

# FUNCTION-DRIVEN APPROACHES TO THE DESIGN OF OPTOGENETIC TOOLS

Frank David Teets III

A dissertation submitted to the faculty at the University of North Carolina at Chapel Hill in partial fulfillment of the requirements for the degree of Ph.D. in the Department of Genetics in the School of Medicine.

Chapel Hill  
2019

Approved by:

Brian Kuhlman

Klaus Hahn

Jeremy Purvis

Saskia Neher

Charlie Carter

© 2019  
Frank David Teets III  
ALL RIGHTS RESERVED

## ABSTRACT

### Frank David Teets III: FUNCTION-DRIVEN APPROACHES TO THE DESIGN OF OPTOGENETIC TOOLS

(Under the direction of Brian Kuhlman and Klaus Hahn)

Proteins play a wide variety of roles in biology despite being produced from a small set of common subunits; this commonality can be exploited to understand the dynamics by which proteins fold into structures and perform their manifold functions and, subsequently, design new proteins for use both in research and as nanoscale machines in industry. While this design process has classically involved residue-level redesign of existing protein backbones and, more recently, the *de novo* design of backbones according to geometrical parameters, the increasing complexity of optogenetic photosystems, biosensors, and other mechanisms for making use of proteins with specific functions has established a need for a design protocol that can reconcile their various structural exigencies with the function-specific elements of as wide an array of proteins as possible in order to make best use of them.

Requirement-driven design eschews specific structural templates in favor of general requirements dependent on the intended function of the design, and so can exploit the vastness of protein structural space in finding solutions to increasingly complex design problems. Here, we present three new advances in the requirement-driven design of proteins as diagnostic tools, including a more general photosystem for the

direct optogenetic control of protein-protein interactions, a series of algorithmic improvements to the leading implementation of requirement-driven design in the Rosetta macromolecular design software suite, and a new version of that algorithm capable of performing requirement-driven backbone design and residue-level backbone optimization simultaneously. These technologies collectively represent a significant improvement in our ability to control the activity of proteins with a wide variety of control schemes and produce functional proteins for arbitrary requirement sets more generally.

## PREFACE

Chapters 2 and 4 are previously published works used with permission from the publishers. Chapter 2 was previously accepted for publication in *Nature Chemical Biology* (Orrin J. Stone et al, 2019). Orrin Stone and Hui Wang conceived of the Z-Lock system, and Orrin Stone designed and tested the cofilin switches, while Neha Pankow created the aTAT switches. I computationally verified the functionality of the cofilin and aTAT switches and elucidated the mechanistic underpinnings of the aTAT switch system with the advice and assistance of Brian Kuhlman, as well as generating the structural figures for the paper.

Chapter 3 has been submitted for publication to the *Journal of Molecular Biology* as “A computational protocol for inhibiting protein binding reactions with a light sensitive protein dimer” (Teets FD et al, 2019). I created the linker design protocol with advice and assistance from Brian Kuhlman, while Takashi Watanabe tested the Wasp CBD switch I designed with advice from Klaus Hahn.

Chapter 4 was originally published in the *Journal of Chemical Informatics and Modeling* as a joint first-author publication with Sharon Guffy (Guffy, S., Teets, F. Langlois, M. and Kuhlman B.(2018) *J. Chem Inf. Model.*) Sharon Guffy, Hermine Langlois, and I all contributed to the protocol outlined in the paper; I created the central data structures for the new version of the algorithm, while Sharon and Hermine implemented the chimerization and scoring processes. I was responsible for the interface design algorithms, while Sharon and Hermine created the ligand-

binding design code. Sharon Guffy and I jointly wrote the manuscript and produced the figures, with Sharon writing the “Ligand Binding” section and me writing the portion on interface design. Brian Kuhlman and Sharon Guffy wrote the introduction. I performed benchmarking and optimization and produced the primary figures for the paper. Brian Kuhlman, Sharon Guffy, and I all edited the paper. Brian Kuhlman provided funding for the research.

## TABLE OF CONTENTS

LIST OF TABLES.....	xi
LIST OF FIGURES.....	xii
LIST OF ABBREVIATIONS.....	xvii
CHAPTER 1: INTRODUCTION TO FUNCTION-DRIVEN DESIGN .....	1
1.1 Proteins as Molecular Machines.....	1
1.2 The Promise of Protein Design.....	3
1.3 <i>De novo</i> Design.....	4
1.4 Optogenetics as a Motivating Case.....	5
1.5 SEWING.....	7
1.6 Ongoing Challenges.....	8
1.7 Goals of Research.....	9
CHAPTER 2: OPTOGENETIC CONTROL OF COFILIN AND ATAT IN LIVING CELLS USING Z-LOCK.....	11
2.1 Introduction.....	11
2.2 Results.....	14
2.3 Discussion.....	23
2.4 Online Methods.....	26
2.5 Figures and Tables.....	38
CHAPTER 3: A COMPUTATIONAL PROTOCOL FOR INHIBITING PROTEIN.....	71
BINDING REACTIONS WITH A LIGHT SENSITIVE PROTEIN DIMER	

3.1 Introduction.....	71
3.2 Results.....	73
3.3 Discussion.....	82
3.4 Materials and Methods.....	84
3.5 Figures and Tables.....	89
CHAPTER 4: PROTOCOLS FOR REQUIREMENT-DRIVEN DESIGN IN THE	
ROSETTA MODELING PROGRAM.....	107
4.1 Introduction.....	107
4.2 Walkthrough and General Guidelines.....	109
4.3 Available Protocols.....	110
4.4 Recommended Options.....	111
4.5 Command Line Options.....	112
4.6 Refinement, Design, and Filtering.....	113
4.7 Input Files for SEWING.....	114
4.7.1 Segment Files.....	114
4.7.2 Motif score databases.....	116
4.7.3 Edge Files.....	116
4.7.4 Scoring .....	116
4.8 Requirements .....	117
4.9 Applications.....	118
4.9.1 Expanding Protein Interfaces.....	118
4.9.2 Adding other proteins to the simulation.....	120
4.10 Guidelines for use.....	121



4.11 Sample Results.....	121
4.12 Ligand Binding.....	122
4.13 Conclusions.....	127
4.14 Supplementary Information.....	128
4.14.1 Segment File Generation.....	128
4.14.2 Edge File Generation.....	129
4.15 Figures and Tables.....	130
CHAPTER 5: ADVANCEMENTS IN FUNCTION-DRIVEN PROTEIN DESIGN.....	148
5.1 Introduction.....	148
5.2 Walkthrough and General Guidelines.....	151
5.3 Vital Residues and Terminus Control: The AddFlankingVirtualResiduesMover.....	151
5.4 Secondary Structural Envelopes: The SewAnythingAddMover.....	152
5.5 Subsets and Scoring: The PoseCompatibleMotifScorer.....	153
5.6 Required Input Files and How to Generate Them.....	154
5.7 Optimal Operational Contexts.....	154
5.8 Results.....	155
5.9 Conclusions.....	156
5.10 Figures and Tables.....	157
CHAPTER 6: CONCLUSIONS.....	321
6.1 The Advantages of Function-Driven Protein Design.....	321
6.2 Present Inadequacies in Function-Driven Design.....	322

6.2a Beta sheets.....	322
6.2b Designing for diversity.....	322
6.2c Intercalated remedial design.....	323
6.3 Final Remarks.....	323
REFERENCES.....	325

## LIST OF TABLES

Supplementary Table 2.1: Change in Gibbs energy of binding for Zdk2 mutants....	48
Supplementary Table 2.2: Plasmids.....	49
Table 4.1. Score functions available during SEWING.....	133
Table 4.2. Requirements available during SEWING.....	134

## LIST OF FIGURES

Figure 2.1. Design of Z-lock cofilin.....	38
Figure 2.2. Optimization of designs based on Zdk1 and Zdk2.....	40
Figure 2.3. Effect of Z-lock cofilin activation on leading edge protrusions and invadopodium formation in tumor cells.....	42
Figure 2.4. Z-lock $\alpha$ TAT.....	46
Figure 2.5 Validation of actin co-sedimentation assay and testing of Zdk-cofilin linkers.....	51
Figure 2.6 Original images of blots and stain-free gels used for Supplementary Fig. 2.1b and 2.1c.....	54
Figure 2.7. Original images of blots and stain-free gels used for Figure 2.5.....	56
Figure 2.8. Original images of blots used for Fig. 2.2c.....	57
Figure 2.9. Original images of blots used for Fig. 2.2a.....	58
Figure 2.10. Original images of blots used for Fig. 2.2g.....	61
Figure 2.11. Actin co-sedimentation assay used to determine the affinity of Z- lock cofilin and F-actin.....	62
Figure 2.12. Rosetta prediction of Z-lock cofilin structure in the dark and lit state...	64
Figure 2.13. F-actin severing by Z-lock cofilin in-vitro.....	65
Figure 2.14. Z-lock cofilin reversibly translocates to the cell edge upon activation..	66
Figure 2.15. Expression of Z-lock cofilin doesn't perturb F-actin level in MTLn3 Cells.....	67
Figure 2.16. Reversibility of Z-lock cofilin.....	69
Figure 2.17. Original images of blots and stain-free gels used for Fig. 2.4b.....	70

Figure 3.1 Inhibiting protein binding interactions with the light sensitive protein pair, LOV-zDK, and linker design.....	89
Figure 3.2. Modeling linkers for z-Lock cofilin.....	91
Figure 3.3 Modeling linkers for z-Lock $\alpha$ TAT.....	93
Figure 3.4. Computational design and experimental testing of a light-sensitive binder for Cdc42.....	95
Figure 3.5: A sample Rosettascripts script for sampling the space accessible to the optogenetic components in a constrained torsion perturbation run.....	97
Figure 3.6. A sample fold tree file for use with the above script.....	99
Figure 3.7. A sample constraint file for use with the above script.....	100
Figure 3.8. A sample script file for use with the ConcatenatePosesMover, an application for assembling protein domains with flexible linkers.....	101
Figure 3.9. A sample RosettaScript for using the ConcatenatePosesMover.....	102
Figure 3.10 A sample flags file for running the ConcatenatePosesMover RosettaScript.....	103
Figure 3.11. A sample flags file for running the structure prediction RosettaScript.....	104
Figure 3.12 A sample command line for running the ConcatenatePosesMover RosettaScript.....	105
Figure 3.13 A sample command line for running the structure prediction RosettaScript.....	106
Figure 4.1. A sample SEWING output, or Assembly.....	130
Figure 4.2. A flowchart showing the general SEWING workflow.....	131

Figure 4.3. Interface expansion with SEWING.....	132
Figure 4.4: Sample flags file used to run SEWING.....	135
Figure 4.5: Sample RosettaScript used to refine and filter backbones that were designed using the SEWING protocol.....	136
Figure 4.6: Sample RosettaScript for expanding the interface between a protein- binding peptide and its binding partner using SEWING Append.....	141
Figure 4.7: Procedure for completing a ligand's coordination environment using LigandSEWING.....	143
Figure 4.8: Sample RosettaScript for using LigandSewing to complete the coordination of a tetrahedral zinc atom.....	145
Figure 4.9: Sample of a ligand coordination file (for use with LigandSEWING).....	147
Figure 5.1 A comparison of the performance of assembly SEWING and SewAnything in extending a two-helix hairpin into a five-helix bundle.....	157
Figure 5.2 A comparison of the performance of assembly SEWING and SewAnything in generating an affinity reagent from the vav1 AID.....	158
Figure 5.3 Selected designs from the small_shear_gsa SewAnything run used in Figure 5.2.....	159
Figure 5.4: The RosettaScript for running assembly SEWING in the 1-cycle case.....	160
Figure 5.5: The flags file for running assembly SEWING.....	162
Figure 5.6: The RosettaScript for running assembly SEWING in the 10-cycle case .....	164

Figure 5.7: The RosettaScript for running assembly SEWING in the 100-cycle case	166
Figure 5.8: The RosettaScript for running assembly SEWING in the 1000-cycle case	167
Figure 5.9: The RosettaScript for running pose SEWING in the 1-to-1 case	170
Figure 5.10: The RosettaScript for running pose SEWING in the 2-to-1 case	175
Figure 5.11: The RosettaScript for running pose SEWING in the 4-to-1 case	180
Figure 5.12: The RosettaScript for running pose SEWING in the 10-to-1 case	185
Figure 5.13: The RosettaScript for running pose SEWING with small and shear moves	190
Figure 5.14: The RosettaScript for running pose SEWING with small and shear moves under the control of the MotifScorer	196
Figure 5.15: The RosettaScript for running assembly SEWING to generate a vav1 AID in the 100-cycle case	202
Figure 5.16: The RosettaScript for running assembly SEWING to generate a vav1 AID in the 1000-cycle case	214
Figure 5.17: The RosettaScript for running assembly SEWING to generate a vav1 AID in the 10000-cycle case	226
Figure 5.18: The RosettaScript for running SewAnything to generate a vav1 affinity reagent in the 1-to-1 case	238
Figure 5.19: The RosettaScript for running SewAnything to generate a vav1 affinity reagent in the 2-to-1 case	250

Figure 5.20: The RosettaScript for running SewAnything to generate a vav1 affinity reagent in the 4-to-1 case.....	263
Figure 5.21: The RosettaScript for running SewAnything to generate a vav1 affinity reagent in the 10-to-1 case.....	276
Figure 5.22: The RosettaScript for running SewAnything to generate a vav1 affinity reagent with intercalated small and shear moves.....	289
Figure 5.23: The RosettaScript for running SewAnything to generate a vav1 affinity reagent with intercalated small and shear moves and an upweighted partner score.....	305



## LIST OF ABBREVIATIONS

$\alpha$ TAT	alpha-tubulin acetyl transferase
CALI	Chromophore-Assisted Laser Inactivation
CBD	Cdc42 Binding Domain
DN $\alpha$ TAT	dominant negative alpha-tubulin acetyl transferase
DSSP	Dynamic Secondary Structure Prediction
EGF	Epidermal Growth Factor
elNemo	elastic network model
HDAC6	histone deacetylase 6
LOV	Light Oxygen Voltage
P	Pellet
POI	Protein of Interest
S	Supernatant
SEWING	Structural Extension With Native-Fragment Graphs
SIRT2	sirtuin type 2

XML      eXtensible Markup Language

## CHAPTER 1: INTRODUCTION TO FUNCTION-DRIVEN DESIGN

### 1.1 Proteins as Molecular Machines

Proteins play a wide variety of roles in the cell; enzymes directly or indirectly mediate all of biochemistry, (Agarwal, 2006; Martínez Cuesta, Cuesta, Rahman, Furnham, & Thornton, 2015), structural proteins physically regulate cellular motion and morphology (Jockusch, 2017; Jreij, 2016; Rodgers & Fanning, 2011), transport proteins regulate the contents of cellular compartments (Aldridge, Cain, & Robinson, 2009; Dalbey, Chen, & Wiedmann, 2002; Görlich & Jäkel, 2002). All of this activity is controlled by a complex network of regulatory proteins, including transcription factors (T. R. Hughes, 2011; Latchman, 1995; Locker, 2012), G proteins and their associated receptors (Berstein, 2019; Iyengar, 2012), kinases (Hodgkin, 2001; Zafar & Zerr, 2013) and other more specialized signaling factors. It is not an exaggeration to say that any biological phenomenon of clinical or scientific interest is either a direct product of or indirectly regulated by proteins and the interactions between proteins, and yet this diverse class of biomolecules consists, in a structural sense, of initially linear arrangements predominantly composed of only twenty different chemical subunits polymerized through a single process common to all known organisms, after which these linear chains adopt an ensemble of three-dimensional structures through a process known to be spontaneous (Anfinsen, Haber, Sela, &

White, 1961) and governed primarily by the interactions between the protein's primary sequence and its environment (Anfinsen, 1973). It is, therefore, possible to explore the relationship between protein sequence, structure, and function by comparing proteins similar in one dimension and evaluating their differences in another; the set of functional proteins in extant organisms is a rich dataset for this purpose, having been continually expanding since its origin, with countless examples of proteins diverging in sequence, structure and function from a common origin across organisms (Fitch, 1978; Martínez Cuesta et al., 2015; Patthy, 2009; Prowse & Byrne, 2012; Saier, 2003; Smith, Mularz, & Hecht, 2015; Stevens, n.d.; Z. Yang, 2014) and to proceed to synthesize from these differences a set of general principles for predicting the structure of a polypeptide from its sequence and its function from its structure. The first of these processes is protein structure prediction, presently conducted through either quantum mechanical modeling or force-field-based approaches in which structures are given scalar scores based on how their relative atomic positions match a set of biophysical criteria typically derived from protein crystal structures and perturbing that structure in an attempt to minimize that score (Bazzoli, Kelow, & Karanicolas, 2015; Rubenstein, Blacklock, Nguyen, Case, & Khare, 2018). Protein structure prediction algorithms have repeatedly demonstrated the ability to accurately predict the structure of a protein from its primary sequence (Bonneau, Ruczinski, Tsai, & Baker, 2002; Bonneau, Strauss, et al., 2002; Bonneau et al., 2001; Bradley et al., 2003; Burguete, Fenn, Brunger, & Pfeffer, 2008; Lesk, Lo Conte, & Hubbard, 2001; Ovchinnikov, Park, Kim, DiMaio, &

Baker, 2018; Rohl, Strauss, Misura, & Baker, 2004), while homology modeling can similarly predict certain types of protein function from structure, making it possible to associate protein sequence, structure, and function in a predictively useful way.

## 1.2 The Promise of Protein Design

If it is possible to predict the structure of a protein from its primary sequence, it follows that predictions of similar confidence may be made for similar sequences regardless of whether they actually exist, and therefore to design proteins by specifying a structure and evaluating different sequences based on the same scoring mechanisms used in protein structure prediction. This is the basis of the foundation of the classical approach to protein design, in which a specific structure undergoes repeated cycles of mutation coupled with scoring and backbone motion; mutations which improve the score are kept and those which worsen the score are discarded until, in the ideal case, the best-scoring sequence represents the sequence most likely to adopt that fold (Loshbaugh & Kortemme, 2019). While in the default case this would be expected to improve stability at the expense of function, it is possible to modify the score function or the simulation conditions to adjust the functionality associated with a given fold; for example, mutations may be made to protein-protein interfaces to improve or attenuate their affinity (Davey & Chica, 2012, 2014; Leaver-Fay, Jacak, Benjamin Stranges, & Kuhlman, 2011). While the confidence of the predictions is correlated with the similarity to a known structure, larger structural changes can be accommodated by linking protein domains together and engineering the interfaces between them, or by grafting parts of one protein

structure into another (Silva, Correia, & Procko, 2016). Protein design can produce affinity reagents to perturb regulatory networks (Babalhavaeji & Woolley, 2018; Boschek et al., 2009; Speltz, Brown, Hajare, Schlieker, & Regan, 2015), improve the catalytic rates or specificity of enzymes for use in synthetic biology (Mak & Siegel, 2014; Nanda & Koder, 2010; Strohmeier, Pichler, May, & Gruber-Khadjawi, 2011), or stabilize portions of antigenic proteins for use in immunotherapy (Golob-Urbanc, Rajčević, Strmšek, & Jerala, 2019; Silva et al., 2019), but classically it is limited to modifying existing protein structures, with larger changes imposed externally prior to residue-level optimization to ameliorate the problem of the design becoming trapped in a local minimum. The individual changes evaluated by the design protocol must necessarily be small to provide high resolution, but folding an entire backbone one torsion perturbation at a time is both computationally intensive and susceptible to noise, as forces that would ordinarily be acting on all parts of the protein simultaneously are effectively simulated piecemeal, and many of the unfolded states of a given protein remain functionally degenerate regardless of their geometrical proximity to the folded state until the relevant residues are close enough to interact.

### 1.3 *De novo* Design

*De novo* design is a catchall term for several protocols intended to avoid the limited scope of residue-level optimization by providing a viable protein backbone as an input to the design process. The primary challenge to designing a protein backbone ex nihilo is the immensity of the search space involved, and so *de novo* design is frequently restricted to a specific fold, either directly according to user

specification (Kuhlman et al., 2003; Liang et al., 2009) or through a set of geometric parameters from which the backbone torsions of each residue may be derived (Lombardi, Pirro, Maglio, Chino, & DeGrado, 2019; Murphy et al., 2015; Rhys et al., 2019). While these protocols can produce protein folds not found in nature, they still produce specific protein folds directly, which may not be relevant to the intended function of the protein. While specific functional regions of proteins are specific, including the active sites of enzymes and the binding interfaces of regulatory proteins, the remainder of the protein need only stabilize those functional moieties in a particular conformation, which requires only that the backbone position the necessary residues within range in a stable arrangement, and further structure needs only stabilize that. While *de novo* design is a powerful tool for converting structure to sequence, it, like residue-level design, only implicitly links structure to function, and can fail to find optimal design solutions if those optima lie outside of the arbitrary criteria constraining their designed folds.

#### 1.4 Optogenetics as a Motivating Case

While it is possible to manually define a structure for a given function, as the desired functions of scientific tools become more complex, it becomes more difficult to postulate structures to fulfill them all simultaneously *a priori*. Optogenetics provides one example of the increasing demands on protein design to make proteins of interest compatible with increasingly complex optogenetic tools. While optogenetics refers broadly to the control of proteins with light, the nature of this control has become more spatiotemporally precise with time (L. Fenno, Yizhar, &

Deisseroth, 2011). Early destructive methods were largely irreversible, including chromophore-assisted laser inactivation (CALI) (Beermann & Jay, 1994; Liao, Roider, & Jay, 1994; Schmucker, Su, Beermann, Jäckle, & Jay, 1994); channelrhodopsins offered reversible control of cell polarization but were neither localizable to the subcellular level nor generalizable to control arbitrary proteins optically (L. Fenno et al., 2011; G. Nagel, 2002; Georg Nagel et al., 2003). More general approaches to controlling protein activity at the subcellular level involved simply attaching the protein of interest to some optically sensitive domain that would, upon illumination, undergo a structural shift that would activate some subcellular transport pathway to shuttle it into a different subcellular compartment, including the nucleus (Yumerefendi et al., 2018) or the mitochondria (H. Wang & Hahn, 2016; H. Wang et al., 2016); these transport-based methods are again limited in spatiotemporal control. Similarly, optically controlled dimerization systems, such as iLiD (Zimmerman et al., 2016) and the pMag/nMag system (Kawano, Suzuki, Furuya, & Sato, 2015), do not directly control protein activity; for that, it has typically been necessary to design a complex in which the interface of interest is sterically occluded by the optically active components (Wu et al., 2009) or allosterically inactivated in either the lit or dark states (Gasser et al., 2014; Möglich, Ayers, & Moffat, 2009; Ryu et al., 2014), which requires that the protein of interest be engineered to be both compatible with the optical components and amenable to the propagation of the optically triggered conformational change through the protein. Other control schemas exist that do not require a specific interface but do require



specific arrangements of the photosystem relative to protein of interest, including the z-Lock system described in Chapter 2, but in all cases, placing a given protein system under direct optical control requires the engineering of the same protein in multiple states, so that it is open when the photosystem is activated and inactive when in the dark state or vice versa. Creating a general set of rules for integrating this functionality into arbitrary proteins is difficult, but simply identifying the conditions under which the requirements of each state are satisfied is relatively straightforward, suggesting that an approach driven directly by those requirements may be better able to adapt existing optogenetic control schemes to new proteins.

## 1.5 SEWING

Structural Extension With Native-Fragment Graphs (SEWING) takes an approach to *de novo* design that operates in an explicitly functional paradigm in contrast to the structural specificity of other protocols; it uses the same loop of structural alteration and scoring-based comparison within a Monte Carlo loop to retain the best-scoring alterations, but the designs are scored based on the relative positions of the designed residues rather than any specific structural schema (Jacobs et al., 2016). The most fundamental form of this score, the MotifScore, uses the geometric relationship between two residues to predict the most favorable interaction two small hydrophobic residues at those positions could have; permutations of this concept restrict the residues compared to those on different chains in order to encourage interface formation or to residues far away in primary sequence space to encourage globularity. In this way, multifunctional protein tools

may be designed without more prior sequence than is necessary for the individual functions, more fully exploiting the vastness of protein structural space to find solutions to increasingly complex functional problems.

## 1.6 SEWING

While requirement-driven design is currently a powerful tool for the design of proteins based wholly on the functions required of them, its particular implementations are limited in the functions they can accommodate; the rigidity assumed of the structural starting points to enable the efficient addition of new structures means that those initial substructures must be provided in a working form, as it is not possible to induce them to fit during the design run. In practical terms, this limits the immediate usage of SEWING to the design of affinity reagents around interfaces of known structure. While score terms do exist in SEWING for designing scaffolds between two points in space in order to stabilize complexes, any attempt to build protein structure around an extant surface necessarily loses efficiency, as the rigid subunits added during the design are not an exhaustive set of all possible subunits and may not conform optimally to a provided surface; with the rest of the designed protein constructed to stabilize this imperfect fit, the entire design must subsequently shift into an optimal backbone configuration *en bloc* during the subsequent design step. Its ability to stabilize interfaces is also limited by the efficacy of residue-level interface design, and therefore the capacity to design arbitrary optogenetic inhibitors is likewise restricted by the compatibility between the extant interface of interest and the photosystem of choice; as detailed in Chapter 2,

the development of more structurally permissive photosystems reduces the dependence of optogenetics on specific structural details, and requirement-driven design can exploit this permissivity to expand the set of proteins that may be optogenetically caged if the dynamics of those photosystems are known with sufficient confidence to permit their reduction to a set of structural requirements.

## 1.7 Goals of Research

In this dissertation, I will outline several solutions to current problems in the design of proteins to serve specific functional roles. In Chapter 2, we use the z-Lock multi-component optogenetic caging scheme to control the activity of several proteins with light without designing an interface between them, and further to use protein design to determine their caging mechanism. In Chapter 3, we introduce a novel method for the design of multi-state proteins generally and z-Lock caged proteins in particular by evaluating the mutual exclusivity of different states as a function of linker geometry. In Chapter 4, we introduce a set of significant improvements to the extant SEWING protocol to improve its computational efficiency and expand its design capabilities; Chapter 5 consists of a description of a new version of the SEWING software with increased interoperability with the rest of the Rosetta design suite and concomitant increase in efficiency when directed toward design goals that are functionally orthogonal, a consequence of increased algorithmic access to the designs during the run. Finally, Chapter 6 outlines future solutions to adapting these algorithms to high-throughput experimental contexts as well as a general overview of the near-future applications of function-driven protein

design.

## CHAPTER 2: OPTOGENETIC CONTROL OF COFILIN AND ATAT IN LIVING CELLS USING Z-LOCK

### 2.1 Introduction

The cofilin pathway has been implicated in tumor cell migration during the early stages of metastasis. (Bravo-Cordero, Magalhaes, Eddy, Hodgson, & Condeelis, 2013) Rapid cofilin activation in specific cellular compartments results in the severing (Oser & Condeelis, 2009; Sidani et al., 2007; Zoncu et al., 2007) of actin filaments. Local F-actin severing by cofilin can produce either polymerization or depolymerization of F-actin depending on the location and timing of cofilin activation. (Bravo-Cordero et al., 2013; Chen & Pollard, 2013; DesMarais, Ghosh, Eddy, & Condeelis, 2005)

Conventional approaches to understand the role of cofilin in cell motility have not been sufficient to decipher important mechanistic questions; cofilin overexpression or suppression are complicated by cellular compensation, lack of kinetic information and a lack of spatial control, which impedes detection of the immediate effects of cofilin activation (H. Aizawa, 1996; Hiroyuki Aizawa et al., 2001; N. Yang et al., 1998). To address these concerns, we sought an optogenetic analog of cofilin that could be activated locally with precise kinetics.

We previously produced a photo-activatable cofilin analog by attaching a photocleavable protecting group to a constitutively active cofilin mutant. (Ghosh, 2004) Irradiation of this analog produced cofilin-dependent localized actin polymerization in living cells, but activation was irreversible and led to accumulation of active cofilin. Furthermore, production of the analog required cofilin isolation, chemical labeling and reinjection. Recently, a genetically-encoded photoactivatable cofilin based on the Lifeact peptide was produced (R. M. Hughes & Lawrence, 2014; Riedl et al., 2008). However, a subsequent study demonstrated that Lifeact significantly alters cofilin severing activity, which prevented us from using the probe to study cofilin-mediated F-actin severing during tumor cell migration (Courtemanche, Pollard, & Chen, 2016).

Due to these issues we developed Z-lock, a new optogenetic approach with potentially broad utility, and applied it to cofilin. To demonstrate the versatility of the method, we also generated a light controlled analog of the alpha tubulin acetylase  $\alpha$ TAT. Z-lock was based on the LOV2 domain of *Avena Sativa* phototropin 1, a protein of proven utility for optogenetics, and the basis of several optogenetics approaches (Dagliyan et al., 2016; Hayashi-Takagi et al., 2015; Devin Strickland et al., 2012; D. Strickland, Moffat, & Sosnick, 2008; X. Wang, Chen, & Yang, 2012; Weitzman & Hahn, 2014; Wu et al., 2009; Yi, Wang, Vilela, Danuser, & Hahn, 2014). The C terminal helix of LOV2 unwinds reversibly in response to irradiation between 400-500 nm. Unwinding is rapid ( $< 0.5$  milliseconds) and the rate of return to the folded state is tunable, with point mutations producing  $t_{1/2}$  of 1.7 – 496 seconds

(Kawano, Aono, Suzuki, & Sato, 2013; H. Wang et al., 2016; Zoltowski, Vaccaro, & Crane, 2009). Z-lock also made use of Zdk, a protein A fragment we developed previously that binds selectively to the dark conformation of LOV2 (H. Wang & Hahn, 2016; H. Wang et al., 2016). As shown in Figure 2.2.1a, Zdk and LOV2 were attached to the C- and N-termini of cofilin such that they bound to each other in the dark and formed a loop occluding the active site. Upon irradiation, Zdk and LOV2 unlinked, freeing the active site. Because photocontrol was based upon steric inhibition, Z-lock could potentially be applied to diverse proteins or protein fragments with specific activity.

The most important requirement was appropriate orientation of the two termini, which we achieved with linker optimization, and which could in some cases benefit from circular permutation (Y. Yu & Lutz, 2011). We previously used Zdk in another optogenetic approach, named LOVTRAP (Xin X. Zhou, Fan, Li, Shen, & Lin, 2017; X. X. Zhou, Chung, Lam, & Lin, 2012), to sequester molecules at particular subcellular locations. LOVTRAP relied on the binding of Zdk to LOV in the dark, which held the target protein on organelles such as mitochondria until it was released by irradiation. Here we use LOV and Zdk to build an intramolecular bridge over the target proteins' active sites. Building this bridge presented challenges in protein engineering different from those encountered with LOVTRAP. We had to adjust the affinity of the binding components to accommodate the intramolecular interaction of Zdk and LOV, and had to properly orient the bridge they formed upon binding. Z-lock produced an active protein that could be turned on and off wherever it was, not a protein that was

sequestered and reversibly released. The fluorescent protein DRONPA has also been used to control proteins through light-regulated homodimerization over their active site (X. X. Zhou et al., 2012). Z-lock's most important advantages are the ability to adjust the affinity and kinetics of the Zdk-LOV interaction, and the relatively small size of LOV and Zdk, facilitating engineering. This paper describes the engineering and placement of a light-modulated bridge over the active sites of both cofilin and  $\alpha$ -Tat. We make use of a linker, built into the middle of the bridge, that can be reversibly perturbed by irradiation, and the engineering involved for proper placement and affinity of the linker components. This results in a set of tools that can be valuable for similarly controlling other protein active sites.

## 2.2 Results

Development and optimization of Z-lock cofilin Cryo-EM structures of cofilin bound to F-actin show that the C-terminus is free and solvent exposed, and previous studies have demonstrated that GFP fused to cofilin's C-terminus does not interfere with cofilin function (Lai et al., 2008). However, cofilin's N-terminus makes several contacts with F-actin. Therefore, we tested whether fusion of Zdk to cofilin's N-terminus affected F-actin binding ( Fig. 2.5a). For this, we used a well characterized assay that measures the ability of cofilin to co-sediment with actin during ultracentrifugation (Fig. 2.5b, Fig. 2.10) (Pope & Weeds, 1986). Zdk fusion affected neither F-actin binding nor cofilin's ability to convert F-actin to G-actin (Figs. 2.5c and 2.5d, Figs. 2.6a-2.6g and 2.7a-2.7d). To produce light-controlled steric inhibition of cofilin, we needed to position the Zdk-LOV complex where it would block



cofilin F-actin binding in the dark state, but not in the lit state. To find appropriate linkers, we performed Rosetta structure prediction simulations and assessed the orientation of the Zdk-LOV complex relative to the active site (Methods) (Huang et al., 2011). We were able to achieve correct positioning by combining a 5 a.a. N-terminal linker and an 8 a.a. C-terminal linker (Fig. 2.1b). With longer linkers, the complex failed to reliably orient over cofilin's actin binding site. For linkers that were too short, the Rosetta algorithm was unable to form the LOV-Zdk complex over cofilin, indicating that Zdk and LOV would not reach each other to bind in the dark. We tested two variants of Zdk (Zdk1 and Zdk2), with different affinity and Zdk-LOV binding sites, to see which produced light-dependent F-actin interactions. They were compared using the co-sedimentation assay and LOV2 mutants that mimicked the lit and dark conformations. Zdk1 achieved a modest ( $\approx 50\%$ ) reduction in F-actin binding in the dark state that fully recovered in the lit state (Fig. 2.2a, GSGGG lane). Zdk2 produced more robust inhibition ( $\approx 80\%$ ) in the dark state, but activity was not recovered in the lit state (Fig. 2.2b, 2.2c, Fig. 2.8, Cofilin and Zdk2 Cofilin LOV lanes). We attempted to improve the dark state inhibition of the Zdk1 design. Comparing structural models of the Zdk1 and Zdk2 designs revealed a potentially strained linker conformation connecting Zdk1 to cofilin, which we hypothesized was hindering Zdk1 binding to dark state LOV2 (Fig. 2.1b). To improve the Zdk1 design, we tested whether removing a proline and/or lysine from the C-terminus of Zdk1 would enhance linker flexibility and improve binding to dark state LOV2. In rapid optimization studies using single tests of multiple different linkers, one linker was

found to produce a 3-fold reduction in dark state binding, but lit state binding was also reduced (Fig. 2.2a and Fig. 2.9a-2.9c). We turned to optimizing the Zdk2 design based on the hypothesis that Zdk2 was binding too tightly to LOV2. This was reasonable given robust inhibition in both the lit and dark conformation, the higher affinity of Zdk2 for lit state LOV (Zdk2:  $761 \pm 78$  nM; Zdk1:  $> 4$   $\mu$ M), (Kellogg, Leaver-Fay, & Baker, 2011; H. Wang et al., 2016) and the fact that in our design the two proteins were physically linked together (Fig. 2.2d). We reasoned that lowering Zdk2 affinity could facilitate dissociation, so we performed Rosetta mutational analysis to identify point mutants that would modestly decrease Zdk2 affinity (Fig. 2.2e). Mutations were chosen based on two criteria: i) avoiding mutation of residues mediating contact between Zdk2 and the J $\alpha$  helix of LOV2, as this could reduce selectivity for dark state LOV2, and ii) mutating non-polar residues to other non-polar residues, as this is more accurately modeled by Rosetta (H. Wang et al., 2016). To assess each potential mutation, we used Rosetta to calculate the change in Gibbs free energy ( $\Delta\Delta G$ ) for Zdk2 in isolation and for the Zdk2-LOV2 complex, which yielded the  $\Delta\Delta G$  of binding (Fig. 2f). We selected several Zdk2 mutants for testing (V15A, V15I, and I32F) based on their predicted reduction in binding affinity and neutral effect on Zdk2 stability (Supplementary Table 2.1). All mutants tested displayed increased F-actin binding in the lit state relative to wild type Zdk2 (Fig. 2g and Supplementary Fig. 6). One mutant, Zdk2 I32F, exhibited a roughly five-fold difference in F-actin binding for the lit versus dark state. This analog was selected for further development and was named Z-lock cofilin (Fig. 2b, 2c, Supplementary

Fig. 2.4, Z-lock cofilin (I32F) lane, Supplementary Table 2.2). Co- sedimentation assays showed that the affinity of Z-lock cofilin for actin was  $285 \pm 44 \mu\text{M}$  in the dark and  $4.5 \pm 2.4 \mu\text{M}$  in the light (Fig. 2.11a – 2.11b). Rosetta modeling indicated that actin binding was inhibited in the dark state of Z-lock cofilin (Figure 2.12a, 2.12b).

#### Z-lock cofilin effects on Actin in vitro and in cells

To test light-dependent severing of actin by Z-lock Cofilin, we deposited in-vitro polymerized F-actin on coverslips and added lysate from MTLn3 cells expressing Z-lock cofilin. Irradiation of these coverslips led to F-actin severing (Fig. 2.13a, 2.13b, 2.13c), while lysates containing Z-lock cofilin dark state mutant (Supplementary Table 2.2) had no effect (Fig. 2.13a, 2.13b, 2.13c). To confirm that the F-actin severing was caused by cofilin, we purified wt cofilin and Z-lock cofilin mutants that mimic the lit and dark conformations (Methods, Supplementary Table 2.2). Wt cofilin and the lit state mutant severed F-actin efficiently, but the dark state mutant did not ( Fig. 2.13d). Consistent with previous studies of cofilin (Amanda Y. Chan, Bailly, Zebda, Segall, & Condeelis, 2000), Z-lock cofilin in living cells was sequestered in the cytosol until irradiation, when it translocated to the cell edge and co-localized with actin. When irradiation ceased, the Z-lock cofilin returned to its initial distribution (Fig. 2.14). Previous studies have shown that cofilin promotes tumor cell migration and directionality by initiating actin polymerization at the cell edge (Ghosh, 2004; Zoncu et al., 2007). We assessed changes in cell migration and F-actin levels following photoactivation of cofilin in live tumor cells. Z-lock cofilin or a dark state mutant control were expressed in the MTLn3 breast cancer cell line, and

the effects of cofilin photoactivation were assessed using time lapse fluorescence microscopy (“Development and Biologic Properties of Malignant Cell Sublines and Clones of a Spontaneously Metastasizing Rat Mammary Adenocarcinoma23,” 1982). Analysis of cell perimeters before and after photoactivation revealed localized protrusion at the site of photoactivation for Z-lock cofilin but not for the Z-lock cofilin dark state mutant (Fig. 2.3a, Supplementary Table 2.2). We assessed the directionality of cell movement before and after photoactivation by measuring the cosine of the angle between the site of photoactivation and the vector indicating the overall direction of cell movement (Fig. 2.3b). Photoactivation resulted in reorientation of the vector towards the spot of photoactivation, and a consequent increase in cosine values. No change in cosine values was observed for the dark state mutant control. We next investigated the effect of global photoactivation on F-actin levels in MTLn3 cells (Fig. 2.3c). The low expression levels of Z-lock cofilin minimally impacted F-actin levels prior to irradiation (Fig. 2.15 a, b, c, d). Photoactivation resulted in a significant increase in F-actin at 3 minutes post-activation, closely paralleling the timing and level of F-actin increase induced by epidermal growth factor (EGF) stimulation (A. Y. Chan et al., 1998). This suggested that cofilin-dependent actin severing and free barbed end formation is sufficient to induce increases in F-actin following EGF stimulation. Irradiation of the cell edge produced protrusions that retracted when the light was turned off, indicating that effects were reversible (Fig. 2.3d). During light-induced protrusion, cells retracted at positions away from the site of irradiation (Fig. 2.3d). Control cells expressing the

dark state mutant of Z-lock cofilin showed no response to light (Fig.2.3e). Sequential photoactivation at the cell edge sometimes induced protrusions multiple times, but this was inconsistent ( Fig. 2.16a – 2.16d).

#### Z-lock induces invadopodia

Metastasis requires dissemination of primary tumor cells to distant organs, where they form secondary tumors. A key step in this process is tumor cell invasion into blood vessels, which is enabled by matrix-degrading protrusions termed 'invadopodia' (Eddy, Weidmann, Sharma, & Condeelis, 2017). Previous studies indicate that this process is dependent on cofilin; cofilin depletion by RNAi inhibits invadopodium precursor stabilization and results in the slow accumulation of actin cytoskeletal defects that directly affect precursor stabilization and maturation (Bravo-Cordero et al., 2013; Yamaguchi et al., 2005). Transient cofilin activation, such as that observed following EGF stimulation, directly contributes to invadopodia precursor stabilization and maturation by driving localized actin polymerization. We therefore assessed the effect of cofilin photoactivation on invadopodium precursor assembly (Fig. 2.3f). MTLn3 cells expressing either Z-lock cofilin or the Z-lock cofilin dark state mutant were globally irradiated for one minute to mimic the kinetics of cofilin activation following EGF stimulation. The number of invadopodium precursors at different time points was measured by quantifying the number of cortactin- and Tks5-positive puncta, made visible by immunostaining (Fig. 2.3g) (Oser et al., 2009). Photoactivation led to a significant increase in invadopodium precursors five minutes after photoactivation (Fig. 2.3h, 2.3i). No significant change was observed for the

Z-lock cofilin dark state mutant control at any time point. Notably, our results closely match previous findings that precursor formation peaks at five minutes following EGF stimulation (Beaty et al., 2013).

#### Z-lock $\alpha$ TAT

To probe the generality of the Z-lock approach, we used it to control a second target, the alpha-tubulin acetyl transferase ( $\alpha$ TAT). The level of tubulin acetylation is regulated by the antagonistic actions of  $\alpha$ TAT and the tubulin deacetylases, histone deacetylase 6 (HDAC6) (Haggarty, Koeller, Wong, Grozinger, & Schreiber, 2003; Hubbert et al., 2002) and sirtuin type 2 (SIRT2)(North, Marshall, Borra, Denu, & Verdin, 2003). SIRT2 has been studied extensively and is known to deacetylate substrates other than tubulin, including the HAT domain of p300 histone acetyl transferase and a histone H4 peptide (Borra, Langer, Slama, & Denu, 2004; Miotto & Struhl, 2010).  $\alpha$ TAT, on the other hand, is highly specific for microtubules and preferentially acetylates microtubules over free tubulin(Shida, Cueva, Xu, Goodman, & Nachury, 2010). The functional consequences of alpha tubulin acetylation have been investigated primarily through over-expression or knockdown of the deacetylase or the acetyl transferase. We sought to help define the role of this dynamic modification by producing a tool that can generate acetylation in living cells.  $\alpha$ TAT catalyzes the transfer of an acetyl moiety from acetyl co-A to tubulin. Full length  $\alpha$ TAT consists of a catalytic core, a C-terminal extension, and a tail domain(Friedmann, Aguilar, Fan, Nachury, & Marmorstein, 2012) . We used the functional core domain (amino acid, a.a. 2 to 236), for which structural information is

available (Szyk et al., 2014) . The flexible, unstructured tail was excluded because it contains multiple phosphorylation sites that could affect activity independent of the Z-lock steric block. The N and C- termini of  $\alpha$ TAT core are on opposite sides of both the tubulin and acetyl coA binding sites. Blocking either site could effectively inhibit  $\alpha$ TAT activity. Fusing the LOV domain alone to the N-terminus of  $\alpha$ TAT did not effectively block tubulin-binding.

To engineer a Z-lock analog of  $\alpha$ TAT, we fused Zdk1 to the N-terminus of the  $\alpha$ TAT core, and LOV2 to the C-terminus (Figure 2.4a). Western blotting showed that acetylation levels in cells were elevated about two-fold when expressing  $\alpha$ TAT core relative to a dominant negative, kinase-dead mutant  $\alpha$ TAT (DN  $\alpha$ TAT) (Figure 4b, 4c, Supplementary Fig. 2.13, Supplementary Table 2.2). Using a fixed 10 a.a. (5GS) linker between Zdk1 and  $\alpha$ TAT, we tested three different linkers between  $\alpha$ TAT and LOV2 (Supplementary Table 2, 4, 6 and 8 a.a., all GS repeats). We found that caging was optimal using the 6 a.a. (3GS) linker. Longer linkers between  $\alpha$ TAT and LOV2 could fully recover  $\alpha$ TAT activity, but were not as effective at reducing activity in the dark state. Further reduction of linker length resulted in lower light-induced activity. (Figure 2.4b, 2.4c, Supplementary Table 2.1.2) Varying the linker between Zdk1 and  $\alpha$ TAT had little effect. The 3GS version was named Z-lock  $\alpha$ TAT, and was used in all subsequent experiments. An initial examination of the  $\alpha$ TAT structure indicated three potential inhibition mechanisms. The Z-lock components could sterically occlude the binding site for tubulin or for acetyl CoA, or the binding of Zdk to LOV could distort the  $\alpha$ TAT structure without occluding these sites. To examine

which of these three mechanisms is most relevant to our designed switch, we performed structure prediction simulations with Rosetta (Kuhlman, Jacobs, & Linskey, 2016; Leaver-Fay, Tyka, Lewis, Lange, Thompson, Jacak, Kaufman, Renfrew, et al., 2011) . The starting models for the simulations were Zdk fused to the N-terminus of  $\alpha$ TAT with the linker (Zdk-GSGSGSGSGS- $\alpha$ TAT) between the two domains in a random conformation and the LOV domain (in the dark state) fused to the C-terminus of  $\alpha$ TAT with the linker ( $\alpha$ TAT-GSGS-LOV) also in a random conformation. Monte Carlo optimization of the backbone torsion angles in the two linkers was then used to search for low energy models in which Zdk was appropriately docked against the LOV domain. Distance constraints derived from the crystal structure of Zdk bound to the LOV domain were used to direct the docking between Zdk and the LOV domain (H. Wang et al., 2016) .

20000 independent simulations were performed and the models output from each simulation were examined to identify what surfaces of  $\alpha$ TAT are occluded when Zdk binds to the LOV domain. In more than 90% of the models the Zdk/LOV complex is adjacent to the tubulin binding on  $\alpha$ TAT and is predicted to sterically occlude binding to tubulin, while the Zdk/LOV complex never came within 6 angstroms of the acetyl CoA site (Figure 2.4d). This result indicates that the Z-lock switch is reducing activity in the dark state by blocking binding to tubulin. To also test if Z-lock may be reducing  $\alpha$ TAT activity by placing strain on the structure we performed the same structure prediction simulations (sampling different linker conformations to dock Zdk against LOV), but with different starting models for  $\alpha$ TAT. The  $\alpha$ TAT starting models were



derived from an elastic network model (elNemo) that predicts conformational changes  $\alpha$ TAT is likely to undergo in the folded state 50 dramatically perturbed its structure, no changes were observed when simulating Zdk docking with the LOV domain. This indicates that the engineered linkers have enough flexibility to accommodate natural structural perturbations  $\alpha$ TAT may undergo, and that the Z-lock switch is not functioning by placing strain on the  $\alpha$ TAT structure.

To examine the ability of Z-lock  $\alpha$ TAT to acetylate tubulin in living cells, we quantified immunofluorescence staining of acetyl-tubulin in COS7 cells transfected with Z-lock  $\alpha$ TAT or controls (Figure 2.4e). Immunofluorescence images showed increased tubulin acetylation in cells expressing full length  $\alpha$ TAT and  $\alpha$ TAT core, relative to  $\alpha$ TAT dominant negative or untreated cells. Irradiation of Z-lock  $\alpha$ TAT induced a two-fold increase in acetylation. (Figure 2.4f, Supplementary Table 2)

### 2.3 Discussion:

In summary, Z-lock provides a versatile means to place a light-controlled, reversible block over important protein sites. Because it is based on steric inhibition, it should be applicable to a wide variety of protein fragments and in some cases complete proteins. We believe the technique will be most useful where a single active site, i.e. a single activity, needs to be controlled and where other regulatory sites are either removed or mutated away. We envision Z-lock will be used to control protein fragments that modulate endogenous targets, like the  $\alpha$ TAT fragment used here. Cofilin is an example of an intact protein, but is a special case in that it has essentially one important active site. The modifications that we made to cofilin had

little effect on the protein's affinity for actin (Fig. 11). As with most optogenetic analogs, upstream regulation was eliminated, so that activity was affected only by irradiation.

The successful completion of the Z-lock analogs here required protein modeling and engineering, primarily to adjust affinities and linkers for intermolecular interactions.

There was a trade-off between residual dark state activity and maximal activation.

This residual activity is an Achilles heel of most nonchannel optogenetics techniques and often requires careful control of expression level to find conditions where the cell is impacted only upon irradiation. Adjusting Z-dk affinity as exemplified here could tune the balance between 'leakiness' and activity.

Expression of a relatively small amount of Z-lock cofilin was sufficient to achieve optogenetic control (Fig. 15). Z-lock cofilin was used to control actin dynamics in vitro and in live cells, and supports a role for cofilin in initiating actin polymerization and in generating invadopodia in tumor cells (Bravo-Cordero et al., 2013; Eddy et al., 2017; Sidani et al., 2007). Precise control of activation kinetics enabled us to show that cofilin activation alone can generate the actin assembly steps produced by EGF stimulation. In the future, we hope to use the technique in live animals to assess cofilin's contribution to tumor cell migration.

Acknowledgments: We thank the NIH for supporting this work (GM122596 to K.M.H., CA150344 and CA216248 to J.S.C.). O.J.S. is a recipient of a Ruth L. Kirschstein National Research Service Award (1F31CA192739). We thank M. Azoitei for assisting with Rosetta modeling, T. Watanabe for assistance with localized

photoactivation experiments, and Chelsea Onyeji for help with cloning and biochemical assays.

Author Contributions: J.C. and K.M.H. conceived of the project. O.S., B.L. and N.P. carried out the experiments, with assistance and advice from V.S. and R.J.E. in live cell imaging and assay of effects on cofilin. K.H. and H.W. conceived of Z-lock, and J.S.C. initiated the focus on cofilin. K.M.H. and J.S.C. supervised the project. O.S., F. T. and B. K. performed Rosetta modeling. A.P. purified proteins. O.S. and K.M.H. wrote the manuscript with input from all authors.

Competing Financial Interests Statement: The authors declare no competing financial interests.

## 2.4 Online Methods:

Antibodies, DNA Constructs, and Transfection Antibodies were from the following sources: Cofilin (D3F9) XP® Rabbit mAb (Cell Signaling #5175),  $\beta$ -Actin (8H10D10) Mouse mAb (Cell Signaling #3700), Tks5 (Santa Cruz Biotechnology; sc-30122), Cortactin (Abcam; ab33333). The cDNA of the LOV2 domain from *Avena sativa* (oat) Phototropin1 (L404-L547) was used to generate photo-sensitive constructs. Three variants of LOV2 were used: wild-type, dark state mutant (C450A, L514K, G528A, L531E, and N538E), and lit state mutant (I510E/I539E) (Supplementary Table 2). 24 The cDNA of full-length rat cofilin was used for all constructs. The Z antibodies that selectively bind dark state LOV2 have been described elsewhere. For transient expression in mammalian cells, constructs were cloned into pmCherry-C1 (Clontech Laboratories, Inc.). Cells were transfected with Lipofectamine 2000 (Life Technologies) using the manufacturer's protocol 24 h before imaging. For imaging of living cells, cells were co-transfected with mCherry Z-lock cofilin and a membrane-anchored yPet (KRas C-terminus) to visualize the cell edge. 51

### DNA construction:

The cDNA of the LOV2 domain from *Avena Sativa* (oat) Phototropin1 (L404-L547) and the cDNA of cofilin from *Rattus norvegicus* (rat) was used to generate photo-sensitive constructs. All plasmids were generated by Gibson assembly using Gibson Assembly® Master Mix (New England Biolabs, Inc.) according to the manufacturer's protocol. Briefly, fragments were amplified by PCR with Q5® Hot Start High-Fidelity 2X Master Mix (New England Biolabs, Inc.) according to the

manufacturer's protocol. Plasmids (specified in Supplementary Table 2) were linearized via restriction digest. Linkers connecting Zdk and LOV2 to cofilin (specified for each construct in Supplementary Table 2) were introduced by including them in the primers used to amplify each fragment for Gibson assembly.

### Cell Culture

Rat mammary adenocarcinoma cells (MTLn3) were cultured in MEM-alpha media (Gibco; cat # 12561-056), supplemented with 5% FBS (Gemini Bio-Products; cat # 100106) and antibiotics as described earlier. 52 Cells were maintained in an incubator at 37 C and 5% CO<sub>2</sub>. Transiently transfected 293T LINXE cells were used for biochemical assessment of tubulin acetylation levels. Cells were transfected using lipofectamine and plus reagents (Invitrogen). HeLa cells transiently transfected using Fugene6 were plated on plain glass cover slips and used for immunofluorescence imaging.

### Protein expression

Proteins were cloned into the bacterial expression vector pET-14b (Novagen) containing an N-terminal His 6 -tag. The proteins were expressed in Escherichia coli strain BL21(DE3) (New England BioLabs). At OD 600 = 0.8 – 1.0, cultures were induced with 0.5 mM isopropyl  $\beta$ -D-1- thiogalactopyranoside (IPTG, Sigma).

Constructs utilizing lit mutant LOV2 (I510E/I539E) were induced for two hours at 37 C and all other constructs were induced for 5 hours at 37 C. Cultures were pelleted at 5000 RCF and stored at -80 C until purification.

### Protein purification

Bacterial pellets were thawed and resuspended in phosphate buffer (20 mM sodium phosphate, 0.5 M NaCl, pH 7.4). A detergent-based protein extraction reagent (10X BugBuster; EMD Millipore) was added to lysate and incubated for 20 minutes at room temperature. Subsequent purification steps were conducted at 4 °C. Lysate was cleared via centrifugation at 10,000 RCF and imidazole was added to a final concentration of 40 mM. Proteins were purified via immobilized metal affinity chromatography with a gravity-flow column packed with Ni Sepharose 6 Fast Flow (His GraviTrap; GE Healthcare). The column was equilibrated with phosphate buffer containing 40 mM imidazole. Lysate was applied to the column and the column was washed with phosphate buffer containing 40 mM imidazole. Proteins were eluted with phosphate buffer containing 500 mM imidazole. Purified proteins were concentrated via centrifugal filtration using a regenerated cellulose membrane with a 3 kDa molecular weight cut-off (Amicon® Ultra-4; Millipore Sigma). The buffer was exchanged (10 mM Tris, 50 mM NaCl, pH 7.4) via gel filtration using a spin column packed with a 7 kDa molecular weight cut-off size exclusion resin (Zeba™ Spin Desalting Columns; Fisher Scientific).

#### Actin Co-sedimentation Assay

This assay was performed using the Actin Binding Protein Spin-Down Assay Biochem Kit (Cytoskeleton, Inc.) according to the manufacturer's protocol with the minor modifications.

