A Hierarchical Approach to the Design of Protein Crystals

Una Wen-lii Nattermann

A dissertation

submitted in partial fulfillment of the

requirements for the degree of

Doctor of Philosophy

University of Washington

2019

Reading Committee:

David Baker, Chair

Jesse Zalatan

Ning Zheng

Program Authorized to Offer Degree:

Biochemistry

University of Washington

Abstract

A Hierarchical Approach to the Design of Protein Crystals

Una Wen-lii Nattermann

Chair of the Supervisory Committee:
Dr. David Baker, Professor of Biochemistry
Biochemistry

Despite the fact that scientists have been using X-ray crystallography for the structure determination of proteins for nearly 80 years, understanding and controlling the crystallization of proteins has remained an outstanding challenge. In order to help address this challenge, the main focus of my doctoral research has been to design self-assembling protein crystals from the bottom up. Inspired by nature's ability to create complicated architectures using the hierarchical assembly of symmetric proteins, my goal has been to design symmetrical protein building blocks that crystalize when mixed together in solution into a predefined lattice architecture and without requiring vapor diffusion conditions. Because of the strict mathematical relationship of symmetry elements in crystal space groups, I designed these protein building blocks to assemble in a way that satisfies the geometric constraints of each lattice architecture. In two cases, I was successful in designing amino acid mutations that resulted in the spontaneous self-assembly of protein crystals; however, they both ended up forming unexpected lattice architectures. To address this challenge of off-target assembly, I developed a hierarchical method that consists of

custom-making building blocks that are geometrically related to one another. I have experimentally characterized tens of these, and am working on continuing to systematically generate crystal designs from these custom building blocks. This new hierarchical design protocol and its results represent a step towards the accurate design of self-assembling protein crystals by using a hierarchical design approach that leverages recent strides in protein design. Achieving a general method to design self-assembling protein crystals would open the door to a new generation of biomaterials that are genetically programmable and also enable unparalleled nanoscale patterning.

TABLE OF CONTENTS

LIST OF FIGURES

List of Tables

ACKNOWLEDGEMENTS

First, I would like to acknowledge that Washington State occupies the land of the Cayuse, Chehalis, Chinook, Colville, Okanagan, Coeur d'Alene, Cowlitz, Kalispel, Spokane Salish, Klallam, Kwalhioqua, Lummi and Samich, Makah, Nez Perce, Nooksack, Quileute, Hoh, Chemakum, Quinault, Skokomish, Thompson Salish, Umatilla, Walla Walla, Palouse, Wasco, Wishram, Yakima, and Puget Sound Salish (Puyallup, Snohomish, Muckleshoot, Snoqualmie, Nisqually, Skagit Suquamish, Squaxi, Swinomish, Stillaguamish, and Sauk-Suiattle) tribes (from http://www.native-languages.org/washington.htm). I am grateful to have been able to inhabit and enjoy this beautiful land over the course of my graduate studies.

As the saying goes, "it takes a village" -- and the statement is no less true with regards to the completion of a graduate degree and the journey involved. First, I would like to thank my primary mentor, David Baker, for the opportunity to work in his group and for truly helpful advice throughout the years related to my project. I would like to thank Jacob Bale, Neil King, and William Sheffler for mentoring me as I started in the Baker Lab. They introduced me to Rosetta, inspired and provided discussion surrounding my research project, and have continued to support me over the years.

This work would not have been possible without the contributions of the large community of developers of the Rosetta macromolecular modeling software, upon which the methods presented here are based. For their assistance and/or advice in computational and/or experimental matters, I would like to specifically thank William Sheffler, Neil King, Jacob Bale, Yang Hsia, Ariel Ben-Sasson, Alexis Courbet, Jorge Fallas, Frank DiMaio, Phil Bradley, Hugh

Haddox, Robby Divine, Fabio Parmeggiani, Jeremy Mills, Daniel Silva, Darwin Alonso, Luki Goldschmidt, Patrick Vecchiato, Andrew Leaver-Fay, Zhe Li, Erin Yang, Radhika Dalal, Hector Vera Torres, Sidney Lisanza, Adam Moyer, Evelyn Tsai, Angelica Yehdego, Lauren Carter, Cameron Chow, Rashmi Ravichandran, Alex Kang, Matthew Bick, Asim Bera, Quinton Dowling, Daniel Ellis, Marc Lajoie, Gabriel Butterfield, George Ueda, Shane Gonen, Christopher Bahl, and Lance Stewart.

For administrative work, I am beyond grateful for the support of Erin Kirschner, Christina Larmore, Michelle Matsunaga, Ratika Krishnamurty, Zari Magness, Leslie Stark, and Lance Stewart.

I would like to give a special thanks to my colleagues: Parisa Hosseinzadeh, Franzsika Seeger, Karla-Luise Herpoldt, Christopher Bahl, Gabriel Rocklin, Marc Lajoie, Gustav Oberdorfer, Scott Boyken, Hugh Haddox, Stephanie Berger, Jorge Fallas, George Ueda, Anindya Roy, and Alexander Ford. You have all been inspirational scientists to work with and have offered invaluable mentorship and advice over the years.

I feel very lucky for the opportunities to collaborate with the labs of Vicki Wysocki, Neil King, Alex Merz, Tamir Gonen, Todd Yeates, and Wes Sundquist.

I would also like to thank the members of my Doctoral Advisory Committee, Chip Asbury, David Veesler, Jesse Zalatan, and Ning Zheng for taking time away from their busy schedules to provide support and advice about various aspects of my research and career development. I want to give special thanks to Barry Stoddard and Ning Zheng for the mentorship and opportunities they provided me during my first year rotations and their labs, and to Gerald

Stubbs for providing me with my first research opportunity as an undergraduate at Vanderbilt University, which helped inspire me to apply to graduate school.

