 8300 hours on a 2.3 GHz processor.

Results

To probe the hypothesis that the structure of the Cdc34-ubiquitin interaction is important for ubiquitin catalysis, we used the UBQ_E2_thioester protocol detailed above

to generate models of the Cdc34-ubiquitin thioester-linked state. These models were constrained to agree with pre-existing mutational data and thus required the interaction of the ubiquitin arginine 42-Cdc34 glutamate 133 pair, as well as burial of ubiquitin isoleucine 44 into the protein-protein interface. The top twenty models by total score generated by this data set collectively strongly suggested Cdc34 serine 129 as a possible interacting partner for ubiquitin isoleucine 44. Figure 6.5 shows one such model.

To assess the validity of this structural model of the ubiquitin-Cdc34 interaction and confirm the mutation-based hypothesis that ubiquitin isoleucine 44 is at the interface, we used these computational models to predict a Cdc34 mutation that could rescue the ubiquitin discharge defect seen in ubiquitin I44A. Specifically, we used Rosetta's fixbb fixed-backbone design application [17] to change the sequence of our Cdc34-ubiquitin models to include the I44A mutation, and then predict what mutation on the surface of Cdc34 would best recreate the wild-type interaction. Rosetta chose to replace Cdc34 serine 129 with leucine, offering a small-to-large mutation to complement the large-to-small mutation I44A. The packing of these residues in the wild-type and I44A/S129L interfaces are detailed in Figure 6.6.

The Cdc34 S129L mutation in isolation produces a ubiquitin discharge defect similar to that caused by ubiquitin I44A [1]. This suggests that Cdc34 serine 129 is indeed in the interface. Additionally, I44A and S129L serve as rescue mutations for each other: when both mutations are present, ubiquitin discharge proceeds with almost wild-type behavior. The defect in S129L, and its ability to rescue I44A, suggest that our Cdc34-ubiquitin model interaction is biologically relevant. The mutational data also

further disproves the hypothesis that the structure of the ubiquitin-E2 interface is unimportant for ubiquitin transfer.

Conclusions

In this work, we used a novel Rosetta3-based tethered docking protocol to model the Cdc34-ubiquitin thioester-linked state and predict a mutation that rescued an experimentally-observed mutation-derived catalytic defect. Our model structure strongly resembles the pre-existing Ubc1-ubiquitin model in PDB 1FXT [5]; this result is trivial because we used 1FXT to guide the selection of the constraints which bias our modeling. Other modeling constraints were derived from experimental mutagenesis results. The multiple mutations presented in this work [1], most of which induce ubiquitin discharge defects, imply that the details of the Cdc34-ubiquitin thioester-linked catalytic conformation are important to its activity.

Rosetta3's modularity and easy-to-develop structure was critical for the production of the UBQ_E2_thioester application used in this work. As in Chapter 3, the drop-in nature of the molecular mechanics code [15] Rosetta uses to evaluate chemical moieties not usually seen in proteins made it simple to direct Rosetta to sample a never-before-modeled (by Rosetta) thioester bond. The placement of chemical moieties into database-organized parameter files [18] which generate `ResidueType` classes [11] readily exposed the necessary program data to be modified to parameterize the thioester bond. Most of the `Mover` classes used inside UBQ_E2_thioester's Monte Carlo sampling were already available when development began; the exception is `TorsionDOFMover` which was created to sample the thioester-related torsions.

As was seen in Chapter 5, Rosetta applications written for specific or specialized modeling purposes may find further use in other experiments. We have created modified versions of the UBQ_E2_thioester code to generate similar tethered-docking models of ubiquitin linked to the small GTPase Ras [19]. This modeling uses either a native isopeptide amide bond between ubiquitin's C-terminus and Ras surface lysines, or an experimentally-induced disulfide bond generated by mutations to cysteine at the same positions. The modeling helps explain the signaling behavior of monoubiquitinated Ras, and indicates that the experimental disulfide complex is a good substitute for the native isopeptide linkage (data not shown).