Briefly, Lyophilized rabbit muscle actin was resuspended to 1 mg/ml in 5 mM Tris-HCL pH 8.0 with 0.2 mM CaCl<sub>2</sub>. The resuspended solution of rabbit muscle

actin was polymerized by adding 10X actin polymerization buffer (500 mM KCL, 20 mM MgCl<sub>2</sub>, and 10 mM ATP) to a final 1X concentration. Following polymerization the pH of the polymerized actin solution was adjusted to pH 6.8 to limit cofilin-mediated actin depolymerization. After pH adjustment, 30 uL of the polymerized actin solution was added to 20 uL of purified cofilin in storage buffer (10 mM Tris, 50 mM NaCl, pH 7.4). The reaction mixture contained test protein at a final concentration of 8 μM and F-actin at a final concentration of 12 μM. Reactions were incubated at room temperature for 30 minutes and centrifuged for 1 hour at 164,000 RCF. Supernatant was removed and pellets were resuspended in 50 uL Milli-Q water. Samples were combined with Laemmli sample buffer, boiled for 5 minutes and stored at 4 C. Samples were resolved by SDS- PAGE followed by Western blot using the above mentioned cofilin and β-Actin antibodies. The results were normalized by first measuring the band density of cofilin in the pellet and supernatant fractions. The relative percent of cofilin bound to F-actin was then calculated by dividing the band density of cofilin in the pellet fraction by the sum of the band densities for both the pellet and supernatant fractions. The average relative percent cofilin bound to F-actin for wild-type cofilin was then used to normalize other measurements. F-actin severing with Z-lock cofilin. F-actin was polymerized using a mixture of unlabeled-actin (1.4 μM), biotin-actin (0.2 μM) and rhodamine-actin (0.4 μM) in actin polymerization buffer (40 mM pH 7.5 Tris HCl, 10 mM EGTA, 4 mM MgCl<sub>2</sub>, 100 mM KCl) for 2 hours at room temperature. MatTek glass- bottomed petri dishes were coated with a layer of PEG/PEG-biotin mixture, dried and stored at 4 C

before use. Before imaging, the coated dishes were incubated with 0.5 mg/ml streptavidin for 5 minutes, and washed 5X with Tris-HCl. F-actin (0.033  $\mu$ M) was deposited on the coated dishes for 15 minutes and rinsed 2X with wash buffer (Actin polymerization buffer, 0.5 mg/ml BSA) twice gently. Wash buffer was removed and purified Cofilin or its mutants was added (Z-lock cofilin lit, Z-lock cofilin dark, see Supplementary Table 2). This was diluted in assay solution (10 mM pH 7.5 Tris HCl, 2.5 mM EGTA, 0.25 mM MgCl<sub>2</sub>, 5 mg/ml BSA, GLOX, 0.5 mM, 10 mM DTT, Protease Inhibitors). Polymerization of F-actin was examined immediately after adding the protein and after indicated times using total internal reflection microscopy (TIRF). To monitor F-actin severing by Z-lock cofilin with light, we transiently transfected MTLn3 cells with Z-lock cofilin or Z-lock cofilin dark (see Supplementary Table 2). Cell lysate was prepared at 4 C and then loaded onto F-actin coated coverglasses. PA-Cofilin was photoactivated by pulsing blue light (on/off, 1s/1s) (Chroma filter HQ470/40x) for 1 min. The images were taken immediately and after the indicated times.

#### Western blot quantification

Cofilin and  $\beta$ -Actin band densities were quantified using ImageStudio, with local background subtraction. To calculate F-actin binding, the cofilin band density for the pellet fraction was divided by the sum of the cofilin band intensities for the pellet and supernatant fractions.

#### Whole-cell photoactivation



Cells were plated on glass-bottom dishes (MatTek Corporation) and allowed to spread overnight protected from light. For F-actin staining cells were plated on acid washed dishes. For invadopodium precursor staining cells were plated on gelatin coated dishes. All work was performed under red light to prevent unintentional photoactivation. Cells were serum-starved 4 hours prior to photoactivation in Leibovitz's L-15 media (Gibco) containing 0.35% BSA. 53 Photoactivation was accomplished with a 470 nm LED array (Mouser Electronics, Inc. part # 828-OVQ12S30B7). The surface of the tissue culture plate was positioned approximately 1 cm away from the LED array, which resulted in a measured power density of 0.064 nW/ $\mu\text{m}^2$  at 608 nm. During photoactivation, cells were maintained in a cell culture incubator set to 37 C and 5% CO<sub>2</sub>.

#### Immunofluorescence

For F-actin and invadopodia analysis MTLn3 cells were fixed with 3.7% paraformaldehyde for 20 min at room temperature and washed 3X with Dulbecco's Phosphate- Buffered Salt Solution (PBS)(Potassium Chloride 0.2g/L, Potassium Dihydrogen Phosphate 0.2g/L, Sodium Chloride 8g/L, Disodium Phosphate 1.15g/L). Cells were permeabilized with Triton-X-100 0.1% solution in PBS for 5 min and washed 3X with PBS. Cells were blocked with 1% BSA and 1% FBS in PBS for 1 h at room temperature. For invadopodia analysis, cells were stained with primary and secondary antibodies at the manufacturer's suggested dilutions in blocking buffer for 1 hour and washed 3X with PBS after each incubation. For F-actin analysis:

DyLight<sup>TM</sup> 488 Phalloidin (Cell Signaling, 12935) was incubated with cells at 1X concentration for 20 min and washed 3X with PBS.

#### Preparation of fluorescent gelatin coated dishes

Gelatin was labeled with Alexa-405 dye and glass bottom MatTek dishes were coated with the fluorescent gelatin as described earlier.<sup>54</sup> Briefly, dishes were acid-washed (1N HCl for 10 min) and coated with poly-L-lysine (50 µg/mL for 20 min) followed by Alexa 405-gelatin coating (0.2% gelatin for 10 min). Gelatin matrix was then crosslinked (0.1% glutaraldehyde for 15 min) and inactivated (5 mg/ml NaBH<sub>4</sub> for 15 min). After each step dishes were washed (3 x 5 min PBS). Dishes were stored at 4°C in 10X Pen-Strep (1,000 IU/mL Penicillin, 1,000 µg/mL Streptomycin; ThermoFisher Scientific). All solutions were prepared fresh immediately before use.

#### Live cell imaging

Cells were plated on acid-washed glass bottom MatTek dishes and allowed to spread overnight, protected from light. Prior to imaging, cells were serum starved in Leibovitz's L-15 media (Gibco) containing 0.35% bovine serum albumin. A closed heated chamber was used during live cell imaging. Imaging was performed using a Zeiss LSM 880 confocal microscope equipped with a Plan-Apochromat 63x oil objective (N.A. 1.40). ZEN software (Zeiss) was used to control the microscope and acquire images at each time point. A GaAsP detector with tunable emission collection windows (Zeiss) was used for detection. YFP images were acquired using a 514 nm Argon laser (25% power) with a collection window of 525 – 580 nm. mCherry images were acquired using a 561 nm DPSS laser (20% laser power) with

a collection window of 580 – 650 nm. LOV2 photoactivation was accomplished with a 488 nm Argon laser (1% power) that irradiated a preselected region every 10 seconds. Images were acquired every 2.5 seconds.

#### Directionality Analysis

Changes in cell directionality were quantified using the directionality index, which is defined as the cosine of the angle between the site of photoactivation and the vector direction of cell movement. 10 The vector direction of cell movement was determined by measuring the cell centroid at two different time points. We used Fiji/Imagej to define the cell centroids, following thresholding of each cell. Fiji/Image calculates the centroid by taking the average of the x and y coordinates of all of the pixels for the thresholded cell. The site of photoactivation was determined by measuring the centroid of the photoactivation ROI using Fiji/ImageJ. The directionality index was assessed for two intervals. First, two minutes prior to photoactivation until the time of photoactivation, which measured cell movement before photoactivation. Second, from the time of photoactivation until two minutes after, which assessed potential changes in directionality in response to photoactivation.

#### Measurement of F-actin content

Cells fixed and stained with phalloidin were imaged on an Olympus IX-81 microscope equipped with a UPlanFLN 40x objective (Pil, N.A 1.30). Metamorph software (Molecular Devices) was used to control the microscope and acquire images. Dylight 488 and mCherry images were acquired using a 100 Watt mercury

arc lamp with a 1% ND filter and a 500-550 nm or 565–595 nm band-pass filter respectively, with 1 second exposure for each channel. Flat field correction was applied to each image using a custom MatLab script. <sup>55</sup> Corrected images were thresholded with Otsu's method <sup>56</sup> using Fiji/ImageJ to generate masks for individual cells. For each image, a region without cells was used to determine background intensity for background subtraction. Mean phalloidin intensity (shade corrected and background subtracted) was measured for individual cells using previously generated masks. Experimental replicates were imaged on the same day to enable comparison based on signal intensity. Cell intensities for each condition were averaged for the sake of comparison.

#### Invadopodium Precursor Analysis

MTLn3 cells transfected with light-insensitive control or mCherry Z-lock cofilin were irradiated for 1 min and fixed at 1, 3, 5- and 30-min post photoactivation. Cells were stained with anti-Cortactin and antiTks5 antibodies and imaged on a Delta Vision epi- fluorescence microscope (Applied Precision Inc.), equipped with a CoolSNAP HQ2 camera and a 60x, NA 1.42 objective lens. Invadopodium precursors were identified as Tks5- and cortactin-positive puncta. The number of invadopodium precursors per cell were quantified in Fiji/ImageJ. Modeling of linkers connecting Zdk and LOV2 to cofilin Structural models were generated using the RosettaRemodel package with the Rosetta3.5 series software. <sup>31,48</sup> This package was designed to provide a framework for flexible protein design utilizing the loop modeling tools in Rosetta. In this case, we used the domain insertion protocol to model the orientation

of the Zdk-LOV complex relative to cofilin with Zdk and LOV attached to the N- and C-terminus of cofilin (PDB 4BEX), respectively. 57 For both Zdk1 (PDB 5EFW) and Zdk2 (PDB 5DJT), we modeled linkers of different length and composition and assessed whether the Zdk-LOV complex reliably orientated over the actin binding interface of cofilin.

#### Computational identification of Zdk2 point mutants

The change in binding energy for Zdk2 point mutants was calculated using the ddG monomer package with the Rosetta3.5 series of software. 32 The package was designed to predict the change in stability (the ddG) of a protein induced by a point mutation. In this case, we calculated the ddG induced by several point mutants for both Zdk2 and the Zdk2-LOV complex, using the PDB structure 5DJT as a starting point. Subtracting the ddG for Zdk2 from the ddG for the Zdk2-LOV complex yielded the change in binding energy. The shift in binding curve was calculated using the following equation:

$$\Delta\Delta G = -RT \ln \frac{K_{d2}}{K_{d1}}$$

where  $\Delta\Delta G$  is the change in binding energy, R is the gas constant, T is the temperature, and  $K_{d2}/K_{d1}$  is the shift in the binding curve.

#### Statistical analysis.

Statistical significance was determined by either t-test (normally distributed dataset) or Mann–Whitney test (non-normally distributed dataset). Normality of each dataset was tested with D'Agostino & Pearson normality test. For selected datasets, we used a Tukey box-and-whisker plot. The ends of the box denote the interquartile range

(IQR) and the median is marked as a line across the box. Whiskers represent either the largest data point less than or equal to 1.5 times the IQR (upper whisker) or the smallest data point more than or equal to -1.5 times the IQR (lower whisker).

Outliers (any value larger than 1.5 times the IQR or smaller than -1.5 times the IQR) are displayed as dots. All statistical analysis was conducted with GraphPad Prism 7. During optimization of linkers and Zdk2 mutants, some constructs were tested only once (Figure 2.2a and 2.2g) . The final design used for cell-based experiments was retested by actin co-sedimentation assay in three independent experiments and the data is shown in Figure 2.2c. Microscopy images are representative of three independent experiments.

#### Biochemical validation of Z-lock $\alpha$ TAT

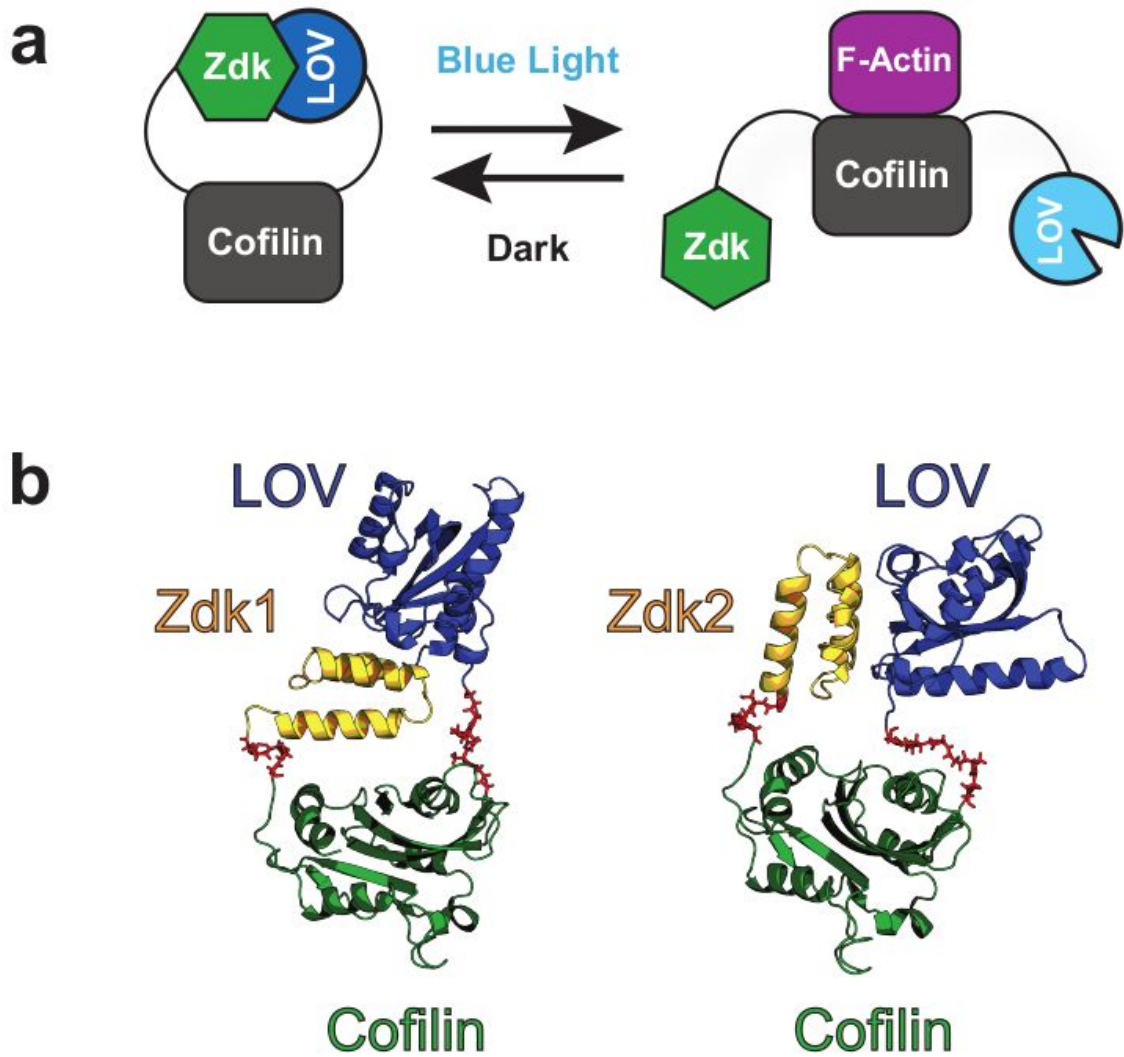
A 5sec on- 5sec off regime of blue light exposure was achieved using an LED-panel constructed for placement in a 37 °C, 5% CO<sub>2</sub> incubator and controlled with an Arduino board. The blue light intensity on the cells was approx. 0.05 nW/ $\mu$ m<sup>2</sup> . Cells were kept under blue light for 30 minutes and then lysed in lysis buffer on ice for 20min. Lysates were loaded on SDS-PAGE gels and transferred to PVDF membranes for Western blotting. The samples were stained with monoclonal anti-AcetylTubulin antibody (Sigma: 6-11B-1,) for acetylated alpha tubulin and anti-FLAG antibody (Abcam: ab49763) for either  $\alpha$ TAT or Z-lock  $\alpha$ TAT at 4 C overnight. The samples were then wash and stained with dye-labeled secondary antibodies (ThermoFisher: Dylight 800; Bio-rad: Starbright 700) at room temperature for 1 hour. Immunofluorescence of microtubule acetylation.

Cells were fixed with ice-cold 100% methanol for 3 minutes before permeabilizing with 0.5% Triton-X100. Staining was performed with the same primary antibodies as above and with a pair of dye-conjugated secondary antibodies (Abcam: ab150105, ab175471). All immunofluorescence buffers were made from a 10x stock of BRB80 with 0.1% Triton-X100. Stained cells were mounted in prolong gold (ThermoFisher Scientific) and imaged on an Olympus spinning disk confocal microscope with a 60x objective. Intensity measurements were made by masking the cells in Fiji/ImageJ and normalizing acetylation intensity to expression level.

Data and Code Availability Statement: The data that support the findings of this study are available from the corresponding authors upon request. Code is available from the authors upon request or at [hahnlab.com](http://hahnlab.com).

## 2.5 Figures and Tables

Figure 2.1. Design of Z-lock cofilin.

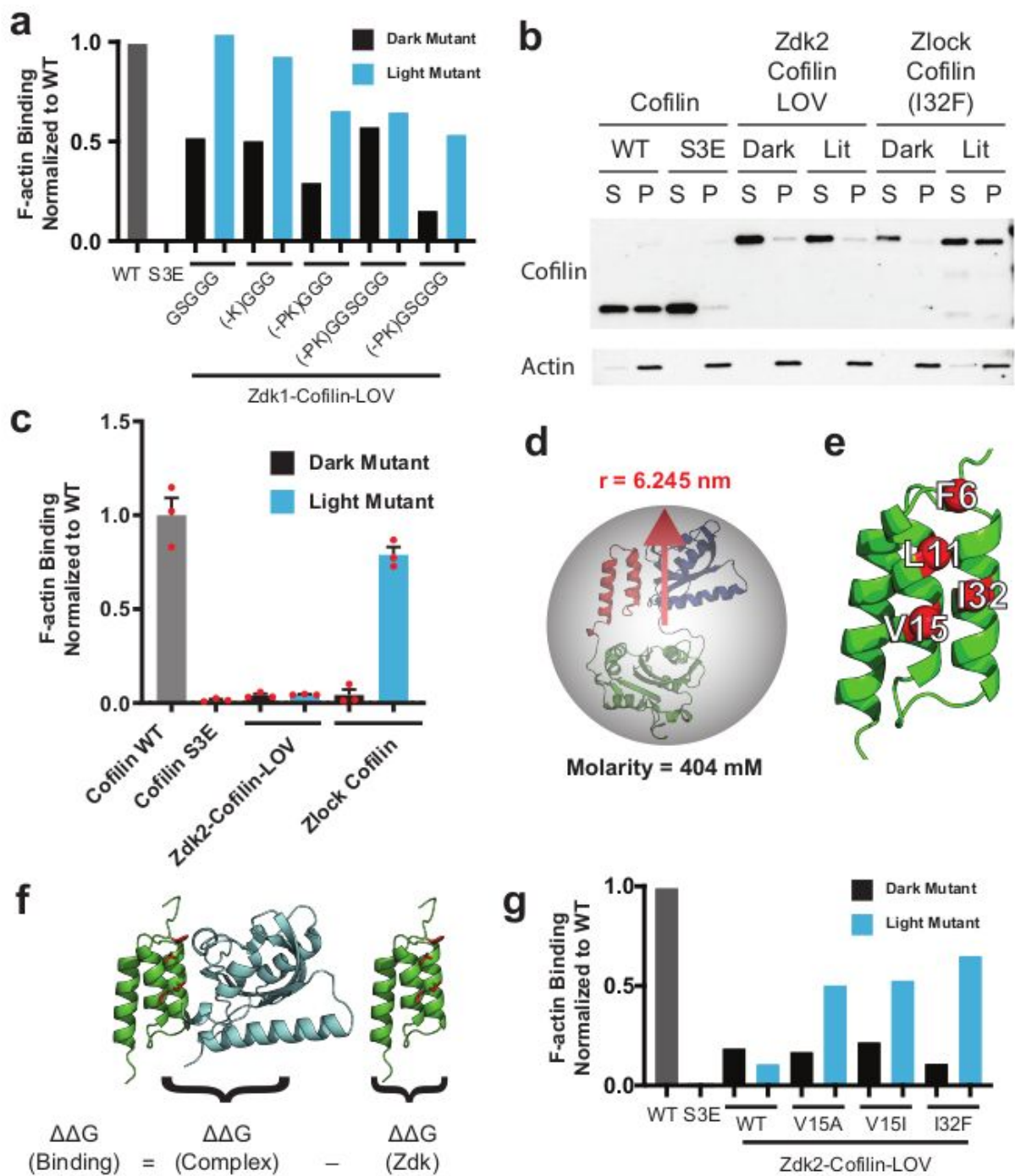


a. Design of Z-lock cofilin. Cofilin (gray) is fused at its N-terminus to Zdk (green) and at its C-terminus to LOV (blue). In the dark, Zdk binding to LOV blocks cofilin binding to F-actin (purple). b. Rosetta-based structure prediction for designs incorporating Zdk1 versus Zdk2. Linker residues connecting Zdk and LOV to cofilin are shown in red.





Figure 2.2. Optimization of designs based on Zdk1 and Zdk2.



a. Actin co-sedimentation assay to measure binding of cofilin to F-actin. Wild type and S3E inactive cofilin mutants are shown as controls. Different linkers connecting

Zdk1 to cofilin are shown on the x-axis. Parentheses indicate residues truncated from the Zdk1 C-terminus. Dark and lit state mutants were used to assess light-dependent changes in cofilin F-actin binding (Supplementary Table 2). In this initial screening study, each variant was tested once.

b. c.) Actin co-sedimentation assay to measure binding of cofilin to F-actin. Wild-type cofilin and S3E constitutively inactive cofilin mutants are shown as controls.

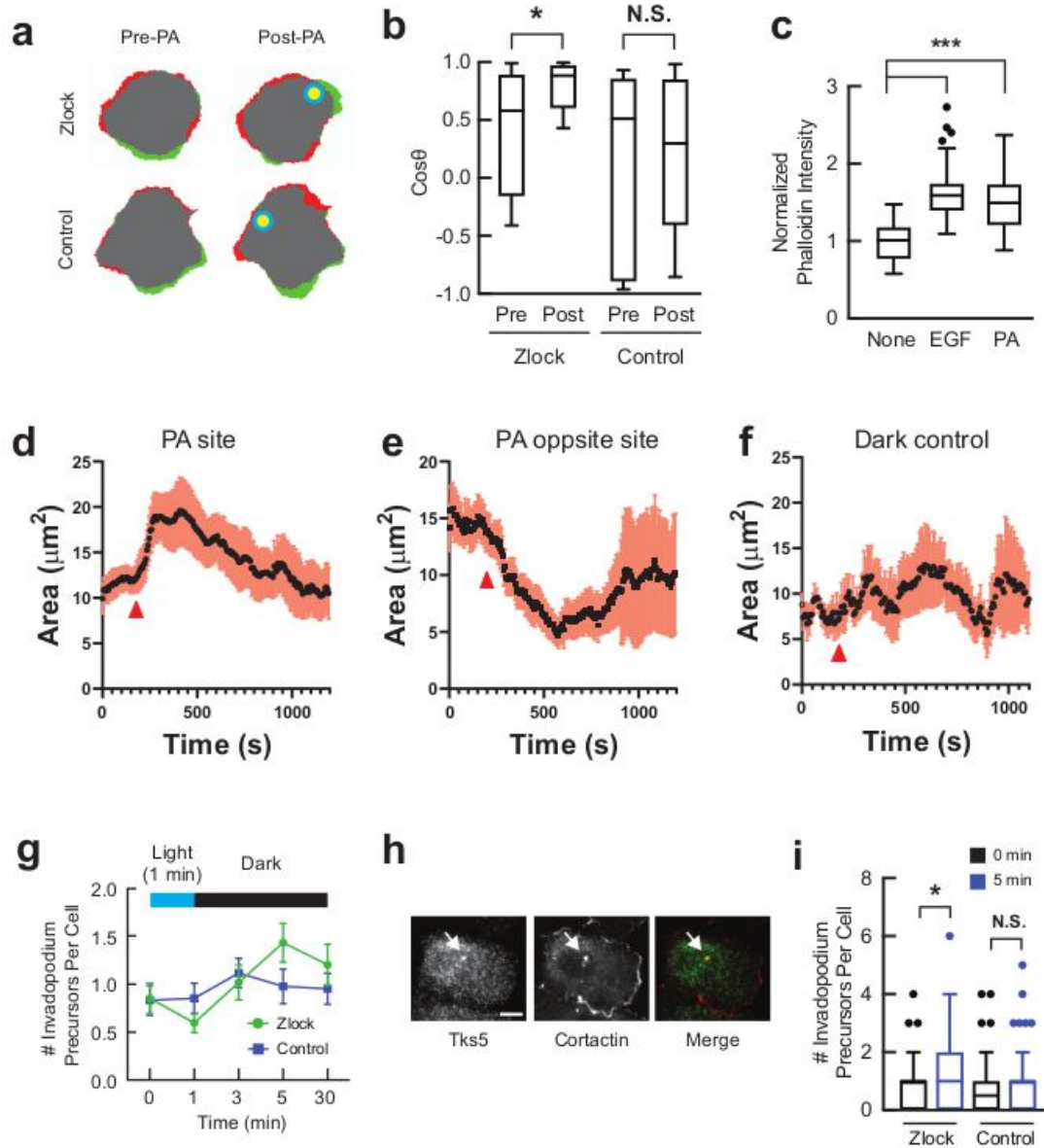
Light-dependent binding to F-actin was tested using LOV2 mutants that mimic the dark state and lit state (Supplementary Table 2). The center and error bars in c represent mean  $\pm$  SEM from three independent experiments. (See full gels in Supplementary Fig. 5)

d. Estimation of LOV2 effective molarity. The volume of a sphere enclosing the Zdk2-based design was determined and used to estimate the effective LOV2 molarity encountered by Zdk2. e. Four residues (shown in red) along the first and second helices of Zdk2 were mutated to generate a reduced affinity variant of Zdk2.

f. The change in Gibbs free energy ( $\Delta\Delta G$ ) for Zdk2 mutants was calculated for the Zdk2-LOV2 complex ( $\Delta\Delta G$  complex) and Zdk2 in isolation ( $\Delta\Delta G$  Zdk2). Subtracting the two values yielded the change in binding energy of the complex ( $\Delta\Delta G$  binding).

g. Actin co-sedimentation assay to measure the binding of cofilin to F-actin. Wild type and S3E inactive cofilin mutant are shown as controls. Zdk2 mutants are shown on the x-axis. Dark and lit state mutants (Supplementary Table 2) were used to assess light-dependent changes in cofilin F-actin binding. In this initial screen, each variant was tested only once.

Figure 2.3. Effect of Z-lock cofilin activation on leading edge protrusions and invadopodium formation in tumor cells.



a. Cell perimeter before (left) and after (right) photoactivation. Retraction = red; protrusion = green; no change = grey. The site of photoactivation is indicated by the

yellow circle. The dark state mutant of Z-lock cofilin is shown as a control. (Z-lock: n= 8 cells; Control: n= 9 cells) (Three independent experiments)

b. Analysis of directional migration in response to irradiation of either Z-lock cofilin or Z-lock cofilin dark state mutant (Supplementary Table 2). The cosine of the angle between the site of photoactivation and the vector of cell movement were calculated for two minutes before (Pre-PA) and after photoactivation (Post-PA). Photoactivation led to an increase in the cosine value for Z-lock cofilin (p-value = 0.0271; n = 8 cells, paired two-tailed t-test) but not for the Z-lock cofilin dark state mutant (p-value = 0.5782; n = 9 cells, paired two-tailed t-test). Cosine value for Pre-PA Z-lock cofilin and Z-lock cofilin dark state mutant were not significantly different (p-value = 0.3523, unpaired two-tailed t-test). Tukey box-and-whisker plot shown with outliers displayed as dots. (Three independent experiments)

c. F-actin content of MTLn3 cells expressing Z-lock cofilin that were either unstimulated (n = 34 cells), stimulated with 5 nM EGF (n = 56 cells), or photoactivated (PA) (n = 51 cells). F-actin content was assessed following fixation and phalloidin staining. PA cells were irradiated for one minute and fixed three minutes after photoactivation. Phalloidin intensity was significantly different for both EGF (p < 0.0001) and PA (p < 0.0001) relative to unstimulated cells. EGF and PA phalloidin intensity were not significantly different from one another (p = 0.0521).

P-value

was calculated using Mann–Whitney test, two-sided. Tukey box-and-whisker plot shown with outliers displayed as dots. (Three independent experiments) d. e. f. ) d

Cell area change within the PA spot. Red arrow indicates the start of photoactivation. Cells were irradiated for 60 s using a 500 ms pulse of blue light every second (n=9). e A spot opposite the site of PA is monitored (n=9). f Cell area change within the PA spot for cells expressing the dark mutant of Z-lock cofilin (n=9) (Supplementary Table 2). Data is shown as mean  $\pm$  SEM. (Three independent experiments)

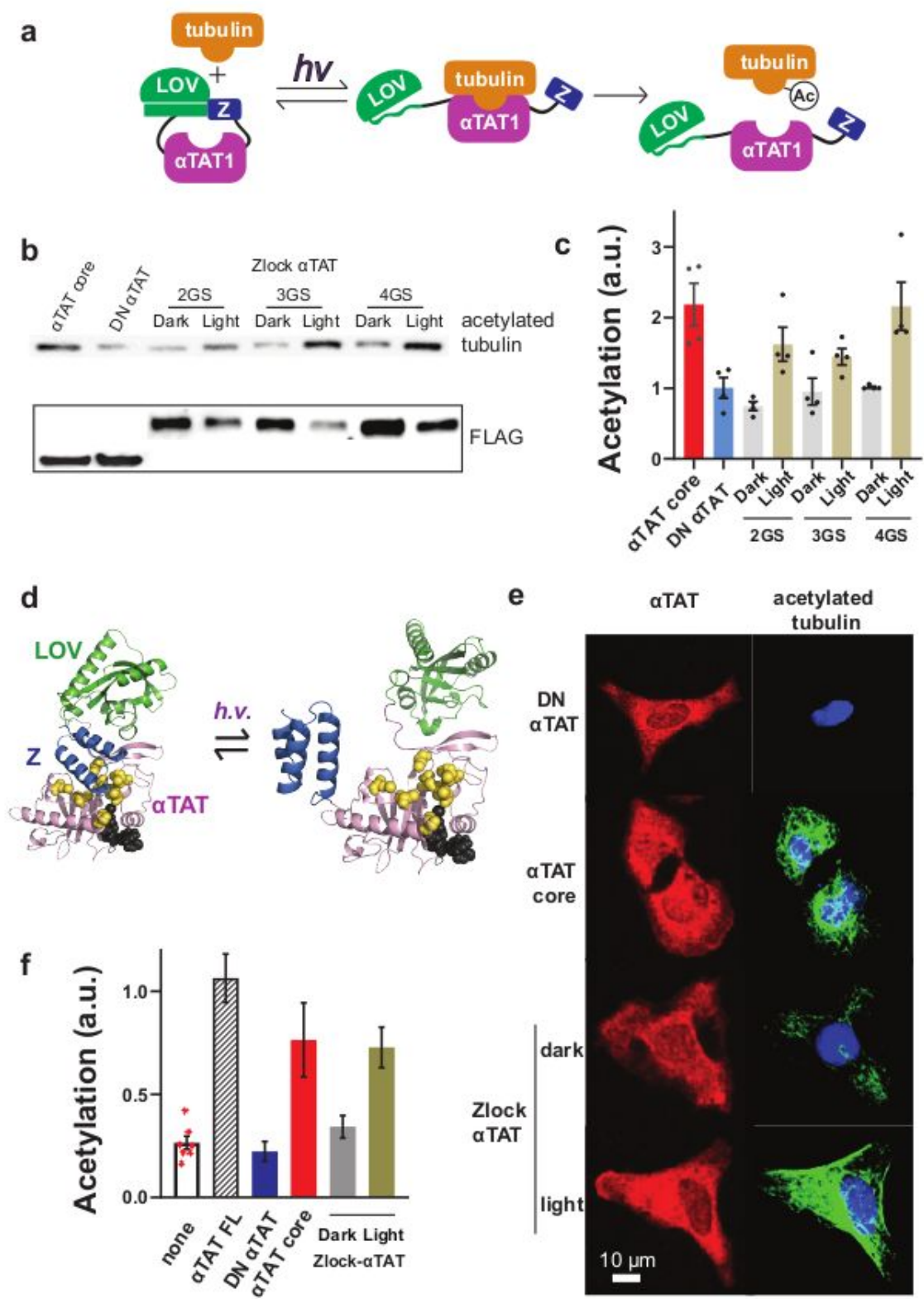
g. Effect of photoactivation on the number of invadopodium precursors in MTLn3 cells expressing Z-lock cofilin versus a dark state mutant. Data is shown as mean  $\pm$  SEM. Z-lock cofilin (0 min: n = 41 cells; 1 min: n = 57 cells; 3 min: n = 51 cells; 5 min: n = 46 cells; 30 min: n = 54). Control (0 min: n = 48; 1 min: n = 48; 3 min: n = 51; 5 min: n = 45; 30 min: n = 43). (Three independent experiments)

h. Representative immunofluorescence images of MTLn3 cell stained for Tks5 and cortactin to identify invadopodium precursors (indicated by the white arrow). Scale bar: 10  $\mu$ m.

i. Change in number of invadopodium precursors per cell following photoactivation, based on the data in Figure 2.3g. Photoactivation resulted in a significant increase of invadopodium precursors for Z-lock cofilin (p = 0.0332, Mann–Whitney test, two-sided) but not for the Z-lock cofilin dark state mutant (p = 0.6001, Mann–Whitney test, two-sided) at 5 min post-photoactivation. Tukey box-and-whisker plot shown with outliers displayed as dots.



Figure 2.4. Z-lock  $\alpha$ TAT





a. Design of Z-lock  $\alpha$ TAT showing light-induced acetylation of tubulin.

b. c.) b Western blotting shows microtubule acetylation in 293T<sub>Lin</sub>XE cells resulting from three versions of LOV- $\alpha$ TAT-Zdk (2GS/3GS/4GS) in the dark vs light

(Supplementary Table 2). The enzyme core (a.a. 2-237) was used as a positive control ( $\alpha$ TAT core) and a core mutant with reduced activity (Q58A/D157N) was used as a negative control (DN  $\alpha$ TAT). c Quantification of Western blotting in b

showed how linker modification affected the activity in the light versus the dark.

Acetylation levels were normalized to construct expression (see FLAG blotting in b).

The 3GS linker version was denoted Z-lock  $\alpha$ TAT. Data is shown as mean  $\pm$  SEM (n=4). (See full gels in Fig. 17) (Three independent experiments)

d. Rosetta-based prediction of Z-lock  $\alpha$ TAT structures in the dark and lit states. e. f.)

e Immunofluorescence images showing the light-mediated acetylation of

microtubules in HeLa cells expressing indicated constructs. Cells expressing Z-lock  $\alpha$ TAT were examined with without blue light activation. Quantification of acetylation

level from immunofluorescence data. Values are normalized for construct expression

level. Data is shown as mean  $\pm$  SEM (none: n = 7;  $\alpha$ TAT FL: n = 11; DN  $\alpha$ TAT: n =

34;  $\alpha$ TAT core: n = 44; Z-lock Dark: n = 43; Z-lock Light: n = 57). (Three independent experiments)

Supplementary Table 2.1: Change in Gibbs energy of binding for Zdk2 mutants.

<b>Mutation</b>	<b>Zdk2 <math>\Delta\Delta G</math></b>	<b>Zdk2-LOV <math>\Delta\Delta G</math></b>	<b>Binding <math>\Delta\Delta G</math></b>	<b>Fold-shift in binding curve</b>
<b>I32W</b>	<b>0.58</b>	<b>14.19</b>	<b>13.61</b>	<b>5222243798.76</b>
I32Y	2.42	9.79	7.37	183171.18
F6C	3.28	5.65	2.37	48.90
<b>I32G</b>	<b>2.03</b>	<b>4.35</b>	<b>2.32</b>	<b>45.04</b>
F6G	4.13	6.32	2.19	36.91
F6M	2.04	4.23	2.19	36.55
<b>F6P</b>	<b>1.88</b>	<b>3.73</b>	<b>1.86</b>	<b>21.17</b>
<b>I32F</b>	<b>0.21</b>	<b>1.92</b>	<b>1.71</b>	<b>16.76</b>
L11Y	0.92	2.50	1.58	13.45
V15G	1.24	2.81	1.56	13.05
V15C	0.60	2.15	1.55	12.86
<b>V15I</b>	<b>-0.06</b>	<b>1.44</b>	<b>1.50</b>	<b>11.79</b>
<b>L11I</b>	<b>0.33</b>	<b>1.76</b>	<b>1.42</b>	<b>10.40</b>
<b>V15A</b>	<b>0.23</b>	<b>1.59</b>	<b>1.36</b>	<b>9.30</b>
F6A	2.92	4.15	1.23	7.59
F6I	2.64	3.79	1.15	6.61
V15M	0.15	1.21	1.06	5.72
V15W	-0.20	0.83	1.04	5.50
F6L	1.08	2.08	1.01	5.25
V15P	5.27	6.25	0.98	5.04
L11C	2.27	3.18	0.91	4.45
F6V	2.77	3.67	0.90	4.39
L11V	1.61	2.47	0.86	4.09
V15L	-0.15	0.65	0.80	3.70
L11G	2.58	3.33	0.76	3.47
I32L	0.22	0.95	0.74	3.37
L11A	1.21	1.77	0.56	2.53
L11P	12.45	12.92	0.47	2.16
I32A	2.39	2.83	0.44	2.05
L11W	0.68	0.96	0.28	1.57
V15V	0.00	0.20	0.20	1.38
I32M	1.20	1.39	0.20	1.38
V15F	0.27	0.46	0.18	1.35
L11L	0.00	0.00	0.00	1.00
V15Y	0.11	-0.18	-0.29	0.62
F6Y	0.09	-0.33	-0.42	0.50
I32I	0.00	-0.43	-0.43	0.49
F6F	-0.06	-0.61	-0.55	0.40
L11F	1.14	0.57	-0.57	0.39
I32P	9.03	8.33	-0.70	0.32
L11M	1.00	0.27	-0.72	0.30
I32C	2.89	2.10	-0.79	0.27
F6W	1.69	0.47	-1.22	0.13
I32V	1.38	-0.15	-1.53	0.08

## Supplementary Table 2.2: Plasmids

See footnotes for names used in text

Constructs details	Associated figures	Backbone
Zdk2-GGG-Cofilin(WT)-6xHis	Supp. Fig. 1 c-d	pET
Zdk2-GGGG-Cofilin WT		pET
Zdk2-GSGGG-Cofilin WT		pET
Zdk2-GSGGGG-Cofilin WT		pET
Zdk2-GGSGGG-Cofilin WT		pET
<sup>1</sup> His-Zdk1-GSGGG-CofilinWT-GGSGGSGG-LOV(sDark)	Fig. 2a	pET
<sup>2</sup> His-Zdk1-GSGGG-CofilinWT-GGSGGSGG-LOV(lit)		pET
His-Zdk1-(-K)GGG-CofilinWT-GGSGGSGG-LOV(sDark)		pET
His-Zdk1-(-K)GGG-CofilinWT-GGSGGSGG-LOV(lit)		pET
His-Zdk1-(-PK)GGG-CofilinWT-GGSGGSGG-LOV(sDark)		pET
His-Zdk1-(-PK)GGG-CofilinWT-GGSGGSGG-LOV(lit)		pET
Zdk1-[-(PK)GSGGG]-CofilinWT-GGSGGSGG-LOV(sDark)		pET
Zdk1-[-(PK)GSGGG]-CofilinWT-GGSGGSGG-LOV(lit)		pET
Zdk1-[-(PK)GSGGG]-CofilinWT-GGSGGSGG-LOV(sDark)		pET
Zdk1-[-(PK)GSGGG]-CofilinWT-GGSGGSGG-LOV(lit)		pET
His-Zdk2-GSGGG-CofilinWT-GGSGGSGG-LOV(sDark)	Fig. 2	pET
His-Zdk2-GSGGG-CofilinWT-GGSGGSGG-LOV(lit)		pET
His-Zdk2(V15A)-GSGGG-CofilinWT-GGSGGSGG-LOV(sDark)	Fig. 2g	pET
His-Zdk2(V15A)-GSGGG-CofilinWT-GGSGGSGG-LOV(lit)		pET
His-Zdk2(V15I)-GSGGG-CofilinWT-GGSGGSGG-LOV(sDark)		pET
His-Zdk2(V15I)-GSGGG-CofilinWT-GGSGGSGG-LOV(lit)		pET
<sup>3</sup> His-Zdk2(I32F)-GSGGG-CofilinWT-GGSGGSGG-LOV(sDark)	Fig. 2; Supp. Fig. 7, 9	pET

His-Zdk2(I32F)-GSGGG-CofilinWT-GGSGGSGG-LOV2Lit	Fig. 2; Supp. Fig. 7, 9	pET
<sup>4</sup> Cofilin-6xHis	Fig. 2;	pET
<sup>5</sup> Cofilin(S3E)-6xHis	Fig. 2;	pET
<sup>6</sup> Cofilin(S3A)-6xHis	Fig. 2; Supp. Fig. 1b	pET
mCherry-Zdk2(I32F)-GSGGG-CofilinWT-GGSGGSGG-LOV(sDark)	Supp. Fig. 9	pEGFP
<sup>7</sup> mCherry-Zdk2(I32F)-GSGGG-CofilinS3A-GGSGGSGG-LOV	Fig. 3; Supp. Fig. 9-12	pEGFP
<sup>8</sup> FLAG-mCherry-TAT	Fig. 4	pTrix4
<sup>9</sup> FLAG-mCherry- $\alpha$ TAT 236	Fig. 4	pTrix4
<sup>10</sup> FLAG-mCherry- $\alpha$ TAT (Q58A/D157N)	Fig. 4	pTrix4
FLAG-mCherry-Zdk1-5GS- $\alpha$ TAT 236-2GS-LOV2	Fig. 4	pTrix4
<sup>11</sup> FLAG-mCherry-Zdk1-5GS- $\alpha$ TAT 236-3GS-LOV2	Fig. 4	pTrix4
FLAG-mCherry-Zdk1-5GS- $\alpha$ TAT 236-4GS-LOV2	Fig. 4	pTrix4

1: LOV sDark mutant: C450A, L514K, G528A, L531E, and N538E.

2: LOV lit mutant: I510E/I539E.

3: Z-lock cofilin dark state mutant

4: Cofilin WT

5: Inactive Cofilin

6: Active Cofilin

7: Z-lock cofilin

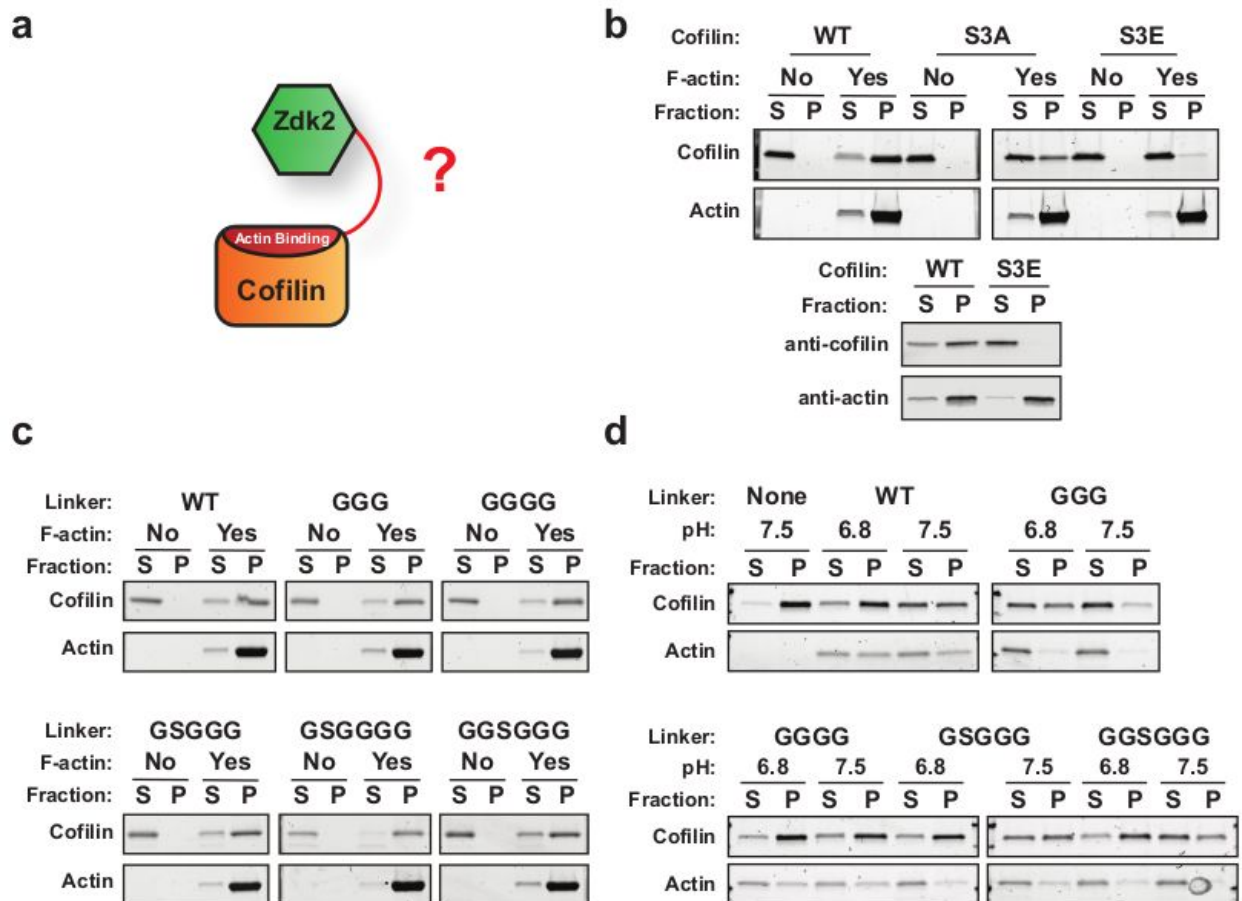
8:  $\alpha$ TAT full length

9:  $\alpha$ TAT core,  $\alpha$ TAT amino acids 2-236

10: DN  $\alpha$ TAT

11: Z-lock  $\alpha$ TAT

Figure 2.5 Validation of actin co-sedimentation assay and testing of Zdk-cofilin linkers.



a. Diagram of Zdk fused to the cofilin N-terminus.

b. Upper: SDS-PAGE gel of supernatant (S) and pellet (P) fractions from actin co-sedimentation assay. Lower: Western blot of supernatant and pellet fractions from actin co-sedimentation assay. Western blot was used to assess Z-lock cofilin designs due to the similar molecular weight of Z-lock cofilin and actin. Wild-type cofilin, S3A constitutively active mutant cofilin, and S3E constitutively inactive mutant cofilin were run with and without actin to test the dependence of cofilin pelleting on

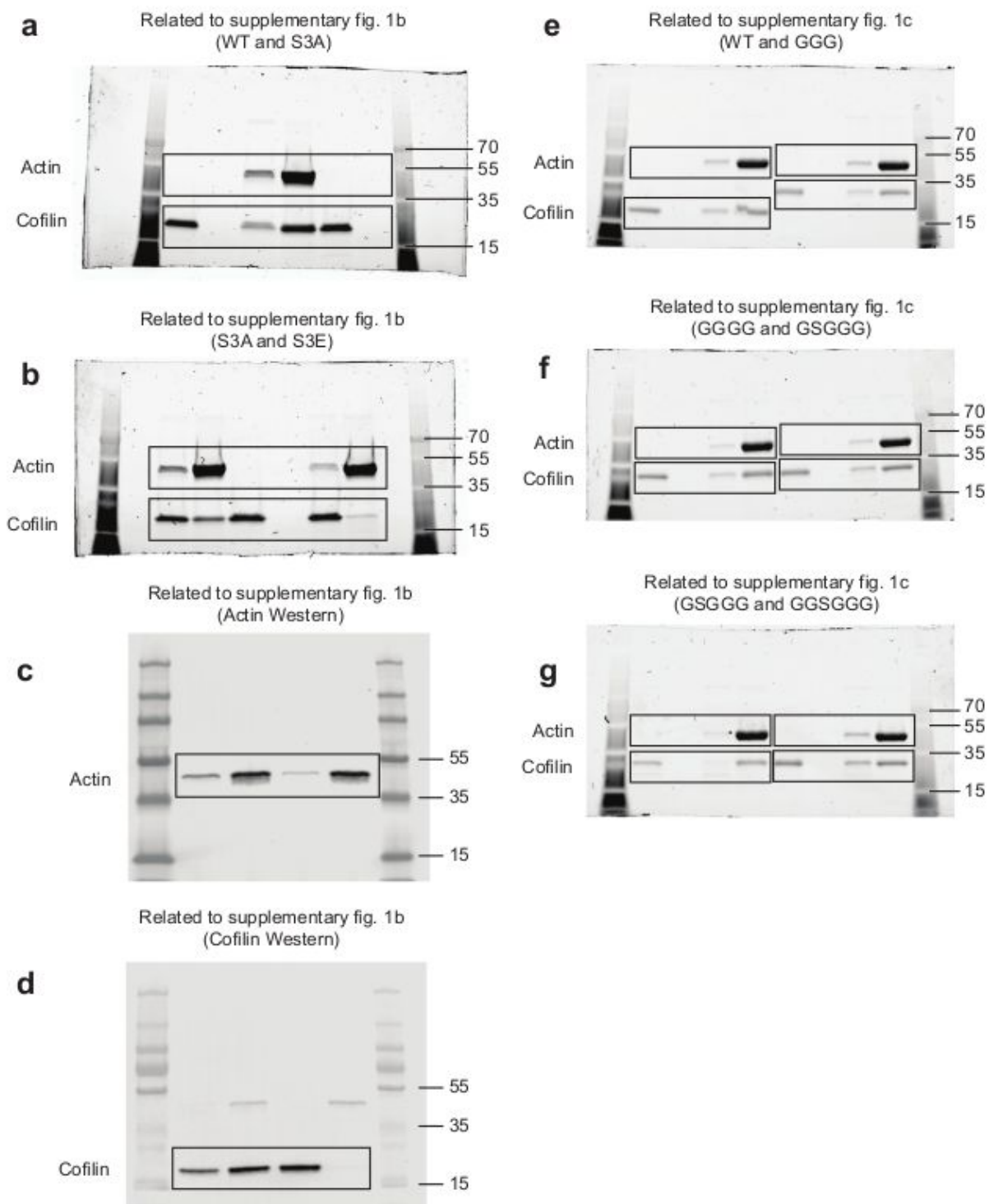
interaction with F-actin. F-actin and purified cofilin were incubated for 30 minutes at room temperature prior to ultracentrifugation. Reactions were run at pH 7.5 to prevent cofilin severing and generation of G-actin. Values were from single measurements. (See full gels in Figure 2.6)

c. Actin co-sedimentation assay for Zdk2-cofilin fusion proteins with different linkers. Values were from single measurements. (See full gels in Figure 2.6)

d. Measurement of F-actin to G-actin conversion for Zdk2-cofilin fusion proteins with different linkers. F-actin and purified cofilin were incubated for 2 hours at room temperature. Samples were tested at pH 6.8 and 7.5 to measure ability of cofilin to convert F-actin to G-actin. Values were from single measurements. (See full gels in Fig. 2.7)



Figure 2.6 Original images of blots and stain-free gels used for Supplementary Fig. 2.1b and 2.1c.

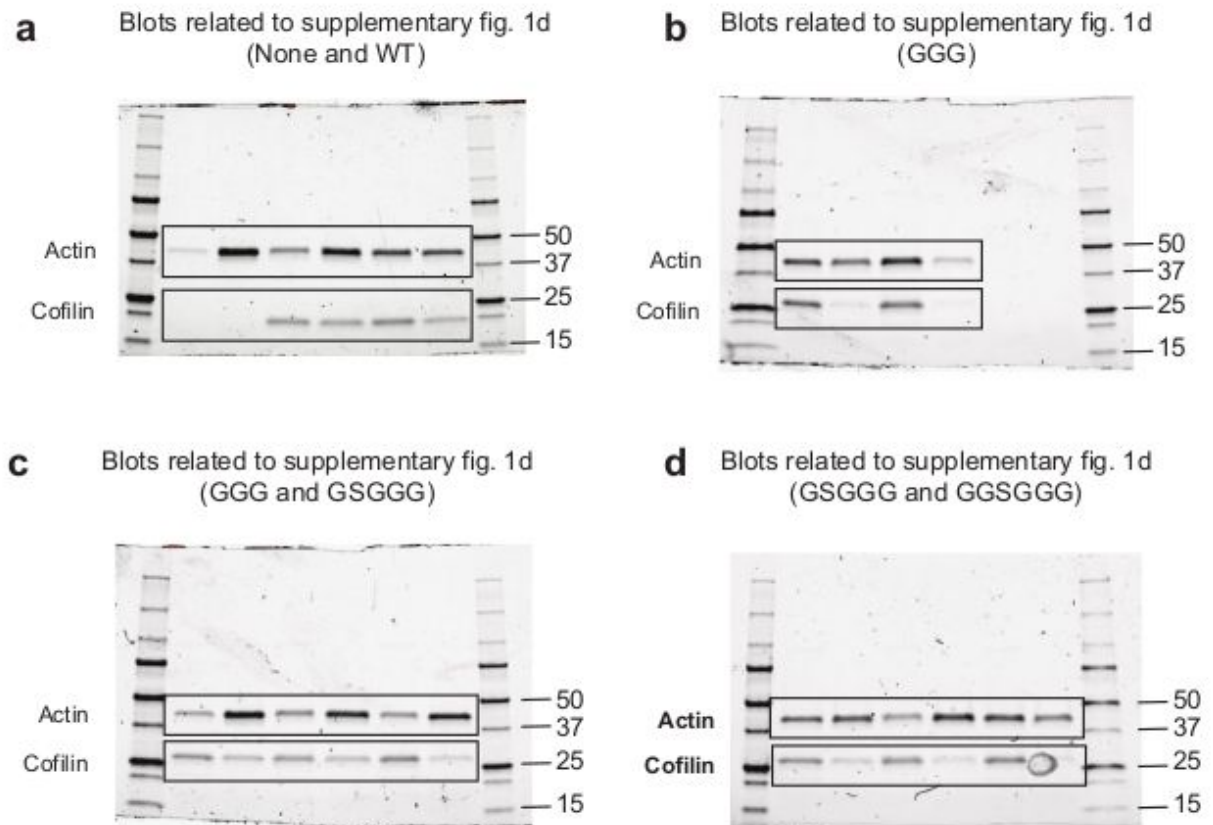




Box is drawn around area used for figure.

- a. Blots related to supplementary fig. 2.1b (WT and S3A)
- b. Blots related to supplementary fig. 2.1b (S3A and S3E)
- c. Blots related to supplementary fig. 2.1b (Actin)
- d. Blots related to supplementary fig. 2.1b (Cofilin)
- e. Blots related to supplementary fig. 2.1c (WT and GGG)
- f. Blots related to supplementary fig. 2.1c (GGGG and GSGGG)
- g. Blots related to supplementary fig. 2.1c (GSGGG and GGSGGG)

Figure 2.7. Original images of blots and stain-free gels used for Figure 2.5.



Box is drawn around area used for figure.

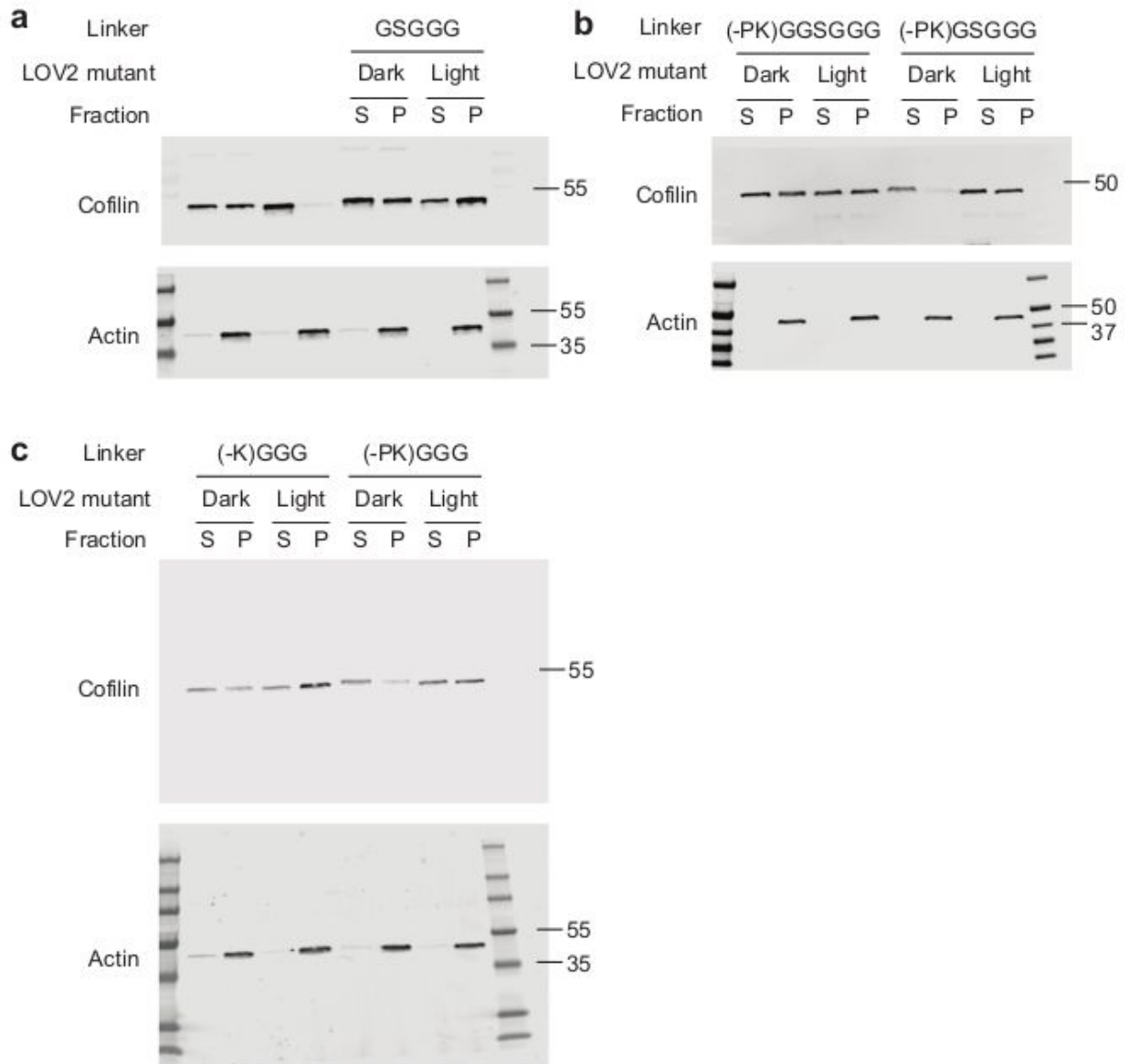
a. Blots related to supplementary fig. 2.1d (None and WT).

b. Blots related to supplementary fig. 2.1d (GGG).

c. Blots related to supplementary fig. 2.1d (GGG and GSGGG).

d. Blots related to supplementary fig. 2.1d (GSGGG and GGSGGG).

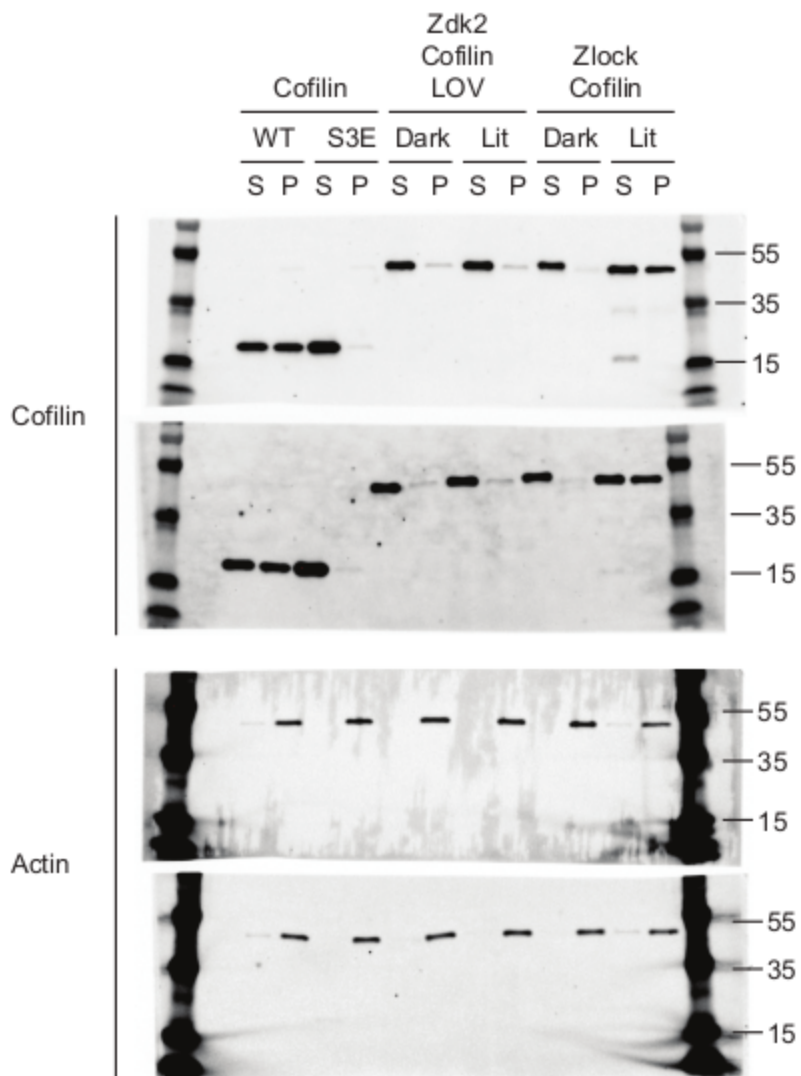
Figure 2.8. Original images of blots used for Fig. 2.2c.



Western blot data for actin co-sedimentation assay. Wild-type cofilin and non-binding S3E cofilin mutant are shown as controls. Zdk2-cofilin-LOV and Z-lock cofilin are identical except Z-lock cofilin contains a Zdk2 mutant (I32F) that reduces affinity for LOV2. Designs were tested using both LOV2 dark and lit mutants to measure light-dependent changes in F-actin binding. Band density for cofilin in both the

supernatant (S) and pellet (P) fractions was quantified and used to calculate F-actin binding as described in methods and displayed as a bar graph in Fig. 2.2c. Data for two independent experiments shown here (the other one in Figure 2.2b).

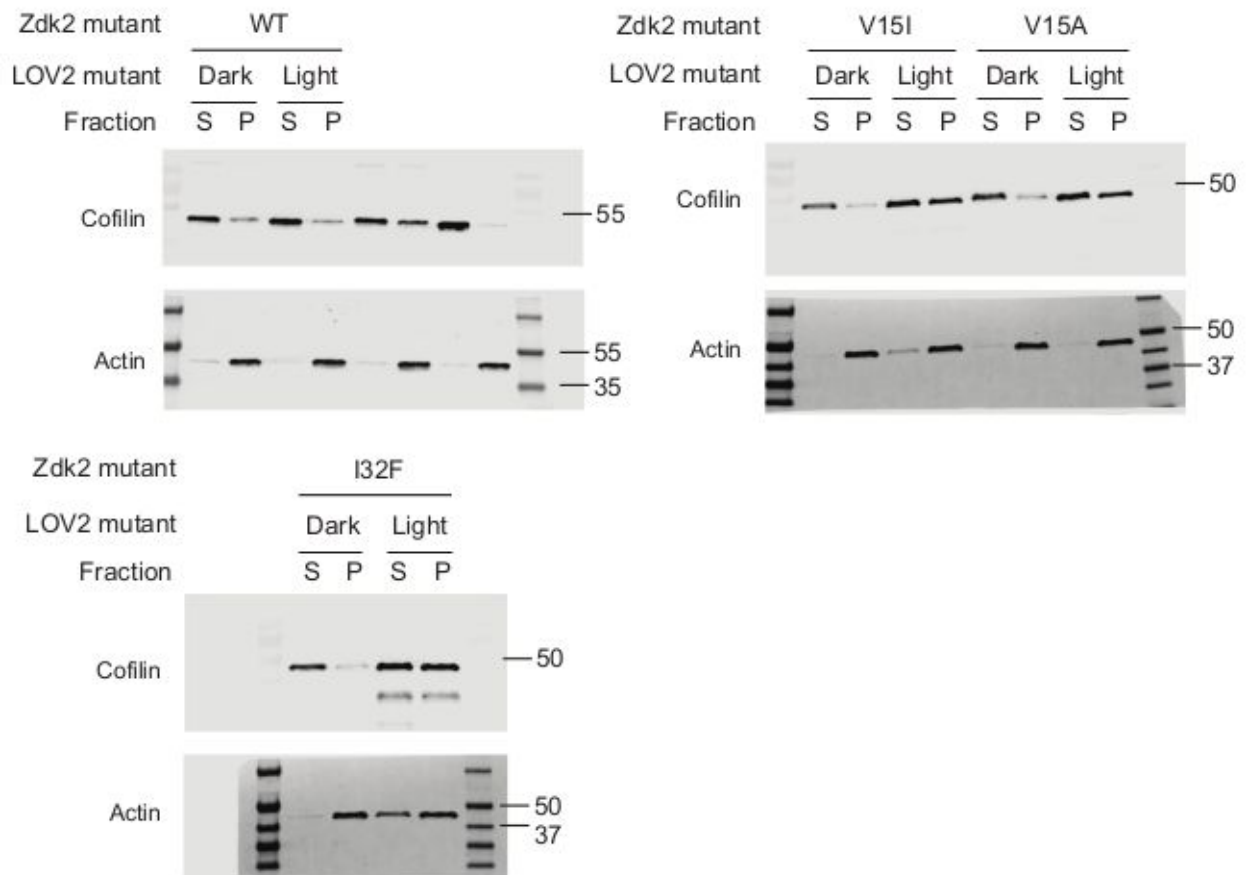
Figure 2.9. Original images of blots used for Fig. 2.2a.



Western blot data for actin co-sedimentation assay is shown for Zdk1-cofilin-LOV designs with indicated linkers connecting Zdk1 and cofilin. Designs were tested using both LOV2 dark and lit mutants (Supplementary Table 2.2) to measure light-dependent changes in F-actin binding. Band density for cofilin in both the supernatant (S) and pellet (P) fractions was quantified and used to

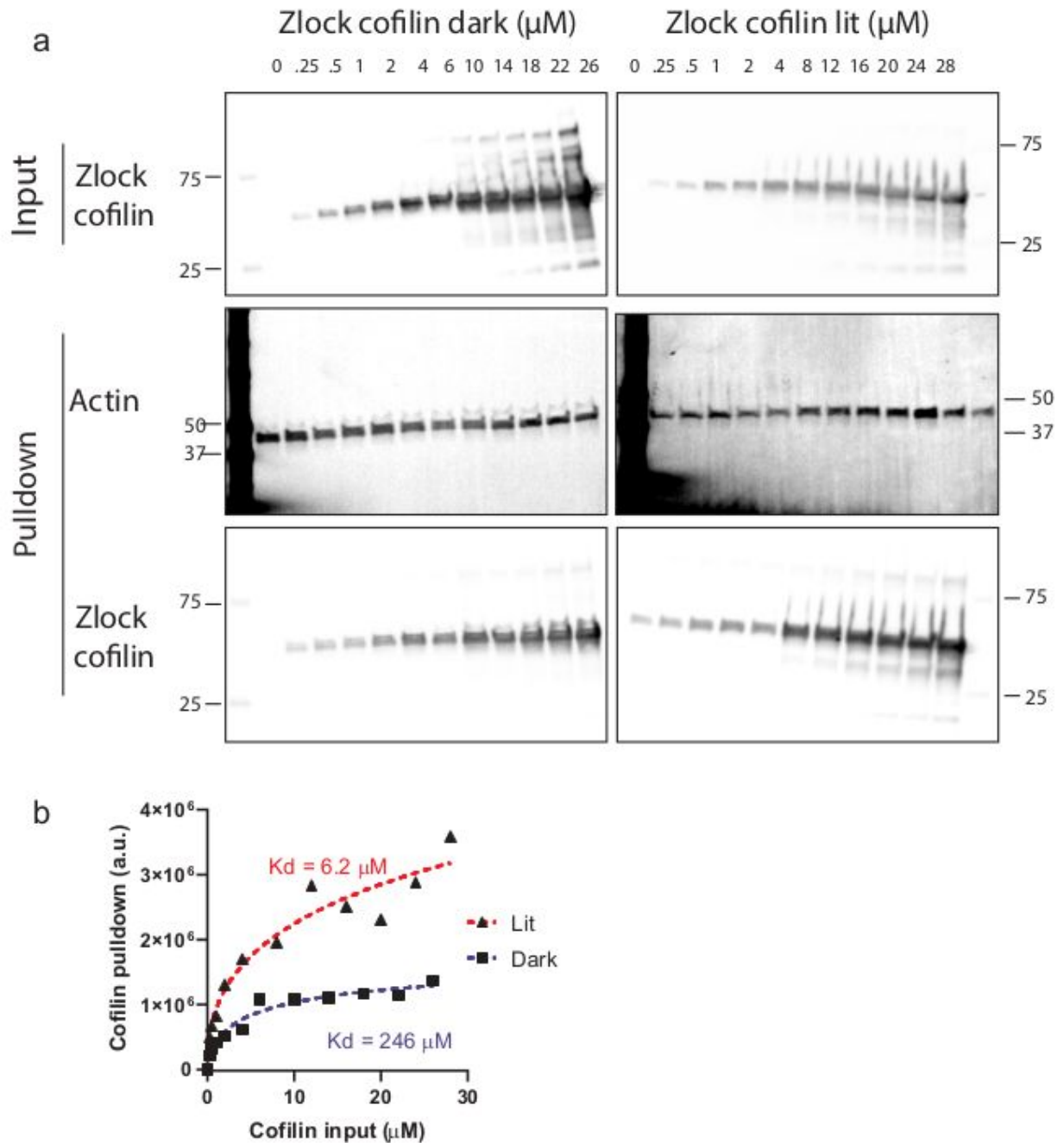
calculate F-actin binding as described in methods and displayed as a bar graph in Fig. 2.2a. In this initial screening study, each variant was tested once.

Figure 2.10. Original images of blots used for Fig. 2.2g.



Western blot data for actin co-sedimentation assay is shown for Zdk2-cofilin-LOV designs with indicated Zdk2 mutants. Designs were tested using LOV2 dark and lit mutants to measure light-dependent changes in F-actin binding. Band density for cofilin in both the supernatant (S) and pellet (P) fractions was quantified and used to calculate F-actin binding as described in methods and displayed as a bar graph in Fig. 2.2g. In this initial screening study, each variant was tested once.

Figure 2.11. Actin co-sedimentation assay used to determine the affinity of Z- lock cofilin and F-actin.



Actin was at 2  $\mu\text{M}$ . Mutants of Z-lock cofilin (dark and lit) were purified (Methods) and used to

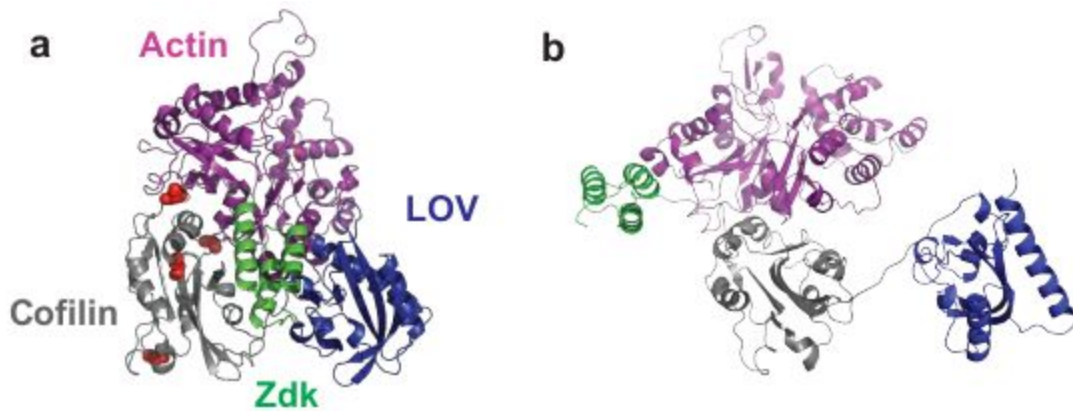


mimic the inactive and active states.

a. Increasing amounts of Z-lock cofilin were incubated with Actin for 30 minutes at room temperature, and 20% of the reaction mixture was examined using SDS-page (Input). The rest was centrifuged down at 112,000 x g for 1.5 hour, and the pellet was examined using Western blot. (Three independent experiments)

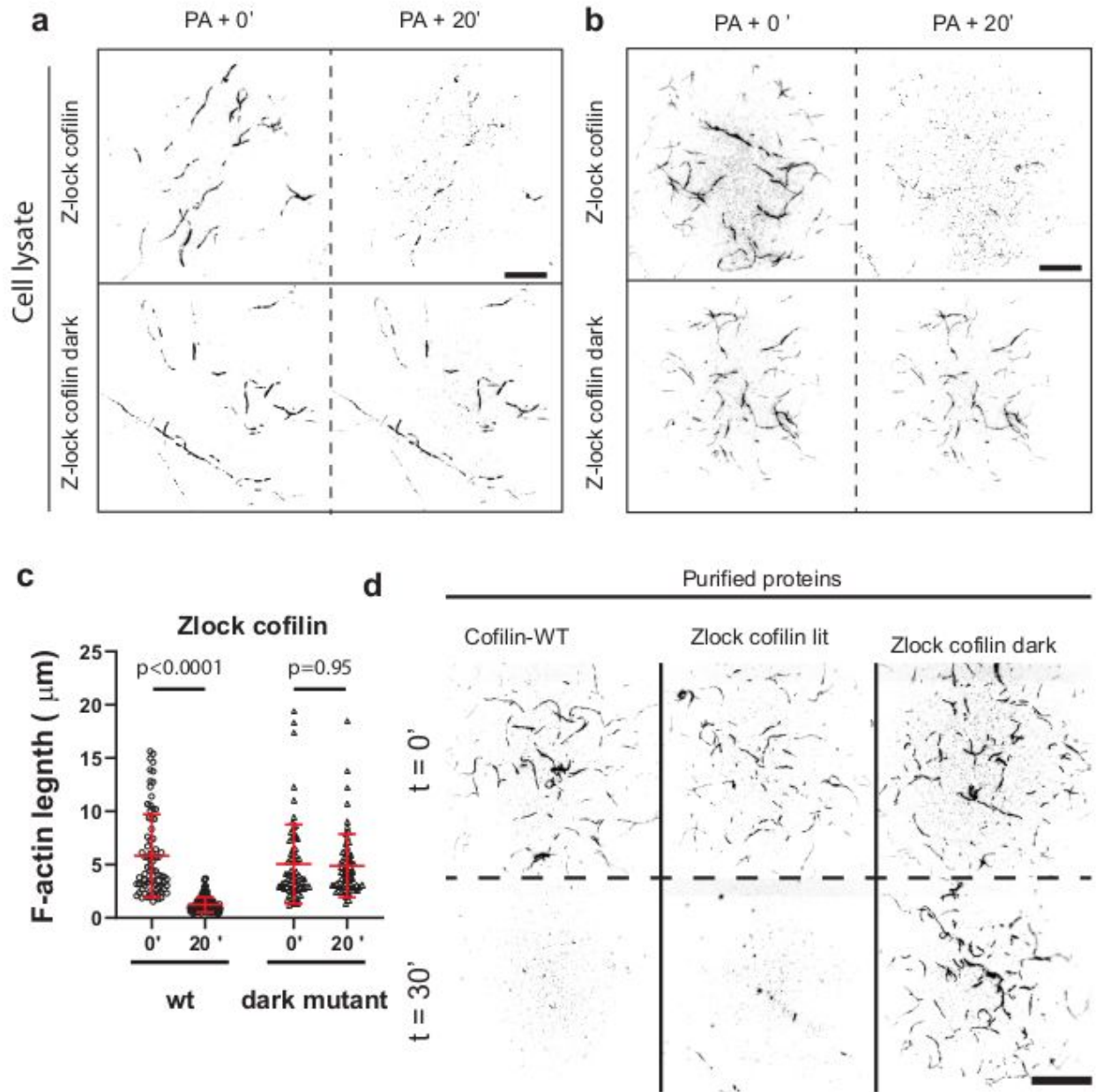
b. The amount of Cofilin in the pellet was plotted against the input of Cofilin. The data was fit to a Hill equation (dashed line). The measured Kds for the dark and lit states were  $285 \pm 44 \mu\text{M}$  and  $4.5 \pm 2.4 \mu\text{M}$ , from three independent experiment. Data is shown as mean  $\pm$  SD.

Figure 2.12. Rosetta prediction of Z-lock cofilin structure in the dark and lit state



- a. In the dark state, LOV interacts with Zdk, occluding the Actin binding site on Cofilin. The phosphorylation sites (red spheres) on Cofilin are not occluded.
- b. In the lit state, LOV dissociates from Zdk, exposing the Actin binding site on Cofilin.

Figure 2.13. F-actin severing by Z-lock cofilin in-vitro



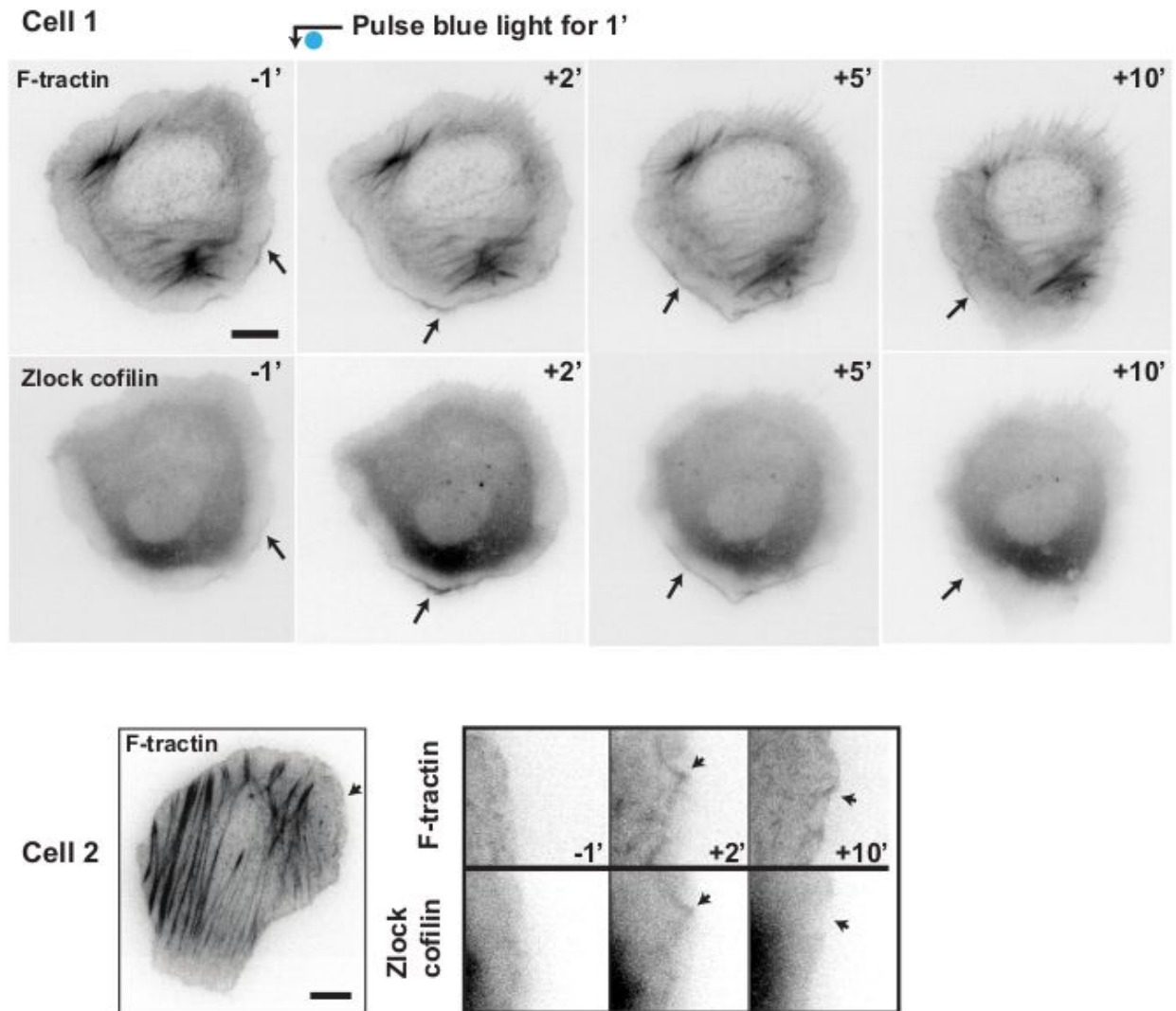
a. b. Polymerized F-actin was deposited on a coverglass (methods). Cell lysate from MTLn3 cells expressing either Zlock cofilin dark state mutant or Zlock cofilin was added. Fluorescent images of F-actin were taken before and after 20 minutes of irradiation. Two independent experiments. Here showing two randomly selected

representative fields of view. Twenty fields of view were examined for each condition.

c. Length of F-actin was measured before (wt: n=80; dark mutant: n=71) and after (n=65) light activation. Data is shown as mean  $\pm$  SD.

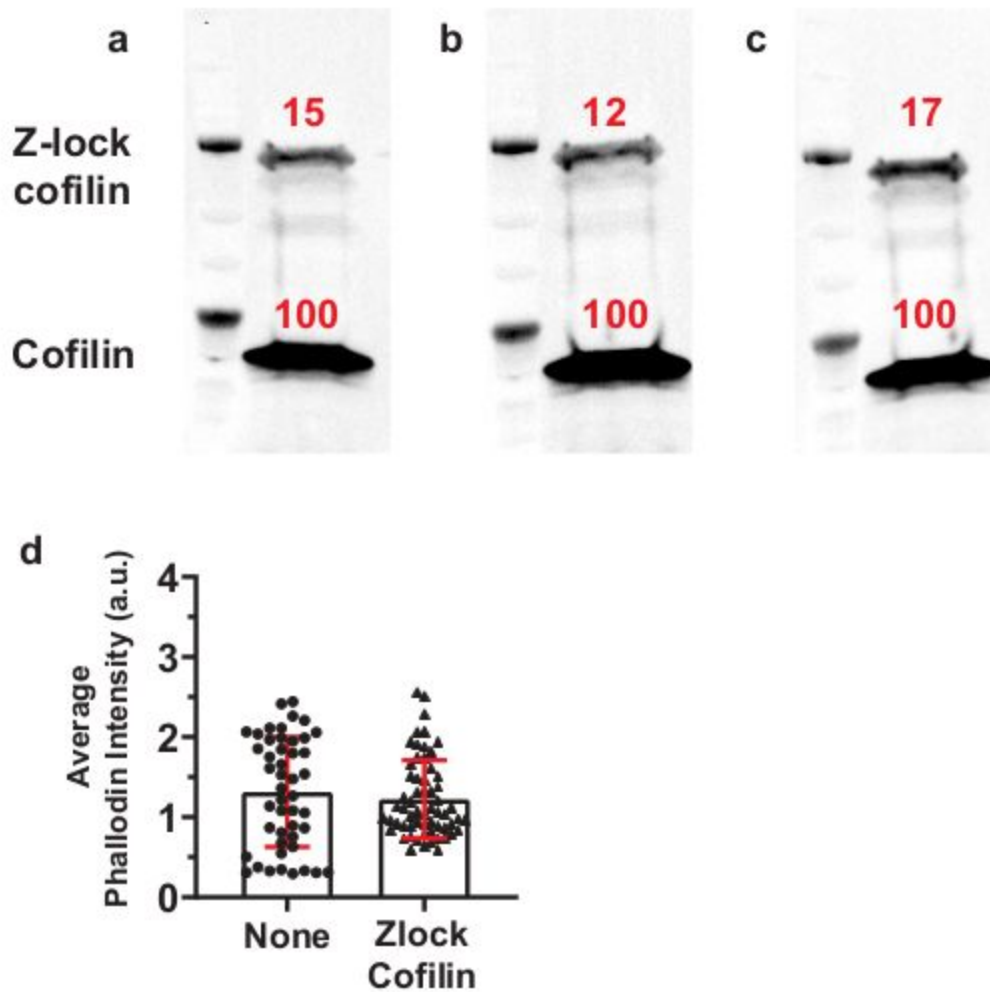
d. Polymerized F-actin was deposited on a coverglass. Purified Cofilin, Z-lock cofilin lit and dark state mutants were added (Supplementary Table 2). Fluorescence images of F-actin were taken at 0 and 30 minutes after addition. Two independent experiments.

Figure 2.14. Z-lock cofilin reversibly translocates to the cell edge upon activation.



MTLn3 cells were transfected with Z-lock cofilin and F-tractin. The cells were activated with blue light for 1 minute globally, causing Z-lock cofilin to move to cell edge, where it co-localized with F-actin filaments. 9 cells from three independent experiments were examined. Scale bar: 10  $\mu$ m.

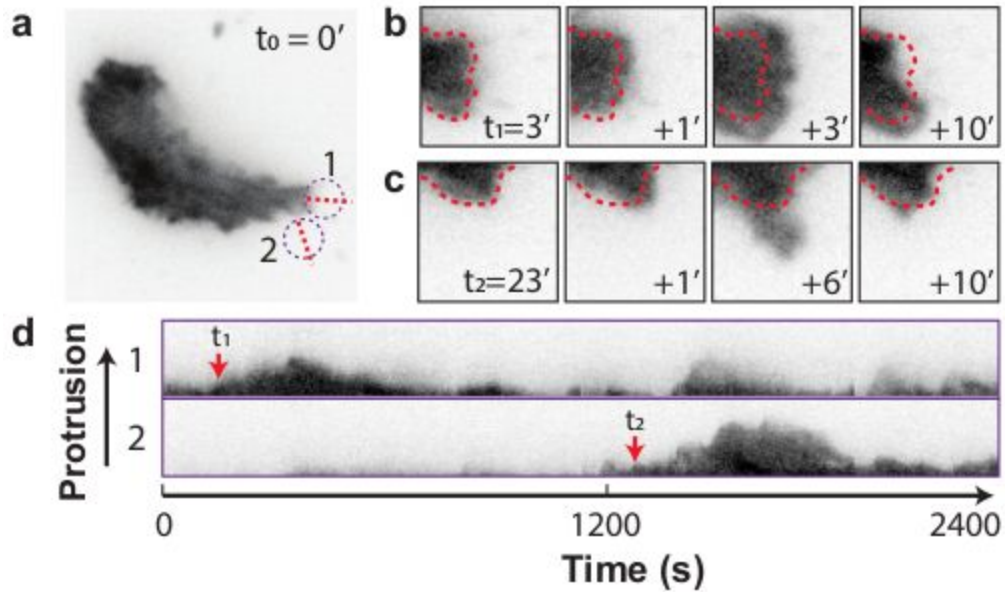
Figure 2.15. Expression of Z-lock cofilin doesn't perturb F-actin level in MTLn3 Cells.



a.b.c.) MTLn3 cells were transfected with Z-lock cofilin overnight. The levels of endogenous Cofilin and Z-lock cofilin were examined by Western blot. Percent endogenous cofilin level is shown. (three independent experiments)

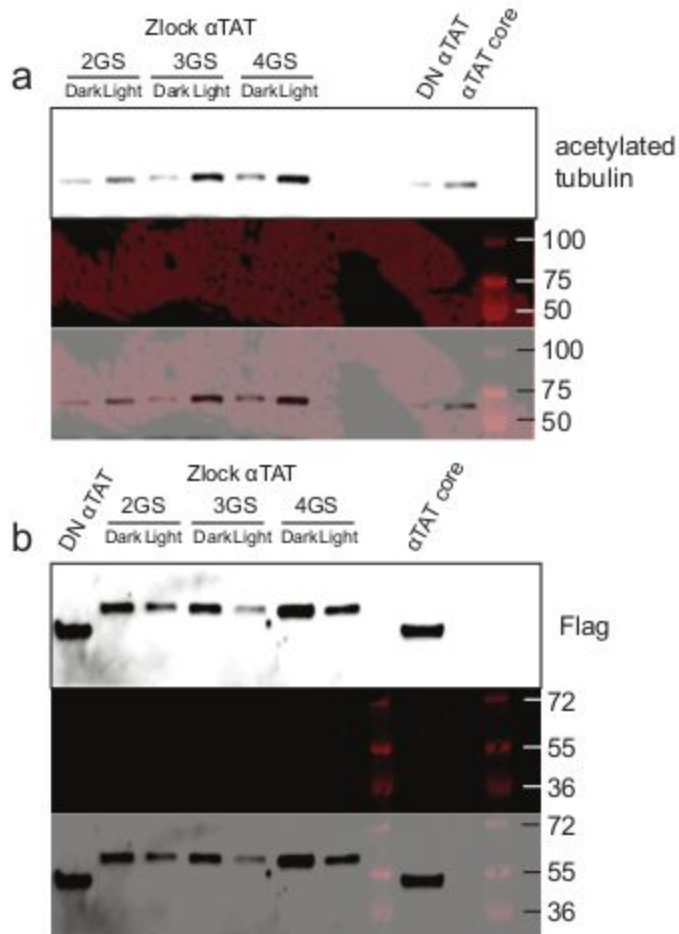
d. F-actin level (a.u.) of individual cells with and without expression of Z-lock cofilin (with:  $503 \pm 262$ ,  $n=47$ ; without:  $468 \pm 186$ ,  $n=61$ ). Data is shown as mean  $\pm$  SD.

Figure 2.16. Reversibility of Z-lock cofilin



a. MTLn3 cells were transfected with Z-lock cofilin and yPet-Stargazin (a membrane marker). Cells were first imaged for 3 minutes. Then, in the purple circle, cells were irradiated with pulsed blue light for 1 min (500 ms irradiation every second). b. Protrusions formed at the cell edge and retracted after irradiation ceased. Cell edges were marked with red-dashed lines. c. Spot 2 in a was activated after spot 1 at the indicated time. Another protrusion formed transiently and retracted afterwards. d. Kymograph along lines in a. Three cells were examined.

Figure 2.17. Original images of blots and stain-free gels used for Fig. 2.4b.



a. Top panel = acetylated tubulin blots. Middle panel = same gel imaged with different wavelengths to show molecular weight markers. Bottom panel = overlay of above two images. (three independent experiments)

b. Top panel = FLAG tag blots. Middle panel = same gel imaged with different wavelengths to show molecular weight markers. Bottom panel = overlay of above two images. (three independent experiments)



## CHAPTER 3: A COMPUTATIONAL PROTOCOL FOR INHIBITING PROTEIN BINDING REACTIONS WITH A LIGHT SENSITIVE PROTEIN DIMER

### 3.1 Introduction

Light-responsive proteins offer spatiotemporal control of protein activity inaccessible to drug-based approaches (Camporeze et al., 2018; Govorunova, Sineshchekov, & Spudich, 2019; Kodani, Soya, & Sakurai, 2019; O'Banion & Lawrence, 2018; Repina, Rosenbloom, Mukherjee, Schaffer, & Kane, 2017), allowing biological processes to be rapidly and reversibly perturbed, in seconds to minutes, in specific locations in a cell or tissue. This has been widely used to interrogate the function of neurons (Assaf & Schiller, 2016; Chang, Chang, & Shyu, 2017), and certain of these tools, particularly channelrhodopsins (Hegemann & Nagel, 2013), have been more broadly applied to controlling protein function with light. Engineered light-activatable proteins have been used to localize proteins to specific cellular components (L. E. Fenno & Deisseroth, 2014; H. Wang & Hahn, 2016; H. Wang et al., 2016; Wehler, Niopek, Eils, & Di Ventura, 2016), oligomerize them (Govorunova et al., 2019; Zimmerman et al., 2016), or control their activity (Jäschke, 2012; Liu & Tucker, 2017), as with the photoactivatable Rac1 (PA-Rac) system (Wu et al., 2009). In PA-Rac, the effector binding surface of the GTPase Rac1 domain is sterically occluded from binding effectors by the light sensitive Light Oxygen Voltage (LOV2) domain from the protein phototropin, which is fused to the

N-terminus of Rac1. When irradiated by 450 nm light, the J $\alpha$  helix at the C-terminus of the LOV2 domain rapidly becomes disordered ( $< 1$  sec), freeing the LOV2 domain from the Rac1 surface and exposing the effector binding site. The process is reversible with the switch returning to the closed state in the dark with a half-time of approximately 30 seconds (Pudasaini, El-Arab, & Zoltowski, 2015). If control on different time scales is needed, there exist mutations to the LOV2 domain that shorten or lengthen the half-life in the lit state (Möglich et al., 2009; Wu et al., 2009; Yazawa, Sadaghiani, Hsueh, & Dolmetsch, 2009). A crystal structure of PA-Rac revealed that the LOV2 domain was able to block effector binding because specific interactions between the surface of the LOV2 domain and the surface of Rac1 helped to pin Rac1 against the LOV2 domain in the dark (Wu et al., 2009). To control the activity of the GTPase Cdc42 with the same approach, it was necessary to redesign the surface of Cdc42 so that similar favorable interactions could be made with the LOV2 domain.

Recently, we developed a general system, called Z-lock, for photocaging proteins with the LOV2 domain that does not require the formation of a specific interface with the protein that is being caged (Stone et al., 2019). Z-lock makes use of a small protein domain, zDK, which we previously engineered to interact with the LOV2 domain in the dark but not in the light. With Z-lock, the LOV2 domain is fused to one terminus of the protein that is being caged and zDK is fused to the other terminus. If the linkers connecting the two domains to the termini of the protein are an appropriate length, zDK interacts with the LOV2 domain in the dark and forms a

complex over the surface of the protein. In the light, zDK releases from the LOV2 domain, exposing the surface of the protein to its binding partners. If the linkers are too short, the LOV2 domain and zDK are unable to interact in the dark and there is little caging. If the linkers are longer than needed, it is more likely that the zDK/LOV2 complex will form in a location that does not sterically occlude the target binding surface. Here, we describe a protocol we have developed in the molecular modeling program Rosetta to aid in the design of linkers for Z-lock. In addition to revealing which linker lengths are most appropriate for caging, the Rosetta protocol determines if the protein termini are appropriately positioned to place the zDK/LOV2 complex over the target binding surface. We first benchmark the computational protocol with data we previously obtained when caging the proteins cofilin and  $\alpha$ TAT with Z-lock, and then use it to design a photosensitive binder of the GTPase Cdc42.

### 3.2 Results

In the Z-lock design protocol, a model is first constructed with zDK and LOV2 fused to the termini of the protein of interest (POI) with the linkers adopting random conformations and zDK not binding with the LOV2 domain. In order to build this model it is necessary to have a high resolution structure or high confidence homology model of the POI. There are two variants of zDK, zDK1 and zDK2, that can be used with the Z-lock system. With zDK1 the N-terminus of the LOV2 domain must be fused to the C-terminus of the POI as the LOV2 domain C-terminus is buried at the interface in the zDK1/LOV2 complex. With zDK2, the LOV2 domain and zDK2 can be fused to either terminus of the POI.

The second stage of the protocol is to perform structure prediction to determine if the linkers are long enough to allow zDK to interact with the LOV2 domain and to determine if the zDK/LOV2 complex is likely to block the surface of interest on the POI. Structure prediction is performed with Monte Carlo sampling with the Rosetta low resolution (centroid) energy function combined with distance constraints to direct the formation of a native-like complex between zDK and LOV. Each Monte Carlo move is either a random perturbation to backbone torsion angles in the two linkers or a fragment insertion. During a fragment insertion, the backbone torsion angles for three consecutive residues in the linker are replaced with torsion angles values pulled from 3-residue pieces of naturally occurring proteins that have similar sequences to the linker that is being designed. Following the Monte Carlo sampling, gradient-based minimization of the torsion angles in the linker is used to fall into a local minimum. The conformations of the individual domains (zDK, LOV2 and the POI) are held constant during the simulation. Many independent trajectories are performed, each starting from different random linker conformations, to map out the range of linker structures that allow for the zDK/LOV2 complex to form. If the linkers are too short for zDK to reach LOV, then none or very few of the trajectories will end up with zDK docked against LOV, even though distance constraints are being used to drive formation of the complex.

To predict if Z-lock is likely to block binding of the POI to its binding partner, we have developed two simulation strategies. The first approach is employed in cases where a high resolution structure (or homology model) is available of the

complex that will be inhibited with Z-lock. In this case, two sets of structure prediction simulations of the zDK/LOV2 linkers are performed: one set with the POI bound to its binding partner and one set with the binding partner absent. During the simulations, if the zDK/LOV2 interaction can form in the absence of the binding partner but can not form in the presence of the binding partner, then it is a good indication that Z-lock will block the binding event in the dark state. This happens in scenarios where the linkers are not long enough to allow the zDK/LOV2 complex to form while the binding partner is present, and indicates that the two binding events are in competition with each other. Our second simulation strategy applies in cases where a structure is not available of the POI bound to its partner, but there is information about which residues on the surface of the POI are important for binding to its partner. In this case, structure prediction is just performed with the POI fused to zDK and LOV2 and the resulting models are interrogated to determine of the zDK/LOV2 complex forms adjacent to the residues on the POI that are important for binding to its partner.

To search for effective Z-lock linkers, structure prediction simulations are performed with a panel of linker lengths going from short to long. As we are not trying to enforce specific conformations upon the linkers, we have computationally and experimentally tested linkers consisting only of serines and glycines. Because of the varied shapes of the POI, zDK and LOV, it can also be helpful to test asymmetric linkers lengths, i.e. where the zDK-POI and the POI-LOV2 linkers are not the same length as each other. Many structure prediction trajectories are performed with each

set of linker lengths and the linker lengths that promote formation of the zDK/LOV2 complex over the desired surface of the POI are good candidates for experimental studies.

To test our computational protocol we performed simulations with three systems that have been caged with Z-lock: the binding of cofilin to actin, the binding of  $\alpha$ -tubulin acetyltransferase ( $\alpha$ TAT) to tubulin, and the binding of a CBD domain to the GTPase Cdc42. We recently reported the experimental characterization of the cofilin and  $\alpha$ TAT switches [Stone et al., 2019], while the caged Cdc42 binder is a new switch that we experimentally characterize here.

*Caging cofilin.* The first optogenetic tool that we examined with our modeling protocol used Z-lock to prevent non-muscle cofilin from binding to actin, which causes depolymerization and severing of actin filaments as part of cytoskeletal remodeling [Carlier, Ressad, & Pantaloni, 1999; Condeelis, 2001]. In vivo, this serves to convert old actin filaments to monomeric actin [Loisel, Boujemaa, Pantaloni, & Carlier, 1999] for re-integration into new filaments, maintaining G-actin levels to enable cell motility [Loisel et al., 1999; Svitkina & Borisy, 1999]. The actin/cofilin complex has an interface area of  $1162 \text{ \AA}^2$  as determined by CoCoMaps [Vangone, Spinelli, Scarano, Cavallo, & Oliva, 2011], with the primary structural component on the cofilin side of the interface being a single alpha helix running from residues 263 to 278 and the beta hairpins flanking it. A line drawn between the terminal residues of cofilin would pass through one end of the helix

forming the majority of the interface, suggesting that the actin interface may be caged by constraining the optogenetic components to the region directly between the termini.

In the development of Z-lock cofilin three sets of linkers with varying lengths were experimentally tested: zDK2-GSGGG-Cofilin-GSG-LOV2 (linker set 1X), zDK2-GSGGG-Cofilin-GGSGG-LOV2 (2X) and zDK2-GSGGG-Cofilin-GGSGGSGG-LOV2 (3X). Binding to actin in the dark state and lit state was initially characterized using mutations to the LOV2 domain that mimic the photo-activated (A532E I536E)(Harper, Christie, & Gardner, 2004) and photo-inactivated (C450A)(Richter et al., 2005) state of the LOV2 domain. Binding to actin was measured using co-sedimentation experiments. In these experiments a mixture of the switch and actin were centrifuged at high speeds and then the sediment was probed to determine what fraction of the switch precipitated along with the actin. All three linker variants of Z-lock cofilin bound more tightly to actin in the lit state, but the strength of binding in the dark state varied significantly. With the shortest linker (1X), 51% of the Z-lock molecules co-sedimented with actin in the dark state, while 31% and 15% co-sedimented with 2X and 3X respectively. These results indicated that the shorter linkers were not providing strong caging in the dark, possibly because the linkers were not long enough to allow the zDK/LOV2 interaction to readily form. The 3X form of the switch was carried forward for additional experimental studies, and was shown to provide light-responsive control of cofilin activity in living cells.

To determine the capacity for our modeling protocol to predict the effect of different length linkers on the caging of Z-lock cofilin, we ran 10,000 independent simulations of the zDK2-cofilin-LOV2 construct in the presence and absence of actin and tabulated what fraction of the simulations produced models with zDK2 bound to LOV2. We refer to this as the closed state. To be considered closed, a model needed to satisfy an energy filter to exclude non-physical models and a constraint filter to verify that zDK2 and LOV2 adopted relative positions consistent with the zDK2-LOV2 co-crystal structure 5DJT; models that passed the first check but failed the second were considered open. The fraction of low-scoring conformations that were closed was compared for the simulations in the absence of actin and for the simulations in the presence of actin. Ideal linkers should allow the zDK2/LOV2 interaction to form (i.e. close) in the absence of actin, but not allow the interaction to form in the presence of actin. An absence of closed models in the presence of actin indicates that the formation of the zDK2/LOV2 interaction and the binding of actin are in competition with each other, a necessary condition for regulating actin binding. To further explore the effects of longer and shorter linker lengths than were experimentally tested, we also computationally tested a single glycine as the cofilin-LOV2 linker (1X), as well as GGSx4(4X) and GGSx5(5X), consisting of the 3X linker set with one and two additional SGG repeats on the cofilin-LOV2 linker respectively.

In the simulations, the 3X linker set showed the largest decrease in interface closure caused by the presence of actin [Figure 3.2]. The 2X linker also



demonstrated some caging capacity, while very little caging was predicted for 1X and 0X. The simulations indicate that the 1X and 0X the linkers are too short to allow the zDK2/LOV2 interaction to form, even in the absence of actin. These predictions are consistent with the co-sedimentation experiments that showed that the 1X linker set binds strongly to actin even in the dark state, and that 3X is more strongly caged in the dark than 2X. The molecular modeling also predicts that 3X will more effectively block actin binding than switches with longer linkers, 4X and 5X. In the case of 4X and 5X, the linkers are long enough that it becomes possible to simultaneously form a complex with actin and form the zDK2/LOV2 complex.

*Caging  $\alpha$ TAT.* Alpha-tubulin acetyltransferase ( $\alpha$ TAT) preferentially acetylates microtubules over free tubulin and promotes microtubule degradation (Kalebic et al., 2013). To create Z-lock  $\alpha$ TAT, zDK1 was fused to the N-terminus of the catalytic domain of  $\alpha$ TAT and the LOV2 domain was fused to the C-terminus. A fixed length linker with 5 gly-ser repeats was placed between zDK1 and  $\alpha$ TAT and three different linkers were experimentally tested for connecting the C-terminus of  $\alpha$ TAT to LOV: a 4 residue linker with 2 gly-ser (GS2) repeats, a 6 residue linker with 3 gly-ser (GS3) repeats, and an 8 residue linker with 4 gly-ser (GS4) repeats. The Z-lock variants were tested by probing levels of microtubule acetylation in 293T<sub>Lin</sub>XE cells in the dark versus light. All three switches showed some light-dependent activity, with the GS3 linker showing the largest fold change in acetylation between the dark and lit state. The GS2 linker demonstrated higher background activity in the dark than the

GS3 linker. To determine if our modeling protocol could predict this behavior we ran simulations with the GS2, GS3 and GS4 linkers as well as a single GS and a 5GS linker. The GS and GS5 linkers were simulated to more completely map out what is occurring when the linkers are shortened or lengthened.

Unlike with cofilin, there is no crystal structure of  $\alpha$ TAT bound to its substrate. However, there is a structure of the  $\alpha$ TAT catalytic domain (Friedmann, Fan, & Marmorstein, 2012) and mutational studies have identified a set of surface residues located in a large groove that are critical for interacting with microtubules. To computationally test which Z-lock constructs would be most effective at regulating acetylation we ran 10000 independent structure prediction trajectories for each linker variant and examined what fraction of the trajectories resulted in models consistent with blocking binding to tubulin. To be consistent with blocking tubulin binding the model needed to have a favorable energy (i.e. few steric clashes), have zDK1 and LOV2 appropriately docked against each other, and be adjacent to the tubulin binding site on  $\alpha$ TAT. We observed clear differences between the linkers in the simulations. The GS and GS2 linkers rarely blocked binding because the linkers were not long enough to allow the zDK1/LOV2 interface to form. This result is consistent with the increased background activity observed experimentally for GS2 in the dark. The GS3 and GS4 linkers most consistently caged the interface, and the GS5 linker resulted in weaker caging because the zDK1/LOV2 complex more often formed in locations that were not adjacent to the tubulin binding site on  $\alpha$ TAT. Overall, the results were in agreement with the trends observed experimentally.

When we originally engineered and tested Z-lock  $\alpha$ TAT, we hypothesized that Z-lock may also regulate the binding of the cofactor, acetyl CoA, to the enzyme. In the Rosetta simulations we did not observe zDK1/LOV2 associating over the acetyl CoA binding site, and we did not observe any strain on the protein that would allosterically regulate acetyl CoA binding. These results suggest that perturbation of cofactor binding is not important to the mechanism of the switch.

*Caging CBD.* To test our linker design protocol on a case without prior experimental data, an inhibitor was constructed around the CDC42-binding CBD domain of WASP, which plays a role in establishing cell polarity in complex with CDC42 and Par3 (Marques & Klefström, 2015) as well as chromosomal protein composition (Dormoy, Tormanen, & Sutterlin, 2013) and more general regulation of cell polarity (Ozdamar, 2005; Thiery & Huang, 2005). To optogenetically control the affinity of the CBD domain to CDC42, zDK2 was added to its N terminus and LOV2 to its C terminus, with glycine-serine dimers placed between the zDK2 and CBD domains. We computationally predicted the effects of adding zero to five glycine-serine dimers. The CBD-based switch is different than the cofilin and  $\alpha$ TAT switches in that the CBD domain is partially disordered when it is not bound to CDC42. Cofilin and  $\alpha$ TAT are well folded in the presence and absence of their binding partners. The intrinsic flexibility of the CBD domain provides an alternative mechanism for controlling its activity with Z-lock. When bound to CDC42, the N-terminal region of the CBD domain forms a long beta strand that partners with a beta strand on the

surface of CDC42. When the CBD domain is not bound to CDC42, this beta strand is intrinsically disordered. We hypothesized that Z-lock could block binding to CDC42 by not only forming a steric block, but also by constraining the flexible N-terminal region of the CBD domain in a distorted, non-binding conformation. To allow for this mechanism of control in our simulations, we allowed the N-terminal beta strand of the CBD domain to adopt alternative conformations in our simulations of the switch not bound to CDC42.

As with cofilin, two types of simulations were performed: simulations with Z-lock Wasp CBD bound to its binding partner, CDC42, and simulations of the unbound state. In the simulations all of the switch variants could form the zDK2/LOV2 complex in the absence of CDC42 and the construct with no additional linker residues (GS0) was least susceptible to forming it in the complex state. The GS0 construct was expressed in LinXe cells and pulldown experiments were used to measure binding to CDC42 in the dark and with blue light illumination [Figure 4]. When zDK2/LOV2 was expressed at lower levels, over a 5-fold reduction in binding to CDC42 was observed in the dark state compared to the lit state.

### 3.3 Discussion

A challenge in optimizing the linkers for the Z-lock system is the ambiguous nature of negative experimental results. Linkers that are too short do not permit zDK and the LOV2 domain to interact and so the switch is permanently “on”. Linkers that are too long allow the zDK/LOV2 interaction to form in locations that do not block binding to the protein of interest and therefore the switch is also constitutively “on”.

Here, we have developed and benchmarked a molecular modeling protocol that can distinguish between the two scenarios and indicates which linker lengths should provide maximal caging with Z-lock. Although it was not the case for the three systems studied here, the modeling should also provide an indication of whether a protein is amenable to caging with Z-lock. If no combination of linker lengths favor the zDK/LOV2 complex forming over the protein binding surface of interest, than that protein is not a good candidate for caging with Z-lock.

The modeling protocol we have developed uses standard conformational sampling strategies in Rosetta and relies on Rosetta's "centroid" mode. In centroid mode each amino acid side chain is treated as a single sphere and detailed side chain van der Waals and hydrogen bonding interactions are not modeled. For our goals it was not necessary to model specific side chain interactions because distance constraints between backbone atoms in the zDK/LOV2 crystal structure could be used to favor a native-like complex during the docking simulations. The advantage of centroid mode is that conformational sampling is rapid and therefore 10,000 independent docking trajectories can be performed in a few hours on a single processor. Performing many independent trajectories allows the linkers to adopt many alternative conformations and provides a robust test of whether formation of the zDK/LOV2 complex is mutually exclusive with binding to the protein of interest.

In two cases we examined, cofilin and the CDB domain, a crystal structure was available of the complex that we wanted to regulate and therefore simulations could be performed with and without the binding partner present. This allowed us to

explicitly look for linkers that allow formation of the zDK/LOV2 complex in the absence of the binding partner, but do not allow zDK/LOV2 formation in the presence of the binding partner because of steric clashes. This explicitly demonstrates that Z-lock should inhibit binding in the dark state. However, in cases where a structure (or homology model) is not available of the complex that one wants to regulate, it is still possible to perform useful simulations if there is some information about the binding surface that needs to be caged. In the case of  $\alpha$ TAT mutational studies had mapped out which residues on the protein were important for binding to tubulin. In this scenario, the modeling trajectories were just probed to determine if the zDK/LOV2 complex was forming adjacent to the tubulin binding surface.

In conclusion, Monte Carlo sampling of alternative linker conformations provides a fast, versatile approach to determine the geometric relationships between rigid protein domains joined by flexible linkers and is an effective method for determining if a protein surface can be blocked with the Z-lock system. We anticipate that the protocol will be useful for engineering a broad variety of light responsive systems based on Z-lock.

### 3.4 Materials and Methods

To validate a given set of linkers for their suitability in caging a given interface, it is necessary to construct a model of those domains in both the presence and absence of the binding partner of interest, then perturb the linker backbone torsions to create an ensemble of models representing possible relative orientations

of the optical components to each other and the protein of interest; the entire ensemble can then be analyzed to estimate the relative ease of formation of the optogenetic interface and fraction of formed interfaces that also cage the desired interface. The first step, that of forming the model itself, can be done manually by aligning the components to each other using standard molecular modeling software, but was done automatically via domain assembly software created for the Rosetta macromolecular modeling suite that automatically generates linker atomic coordinates for a provided sequence, links them to the provided domains via idealized peptide bonds, and outputs the relevant kinematic inheritance tree for downstream linker perturbation. An example of this component file is provided as figure 3.8.