Figure 6.1: Disagreement about E2-ubiquitin interface models
This figure, rendered in PyMOL [20], captures the disagreement in the literature about what E2-ubiquitin interfaces look like. These PDB [21] structures represent a variety of E2 proteins, structural analysis techniques, and chemical linkages, so some variation is to be expected. Nevertheless, there is essentially no consensus as to how ubiquitin interacts with an E2 to which it is chemically bound, giving rise to the hypothesis that said interaction is unimportant. The structures are aligned on ubiquitin, which is in forest green in the background of the image. Presented here (clockwise from top) are structures from PDBs 3A33 [10] (salmon), 3JVZ [8] (blue), 1FXT [5] (cyan), 2KJH [9] (yellow), and 2GMI [7] (magenta).

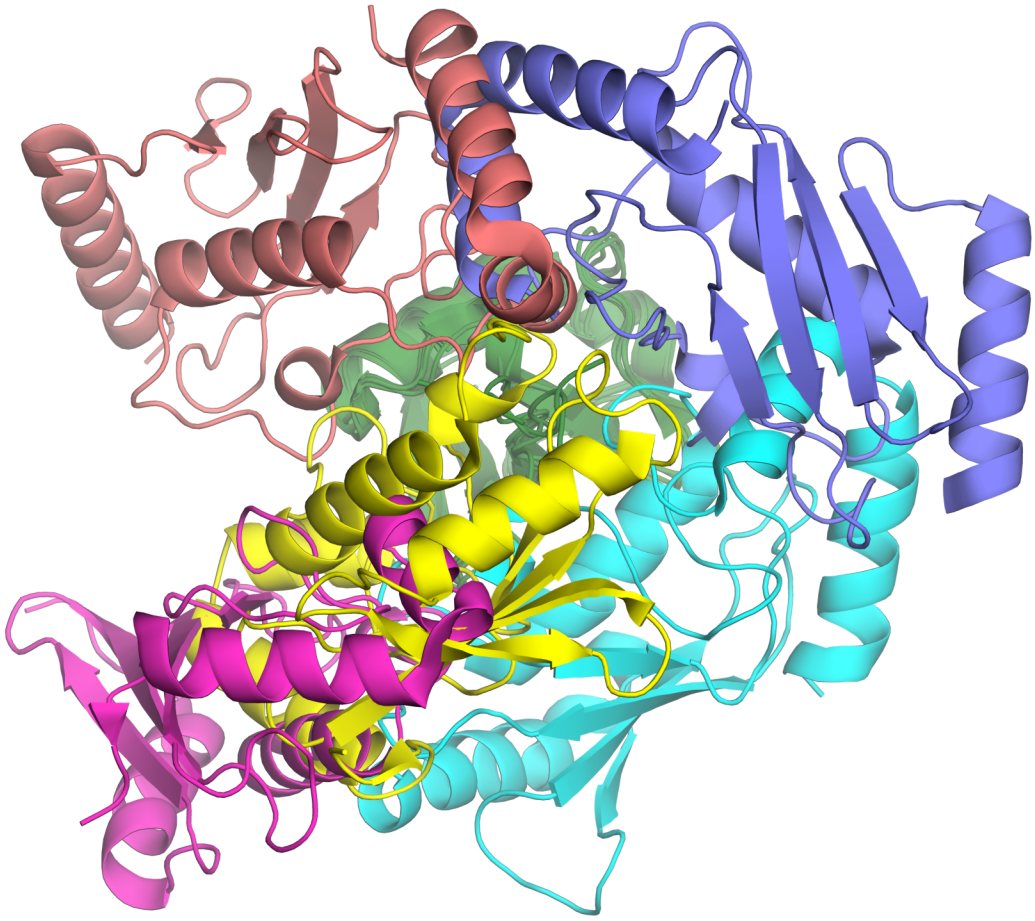


Figure 6.2: Modeled torsions near the thioester bond
This figure demonstrates the thioester linkage between ubiquitin C-terminal glycine 76 (Ubq G76, green at left) and the catalytic E2 cysteine 93 on Cdc34 (Cdc34 C93, blue at right). Torsions sampled by the UBQ_E2_thioester protocol are marked with rotation arrows. From the right, these are χ_1 and χ_2 of C93, the thioester bond, and φ and ψ of ubiquitin G76, respectively. Not visible in the figure, but also sampled, are φ and ψ for the penultimate and antepenultimate G75 and R74 residues of ubiquitin. Sidechains of either protein near the interface are allowed to repack as well.

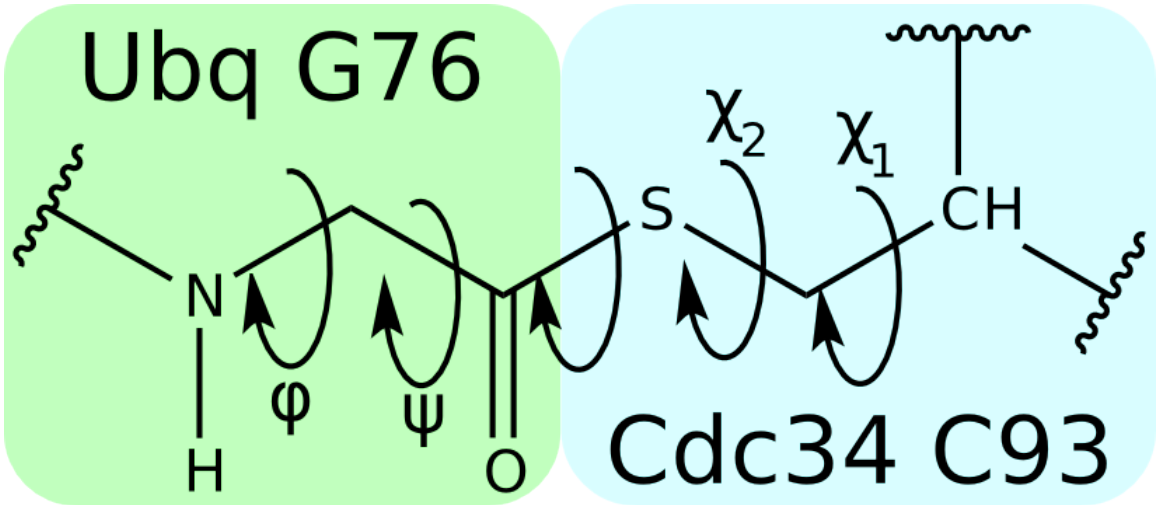


Figure 6.3: UBQ_E2_thioester protocol flow
This flowchart details control flow for the UBQ_E2_thioester protocol. Two PDBs, representing the E2 Cdc34 and ubiquitin, are assembled with the help of thioester parameters to create a starting state tethered with the correct chemical linkage. The protocol then enters 20,000 Monte Carlo cycles. In most cycles, the protocol changes a linker torsion (those detailed in Figure 6.2) with `TorsionDOFMover`, `SmallMover`, or `ShearMover`, runs a fast single-pass rotamer relaxation scheme (`EnergyCutRotamerTrialsMover`), and if the structure energy has not increased unduly (meaning, if there is no large clash due to the proteins colliding, as measured by `JumpOutMover`) the protocol minimizes the G75 and R74 backbone torsions with `MinMover`. Thioester-related torsions cannot be minimized due to the absence of thioester parameters in the standard scorefunction; instead the Monte Carlo proposals for thioester-related torsions are dampened to only propose low-energy torsions. On every hundredth cycle, the protocol performs an intensive rotamer repacking operation at the interface and minimizes those sidechains (plus the G75 and R74 backbone torsions). Completed structures are subject to filters for score and SASA; only those that pass the filters are output.