The RosettaScripts executable was then used to perturb the backbone torsional dihedrals of the linker residues in centroid mode according to the RosettaScript listed as supplementary figure S1. As the force field terms required to favor the formation of an interface only function at ranges that are small in comparison to the entire sampleable space, the score function was biased towards favorable interfaces by including constraints to favor distances between the alpha carbons of the LOV-zDK interface residues seen in the relevant LOV-zDK crystal structure. A sample constraint file of this type is included in figure 3.7, and an example fold tree is provided as figure 3.6. Flags files and command lines for running both scripts are figures S3.10-S3.13. Decoy generation itself consisted of a combination of fragment insertion into the linkers and random backbone torsion

perturbation under the control of a simulated annealing Monte Carlo function. This step is followed by gradient-based minimization under the same constraints to optimize the linker torsions. As these constraints also artificially force the optogenetic components closer than they would be randomly in the absence of an interface, it is necessary to differentiate formed interfaces from unformed interfaces through a low numerical cutoff on the score term representing the total deviation of the interface alpha carbons from their ideal positions. Typically, 2000 such models are independently generated to assay a single set of linkers, 1000 in the monomer state and 1000 in the complex state; if the complex state is unknown, 2000 trajectories of the monomer state are used instead. In many cases, the set of 2000 trajectories was then repeated 10 times to establish error bars. The fraction of models that successfully formed interfaces (as determined by the constraint term being below a numerical cutoff, usually 5) was then compared in the case of the sensor alone relative to the sensor with the binding partner of interest positioned over the interface. If the selected linkers are too short, a large fraction of models will fail to form interfaces in both the monomer and complex cases as the protein of interest separates the optogenetic components too widely; if they are too long, the interface will form away from the interface of interest, creating a “leaky” sensor in which the closed state is not mutually exclusive with binding. This state is detectable as a high proportion of models with formed interface in both the monomer and complex states. The linker residues of optimal length will display a high proportion of formed interfaces in the monomer state but not in the complex state.



For the cofilin sensor, the complex state used was that of the cofilin-actin complex in the the co-crystal structure 5YU8. For the CBD sensor, structures of CDC42 (2ODB) and WASP CBD (1CEE) were aligned according to the complex in the structure 1NF3. For the ATAT sensor, the structure of ATAT alone (4GS4) was used. The LOV2 and zdk structures were extracted from 5EFW.

#### Plasmid construction

LOV2 dark mutant (C450A, L514K, G528A, L531E, and N538E) and lit mutant (I510E and I539E) were used for Z-lock-WASP dark and lit, respectively. These cDNA fused with zDK1-WASP CBD (Cdc42 binding domain) was cloned into pTriEx-mVenus vector. pEF-BOS-GST and pGEX-Cdc42 Q61L were provided by Kozo Kaibuchi (Nagoya University, Japan). Human Cdc42 Q61L cDNA was excised from pGEX plasmid and cloned into pEF-BOS-GST to express GST tagged Cdc42 Q61L in mammalian cells.

#### Pulldown assay

LinXe cells were cultured in DMEM supplemented with 10% fetal bovine serum (Gemini Bio-Product, CA, USA) and GlutaMax (Thermofisher Scientific, MA, USA). For pull down assay, the cells were seeded in 6-well plate one day before the transfection. The plasmids were transfected by Fugene 6 transfection reagent (Promega, WI, USA) according to the manufacture's instruction. The following amount of the plasmid was used for transfection. pTriEx-mVenus-Z-lock-WASP lit, 400 to 1600 ng; pTriEx-mVenus-Z-lock-WASP dark, 50 to 200 ng; pTriEx-mVenus-CBD and mutant CBD, 150 ng; pEF-BOS-GST and

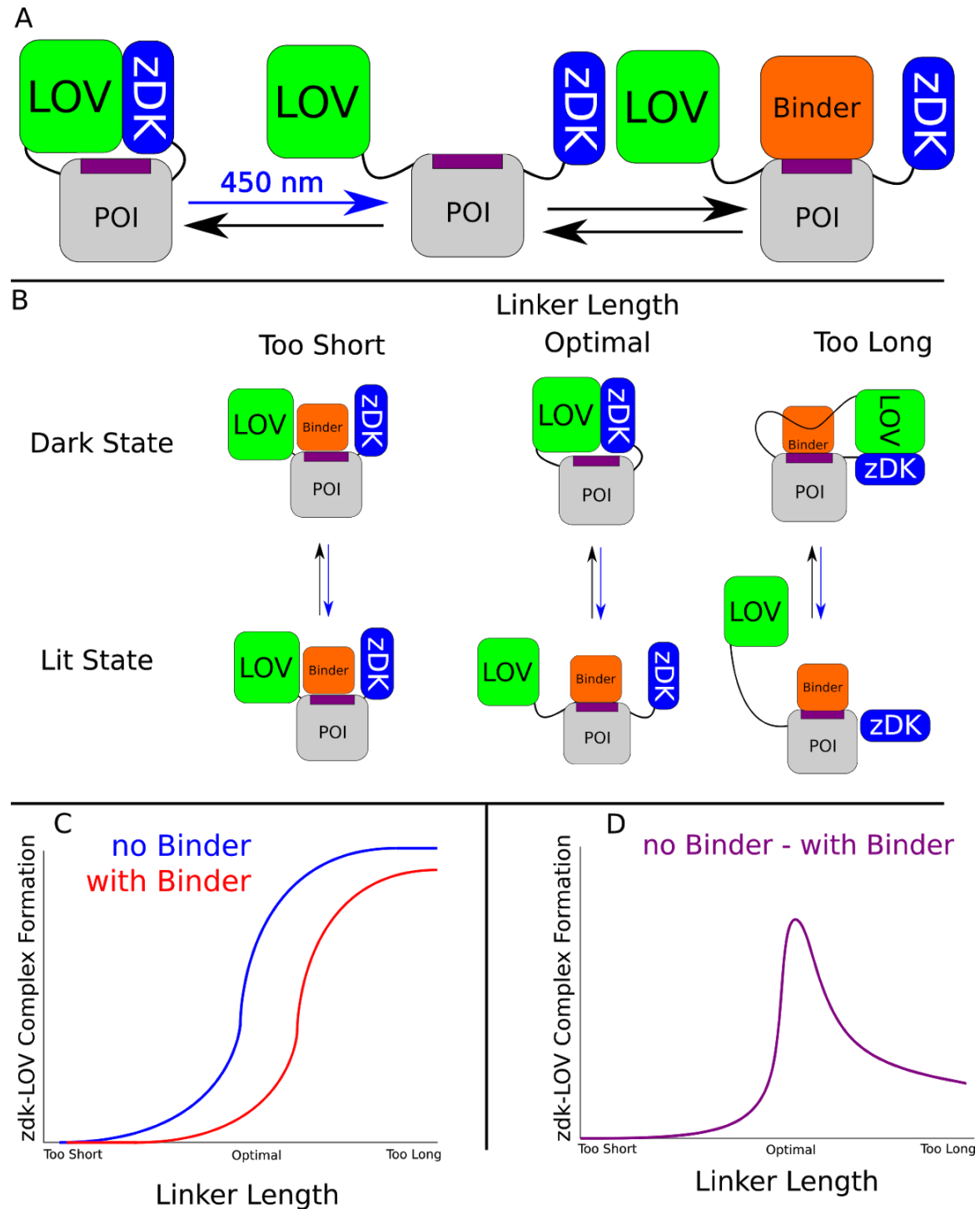
pEF-BOS-GST-Cdc42 Q61L, 500 ng. pBabePuro was used to equalize the total amount of the plasmid. 2.1 mg plasmids in total was mixed with 10.5 ml of Eugene 6 in Opti-MEM (ThermoFisher Scientific). After 24 hours transfection, the cells were lysed with the lysis buffer (25 mM Tris-HCl [pH 7.5], 1% NP-40, 10 mM MgCl<sub>2</sub>, 0.1% sodium dodecyl sulfate, 100 mM NaCl) supplemented with the protease inhibitor cocktail (Sigma, MA) by rocking the plate for 30 minutes at 4°C. The lysates were centrifuged by 12,000g for 5 minutes at 4°C to remove debris. The supernatant was incubated with glutathione agarose beads (ThermoFisher Scientific) at 4°C for 1 hour. After washing the beads three times with the lysis buffer, the bound proteins were dissolved in SDS-PAGE sample buffer. Immunoblot was used to detect the bound proteins. The following antibodies were used; anti-GFP antibody (JF-8; Takara, Japan), anti-GST antibody (91G1; Cell Signaling Technology, MA, USA), DyLight 800-conjugated anti-rabbit IgG antibody (Cell Signaling Technology), and DyLight 680-conjugated anti-mouse IgG antibody (Cell Signaling Technology). The fluorescence was visualized by ChemiDoc (BioRad, CA).

#### Acknowledgements

We thank Orrin Stone, Neha Pankow and Bei Liu for helpful discussions regarding the z-Lock system. This work was supported by the NIH grants R35GM131923(BK) and R35GM122596 (KH).

### 3.5 Figures and Tables

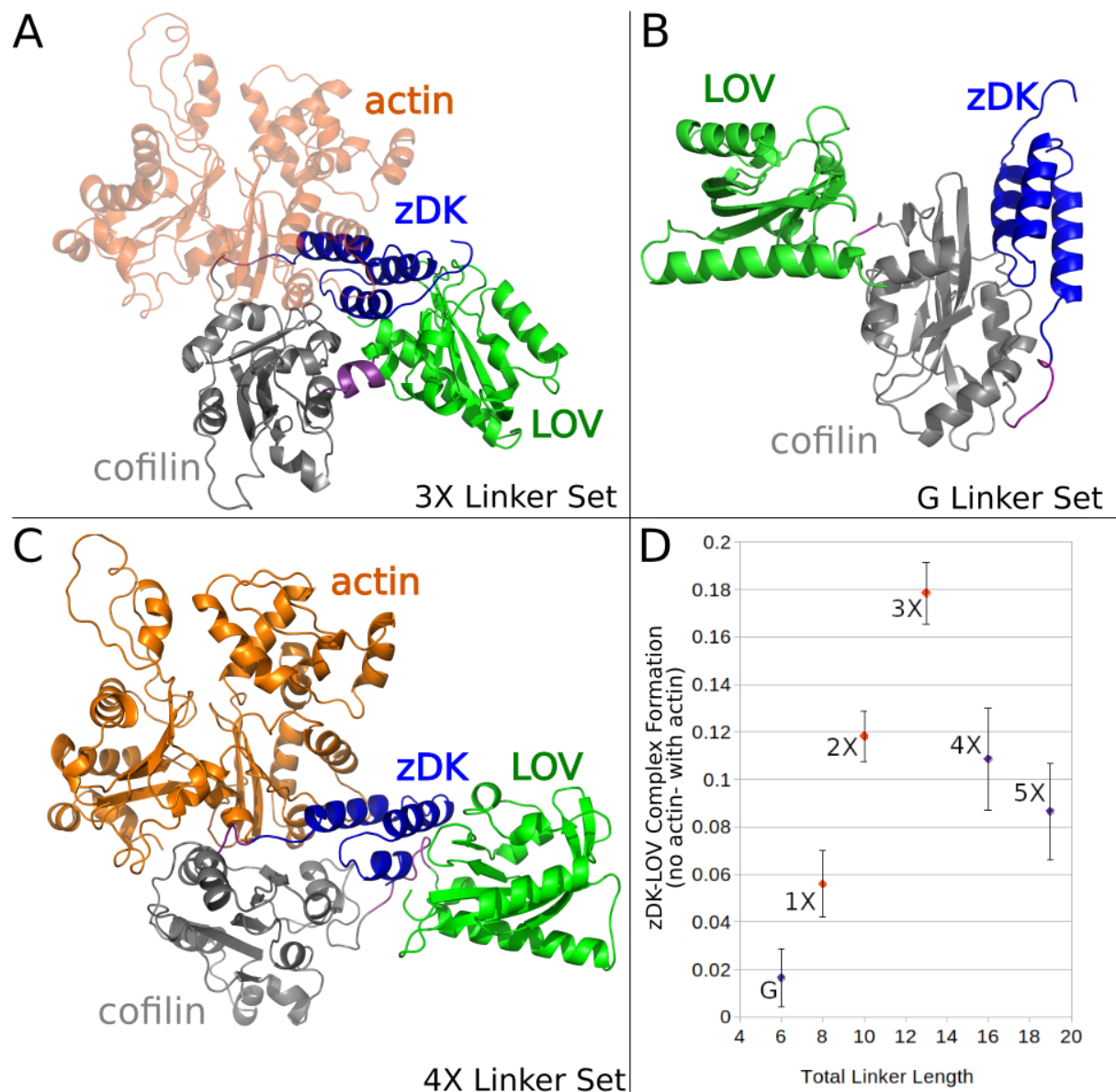
Figure 3.1 Inhibiting protein binding interactions with the light sensitive protein pair, LOV-zDK, and linker design.



(a) By physically coupling the light-sensitive binding between zDK and the LOV2 domain to a protein of interest via flexible linkers, the binding of specific interactors can be reversibly mediated with light. (b) The behavior of a given switch design is a function of the length of the linkers used. Linkers that are too short to permit the formation of the optogenetic interface will fail to cage the protein of interest in either state, as LOV and zDK will never interact; if the length of the linkers is excessive, the formation of the LOV-zDK interface will not prevent the binding of the binding partner to the protein of interest. In the optimal case, the formation of the LOV-zDK complex is mutually exclusive with the accessibility of the binding site. (c) The fraction of models from a large number of docking trajectories that successfully form the optogenetic zDK-LOV complex can then be compared in the presence and absence of the binder. If the linkers are too short, no complex will be seen in either the presence or absence of the binder. If they are too long, a high proportion of complexes will form in both cases. (d) The fraction forming in the presence of the binder can then be subtracted from that forming in the absence of the binder; the linker set with the biggest difference between the two states is predicted to be optimal.



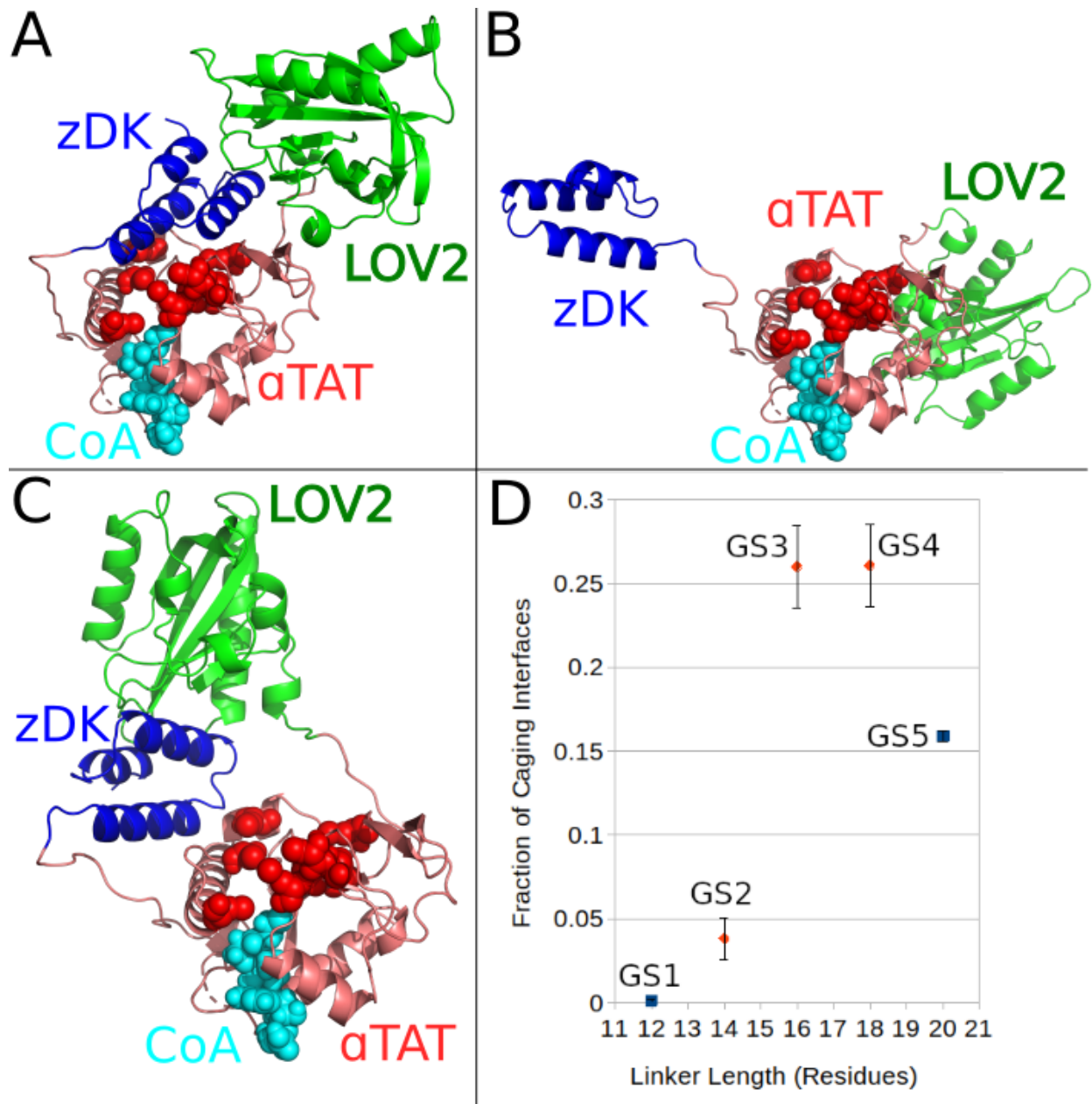
Figure 3.2. Modeling linkers for z-Lock cofilin.



Various linkers were modeled in the presence and absence of actin to establish which linkers should most effectively regulate binding to cofilin. (a) The optimal-length linker set “3X” positioned the zDK(blue)/LOV(green) interface over cofilin (grey) such that zDK sterically clashed with actin (orange.) (b) The shorter linker set “G” did not permit the formation of a zDK/LOV2 interface even in the monomer state (the pictured model is with the constraint term downweighted to 0 to

emulate the predicted behavior of this construct in cells). (c) The longer linker sets permitted the simultaneous formation of an actin-cofilin interface and a zDK/LOV2 interface. (d) The fraction of modeling trajectories that resulted in a well-formed interface between zDK and LOV2 was calculated for simulations with and without actin and the difference of these two fractions was plotted to establish the linker length most likely to inhibit cofilin binding in the dark. The linker length predicted to be most effective (3X) is in good agreement with experimental results that indicated that the 3X showed less binding to actin in the dark than the 2X and 1X linker sets. As expected, the dark-state efficacy of the linkers decreased sharply with decreasing length, a trend that continued into our short linker (blue, "G"), and the longer linkers (4X and 5X) exhibited a more gradual decline in predicted efficacy.

Figure 3.3 Modeling linkers for z-Lock  $\alpha$ TAT.



In lieu of a structure of the  $\alpha$ TAT/tubulin complex, interface caging was determined by counting contacts between residues of zDK/LOV and surface residues of  $\alpha$ TAT that are known to be critical for acetyltransferase activity (red spheres). (A) Models were considered to be caging if at least 30  $\alpha$ -carbons of either LOV2 or zDK were



within 20 Å of any critical residue, (B) while an interface was considered open if the zDK/LOV2 interaction could not form or (C) if the zDK/LOV2 complex was not adjacent to the tubulin binding site. While the fraction of caged models did not differ between the GS3 and GS4 linker sets, with dynamic ranges of 2 and 1.87 respectively, the GS2 linker set, with a dynamic range of 1.33, did show significantly fewer caging interfaces. GS3 was shown to be the most effective linker set experimentally (Stone et al., preprint).

**A** Ribbon diagram of the zDKLOV complex. The LOV domain is shown in green, the CBD domain in grey, and the zDK domain in blue.

**B** Ribbon diagram of the zDKLOV complex with Cdc42 (orange) bound to the zDK domain.

**C** Scatter plot showing zDKLOV Complex Formation (no Cdc42- with Cdc42) vs. total linker length for constructs GS0 to GS4. The y-axis ranges from 0 to 0.08, and the x-axis ranges from 0 to 9. GS0 is at 0 linker length, GS1 at 3, GS2 at 4, GS3 at 5, and GS4 at 8.

**D** Bar graph showing CBD binding to active Cdc42 for various constructs. The y-axis ranges from 0 to 1.0. The x-axis shows lit and dark conditions for zlock-Wasp CBD, Wasp CBD, and Wasp mCBD. Error bars represent standard deviation.

**E** Input assay showing zDKLOV-Cdc42 interaction. The top panel shows mVen-zlock-Wasp CBD and mVen-Wasp CBD. The bottom panel shows GST-Cdc42 Q61L and GST. The x-axis shows lit and dark conditions for zlock-Wasp CBD, Wasp CBD, and Wasp mCBD. The y-axis shows vs GST-Cdc42 Q61L and vs GST.

96

zDK2/LOV2 formation will be in competition and that this should be an effective switch. (C) Simulations with longer linkers (GS1-GS4) indicates that caging will be diminished as the linkers are lengthened. (D) Experimental analysis of the zDK2-GS0-CBD-LOV2 construct shows a light-dependent change in affinity for Cdc42 as determined by pulldown that varied directly with protein expression level (wedges, corresponding to 400, 800, and 1600 ng DNA for the lit state and 50, 100, and 200 ng DNA for the dark state), indicating that our prediction accurately determined the suitability of this linker set for optogenetically caging CBD.

Figure 3.5: A sample Rosettascripts script for sampling the space accessible to the optogenetic components in a constrained torsion perturbation run.

```

<ROSETTASCRIPTS>

  <SCOREFXNS>
    <ScoreFunction name="score3" weights="score3">
      <Reweight scoretype="atom_pair_constraint" weight="3.0"/>
    </ScoreFunction>
  </SCOREFXNS>

  <RESIDUE_SELECTORS>
</RESIDUE_SELECTORS>

  <SIMPLE_METRICS>
</SIMPLE_METRICS>

  <MOVERS>
    <SwitchResidueTypeSetMover name="switch_repr"
set="centroid" />
    <SwitchResidueTypeSetMover name="switch_repr_back"
set="fa_standard" />
    <AtomTree name="tree" fold_tree_file="ft"/>
    <ConstraintSetMover name = "add_constraints" cst_file =
"cst_file"/>
    <SingleFragmentMover name="frag_3mer"
fragments="fragment_file" policy="uniform">
      <MoveMap jump="0">
        <Span begin = "1" end="46" chi="0" bb="0"/>
        <Span begin = "63" end="210" chi="0" bb="0"/>
        <Span begin = "218" end="396" chi="0" bb="0"/>
        <Span begin = "397" end="397" chi="0" bb="0"/>
      </MoveMap>
    </SingleFragmentMover>

    <Small name = "smallmover_coarse" scorefxn = "score3"
nmoves = "1" angle_max="30">
      <MoveMap jump="0">
        <Span begin = "1" end="45" chi="0" bb="0"/>

```

```

    <Span begin = "56" end="203" chi="0" bb="0"/>
    <Span begin = "221" end="399" chi="0" bb="0"/>
    <Span begin = "400" end="400" chi="0" bb="0"/>
  </MoveMap>
</Small>

  <Small name = "smallmover_fine" scorefxn = "score3"
nmoves = "1" angle_max="15" temperature="2">
  <MoveMap jump="0">
    <Span begin = "1" end="45" chi="0" bb="0"/>
    <Span begin = "56" end="203" chi="0" bb="0"/>
    <Span begin = "221" end="399" chi="0" bb="0"/>
    <Span begin = "400" end="400" chi="0" bb="0"/>
  </MoveMap>
</Small>

  GenericMonteCarlo name="stage0" scorefxn_name="score3"
mover_name="frag_3mer" temperature="4.0" trials="400"
recover_low="1"/>
  <GenericMonteCarlo name="stage1" scorefxn_name="score3"
mover_name="smallmover_coarse" temperature="2.0" trials="100"
recover_low="1"/>
  <GenericMonteCarlo name="stage2" scorefxn_name="score3"
mover_name="smallmover_fine" temperature="1.0" trials="400"
recover_low="1" />

</MOVERS>

<PROTOCOLS>
  <Add mover_name = "switch_repr" />
  <Add mover_name = "tree" />
  <Add mover_name = "add_constraints" />
  Add mover_name = "stage0" />
  <Add mover_name = "stage1" />
  <Add mover_name = "stage2" />
</PROTOCOLS>

  <OUTPUT scorefxn="score3"/>
</ROSETTASCRIPTS>

```

Figure 3.6. A sample fold tree file for use with the above script

```
FOLD_TREE EDGE 200 1 -1 EDGE 200 400 -1 EDGE 200 401 1  
EDGE 401 600 -1
```

**Figure 3.7. A sample constraint file for use with the above script**

```
AtomPair CA 10 CA 265 HARMONIC 4.8 2
AtomPair CA 13 CA 263 HARMONIC 5.1 2
AtomPair CA 17 CA 260 HARMONIC 5.0 2
AtomPair CA 21 CA 257 HARMONIC 5.5 2
AtomPair CA 19 CA 259 HARMONIC 5.0 2
```

**Figure 3.8. A sample script file for use with the ConcatenatePosesMover, an application for assembling protein domains with flexible linkers.**

```
IGNORE This is a sample components file for the pose
concatenator. DOMAIN lines control the domains; specify a
filename and a chain number. LINKER lines work as before.
BINDER lines are DOMAIN lines with the binding name/chain
prepended by a domain/chain that exists already in a
DOMAIN line. IGNORE lines are ignored.
DOMAIN domain_1.pdb 1
LINKER GSG
DOMAIN domain_2.pdb 1
LINKER GSG
DOMAIN domain_3.pdb 1
BINDER domain_2.pdb binding_partner.pdb
```



Figure 3.9. A sample RosettaScript for using the ConcatenatePosesMover.

```
<ROSETTASCRIPTS>
  <MOVERS>
    <ConcatenatePosesMover name="build"
component_file="component_file"/>
  </MOVERS>
  <PROTOCOLS>
    <Add mover="build"/>
  </PROTOCOLS>
</ROSETTASCRIPTS>
```

Figure 3.10 A sample flags file for running the ConcatenatePosesMover  
RosettaScript

```
-parser::protocol path/to/concatenate_poses.xml  
-s path/to/dummy.pdb  
-nstruct 1
```

Figure 3.11. A sample flags file for running the structure prediction RosettaScript

```
-parser::protocol path/to/fragment_minimize.xml  
-s path/to/concatenated_pose.pdb  
-nstruct 1000
```

Figure 3.12 A sample command line for running the ConcatenatePosesMover  
RosettaScript

```
rosettascripts.linuxgccrelease @concatenate_poses.flags
```

Figure 3.13 A sample command line for running the structure prediction  
RosettaScript

```
rosettascripts.linuxgccrelease @fragment_minimize.flags
```

## CHAPTER 4: PROTOCOLS FOR REQUIREMENT-DRIVEN DESIGN IN THE ROSETTA MODELING PROGRAM

### 4.1 Introduction

Most studies in *de novo* protein design have begun with the researcher specifying a protein fold they would like to create and then using molecular modeling to identify amino acid sequences that will adopt that fold (Der et al., 2012; Joh et al., 2014; Kuhlman et al., 2003). While such *de novo* designs provide valuable information about the minimal determinants of protein folding and stability, the long term goal for the field is to create *de novo* proteins that also perform useful functions. When designing for protein function, it is no longer critical that the protein adopt a particular type of fold, but rather it must contain specific binding sites that allow it to perform its function, whether it be catalysis, transport, scaffolding, etc. We refer to this type of protein design as requirement-driven design, where the goal is not to create a specific protein topology, but rather create a well-folded protein (from many possible alternative topologies) that satisfies a set of user-specified requirements. These requirements can be quite general, for instance create a protein with a groove of a certain size, or they can be more detailed and require the precise placement of a set of amino acids to create a ligand binding site. Here, we describe a set of protocols we have written in the molecular modeling program Rosetta (Leaver-Fay, Tyka, Lewis, Lange, Thompson, Jacak, Kaufman, Douglas Renfrew, et al., 2011) for performing requirement-driven protein design.

A key component of our approach is a method, called SEWING, we developed for rapidly generating a large set of protein models that have varying sizes and three-dimensional structures, and are inherently designable because they have been assembled from pieces of naturally occurring proteins. In a previous study we showed that SEWING can be used to create novel helical bundles with structures that closely match the design models(Jacobs et al., 2016). SEWING begins by extracting supersecondary structure elements (e.g. helix-loop-helix motifs), called substructures, from native proteins. These substructures are then candidates for combination when the C-terminal secondary structural element of one substructure structurally aligns with the N-terminal secondary structural element of another substructure, and the substructures can be combined without clashes (Figure 4.1). Note that SEWING is currently only compatible with helical substructures. Beta strands may be present in the segments, starting structure, or partner protein(s), but SEWING cannot hybridize beta strands. Complete backbones are assembled using a Monte Carlo method of adding and deleting substructures using a user-specified score function. In a final step, rotamer-based sequence optimization protocols in Rosetta are used to identify low energy amino acid sequences for the SEWING generated backbone.

To use SEWING for requirement-driven design, filters and/or score terms are included during the assembly process to favor the generation of structures that satisfy the requirements. For instance, to build structures with a metal binding site a score term is added that favors structures with residues placed in the correct

position to coordinate with the metal. Additionally, the user can specify a starting substructure for SEWING that includes some of the structural features needed to satisfy a requirement. For example, the starting structure may include one or two critical contacts with a ligand, and then SEWING can be used to complete the ligand binding site.

In this paper we outline how to use SEWING to perform requirement-driven design (Figure 4.2). The primary interface to SEWING is RosettaScripts, an XML-based scripting interface for Rosetta(Fleishman et al., 2011). A typical RosettaScript for SEWING contains sections to specify the database of substructures that will be used, the score functions and filters that will be used to bias the assembly process, a starting structure if desired, and parameters that control the Monte Carlo assembly. After generating a set of backbone models, a separate RosettaScript is used to perform the final sequence optimization. To make this process more concrete, we also present two specific applications of SEWING: the design of new contacts at a protein-protein interface and the design of ligand binding sites.

#### 4.2 Walkthrough and General Guidelines

To run SEWING, users must have a compiled copy of Rosetta(Leaver-Fay, Tyka, Lewis, Lange, Thompson, Jacak, Kaufman, Renfrew, et al., 2011). Directions for obtaining a Rosetta license (free for academic users), downloading Rosetta source code, and compiling the code are available on the RosettaCommons web site (<https://www.rosettacommons.org/software/license-and-download>). Rosetta



applications, including SEWING, are run through a command-line interface using command-line options and associated input files. A single compute node can be sufficient to generate new backbones with SEWING, but the full design pipeline with rotamer-based sequence optimization generally requires access to a cluster with multiple processors.

SEWING is accessed through the RosettaScripts application which allows users to specify their protocol as an XML document and provides support for distributed computing.(Fleishman et al., 2011) Example RosettaScripts for several SEWING applications can be found in the Supplementary Materials. Briefly, each script will include a SEWING mover object (described below) in the MOVERS section. Options are defined within the mover tag, and the AssemblyScorers and AssemblyRequirements subelements allow score functions and requirements to be defined as separate subtags. Similarly, Ligand elements can be defined within the Ligands subelement and contain subtags listing their contacts and, when using LigandBindingAssemblyMover, their ideal coordination environment. Certain options also allow users to specify behaviors for particular residues using Rosetta ResidueSelector objects. For more general information on RosettaScripts, users should consult the Rosetta documentation.

#### 4.3 Available Protocols

SEWING is available through three Rosetta Mover objects,(Leaver-Fay, Tyka, Lewis, Lange, Thompson, Jacak, Kaufman, Douglas Renfrew, et al., 2011) each with

its own applications and available parameters. Here, we describe the basic AssemblyMover; AppendAssemblyMover and LigandBindingAssemblyMover will be described in the Applications section below.

The base protocol for SEWING is accessible through the AssemblyMover. This mover is used to generate backbones that fulfill a set of user-defined requirements without incorporating any particular starting structure. Given an input segment file (specified using the `model_file_name` option), this mover selects a random substructure as a starting point for each new backbone. It then performs a Monte Carlo simulation in which substructures are added, deleted, or “switched” (replaced) at the termini of the structure for a user-defined number of steps (defined by the `min_cycles` option). If the requirements have not been satisfied at this point, the simulation will continue until the structure satisfies the requirements or until the user-defined maximum number of cycles (`max_cycles`) has been reached. To see all of the structures along the path of a SEWING trajectory for illustrative purposes use the `output_pose_per_move` option.

AssemblyMover performs temperature cooling from the beginning of the simulation until `min_cycles` is reached (specified by the `start_temperature` and `end_temperature` options). At the end of the simulation, the lowest scoring assembly is recovered. The `window_width` option is used to indicate the minimum number of overlapping residues which should be present when combining two substructures.

#### 4.4 Recommended Options

A list of options available for SEWING movers and their defaults can be found in [Table S1](#). The default values for SEWING options are appropriate for typical applications, but modifications may benefit users in certain cases. By default, SEWING trajectories run for a minimum of 10,000 cycles and a maximum of 20,000 cycles. This value is appropriate for users generating ~10,000 to 100,000 structures; however, in some cases better performance can be achieved by generating more structures (>1,000,000) using fewer cycles. When using the recommended number of cycles and temperature schedule, SEWING performs best with relatively low add and delete probabilities (each 0.1 or less). The default values (add\_probability = 0.05, delete\_probability = 0.005) work well in most cases; however, greater add probabilities may be necessary for large assemblies or small numbers of cycles, and larger delete probabilities may help increase sampling of interior segments. By default, SEWING uses a window width of four residues (approximately one helical turn), which is sufficient for canonical helices. Note that SEWING is not currently recommended for use with beta strands as chimerizable segments.

#### 4.5 Command Line Options

Command-line flags for the RosettaScripts application can be provided as a flags file (Figure 4.4). The -in:file:s or -in:file:l flag indicates the starting structure to be used in AppendAssemblyMover or LigandBindingAssemblyMover; AssemblyMover does not use the input structure, but one must still be specified. Users must also specify the path to their RosettaScripts XML file using the -parser:protocol flag. Motif scoring

(-mh) flags should be specified as described in Figure S4.1. A list of options for other SEWING-related applications, such as the `segment_file_generator` application, can be found in Table 4.2.

#### 4.6 Refinement, Design, and Filtering

Since SEWING produces backbones but does not perform side chain design, users will also need to perform additional analysis and design; however, we recommend that this analysis be performed in a separate RosettaScript so that SEWING score terms will be output appropriately in the Rosetta score file. Since backbone generation is rapid compared to structure refinement and side chain design, we recommend that users only refine a fraction of generated backbones. For a typical production run we recommend that users generate at least 10,000 assemblies and select the top 10% by total SEWING score for further refinement; for a protein between 100–150 residues it will take on the order of one hundred CPU-hours to generate 10,000 Assemblies, and approximately 500 to 1000 CPU-hours to perform refinement on 1000 Assemblies (Table 4.3).

An example of a typical refinement and filtering script for SEWING designs is shown in Figure 4.6. Although SEWING ignores side chains, it does preserve their coordinates; therefore, since they are packed differently in SEWING assemblies than in their original context, these structures will initially have numerous side chain clashes. First, an initial fixed-backbone design step is performed to remove clashes within the structure. Cartesian minimization is then used to repair any bond length

and angle changes introduced during SEWING. Finally, Rosetta FastRelax performs additional rounds of iterative side chain design and backbone minimization.(Nivón, Moretti, & Baker, 2013; Tyka et al., 2011) Loop residues are constrained to their native identities throughout the design process to encourage proper folding of the structure, and any functional residues (i.e. “required residues” from the initial SEWING run) are held fixed. We recommend including a filter for fragment quality (FragmentLookupFilter), which verifies that each four-residue window adopts a structure compatible with its sequence(Dang et al., 2017):

## 4.7 Input Files for SEWING

### 4.7.1 Segment Files

Before running SEWING, users must generate a “segment file”, which contains coordinates, sequences, and connectivity information for a set of substructures from native proteins. A pre-generated helical segment file can be found in the Rosetta database in `main/database/additional_protocol_data/sewing`. This segment database was generated using a set of high-resolution structures selected using the PISCES protein sequence culling server with at least 2.2 Ångstrom resolution and no more than 30% sequence identity(G. Wang & Dunbrack, 2003).

To generate custom segment files, use the `segment_file_generator` Rosetta application along with a set of input structures. Since beta sheet specific

requirements have not yet been developed, we recommend that users do not include beta strands in segment files at this time.

#### 4.7.2 Motif score databases

SEWING also requires files that provide information necessary for scoring using the Motif Score, a sequence-independent metric of designability of neighboring secondary structural elements.(Fallas et al., 2017) These files are specified using command-line options (Figure 4.5) and can be found in the Rosetta database in `main/database/additional_protocol_data/sewing`.

#### 4.7.3 Edge Files

By default, SEWING computes alignments between helices on-the-fly, which improves performance for most helical assemblies. SEWING may also be run with alignments precomputed to reduce redundancy, providing more thorough checks of secondary structure alignment at the expense of speed and memory usage. If this option is enabled, then SEWING will also require an “edge file” (described in the Supplementary Materials) defining which helices can be chimerized. While this feature is generally unnecessary for generating helical proteins, it will likely be necessary for future applications involving more irregular secondary structure elements.

#### 4.7.4 Scoring

SEWING does not model amino acid side chains during assembly; all scorers and binary requirements are evaluated based on backbone coordinates, and side chain coordinates are restored at the end of each simulation. Therefore, it must rely

on a low-resolution score function that evaluates potential packing based solely on these backbone coordinates. For this we primarily use MotifScore, which based on the relative position of two residues, approximates the best possible attractive energy that could be achieved between the residues by mutating the residues to alanine, valine, isoleucine, or leucine.(Fallas et al., 2017) Specialized score terms are also included for interface design and/or ligand binding protein design. A list of all available scorers can be found in Table 4.4. For each protocol, users may specify any combination of scorers, and design trajectories will be based on the weighted sum of those scores. If no scorers are specified, SEWING will use the MotifScorer and InterModelMotifScorer with the recommended weights. This default score function is appropriate for simple backbone design when no binding partners or ligands are present. Note that specifying any scorers will override the default score function, so the MotifScorer and InterModelMotifScorer will not be included unless they are also specified.

## 4.8 Requirements

One of the key features of SEWING is the user's ability to place boolean restrictions on designed structures without specifying a particular fold, to be checked after every change to the Assembly and trigger a reversion to the previous state if failed. We refer to these restrictions as requirements. Requirements can range from simple (e.g. the ClashRequirement, which prevents clashes in the designed protein) to more complex requirements targeting specific regions of the structure. A full list of



available requirements is found in Table 4.5. If no requirements are specified, then the ClashRequirement will be applied by default; we recommend this requirement for all design cases. Note that specifying any additional requirements will override the default, and the ClashRequirement will not be included unless it is also specified by the user. Note also that these requirements are not to be confused with the “requirements” in the more general sense referenced previously, which refer to all of the information provided to the SEWING protocol and used to guide backbone generation. Requirements are intended to work in concert with Scorers and the various options available to the SEWING movers to define the desired features of the finished Assembly.

## 4.9 Applications

### 4.9.1 Expanding Protein Interfaces

AppendAssemblyMover, also known as SEWING Append, uses the same method as AssemblyMover to generate a structure from native elements; however, instead of beginning with a random substructure, it incorporates a user-specified starting structure into all designed backbones as shown in Figure 4.3. Therefore, users can incorporate functional motifs, such as protein-binding peptides or ligand binding sites, into their designed backbones provided that the motifs are local in sequence. Since the only requirement for this starting structure is that SEWING can append to at least one of its termini using the provided segment file, this protocol can produce final structures containing elements that SEWING would normally not

be able to generate and optimize those elements' packing using the SEWING score function. For instance, although users may not yet use SEWING to chimerize across beta sheets, they can use AppendAssemblyMover to add structure to an existing beta sheet, provided that the starting structure has at least one terminal helix available for chimerization.

The `required_resnums` or `required_selector` options in AppendAssemblyMover can be used to guarantee that a set of residues will be preserved during assembly. To enforce this guarantee, SEWING must not include these residues in the overlapping regions of segments being combined; therefore, if a required residue is within `window_width` residues of the terminus of the user's starting structure, then SEWING will be unable to append to that terminus. Users in this scenario may need to append additional helical structure to the appropriate terminus before running SEWING Append.

The starting structure will by default be converted into a series of segments according to its secondary structure determined by the Dynamic Secondary Structure Prediction (DSSP) algorithm (Kabsch & Sander, 1983). As in segment file generation, SEWING will ignore single-residue loops by default. This can be disabled by setting the `strict_dssp_changes` option (in this case, an XML option) to false. Users may also override the automatically detected segments by providing a series of segment start points, end points, and DSSP codes (Figure S4.3). The segment start and end points are provided as comma-separated lists to the

pose\_segment\_starts and pose\_segment\_ends options, respectively, and the segments' secondary structure types can optionally be provided to the pose\_segment\_dssp option as a string with each character corresponding to one segment.

The modifiable\_terminus option can be used to specify which termini can be extended during SEWING. Note that the secondary structure of this terminus should match that of the segment file being used. Extend mode is available for cases in which users simply want to extend a secondary structure element.

#### 4.9.2 Adding other proteins to the simulation

One potential application for SEWING Append is to extend protein interfaces to improve binding affinity or specificity (Figure 4.7). In this scenario the aligned binding partner(s) is included as a separate input file using the partner\_pdb option. Binding partners are considered by certain scorers and requirements but are not modified or moved relative to the starting structure. This can be used to ensure that no part of the Assembly will clash with the partner, but it can also be used in concert with the PartnerMotifScorer to bias the assembly towards placing segments in proximity to the partner, thereby allowing residue-level design to create new interface contacts. In cases where contacts to a particular region of the partner protein are desirable, the SubsetPartnerMotifScorer biases the assembly toward placing structure in proximity to a specified region of the protein. Furthermore, the termini of the designed protein may be positioned through the use of the

TerminusMotifScorer, which biases the placement of terminal residues within a user-defined volume as measured by proximity to specific residues on the partner protein. This can be used to produce designs that are more likely to accommodate FRET pairs or other useful motifs without clashing with either the design or the partner.

#### 4.10 Guidelines for use

AppendAssemblyMover will by its nature restrict the sample space available to the algorithm, since every run starts with the same substructure and potentially the same excluded volume. The addition of a partner PDB, and therefore of further constraints on the volume available for the design to fill, further restricts the SEWING algorithm. Users should therefore run many short SEWING runs in preference to fewer runs with more cycles, as longer runs may produce duplicate structures. Users should adjust the TerminusMotifScorer weights such that the range of expected score values results in a TerminusMotifScorer value similar to that of the MotifScorer terms, which can be expected to return values from -2 to 0 multiplied by the weight; while other scorers are constrained by the nature of the MotifScore to comparable ranges of possible values, TerminusMotifScorers are geometric in nature and can therefore return values large enough to dominate the score and return otherwise undesirable designs that satisfy the TerminusMotifScore requirements at the expense of all other terms.

#### 4.11 Sample Results

To demonstrate the SEWING-Append protocol, we generated helical backbones starting from the VBS1 helix of talin in complex with vinculin (PDB code 1T01). The script shown in Figure S4.3 was used to extend the existing helix to create up to 5-helical bundles, producing a total of 4627 designs; the top 463 designs by total weighted SEWING score were then refined as described in the Refinement, Design, and Filtering section (Figure 4.6) to produce designs with an average per-residue score of  $-3.03 \pm 0.4$  kcal/mol with the Ref2015 score function. The entire design process used 252 CPU-hours.

#### 4.12 Ligand Binding

##### Compatibility with existing ligands

Partial and complete ligand binding sites offer another set of possible functions to be incorporated using SEWING Append. Incorporating non-protein residues raises additional challenges, however, which have required special adaptations to the SEWING framework.

Since ligands often bind with multiple protein residues that can be distant in sequence space, ligands are stored as specialized residues which are not part of any one segment. These ligand residues store their contacts, which can either be specified manually by the user or, in the case of coordinated metal ions or covalent bonds to ligand atoms, can be automatically detected. Contacts are stored as required residues which must be preserved throughout the protocol, and the ligand

is anchored to its most N-terminal contact residue. Any movements of that residue will cause the ligand to move as well. Ligand-protein contacts are scored using a new LigandMotifScorer (Table 4.4) which uses the distance and angle of C $\alpha$ -C $\beta$  bond vectors with hydrophobic atoms in the ligand to favor new hydrophobic ligand contacts; hydrophilic ligand atoms are not scored. A new LigandClashRequirement (Table 4.5) identifies protein-ligand clashes by detecting ligand atoms and protein C $\alpha$  atoms within a user-specified clash radius.

#### Incorporating new ligand contacts

SEWING can also add contacts to partially-coordinated ligands using the new LigandBindingAssemblyMover, or LigandSEWING (Figure 4.8B). This mover is a specialization of SEWING Append which, in addition to adding new structural elements to a ligand binding site, also adds new ligand-protein contacts that satisfy user-specified geometric requirements. LigandBindingAssemblyMover is solely intended for adding new ligand contacts and is typically used as a first step in a larger protocol (Figure 4.2); users who begin with a fully coordinated ligand, as well as those who have already completed their ligand's coordination using this mover, should use AppendAssemblyMover as described above.

To add new ligand contacts using SEWING, the user must specify the ideal geometry of the contacts. This geometry is treated as two components. First, the ligand stores ideal bond lengths and angles for each atom that will form new contacts; these can be specified as IdealCoordination subtags of the Ligand tag in

RosettaScripts (Figure 4.9). The user must also specify the possible positions of new coordinating residues relative to these atoms. This information is pre-generated by the user in a separate Rosetta application (the `zinc_statistic_generator` application is available for polar tetrahedral atoms) using the following procedure: An isolated residue is mutated in turn to all possible coordinating residues for the ligand atom. This residue then cycles through its most common rotamers. For each rotamer, the application then builds each possible position of the coordinated ligand atom and converts that atom's coordinates into the coordinate frame defined by the residue's three main-chain atoms (Figure 4.8A). These coordinates, along with the residue type, chi angles, atom names, and secondary structure type, are stored in a ligand coordination file (Figure 4.10). Each `IdealCoordination` tag is provided with the name(s) of these files for the ligand atom being considered.

During assembly, `LigandSEWING` adheres to a user-specified "segment distance cutoff" which determines how much additional structure can be built on either side of the original site. This cutoff helps prevent the mover from continuing to add structural elements that are too far from the ligand to potentially form new contacts. Each time new substructures are added to the assembly, they are first checked to see whether any added segment comes within a user-specified distance of the ligand atom being coordinated. If a segment meets this cutoff, then each added residue's position relative to the ligand is compared to the possible positions defined in the provided ligand coordination file. If a match is found, the residue is then mutated to the residue type and rotamer specified in the hash, and the overall

geometry of the ligand binding site is scored. Once this process is complete for all new residues, the best-scoring mutation is kept if its geometry score meets a user-specified score threshold. When the ligand atom's coordination is complete, LigandSEWING either outputs the completed site immediately or, optionally, transitions into a SEWING Append protocol using the completed ligand binding site as the starting structure.

#### Guidelines for use

Since the sampling space of LigandBindingAssemblyMover is highly restricted, users should use relatively high temperature values (between 2.0 and 3.0) to ensure diverse sampling. Due to this restricted (usually exhaustive) sampling and because it accepts the first ligand contact discovered that passes its scoring thresholds, LigandSEWING does not perform temperature ramping or cooling; instead, the `max_temperature` option sets the value for the entire protocol. Instead, temperature cooling is performed using AppendAssemblyMover in the latter steps of the protocol, either by setting the `build_site_only` option to true or by performing a separate Append step after site completion.

The `ligand_binding_cycles` option sets the maximum number of moves that the mover will perform (not counting the optional Append step, which uses the `min_cycles` and `max_cycles` options); we recommend a value of 20000 cycles for protocols adding a single ligand contact.



To prevent wasted sampling, a segment distance cutoff should not exceed two. Note that this cutoff counts the number of substructures added to each terminus of the starting structure rather than the number of segments added. A max\_distance of 8.0 Å is also recommended to prevent unnecessary calculations in segments too distant to form ligand contacts.

Recommended geometry score thresholds will vary depending on the coordination environment of the specified atom. Lower scores generally indicate more ideal geometry (and therefore a stricter cutoff). Since this score includes angles and dihedrals between all pairs of contacts to the atom in question, it scales nonlinearly with the number of contacts; for example, a value of 5.0 is appropriate for a tetrahedrally coordinated atom forming a third contact whereas a value of 20.0 will give similarly ideal geometries when forming a fourth contact. Non-ideal coordination geometry in the starting contacts also lead to increased scores. Therefore, we recommend that users try several thresholds for their specific use case before running full-scale simulations and select one which gives the highest possible success rate while yielding satisfactory ligand geometry.

## Sample Results

To demonstrate the Ligand SEWING protocol (Figure 4.8B), we generated a set of helical proteins containing a tetravalent zinc ion beginning with a partially coordinated zinc. The script shown in Figure S4.9 was first used to add a fourth coordinating residue to zinc ions with three coordinating histidines using

LigandBindingAssemblyMover. Five starting sites were used as input and were run for 2000 trajectories each; however, only 99 unique new contacts were identified due to the highly limited sampling space of LigandBindingAssemblyMover. These 99 sites were then used as input to AppendAssemblyMover to generate 1000 assemblies per starting site, and the top 10% of generated assemblies proceeded to refinement. After filtering designs as described in the Refinement, Design, and Filtering section (Figure 4.6) along with additional project-specific filters, 139 refined assemblies were available for further filtration and selection. Scores of these designs ranged from  $-4.2$  to  $-3.4$  kcal/mol per residue (median =  $-3.9$  kcal/mol/res, mean =  $-3.9$  kcal/mol/res) with the REF2015 score function.

#### 4.13 Conclusions

SEWING provides an alternative to traditional methods for *de novo* protein design which rely on strict predetermined structural restrictions. Due to its ability to quickly generate diverse backbones, it is a promising approach for applications that benefit from having a large library of possible starting structures (e.g. enzyme design). Importantly, SEWING is also able to combine multiple functional elements (e.g. a protein binding partner and a ligand) into a single protein. By rapidly sampling a large structural space, SEWING is able to find ways to accommodate arbitrary sets of user requirements, thus encouraging functional protein design.

## 4.14 Supplementary Information

### 4.14.1 Segment File Generation

Before generating a segment file, users must first decide on the number, type, and lengths of secondary structure elements that will compose their substructures. These are delineated in a “motif file”, a text file in which each line describes the composition of substructures for a segment file. Each secondary structure element in the substructure is specified by a DSSP code, a minimum length in residues, and a maximum length in residues separated by spaces; secondary structure elements are separated by commas. Users should provide the path to this file using the `-motif_file` command line option; if multiple motifs (lines) were specified in the file, then one segment file will be output for each motif.

The native structures used to extract substructures should be provided as a text file containing a list of PDB files, specified by the `pdb_list_file` option. To ensure that the generated segment file is both accurate and nonredundant, input crystal structures should be restricted to high-resolution crystal structures with limited homology. The TOP8000 set of high-resolution crystal structures provides an appropriate starting set;<sup>11</sup> alternatively, users may choose their own structures using tools such as the PISCES protein sequence culling server.<sup>12</sup>

By default, SEWING only recognizes loops of two or more residues to allow for kinks in helices that DSSP may identify as loops. This behavior is controlled by

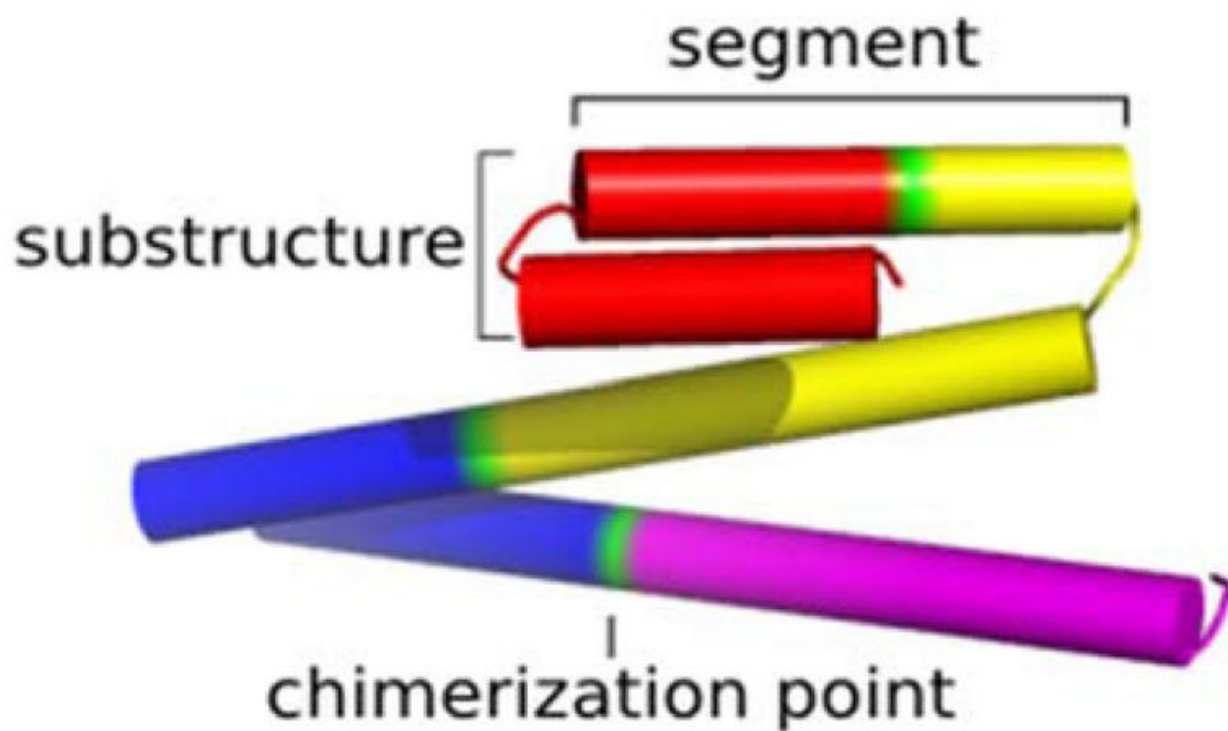
the `strict_dssp_changes` command line option and can be disabled by setting the option to `false`. Users should also process all structures using the `clean_pdb.py` script found in the Rosetta tools repository before generating segment files to remove non-supported residue types.

#### 4.14.2 Edge File Generation

Edge files for hashed SEWING can be generated using the `edge_file_generator` application. To run the application, the user must provide the path to the input segment file using the `-sewing:segment_file_name` option and a path for the output edge file using the `-sewing:edge_file_name` option. The default values for the `-sewing:max_clash_score` and `-sewing:box_length` options (0 and 3, respectively) should be appropriate for most use cases; however, we recommend increasing the `-sewing:min_hash_score` option to 20 so that at least 20 overlapping atoms are required to allow two helices to hybridize.