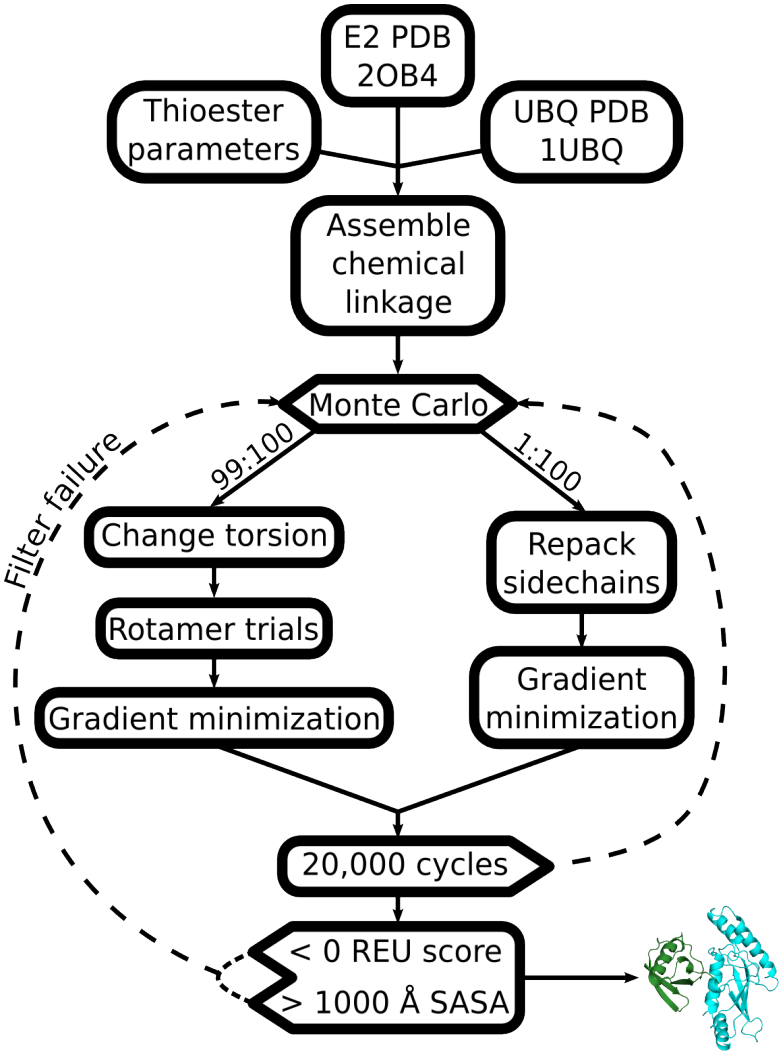


Figure 6.4: A comparison of constrained and unconstrained models
In this figure, E2 Cdc34 (from 2OB4 [13]) is in cyan, and E2 Ubc1 (from 1FXT [5]) is in blue. 1FXT's ubiquitin is white, and other colors indicate model ubiquitins from our cdc34-ubiquitin models. At left in panel A, the top 5 models from the final model population are shown along with 1FXT, demonstrating the small spread in model placement and their similarity to 1FXT. On the right in panel B are the top 5 scoring models from an earlier version of the protocol which lacked experimentally-derived constraints (but did still include the thioester tethered docking element). These models show much more spread in conformation. There are models (none shown here) which adopt a 1FXT-like conformation in this unconstrained population.

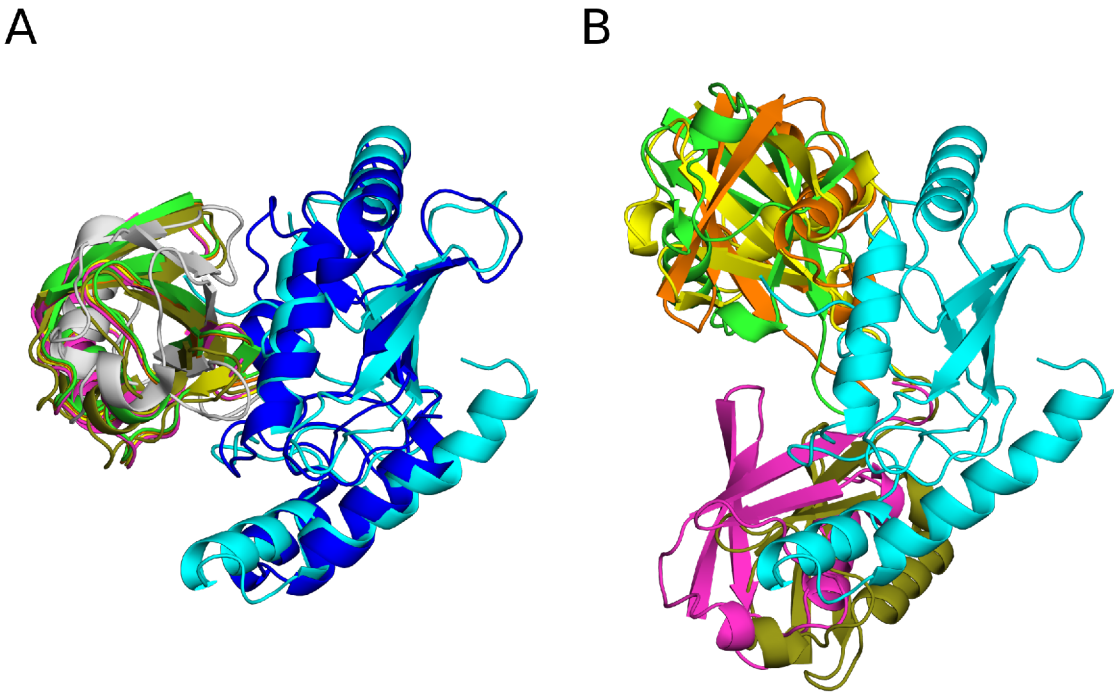


Figure 6.5: Model interface suggests I44 interactions
This figure zooms in on a few relevant details in the ubiquitin-Cdc34 interface. Ubiquitin is in green at left, and Cdc34 in cyan on the right. The R42/E113 charge