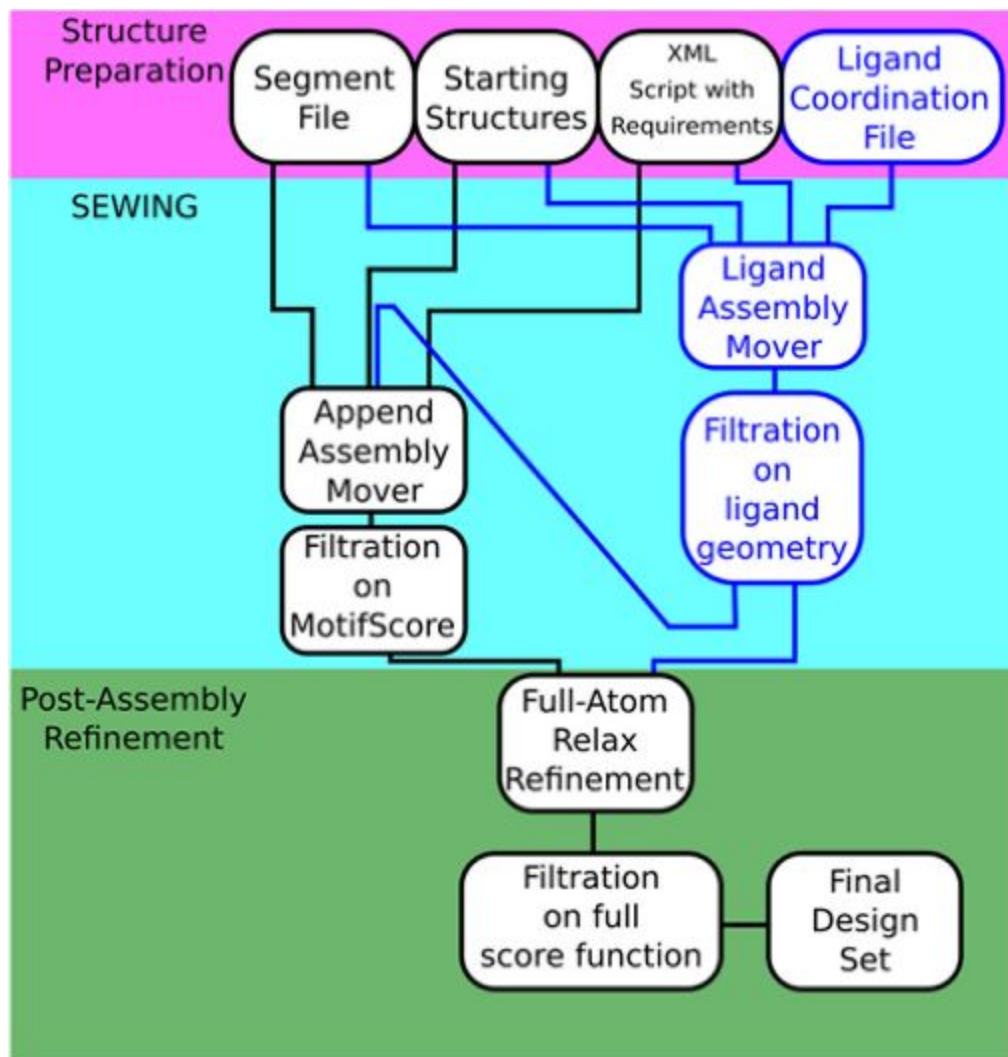
#### 4.15 Figures and Tables

Figure 4.1. A sample SEWING output, or Assembly.



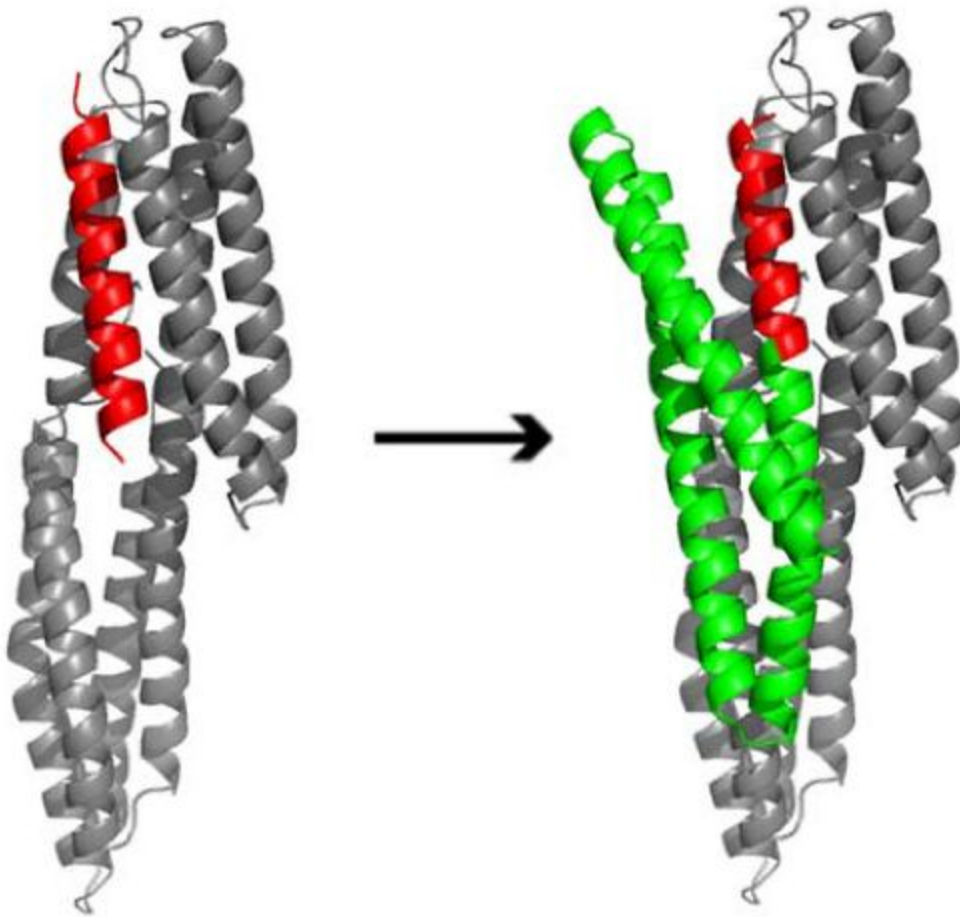
Individual segments, contiguous sets of residues with a common secondary structure, are sourced from crystal structures as contiguous substructures and recombined into an Assembly by transforming a residue of one segment onto a residue of another (collectively a basis pair) by pair-fitting the backbone atoms and saving the N-terminal region of one segment and the C-terminal region of another.

Figure 4.2. A flowchart showing the general SEWING workflow.



Portions unique to ligand binding site design are shown in blue; after generating assemblies containing the site, design may proceed through AppendAssemblyMover or directly to full-atom design.

Figure 4.3. Interface expansion with SEWING.



Starting from a fragment of talin(red) bound to vinculin(grey), the starting fragment is expanded (green) to make additional contacts with vinculin.

<b>Table 4.1. Score functions available during SEWING</b>				
<b>Scorer</b>	Score Range <sup>1</sup>	Suggested Weight	Requires <sup>2</sup>	Measures <sup>4</sup>
<b>MotifScorer</b>	-2,0	1	Assembly	all intra-assembly interactions
<b>InterModelMotifScorer</b>	-2,0	10	Segments>2	interactions between distant segments
<b>PartnerMotifScorer</b>	-2,0	10	Partner	potential interface to partner protein
<b>SubsetPartnerMotifScorer</b>	-2,0	10	Partner	potential interface to specific residues
<b>LigandScorer</b>	-2,0	1	ligands	ideality of ligand geometry
<b>NtoCTerminusMotifScorer</b>	Varies	Varies	Assembly	distance between termini
<b>TerminusMotifScorer</b>	Varies	Varies	partner	distance from terminus to single partner residue

<sup>1</sup>Range of scores that can result from the requirement. “Varies” indicates that values are unbounded and should be benchmarked case-by-case. <sup>2</sup>Requirements (beyond those demanded by the SEWING Mover) needed for the scorer to return meaningful results.



<b>Table 4.2. Requirements available during SEWING</b>		
<b>Requirement</b>	<b>Requires</b>	<b>Measures</b>
<b>SizeInSegmentsRequirement</b>	Min,max	Size by number of secondary structural elements
<b>LengthInResiduesRequirement</b>	Min,max	Size in residues
<b>DsspSpecificLengthRequirement</b>	Int, DSSP code	Loop length or helix length
<b>ClashRequirement</b>	Max, radius	Clashes. On by default.
<b>LigandClashRequirement</b>	Max, radius	Clashes with ligand
<b>NonTerminalStartingSegmentRequirement</b>	none	Location of starting segment in assembly

## **Supplementary Figures**

**Figure 4.4:** Sample flags file used to run SEWING.

```
-ignore_unrecognized_res

-parser:protocol zn_sewing_script.xml

-detect_disulf false

-mh

-score

    -use_ss1 true

    -use_ss2 true

    -use_aa1 false

    -use_aa2 false

    -path

    -motifs
    /home/guffy/Rosetta/main/database/additional_protocol_data/
    sewing/xsmax_bb_ss_AILV_res10.8_msc0.3.rpm.bin.gz

    -scores_BB_BB

    /home/guffy/Rosetta/main/database/additional_protocol_data/
    sewing /xsmax_bb_ss_AILV_res10.8_msc0.3

-gen_reverse_motifs_on_load false
```

**Figure 4.5:** Sample RosettaScript used to refine and filter backbones that were designed using the SEWING protocol. We recommend that users refine only the best 10% of backbones by total SEWING score.

```
<ROSETTASCRIPTS>

  <SCOREFXNS>

    <ScoreFunction name="ref2015_cart" weights="ref2015_cart">
    </ScoreFunction>

    <ScoreFunction name="ref2015" weights="ref2015">
    </ScoreFunction>

    <ScoreFunction name="ref2015_soft" weights="ref2015_soft">
      <Reweight scoretype="res_type_constraint" weight="1.0" />
    </ScoreFunction>
  </SCOREFXNS>

  <RESIDUE_SELECTORS>

    <SecondaryStructure      name="loops"      overlap="1"
    include_terminal_loops="false" ss="L" use_dssp="true" />

    <Layer      name="core"      select_core="true"
    use_sidechain_neighbors="true"/>

    <Layer      name="boundary"      select_boundary="true"
    use_sidechain_neighbors="true"/>

    <Layer      name="surface"      select_surface="true"
    use_sidechain_neighbors="true"/>

    <Not name="not_loops" selector="loops"/>
```

```

    <Chain name="chB" chains="B"/>

    <ResiduePDBInfoHasLabel name = "vital_residues" property
    = "VITAL" />

</RESIDUE_SELECTORS>

<TASKOPERATIONS>

    <InitializeFromCommandline name="init"/>

    <RestrictAbsentCanonicalAAS                                name="nocys"
    keep_aas="ADEFBGHIKLMNPQRSTVWY" />

    <IncludeCurrent name="current" />

    Begin layer design operations

    <OperateOnResidueSubset                                name="surface_to"
    selector="surface" >

        <RestrictAbsentCanonicalAASRLT aas="DEGHKNPQRST"/>

    </OperateOnResidueSubset>

    <OperateOnResidueSubset                                name="boundary_to"
    selector="boundary" >

        <RestrictAbsentCanonicalAASRLT
        aas="ADEFBGIKLNPQRSTVWY"/>

    </OperateOnResidueSubset>

    <OperateOnResidueSubset name="core_to" selector="core" >

        <RestrictAbsentCanonicalAASRLT aas="AFILPVWY"/>

    </OperateOnResidueSubset>

    <OperateOnResidueSubset                                name="helix_to"
    selector="not_loops" >

        <RestrictAbsentCanonicalAASRLT
        aas="ADEHIKLNQRSTVWY"/>

    </OperateOnResidueSubset>

```

```

<OperateOnResidueSubset name="only_B" selector="chB" >
    <PreventRepackingRLT/>
</OperateOnResidueSubset>

<OperateOnResidueSubset name = "vitals" selector =
"vital_residues" >

<RestrictToRepackingRLT/>

</OperateOnResidueSubset>

</TASKOPERATIONS>

<FILTERS>

    <SecondaryStructureHasResidue name="require_core"
confidence="0" /> Defaults will ensure hydrophobic
residues in each helix

    <PackStat name="pstat" threshold="0.60" repeats="1"
confidence="0" />

    <SSPrediction name="sspred" threshold="0.35" use_svm="1"
use_probability="1" mismatch_probability="1"
confidence="0" />

    <CavityVolume name="cavvol" />

    <BuriedUnsatHbonds name="bunsat" scorefxn="ref2015"
cutoff="5" confidence="1" />

</FILTERS>

<MOVERS>

<AddConstraints name="constrain_loop_identities" >

<ResidueTypeConstraintGenerator name="loop_csts"
residue_selector="loops" favor_native_bonus="1.0" />

</AddConstraints>

<PackRotamersMover name="prepack" scorefxn="ref2015_soft"
task_operations="init,nocys,current,core_to,surface_to,bo
undary_to,helix_to,only_B" />

```

```

<MinMover    name="cartesian_min"    scorefxn="ref2015_cart"
chi="false"    bb="false"    omega="false"    jump="ALL"
cartesian="true"    bondangle="true"    bondlength="true"
tolerance="0.001" max_iter="1000" />

<FastDesign          name="relax"          scorefxn="ref2015"
disable_design="false"          repeats="2"
task_operations="init,nocys,current,core_to,surface_to,bo
undary_to,helix_to,only_B,vitals"
min_type="lbfgs_armijo_nonmonotone"
ramp_down_constraints="false" />

</MOVERS>

<APPLY_TO_POSE>

</APPLY_TO_POSE>

<PROTOCOLS>

  <Add mover_name="constrain_loop_identities" />

  <Add mover_name="prepack" /> This is added so that side
chain clashes will be resolved before we try to fix
chainbreaks

  Add mover_name="cartesian_min" /> This is ONLY being used
to fix bad bond lengths/angles at chimerization points
(no torsion angle minimization before design)

  <Add mover_name="relax" /> FastDesign gives better tracer
output for design

  <Add filter_name="require_core" />

  <Add filter_name="pstat" />

  <Add filter_name="cavvol" />

  <Add filter_name="bunsat" />

  <Add filter_name="sspred" />

</PROTOCOLS>

<OUTPUT scorefxn="ref2015" />

```

</ROSETTASCRIPTS>

**Figure 4.6:** Sample RosettaScript for expanding the interface between a protein-binding peptide and its binding partner using SEWING Append.

```

<ROSETTASCRIPTS>

<SCOREFXNS>

</SCOREFXNS>

<FILTERS>

</FILTERS>

<MOVERS>

    <AppendAssemblyMover
        model_file_name="test.model"  partner_pdb   =  "test.pdb"
        hashed="false"      required_resnums      =      "7,8,11,12"
        max_segments      =      "11"      minimum_cycles      =      "1000"
        maximum_cycles      ="1100"      modifiable_terminus      =      "B"
        output_partner = "true"  >

        <AssemblyScorers>

            <MotifScorer weight = "1" />

            <InterModelMotifScorer weight = "10" />

            <PartnerMotifScorer weight = "10" />

        </AssemblyScorers>

        <AssemblyRequirements>

            <ClashRequirement clash_radius = "4" />

            <DsspSpecificLengthRequirement dssp_code = "L"
                maximum_length = "5" />

        </AssemblyRequirements>

    </AppendAssemblyMover>

```



</MOVERS>

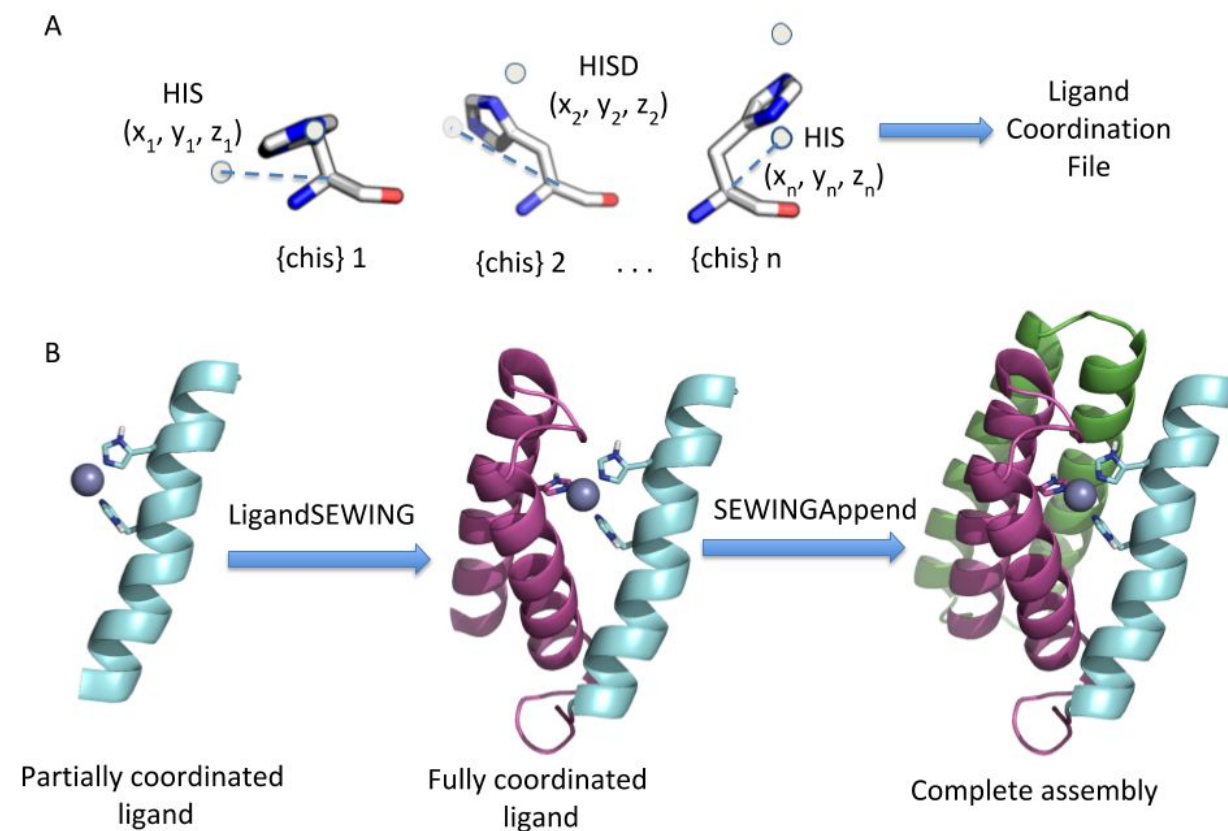
<PROTOCOLS>

<Add mover\_name="aam" />

</PROTOCOLS>

</ROSETTASCRIPTS>

**Figure 4.7:** Procedure for completing a ligand's coordination environment using LigandSEWING.



A) Users first generate a LigandCoordinationFile which stores the ligand atom's coordinates relative to the backbone of the coordinating residue in its most common rotamers. These files are prepared using an external Rosetta application and converted into a three-dimensional geometric hash to determine potential placements of ligand-coordinating residues during

LigandSEWING. B) Users begin with a partial ligand binding site. LigandBindingAssemblyMover (LigandSEWING) is used to complete the ligand's coordination, and SEWINGAppend (either separately or as part of the same protocol) is then used to complete the assembly.

**Figure 4.8:** Sample RosettaScript for using LigandSewing to complete the coordination of a tetrahedral zinc atom.

```

<ROSETTASCRIPTS>

<SCOREFXNS>

</SCOREFXNS>

<RESIDUE_SELECTORS>

    <ResidueName name="select_zn" residue_name3=" ZN" />

</RESIDUE_SELECTORS>

<FILTERS>

</FILTERS>

<MOVERS>

    <LigandBindingAssemblyMover                                name="assemble"
    binding_cycles="20000"
    model_file_name="smotifs_H_5_40_L_1_6_H_5_40.segments"
    start_temperature="2.0">

    <Ligands>

        <Ligand                                ligand_selector="select_zn"
        auto_detect_contacts="true" >

            <Coordination
            coordination_files="H_NEW_stats.txt"
            geometry_score_threshold="5" >

                <IdealContacts                    ligand_atom_name="ZN"
                max_coordinating_atoms="4"        angle="109.5"

```

```

        distance="2.2"                dihedral_1="30"
        dihedral_2="120" />

    </Coordination>

</Ligand>

</Ligands>

<AssemblyRequirements>

    <DsspSpecificLengthRequirement      dssp_code="L"
    maximum_length="6" /> Prevents super-long loops

    <DsspSpecificLengthRequirement      dssp_code="H"
    minimum_length="10" /> Prevents super-short helices

    <ClashRequirement />

    <SizeInSegmentsRequirement          maximum_size="9"
    minimum_size="5" />

    <LigandClashRequirement />

</AssemblyRequirements>

</LigandBindingAssemblyMover>

</MOVERS>

<APPLY_TO_POSE>

</APPLY_TO_POSE>

<PROTOCOLS>

    <Add mover_name="assemble" />

</PROTOCOLS>

</ROSETTASCRIPTS>

```

**Figure 4.9:** Sample of a ligand coordination file (for use with LigandSEWING). The first two lines of the file indicate the secondary structure (H, L, or E for helix, loop, or beta strand) and the ligand residue name, respectively, for which the file should be used. The third line contains headings labeling each column of the remaining lines in the file.

H

ZN

Coord_res_name	Coord_atom_number	Ligand_atom_number	Local_x	Local_y	Local_z	Chi1	Chi2	Chi3	Chi4
----------------	-------------------	--------------------	---------	---------	---------	------	------	------	------

HIS	71	-2.53973	-2.17178	-1.25977	1.70	.466	.200		
-----	----	----------	----------	----------	------	------	------	--	--

HIS	71	-2.54049	-2.2065	-1.23906	1.70	.466	.9400		
-----	----	----------	---------	----------	------	------	-------	--	--

HIS	71	-2.54116	-2.24095	-1.21791	1.70	.467	.6800		
-----	----	----------	----------	----------	------	------	-------	--	--

.

.

.

GLU91		-0.736518	-5.97938	1.14256	-63187.8	-23.30			
-------	--	-----------	----------	---------	----------	--------	--	--	--

## CHAPTER 5: ADVANCEMENTS IN REQUIREMENT-DRIVEN PROTEIN DESIGN

### 5.1 Introduction

The fundamental problem of protein design as it applies to large molecules, including entire proteins, is the sheer number of degrees of freedom involved (Debrunner (ed. ), Tsibris (ed., & Münck (ed. ), 1969); proteins are structured according to the sum total of interactions between individual atoms positioned according to the cumulative effect of dozens of torsional degrees of freedom, any one of which can have a profound impact on the overall predicted stability of the molecule when shifted even slightly (Anfinsen, 1973). Accordingly, the classical approach to the design of whole proteins has involved first the coarse positioning of backbone elements followed by progressively finer-grained perturbation of backbone and side chain torsions in an attempt to balance accuracy and speed; for example, a given structural prediction may begin by inserting 3mer or 9mer fragments with all of their associated torsions *en bloc* and finish by making small perturbations to backbone rotamers (Kicinski, n.d.).

While this one-directional approach is appropriate for designs that have some form of constraints on the final fold and therefore a single target conformation, requirement-driven design has no theoretical single energy minimum toward which

the algorithm is progressing and can therefore theoretically cycle through scales of structural change, at minimum alternating between the addition of new substructures and the optimization of the backbone torsions in the added components. This has several potential benefits for requirement-driven design as currently employed, the most significant being the ability to compensate for the granularity inherent to a finite set of substructures: under conventional requirement-driven design, all torsions must reach their optima simultaneously, despite having been in the optimal case built on a close approximation to the lowest-energy structure. Progressive structural optimization ameliorates the buildup of future structural strain that applies once the design is allowed to move in torsion space. This most significantly benefits proteins designed around collections of residues that must be placed with atomic precision but may be distant in sequence space, as with enzyme active sites and metal binding sites, as the flexibility of protein substructures may be exploited to fix the relative positions of those residues once they are in place and subsequently design further structure around the backbone conformation required to enable those relative orientations. Similarly, structures with high contact order interactions, including beta strand-containing structures, can be more efficiently optimized piecemeal; attempting to relax an entire beta sheet into existence would most probably result in the formation of localized beta hairpins of different twists that would be energetically discouraged from forming into sheets.

While there is no theoretical reason for requirement-driven design to be incapable of intercalating structural chimerization steps with torsion minimization,



there has been an insurmountable algorithmic barrier in the form of the different data structures under which each step operates. The chimerization of new structures has heretofore required a minimal data structure in which only the atomic coordinates of the backbone atoms are retained, transformed, and compared; since the substructures are effectively rigid, there is no need to store the kinematic layer defining the geometric relationships between atoms, and that data would add to the computational load involved in aligning substructures. That data is reintroduced to the simulation at the end of the requirement-driven design portion through a computationally expensive process of realigning fully kinematically-enabled residues according to the transformed coordinates, at which point the simulation proceeds as with normal residue-level design.

Fortunately, recent developments in Rosetta have made it possible to manipulate the substructures as full protein fragments in a way that is no longer prohibitively computationally intensive, making it possible to produce a new version of requirement-driven design in which the structural extension steps are modular, and may be inserted at arbitrary points in a residue-level design algorithm. This allows for a more progressive design methodology in which each addition may be locally optimized before the entire design is relaxed, and should consequently significantly improve the final design ensembles. Here, we present one implementation of such an algorithm, called Structural Extension With Anything (SewAnything), functionally identical to the older SEWING algorithm (Guffy, Teets, Langlois, & Kuhlman, 2018) except for operating on Rosetta's common data

structures rather than structures unique to itself. We have also expanded the algorithm's ability to insert metal-binding sites into proteins with atomic-level accuracy, and herein we describe how to enable that functionality.

## 5.2 Walkthrough and General Guidelines

SewAnything runs in either the RosettaScripts or the MultiStageRosettaScripts interface to Rosetta, and the syntax of the components is the same for both. The monolithic AssemblyMover of the original SEWING has been split into three components: the AddFlankingVirtualResiduesMover, which prepares the starting structures for design by adding placeholder “virtual” residues to one or both ends of the starting chain; the SewAnythingAddMover, which selects virtual residues in the chain and attempts to add non-clashing subunits from a provided file into them; and the PoseCompatibleMotifScorer, which scores the entire design or subsets of it. The basic workflow, then, is to run the AddFlankingVirtualResiduesMover once to prepare the design, then run as many cycles of the SewAnythingAddMover as are required to build a protein of the desired length, using the PoseCompatibleMotifScorer to select the best-scoring subset for downstream design. Backbone torsional sampling can be introduced in between calls to SewAnythingAddMover, and should be restricted to loops.

## 5.3 Vital Residues and Terminus Control: The AddFlankingVirtualResiduesMover

While SewAnything operates on the same Pose representation of a protein as the rest of Rosetta, it still requires additional information to set up the design run: specifically, the maximum size of the protein and the position of the initial structure

within that envelope is controlled by the spans of virtual residues added to the Pose in this setup step. If, for example, only C-terminal additions are desired, virtual residues need to be added only to the C terminus. Furthermore, those residues which cannot be modified or chimerized over during the structural extension process, termed “vital” residues, can be marked at this step. The `AddFlankingVirtualResiduesMover` does both of these processes simultaneously, taking a number of virtual residues to be added to either terminus and a list of residues to mark as unmodifiable and annotating the Pose accordingly.

#### 5.4 Secondary Structural Envelopes: The `SewAnythingAddMover`

Unlike in previous iterations of SEWING, in `SewAnything` structural extension need not take place within a Monte Carlo context; instead of running a complete design, the `SewAnythingAddMover` attempts to add a single substructure from a provided list into any of the `SewAnything`-marked stretches of virtual residues added by `AddFlankingVirtualResiduesMover`. The only checks it performs are the terminal compatibility check and global clash check from classical SEWING, for which reason there is a tunable number of attempts per addition. What is returned is not the lowest-scoring result of these attempts but instead the first to pass both checks.

With the transition to a multi-step process for protein construction comes the ability to tailor the fragment set available to the `SewAnythingAddMover` on a per-substructure basis, for which reason a control is provided to specify the envelope of acceptable termini for each step; it is now technically possible, for example, to alternate helix- and strand-ended substructures to produce an

alpha/beta protein. For applications comparable to SEWING, however, this should remain fixed at “H” only.

### 5.5 Subsets and Scoring: The PoseCompatibleMotifScorer

Scoring is unchanged in a functional sense from SEWING’s array of MotifScorers, but as the underlying data structure is now a Pose and therefore accessible to ResidueSelectors, those individual MotifScorers are now represented by a single Scorer, the behavior of which may be adjusted through the residue subsets passed to it. Absent any selections, the PoseCompatibleMotifScorer will score all residues in the Pose against all other residues, provided they lie within the specified distance cutoff; given a single selector, it will score all the residues selected against all of the other residues selected. Given two selectors, it will score all of the residues of the first against all of the residues in the second. The PoseCompatibleMotifScorer also accepts a sequence gap parameter, excluding from the score tabulation any pair of residues separated by that many residues in sequence space or fewer. The classical MotifScorer may therefore be replicated by simply running the PoseCompatibleMotifScorer with default settings, while the PartnerMotifScorer equivalent is to select the sewn design and the partner chain and run the PoseCompatibleMotifScorer with those two selectors. The InterModelMotifScorer can be emulated by using the sequence gap parameter to exclude adjacent secondary structural elements. All of these MotifScorers can be combined into a weighted sum using a CombinedValueFilter, the value of which can subsequently be passed to a GenericSimulatedAnnealer or similar Monte Carlo

wrapper to serve as a sorting parameter.

## 5.6 Required Input Files and How to Generate Them

As SewAnything only supports the “hashless” version of the SEWING algorithm, the only new file required by the algorithm itself is the segment file, an annotated list of Protein DataBank (PDB) files from which the SewAnythingAddMover will draw when constructing the protein according to the provided information about terminal secondary structure length and Dynamic Secondary Structure Prediction (DSSP) code. SewAnythingAddMover can generate this file internally, but it can be prepared separately with the MakeSegmentFile application; in either case, the requisite files include a newline-delimited list of PDBs from which to extract segments and a motif file listing the permissible lengths and DSSP codes for a substructure to be valid, along with a directory into which the segment file PDBs and file listing will be loaded.

## 5.7 Optimal Operational Contexts

SewAnything is designed to operate in a parallel processing context, with the results of each step pooled and the best-scoring fraction selected for the next step. Accordingly, to make best use of computing resources throughout the run, the total number of trajectories remains constant, and so the number of trajectories per input is the reciprocal of the fraction of the outputs from the previous step kept and used as inputs; if the top 10% of the results are kept from one step, each will generate 10 parallel inputs for the next step. Thus it is convenient when planning a SewAnything run to consider this fraction as a relevant parameter rather than the actual number of

runs used, instead scaling the latter parameter to the resources available. As with previous versions of this algorithm, the number of steps is correlated with both the intended size of the final design and with the expected runtime; if the runtime of the algorithm must be adjusted, the torsional sampling steps can be scaled without impacting the size of the design.

## 5.8 Results

To test the impact of intercalating torsion optimization on the computational expense and predicted designability of the results of a SEWING run, we set up both SEWING and SewAnything design runs to produce a five-helix bundle starting from a two-helix hairpin, and proceeded to compare both the CPU-seconds per assembly and the median MotifScore (Figure 5.1). Each case was set to output 1000 independent trajectories; the assembly SEWING runs varied by the number of cycles per addition (1, 10, 100 and 1000), while the SewAnything runs varied in the fraction of each addition step which was propagated to start the next round of additions (1, 0.5, 0.25 and 0.1). Additional SewAnything runs included small and shear backbone torsion perturbation steps between each structural addition, restricted to loop residues. While SewAnything exhibited a modest improvement in median MotifScore for comparable runtime, the addition of torsion perturbation steps dramatically improved the scores without significantly increasing runtime.

To evaluate the degree to which the improvements in MotifScore realized by SewAnything translated into improved scores, we evaluated the performance of both SEWING and SewAnything at a standard requirement-driven design task: the

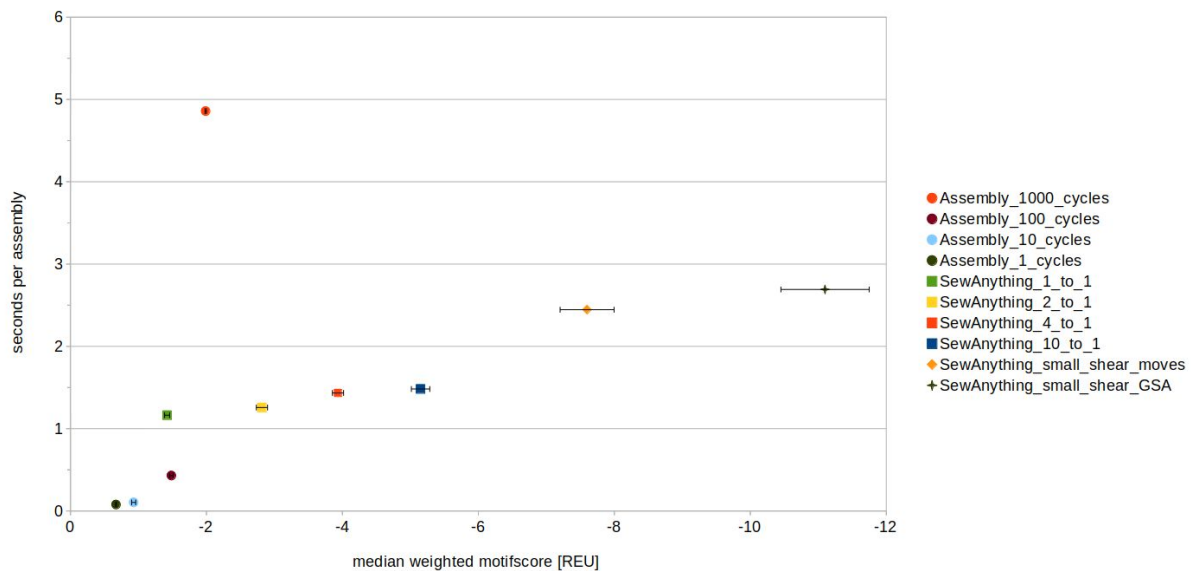
extension of one half of a protein-protein interface into a stable monomer to serve as an affinity reagent. In this case, the interface selected was that between the Dbl homology (DH) domain of the guanidine nucleotide exchange factor Vav1 and its autoinhibitory domain (AID) as represented in the crystal structure 3KY9(B. Yu et al., 2010). All SEWING and SewAnything runs were configured to produce 1000 candidate backbones, all of which were then designed at the residue level through the standard Rosetta FastRelax protocol. The average Rosetta per-residue score was evaluated for the designed monomer in each case, as was the predicted delta delta-G of the complex (Figure 5.2). While the results indicated that SEWING runs with high cycle numbers can exceed baseline SewAnything runs in monomer score, the addition of small and shear backbone torsion perturbations under Monte Carlo control again produced significant improvements in either monomer score or ddG depending on the weights assigned to each score term while retaining noticeable variability (Figure 5.3)

## 5.9 Conclusions

SewAnything represents a significant improvement over the standard SEWING algorithm in its ability to incorporate other protocols into any point in the requirement-driven design process, but by itself the improvements are marginal and driven primarily by the multistate design framework; it would, in theory, suffice to run SEWING to generate a much larger set of trajectories. Using this new capability to best advantage will require careful parameterization of the inserted residue-level design steps.

## 5.10 Figures and Tables

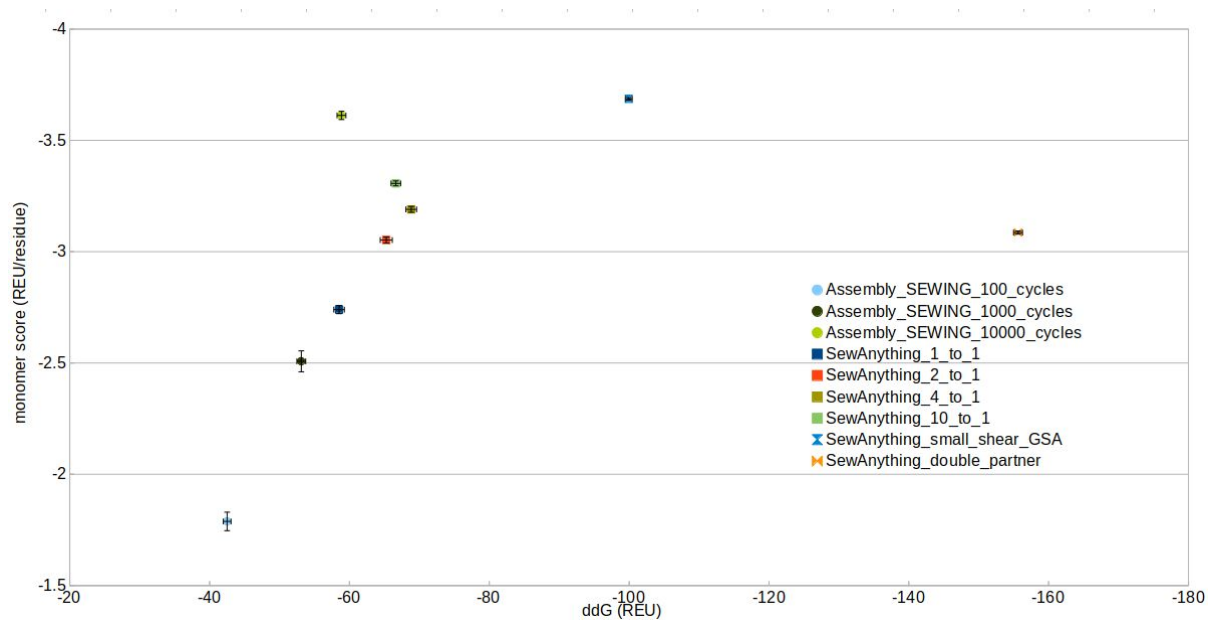
Figure 5.1 A comparison of the performance of assembly SEWING and SewAnything in extending a two-helix hairpin into a five-helix bundle.



A comparison of the performance of assembly SEWING and SewAnything in extending a two-helix hairpin into a five-helix bundle. While the SewAnything design runs (squares) were comparable in runtime to the Assembly SEWING runs (circles), the addition of small and shear backbone perturbation steps (diamond) dramatically improved the MotifScore, and placing those steps under the control of simulated annealing (cross) improved the score still further for little additional runtime.

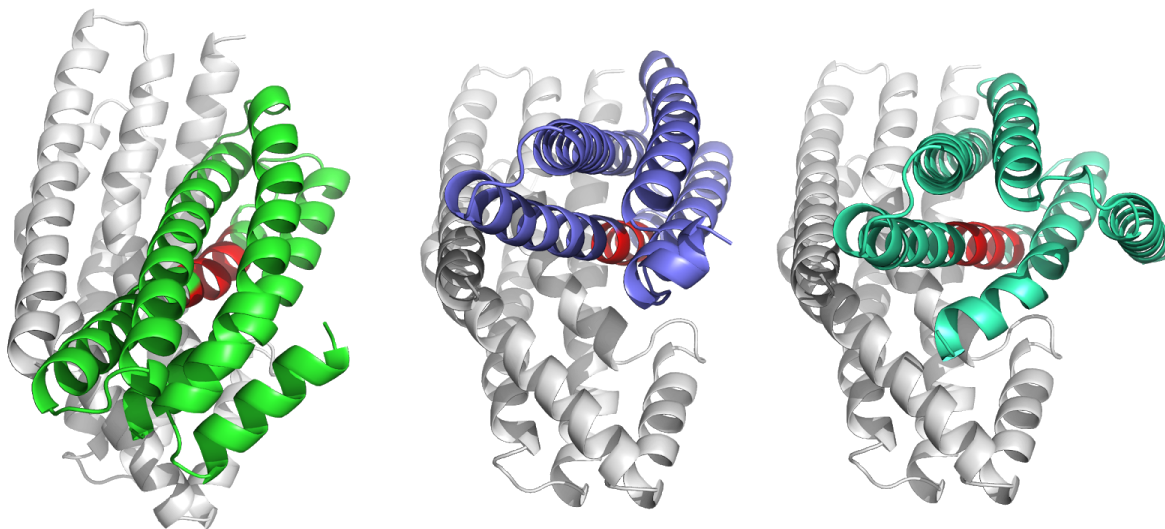


Figure 5.2 A comparison of the performance of assembly SEWING and SewAnything in generating an affinity reagent from the vav1 AID



A comparison of the performance of assembly SEWING and SewAnything in generating an affinity reagent from the vav1 AID. SEWING, when run with high cycle numbers, finds better-scoring designs than SewAnything without small and shear moves, but the predicted affinities of the SewAnything designs improve dramatically with the addition of small and shear moves, consistent with small backbone perturbations facilitating shape complementarity.

Figure 5.3 Selected designs from the small\_shear\_gsa SewAnything run used in Figure 5.2



Selected designs from the small\_shear\_gsa SewAnything run used in Figure 5.2; the autoinhibitory domain (red) was extended to form additional contacts to the Vav1 DH domain (grey).

Figure 5.4: The RosettaScript for running assembly SEWING in the 1-cycle case

```
<ROSETTASCRIPTS>

<RESIDUE_SELECTORS>

</RESIDUE_SELECTORS>

<TASKOPERATIONS>

</TASKOPERATIONS>

<SCOREFXNS>

</SCOREFXNS>

<FILTERS>

    <PoseCompatibleMotifScorer name="local" confidence = "0"/>

    <PoseCompatibleMotifScorer name="longrange" sequence_gap =
"20" confidence = "0"/>

    <CombinedValue name="weighted_motif" >

        <Add filter_name="local"/>

        <Add filter_name="longrange" factor = "10" />

    </CombinedValue>

</FILTERS>

<MOVERS>

    <AppendAssemblyMover name="aam"

model_file_name="/home/frank/generate_motif_file/smotifs_H_5_4
0_L_2_6_H_5_40.segments" hashed="false" max_segments="9"
minimum_cycles="1" maximum_cycles="1">
```

```

    <AssemblyRequirements>

        <ClashRequirement clash_radius = "4" />

        <SizeInSegmentsRequirement maximum_size="9"
minimum_size="9"/>

    </AssemblyRequirements>

</AppendAssemblyMover>

<FilterReportAsPoseExtraScoresMover
name="report_motifscore" report_as = "MS_local"
filter_name="weighted_motif"/>

</MOVERS>

<PROTOCOLS>

    <Add mover="aam"/>

    <Add mover="report_motifscore"/>

</PROTOCOLS>

</ROSETTASCRIPTS>

```

**Figure 5.5: The flags file for running assembly SEWING**

```
-parser:protocol assembly_sewing.xml

-l /home/frank/SewAnything_testing/start_pdb_list

-overwrite

-detect_disulf false


-mh:score:use_ss1 true          # applicable only to "BB_BB
motifs, match secondary structure on first (target) res"

-mh:score:use_ss2 true          # applicable only to "BB_BB
motifs, match secondary structure on second (binder) res"

-mh:score:use_aa1 false         # applicable only to "BB_BB
motifs, match AA identity on first (target) res"

-mh:score:use_aa2 false         # applicable only to "BB_BB
motifs, match AA identity on second (binder) res"


-mh:path:motifs

/home/frank/Rosetta/main/database/additional_protocol_data/sew
ing/xsmax_bb_ss_AILV_res10.8_msc0.3/xsmax_bb_ss_AILV_res10.8_m
sc0.3.rpm.bin.gz

-mh:path:scores_BB_BB

/home/frank/Rosetta/main/database/additional_protocol_data/sew
```

```
ing/xsmax_bb_ss_AILV_res10.8_msc0.3/xsmax_bb_ss_AILV_res10.8_m  
sc0.3
```

```
-mh:gen_reverse_motifs_on_load false    # I think the inverse  
motifs are already in the datafiles
```

```
-nstruct 1000
```

Figure 5.6: The RosettaScript for running assembly SEWING in the 10-cycle case

```
<ROSETTASCRIPTS>

<RESIDUE_SELECTORS>

</RESIDUE_SELECTORS>

<TASKOPERATIONS>

</TASKOPERATIONS>

<SCOREFXNS>

</SCOREFXNS>

<FILTERS>

    <PoseCompatibleMotifScorer name="local" confidence = "0"/>

    <PoseCompatibleMotifScorer name="longrange" sequence_gap =
"20" confidence = "0"/>

    <CombinedValue name="weighted_motif" >

        <Add filter_name="local"/>

        <Add filter_name="longrange" factor = "10" />

    </CombinedValue>

</FILTERS>

<MOVERS>

    <AppendAssemblyMover name="aam"

model_file_name="/home/frank/generate_motif_file/smotifs_H_5_4
0_L_2_6_H_5_40.segments" hashed="false" max_segments="9"

minimum_cycles="10" maximum_cycles="10">
```

```

    <AssemblyRequirements>

        <ClashRequirement clash_radius = "4" />

        <SizeInSegmentsRequirement maximum_size="9"
minimum_size="9"/>

    </AssemblyRequirements>

</AppendAssemblyMover>

<FilterReportAsPoseExtraScoresMover
name="report_motifscore" report_as = "MS_local"
filter_name="weighted_motif"/>

</MOVERS>

<PROTOCOLS>

    <Add mover="aam"/>

    <Add mover="report_motifscore"/>

</PROTOCOLS>

</ROSETTASCRIPTS>

```



Figure 5.7: The RosettaScript for running assembly SEWING in the 100-cycle case

```
<ROSETTASCRIPTS>

<RESIDUE_SELECTORS>

</RESIDUE_SELECTORS>

<TASKOPERATIONS>

</TASKOPERATIONS>

<SCOREFXNS>

</SCOREFXNS>

<FILTERS>

    <PoseCompatibleMotifScorer name="local" confidence = "0"/>

    <PoseCompatibleMotifScorer name="longrange" sequence_gap =
"20" confidence = "0"/>

    <CombinedValue name="weighted_motif" >

        <Add filter_name="local"/>

        <Add filter_name="longrange" factor = "10" />

    </CombinedValue>

</FILTERS>

<MOVERS>

    <AppendAssemblyMover name="aam"

model_file_name="/home/frank/generate_motif_file/smotifs_H_5_4
0_L_2_6_H_5_40.segments" hashed="false" max_segments="9"

minimum_cycles="100" maximum_cycles="100">
```

```

    <AssemblyRequirements>

        <ClashRequirement clash_radius = "4" />

        <SizeInSegmentsRequirement maximum_size="9"
minimum_size="9"/>

    </AssemblyRequirements>

</AppendAssemblyMover>

<FilterReportAsPoseExtraScoresMover
name="report_motifscore" report_as = "MS_local"
filter_name="weighted_motif"/>

</MOVERS>

<PROTOCOLS>

    <Add mover="aam"/>

    <Add mover="report_motifscore"/>

</PROTOCOLS>

</ROSETTASCRIPTS>

```

Figure 5.8: The RosettaScript for running assembly SEWING in the 1000-cycle case

```
<ROSETTASCRIPTS>

<RESIDUE_SELECTORS>

</RESIDUE_SELECTORS>

<TASKOPERATIONS>

</TASKOPERATIONS>

<SCOREFXNS>

</SCOREFXNS>

<FILTERS>

    <PoseCompatibleMotifScorer name="local" confidence = "0"/>

    <PoseCompatibleMotifScorer name="longrange" sequence_gap =
"20" confidence = "0"/>

    <CombinedValue name="weighted_motif" >

        <Add filter_name="local"/>

        <Add filter_name="longrange" factor = "10" />

    </CombinedValue>

</FILTERS>

<MOVERS>

    <AppendAssemblyMover name="aam"

model_file_name="/home/frank/generate_motif_file/smotifs_H_5_4
0_L_2_6_H_5_40.segments" hashed="false" max_segments="9"

minimum_cycles="1000" maximum_cycles="1000">
```

```

    <AssemblyRequirements>

        <ClashRequirement clash_radius = "4" />

        <SizeInSegmentsRequirement maximum_size="9"
minimum_size="9"/>

    </AssemblyRequirements>

</AppendAssemblyMover>

<FilterReportAsPoseExtraScoresMover
name="report_motifscore" report_as = "MS_local"
filter_name="weighted_motif"/>

</MOVERS>

<PROTOCOLS>

    <Add mover="aam"/>

    <Add mover="report_motifscore"/>

</PROTOCOLS>

</ROSETTASCRIPTS>

```

Figure 5.9: The RosettaScript for running pose SEWING in the 1-to-1 case

```
<JobDefinitionFile>

  <Job>

    <Input>

      <PDB

filename="/home/frank/sewanything_params/test_starting_hairpin
/starting_hairpin.pdb" />

      </Input>
    </Job>

  <Common>

    <SCOREFXNS>

  </SCOREFXNS>

  <RESIDUE_SELECTORS>

    <SecondaryStructure name="loops" overlap="0" minH="1"
minE="1" include_terminal_loops="false" ss="L" />

    <VirtualResidueSelector name="virtuals" />

    <Not name = "real" selector="virtuals"/>

    <And name = "real_loops" selectors="real,loops"/>

  </RESIDUE_SELECTORS>
```

```
<TASKOPERATIONS> 'to' is a suffix for all task-ops | repack  
interface, minimize all atoms
```

```
</TASKOPERATIONS>
```

```
<FILTERS>
```

```
<PoseCompatibleMotifScorer name="local" confidence = "0"/>
```

```
<PoseCompatibleMotifScorer name="longrange" sequence_gap =  
"20" confidence = "0"/>
```

```
<CombinedValue name="weighted_motif" >
```

```
<Add filter_name="local"/>
```

```
<Add filter_name="longrange" factor = "10" />
```

```
</CombinedValue>
```

```
<TrueFilter name="true_filter"/>
```

```
</FILTERS>
```

```
<MOVERS>
```

```
<AddFlankingVirtualResiduesMover name="virt"
```

```
N_term_residues="200" C_term_residues="200"/>
```

```
<ChimerizeSuperSecondaryStructuralSegmentsIntoFlankingVirtualR  
esiduesMover name="chim" window_width="2"
```

```

alignment_max_distance="1"

decompose_path="/home/frank/SewAnything_testing/decompose_test

" read_segments_from_decompose_file="true"

read_segments_from_pdb_list = "false"

permissible_segment_ends="H" model_file_name =

"test_motif_file" pdb_file_name = "test_pdbs"/>

  <FilterReportAsPoseExtraScoresMover

name="report_weighted_motifscore" report_as = "MS_weighted"

filter_name="weighted_motif"/>

  <FilterReportAsPoseExtraScoresMover

name="report_motifscore" report_as = "MS_local"

filter_name="local"/>

  <FilterReportAsPoseExtraScoresMover

name="report_longrange_motifscore" report_as = "MS_longrange"

filter_name="longrange"/>

  <Small name = "small_mover" residue_selector="real_loops"

nmoves = "10" angle_max="12"/>

  <GenericSimulatedAnnealer name="small_move_gsa"

mover_name="small_mover" filter_name="local" trials="10"

sample_type="low" temperature="0.1"/>

  <Shear name = "shear_mover" residue_selector="real_loops"

nmoves = "10" angle_max="12"/>

```

```

    <GenericSimulatedAnnealer name="shear_move_gsa"
mover_name="shear_mover" filter_name="local" trials="10"
sample_type="low" temperature="0.1"/>

</MOVERS>

<PROTOCOLS>

    <Stage num_runs_per_input_struct="1"
total_num_results_to_keep="1">

        <Add mover_name="virt"/>

        <Sort filter="true_filter"/>

    </Stage>

    <Stage num_runs_per_input_struct="1000"
total_num_results_to_keep="1000">

        <Add mover_name="chim"/>

        <Sort filter="true_filter"/>

    </Stage>

    <Stage num_runs_per_input_struct="1"
total_num_results_to_keep="1000">

        <Add mover_name="chim"/>

        <Sort filter="true_filter"/>

    </Stage>

```



```
<Stage num_runs_per_input_struct="1"
total_num_results_to_keep="1000">

    <Add mover_name="chim"/>

    <Add mover_name="report_motifscore"/>

    <Add mover_name="report_weighted_motifscore"/>

    <Add mover_name="report_longrange_motifscore"/>

    <Sort filter="true_filter"/>

</Stage>

</PROTOCOLS>

</Common>

</JobDefinitionFile>
```

Figure 5.10: The RosettaScript for running pose SEWING in the 2-to-1 case

```
<JobDefinitionFile>

<Job>

  <Input>

    <PDB

filename="/home/frank/sewanything_params/test_starting_hairpin
/starting_hairpin.pdb" />

    </Input>

  </Job>

<Common>

  <SCOREFXNS>

  </SCOREFXNS>

  <RESIDUE_SELECTORS>

    <SecondaryStructure name="loops" overlap="0" minH="1"
minE="1" include_terminal_loops="false" ss="L" />

    <VirtualResidueSelector name="virtuals" />

    <Not name = "real" selector="virtuals"/>

    <And name = "real_loops" selectors="real,loops"/>

  </RESIDUE_SELECTORS>
```

```
<TASKOPERATIONS> 'to' is a suffix for all task-ops | repack  
interface, minimize all atoms
```

```
</TASKOPERATIONS>
```

```
<FILTERS>
```

```
<PoseCompatibleMotifScorer name="local" confidence = "0"/>
```

```
<PoseCompatibleMotifScorer name="longrange" sequence_gap =  
"20" confidence = "0"/>
```

```
<CombinedValue name="weighted_motif" >
```

```
<Add filter_name="local"/>
```

```
<Add filter_name="longrange" factor = "10" />
```

```
</CombinedValue>
```

```
<TrueFilter name="true_filter"/>
```

```
</FILTERS>
```

```
<MOVERS>
```

```
<AddFlankingVirtualResiduesMover name="virt"
```

```
N_term_residues="200" C_term_residues="200"/>
```

```
<ChimerizeSuperSecondaryStructuralSegmentsIntoFlankingVirtualR  
esiduesMover name="chim" window_width="2"
```

```

alignment_max_distance="1"

decompose_path="/home/frank/SewAnything_testing/decompose_test

" read_segments_from_decompose_file="true"

read_segments_from_pdb_list = "false"

permissible_segment_ends="H" model_file_name =

"test_motif_file" pdb_file_name = "test_pdbs"/>

    <FilterReportAsPoseExtraScoresMover

name="report_weighted_motifscore" report_as = "MS_weighted"

filter_name="weighted_motif"/>

    <FilterReportAsPoseExtraScoresMover

name="report_motifscore" report_as = "MS_local"

filter_name="local"/>

    <FilterReportAsPoseExtraScoresMover

name="report_longrange_motifscore" report_as = "MS_longrange"

filter_name="longrange"/>

    <Small name = "small_mover" residue_selector="real_loops"

nmoves = "10" angle_max="12"/>

    <GenericSimulatedAnnealer name="small_move_gsa"

mover_name="small_mover" filter_name="local" trials="10"

sample_type="low" temperature="0.1"/>

    <Shear name = "shear_mover" residue_selector="real_loops"

nmoves = "10" angle_max="12"/>

```

```

    <GenericSimulatedAnnealer name="shear_move_gsa"
mover_name="shear_mover" filter_name="local" trials="10"
sample_type="low" temperature="0.1"/>

</MOVERS>

<PROTOCOLS>

    <Stage num_runs_per_input_struct="1"
total_num_results_to_keep="1">

        <Add mover_name="virt"/>

        <Sort filter="weighted_motif"/>

    </Stage>

    <Stage num_runs_per_input_struct="1000"
total_num_results_to_keep="500">

        <Add mover_name="chim"/>

        <Sort filter="weighted_motif"/>

    </Stage>

    <Stage num_runs_per_input_struct="2"
total_num_results_to_keep="500">

        <Add mover_name="chim"/>

        <Sort filter="weighted_motif"/>

    </Stage>

```

```

        <Stage num_runs_per_input_struct="2"
total_num_results_to_keep="1000">

            <Add mover_name="chim"/>

            <Sort filter="weighted_motif"/>

        </Stage>

        <Stage num_runs_per_input_struct="1"
total_num_results_to_keep="1000">

            <Add mover_name="report_motifscore"/>

            <Add mover_name="report_weighted_motifscore"/>

            <Add mover_name="report_longrange_motifscore"/>

            <Sort filter="true_filter"/>

        </Stage>

    </PROTOCOLS>

</Common>

</JobDefinitionFile>

```

Figure 5.11: The RosettaScript for running pose SEWING in the 4-to-1 case

```
<JobDefinitionFile>

<Job>

  <Input>

    <PDB

filename="/home/frank/sewanything_params/test_starting_hairpin
/starting_hairpin.pdb" />

  </Input>

</Job>

<Common>

  <SCOREFXNS>

  </SCOREFXNS>

  <RESIDUE_SELECTORS>

    <SecondaryStructure name="loops" overlap="0" minH="1"
minE="1" include_terminal_loops="false" ss="L" />

    <VirtualResidueSelector name="virtuals" />

    <Not name = "real" selector="virtuals"/>

    <And name = "real_loops" selectors="real,loops"/>

  </RESIDUE_SELECTORS>
```

```
<TASKOPERATIONS> 'to' is a suffix for all task-ops | repack  
interface, minimize all atoms
```

```
</TASKOPERATIONS>
```

```
<FILTERS>
```

```
<PoseCompatibleMotifScorer name="local" confidence = "0"/>
```

```
<PoseCompatibleMotifScorer name="longrange" sequence_gap =  
"20" confidence = "0"/>
```

```
<CombinedValue name="weighted_motif" >
```

```
<Add filter_name="local"/>
```

```
<Add filter_name="longrange" factor = "10" />
```

```
</CombinedValue>
```

```
<TrueFilter name="true_filter"/>
```

```
</FILTERS>
```

```
<MOVERS>
```

```
<AddFlankingVirtualResiduesMover name="virt"  
N_term_residues="200" C_term_residues="200"/>
```

```
<ChimerizeSuperSecondaryStructuralSegmentsIntoFlankingVirtualR  
esiduesMover name="chim" window_width="2"
```



```

alignment_max_distance="1"

decompose_path="/home/frank/SewAnything_testing/decompose_test

" read_segments_from_decompose_file="true"

read_segments_from_pdb_list = "false"

permissible_segment_ends="H" model_file_name =

"test_motif_file" pdb_file_name = "test_pdbs"/>

    <FilterReportAsPoseExtraScoresMover

name="report_weighted_motifscore" report_as = "MS_weighted"

filter_name="weighted_motif"/>

    <FilterReportAsPoseExtraScoresMover

name="report_motifscore" report_as = "MS_local"

filter_name="local"/>

    <FilterReportAsPoseExtraScoresMover

name="report_longrange_motifscore" report_as = "MS_longrange"

filter_name="longrange"/>

    <Small name = "small_mover" residue_selector="real_loops"

nmoves = "10" angle_max="12"/>

    <GenericSimulatedAnnealer name="small_move_gsa"

mover_name="small_mover" filter_name="local" trials="10"

sample_type="low" temperature="0.1"/>

    <Shear name = "shear_mover" residue_selector="real_loops"

nmoves = "10" angle_max="12"/>

```

```

    <GenericSimulatedAnnealer name="shear_move_gsa"
mover_name="shear_mover" filter_name="local" trials="10"
sample_type="low" temperature="0.1"/>

</MOVERS>

<PROTOCOLS>

    <Stage num_runs_per_input_struct="1"
total_num_results_to_keep="1">

        <Add mover_name="virt"/>

        <Sort filter="weighted_motif"/>

    </Stage>

    <Stage num_runs_per_input_struct="1000"
total_num_results_to_keep="250">

        <Add mover_name="chim"/>

        <Sort filter="weighted_motif"/>

    </Stage>

    <Stage num_runs_per_input_struct="4"
total_num_results_to_keep="250">

        <Add mover_name="chim"/>

        <Sort filter="weighted_motif"/>

    </Stage>

```

```

        <Stage num_runs_per_input_struct="4"
total_num_results_to_keep="1000">

            <Add mover_name="chim"/>

            <Sort filter="weighted_motif"/>

        </Stage>

        <Stage num_runs_per_input_struct="1"
total_num_results_to_keep="1000">

            <Add mover_name="report_motifscore"/>

            <Add mover_name="report_weighted_motifscore"/>

            <Add mover_name="report_longrange_motifscore"/>

            <Sort filter="true_filter"/>

        </Stage>

    </PROTOCOLS>

</Common>

</JobDefinitionFile>

```

Figure 5.12: The RosettaScript for running pose SEWING in the 10-to-1 case

```
<JobDefinitionFile>

<Job>

  <Input>

    <PDB

filename="/home/frank/sewanything_params/test_starting_hairpin
/starting_hairpin.pdb" />

  </Input>

</Job>

<Common>

  <SCOREFXNS>

  </SCOREFXNS>

  <RESIDUE_SELECTORS>

    <SecondaryStructure name="loops" overlap="0" minH="1"
minE="1" include_terminal_loops="false" ss="L" />

    <VirtualResidueSelector name="virtuals" />

    <Not name = "real" selector="virtuals"/>

    <And name = "real_loops" selectors="real,loops"/>

  </RESIDUE_SELECTORS>
```

```
<TASKOPERATIONS> 'to' is a suffix for all task-ops | repack  
interface, minimize all atoms
```

```
</TASKOPERATIONS>
```

```
<FILTERS>
```

```
<PoseCompatibleMotifScorer name="local" confidence = "0"/>
```

```
<PoseCompatibleMotifScorer name="longrange" sequence_gap =  
"20" confidence = "0"/>
```

```
<CombinedValue name="weighted_motif" >
```

```
<Add filter_name="local"/>
```

```
<Add filter_name="longrange" factor = "10" />
```

```
</CombinedValue>
```

```
<TrueFilter name="true_filter"/>
```

```
</FILTERS>
```

```
<MOVERS>
```

```
<AddFlankingVirtualResiduesMover name="virt"
```

```
N_term_residues="200" C_term_residues="200"/>
```

```
<ChimerizeSuperSecondaryStructuralSegmentsIntoFlankingVirtualR  
esiduesMover name="chim" window_width="2"
```

```

alignment_max_distance="1"

decompose_path="/home/frank/SewAnything_testing/decompose_test

" read_segments_from_decompose_file="true"

read_segments_from_pdb_list = "false"

permissible_segment_ends="H" model_file_name =

"test_motif_file" pdb_file_name = "test_pdbs"/>

    <FilterReportAsPoseExtraScoresMover

name="report_weighted_motifscore" report_as = "MS_weighted"

filter_name="weighted_motif"/>

    <FilterReportAsPoseExtraScoresMover

name="report_motifscore" report_as = "MS_local"

filter_name="local"/>

    <FilterReportAsPoseExtraScoresMover

name="report_longrange_motifscore" report_as = "MS_longrange"

filter_name="longrange"/>

    <Small name = "small_mover" residue_selector="real_loops"

nmoves = "10" angle_max="12"/>

    <GenericSimulatedAnnealer name="small_move_gsa"

mover_name="small_mover" filter_name="local" trials="10"

sample_type="low" temperature="0.1"/>

    <Shear name = "shear_mover" residue_selector="real_loops"

nmoves = "10" angle_max="12"/>

```

```

    <GenericSimulatedAnnealer name="shear_move_gsa"
mover_name="shear_mover" filter_name="local" trials="10"
sample_type="low" temperature="0.1"/>

</MOVERS>

<PROTOCOLS>

    <Stage num_runs_per_input_struct="1"
total_num_results_to_keep="1">

        <Add mover_name="virt"/>

        <Sort filter="weighted_motif"/>

    </Stage>

    <Stage num_runs_per_input_struct="1000"
total_num_results_to_keep="100">

        <Add mover_name="chim"/>

        <Sort filter="weighted_motif"/>

    </Stage>

    <Stage num_runs_per_input_struct="10"
total_num_results_to_keep="100">

        <Add mover_name="chim"/>

        <Sort filter="weighted_motif"/>

    </Stage>

```

```

        <Stage num_runs_per_input_struct="10"
total_num_results_to_keep="1000">

            <Add mover_name="chim"/>

            <Sort filter="weighted_motif"/>

        </Stage>

        <Stage num_runs_per_input_struct="1"
total_num_results_to_keep="1000">

            <Add mover_name="report_motifscore"/>

            <Add mover_name="report_weighted_motifscore"/>

            <Add mover_name="report_longrange_motifscore"/>

            <Sort filter="true_filter"/>

        </Stage>

    </PROTOCOLS>

</Common>

</JobDefinitionFile>

```



Figure 5.13: The RosettaScript for running pose SEWING with small and shear moves

```
<JobDefinitionFile>

<Job>

  <Input>

    <PDB

filename="/home/frank/sewanything_params/test_starting_hairpin
/starting_hairpin.pdb" />

  </Input>

</Job>

<Common>

  <SCOREFXNS>

  </SCOREFXNS>

  <RESIDUE_SELECTORS>

    <SecondaryStructure name="loops" overlap="0" minH="1"
minE="1" include_terminal_loops="false" ss="L" />

    <VirtualResidueSelector name="virtuals" />

    <Not name = "real" selector="virtuals"/>

    <And name = "real_loops" selectors="real,loops"/>
```

```

</RESIDUE_SELECTORS>

<TASKOPERATIONS> 'to' is a suffix for all task-ops | repack
interface, minimize all atoms

</TASKOPERATIONS>

<FILTERS>

  <PoseCompatibleMotifScorer name="local" confidence = "0"/>

  <PoseCompatibleMotifScorer name="longrange" sequence_gap =
"20" confidence = "0"/>

  <CombinedValue name="weighted_motif" >

    <Add filter_name="local"/>

    <Add filter_name="longrange" factor = "10" />

  </CombinedValue>

  <TrueFilter name="true_filter"/>

</FILTERS>

<MOVERS>

  <AddFlankingVirtualResiduesMover name="virt"
N_term_residues="200" C_term_residues="200"/>

<ChimerizeSuperSecondaryStructuralSegmentsIntoFlankingVirtualR

```

```

esiduesMover name="chim" window_width="2"

alignment_max_distance="1"

decompose_path="/home/frank/SewAnything_testing/decompose_test
" read_segments_from_decompose_file="true"

read_segments_from_pdb_list = "false"

permissible_segment_ends="H" model_file_name =
"test_motif_file" pdb_file_name = "test_pdbs"/>

  <FilterReportAsPoseExtraScoresMover

name="report_weighted_motifscore" report_as = "MS_weighted"

filter_name="weighted_motif"/>

  <FilterReportAsPoseExtraScoresMover

name="report_motifscore" report_as = "MS_local"

filter_name="local"/>

  <FilterReportAsPoseExtraScoresMover

name="report_longrange_motifscore" report_as = "MS_longrange"

filter_name="longrange"/>

  <Shear name = "shear_mover" residue_selector="real_loops"

nmoves = "10" angle_max="6"/>

  <Small name = "small_mover" residue_selector="real_loops"

nmoves = "10" angle_max="6"/>

</MOVERS>

```

```

<PROTOCOLS>

  <Stage num_runs_per_input_struct="1"
total_num_results_to_keep="1">

    <Add mover_name="virt"/>

    <Sort filter="weighted_motif"/>

  </Stage>

  <Stage num_runs_per_input_struct="1000"
total_num_results_to_keep="100">

    <Add mover_name="chim"/>

    <Sort filter="weighted_motif"/>

  </Stage>


  <Stage num_runs_per_input_struct="10"
total_num_results_to_keep="100">

    <Add mover_name="shear_mover"/>

    <Add mover_name="small_mover"/>

    <Sort filter="weighted_motif"/>

  </Stage>


  <Stage num_runs_per_input_struct="10"
total_num_results_to_keep="100">

    <Add mover_name="chim"/>

```

```

        <Sort filter="weighted_motif"/>
    </Stage>

    <Stage num_runs_per_input_struct="10"
total_num_results_to_keep="100">

        <Add mover_name="shear_mover"/>

        <Add mover_name="small_mover"/>

        <Sort filter="weighted_motif"/>
    </Stage>

    <Stage num_runs_per_input_struct="10"
total_num_results_to_keep="100">

        <Add mover_name="chim"/>

        <Sort filter="weighted_motif"/>
    </Stage>

    <Stage num_runs_per_input_struct="10"
total_num_results_to_keep="1000">

        <Add mover_name="shear_mover"/>

        <Add mover_name="small_mover"/>

        <Sort filter="weighted_motif"/>
    </Stage>

```

```
<Stage num_runs_per_input_struct="1"
total_num_results_to_keep="1000">

    <Add mover_name="report_motifscore"/>

    <Add mover_name="report_weighted_motifscore"/>

    <Add mover_name="report_longrange_motifscore"/>

    <Sort filter="true_filter"/>

</Stage>

</PROTOCOLS>

</Common>

</JobDefinitionFile>
```

Figure 5.14: The RosettaScript for running pose SEWING with small and shear moves under the control of the MotifScorer

```
<JobDefinitionFile>

<Job>

  <Input>

    <PDB

filename="/home/frank/sewanything_params/test_starting_hairpin
/starting_hairpin.pdb" /> EDIT THIS (i.e. scaffold.pdb)

    </Input>

  </Job>

<Common>

  <SCOREFXNS>

  </SCOREFXNS>

  <RESIDUE_SELECTORS>

    <SecondaryStructure name="loops" overlap="0" minH="1"
minE="1" include_terminal_loops="false" ss="L" />

    <VirtualResidueSelector name="virtuals" />

    <Not name = "real" selector="virtuals"/>

    <And name = "real_loops" selectors="real,loops"/>
```

```

</RESIDUE_SELECTORS>

<TASKOPERATIONS> 'to' is a suffix for all task-ops | repack
interface, minimize all atoms

</TASKOPERATIONS>

<FILTERS>

  <PoseCompatibleMotifScorer name="local" confidence = "0"/>

  <PoseCompatibleMotifScorer name="longrange" sequence_gap =
"20" confidence = "0"/>

  <CombinedValue name="weighted_motif" >

    <Add filter_name="local"/>

    <Add filter_name="longrange" factor = "10" />

  </CombinedValue>

  <TrueFilter name="true_filter"/>

</FILTERS>

<MOVERS>

  <AddFlankingVirtualResiduesMover name="virt"
N_term_residues="100" C_term_residues="100"/>

<ChimerizeSuperSecondaryStructuralSegmentsIntoFlankingVirtualR

```



```

esiduesMover name="chim" window_width="2"

alignment_max_distance="1"

decompose_path="/home/frank/SewAnything_testing/decompose_test
" read_segments_from_decompose_file="true"

read_segments_from_pdb_list = "false"

permissible_segment_ends="H" model_file_name =
"test_motif_file" pdb_file_name = "test_pdbs"/>

    <FilterReportAsPoseExtraScoresMover

name="report_weighted_motifscore" report_as = "MS_weighted"

filter_name="weighted_motif"/>

    <FilterReportAsPoseExtraScoresMover

name="report_motifscore" report_as = "MS_local"

filter_name="local"/>

    <FilterReportAsPoseExtraScoresMover

name="report_longrange_motifscore" report_as = "MS_longrange"

filter_name="longrange"/>

    <Shear name = "shear_mover" residue_selector="real_loops"

nmoves = "1" angle_max="6"/>

    <Small name = "small_mover" residue_selector="real_loops"

nmoves = "1" angle_max="6"/>

```

```
<GenericSimulatedAnnealer name="shear_move_gsa"
mover_name="shear_mover" filter_name="weighted_motif"
trials="10" sample_type="low" temperature="0.1"/>
```

```
<GenericSimulatedAnnealer name="small_move_gsa"
mover_name="small_mover" filter_name="weighted_motif"
trials="10" sample_type="low" temperature="0.1"/>
```

```
</MOVERS>
```

```
<PROTOCOLS>
```

```
<Stage num_runs_per_input_struct="1"
total_num_results_to_keep="1">
```

```
<Add mover_name="virt"/>
```

```
<Sort filter="weighted_motif"/>
```

```
</Stage>
```

```
<Stage num_runs_per_input_struct="1000"
total_num_results_to_keep="100">
```

```
<Add mover_name="chim"/>
```

```
<Sort filter="weighted_motif"/>
```

```
</Stage>
```

```
<Stage num_runs_per_input_struct="10"
total_num_results_to_keep="100">
```

```

        <Add mover_name="shear_move_gsa"/>
        <Add mover_name="small_move_gsa"/>
        <Sort filter="weighted_motif"/>
    </Stage>

    <Stage num_runs_per_input_struct="10"
total_num_results_to_keep="100">
        <Add mover_name="chim"/>
        <Sort filter="weighted_motif"/>
    </Stage>

    <Stage num_runs_per_input_struct="10"
total_num_results_to_keep="100">
        <Add mover_name="shear_move_gsa"/>
        <Add mover_name="small_move_gsa"/>
        <Sort filter="weighted_motif"/>
    </Stage>

    <Stage num_runs_per_input_struct="10"
total_num_results_to_keep="100">
        <Add mover_name="chim"/>
        <Sort filter="weighted_motif"/>

```

```

</Stage>

    <Stage num_runs_per_input_struct="10"
total_num_results_to_keep="1000">

        <Add mover_name="shear_move_gsa"/>

        <Add mover_name="small_move_gsa"/>

        <Sort filter="weighted_motif"/>
</Stage>

    <Stage num_runs_per_input_struct="1"
total_num_results_to_keep="1000">

        <Add mover_name="report_motifscore"/>

        <Add mover_name="report_weighted_motifscore"/>

        <Add mover_name="report_longrange_motifscore"/>

        <Sort filter="true_filter"/>
</Stage>

</PROTOCOLS>

</Common>

</JobDefinitionFile>

```

Figure 5.15: The RosettaScript for running assembly SEWING to generate a vav1 AID in the 100-cycle case

```
<ROSETTASCRIPTS>

<SCOREFXNS>

    <ScoreFunction name="common_sfxn"
weights="ref2015.wts"/>

</SCOREFXNS>

<RESIDUE_SELECTORS>

    <Chain name="chain_A" chains="2"/>

    <Chain name="chain_B" chains="1"/>

    <ResidueName name="critical_trp" residue_name3="TYR"/>

    <SecondaryStructure name="loops" overlap="0" minH="1"
minE="1" include_terminal_loops="false" ss="L" />

    <VirtualResidueSelector name="virtuals" />

    <Not name = "real" selector="virtuals"/>

    <And name = "real_loops" selectors="real,loops"/>

    <Neighborhood name="near_B" selector="chain_B" distance =
"6.0"/>

    <Not name = "not_near_B" selector="near_B"/>

    <And name = "A_distant_B" selectors="not_near_B,chain_A"/>
```

```

    <And name = "A_near_B" selectors="near_B,chain_A"/>

    <Or name = "A_and_A_near_B" selectors="A_near_B,chain_A"/>
</RESIDUE_SELECTORS>


<MOVE_MAP_FACTORIES>

    <MoveMapFactory name="partner_sewing_movemap" bb="0"
chi="0">

        <Backbone residue_selector="chain_B"/>

        <Chi residue_selector="A_and_A_near_B"/>

    </MoveMapFactory>
</MOVE_MAP_FACTORIES>


<TASKOPERATIONS> 'to' is a suffix for all task-ops | repack
interface, minimize all atoms

    <InitializeFromCommandline name="ifc"/>

    <IncludeCurrent name="ic"/>

    <ExtraRotamersGeneric ex1="true" ex2="false"
name="exlex2"/>

    <OperateOnResidueSubset name="repack_partner" selector =
"chain_A">

        <RestrictToRepackingRLT/>

    </OperateOnResidueSubset>

```

```

    <OperateOnResidueSubset
name="do_not_repack_distant_partner" selector = "A_distant_B">
        <PreventRepackingRLT/>
    </OperateOnResidueSubset>
</TASKOPERATIONS>

<FILTERS>

    <PoseCompatibleMotifScorer name="local_by_interaction"
first_selector="chain_B" normalize_by_interactions="true"
confidence = "0"/>

    <PoseCompatibleMotifScorer
name="longrange_by_interaction" first_selector="chain_B"
sequence_gap = "20" normalize_by_interactions="true"
confidence = "0"/>

    <PoseCompatibleMotifScorer name="partner_by_interaction"
first_selector="chain_A" second_selector="chain_B"
normalize_by_interactions="true" confidence = "0"/>

    <CombinedValue name="weighted_motif_by_interaction" >
        <Add filter_name="local_by_interaction"/>

```

```

        <Add filter_name="longrange_by_interaction" factor =
"10" />

        <Add filter_name="partner_by_interaction" factor = "10"
/>

    </CombinedValue>

    <PoseCompatibleMotifScorer name="local_by_residue"
first_selector="chain_B" normalize_by_interactions="false"
confidence = "0"/>

    <PoseCompatibleMotifScorer name="longrange_by_residue"
first_selector="chain_B" sequence_gap = "20"
normalize_by_interactions="false" confidence = "0"/>

    <PoseCompatibleMotifScorer name="partner_by_residue"
first_selector="chain_A" second_selector="chain_B"
normalize_by_interactions="false" confidence = "0"/>

    <CombinedValue name="weighted_motif_by_residue" >

        <Add filter_name="local_by_residue"/>

        <Add filter_name="longrange_by_residue" factor = "10" />

        <Add filter_name="partner_by_residue" factor = "10" />

    </CombinedValue>

```



```

    <PoseCompatibleMotifScorer name="local"
first_selector="chain_B" normalize_by_interactions="false"
normalize_by_residues="false" confidence = "0"/>

    <PoseCompatibleMotifScorer name="longrange"
first_selector="chain_B" sequence_gap = "20"
normalize_by_interactions="false"
normalize_by_residues="false" confidence = "0"/>

    <PoseCompatibleMotifScorer name="partner"
first_selector="chain_A" second_selector="chain_B"
normalize_by_interactions="false"
normalize_by_residues="false" confidence = "0"/>

    <CombinedValue name="weighted_motif" >
        <Add filter_name="local"/>
        <Add filter_name="longrange" factor = "10" />
        <Add filter_name="partner" factor = "10" />
    </CombinedValue>

    <ResidueCount name="size" confidence="0"/>

    <ResidueCount name="size_of_A" confidence="0"
residue_selector="chain_A"/>

```

```

    <ResidueCount name="size_of_B" confidence="0"
residue_selector="chain_B"/>

    <ScoreType name="sfxn_filter" score_type="total_score"
scorefxn="common_sfxn" threshold="999999"/>

    <CalculatorFilter name="normalized_score"
equation="total_score / res" confidence="0">

        <VAR name="total_score" filter="sfxn_filter"/>

        <VAR name="res" filter="size"/>

    </CalculatorFilter>

</FILTERS>

<MOVERS>

    <AppendAssemblyMover name="aam"

model_file_name="/nas/longleaf/home/frankdt/scratch/multistage
_SEWING/relax_tests/assembly_sewing/inputs/smotifs_H_5_40_L_2_
6_H_5_40.segments" hashed="false" max_segments="9"
minimum_cycles="100" maximum_cycles="100"
recover_lowest_assembly = "true" partner_pdb =
"vav1_partner.pdb" required_resnums="8">

        <AssemblyRequirements>

```

```

        <ClashRequirement clash_radius =
"4" />

        <SizeInSegmentsRequirement
maximum_size="9" minimum_size="9"/>

    </AssemblyRequirements>

    <AssemblyScorers>

        <MotifScorer weight="1"/>

        <InterModelMotifScorer weight="10"/>

        <PartnerMotifScorer weight="10"/>

    </AssemblyScorers>

</AppendAssemblyMover>

    <AddFlankingVirtualResiduesMover name="virt"
N_term_residues="200" C_term_residues="200"/>

<ChimerizeSuperSecondaryStructuralSegmentsIntoFlankingVirtualR
esiduesMover name="chim" window_width="2"
alignment_max_distance="0.5"
decompose_path="/nas/longleaf/home/frankdt/scratch/multistage_
SEWING/decompose_test"
read_segments_from_decompose_file="true"
read_segments_from_pdb_list = "false"

```

```

permissible_segment_ends="H" model_file_name =
"test_motif_file" pdb_file_name = "test_pdbs"/>

    <FilterReportAsPoseExtraScoresMover
name="report_weighted_motifscore_by_residue" report_as =
"MS_weighted_by_residue"
filter_name="weighted_motif_by_residue"/>

    <FilterReportAsPoseExtraScoresMover
name="report_motifscore_by_residue" report_as =
"MS_local_by_residue" filter_name="local_by_residue"/>

    <FilterReportAsPoseExtraScoresMover
name="report_longrange_motifscore_by_residue" report_as =
"MS_longrange_by_residue" filter_name="longrange_by_residue"/>

    <FilterReportAsPoseExtraScoresMover
name="report_partner_motifscore_by_residue" report_as =
"MS_partner_by_residue" filter_name="partner_by_residue"/>

    <FilterReportAsPoseExtraScoresMover
name="report_weighted_motifscore_by_interaction" report_as =
"MS_weighted_by_interaction"
filter_name="weighted_motif_by_interaction"/>

```

```

    <FilterReportAsPoseExtraScoresMover
name="report_motifscore_by_interaction" report_as =
"MS_local_by_interaction" filter_name="local_by_interaction"/>

    <FilterReportAsPoseExtraScoresMover
name="report_longrange_motifscore_by_interaction" report_as =
"MS_longrange_by_interaction"
filter_name="longrange_by_interaction"/>

    <FilterReportAsPoseExtraScoresMover
name="report_partner_motifscore_by_interaction" report_as =
"MS_partner_by_interaction"
filter_name="partner_by_interaction"/>

    <FilterReportAsPoseExtraScoresMover
name="report_weighted_motifscore" report_as = "MS_weighted"
filter_name="weighted_motif"/>

    <FilterReportAsPoseExtraScoresMover
name="report_motifscore" report_as = "MS_local"
filter_name="local"/>

    <FilterReportAsPoseExtraScoresMover
name="report_longrange_motifscore" report_as = "MS_longrange"
filter_name="longrange"/>

```

```

    <FilterReportAsPoseExtraScoresMover
name="report_partner_motifscore" report_as = "MS_partner"
filter_name="partner"/>

    <FilterReportAsPoseExtraScoresMover name="report_size"
report_as = "size" filter_name="size"/>

    <FilterReportAsPoseExtraScoresMover
name="report_size_of_A" report_as = "size_of_A"
filter_name="size_of_A"/>

    <FilterReportAsPoseExtraScoresMover
name="report_size_of_B" report_as = "size_of_B"
filter_name="size_of_B"/>

    <FilterReportAsPoseExtraScoresMover
name="report_normalized_score_of_complex" report_as =
"normalized_score_of_complex" filter_name="normalized_score"/>

    <FilterReportAsPoseExtraScoresMover
name="report_normalized_score_of_design" report_as =
"normalized_score_of_design" filter_name="normalized_score"/>

    <InterfaceAnalyzerMover name="interface_analyze" scorefxn
= "common_sfxn" pack_separated="false" pack_input = "false"
jump="1"/>

```

```

    <FastRelax disable_design="false" name="relax_1"
repeats="5" scorefxn="common_sfxn"
task_operations="ifc,ic,exlex2,repack_partner,do_not_repack_di
stant_partner" movemap_factory = "partner_sewing_movemap"
relaxscript="rosettacon2018"/>

```

```

    <DeleteRegionMover name="delete_partner"
residue_selector="chain_A"/>

```

```

</MOVERS>

```

```

<PROTOCOLS>

```

```

    <Add mover_name="aam"/>

```

```

    <Add

```

```

mover_name="report_weighted_motifscore_by_residue"/>

```

```

    <Add

```

```

mover_name="report_longrange_motifscore_by_residue"/>

```

```

    <Add mover_name="report_partner_motifscore_by_residue"/>

```

```

    <Add

```

```

mover_name="report_weighted_motifscore_by_residue"/>

```

```

    <Add

```

```

mover_name="report_weighted_motifscore_by_interaction"/>

```

```

    <Add
mover_name="report_longrange_motifscore_by_interaction"/>

    <Add
mover_name="report_partner_motifscore_by_interaction"/>

    <Add
mover_name="report_weighted_motifscore_by_interaction"/>

    <Add mover_name="report_weighted_motifscore"/>

    <Add mover_name="report_longrange_motifscore"/>

    <Add mover_name="report_partner_motifscore"/>

    <Add mover_name="report_weighted_motifscore"/>

    <Add mover_name="relax_1"/>

    <Add mover_name="interface_analyze"/>

    <Add mover_name="report_size"/>

    <Add mover_name="report_normalized_score_of_complex"/>

    <Add mover_name="delete_partner"/>

    <Add mover_name="report_normalized_score_of_design"/>

    <Add filter_name ="sfxn_filter"/>

</PROTOCOLS>

</ROSETTASCRIPTS>

```



Figure 5.16: The RosettaScript for running assembly SEWING to generate a vav1 AID in the 1000-cycle case

```
<ROSETTASCRIPTS>

  <SCOREFXNS>

    <ScoreFunction name="common_sfxn"
weights="ref2015.wts"/>

  </SCOREFXNS>

  <RESIDUE_SELECTORS>

    <Chain name="chain_A" chains="2"/>

    <Chain name="chain_B" chains="1"/>

    <ResidueName name="critical_trp" residue_name3="TYR"/>

    <SecondaryStructure name="loops" overlap="0" minH="1"
minE="1" include_terminal_loops="false" ss="L" />

    <VirtualResidueSelector name="virtuals" />

    <Not name = "real" selector="virtuals"/>

    <And name = "real_loops" selectors="real,loops"/>

    <Neighborhood name="near_B" selector="chain_B" distance =
"6.0"/>

    <Not name = "not_near_B" selector="near_B"/>

    <And name = "A_distant_B" selectors="not_near_B,chain_A"/>
```

```

    <And name = "A_near_B" selectors="near_B,chain_A"/>

    <Or name = "A_and_A_near_B" selectors="A_near_B,chain_A"/>
</RESIDUE_SELECTORS>


<MOVE_MAP_FACTORIES>

    <MoveMapFactory name="partner_sewing_movemap" bb="0"
chi="0">

        <Backbone residue_selector="chain_B"/>

        <Chi residue_selector="A_and_A_near_B"/>

    </MoveMapFactory>
</MOVE_MAP_FACTORIES>


<TASKOPERATIONS> 'to' is a suffix for all task-ops | repack
interface, minimize all atoms

    <InitializeFromCommandline name="ifc"/>

    <IncludeCurrent name="ic"/>

    <ExtraRotamersGeneric ex1="true" ex2="false"
name="exlex2"/>

    <OperateOnResidueSubset name="repack_partner" selector =
"chain_A">

        <RestrictToRepackingRLT/>

    </OperateOnResidueSubset>

```

```

    <OperateOnResidueSubset
name="do_not_repack_distant_partner" selector = "A_distant_B">
        <PreventRepackingRLT/>
    </OperateOnResidueSubset>
</TASKOPERATIONS>

<FILTERS>

    <PoseCompatibleMotifScorer name="local_by_interaction"
first_selector="chain_B" normalize_by_interactions="true"
confidence = "0"/>

    <PoseCompatibleMotifScorer
name="longrange_by_interaction" first_selector="chain_B"
sequence_gap = "20" normalize_by_interactions="true"
confidence = "0"/>

    <PoseCompatibleMotifScorer name="partner_by_interaction"
first_selector="chain_A" second_selector="chain_B"
normalize_by_interactions="true" confidence = "0"/>

    <CombinedValue name="weighted_motif_by_interaction" >
        <Add filter_name="local_by_interaction"/>

```

```

        <Add filter_name="longrange_by_interaction" factor =
"10" />

        <Add filter_name="partner_by_interaction" factor = "10"
/>

    </CombinedValue>

    <PoseCompatibleMotifScorer name="local_by_residue"
first_selector="chain_B" normalize_by_interactions="false"
confidence = "0"/>

    <PoseCompatibleMotifScorer name="longrange_by_residue"
first_selector="chain_B" sequence_gap = "20"
normalize_by_interactions="false" confidence = "0"/>

    <PoseCompatibleMotifScorer name="partner_by_residue"
first_selector="chain_A" second_selector="chain_B"
normalize_by_interactions="false" confidence = "0"/>

    <CombinedValue name="weighted_motif_by_residue" >

        <Add filter_name="local_by_residue"/>

        <Add filter_name="longrange_by_residue" factor = "10" />

        <Add filter_name="partner_by_residue" factor = "10" />

    </CombinedValue>

```

```

    <PoseCompatibleMotifScorer name="local"
first_selector="chain_B" normalize_by_interactions="false"
normalize_by_residues="false" confidence = "0"/>

    <PoseCompatibleMotifScorer name="longrange"
first_selector="chain_B" sequence_gap = "20"
normalize_by_interactions="false"
normalize_by_residues="false" confidence = "0"/>

    <PoseCompatibleMotifScorer name="partner"
first_selector="chain_A" second_selector="chain_B"
normalize_by_interactions="false"
normalize_by_residues="false" confidence = "0"/>

    <CombinedValue name="weighted_motif" >
        <Add filter_name="local"/>
        <Add filter_name="longrange" factor = "10" />
        <Add filter_name="partner" factor = "10" />
    </CombinedValue>

    <ResidueCount name="size" confidence="0"/>

    <ResidueCount name="size_of_A" confidence="0"
residue_selector="chain_A"/>

```

```

    <ResidueCount name="size_of_B" confidence="0"
residue_selector="chain_B"/>

    <ScoreType name="sfxn_filter" score_type="total_score"
scorefxn="common_sfxn" threshold="999999"/>

    <CalculatorFilter name="normalized_score"
equation="total_score / res" confidence="0">

        <VAR name="total_score" filter="sfxn_filter"/>

        <VAR name="res" filter="size"/>

    </CalculatorFilter>

</FILTERS>

<MOVERS>

    <AppendAssemblyMover name="aam"

model_file_name="/nas/longleaf/home/frankdt/scratch/multistage
_SEWING/relax_tests/assembly_sewing/inputs/smotifs_H_5_40_L_2_
6_H_5_40.segments" hashed="false" max_segments="9"
minimum_cycles="1000" maximum_cycles="1000"
recover_lowest_assembly = "true" partner_pdb =
"vav1_partner.pdb" required_resnums="8">

        <AssemblyRequirements>

```

```

        <ClashRequirement clash_radius =
"4" />

        <SizeInSegmentsRequirement
maximum_size="9" minimum_size="9"/>

    </AssemblyRequirements>

    <AssemblyScorers>

        <MotifScorer weight="1"/>

        <InterModelMotifScorer weight="10"/>

        <PartnerMotifScorer weight="10"/>

    </AssemblyScorers>

</AppendAssemblyMover>

    <AddFlankingVirtualResiduesMover name="virt"
N_term_residues="200" C_term_residues="200"/>

<ChimerizeSuperSecondaryStructuralSegmentsIntoFlankingVirtualR
esiduesMover name="chim" window_width="2"
alignment_max_distance="0.5"
decompose_path="/nas/longleaf/home/frankdt/scratch/multistage_
SEWING/decompose_test"
read_segments_from_decompose_file="true"
read_segments_from_pdb_list = "false"

```

```

permissible_segment_ends="H" model_file_name =
"test_motif_file" pdb_file_name = "test_pdbs"/>

<FilterReportAsPoseExtraScoresMover
name="report_weighted_motifscore_by_residue" report_as =
"MS_weighted_by_residue"
filter_name="weighted_motif_by_residue"/>

<FilterReportAsPoseExtraScoresMover
name="report_motifscore_by_residue" report_as =
"MS_local_by_residue" filter_name="local_by_residue"/>

<FilterReportAsPoseExtraScoresMover
name="report_longrange_motifscore_by_residue" report_as =
"MS_longrange_by_residue" filter_name="longrange_by_residue"/>

<FilterReportAsPoseExtraScoresMover
name="report_partner_motifscore_by_residue" report_as =
"MS_partner_by_residue" filter_name="partner_by_residue"/>

<FilterReportAsPoseExtraScoresMover
name="report_weighted_motifscore_by_interaction" report_as =
"MS_weighted_by_interaction"
filter_name="weighted_motif_by_interaction"/>

```



```

    <FilterReportAsPoseExtraScoresMover
name="report_motifscore_by_interaction" report_as =
"MS_local_by_interaction" filter_name="local_by_interaction"/>

    <FilterReportAsPoseExtraScoresMover
name="report_longrange_motifscore_by_interaction" report_as =
"MS_longrange_by_interaction"
filter_name="longrange_by_interaction"/>

    <FilterReportAsPoseExtraScoresMover
name="report_partner_motifscore_by_interaction" report_as =
"MS_partner_by_interaction"
filter_name="partner_by_interaction"/>

    <FilterReportAsPoseExtraScoresMover
name="report_weighted_motifscore" report_as = "MS_weighted"
filter_name="weighted_motif"/>

    <FilterReportAsPoseExtraScoresMover
name="report_motifscore" report_as = "MS_local"
filter_name="local"/>

    <FilterReportAsPoseExtraScoresMover
name="report_longrange_motifscore" report_as = "MS_longrange"
filter_name="longrange"/>

```

```

    <FilterReportAsPoseExtraScoresMover
name="report_partner_motifscore" report_as = "MS_partner"
filter_name="partner"/>

    <FilterReportAsPoseExtraScoresMover name="report_size"
report_as = "size" filter_name="size"/>

    <FilterReportAsPoseExtraScoresMover
name="report_size_of_A" report_as = "size_of_A"
filter_name="size_of_A"/>

    <FilterReportAsPoseExtraScoresMover
name="report_size_of_B" report_as = "size_of_B"
filter_name="size_of_B"/>

    <FilterReportAsPoseExtraScoresMover
name="report_normalized_score_of_complex" report_as =
"normalized_score_of_complex" filter_name="normalized_score"/>

    <FilterReportAsPoseExtraScoresMover
name="report_normalized_score_of_design" report_as =
"normalized_score_of_design" filter_name="normalized_score"/>

    <InterfaceAnalyzerMover name="interface_analyze" scorefxn
= "common_sfxn" pack_separated="false" pack_input = "false"
jump="1"/>

```

```

    <FastRelax disable_design="false" name="relax_1"
repeats="5" scorefxn="common_sfxn"
task_operations="ifc,ic,exlex2,repack_partner,do_not_repack_di
stant_partner" movemap_factory = "partner_sewing_movemap"
relaxscript="rosettacon2018"/>

```

```

    <DeleteRegionMover name="delete_partner"
residue_selector="chain_A"/>

```

```

</MOVERS>

```

```

<PROTOCOLS>

```

```

    <Add mover_name="aam"/>

```

```

    <Add

```

```

mover_name="report_weighted_motifscore_by_residue"/>

```

```

    <Add

```

```

mover_name="report_longrange_motifscore_by_residue"/>

```

```

    <Add mover_name="report_partner_motifscore_by_residue"/>

```

```

    <Add

```

```

mover_name="report_weighted_motifscore_by_residue"/>

```

```

    <Add

```

```

mover_name="report_weighted_motifscore_by_interaction"/>

```

```

    <Add
mover_name="report_longrange_motifscore_by_interaction"/>

    <Add
mover_name="report_partner_motifscore_by_interaction"/>

    <Add
mover_name="report_weighted_motifscore_by_interaction"/>

    <Add mover_name="report_weighted_motifscore"/>

    <Add mover_name="report_longrange_motifscore"/>

    <Add mover_name="report_partner_motifscore"/>

    <Add mover_name="report_weighted_motifscore"/>

    <Add mover_name="relax_1"/>

    <Add mover_name="interface_analyze"/>

    <Add mover_name="report_size"/>

    <Add mover_name="report_normalized_score_of_complex"/>

    <Add mover_name="delete_partner"/>

    <Add mover_name="report_normalized_score_of_design"/>

    <Add filter_name ="sfxn_filter"/>

</PROTOCOLS>

</ROSETTASCRIPTS>

```

Figure 5.17: The RosettaScript for running assembly SEWING to generate a vav1 AID in the 10000-cycle case

```
<ROSETTASCRIPTS>

  <SCOREFXNS>

    <ScoreFunction name="common_sfxn"
weights="ref2015.wts"/>

  </SCOREFXNS>

  <RESIDUE_SELECTORS>

    <Chain name="chain_A" chains="2"/>

    <Chain name="chain_B" chains="1"/>

    <ResidueName name="critical_trp" residue_name3="TYR"/>

    <SecondaryStructure name="loops" overlap="0" minH="1"
minE="1" include_terminal_loops="false" ss="L" />

    <VirtualResidueSelector name="virtuals" />

    <Not name = "real" selector="virtuals"/>

    <And name = "real_loops" selectors="real,loops"/>

    <Neighborhood name="near_B" selector="chain_B" distance =
"6.0"/>

    <Not name = "not_near_B" selector="near_B"/>

    <And name = "A_distant_B" selectors="not_near_B,chain_A"/>
```

```

    <And name = "A_near_B" selectors="near_B,chain_A"/>

    <Or name = "A_and_A_near_B" selectors="A_near_B,chain_A"/>
</RESIDUE_SELECTORS>


<MOVE_MAP_FACTORIES>

    <MoveMapFactory name="partner_sewing_movemap" bb="0"
chi="0">

        <Backbone residue_selector="chain_B"/>

        <Chi residue_selector="A_and_A_near_B"/>

    </MoveMapFactory>
</MOVE_MAP_FACTORIES>


<TASKOPERATIONS> 'to' is a suffix for all task-ops | repack
interface, minimize all atoms

    <InitializeFromCommandline name="ifc"/>

    <IncludeCurrent name="ic"/>

    <ExtraRotamersGeneric ex1="true" ex2="false"
name="exlex2"/>

    <OperateOnResidueSubset name="repack_partner" selector =
"chain_A">

        <RestrictToRepackingRLT/>

    </OperateOnResidueSubset>

```

```

    <OperateOnResidueSubset
name="do_not_repack_distant_partner" selector = "A_distant_B">
        <PreventRepackingRLT/>
    </OperateOnResidueSubset>
</TASKOPERATIONS>

<FILTERS>

    <PoseCompatibleMotifScorer name="local_by_interaction"
first_selector="chain_B" normalize_by_interactions="true"
confidence = "0"/>

    <PoseCompatibleMotifScorer
name="longrange_by_interaction" first_selector="chain_B"
sequence_gap = "20" normalize_by_interactions="true"
confidence = "0"/>

    <PoseCompatibleMotifScorer name="partner_by_interaction"
first_selector="chain_A" second_selector="chain_B"
normalize_by_interactions="true" confidence = "0"/>

    <CombinedValue name="weighted_motif_by_interaction" >
        <Add filter_name="local_by_interaction"/>

```

```

        <Add filter_name="longrange_by_interaction" factor =
"10" />

        <Add filter_name="partner_by_interaction" factor = "10"
/>

    </CombinedValue>

    <PoseCompatibleMotifScorer name="local_by_residue"
first_selector="chain_B" normalize_by_interactions="false"
confidence = "0"/>

    <PoseCompatibleMotifScorer name="longrange_by_residue"
first_selector="chain_B" sequence_gap = "20"
normalize_by_interactions="false" confidence = "0"/>

    <PoseCompatibleMotifScorer name="partner_by_residue"
first_selector="chain_A" second_selector="chain_B"
normalize_by_interactions="false" confidence = "0"/>

    <CombinedValue name="weighted_motif_by_residue" >

        <Add filter_name="local_by_residue"/>

        <Add filter_name="longrange_by_residue" factor = "10" />

        <Add filter_name="partner_by_residue" factor = "10" />

    </CombinedValue>

```



```

    <PoseCompatibleMotifScorer name="local"
first_selector="chain_B" normalize_by_interactions="false"
normalize_by_residues="false" confidence = "0"/>

    <PoseCompatibleMotifScorer name="longrange"
first_selector="chain_B" sequence_gap = "20"
normalize_by_interactions="false"
normalize_by_residues="false" confidence = "0"/>

    <PoseCompatibleMotifScorer name="partner"
first_selector="chain_A" second_selector="chain_B"
normalize_by_interactions="false"
normalize_by_residues="false" confidence = "0"/>

    <CombinedValue name="weighted_motif" >
        <Add filter_name="local"/>
        <Add filter_name="longrange" factor = "10" />
        <Add filter_name="partner" factor = "10" />
    </CombinedValue>

    <ResidueCount name="size" confidence="0"/>

    <ResidueCount name="size_of_A" confidence="0"
residue_selector="chain_A"/>

```

```

    <ResidueCount name="size_of_B" confidence="0"
residue_selector="chain_B"/>

    <ScoreType name="sfxn_filter" score_type="total_score"
scorefxn="common_sfxn" threshold="999999"/>

    <CalculatorFilter name="normalized_score"
equation="total_score / res" confidence="0">

        <VAR name="total_score" filter="sfxn_filter"/>

        <VAR name="res" filter="size"/>

    </CalculatorFilter>

</FILTERS>

<MOVERS>

    <AppendAssemblyMover name="aam"

model_file_name="/nas/longleaf/home/frankdt/scratch/multistage
_SEWING/relax_tests/assembly_sewing/inputs/smotifs_H_5_40_L_2_
6_H_5_40.segments" hashed="false" max_segments="9"
minimum_cycles="10000" maximum_cycles="10000"
recover_lowest_assembly = "true" partner_pdb =
"vav1_partner.pdb" required_resnums="8">

        <AssemblyRequirements>

```

```

        <ClashRequirement clash_radius =
"4" />

        <SizeInSegmentsRequirement
maximum_size="9" minimum_size="9"/>

    </AssemblyRequirements>

    <AssemblyScorers>

        <MotifScorer weight="1"/>

        <InterModelMotifScorer weight="10"/>

        <PartnerMotifScorer weight="10"/>

    </AssemblyScorers>

</AppendAssemblyMover>

    <AddFlankingVirtualResiduesMover name="virt"
N_term_residues="200" C_term_residues="200"/>

<ChimerizeSuperSecondaryStructuralSegmentsIntoFlankingVirtualR
esiduesMover name="chim" window_width="2"
alignment_max_distance="0.5"
decompose_path="/nas/longleaf/home/frankdt/scratch/multistage_
SEWING/decompose_test"
read_segments_from_decompose_file="true"
read_segments_from_pdb_list = "false"

```

```

permissible_segment_ends="H" model_file_name =
"test_motif_file" pdb_file_name = "test_pdbs"/>

<FilterReportAsPoseExtraScoresMover
name="report_weighted_motifscore_by_residue" report_as =
"MS_weighted_by_residue"
filter_name="weighted_motif_by_residue"/>

<FilterReportAsPoseExtraScoresMover
name="report_motifscore_by_residue" report_as =
"MS_local_by_residue" filter_name="local_by_residue"/>

<FilterReportAsPoseExtraScoresMover
name="report_longrange_motifscore_by_residue" report_as =
"MS_longrange_by_residue" filter_name="longrange_by_residue"/>

<FilterReportAsPoseExtraScoresMover
name="report_partner_motifscore_by_residue" report_as =
"MS_partner_by_residue" filter_name="partner_by_residue"/>

<FilterReportAsPoseExtraScoresMover
name="report_weighted_motifscore_by_interaction" report_as =
"MS_weighted_by_interaction"
filter_name="weighted_motif_by_interaction"/>

```

```

    <FilterReportAsPoseExtraScoresMover
name="report_motifscore_by_interaction" report_as =
"MS_local_by_interaction" filter_name="local_by_interaction"/>

    <FilterReportAsPoseExtraScoresMover
name="report_longrange_motifscore_by_interaction" report_as =
"MS_longrange_by_interaction"
filter_name="longrange_by_interaction"/>

    <FilterReportAsPoseExtraScoresMover
name="report_partner_motifscore_by_interaction" report_as =
"MS_partner_by_interaction"
filter_name="partner_by_interaction"/>

    <FilterReportAsPoseExtraScoresMover
name="report_weighted_motifscore" report_as = "MS_weighted"
filter_name="weighted_motif"/>

    <FilterReportAsPoseExtraScoresMover
name="report_motifscore" report_as = "MS_local"
filter_name="local"/>

    <FilterReportAsPoseExtraScoresMover
name="report_longrange_motifscore" report_as = "MS_longrange"
filter_name="longrange"/>

```

```

    <FilterReportAsPoseExtraScoresMover
name="report_partner_motifscore" report_as = "MS_partner"
filter_name="partner"/>

    <FilterReportAsPoseExtraScoresMover name="report_size"
report_as = "size" filter_name="size"/>

    <FilterReportAsPoseExtraScoresMover
name="report_size_of_A" report_as = "size_of_A"
filter_name="size_of_A"/>

    <FilterReportAsPoseExtraScoresMover
name="report_size_of_B" report_as = "size_of_B"
filter_name="size_of_B"/>

    <FilterReportAsPoseExtraScoresMover
name="report_normalized_score_of_complex" report_as =
"normalized_score_of_complex" filter_name="normalized_score"/>

    <FilterReportAsPoseExtraScoresMover
name="report_normalized_score_of_design" report_as =
"normalized_score_of_design" filter_name="normalized_score"/>

    <InterfaceAnalyzerMover name="interface_analyze" scorefxn
= "common_sfxn" pack_separated="false" pack_input = "false"
jump="1"/>

```

```

    <FastRelax disable_design="false" name="relax_1"
repeats="5" scorefxn="common_sfxn"
task_operations="ifc,ic,exlex2,repack_partner,do_not_repack_di
stant_partner" movemap_factory = "partner_sewing_movemap"
relaxscript="rosettacon2018"/>

```

```

    <DeleteRegionMover name="delete_partner"
residue_selector="chain_A"/>

```

```

</MOVERS>

```

```

<PROTOCOLS>

```

```

    <Add mover_name="aam"/>

```

```

    <Add

```

```

mover_name="report_weighted_motifscore_by_residue"/>

```

```

    <Add

```

```

mover_name="report_longrange_motifscore_by_residue"/>

```

```

    <Add mover_name="report_partner_motifscore_by_residue"/>

```

```

    <Add

```

```

mover_name="report_weighted_motifscore_by_residue"/>

```

```

    <Add

```

```

mover_name="report_weighted_motifscore_by_interaction"/>

```

```

    <Add
mover_name="report_longrange_motifscore_by_interaction"/>

    <Add
mover_name="report_partner_motifscore_by_interaction"/>

    <Add
mover_name="report_weighted_motifscore_by_interaction"/>

    <Add mover_name="report_weighted_motifscore"/>

    <Add mover_name="report_longrange_motifscore"/>

    <Add mover_name="report_partner_motifscore"/>

    <Add mover_name="report_weighted_motifscore"/>

    <Add mover_name="relax_1"/>

    <Add mover_name="interface_analyze"/>

    <Add mover_name="report_size"/>

    <Add mover_name="report_normalized_score_of_complex"/>

    <Add mover_name="delete_partner"/>

    <Add mover_name="report_normalized_score_of_design"/>

    <Add filter_name ="sfxn_filter"/>

</PROTOCOLS>

</ROSETTASCRIPTS>

```



Figure 5.18: The RosettaScript for running SewAnything to generate a vav1 affinity reagent in the 1-to-1 case

```
<JobDefinitionFile>

<Job>

  <Input>

    <PDB filename="vav1_start_node_and_partner.pdb" /> EDIT
THIS (i.e. scaffold.pdb)

  </Input>

</Job>

<Common>

  <SCOREFXNS>

    <ScoreFunction name="common_sfxn"
weights="ref2015.wts"/>

  </SCOREFXNS>

  <RESIDUE_SELECTORS>

    <Chain name="chain_A" chains="1"/>

    <Chain name="chain_B" chains="2"/>

    <ResidueName name="critical_trp" residue_name3="TYR"/>
```

```

    <SecondaryStructure name="loops" overlap="0" minH="1"
minE="1" include_terminal_loops="false" ss="L" />

    <VirtualResidueSelector name="virtuals" />

    <Not name = "real" selector="virtuals"/>

    <And name = "real_loops" selectors="real,loops"/>

    <Neighborhood name="near_B" selector="chain_B" distance =
"6.0"/>

    <Not name = "not_near_B" selector="near_B"/>

    <And name = "A_distant_B" selectors="not_near_B,chain_A"/>

    <And name = "A_near_B" selectors="near_B,chain_A"/>

    <Or name = "A_and_A_near_B" selectors="A_near_B,chain_A"/>

</RESIDUE_SELECTORS>

<MOVE_MAP_FACTORIES>

    <MoveMapFactory name="partner_sewing_movemap" bb="0"
chi="0">

        <Backbone residue_selector="chain_B"/>

        <Chi residue_selector="A_and_A_near_B"/>

    </MoveMapFactory>

</MOVE_MAP_FACTORIES>

```

```

<TASKOPERATIONS> 'to' is a suffix for all task-ops | repack
interface, minimize all atoms

  <InitializeFromCommandline name="ifc"/>

    <IncludeCurrent name="ic"/>

      <ExtraRotamersGeneric ex1="true" ex2="false"
name="exlex2"/>

        <OperateOnResidueSubset name="repack_partner" selector =
"chain_A">

          <RestrictToRepackingRLT/>

        </OperateOnResidueSubset>

        <OperateOnResidueSubset
name="do_not_repack_distant_partner" selector = "A_distant_B">

          <PreventRepackingRLT/>

        </OperateOnResidueSubset>

      </TASKOPERATIONS>

<FILTERS>

  <PoseCompatibleMotifScorer name="local_by_interaction"
first_selector="chain_B" normalize_by_interactions="true"
confidence = "0"/>

  <PoseCompatibleMotifScorer
name="longrange_by_interaction" first_selector="chain_B"

```

```

sequence_gap = "20" normalize_by_interactions="true"
confidence = "0"/>

    <PoseCompatibleMotifScorer name="partner_by_interaction"
first_selector="chain_A" second_selector="chain_B"
normalize_by_interactions="true" confidence = "0"/>

    <CombinedValue name="weighted_motif_by_interaction" >
        <Add filter_name="local_by_interaction"/>
        <Add filter_name="longrange_by_interaction" factor =
"10" />
        <Add filter_name="partner_by_interaction" factor = "10"
/>
    </CombinedValue>

    <PoseCompatibleMotifScorer name="local_by_residue"
first_selector="chain_B" normalize_by_interactions="false"
confidence = "0"/>

    <PoseCompatibleMotifScorer name="longrange_by_residue"
first_selector="chain_B" sequence_gap = "20"
normalize_by_interactions="false" confidence = "0"/>

```

```

    <PoseCompatibleMotifScorer name="partner_by_residue"
first_selector="chain_A" second_selector="chain_B"
normalize_by_interactions="false" confidence = "0"/>

    <CombinedValue name="weighted_motif_by_residue" >
        <Add filter_name="local_by_residue"/>
        <Add filter_name="longrange_by_residue" factor = "10" />
        <Add filter_name="partner_by_residue" factor = "10" />
    </CombinedValue>

    <TrueFilter name="true_filter"/>

    <ResidueCount name="size" confidence="0"/>
    <ResidueCount name="size_of_A" confidence="0"
residue_selector="chain_A"/>
    <ResidueCount name="size_of_B" confidence="0"
residue_selector="chain_B"/>
    <ScoreType name="sfxn_filter" score_type="total_score"
scorefxn="common_sfxn" threshold="999999"/>

    <CalculatorFilter name="normalized_score"
equation="total_score / res" confidence="0">