pair, suggested by a charge-swap experiment, is labeled. I44's primary interacting partner was found to be S129 on Cdc34. In the lower left is the thioester bond; it is behind the interface.

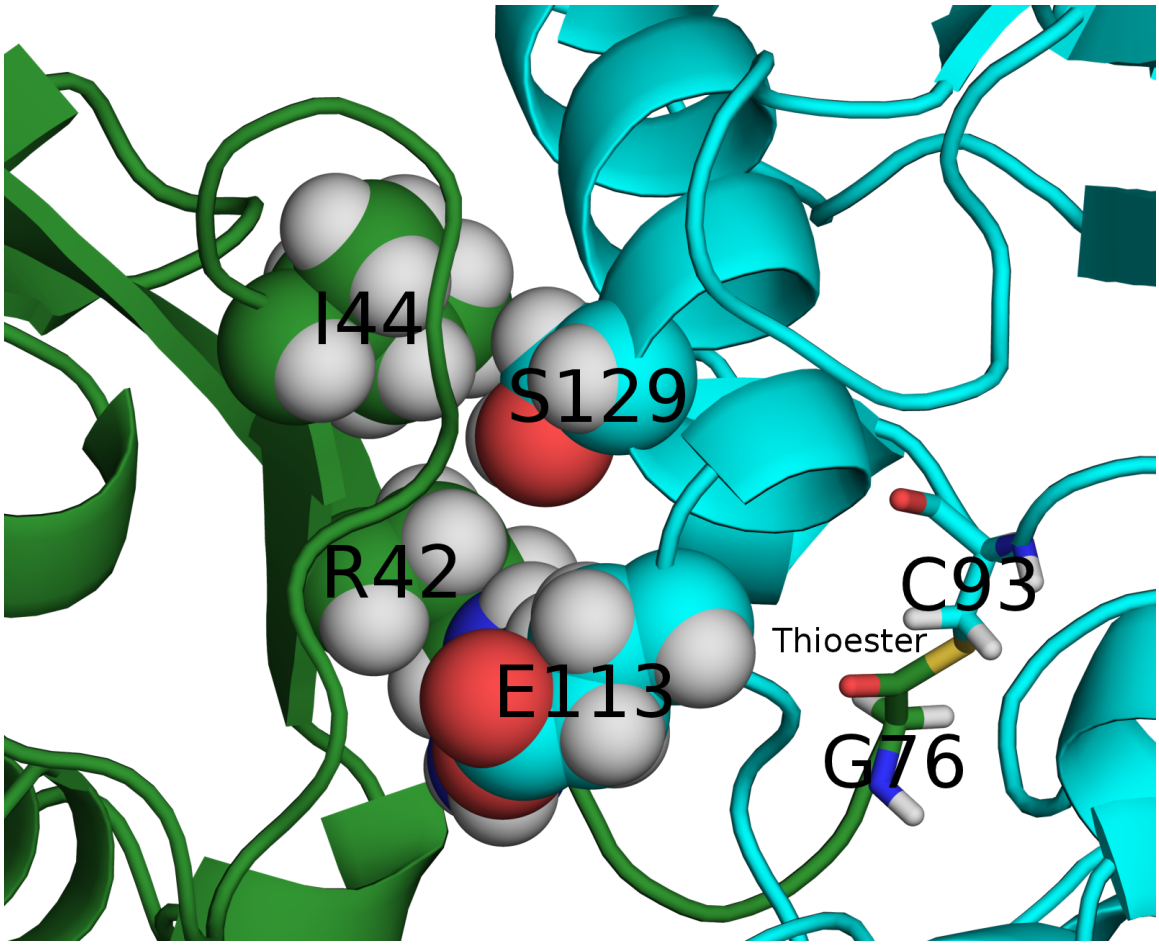
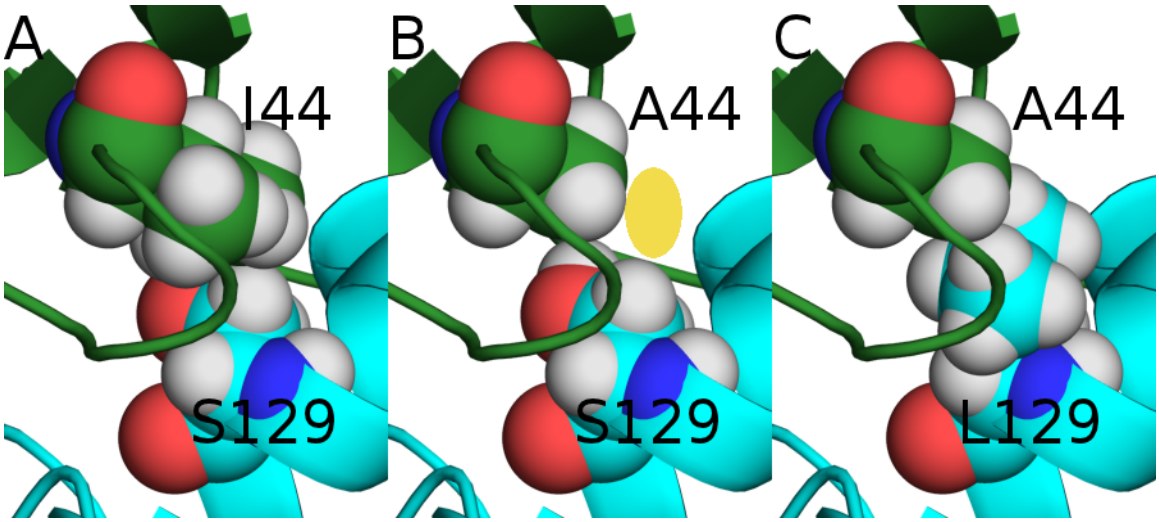