```

```

        <VAR name="total_score" filter="sfxn_filter"/>

        <VAR name="res" filter="size"/>

    </CalculatorFilter>

</FILTERS>

<MOVERS>

    <AddFlankingVirtualResiduesMover name="virt"
N_term_residues="200" C_term_residues="200"
chain_to_modify="2" vital_selector="critical_trp"/>

    <ChimerizeSuperSecondaryStructuralSegmentsIntoFlankingVirtualR
esiduesMover name="chim" window_width="2"
alignment_max_distance="0.5"
decompose_path="/nas/longleaf/home/frankdt/scratch/multistage_
SEWING/decompose_test"
read_segments_from_decompose_file="true"
read_segments_from_pdb_list = "false"
permissible_segment_ends="H" model_file_name =
"test_motif_file" pdb_file_name = "test_pdbs"/>

```

```

    <FilterReportAsPoseExtraScoresMover
name="report_weighted_motifscore_by_residue" report_as =
"MS_weighted_by_residue"
filter_name="weighted_motif_by_residue"/>

    <FilterReportAsPoseExtraScoresMover
name="report_motifscore_by_residue" report_as =
"MS_local_by_residue" filter_name="local_by_residue"/>

    <FilterReportAsPoseExtraScoresMover
name="report_longrange_motifscore_by_residue" report_as =
"MS_longrange_by_residue" filter_name="longrange_by_residue"/>

    <FilterReportAsPoseExtraScoresMover
name="report_partner_motifscore_by_residue" report_as =
"MS_partner_by_residue" filter_name="partner_by_residue"/>


    <FilterReportAsPoseExtraScoresMover
name="report_weighted_motifscore_by_interaction" report_as =
"MS_weighted_by_interaction"
filter_name="weighted_motif_by_interaction"/>

    <FilterReportAsPoseExtraScoresMover
name="report_motifscore_by_interaction" report_as =
"MS_local_by_interaction" filter_name="local_by_interaction"/>

```

```
<FilterReportAsPoseExtraScoresMover  
name="report_longrange_motifscore_by_interaction" report_as =  
"MS_longrange_by_interaction"  
filter_name="longrange_by_interaction"/>
```

```
<FilterReportAsPoseExtraScoresMover  
name="report_partner_motifscore_by_interaction" report_as =  
"MS_partner_by_interaction"  
filter_name="partner_by_interaction"/>
```

```
<FilterReportAsPoseExtraScoresMover name="report_size"  
report_as = "size" filter_name="size"/>
```

```
<FilterReportAsPoseExtraScoresMover  
name="report_size_of_A" report_as = "size_of_A"  
filter_name="size_of_A"/>
```

```
<FilterReportAsPoseExtraScoresMover  
name="report_size_of_B" report_as = "size_of_B"  
filter_name="size_of_B"/>
```

```
<FilterReportAsPoseExtraScoresMover  
name="report_normalized_score_of_complex" report_as =  
"normalized_score_of_complex" filter_name="normalized_score"/>
```



```

    <FilterReportAsPoseExtraScoresMover
name="report_normalized_score_of_design" report_as =
"normalized_score_of_design" filter_name="normalized_score"/>

    <Small name = "small_mover" residue_selector="real_loops"
nmoves = "10" angle_max="12"/>

    <GenericSimulatedAnnealer name="small_move_gsa"
mover_name="small_mover"
filter_name="weighted_motif_by_residue" trials="10"
sample_type="low" temperature="0.1"/>

    <Shear name = "shear_mover" residue_selector="real_loops"
nmoves = "10" angle_max="12"/>

    <GenericSimulatedAnnealer name="shear_move_gsa"
mover_name="shear_mover"
filter_name="weighted_motif_by_residue" trials="10"
sample_type="low" temperature="0.1"/>

    <DeleteRegionMover name="unvirt"
residue_selector="virtuals"/>

    <DeleteRegionMover name="delete_partner"
residue_selector="chain_A"/>

```

```
<InterfaceAnalyzerMover name="interface_analyze" scorefxn
= "common_sfxn" pack_separated="false" pack_input = "false"
jump="1"/>
```

```
<FastRelax disable_design="false" name="relax_1"
repeats="5" scorefxn="common_sfxn"
task_operations="ifc,ic,exlex2,repack_partner,do_not_repack_di
stant_partner" movemap_factory="partner_sewing_movemap"
relaxscript="rosettacon2018"/>
```

```
<SwitchResidueTypeSetMover name="to_fa"
set="fa_standard"/>
```

```
</MOVERS>
```

```
<PROTOCOLS>
```

```
<Stage num_runs_per_input_struct="1"
total_num_results_to_keep="1">
```

```
<Add mover_name="virt"/>
```

```
Sort filter="weighted_motif"/>
```

```
<Sort filter="true_filter"/>
```

```
</Stage>
```

```
<Stage num_runs_per_input_struct="1000"
total_num_results_to_keep="1000">
```

```

        <Add mover_name="chim"/>

    <Sort filter="true_filter"/>

</Stage>

    <Stage num_runs_per_input_struct="1"
total_num_results_to_keep="1000">

        <Add mover_name="chim"/>

    <Sort filter="true_filter"/>

</Stage>

    <Stage num_runs_per_input_struct="1"
total_num_results_to_keep="1000">

        <Add mover_name="chim"/>

    <Sort filter="true_filter"/>

</Stage>

    <Stage num_runs_per_input_struct="1"
total_num_results_to_keep="1000">

        <Add mover_name="unvirt"/>

        <Add
mover_name="report_weighted_motifscore_by_residue"/>

        <Add
mover_name="report_longrange_motifscore_by_residue"/>

        <Add
mover_name="report_partner_motifscore_by_residue"/>

```

```

        <Add
mover_name="report_weighted_motifscore_by_residue"/>

        <Add
mover_name="report_weighted_motifscore_by_interaction"/>

        <Add
mover_name="report_longrange_motifscore_by_interaction"/>

        <Add
mover_name="report_partner_motifscore_by_interaction"/>

        <Add
mover_name="report_weighted_motifscore_by_interaction"/>

        <Add mover_name="relax_1"/>

        <Add mover_name="interface_analyze"/>

        <Add mover_name="report_size"/>

        <Add mover_name="report_normalized_score_of_complex"/>

        <Add mover_name="delete_partner"/>

        <Add mover_name="report_normalized_score_of_design"/>

        <Sort filter="true_filter"/>

    </Stage>

</PROTOCOLS>

</Common>

</JobDefinitionFile>

```

Figure 5.19: The RosettaScript for running SewAnything to generate a vav1 affinity reagent in the 2-to-1 case

```
<JobDefinitionFile>

<Job>

  <Input>

    <PDB filename="vav1_start_node_and_partner.pdb" /> EDIT
THIS (i.e. scaffold.pdb)

  </Input>

</Job>

<Common>

  <SCOREFXNS>

    <ScoreFunction name="common_sfxn"
weights="ref2015.wts"/>

  </SCOREFXNS>

  <RESIDUE_SELECTORS>

    <Chain name="chain_A" chains="1"/>

    <Chain name="chain_B" chains="2"/>

    <ResidueName name="critical_trp" residue_name3="TYR"/>
```

```

    <SecondaryStructure name="loops" overlap="0" minH="1"
minE="1" include_terminal_loops="false" ss="L" />

    <VirtualResidueSelector name="virtuals" />

    <Not name = "real" selector="virtuals"/>

    <And name = "real_loops" selectors="real,loops"/>

    <Neighborhood name="near_B" selector="chain_B" distance =
"6.0"/>

    <Not name = "not_near_B" selector="near_B"/>

    <And name = "A_distant_B" selectors="not_near_B,chain_A"/>

    <And name = "A_near_B" selectors="near_B,chain_A"/>

    <Or name = "A_and_A_near_B" selectors="A_near_B,chain_A"/>

</RESIDUE_SELECTORS>

<MOVE_MAP_FACTORIES>

    <MoveMapFactory name="partner_sewing_movemap" bb="0"
chi="0">

        <Backbone residue_selector="chain_B"/>

        <Chi residue_selector="A_and_A_near_B"/>

    </MoveMapFactory>

</MOVE_MAP_FACTORIES>

```

```

<TASKOPERATIONS> 'to' is a suffix for all task-ops | repack
interface, minimize all atoms

  <InitializeFromCommandline name="ifc"/>

    <IncludeCurrent name="ic"/>

      <ExtraRotamersGeneric ex1="true" ex2="false"
name="exlex2"/>

        <OperateOnResidueSubset name="repack_partner" selector =
"chain_A">

          <RestrictToRepackingRLT/>

        </OperateOnResidueSubset>

        <OperateOnResidueSubset
name="do_not_repack_distant_partner" selector = "A_distant_B">

          <PreventRepackingRLT/>

        </OperateOnResidueSubset>

      </TASKOPERATIONS>

<FILTERS>

  <PoseCompatibleMotifScorer name="local_by_interaction"
first_selector="chain_B" normalize_by_interactions="true"
confidence = "0"/>

  <PoseCompatibleMotifScorer
name="longrange_by_interaction" first_selector="chain_B"

```

```

sequence_gap = "20" normalize_by_interactions="true"
confidence = "0"/>

    <PoseCompatibleMotifScorer name="partner_by_interaction"
first_selector="chain_A" second_selector="chain_B"
normalize_by_interactions="true" confidence = "0"/>

    <CombinedValue name="weighted_motif_by_interaction" >
        <Add filter_name="local_by_interaction"/>
        <Add filter_name="longrange_by_interaction" factor =
"10" />
        <Add filter_name="partner_by_interaction" factor = "10"
/>
    </CombinedValue>

    <PoseCompatibleMotifScorer name="local_by_residue"
first_selector="chain_B" normalize_by_interactions="false"
confidence = "0"/>

    <PoseCompatibleMotifScorer name="longrange_by_residue"
first_selector="chain_B" sequence_gap = "20"
normalize_by_interactions="false" confidence = "0"/>

```



```

    <PoseCompatibleMotifScorer name="partner_by_residue"
first_selector="chain_A" second_selector="chain_B"
normalize_by_interactions="false" confidence = "0"/>

    <CombinedValue name="weighted_motif_by_residue" >
        <Add filter_name="local_by_residue"/>
        <Add filter_name="longrange_by_residue" factor = "10" />
        <Add filter_name="partner_by_residue" factor = "10" />
    </CombinedValue>

    <TrueFilter name="true_filter"/>

    <ResidueCount name="size" confidence="0"/>
    <ResidueCount name="size_of_A" confidence="0"
residue_selector="chain_A"/>
    <ResidueCount name="size_of_B" confidence="0"
residue_selector="chain_B"/>
    <ScoreType name="sfxn_filter" score_type="total_score"
scorefxn="common_sfxn" threshold="999999"/>

    <CalculatorFilter name="normalized_score"
equation="total_score / res" confidence="0">

```

```

        <VAR name="total_score" filter="sfxn_filter"/>

        <VAR name="res" filter="size"/>

    </CalculatorFilter>

</FILTERS>

<MOVERS>

    <AddFlankingVirtualResiduesMover name="virt"
N_term_residues="200" C_term_residues="200"
chain_to_modify="2" vital_selector="critical_trp"/>

    <ChimerizeSuperSecondaryStructuralSegmentsIntoFlankingVirtualR
esiduesMover name="chim" window_width="2"
alignment_max_distance="0.5"
decompose_path="/nas/longleaf/home/frankdt/scratch/multistage_
SEWING/decompose_test"
read_segments_from_decompose_file="true"
read_segments_from_pdb_list = "false"
permissible_segment_ends="H" model_file_name =
"test_motif_file" pdb_file_name = "test_pdbs"/>

```

```

    <FilterReportAsPoseExtraScoresMover
name="report_weighted_motifscore_by_residue" report_as =
"MS_weighted_by_residue"
filter_name="weighted_motif_by_residue"/>

    <FilterReportAsPoseExtraScoresMover
name="report_motifscore_by_residue" report_as =
"MS_local_by_residue" filter_name="local_by_residue"/>

    <FilterReportAsPoseExtraScoresMover
name="report_longrange_motifscore_by_residue" report_as =
"MS_longrange_by_residue" filter_name="longrange_by_residue"/>

    <FilterReportAsPoseExtraScoresMover
name="report_partner_motifscore_by_residue" report_as =
"MS_partner_by_residue" filter_name="partner_by_residue"/>


    <FilterReportAsPoseExtraScoresMover
name="report_weighted_motifscore_by_interaction" report_as =
"MS_weighted_by_interaction"
filter_name="weighted_motif_by_interaction"/>

    <FilterReportAsPoseExtraScoresMover
name="report_motifscore_by_interaction" report_as =
"MS_local_by_interaction" filter_name="local_by_interaction"/>

```

```

    <FilterReportAsPoseExtraScoresMover
name="report_longrange_motifscore_by_interaction" report_as =
"MS_longrange_by_interaction"
filter_name="longrange_by_interaction"/>

    <FilterReportAsPoseExtraScoresMover
name="report_partner_motifscore_by_interaction" report_as =
"MS_partner_by_interaction"
filter_name="partner_by_interaction"/>


    <FilterReportAsPoseExtraScoresMover name="report_size"
report_as = "size" filter_name="size"/>

    <FilterReportAsPoseExtraScoresMover
name="report_size_of_A" report_as = "size_of_A"
filter_name="size_of_A"/>

    <FilterReportAsPoseExtraScoresMover
name="report_size_of_B" report_as = "size_of_B"
filter_name="size_of_B"/>


    <FilterReportAsPoseExtraScoresMover
name="report_normalized_score_of_complex" report_as =
"normalized_score_of_complex" filter_name="normalized_score"/>

```

```

    <FilterReportAsPoseExtraScoresMover
name="report_normalized_score_of_design" report_as =
"normalized_score_of_design" filter_name="normalized_score"/>

    <Small name = "small_mover" residue_selector="real_loops"
nmoves = "10" angle_max="12"/>

    <GenericSimulatedAnnealer name="small_move_gsa"
mover_name="small_mover"
filter_name="weighted_motif_by_residue" trials="10"
sample_type="low" temperature="0.1"/>

    <Shear name = "shear_mover" residue_selector="real_loops"
nmoves = "10" angle_max="12"/>

    <GenericSimulatedAnnealer name="shear_move_gsa"
mover_name="shear_mover"
filter_name="weighted_motif_by_residue" trials="10"
sample_type="low" temperature="0.1"/>

    <DeleteRegionMover name="unvirt"
residue_selector="virtuals"/>

    <DeleteRegionMover name="delete_partner"
residue_selector="chain_A"/>

```

```

    <InterfaceAnalyzerMover name="interface_analyze" scorefxn
= "common_sfxn" pack_separated="false" pack_input = "false"
jump="1"/>

    <FastRelax disable_design="false" name="relax_1"
repeats="5" scorefxn="common_sfxn"
task_operations="ifc,ic,exlex2,repack_partner,do_not_repack_di
stant_partner" movemap_factory="partner_sewing_movemap"
relaxscript="rosettacon2018"/>

    <SwitchResidueTypeSetMover name="to_fa"
set="fa_standard"/>

</MOVERS>

<PROTOCOLS>

    <Stage num_runs_per_input_struct="1"
total_num_results_to_keep="1">

        <Add mover_name="virt"/>

        Sort filter="weighted_motif"/>

        <Sort filter="true_filter"/>

    </Stage>

    <Stage num_runs_per_input_struct="1000"
total_num_results_to_keep="500">

```

```

        <Add mover_name="chim"/>

    <Sort filter="weighted_motif_by_residue"/>

</Stage>

<Stage num_runs_per_input_struct="2"
total_num_results_to_keep="500">

    <Add mover_name="chim"/>

    <Sort filter="weighted_motif_by_residue"/>

</Stage>

<Stage num_runs_per_input_struct="2"
total_num_results_to_keep="500">

    <Add mover_name="chim"/>

    <Sort filter="weighted_motif_by_residue"/>

</Stage>

<Stage num_runs_per_input_struct="2"
total_num_results_to_keep="1000">

    <Add mover_name="chim"/>

    <Sort filter="true_filter"/>

</Stage>

<Stage num_runs_per_input_struct="1"
total_num_results_to_keep="1000">

    <Add mover_name="unvirt"/>

```

```

        <Add
mover_name="report_weighted_motifscore_by_residue"/>

        <Add
mover_name="report_longrange_motifscore_by_residue"/>

        <Add
mover_name="report_partner_motifscore_by_residue"/>

        <Add
mover_name="report_weighted_motifscore_by_residue"/>

        <Add
mover_name="report_weighted_motifscore_by_interaction"/>

        <Add
mover_name="report_longrange_motifscore_by_interaction"/>

        <Add
mover_name="report_partner_motifscore_by_interaction"/>

        <Add
mover_name="report_weighted_motifscore_by_interaction"/>

        <Add mover_name="relax_1"/>

        <Add mover_name="interface_analyze"/>

        <Add mover_name="report_size"/>

        <Add mover_name="report_normalized_score_of_complex"/>

        <Add mover_name="delete_partner"/>

        <Add mover_name="report_normalized_score_of_design"/>

```



```
<Sort filter="true_filter"/>
</Stage>
</PROTOCOLS>
</Common>
</JobDefinitionFile>
```

Figure 5.20: The RosettaScript for running SewAnything to generate a vav1 affinity reagent in the 4-to-1 case

```
<JobDefinitionFile>

<Job>

  <Input>

    <PDB filename="vav1_start_node_and_partner.pdb" /> EDIT
THIS (i.e. scaffold.pdb)

  </Input>

</Job>

<Common>

  <SCOREFXNS>

    <ScoreFunction name="common_sfxn"
weights="ref2015.wts"/>

  </SCOREFXNS>

  <RESIDUE_SELECTORS>

    <Chain name="chain_A" chains="1"/>

    <Chain name="chain_B" chains="2"/>

    <ResidueName name="critical_trp" residue_name3="TYR"/>
```

```

    <SecondaryStructure name="loops" overlap="0" minH="1"
minE="1" include_terminal_loops="false" ss="L" />

    <VirtualResidueSelector name="virtuals" />

    <Not name = "real" selector="virtuals"/>

    <And name = "real_loops" selectors="real,loops"/>

    <Neighborhood name="near_B" selector="chain_B" distance =
"6.0"/>

    <Not name = "not_near_B" selector="near_B"/>

    <And name = "A_distant_B" selectors="not_near_B,chain_A"/>

    <And name = "A_near_B" selectors="near_B,chain_A"/>

    <Or name = "A_and_A_near_B" selectors="A_near_B,chain_A"/>

</RESIDUE_SELECTORS>

<MOVE_MAP_FACTORIES>

    <MoveMapFactory name="partner_sewing_movemap" bb="0"
chi="0">

        <Backbone residue_selector="chain_B"/>

        <Chi residue_selector="A_and_A_near_B"/>

    </MoveMapFactory>

</MOVE_MAP_FACTORIES>

```

```

<TASKOPERATIONS> 'to' is a suffix for all task-ops | repack
interface, minimize all atoms

  <InitializeFromCommandline name="ifc"/>

    <IncludeCurrent name="ic"/>

      <ExtraRotamersGeneric ex1="true" ex2="false"
name="exlex2"/>

        <OperateOnResidueSubset name="repack_partner" selector =
"chain_A">

          <RestrictToRepackingRLT/>

        </OperateOnResidueSubset>

        <OperateOnResidueSubset
name="do_not_repack_distant_partner" selector = "A_distant_B">

          <PreventRepackingRLT/>

        </OperateOnResidueSubset>

      </TASKOPERATIONS>

<FILTERS>

  <PoseCompatibleMotifScorer name="local_by_interaction"
first_selector="chain_B" normalize_by_interactions="true"
confidence = "0"/>

  <PoseCompatibleMotifScorer
name="longrange_by_interaction" first_selector="chain_B"

```

```

sequence_gap = "20" normalize_by_interactions="true"
confidence = "0"/>

    <PoseCompatibleMotifScorer name="partner_by_interaction"
first_selector="chain_A" second_selector="chain_B"
normalize_by_interactions="true" confidence = "0"/>

    <CombinedValue name="weighted_motif_by_interaction" >
        <Add filter_name="local_by_interaction"/>
        <Add filter_name="longrange_by_interaction" factor =
"10" />
        <Add filter_name="partner_by_interaction" factor = "10"
/>
    </CombinedValue>

    <PoseCompatibleMotifScorer name="local_by_residue"
first_selector="chain_B" normalize_by_interactions="false"
confidence = "0"/>

    <PoseCompatibleMotifScorer name="longrange_by_residue"
first_selector="chain_B" sequence_gap = "20"
normalize_by_interactions="false" confidence = "0"/>

```

```

    <PoseCompatibleMotifScorer name="partner_by_residue"
first_selector="chain_A" second_selector="chain_B"
normalize_by_interactions="false" confidence = "0"/>

    <CombinedValue name="weighted_motif_by_residue" >
        <Add filter_name="local_by_residue"/>
        <Add filter_name="longrange_by_residue" factor = "10" />
        <Add filter_name="partner_by_residue" factor = "10" />
    </CombinedValue>

    <TrueFilter name="true_filter"/>

    <ResidueCount name="size" confidence="0"/>
    <ResidueCount name="size_of_A" confidence="0"
residue_selector="chain_A"/>
    <ResidueCount name="size_of_B" confidence="0"
residue_selector="chain_B"/>
    <ScoreType name="sfxn_filter" score_type="total_score"
scorefxn="common_sfxn" threshold="999999"/>

    <CalculatorFilter name="normalized_score"
equation="total_score / res" confidence="0">

```

```

        <VAR name="total_score" filter="sfxn_filter"/>

        <VAR name="res" filter="size"/>

    </CalculatorFilter>

</FILTERS>

<MOVERS>

    <AddFlankingVirtualResiduesMover name="virt"
N_term_residues="200" C_term_residues="200"
chain_to_modify="2" vital_selector="critical_trp"/>

    <ChimerizeSuperSecondaryStructuralSegmentsIntoFlankingVirtualR
esiduesMover name="chim" window_width="2"
alignment_max_distance="0.5"
decompose_path="/nas/longleaf/home/frankdt/scratch/multistage_
SEWING/decompose_test"
read_segments_from_decompose_file="true"
read_segments_from_pdb_list = "false"
permissible_segment_ends="H" model_file_name =
"test_motif_file" pdb_file_name = "test_pdbs"/>

```

```

    <FilterReportAsPoseExtraScoresMover
name="report_weighted_motifscore_by_residue" report_as =
"MS_weighted_by_residue"
filter_name="weighted_motif_by_residue"/>

    <FilterReportAsPoseExtraScoresMover
name="report_motifscore_by_residue" report_as =
"MS_local_by_residue" filter_name="local_by_residue"/>

    <FilterReportAsPoseExtraScoresMover
name="report_longrange_motifscore_by_residue" report_as =
"MS_longrange_by_residue" filter_name="longrange_by_residue"/>

    <FilterReportAsPoseExtraScoresMover
name="report_partner_motifscore_by_residue" report_as =
"MS_partner_by_residue" filter_name="partner_by_residue"/>


    <FilterReportAsPoseExtraScoresMover
name="report_weighted_motifscore_by_interaction" report_as =
"MS_weighted_by_interaction"
filter_name="weighted_motif_by_interaction"/>

    <FilterReportAsPoseExtraScoresMover
name="report_motifscore_by_interaction" report_as =
"MS_local_by_interaction" filter_name="local_by_interaction"/>

```



```
<FilterReportAsPoseExtraScoresMover
name="report_longrange_motifscore_by_interaction" report_as =
"MS_longrange_by_interaction"
filter_name="longrange_by_interaction"/>
```

```
<FilterReportAsPoseExtraScoresMover
name="report_partner_motifscore_by_interaction" report_as =
"MS_partner_by_interaction"
filter_name="partner_by_interaction"/>
```

```
<FilterReportAsPoseExtraScoresMover name="report_size"
report_as = "size" filter_name="size"/>
```

```
<FilterReportAsPoseExtraScoresMover
name="report_size_of_A" report_as = "size_of_A"
filter_name="size_of_A"/>
```

```
<FilterReportAsPoseExtraScoresMover
name="report_size_of_B" report_as = "size_of_B"
filter_name="size_of_B"/>
```

```
<FilterReportAsPoseExtraScoresMover
name="report_normalized_score_of_complex" report_as =
"normalized_score_of_complex" filter_name="normalized_score"/>
```

```

    <FilterReportAsPoseExtraScoresMover
name="report_normalized_score_of_design" report_as =
"normalized_score_of_design" filter_name="normalized_score"/>

    <Small name = "small_mover" residue_selector="real_loops"
nmoves = "10" angle_max="12"/>

    <GenericSimulatedAnnealer name="small_move_gsa"
mover_name="small_mover"
filter_name="weighted_motif_by_residue" trials="10"
sample_type="low" temperature="0.1"/>

    <Shear name = "shear_mover" residue_selector="real_loops"
nmoves = "10" angle_max="12"/>

    <GenericSimulatedAnnealer name="shear_move_gsa"
mover_name="shear_mover"
filter_name="weighted_motif_by_residue" trials="10"
sample_type="low" temperature="0.1"/>

    <DeleteRegionMover name="unvirt"
residue_selector="virtuals"/>

    <DeleteRegionMover name="delete_partner"
residue_selector="chain_A"/>

```

```

    <InterfaceAnalyzerMover name="interface_analyze" scorefxn
= "common_sfxn" pack_separated="false" pack_input = "false"
jump="1"/>

    <FastRelax disable_design="false" name="relax_1"
repeats="5" scorefxn="common_sfxn"
task_operations="ifc,ic,exlex2,repack_partner,do_not_repack_di
stant_partner" movemap_factory="partner_sewing_movemap"
relaxscript="rosettacon2018"/>

    <SwitchResidueTypeSetMover name="to_fa"
set="fa_standard"/>

</MOVERS>

<PROTOCOLS>

    <Stage num_runs_per_input_struct="1"
total_num_results_to_keep="1">

        <Add mover_name="virt"/>

        <Sort filter="weighted_motif"/>

        <Sort filter="true_filter"/>

    </Stage>

    <Stage num_runs_per_input_struct="1000"
total_num_results_to_keep="250">

```

```

        <Add mover_name="chim"/>

    <Sort filter="weighted_motif_by_residue"/>

</Stage>

    <Stage num_runs_per_input_struct="4"
total_num_results_to_keep="250">

        <Add mover_name="chim"/>

    <Sort filter="weighted_motif_by_residue"/>

</Stage>

    <Stage num_runs_per_input_struct="4"
total_num_results_to_keep="250">

        <Add mover_name="chim"/>

    <Sort filter="weighted_motif_by_residue"/>

</Stage>

    <Stage num_runs_per_input_struct="4"
total_num_results_to_keep="1000">

        <Add mover_name="chim"/>

    <Sort filter="true_filter"/>

</Stage>

    <Stage num_runs_per_input_struct="1"
total_num_results_to_keep="1000">

        <Add mover_name="unvirt"/>

```

```

        <Add
mover_name="report_weighted_motifscore_by_residue"/>
        <Add
mover_name="report_longrange_motifscore_by_residue"/>
        <Add
mover_name="report_partner_motifscore_by_residue"/>
        <Add
mover_name="report_weighted_motifscore_by_residue"/>
        <Add
mover_name="report_weighted_motifscore_by_interaction"/>
        <Add
mover_name="report_longrange_motifscore_by_interaction"/>
        <Add
mover_name="report_partner_motifscore_by_interaction"/>
        <Add
mover_name="report_weighted_motifscore_by_interaction"/>
        <Add mover_name="relax_1"/>
        <Add mover_name="interface_analyze"/>
        <Add mover_name="report_size"/>
        <Add mover_name="report_normalized_score_of_complex"/>
        <Add mover_name="delete_partner"/>
        <Add mover_name="report_normalized_score_of_design"/>

```

```
        <Sort filter="true_filter"/>
    </Stage>
</PROTOCOLS>
</Common>
</JobDefinitionFile>
```

Figure 5.21: The RosettaScript for running SewAnything to generate a vav1 affinity reagent in the 10-to-1 case

```
<JobDefinitionFile>

<Job>

  <Input>

    <PDB filename="vav1_start_node_and_partner.pdb" /> EDIT
THIS (i.e. scaffold.pdb)

  </Input>

</Job>

<Common>

  <SCOREFXNS>

    <ScoreFunction name="common_sfxn"
weights="ref2015.wts"/>

  </SCOREFXNS>

  <RESIDUE_SELECTORS>

    <Chain name="chain_A" chains="1"/>

    <Chain name="chain_B" chains="2"/>

    <ResidueName name="critical_trp" residue_name3="TYR"/>
```

```

    <SecondaryStructure name="loops" overlap="0" minH="1"
minE="1" include_terminal_loops="false" ss="L" />

    <VirtualResidueSelector name="virtuals" />

    <Not name = "real" selector="virtuals"/>

    <And name = "real_loops" selectors="real,loops"/>

    <Neighborhood name="near_B" selector="chain_B" distance =
"6.0"/>

    <Not name = "not_near_B" selector="near_B"/>

    <And name = "A_distant_B" selectors="not_near_B,chain_A"/>

    <And name = "A_near_B" selectors="near_B,chain_A"/>

    <Or name = "A_and_A_near_B" selectors="A_near_B,chain_A"/>

</RESIDUE_SELECTORS>

<MOVE_MAP_FACTORIES>

    <MoveMapFactory name="partner_sewing_movemap" bb="0"
chi="0">

        <Backbone residue_selector="chain_B"/>

        <Chi residue_selector="A_and_A_near_B"/>

    </MoveMapFactory>

</MOVE_MAP_FACTORIES>

```



```

<TASKOPERATIONS> 'to' is a suffix for all task-ops | repack
interface, minimize all atoms

  <InitializeFromCommandline name="ifc"/>

    <IncludeCurrent name="ic"/>

      <ExtraRotamersGeneric ex1="true" ex2="false"
name="exlex2"/>

        <OperateOnResidueSubset name="repack_partner" selector =
"chain_A">

          <RestrictToRepackingRLT/>

        </OperateOnResidueSubset>

        <OperateOnResidueSubset
name="do_not_repack_distant_partner" selector = "A_distant_B">

          <PreventRepackingRLT/>

        </OperateOnResidueSubset>

      </TASKOPERATIONS>

<FILTERS>

  <PoseCompatibleMotifScorer name="local_by_interaction"
first_selector="chain_B" normalize_by_interactions="true"
confidence = "0"/>

  <PoseCompatibleMotifScorer
name="longrange_by_interaction" first_selector="chain_B"

```

```

sequence_gap = "20" normalize_by_interactions="true"
confidence = "0"/>

    <PoseCompatibleMotifScorer name="partner_by_interaction"
first_selector="chain_A" second_selector="chain_B"
normalize_by_interactions="true" confidence = "0"/>

    <CombinedValue name="weighted_motif_by_interaction" >
        <Add filter_name="local_by_interaction"/>
        <Add filter_name="longrange_by_interaction" factor =
"10" />
        <Add filter_name="partner_by_interaction" factor = "10"
/>
    </CombinedValue>

    <PoseCompatibleMotifScorer name="local_by_residue"
first_selector="chain_B" normalize_by_interactions="false"
confidence = "0"/>

    <PoseCompatibleMotifScorer name="longrange_by_residue"
first_selector="chain_B" sequence_gap = "20"
normalize_by_interactions="false" confidence = "0"/>

```

```

    <PoseCompatibleMotifScorer name="partner_by_residue"
first_selector="chain_A" second_selector="chain_B"
normalize_by_interactions="false" confidence = "0"/>

    <CombinedValue name="weighted_motif_by_residue" >
        <Add filter_name="local_by_residue"/>
        <Add filter_name="longrange_by_residue" factor = "10" />
        <Add filter_name="partner_by_residue" factor = "10" />
    </CombinedValue>

    <TrueFilter name="true_filter"/>

    <ResidueCount name="size" confidence="0"/>
    <ResidueCount name="size_of_A" confidence="0"
residue_selector="chain_A"/>
    <ResidueCount name="size_of_B" confidence="0"
residue_selector="chain_B"/>
    <ScoreType name="sfxn_filter" score_type="total_score"
scorefxn="common_sfxn" threshold="999999"/>

    <CalculatorFilter name="normalized_score"
equation="total_score / res" confidence="0">

```

```

        <VAR name="total_score" filter="sfxn_filter"/>

        <VAR name="res" filter="size"/>

    </CalculatorFilter>

</FILTERS>

<MOVERS>

    <AddFlankingVirtualResiduesMover name="virt"
N_term_residues="200" C_term_residues="200"
chain_to_modify="2" vital_selector="critical_trp"/>

    <ChimerizeSuperSecondaryStructuralSegmentsIntoFlankingVirtualR
esiduesMover name="chim" window_width="2"
alignment_max_distance="0.5"
decompose_path="/nas/longleaf/home/frankdt/scratch/multistage_
SEWING/decompose_test"
read_segments_from_decompose_file="true"
read_segments_from_pdb_list = "false"
permissible_segment_ends="H" model_file_name =
"test_motif_file" pdb_file_name = "test_pdbs"/>

```

```

    <FilterReportAsPoseExtraScoresMover
name="report_weighted_motifscore_by_residue" report_as =
"MS_weighted_by_residue"
filter_name="weighted_motif_by_residue"/>

    <FilterReportAsPoseExtraScoresMover
name="report_motifscore_by_residue" report_as =
"MS_local_by_residue" filter_name="local_by_residue"/>

    <FilterReportAsPoseExtraScoresMover
name="report_longrange_motifscore_by_residue" report_as =
"MS_longrange_by_residue" filter_name="longrange_by_residue"/>

    <FilterReportAsPoseExtraScoresMover
name="report_partner_motifscore_by_residue" report_as =
"MS_partner_by_residue" filter_name="partner_by_residue"/>


    <FilterReportAsPoseExtraScoresMover
name="report_weighted_motifscore_by_interaction" report_as =
"MS_weighted_by_interaction"
filter_name="weighted_motif_by_interaction"/>

    <FilterReportAsPoseExtraScoresMover
name="report_motifscore_by_interaction" report_as =
"MS_local_by_interaction" filter_name="local_by_interaction"/>

```

```
<FilterReportAsPoseExtraScoresMover
name="report_longrange_motifscore_by_interaction" report_as =
"MS_longrange_by_interaction"
filter_name="longrange_by_interaction"/>
```

```
<FilterReportAsPoseExtraScoresMover
name="report_partner_motifscore_by_interaction" report_as =
"MS_partner_by_interaction"
filter_name="partner_by_interaction"/>
```

```
<FilterReportAsPoseExtraScoresMover name="report_size"
report_as = "size" filter_name="size"/>
```

```
<FilterReportAsPoseExtraScoresMover
name="report_size_of_A" report_as = "size_of_A"
filter_name="size_of_A"/>
```

```
<FilterReportAsPoseExtraScoresMover
name="report_size_of_B" report_as = "size_of_B"
filter_name="size_of_B"/>
```

```
<FilterReportAsPoseExtraScoresMover
name="report_normalized_score_of_complex" report_as =
"normalized_score_of_complex" filter_name="normalized_score"/>
```

```

    <FilterReportAsPoseExtraScoresMover
name="report_normalized_score_of_design" report_as =
"normalized_score_of_design" filter_name="normalized_score"/>

    <Small name = "small_mover" residue_selector="real_loops"
nmoves = "10" angle_max="12"/>

    <GenericSimulatedAnnealer name="small_move_gsa"
mover_name="small_mover"
filter_name="weighted_motif_by_residue" trials="10"
sample_type="low" temperature="0.1"/>

    <Shear name = "shear_mover" residue_selector="real_loops"
nmoves = "10" angle_max="12"/>

    <GenericSimulatedAnnealer name="shear_move_gsa"
mover_name="shear_mover"
filter_name="weighted_motif_by_residue" trials="10"
sample_type="low" temperature="0.1"/>

    <DeleteRegionMover name="unvirt"
residue_selector="virtuals"/>

    <DeleteRegionMover name="delete_partner"
residue_selector="chain_A"/>

```

```

    <InterfaceAnalyzerMover name="interface_analyze" scorefxn
= "common_sfxn" pack_separated="false" pack_input = "false"
jump="1"/>

    <FastRelax disable_design="false" name="relax_1"
repeats="5" scorefxn="common_sfxn"
task_operations="ifc,ic,exlex2,repack_partner,do_not_repack_di
stant_partner" movemap_factory="partner_sewing_movemap"
relaxscript="rosettacon2018"/>

    <SwitchResidueTypeSetMover name="to_fa"
set="fa_standard"/>

</MOVERS>

<PROTOCOLS>

    <Stage num_runs_per_input_struct="1"
total_num_results_to_keep="1">

        <Add mover_name="virt"/>

        Sort filter="weighted_motif"/>

        <Sort filter="true_filter"/>

    </Stage>

    <Stage num_runs_per_input_struct="1000"
total_num_results_to_keep="100">

```



```

        <Add mover_name="chim"/>

    <Sort filter="weighted_motif_by_residue"/>

</Stage>

    <Stage num_runs_per_input_struct="10"
total_num_results_to_keep="100">

        <Add mover_name="chim"/>

    <Sort filter="weighted_motif_by_residue"/>

</Stage>

    <Stage num_runs_per_input_struct="10"
total_num_results_to_keep="100">

        <Add mover_name="chim"/>

    <Sort filter="weighted_motif_by_residue"/>

</Stage>

    <Stage num_runs_per_input_struct="10"
total_num_results_to_keep="1000">

        <Add mover_name="chim"/>

    <Sort filter="true_filter"/>

</Stage>

    <Stage num_runs_per_input_struct="1"
total_num_results_to_keep="1000">

        <Add mover_name="unvirt"/>

```

```

        <Add
mover_name="report_weighted_motifscore_by_residue"/>
        <Add
mover_name="report_longrange_motifscore_by_residue"/>
        <Add
mover_name="report_partner_motifscore_by_residue"/>
        <Add
mover_name="report_weighted_motifscore_by_residue"/>
        <Add
mover_name="report_weighted_motifscore_by_interaction"/>
        <Add
mover_name="report_longrange_motifscore_by_interaction"/>
        <Add
mover_name="report_partner_motifscore_by_interaction"/>
        <Add
mover_name="report_weighted_motifscore_by_interaction"/>
        <Add mover_name="relax_1"/>
        <Add mover_name="interface_analyze"/>
        <Add mover_name="report_size"/>
        <Add mover_name="report_normalized_score_of_complex"/>
        <Add mover_name="delete_partner"/>
        <Add mover_name="report_normalized_score_of_design"/>

```

```
<Sort filter="true_filter"/>
</Stage>
</PROTOCOLS>
</Common>
</JobDefinitionFile>
```

Figure 5.22: The RosettaScript for running SewAnything to generate a vav1 affinity reagent with intercalated small and shear moves

```
<JobDefinitionFile>

<Job>

  <Input>

    <PDB filename="vav1_start_node_and_partner.pdb" /> EDIT
THIS (i.e. scaffold.pdb)

  </Input>

</Job>

<Common>

  <SCOREFXNS>

    <ScoreFunction name="common_sfxn"
weights="ref2015.wts"/>

  </SCOREFXNS>

  <RESIDUE_SELECTORS>

    <Chain name="chain_A" chains="1"/>

    <Chain name="chain_B" chains="2"/>

    <ResidueName name="critical_trp" residue_name3="TYR"/>
```

```

    <SecondaryStructure name="loops" overlap="0" minH="1"
minE="1" include_terminal_loops="false" ss="L" />

    <VirtualResidueSelector name="virtuals" />

    <Not name = "real" selector="virtuals"/>

    <And name = "real_loops" selectors="real,loops"/>

    <And name = "real_loops_of_B"
selectors="real,loops,chain_B"/>

    <Neighborhood name="near_B" selector="chain_B" distance =
"6.0"/>

    <Not name = "not_near_B" selector="near_B"/>

    <And name = "A_distant_B" selectors="not_near_B,chain_A"/>

    <And name = "A_near_B" selectors="near_B,chain_A"/>

    <Or name = "A_and_A_near_B" selectors="A_near_B,chain_A"/>

</RESIDUE_SELECTORS>

<MOVE_MAP_FACTORIES>

    <MoveMapFactory name="partner_sewing_movemap" bb="0"
chi="0">

        <Backbone residue_selector="chain_B"/>

        <Chi residue_selector="A_and_A_near_B"/>

    </MoveMapFactory>

</MOVE_MAP_FACTORIES>

```

<TASKOPERATIONS> 'to' is a suffix for all task-ops | repack  
interface, minimize all atoms

```
<InitializeFromCommandline name="ifc"/>
    <IncludeCurrent name="ic"/>
    <ExtraRotamersGeneric ex1="true" ex2="false"
name="exlex2"/>
    <OperateOnResidueSubset name="repack_partner" selector =
"chain_A">
        <RestrictToRepackingRLT/>
    </OperateOnResidueSubset>
    <OperateOnResidueSubset
name="do_not_repack_distant_partner" selector = "A_distant_B">
        <PreventRepackingRLT/>
    </OperateOnResidueSubset>
</TASKOPERATIONS>
```

```
<FILTERS>
    <PoseCompatibleMotifScorer name="local_by_interaction"
first_selector="chain_B" normalize_by_interactions="true"
confidence = "0"/>
```

```

    <PoseCompatibleMotifScorer
name="longrange_by_interaction" first_selector="chain_B"
sequence_gap = "20" normalize_by_interactions="true"
confidence = "0"/>

    <PoseCompatibleMotifScorer name="partner_by_interaction"
first_selector="chain_A" second_selector="chain_B"
normalize_by_interactions="true" confidence = "0"/>

    <CombinedValue name="weighted_motif_by_interaction" >
        <Add filter_name="local_by_interaction"/>
        <Add filter_name="longrange_by_interaction" factor =
"10" />
        <Add filter_name="partner_by_interaction" factor = "10"
/>
    </CombinedValue>

    <PoseCompatibleMotifScorer name="local_by_residue"
first_selector="chain_B" normalize_by_interactions="false"
confidence = "0"/>

    <PoseCompatibleMotifScorer name="longrange_by_residue"
first_selector="chain_B" sequence_gap = "20"
normalize_by_interactions="false" confidence = "0"/>

```

```

    <PoseCompatibleMotifScorer name="partner_by_residue"
first_selector="chain_A" second_selector="chain_B"
normalize_by_interactions="false" confidence = "0"/>

    <CombinedValue name="weighted_motif_by_residue" >

        <Add filter_name="local_by_residue"/>

        <Add filter_name="longrange_by_residue" factor = "10" />

        <Add filter_name="partner_by_residue" factor = "10" />

    </CombinedValue>

    <PoseCompatibleMotifScorer name="local"
first_selector="chain_B" normalize_by_interactions="false"
normalize_by_residues="false" confidence = "0"/>

    <PoseCompatibleMotifScorer name="longrange"
first_selector="chain_B" sequence_gap = "20"
normalize_by_interactions="false"
normalize_by_residues="false" confidence = "0"/>

    <PoseCompatibleMotifScorer name="partner"
first_selector="chain_A" second_selector="chain_B"
normalize_by_interactions="false"
normalize_by_residues="false" confidence = "0"/>

```



```

<CombinedValue name="weighted_motif" >

  <Add filter_name="local"/>

  <Add filter_name="longrange" factor = "10" />

  <Add filter_name="partner" factor = "10" />

</CombinedValue>


<TrueFilter name="true_filter"/>


<ResidueCount name="size" confidence="0"/>

<ResidueCount name="size_of_A" confidence="0"
residue_selector="chain_A"/>

<ResidueCount name="size_of_B" confidence="0"
residue_selector="chain_B"/>

<ScoreType name="sfxn_filter" score_type="total_score"
scorefxn="common_sfxn" threshold="999999"/>


<CalculatorFilter name="normalized_score"
equation="total_score / res" confidence="0">

  <VAR name="total_score" filter="sfxn_filter"/>

  <VAR name="res" filter="size"/>

</CalculatorFilter>

```

</FILTERS>

<MOVERS>

```
<AddFlankingVirtualResiduesMover name="virt"
N_term_residues="200" C_term_residues="200"
chain_to_modify="2" vital_selector="critical_trp"/>

<ChimerizeSuperSecondaryStructuralSegmentsIntoFlankingVirtualR
esiduesMover name="chim" window_width="2"
alignment_max_distance="0.5"
decompose_path="/nas/longleaf/home/frankdt/scratch/multistage_
SEWING/decompose_test"
read_segments_from_decompose_file="true"
read_segments_from_pdb_list = "false"
permissible_segment_ends="H" model_file_name =
"test_motif_file" pdb_file_name = "test_pdbs"
max_attempts="1000"/>

<FilterReportAsPoseExtraScoresMover
name="report_weighted_motifscore_by_residue" report_as =
"MS_weighted_by_residue"
filter_name="weighted_motif_by_residue"/>
```

```

    <FilterReportAsPoseExtraScoresMover
name="report_motifscore_by_residue" report_as =
"MS_local_by_residue" filter_name="local_by_residue"/>

    <FilterReportAsPoseExtraScoresMover
name="report_longrange_motifscore_by_residue" report_as =
"MS_longrange_by_residue" filter_name="longrange_by_residue"/>

    <FilterReportAsPoseExtraScoresMover
name="report_partner_motifscore_by_residue" report_as =
"MS_partner_by_residue" filter_name="partner_by_residue"/>


    <FilterReportAsPoseExtraScoresMover
name="report_weighted_motifscore_by_interaction" report_as =
"MS_weighted_by_interaction"
filter_name="weighted_motif_by_interaction"/>

    <FilterReportAsPoseExtraScoresMover
name="report_motifscore_by_interaction" report_as =
"MS_local_by_interaction" filter_name="local_by_interaction"/>

    <FilterReportAsPoseExtraScoresMover
name="report_longrange_motifscore_by_interaction" report_as =
"MS_longrange_by_interaction"
filter_name="longrange_by_interaction"/>

```

```
<FilterReportAsPoseExtraScoresMover  
name="report_partner_motifscore_by_interaction" report_as =  
"MS_partner_by_interaction"  
filter_name="partner_by_interaction"/>
```

```
<FilterReportAsPoseExtraScoresMover  
name="report_weighted_motifscore" report_as = "MS_weighted"  
filter_name="weighted_motif"/>
```

```
<FilterReportAsPoseExtraScoresMover  
name="report_motifscore" report_as = "MS_local"  
filter_name="local"/>
```

```
<FilterReportAsPoseExtraScoresMover  
name="report_longrange_motifscore" report_as = "MS_longrange"  
filter_name="longrange"/>
```

```
<FilterReportAsPoseExtraScoresMover  
name="report_partner_motifscore" report_as = "MS_partner"  
filter_name="partner"/>
```

```
<FilterReportAsPoseExtraScoresMover name="report_size"  
report_as = "size" filter_name="size"/>
```

```

    <FilterReportAsPoseExtraScoresMover
name="report_size_of_A" report_as = "size_of_A"
filter_name="size_of_A"/>

    <FilterReportAsPoseExtraScoresMover
name="report_size_of_B" report_as = "size_of_B"
filter_name="size_of_B"/>

    <FilterReportAsPoseExtraScoresMover
name="report_normalized_score_of_complex" report_as =
"normalized_score_of_complex" filter_name="normalized_score"/>

    <FilterReportAsPoseExtraScoresMover
name="report_normalized_score_of_design" report_as =
"normalized_score_of_design" filter_name="normalized_score"/>

    <Small name = "small_mover"
residue_selector="real_loops_of_B" nmoves = "1"
angle_max="12"/>

    <GenericSimulatedAnnealer name="small_move_gsa"
mover_name="small_mover"
filter_name="weighted_motif_by_interaction" trials="10"
sample_type="low" temperature="0.1"/>

```

```

    <Shear name = "shear_mover"
residue_selector="real_loops_of_B" nmoves = "1"
angle_max="12"/>

    <GenericSimulatedAnnealer name="shear_move_gsa"
mover_name="shear_mover"
filter_name="weighted_motif_by_interaction" trials="10"
sample_type="low" temperature="0.1"/>

    <DeleteRegionMover name="unvirt"
residue_selector="virtuals"/>

    <DeleteRegionMover name="delete_partner"
residue_selector="chain_A"/>

    <InterfaceAnalyzerMover name="interface_analyze" scorefxn
= "common_sfxn" pack_separated="false" pack_input = "false"
jump="1"/>

    <FastRelax disable_design="false" name="relax_1"
repeats="5" scorefxn="common_sfxn"
task_operations="ifc,ic,exlex2,repack_partner,do_not_repack_distant_partner" movemap_factory="partner_sewing_movemap"
relaxscript="rosettacon2018"/>

    <SwitchResidueTypeSetMover name="to_fa"
set="fa_standard"/>

```

</MOVERS>

<PROTOCOLS>

```
    <Stage num_runs_per_input_struct="1"
total_num_results_to_keep="1">
    <Add mover_name="virt"/>
    Sort filter="weighted_motif"/>
    <Sort filter="true_filter"/>
    </Stage>
    <Stage num_runs_per_input_struct="1000"
total_num_results_to_keep="100">
    <Add mover_name="chim"/>
    <Sort filter="weighted_motif"/>
    </Stage>
    <Stage num_runs_per_input_struct="10"
total_num_results_to_keep="100">
    <Add mover_name="shear_move_gsa"/>
    <Sort filter="weighted_motif"/>
    </Stage>
    <Stage num_runs_per_input_struct="10"
total_num_results_to_keep="100">
```

```

        <Add mover_name="small_move_gsa"/>

    <Sort filter="weighted_motif"/>

</Stage>

    <Stage num_runs_per_input_struct="10"
total_num_results_to_keep="100">

        <Add mover_name="chim"/>

    <Sort filter="weighted_motif"/>

</Stage>

    <Stage num_runs_per_input_struct="10"
total_num_results_to_keep="100">

        <Add mover_name="shear_move_gsa"/>

    <Sort filter="weighted_motif"/>

</Stage>

    <Stage num_runs_per_input_struct="10"
total_num_results_to_keep="100">

        <Add mover_name="small_move_gsa"/>

    <Sort filter="weighted_motif"/>

</Stage>

    <Stage num_runs_per_input_struct="10"
total_num_results_to_keep="100">

        <Add mover_name="chim"/>

    <Sort filter="weighted_motif"/>

```



```

    </Stage>

    <Stage num_runs_per_input_struct="10"
total_num_results_to_keep="100">

        <Add mover_name="shear_move_gsa"/>

        <Sort filter="weighted_motif"/>

    </Stage>

    <Stage num_runs_per_input_struct="10"
total_num_results_to_keep="100">

        <Add mover_name="small_move_gsa"/>

        <Sort filter="weighted_motif"/>

    </Stage>

    <Stage num_runs_per_input_struct="10"
total_num_results_to_keep="100">

        <Add mover_name="chim"/>

        <Sort filter="true_filter"/>

    </Stage>

    <Stage num_runs_per_input_struct="10"
total_num_results_to_keep="100">

        <Add mover_name="shear_move_gsa"/>

        <Sort filter="weighted_motif"/>

    </Stage>

```

```

    <Stage num_runs_per_input_struct="10"
total_num_results_to_keep="1000">

        <Add mover_name="small_move_gsa"/>

        <Sort filter="weighted_motif"/>

    </Stage>

    <Stage num_runs_per_input_struct="1"
total_num_results_to_keep="1000">

        <Add mover_name="unvirt"/>

        <Add
mover_name="report_weighted_motifscore_by_residue"/>

        <Add
mover_name="report_longrange_motifscore_by_residue"/>

        <Add
mover_name="report_partner_motifscore_by_residue"/>

        <Add
mover_name="report_weighted_motifscore_by_residue"/>

        <Add
mover_name="report_weighted_motifscore_by_interaction"/>

        <Add
mover_name="report_longrange_motifscore_by_interaction"/>

        <Add
mover_name="report_partner_motifscore_by_interaction"/>

```

```

        <Add
mover_name="report_weighted_motifscore_by_interaction"/>
        <Add mover_name="report_weighted_motifscore"/>
        <Add mover_name="report_longrange_motifscore"/>
        <Add mover_name="report_partner_motifscore"/>
        <Add mover_name="report_weighted_motifscore"/>
        <Add mover_name="relax_1"/>
        <Add mover_name="interface_analyze"/>
        <Add mover_name="report_size"/>
        <Add mover_name="report_normalized_score_of_complex"/>
        <Add mover_name="delete_partner"/>
        <Add mover_name="report_normalized_score_of_design"/>
        <Sort filter="true_filter"/>
    </Stage>
</PROTOCOLS>
</Common>
</JobDefinitionFile>

```

Figure 5.23: The RosettaScript for running SewAnything to generate a vav1 affinity reagent with intercalated small and shear moves and an upweighted partner score

```
<JobDefinitionFile>

<Job>

  <Input>

    <PDB filename="vav1_start_node_and_partner.pdb" /> EDIT
THIS (i.e. scaffold.pdb)

  </Input>

</Job>

<Common>

  <SCOREFXNS>

    <ScoreFunction name="common_sfxn"
weights="ref2015.wts"/>

  </SCOREFXNS>

  <RESIDUE_SELECTORS>

    <Chain name="chain_A" chains="1"/>

    <Chain name="chain_B" chains="2"/>

    <ResidueName name="critical_trp" residue_name3="TYR"/>
```

```

    <SecondaryStructure name="loops" overlap="0" minH="1"
minE="1" include_terminal_loops="false" ss="L" />

    <VirtualResidueSelector name="virtuals" />

    <Not name = "real" selector="virtuals"/>

    <And name = "real_loops" selectors="real,loops"/>

    <And name = "real_loops_of_B"
selectors="real,loops,chain_B"/>

    <Neighborhood name="near_B" selector="chain_B" distance =
"6.0"/>

    <Not name = "not_near_B" selector="near_B"/>

    <And name = "A_distant_B" selectors="not_near_B,chain_A"/>

    <And name = "A_near_B" selectors="near_B,chain_A"/>

    <Or name = "A_and_A_near_B" selectors="A_near_B,chain_A"/>

</RESIDUE_SELECTORS>

<MOVE_MAP_FACTORIES>

    <MoveMapFactory name="partner_sewing_movemap" bb="0"
chi="0">

        <Backbone residue_selector="chain_B"/>

        <Chi residue_selector="A_and_A_near_B"/>

    </MoveMapFactory>

</MOVE_MAP_FACTORIES>

```

<TASKOPERATIONS> 'to' is a suffix for all task-ops | repack  
interface, minimize all atoms

```
<InitializeFromCommandline name="ifc"/>
    <IncludeCurrent name="ic"/>
    <ExtraRotamersGeneric ex1="true" ex2="false"
name="exlex2"/>
    <OperateOnResidueSubset name="repack_partner" selector =
"chain_A">
        <RestrictToRepackingRLT/>
    </OperateOnResidueSubset>
    <OperateOnResidueSubset
name="do_not_repack_distant_partner" selector = "A_distant_B">
        <PreventRepackingRLT/>
    </OperateOnResidueSubset>
</TASKOPERATIONS>
```

```
<FILTERS>
    <PoseCompatibleMotifScorer name="local_by_interaction"
first_selector="chain_B" normalize_by_interactions="true"
confidence = "0"/>
```

```

    <PoseCompatibleMotifScorer
name="longrange_by_interaction" first_selector="chain_B"
sequence_gap = "20" normalize_by_interactions="true"
confidence = "0"/>

    <PoseCompatibleMotifScorer name="partner_by_interaction"
first_selector="chain_A" second_selector="chain_B"
normalize_by_interactions="true" confidence = "0"/>

    <CombinedValue name="weighted_motif_by_interaction" >
        <Add filter_name="local_by_interaction"/>
        <Add filter_name="longrange_by_interaction" factor =
"10" />
        <Add filter_name="partner_by_interaction" factor = "10"
/>
    </CombinedValue>

    <PoseCompatibleMotifScorer name="local_by_residue"
first_selector="chain_B" normalize_by_interactions="false"
confidence = "0"/>

    <PoseCompatibleMotifScorer name="longrange_by_residue"
first_selector="chain_B" sequence_gap = "20"
normalize_by_interactions="false" confidence = "0"/>

```

```

    <PoseCompatibleMotifScorer name="partner_by_residue"
first_selector="chain_A" second_selector="chain_B"
normalize_by_interactions="false" confidence = "0"/>

    <CombinedValue name="weighted_motif_by_residue" >

        <Add filter_name="local_by_residue"/>

        <Add filter_name="longrange_by_residue" factor = "10" />

        <Add filter_name="partner_by_residue" factor = "10" />

    </CombinedValue>

    <PoseCompatibleMotifScorer name="local"
first_selector="chain_B" normalize_by_interactions="false"
normalize_by_residues="false" confidence = "0"/>

    <PoseCompatibleMotifScorer name="longrange"
first_selector="chain_B" sequence_gap = "20"
normalize_by_interactions="false"
normalize_by_residues="false" confidence = "0"/>

    <PoseCompatibleMotifScorer name="partner"
first_selector="chain_A" second_selector="chain_B"
normalize_by_interactions="false"
normalize_by_residues="false" confidence = "0"/>

```



```

<CombinedValue name="weighted_motif" >

  <Add filter_name="local"/>

  <Add filter_name="longrange" factor = "10" />

  <Add filter_name="partner" factor = "20" />

</CombinedValue>


<TrueFilter name="true_filter"/>


<ResidueCount name="size" confidence="0"/>

<ResidueCount name="size_of_A" confidence="0"
residue_selector="chain_A"/>

<ResidueCount name="size_of_B" confidence="0"
residue_selector="chain_B"/>

<ScoreType name="sfxn_filter" score_type="total_score"
scorefxn="common_sfxn" threshold="999999"/>


<CalculatorFilter name="normalized_score"
equation="total_score / res" confidence="0">

  <VAR name="total_score" filter="sfxn_filter"/>

  <VAR name="res" filter="size"/>

</CalculatorFilter>

```

</FILTERS>

<MOVERS>

```
<AddFlankingVirtualResiduesMover name="virt"
N_term_residues="200" C_term_residues="200"
chain_to_modify="2" vital_selector="critical_trp"/>

<ChimerizeSuperSecondaryStructuralSegmentsIntoFlankingVirtualR
esiduesMover name="chim" window_width="2"
alignment_max_distance="0.5"
decompose_path="/nas/longleaf/home/frankdt/scratch/multistage_
SEWING/decompose_test"
read_segments_from_decompose_file="true"
read_segments_from_pdb_list = "false"
permissible_segment_ends="H" model_file_name =
"test_motif_file" pdb_file_name = "test_pdbs"
max_attempts="1000"/>

<FilterReportAsPoseExtraScoresMover
name="report_weighted_motifscore_by_residue" report_as =
"MS_weighted_by_residue"
filter_name="weighted_motif_by_residue"/>
```

```

    <FilterReportAsPoseExtraScoresMover
name="report_motifscore_by_residue" report_as =
"MS_local_by_residue" filter_name="local_by_residue"/>

    <FilterReportAsPoseExtraScoresMover
name="report_longrange_motifscore_by_residue" report_as =
"MS_longrange_by_residue" filter_name="longrange_by_residue"/>

    <FilterReportAsPoseExtraScoresMover
name="report_partner_motifscore_by_residue" report_as =
"MS_partner_by_residue" filter_name="partner_by_residue"/>


    <FilterReportAsPoseExtraScoresMover
name="report_weighted_motifscore_by_interaction" report_as =
"MS_weighted_by_interaction"
filter_name="weighted_motif_by_interaction"/>

    <FilterReportAsPoseExtraScoresMover
name="report_motifscore_by_interaction" report_as =
"MS_local_by_interaction" filter_name="local_by_interaction"/>

    <FilterReportAsPoseExtraScoresMover
name="report_longrange_motifscore_by_interaction" report_as =
"MS_longrange_by_interaction"
filter_name="longrange_by_interaction"/>

```

```
<FilterReportAsPoseExtraScoresMover  
name="report_partner_motifscore_by_interaction" report_as =  
"MS_partner_by_interaction"  
filter_name="partner_by_interaction"/>
```

```
<FilterReportAsPoseExtraScoresMover  
name="report_weighted_motifscore" report_as = "MS_weighted"  
filter_name="weighted_motif"/>
```

```
<FilterReportAsPoseExtraScoresMover  
name="report_motifscore" report_as = "MS_local"  
filter_name="local"/>
```

```
<FilterReportAsPoseExtraScoresMover  
name="report_longrange_motifscore" report_as = "MS_longrange"  
filter_name="longrange"/>
```

```
<FilterReportAsPoseExtraScoresMover  
name="report_partner_motifscore" report_as = "MS_partner"  
filter_name="partner"/>
```

```
<FilterReportAsPoseExtraScoresMover name="report_size"  
report_as = "size" filter_name="size"/>
```

```

    <FilterReportAsPoseExtraScoresMover
name="report_size_of_A" report_as = "size_of_A"
filter_name="size_of_A"/>

    <FilterReportAsPoseExtraScoresMover
name="report_size_of_B" report_as = "size_of_B"
filter_name="size_of_B"/>

    <FilterReportAsPoseExtraScoresMover
name="report_normalized_score_of_complex" report_as =
"normalized_score_of_complex" filter_name="normalized_score"/>

    <FilterReportAsPoseExtraScoresMover
name="report_normalized_score_of_design" report_as =
"normalized_score_of_design" filter_name="normalized_score"/>

    <Small name = "small_mover"
residue_selector="real_loops_of_B" nmoves = "1"
angle_max="12"/>

    <GenericSimulatedAnnealer name="small_move_gsa"
mover_name="small_mover"
filter_name="weighted_motif_by_interaction" trials="10"
sample_type="low" temperature="0.1"/>

```

```

    <Shear name = "shear_mover"
residue_selector="real_loops_of_B" nmoves = "1"
angle_max="12"/>

    <GenericSimulatedAnnealer name="shear_move_gsa"
mover_name="shear_mover"
filter_name="weighted_motif_by_interaction" trials="10"
sample_type="low" temperature="0.1"/>

    <DeleteRegionMover name="unvirt"
residue_selector="virtuals"/>

    <DeleteRegionMover name="delete_partner"
residue_selector="chain_A"/>

    <InterfaceAnalyzerMover name="interface_analyze" scorefxn
= "common_sfxn" pack_separated="false" pack_input = "false"
jump="1"/>

    <FastRelax disable_design="false" name="relax_1"
repeats="5" scorefxn="common_sfxn"
task_operations="ifc,ic,exlex2,repack_partner,do_not_repack_di
stant_partner" movemap_factory="partner_sewing_movemap"
relaxscript="rosettacon2018"/>

    <SwitchResidueTypeSetMover name="to_fa"
set="fa_standard"/>

```

</MOVERS>

<PROTOCOLS>

```
    <Stage num_runs_per_input_struct="1"
total_num_results_to_keep="1">
    <Add mover_name="virt"/>
    Sort filter="weighted_motif"/>
    <Sort filter="true_filter"/>
    </Stage>
    <Stage num_runs_per_input_struct="1000"
total_num_results_to_keep="100">
    <Add mover_name="chim"/>
    <Sort filter="weighted_motif"/>
    </Stage>
    <Stage num_runs_per_input_struct="10"
total_num_results_to_keep="100">
    <Add mover_name="shear_move_gsa"/>
    <Sort filter="weighted_motif"/>
    </Stage>
    <Stage num_runs_per_input_struct="10"
total_num_results_to_keep="100">
```

```

        <Add mover_name="small_move_gsa"/>

    <Sort filter="weighted_motif"/>

</Stage>

    <Stage num_runs_per_input_struct="10"
total_num_results_to_keep="100">

        <Add mover_name="chim"/>

    <Sort filter="weighted_motif"/>

</Stage>

    <Stage num_runs_per_input_struct="10"
total_num_results_to_keep="100">

        <Add mover_name="shear_move_gsa"/>

    <Sort filter="weighted_motif"/>

</Stage>

    <Stage num_runs_per_input_struct="10"
total_num_results_to_keep="100">

        <Add mover_name="small_move_gsa"/>

    <Sort filter="weighted_motif"/>

</Stage>

    <Stage num_runs_per_input_struct="10"
total_num_results_to_keep="100">

        <Add mover_name="chim"/>

    <Sort filter="weighted_motif"/>

```



```

    </Stage>

    <Stage num_runs_per_input_struct="10"
total_num_results_to_keep="100">

        <Add mover_name="shear_move_gsa"/>

        <Sort filter="weighted_motif"/>

    </Stage>

    <Stage num_runs_per_input_struct="10"
total_num_results_to_keep="100">

        <Add mover_name="small_move_gsa"/>

        <Sort filter="weighted_motif"/>

    </Stage>

    <Stage num_runs_per_input_struct="10"
total_num_results_to_keep="100">

        <Add mover_name="chim"/>

        <Sort filter="true_filter"/>

    </Stage>

    <Stage num_runs_per_input_struct="10"
total_num_results_to_keep="100">

        <Add mover_name="shear_move_gsa"/>

        <Sort filter="weighted_motif"/>

    </Stage>

```

```

    <Stage num_runs_per_input_struct="10"
total_num_results_to_keep="1000">

        <Add mover_name="small_move_gsa"/>

        <Sort filter="weighted_motif"/>

    </Stage>

    <Stage num_runs_per_input_struct="1"
total_num_results_to_keep="1000">

        <Add mover_name="unvirt"/>

        <Add
mover_name="report_weighted_motifscore_by_residue"/>

        <Add
mover_name="report_longrange_motifscore_by_residue"/>

        <Add
mover_name="report_partner_motifscore_by_residue"/>

        <Add
mover_name="report_weighted_motifscore_by_residue"/>

        <Add
mover_name="report_weighted_motifscore_by_interaction"/>

        <Add
mover_name="report_longrange_motifscore_by_interaction"/>

        <Add
mover_name="report_partner_motifscore_by_interaction"/>

```

```

        <Add
mover_name="report_weighted_motifscore_by_interaction"/>
        <Add mover_name="report_weighted_motifscore"/>
        <Add mover_name="report_longrange_motifscore"/>
        <Add mover_name="report_partner_motifscore"/>
        <Add mover_name="report_weighted_motifscore"/>
        <Add mover_name="relax_1"/>
        <Add mover_name="interface_analyze"/>
        <Add mover_name="report_size"/>
        <Add mover_name="report_normalized_score_of_complex"/>
        <Add mover_name="delete_partner"/>
        <Add mover_name="report_normalized_score_of_design"/>
        <Sort filter="true_filter"/>
    </Stage>
</PROTOCOLS>
</Common>
</JobDefinitionFile>

```

## CHAPTER 6: CONCLUSIONS

### 6.1 The Advantages of Function-Driven Protein Design

The currently dominant paradigm in protein design is one in which as much of the native structure of the protein as possible is kept unless there is a compelling reason to mutate it, effectively relying at least in part on idiosyncratic compatibilities in protein structure to make it possible to integrate novel functionality into existing proteins, which limits both the range of available targets and the amenability of the entire process to automation. By contrast, function-driven design eschews as much of the native protein as possible, and with it all of the structural and functional compromises that accompany natural proteins, in favor of entirely artificial structures designed *de novo* to facilitate a specific function set by allowing the direct computational optimization the geometric relationships between functional moieties and consequent sequence-agnostic stabilization of the optimal substructure arrangement. Linker design can do this with whole proteins as substructures while SEWING and its new iteration SewAnything can do so on the level of individual protein domains; in either case, the primary advantage of these design protocols is the efficiency with which they navigate vast regions of structural space. Other protocols exist for designing to specific geometric parameters, but functional design is unparalleled in its ability to exploit the structural versatility of proteins for arbitrary

functional purposes.

## 6.2 Present Inadequacies in Function-Driven Design

### 6.2a Beta sheets

SEWING cannot incorporate beta strand-containing structural motifs into its designs except as part of the starting node; while SewAnything is algorithmically capable of incorporating beta-containing motifs, the relatively higher contact order of beta strands still poses a significant problem for including them in designs, which limits the contexts in which domain-level function-driven design can be used. While initial trials with SewAnything have produced beta meanders, none have yet been tested, and optimal parameters for including beta sheet structures in designs have not yet been determined.

### 6.2b Designing for diversity

At present, function-driven design still operates in a broadly classical paradigm in that all designs are compared to find the single best fit for the intended functional set; while this is a suitable approach for designing to fully known parameters, there is presently no way, for example, to construct a library of interfaces that might bind a protein of unknown structure and then design a set of backbones to support them all with minimal redundancy. As the per-sequence cost of gene synthesis drops and directed evolution becomes more feasible for a broad variety of functional parameters, it may be worthwhile to take advantage of the all-vs-all comparative capability of the multistage RosettaScripts framework to design libraries at once rather than one member at a time; furthermore,

SewAnything can be tuned to overpopulate the design pool with good-scoring designs for diversification at the backbone level during the run.

### 6.2c Intercalated remedial design

Fragment-based *ab initio* folding is frequently used to validate the overall feasibility of a design, identifying regions where the intended sequence cannot adopt the intended structure. At present, such designs are typically discarded, but SEWING can design around large proteins as easily as small ones, and it may be worthwhile in certain cases to simply excise the dysfunctional region and replace it with a different structure that holds the remaining protein substructures in the same relative orientation. While scoring functionality exists to drive SEWING designs toward a specific residue, the overall utility of point-to-point design to scaffold protein subunits together remains untested, and no protocol yet exists to identify and constrain corresponding residues between the SEWING-created portion and the native protein to overlay each other, although the existing ConvertRealtoVirtualResiduesMover suggests the solution may be to adapt the AddFlankingVirtualResiduesMover to transfer residue identity information to the remaining residues either side of the replaced region.

### 6.3 Final Remarks

The incredible structural and functional diversity of proteins both motivates extensive study of the spatiotemporal dynamics governing their activity and provides a versatile toolset with which to construct nanoscale tools to facilitate that study, from affinity reagents to optogenetic switches to biosensors of protein activity. The

complexity of the photosystems and other constructs for transducing macroscale signals into the cellular environment and retrieving information precisely from it will only increase as further details of the dynamics of these systems become apparent and of scientific interest, and the present emphasis on high-throughput experimentation suggests that any attempt to design the application-specific components of such tools must be robust and generally applicable if it is to be most useful, militating against structure-driven approaches to identify optimal backbone conformations *a priori*. Requirement-driven design offers a way to maximize the general applicability of sensing and perturbation schemes to a wide range of proteins by abandoning structural specificity entirely and instead designing backbones for subsequent residue-level design with solely functional constraints, making it possible to integrate disparate functional requirements that never occur together in nature into either single backbones or single complexes in a way that appears to also allow for concomitant improvements in stability. When coupled with more generally applicable optogenetic photosystems such as the z-Lock system described in Chapter 2 and more sensitive biosensor modalities, requirement-driven design has the potential to translate algorithmic improvements in search speed into immediately useful increases in our capacity to integrate more demanding functional requirements into arbitrary proteins, and may be expected to play an increasingly prominent role in functional protein design as new advances in protein structure prediction more fully realize the potential of proteins as nanoscale machines.

## REFERENCES

- Agarwal, P. K. (2006). Enzymes: An integrated view of structure, dynamics and function. *Microbial Cell Factories*, 5, 2.
- Aizawa, H. (1996). Overexpression of cofilin stimulates bundling of actin filaments, membrane ruffling, and cell movement in Dictyostelium. *The Journal of Cell Biology*, Vol. 132, pp. 335–344. <https://doi.org/10.1083/jcb.132.3.335>
- Aizawa, H., Wakatsuki, S., Ishii, A., Moriyama, K., Sasaki, Y., Ohashi, K., ... Yahara, I. (2001). Phosphorylation of cofilin by LIM-kinase is necessary for semaphorin 3A-induced growth cone collapse. *Nature Neuroscience*, Vol. 4, pp. 367–373. <https://doi.org/10.1038/86011>
- Aldridge, C., Cain, P., & Robinson, C. (2009). Protein transport in organelles: Protein transport into and across the thylakoid membrane. *FEBS Journal*, Vol. 276, pp. 1177–1186. <https://doi.org/10.1111/j.1742-4658.2009.06875.x>
- Anfinsen, C. B. (1973). Principles that govern the folding of protein chains. *Science*, 181(4096), 223–230.
- Anfinsen, C. B., Haber, E., Sela, M., & White, F. H., Jr. (1961). The kinetics of formation of native ribonuclease during oxidation of the reduced polypeptide chain. *Proceedings of the National Academy of Sciences of the United States of America*, 47, 1309–1314.
- Assaf, F., & Schiller, Y. (2016). The antiepileptic and ictogenic effects of optogenetic neurostimulation of PV-expressing interneurons. *Journal of Neurophysiology*, 116(4), 1694–1704.
- Babalhavaeji, A., & Woolley, G. A. (2018). Modular design of optically controlled protein affinity reagents. *Chemical Communications*, 54(13), 1591–1594.
- Bazzoli, A., Kelow, S. P., & Karanicolas, J. (2015). Enhancements to the Rosetta Energy Function Enable Improved Identification of Small Molecules that Inhibit Protein-Protein Interactions. *PLOS ONE*, Vol. 10, p. e0140359. <https://doi.org/10.1371/journal.pone.0140359>
- Beaty, B. T., Sharma, V. P., Bravo-Cordero, J. J., Simpson, M. A., Eddy, R. J., Koleske, A. J., & Condeelis, J. (2013).  $\beta$ 1 integrin regulates Arg to promote invadopodial maturation and matrix degradation. *Molecular Biology of the Cell*, Vol. 24, pp. 1661–1675. <https://doi.org/10.1091/mbc.e12-12-0908>



- Beermann, A. E., & Jay, D. G. (1994). Chromophore-assisted laser inactivation of cellular proteins. *Methods in Cell Biology*, 44, 715–732.
- Berstein, G. (2019). *G Protein-Coupled Receptors*. CRC Press.
- Bonneau, R., Ruczinski, I., Tsai, J., & Baker, D. (2002). Contact order and ab initio protein structure prediction. *Protein Science*, Vol. 11, pp. 1937–1944. <https://doi.org/10.1110/ps.3790102>
- Bonneau, R., Strauss, C. E. M., Rohl, C. A., Chivian, D., Bradley, P., Malmström, L., ... Baker, D. (2002). De Novo Prediction of Three-dimensional Structures for Major Protein Families. *Journal of Molecular Biology*, Vol. 322, pp. 65–78. [https://doi.org/10.1016/s0022-2836\(02\)00698-8](https://doi.org/10.1016/s0022-2836(02)00698-8)
- Bonneau, R., Tsai, J., Ruczinski, I., Chivian, D., Rohl, C., Strauss, C. E. M., & Baker, D. (2001). Rosetta in CASP4: Progress in ab initio protein structure prediction. *Proteins: Structure, Function, and Genetics*, Vol. 45, pp. 119–126. <https://doi.org/10.1002/prot.1170>
- Borra, M. T., Langer, M. R., Slama, J. T., & Denu, J. M. (2004). Substrate Specificity and Kinetic Mechanism of the Sir2 Family of NAD<sup>+</sup>-Dependent Histone/Protein Deacetylases†. *Biochemistry*, Vol. 43, pp. 9877–9887. <https://doi.org/10.1021/bi049592e>
- Boschek, C. B., Apiyo, D. O., Soares, T. A., Engelmann, H. E., Pefaur, N. B., Straatsma, T. P., & Baird, C. L. (2009). Engineering an ultra-stable affinity reagent based on Top7. *Protein Engineering, Design & Selection: PEDS*, 22(5), 325–332.
- Bradley, P., Chivian, D., Meiler, J., Misura, K. M. S., Rohl, C. A., Schief, W. R., ... Baker, D. (2003). Rosetta predictions in CASP5: Successes, failures, and prospects for complete automation. *Proteins: Structure, Function, and Genetics*, Vol. 53, pp. 457–468. <https://doi.org/10.1002/prot.10552>
- Bravo-Cordero, J. J., Magalhaes, M. A. O., Eddy, R. J., Hodgson, L., & Condeelis, J. (2013). Functions of cofilin in cell locomotion and invasion. *Nature Reviews Molecular Cell Biology*, Vol. 14, pp. 405–415. <https://doi.org/10.1038/nrm3609>
- Burguete, A. S., Fenn, T. D., Brunger, A. T., & Pfeffer, S. R. (2008). Rab and Arl GTPase Family Members Cooperate in the Localization of the Golgin GCC185. *Cell*, Vol. 132, pp. 286–298. <https://doi.org/10.1016/j.cell.2007.11.048>
- Camporeze, B., Manica, B. A., Bonafé, G. A., Ferreira, J. J. C., Diniz, A. L., de Oliveira, C. T. P., ... Ortega, M. M. (2018). Optogenetics: the new molecular approach to control functions of neural cells in epilepsy, depression and tumors

- of the central nervous system. *American Journal of Cancer Research*, 8(10), 1900–1918.
- Carlier, M. F., Ressad, F., & Pantaloni, D. (1999). Control of actin dynamics in cell motility. Role of ADF/cofilin. *The Journal of Biological Chemistry*, 274(48), 33827–33830.
- Chan, A. Y., Bailly, M., Zebda, N., Segall, J. E., & Condeelis, J. S. (2000). Role of Cofilin in Epidermal Growth Factor–Stimulated Actin Polymerization and Lamellipod Protrusion. *The Journal of Cell Biology*, Vol. 148, pp. 531–542. <https://doi.org/10.1083/jcb.148.3.531>
- Chan, A. Y., Raft, S., Bailly, M., Wyckoff, J. B., Segall, J. E., & Condeelis, J. S. (1998). EGF stimulates an increase in actin nucleation and filament number at the leading edge of the lamellipod in mammary adenocarcinoma cells. *Journal of Cell Science*, 111 ( Pt 2), 199–211.
- Chang, W. J., Chang, W. P., & Shyu, B. C. (2017). Suppression of cortical seizures by optic stimulation of the reticular thalamus in PV-mhChR2-YFP BAC transgenic mice. *Molecular Brain*, 10(1), 42.
- Chen, Q., & Pollard, T. D. (2013). Actin Filament Severing by Cofilin Dismantles Actin Patches and Produces Mother Filaments for New Patches. *Current Biology*, Vol. 23, pp. 1154–1162. <https://doi.org/10.1016/j.cub.2013.05.005>
- Condeelis, J. (2001). How is actin polymerization nucleated in vivo? *Trends in Cell Biology*, 11(7), 288–293.
- Courtemanche, N., Pollard, T. D., & Chen, Q. (2016). Avoiding artefacts when counting polymerized actin in live cells with LifeAct fused to fluorescent proteins. *Nature Cell Biology*, Vol. 18, pp. 676–683. <https://doi.org/10.1038/ncb3351>
- Dagliyan, O., Tarnawski, M., Chu, P.-H., Shirvanyants, D., Schlichting, I., Dokholyan, N. V., & Hahn, K. M. (2016). Engineering extrinsic disorder to control protein activity in living cells. *Science*, Vol. 354, pp. 1441–1444. <https://doi.org/10.1126/science.aah3404>
- Dalbey, R. E., Chen, M., & Wiedmann, M. (2002). Methods in Protein Targeting, Translocation and Transport. *Protein Targeting, Transport, and Translocation*, pp. 5–34. <https://doi.org/10.1016/b978-012200731-6.50004-5>
- Dang, B., Wu, H., Mulligan, V. K., Mravic, M., Wu, Y., Lemmin, T., ... DeGrado, W. F. (2017). De novo design of covalently constrained mesosize protein scaffolds with unique tertiary structures. *Proceedings of the National Academy of Sciences of the United States of America*, 114(41), 10852–10857.

- Davey, J. A., & Chica, R. A. (2012). Multistate approaches in computational protein design. *Protein Science*, Vol. 21, pp. 1241–1252.  
<https://doi.org/10.1002/pro.2128>
- Davey, J. A., & Chica, R. A. (2014). Improving the accuracy of protein stability predictions with multistate design using a variety of backbone ensembles. *Proteins: Structure, Function, and Bioinformatics*, Vol. 82, pp. 771–784.  
<https://doi.org/10.1002/prot.24457>
- Debrunner (ed. ), P., Tsibris (ed., J. C., & Münck (ed. ), E. (1969). *Mössbauer Spectroscopy in Biological Systems: Proceedings of a Meeting Held at Allerton House, March 17 and 18, 1969, Monticello, Ill., Organized by the University of Illinois at Urbana-Champaign, Departments of Chemistry and Physics*.
- Der, B. S., Machius, M., Miley, M. J., Mills, J. L., Szyperski, T., & Kuhlman, B. (2012). Metal-Mediated Affinity and Orientation Specificity in a Computationally Designed Protein Homodimer. *Journal of the American Chemical Society*, Vol. 134, pp. 375–385. <https://doi.org/10.1021/ja208015j>
- DesMarais, V., Ghosh, M., Eddy, R., & Condeelis, J. (2005). Cofilin takes the lead. *Journal of Cell Science*, Vol. 118, pp. 19–26. <https://doi.org/10.1242/jcs.01631>
- Development and Biologic Properties of Malignant Cell Sublines and Clones of a Spontaneously Metastasizing Rat Mammary Adenocarcinoma<sup>23</sup>. (1982). *JNCI: Journal of the National Cancer Institute*. <https://doi.org/10.1093/jnci/68.3.507>
- Dormoy, V., Tormanen, K., & Sutterlin, C. (2013). Par6 is at the mother centriole and controls centrosomal protein composition through a Par6 -dependent pathway. *Journal of Cell Science*, Vol. 126, pp. 860–870.  
<https://doi.org/10.1242/jcs.121186>
- Eddy, R. J., Weidmann, M. D., Sharma, V. P., & Condeelis, J. S. (2017). Tumor Cell Invadopodia: Invasive Protrusions that Orchestrate Metastasis. *Trends in Cell Biology*, Vol. 27, pp. 595–607. <https://doi.org/10.1016/j.tcb.2017.03.003>
- Fallas, J. A., Ueda, G., Sheffler, W., Nguyen, V., McNamara, D. E., Sankaran, B., ... Baker, D. (2017). Computational design of self-assembling cyclic protein homo-oligomers. *Nature Chemistry*, 9(4), 353–360.
- Fenno, L. E., & Deisseroth, K. (2014). Optogenetic Tools for Control of Neural Activity. *Neuromethods*, pp. 73–86.  
[https://doi.org/10.1007/978-1-62703-785-3\\_5](https://doi.org/10.1007/978-1-62703-785-3_5)
- Fenno, L., Yizhar, O., & Deisseroth, K. (2011). The Development and Application of Optogenetics. *Annual Review of Neuroscience*, Vol. 34, pp. 389–412.

<https://doi.org/10.1146/annurev-neuro-061010-113817>

Fitch, W. M. (1978). PROTEIN STRUCTURE AND EVOLUTION. *Evolution*, Vol. 32, pp. 225–227. <https://doi.org/10.1111/j.1558-5646.1978.tb01118.x>

Fleishman, S. J., Leaver-Fay, A., Corn, J. E., Strauch, E.-M., Khare, S. D., Koga, N., ... Baker, D. (2011). RosettaScripts: A Scripting Language Interface to the Rosetta Macromolecular Modeling Suite. *PLoS ONE*, Vol. 6, p. e20161. <https://doi.org/10.1371/journal.pone.0020161>

Friedmann, D. R., Aguilar, A., Fan, J., Nachury, M. V., & Marmorstein, R. (2012). Structure of the  $\alpha$ -tubulin acetyltransferase, TAT1, and implications for tubulin-specific acetylation. *Proceedings of the National Academy of Sciences*, Vol. 109, pp. 19655–19660. <https://doi.org/10.1073/pnas.1209357109>

Friedmann, D. R., Fan, J., & Marmorstein, R. (2012). *Structure of the alpha-tubulin acetyltransferase, alpha-TAT1*. <https://doi.org/10.2210/pdb4gs4/pdb>

Gasser, C., Taiber, S., Yeh, C.-M., Wittig, C. H., Hegemann, P., Ryu, S., ... Möglich, A. (2014). Engineering of a red-light-activated human cAMP/cGMP-specific phosphodiesterase. *Proceedings of the National Academy of Sciences of the United States of America*, 111(24), 8803–8808.

Ghosh, M. (2004). Cofilin Promotes Actin Polymerization and Defines the Direction of Cell Motility. *Science*, Vol. 304, pp. 743–746. <https://doi.org/10.1126/science.1094561>

Golob-Urbanc, A., Rajčević, U., Strmšek, Ž., & Jerala, R. (2019). Design of split superantigen fusion proteins for cancer immunotherapy. *The Journal of Biological Chemistry*, 294(16), 6294–6305.

Görllich, D., & Jäkel, S. (2002). Nucleocytoplasmic Transport. *Protein Targeting, Transport, and Translocation*, pp. 293–321. <https://doi.org/10.1016/b978-012200731-6.50015-x>

Govorunova, E. G., Sineshchekov, O. A., & Spudich, J. L. (2019). Three Families of Channelrhodopsins and Their Use in Optogenetics (review). *Neuroscience and Behavioral Physiology*, 49(2), 163–168.

Guffy, S. L., Teets, F. D., Langlois, M. I., & Kuhlman, B. (2018). Protocols for Requirement-Driven Protein Design in the Rosetta Modeling Program. *Journal of Chemical Information and Modeling*, 58(5), 895–901.

Haggarty, S. J., Koeller, K. M., Wong, J. C., Grozinger, C. M., & Schreiber, S. L. (2003). Domain-selective small-molecule inhibitor of histone deacetylase 6

(HDAC6)-mediated tubulin deacetylation. *Proceedings of the National Academy of Sciences*, Vol. 100, pp. 4389–4394. <https://doi.org/10.1073/pnas.0430973100>

Harper, S. M., Christie, J. M., & Gardner, K. H. (2004). Disruption of the LOV-Jalpha helix interaction activates phototropin kinase activity. *Biochemistry*, 43(51), 16184–16192.

Hayashi-Takagi, A., Yagishita, S., Nakamura, M., Shirai, F., Wu, Y. I., Loshbaugh, A. L., ... Kasai, H. (2015). Labelling and optical erasure of synaptic memory traces in the motor cortex. *Nature*, Vol. 525, pp. 333–338. <https://doi.org/10.1038/nature15257>

Hegemann, P., & Nagel, G. (2013). From channelrhodopsins to optogenetics. *EMBO Molecular Medicine*, Vol. 5, pp. 173–176. <https://doi.org/10.1002/emmm.201202387>

Hodgkin, J. (2001). Kinases (Protein Kinases). *Encyclopedia of Genetics*, pp. 1063–1064. <https://doi.org/10.1006/rwgn.2001.0730>

Huang, P.-S., Ban, Y.-E. A., Richter, F., Andre, I., Vernon, R., Schief, W. R., & Baker, D. (2011). RosettaRemodel: a generalized framework for flexible backbone protein design. *PloS One*, 6(8), e24109.

Hubbert, C., Guardiola, A., Shao, R., Kawaguchi, Y., Ito, A., Nixon, A., ... Yao, T.-P. (2002). HDAC6 is a microtubule-associated deacetylase. *Nature*, Vol. 417, pp. 455–458. <https://doi.org/10.1038/417455a>

Hughes, R. M., & Lawrence, D. S. (2014). Optogenetic Engineering: Light-Directed Cell Motility. *Angewandte Chemie International Edition*, Vol. 53, pp. 10904–10907. <https://doi.org/10.1002/anie.201404198>

Hughes, T. R. (2011). *A Handbook of Transcription Factors*. Springer Science & Business Media.

Iyengar, R. (2012). *G Proteins*. Elsevier.

Jacobs, T. M., Williams, B., Williams, T., Xu, X., Eletsky, A., Federizon, J. F., ... Kuhlman, B. (2016). Design of structurally distinct proteins using strategies inspired by evolution. *Science*, 352(6286), 687–690.

Jäschke, A. (2012). Genetically encoded RNA photoswitches as tools for the control of gene expression. *FEBS Letters*, Vol. 586, pp. 2106–2111. <https://doi.org/10.1016/j.febslet.2012.05.040>

Jockusch, B. M. (2017). *The Actin Cytoskeleton*. Springer.

- Joh, N. H., Wang, T., Bhate, M. P., Acharya, R., Wu, Y., Grabe, M., ... DeGrado, W. F. (2014). De novo design of a transmembrane Zn<sup>2+</sup>-transporting four-helix bundle. *Science*, Vol. 346, pp. 1520–1524. <https://doi.org/10.1126/science.1261172>
- Jreij, P. (2016). *The Mechanoregulation of the Actin Cytoskeleton*.
- Kabsch, W., & Sander, C. (1983). Dictionary of protein secondary structure: Pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers*, Vol. 22, pp. 2577–2637. <https://doi.org/10.1002/bip.360221211>
- Kalebic, N., Martinez, C., Perlas, E., Hublitz, P., Bilbao-Cortes, D., Fiedorczuk, K., ... Heppenstall, P. A. (2013). Tubulin acetyltransferase  $\alpha$ TAT1 destabilizes microtubules independently of its acetylation activity. *Molecular and Cellular Biology*, 33(6), 1114–1123.
- Kawano, F., Aono, Y., Suzuki, H., & Sato, M. (2013). Fluorescence imaging-based high-throughput screening of fast- and slow-cycling LOV proteins. *PloS One*, 8(12), e82693.
- Kawano, F., Suzuki, H., Furuya, A., & Sato, M. (2015). Engineered pairs of distinct photoswitches for optogenetic control of cellular proteins. *Nature Communications*, 6, 6256.
- Kellogg, E. H., Leaver-Fay, A., & Baker, D. (2011). Role of conformational sampling in computing mutation-induced changes in protein structure and stability. *Proteins: Structure, Function, and Bioinformatics*, Vol. 79, pp. 830–838. <https://doi.org/10.1002/prot.22921>
- Kicinski, M. (n.d.). *AB INITIO PROTEIN STRUCTURE PREDICTION ALGORITHMS*. <https://doi.org/10.31979/etd.dycd-k9fd>
- Kodani, S., Soya, S., & Sakurai, T. (2019). Optogenetic Manipulation of Neural Circuits During Monitoring Sleep/wakefulness States in Mice. *Journal of Visualized Experiments: JoVE*, (148). <https://doi.org/10.3791/58613>
- Kuhlman, B., Dantas, G., Ireton, G. C., Varani, G., Stoddard, B. L., & Baker, D. (2003). Design of a novel globular protein fold with atomic-level accuracy. *Science*, 302(5649), 1364–1368.
- Kuhlman, B., Jacobs, T., & Linskey, T. (2016). Computational Design of Protein Linkers. *Methods in Molecular Biology*, pp. 341–351. [https://doi.org/10.1007/978-1-4939-3569-7\\_20](https://doi.org/10.1007/978-1-4939-3569-7_20)

- Lai, F. P. L., Szczodrak, M., Block, J., Faix, J., Breitsprecher, D., Mannherz, H. G., ... Rottner, K. (2008). Arp2/3 complex interactions and actin network turnover in lamellipodia. *The EMBO Journal*, Vol. 27, pp. 982–992. <https://doi.org/10.1038/emboj.2008.34>
- Latchman, D. S. (1995). *Eukaryotic transcription factors*.
- Leaver-Fay, A., Jacak, R., Benjamin Stranges, P., & Kuhlman, B. (2011). A Generic Program for Multistate Protein Design. *PLoS ONE*, Vol. 6, p. e20937. <https://doi.org/10.1371/journal.pone.0020937>
- Leaver-Fay, A., Tyka, M., Lewis, S. M., Lange, O. F., Thompson, J., Jacak, R., ... Bradley, P. (2011). Rosetta3. *Computer Methods, Part C*, pp. 545–574. <https://doi.org/10.1016/b978-0-12-381270-4.00019-6>
- Leaver-Fay, A., Tyka, M., Lewis, S. M., Lange, O. F., Thompson, J., Jacak, R., ... Bradley, P. (2011). ROSETTA3: an object-oriented software suite for the simulation and design of macromolecules. *Methods in Enzymology*, 487, 545–574.
- Lesk, A. M., Lo Conte, L., & Hubbard, T. J. P. (2001). Assessment of novel fold targets in CASP4: Predictions of three-dimensional structures, secondary structures, and interresidue contacts. *Proteins: Structure, Function, and Genetics*, Vol. 45, pp. 98–118. <https://doi.org/10.1002/prot.10056>
- Liang, H., Chen, H., Fan, K., Wei, P., Guo, X., Jin, C., ... Lai, L. (2009). De novo design of a beta alpha beta motif. *Angewandte Chemie*, 48(18), 3301–3303.
- Liao, J. C., Roider, J., & Jay, D. G. (1994). Chromophore-assisted laser inactivation of proteins is mediated by the photogeneration of free radicals. *Proceedings of the National Academy of Sciences of the United States of America*, 91(7), 2659–2663.
- Liu, Q., & Tucker, C. L. (2017). Engineering genetically-encoded tools for optogenetic control of protein activity. *Current Opinion in Chemical Biology*, 40, 17–23.
- Locker, J. (2012). *Transcription Factors*. Garland Science.
- Loisel, T. P., Boujemaa, R., Pantaloni, D., & Carlier, M. F. (1999). Reconstitution of actin-based motility of *Listeria* and *Shigella* using pure proteins. *Nature*, 401(6753), 613–616.
- Lombardi, A., Pirro, F., Maglio, O., Chino, M., & DeGrado, W. F. (2019). De Novo Design of Four-Helix Bundle Metalloproteins: One Scaffold, Diverse

Reactivities. *Accounts of Chemical Research*, 52(5), 1148–1159.

Loshbaugh, A. L., & Kortemme, T. (2019). Comparison of Rosetta flexible-backbone computational protein design methods on binding interactions. *Proteins*.  
<https://doi.org/10.1002/prot.25790>

Mak, W. S., & Siegel, J. B. (2014). Computational enzyme design: transitioning from catalytic proteins to enzymes. *Current Opinion in Structural Biology*, 27, 87–94.

Marques, E., & Klefström, J. (2015). Par6 family proteins in cancer. *Oncoscience*, 2(11), 894–895.

Martínez Cuesta, S., Cuesta, S. M., Rahman, S. A., Furnham, N., & Thornton, J. M. (2015). The Classification and Evolution of Enzyme Function. *Biophysical Journal*, Vol. 109, pp. 1082–1086. <https://doi.org/10.1016/j.bpj.2015.04.020>

Miotto, B., & Struhl, K. (2010). HBO1 Histone Acetylase Activity Is Essential for DNA Replication Licensing and Inhibited by Geminin. *Molecular Cell*, Vol. 37, pp. 57–66. <https://doi.org/10.1016/j.molcel.2009.12.012>

Möglich, A., Ayers, R. A., & Moffat, K. (2009). Design and signaling mechanism of light-regulated histidine kinases. *Journal of Molecular Biology*, 385(5), 1433–1444.

Murphy, G. S., Sathyamoorthy, B., Der, B. S., Machius, M. C., Pulavarti, S. V., Szyperski, T., & Kuhlman, B. (2015). Computational de novo design of a four-helix bundle protein--DND\_4HB. *Protein Science: A Publication of the Protein Society*, 24(4), 434–445.

Nagel, G. (2002). Channelrhodopsin-1: A Light-Gated Proton Channel in Green Algae. *Science*, Vol. 296, pp. 2395–2398.  
<https://doi.org/10.1126/science.1072068>

Nagel, G., Szellas, T., Huhn, W., Kateriya, S., Adeishvili, N., Berthold, P., ... Bamberg, E. (2003). Channelrhodopsin-2, a directly light-gated cation-selective membrane channel. *Proceedings of the National Academy of Sciences of the United States of America*, 100(24), 13940–13945.

Nanda, V., & Koder, R. L. (2010). Designing artificial enzymes by intuition and computation. *Nature Chemistry*, 2(1), 15–24.

Nivón, L. G., Moretti, R., & Baker, D. (2013). A Pareto-Optimal Refinement Method for Protein Design Scaffolds. *PLoS ONE*, Vol. 8, p. e59004.  
<https://doi.org/10.1371/journal.pone.0059004>



- North, B. J., Marshall, B. L., Borra, M. T., Denu, J. M., & Verdin, E. (2003). The Human Sir2 Ortholog, SIRT2, Is an NAD<sup>+</sup>-Dependent Tubulin Deacetylase. *Molecular Cell*, Vol. 11, pp. 437–444. [https://doi.org/10.1016/s1097-2765\(03\)00038-8](https://doi.org/10.1016/s1097-2765(03)00038-8)
- O'Banion, C. P., & Lawrence, D. S. (2018). Optogenetics: A Primer for Chemists. *Chembiochem: A European Journal of Chemical Biology*, 19(12), 1201–1216.
- Oser, M., & Condeelis, J. (2009). The cofilin activity cycle in lamellipodia and invadopodia. *Journal of Cellular Biochemistry*, Vol. 108, pp. 1252–1262. <https://doi.org/10.1002/jcb.22372>
- Oser, M., Yamaguchi, H., Mader, C. C., Bravo-Cordero, J. J., Arias, M., Chen, X., ... Condeelis, J. (2009). Cortactin regulates cofilin and N-WASp activities to control the stages of invadopodium assembly and maturation. *The Journal of Cell Biology*, Vol. 186, pp. 571–587. <https://doi.org/10.1083/jcb.200812176>
- Ovchinnikov, S., Park, H., Kim, D. E., DiMaio, F., & Baker, D. (2018). Protein structure prediction using Rosetta in CASP12. *Proteins: Structure, Function, and Bioinformatics*, Vol. 86, pp. 113–121. <https://doi.org/10.1002/prot.25390>
- Ozdamar, B. (2005). Regulation of the Polarity Protein Par6 by TGF Receptors Controls Epithelial Cell Plasticity. *Science*, Vol. 307, pp. 1603–1609. <https://doi.org/10.1126/science.1105718>
- Patthy, L. (2009). *Protein Evolution*. John Wiley & Sons.
- Pope, B., & Weeds, A. G. (1986). Binding of pig plasma gelsolin to F-actin and partial fractionation into calcium-dependent and calcium-independent forms. *European Journal of Biochemistry*, Vol. 161, pp. 85–93. <https://doi.org/10.1111/j.1432-1033.1986.tb10127.x>
- Prowse, T. A. A., & Byrne, M. (2012). Evolution of yolk protein genes in the Echinodermata. *Evolution & Development*, Vol. 14, pp. 139–151. <https://doi.org/10.1111/j.1525-142x.2012.00531.x>
- Pudasaini, A., El-Arab, K. K., & Zoltowski, B. D. (2015). LOV-based optogenetic devices: light-driven modules to impart photoregulated control of cellular signaling. *Frontiers in Molecular Biosciences*, 2, 18.
- Repina, N. A., Rosenbloom, A., Mukherjee, A., Schaffer, D. V., & Kane, R. S. (2017). At Light Speed: Advances in Optogenetic Systems for Regulating Cell Signaling and Behavior. *Annual Review of Chemical and Biomolecular Engineering*, 8, 13–39.

- Rhys, G. G., Wood, C. W., Beesley, J. L., Zaccai, N. R., Burton, A. J., Brady, R. L., ... Woolfson, D. N. (2019). Navigating the Structural Landscape of De Novo  $\alpha$ -Helical Bundles. *Journal of the American Chemical Society*, 141(22), 8787–8797.
- Richter, G., Weber, S., Römisch, W., Bacher, A., Fischer, M., & Eisenreich, W. (2005). Photochemically Induced Dynamic Nuclear Polarization in a C450A Mutant of the LOV2 Domain of the *Avena sativa* Blue-Light Receptor Phototropin. *Journal of the American Chemical Society*, Vol. 127, pp. 17245–17252. <https://doi.org/10.1021/ja053785n>
- Riedl, J., Crevenna, A. H., Kessenbrock, K., Yu, J. H., Neukirchen, D., Bista, M., ... Wedlich-Soldner, R. (2008). Lifeact: a versatile marker to visualize F-actin. *Nature Methods*, Vol. 5, pp. 605–607. <https://doi.org/10.1038/nmeth.1220>
- Rodgers, L. S., & Fanning, A. S. (2011). Regulation of epithelial permeability by the actin cytoskeleton. *Cytoskeleton*, Vol. 68, pp. 653–660. <https://doi.org/10.1002/cm.20547>
- Rohl, C. A., Strauss, C. E. M., Misura, K. M. S., & Baker, D. (2004). Protein Structure Prediction Using Rosetta. *Methods in Enzymology*, pp. 66–93. [https://doi.org/10.1016/s0076-6879\(04\)83004-0](https://doi.org/10.1016/s0076-6879(04)83004-0)
- Rubenstein, A. B., Blacklock, K., Nguyen, H., Case, D. A., & Khare, S. D. (2018). Systematic Comparison of Amber and Rosetta Energy Functions for Protein Structure Evaluation. *Journal of Chemical Theory and Computation*, 14(11), 6015–6025.
- Ryu, M.-H., Kang, I.-H., Nelson, M. D., Jensen, T. M., Lyuksyutova, A. I., Siltberg-Liberles, J., ... Gomelsky, M. (2014). Engineering adenylate cyclases regulated by near-infrared window light. *Proceedings of the National Academy of Sciences of the United States of America*, 111(28), 10167–10172.
- Saier, M. H. (2003). Tracing pathways of transport protein evolution. *Molecular Microbiology*, Vol. 48, pp. 1145–1156. <https://doi.org/10.1046/j.1365-2958.2003.03499.x>
- Schmucker, D., Su, A. L., Beermann, A., Jäckle, H., & Jay, D. G. (1994). Chromophore-assisted laser inactivation of patched protein switches cell fate in the larval visual system of *Drosophila*. *Proceedings of the National Academy of Sciences of the United States of America*, 91(7), 2664–2668.
- Shida, T., Cueva, J. G., Xu, Z., Goodman, M. B., & Nachury, M. V. (2010). The major  $\gamma$ -tubulin K40 acetyltransferase TAT1 promotes rapid ciliogenesis and efficient mechanosensation. *Proceedings of the National Academy of Sciences*,

Vol. 107, pp. 21517–21522. <https://doi.org/10.1073/pnas.1013728107>

- Sidani, M., Wessels, D., Mouneimne, G., Ghosh, M., Goswami, S., Sarmiento, C., ... Condeelis, J. (2007). Cofilin determines the migration behavior and turning frequency of metastatic cancer cells. *The Journal of Cell Biology*, Vol. 179, pp. 777–791. <https://doi.org/10.1083/jcb.200707009>
- Silva, D.-A., Correia, B. E., & Procko, E. (2016). Motif-Driven Design of Protein-Protein Interfaces. *Methods in Molecular Biology*, 1414, 285–304.
- Silva, D.-A., Yu, S., Ulge, U. Y., Spangler, J. B., Jude, K. M., Labão-Almeida, C., ... Baker, D. (2019). De novo design of potent and selective mimics of IL-2 and IL-15. *Nature*, 565(7738), 186–191.
- Smith, B. A., Mularz, A. E., & Hecht, M. H. (2015). Divergent evolution of a bifunctional de novo protein. *Protein Science*, Vol. 24, pp. 246–252. <https://doi.org/10.1002/pro.2611>
- Speltz, E. B., Brown, R. S. H., Hajare, H. S., Schlieker, C., & Regan, L. (2015). A designed repeat protein as an affinity capture reagent. *Biochemical Society Transactions*, 43(5), 874–880.
- Stevens, L. (n.d.). Protein evolution and polymorphism. *Genetics and Evolution of the Domestic Fowl*, pp. 170–207. <https://doi.org/10.1017/cbo9780511525780.011>
- Strickland, D., Lin, Y., Wagner, E., Matthew Hope, C., Zayner, J., Antoniou, C., ... Glotzer, M. (2012). TULIPs: tunable, light-controlled interacting protein tags for cell biology. *Nature Methods*, Vol. 9, pp. 379–384. <https://doi.org/10.1038/nmeth.1904>
- Strickland, D., Moffat, K., & Sosnick, T. R. (2008). Light-activated DNA binding in a designed allosteric protein. *Proceedings of the National Academy of Sciences*, Vol. 105, pp. 10709–10714. <https://doi.org/10.1073/pnas.0709610105>
- Strohmeier, G. A., Pichler, H., May, O., & Gruber-Khadjawi, M. (2011). Application of Designed Enzymes in Organic Synthesis. *Chemical Reviews*, Vol. 111, pp. 4141–4164. <https://doi.org/10.1021/cr100386u>
- Svitkina, T. M., & Borisy, G. G. (1999). Arp2/3 complex and actin depolymerizing factor/cofilin in dendritic organization and treadmilling of actin filament array in lamellipodia. *The Journal of Cell Biology*, 145(5), 1009–1026.
- Szyk, A., Deaconescu, A. M., Spector, J., Goodman, B., Valenstein, M. L., Ziolkowska, N. E., ... Roll-Mecak, A. (2014). Molecular Basis for

- Age-Dependent Microtubule Acetylation by Tubulin Acetyltransferase. *Cell*, Vol. 157, pp. 1405–1415. <https://doi.org/10.1016/j.cell.2014.03.061>
- Thiery, J. P., & Huang, R. (2005). Linking Epithelial-Mesenchymal Transition to the Well-Known Polarity Protein Par6. *Developmental Cell*, Vol. 8, pp. 456–458. <https://doi.org/10.1016/j.devcel.2005.03.002>
- Tyka, M. D., Keedy, D. A., André, I., DiMaio, F., Song, Y., Richardson, D. C., ... Baker, D. (2011). Alternate States of Proteins Revealed by Detailed Energy Landscape Mapping. *Journal of Molecular Biology*, Vol. 405, pp. 607–618. <https://doi.org/10.1016/j.jmb.2010.11.008>
- Vangone, A., Spinelli, R., Scarano, V., Cavallo, L., & Oliva, R. (2011). COCOMAPS: a web application to analyze and visualize contacts at the interface of biomolecular complexes. *Bioinformatics*, 27(20), 2915–2916.
- Wang, G., & Dunbrack, R. L. (2003). PISCES: a protein sequence culling server. *Bioinformatics*, Vol. 19, pp. 1589–1591. <https://doi.org/10.1093/bioinformatics/btg224>
- Wang, H., & Hahn, K. M. (2016). LOVTRAP: A Versatile Method to Control Protein Function with Light. *Current Protocols in Cell Biology / Editorial Board, Juan S. Bonifacino ... [et Al.]*, 73, 21.10.1–21.10.14.
- Wang, H., Vilela, M., Winkler, A., Tarnawski, M., Schlichting, I., Yumerefendi, H., ... Hahn, K. M. (2016). LOVTRAP: an optogenetic system for photoinduced protein dissociation. *Nature Methods*, 13(9), 755–758.
- Wang, X., Chen, X., & Yang, Y. (2012). Spatiotemporal control of gene expression by a light-switchable transgene system. *Nature Methods*, Vol. 9, pp. 266–269. <https://doi.org/10.1038/nmeth.1892>
- Wehler, P., Niopek, D., Eils, R., & Di Ventura, B. (2016). Optogenetic Control of Nuclear Protein Import in Living Cells Using Light-Inducible Nuclear Localization Signals (LINuS). *Current Protocols in Chemical Biology*, pp. 131–145. <https://doi.org/10.1002/cpch.4>
- Weitzman, M., & Hahn, K. M. (2014). Optogenetic approaches to cell migration and beyond. *Current Opinion in Cell Biology*, Vol. 30, pp. 112–120. <https://doi.org/10.1016/j.ceb.2014.08.004>
- Wu, Y. I., Frey, D., Lungu, O. I., Jaehrig, A., Schlichting, I., Kuhlman, B., & Hahn, K. M. (2009). A genetically encoded photoactivatable Rac controls the motility of living cells. *Nature*, 461(7260), 104–108.

- Yamaguchi, H., Lorenz, M., Kempiak, S., Sarmiento, C., Coniglio, S., Symons, M., ... Condeelis, J. (2005). Molecular mechanisms of invadopodium formation. *The Journal of Cell Biology*, Vol. 168, pp. 441–452.  
<https://doi.org/10.1083/jcb.200407076>
- Yang, N., Higuchi, O., Ohashi, K., Nagata, K., Wada, A., Kangawa, K., ... Mizuno, K. (1998). Cofilin phosphorylation by LIM-kinase 1 and its role in Rac-mediated actin reorganization. *Nature*, Vol. 393, pp. 809–812.  
<https://doi.org/10.1038/31735>
- Yang, Z. (2014). Neutral and adaptive protein evolution. *Molecular Evolution*, pp. 390–417. <https://doi.org/10.1093/acprof:oso/9780199602605.003.0011>
- Yazawa, M., Sadaghiani, A. M., Hsueh, B., & Dolmetsch, R. E. (2009). Induction of protein-protein interactions in live cells using light. *Nature Biotechnology*, 27(10), 941–945.
- Yi, J. J., Wang, H., Vilela, M., Danuser, G., & Hahn, K. M. (2014). Manipulation of Endogenous Kinase Activity in Living Cells Using Photoswitchable Inhibitory Peptides. *ACS Synthetic Biology*, Vol. 3, pp. 788–795.  
<https://doi.org/10.1021/sb5001356>
- Yu, B., Martins, I. R. S., Li, P., Amarasinghe, G. K., Umetani, J., Fernandez-Zapico, M. E., ... Rosen, M. K. (2010). Structural and energetic mechanisms of cooperative autoinhibition and activation of Vav1. *Cell*, 140(2), 246–256.
- Yumerefendi, H., Wang, H., Dickinson, D. J., Lerner, A. M., Malkus, P., Goldstein, B., ... Kuhlman, B. (2018). Light-Dependent Cytoplasmic Recruitment Enhances the Dynamic Range of a Nuclear Import Photoswitch. *Chembiochem: A European Journal of Chemical Biology*, 19(12), 1319–1325.
- Yu, Y., & Lutz, S. (2011). Circular permutation: a different way to engineer enzyme structure and function. *Trends in Biotechnology*, Vol. 29, pp. 18–25.  
<https://doi.org/10.1016/j.tibtech.2010.10.004>
- Zafar, S., & Zerr, I. (2013). Kinases (Protein Kinases). *Brenner's Encyclopedia of Genetics*, pp. 166–167. <https://doi.org/10.1016/b978-0-12-374984-0.00833-0>
- Zhou, X. X., Chung, H. K., Lam, A. J., & Lin, M. Z. (2012). Optical Control of Protein Activity by Fluorescent Protein Domains. *Science*, Vol. 338, pp. 810–814.  
<https://doi.org/10.1126/science.1226854>
- Zhou, X. X., Fan, L. Z., Li, P., Shen, K., & Lin, M. Z. (2017). Optical control of cell signaling by single-chain photoswitchable kinases. *Science*, Vol. 355, pp. 836–842. <https://doi.org/10.1126/science.aah3605>

- Zimmerman, S. P., Hallett, R. A., Bourke, A. M., Bear, J. E., Kennedy, M. J., & Kuhlman, B. (2016). Tuning the Binding Affinities and Reversion Kinetics of a Light Inducible Dimer Allows Control of Transmembrane Protein Localization. *Biochemistry*, 55(37), 5264–5271.
- Zoltowski, B. D., Vaccaro, B., & Crane, B. R. (2009). Mechanism-based tuning of a LOV domain photoreceptor. *Nature Chemical Biology*, Vol. 5, pp. 827–834. <https://doi.org/10.1038/nchembio.210>
- Zoncu, R., Perera, R. M., Sebastian, R., Nakatsu, F., Chen, H., Balla, T., ... De Camilli, P. V. (2007). Loss of endocytic clathrin-coated pits upon acute depletion of phosphatidylinositol 4,5-bisphosphate. *Proceedings of the National Academy of Sciences*, Vol. 104, pp. 3793–3798. <https://doi.org/10.1073/pnas.0611733104>