Figure 6.6: How S129L rescues I44A
These panels show how ubiquitin I44A (green) can be rescued by Cdc34 S129L (cyan). Panel A shows a model of the wild-type interface. In panel B, the gap introduced by I44A is highlighted in yellow. In panel C, S129L fills the gap to rescue I44A.



References

1. Saha A, Lewis S, Kleiger G, Kuhlman B, Deshaies RJ. (2011) Essential role for ubiquitin-ubiquitin-conjugating enzyme interaction in ubiquitin discharge from Cdc34 to substrate. Mol Cell 42(1): 75-83. 10.1016/j.molcel.2011.03.016.

2. Chau V, Tobias JW, Bachmair A, Marriott D, Ecker DJ, et al. (1989) A multiubiquitin chain is confined to specific lysine in a targeted short-lived protein. Science 243(4898): 1576-1583.

3. Thrower JS, Hoffman L, Rechsteiner M, Pickart CM. (2000) Recognition of the polyubiquitin proteolytic signal. EMBO J 19(1): 94-102. 10.1093/emboj/19.1.94.

4. Behrends C, Harper JW. (2011) Constructing and decoding unconventional ubiquitin chains. Nat Struct Mol Biol 18(5): 520-528. 10.1038/nsmb.2066.

5. Hamilton KS, Ellison MJ, Barber KR, Williams RS, Huzil JT, et al. (2001) Structure of a conjugating enzyme-ubiquitin thiolester intermediate reveals a novel role for the ubiquitin tail. Structure 9(10): 897-904.

6. Brzovic PS, Lissounov A, Christensen DE, Hoyt DW, Klevit RE. (2006) A UbcH5/ubiquitin noncovalent complex is required for processive BRCA1-directed ubiquitination. Mol Cell 21(6): 873-880. 10.1016/j.molcel.2006.02.008.

7. Eddins MJ, Carlile CM, Gomez KM, Pickart CM, Wolberger C. (2006) Mms2-Ubc13 covalently bound to ubiquitin reveals the structural basis of linkage-specific polyubiquitin chain formation. Nat Struct Mol Biol 13(10): 915-920. 10.1038/nsmb1148.

8. Kamadurai HB, Souphron J, Scott DC, Duda DM, Miller DJ, et al. (2009) Insights into ubiquitin transfer cascades from a structure of a UbcH5B approximately ubiquitin-HECT(NEDD4L) complex. Mol Cell 36(6): 1095-1102. 10.1016/j.molcel.2009.11.010.

9. Serniwka SA, Shaw GS. (2009) The structure of the UbcH8-ubiquitin complex shows a unique ubiquitin interaction site. Biochemistry 48(51): 12169-12179. 10.1021/bi901686j.

10. Sakata E, Satoh T, Yamamoto S, Yamaguchi Y, Yagi-Utsumi M, et al. (2010) Crystal structure of UbcH5b~ubiquitin intermediate: Insight into the formation of the self-assembled E2~Ub conjugates. Structure 18(1): 138-147. 10.1016/j.str.2009.11.007.

11. Leaver-Fay A, Tyka M, Lewis SM, Lange OF, Thompson J, et al. (2011) ROSETTA3: An object-oriented software suite for the simulation and design of macromolecules. Methods Enzymol 487: 545-574. 10.1016/B978-0-12-381270-4.00019-6.

12. Vijay-Kumar S, Bugg CE, Cook WJ. (1987) Structure of ubiquitin refined at 1.8 A resolution. J Mol Biol 194(3): 531-544.

13. Ceccarelli DF, Tang X, Pelletier B, Orlicky S, Xie W, et al. (2011) An allosteric inhibitor of the human Cdc34 ubiquitin-conjugating enzyme. Cell 145(7): 1075-1087. 10.1016/j.cell.2011.05.039.

14. Rohl CA, Strauss CEM, Misura KMS, Baker D. (2004) Protein structure prediction using rosetta. Methods Enzymol 383: 66-+.

15. Renfrew PD, Choi EJ, Bonneau R, Kuhlman B,. (2012) Incorporation of noncanonical amino acids into rosetta and use in computational protein-peptide interface design. PLoS ONE 7(3): e32637. Available: http://dx.doi.org/10.1371%2Fjournal.pone.0032637 via the Internet.

16. Yang W, Drueckhammer DG. (2001) Understanding the relative acyl-transfer reactivity of oxoesters and thioesters: Computational analysis of transition state delocalization effects. J Am Chem Soc 123(44): 11004-11009. 10.1021/ja010726a. Available: http://dx.doi.org/10.1021/ja010726a via the Internet.

17. Kuhlman B, Dantas G, Ireton GC, Varani G, Stoddard BL, et al. (2003) Design of a novel globular protein fold with atomic-level accuracy. Science 302(5649): 1364-1368.

18. Davis IW, Baker D. (2009) RosettaLigand docking with full ligand and receptor flexibility. J Mol Biol 385(2): 381-392. 10.1016/j.jmb.2008.11.010.

19. Baker R, Lewis SM, Wilkerson EM, Sasaki AT, Cantley LC, et al. (Submitted.) Site-specific monoubiquitination activates ras by impeding GTPase activating protein function. Nature Structural & Molecular Biology .

20. Schrödinger L. (2002) The PyMOL molecular graphics system. 1.4.1. Available: http://www.pymol.org; via the Internet.

21. Berman HM, Westbrook J, Feng Z, Gilliland G, Bhat TN, et al. (2000) The protein data bank. Nucleic Acids Res 28(1): 235-242.