I would like to thank my family -- my father, mother, and brother -- who have provided me with endless support and encouragement throughout my studies and chosen career path. My other family members around the world have continuously offered words of encouragement and support throughout my life.

Finally, I would like to thank my amazing friends who I am so grateful to have had the constant support of during my graduate career. Thank you for your friendship and love over the years. Without you, graduate school would have been much more difficult. You truly enriched my experience here at the University of Washington, and in Seattle.

DEDICATION

For my father, who encouraged me in everything.

For my mother and brother, who continue to do so.

INTRODUCTION

There is growing interest in the field of biomaterials and their ability to endow highly accurate spatial patterning at the nanometer scale. In comparison to canonical metal- and polymer-based materials, macromolecules present an intrinsically biodegradable platform with potential for sub-atomic precision and sophisticated functionality. Additionally, natural materials – specifically silks – have been shown to outperform synthetic materials by being both strong and flexible in ways that synthetic materials cannot yet mimic [1]. In the last two decades, nucleic acid nanomaterials, and specifically those made from DNA, have had a surge in examples of the impressively complicated supramolecular structures that can be accurately engineered by researchers [2]. This is due to the stereotyped Watson-Crick base-pairing interactions of DNA that allow for the straightforward design of various shapes and materials. While DNA-based materials certainly have a plethora of applications, it is notable that in nature, cells use nucleic acids primarily for information storage and use proteins as the building blocks for complex architectures and nanomachines. Additionally, despite the ability of researchers to engineer an impressive slew of structures using DNA origami and nanotechnology, there is only one report of an engineered entirely DNA-based three-dimensional crystal [16]. Proteins are macromolecular machines that naturally display unparalleled self-assembly properties. This suggests that engineering protein-based materials could open new areas of materials mechanical and chemical properties.

Over the past five years, protein design [3] has been steadily catching up to the versatility of DNA nanotechnology. For example, there have been multiple examples of successes in the

computational design of cyclic [4,5], dihedral [6], and polyhedral [7,8,9,10] bounded protein assemblies as well as of unbounded one-dimensional filaments [11], and two-dimensional sheets [12,13] made entirely from protein building blocks . These research reports all take advantage of a design strategy exhibited by nature in that producing symmetric assemblies reduces the number of unique interfaces required to hold the architectures together [14,15]. Here, I try to leverage these same theories to computationally model and design proteins to self-assemble into three-dimensional crystalline materials.

Protein crystallography has been a hallmark of structural biology for nearly a century; and yet, scientists do not have a generalizable method for controlling the crystallization process. Self-assembling macromolecular crystals have a lot of potential for use in a wide variety of basic sciences, health, and environmental applications because of their high degree of order at the nanoscale. While many proteins will crystallize under artificial vapor diffusion conditions, no method currently exists for robust generation of protein crystals with desired architectures designed with high accuracy. Being able to engineer crystals from the bottom up has implications for deepening our understanding of the nucleation and growth of macromolecular crystals in general, chaperoning the crystallization of target proteins for structure determination by X-ray crystallography [17], scaffolding metabolic pathways [18], and patterning functional elements such as metal nanoparticles, among other applications. Because of this, scientists have been pursuing the engineering of nucleic acid- and protein- based crystals for the past two decades.

Other research groups have tackled this oustanding challenge by integrating favorable crystallization motifs consisting mostly of non-protein additions, such as nucleic acids, metals,

and polymers. In some cases where only protein-based motifs were used, researchers relied on nonspecific electrostatic interactions [19] or aromatic motifs. More success has been achieved using hybrid building blocks to construct macromolecular crystalline materials. This has been demonstrated with metal- [20,21] , nucleic acid- [22] , and polymer-protein hybrids [23, 24]. While some of these efforts have yielded crystalline materials, they have also exhibited a number of shortcomings, including requiring special conditions for crystallization such as vapor diffusion methods, forming architectures whose structures were not predicted with high accuracy ahead of time, or yielding structures with insufficient long-range order to be verified with atomic resolution. The only successful example of a protein that crystallized into a designed lattice was a coiled-coil protein into P6 lattice [25]. However, the assembly required artificial vapor diffusion condition. Controlling self-assembly by protein-protein interactions alone could allow for higher accuracy of the orientation of building blocks and allow more readily the design of tunable assembly and disassembly. However, to date, there are only a few examples of entirely protein-based engineered crystals. In addition, there is no generalizable method for the design of all possible lattice architectures. The aim of my doctoral research was to address these current challenges to make protein crystal design a more viable route for the development of functional biomaterials.

Inspired by prior successes in computational design of self-assembling biomaterials, I developed a framework for applying analogous protocols to the design of three-dimensional crystalline materials. I focused on multi-component systems because they expand the theoretical crystal design space, allow for built-in inducible assembly, and intrinsically have more termini available for downstream functionalization via genetic fusions. Within the many options for

three-dimensional crystal lattices formed by combining two symmetric building blocks (Table 1.1), space groups that had a lower number of subunit-subunit interactions required to close the system, or cycle connectivities (Fig. 1.1), and that formed cubic crystal space groups were chosen as the best places to start. Cycle connectivities range from as low as six for combining an octahedron with a D2 oligomer to form an F432 lattice to as high as 20 for combining a C2 and C3 oligomer into a $P4_132$ lattice. Cubic crystal space groups were focused on because all six of the unit cell dimensions of the lattice architecture could be symmetrically sampled, which simplified setting up the computational framework.

First, I sought to use an analogous computational design strategy that was sufficient for the creation of zero- [7,8,9,10], one- [11], and two- [12,13] dimensional self-assembling protein complexes. This research resulted in the development of a framework for the modeling, docking, and design of various crystal space groups formed by combining two symmetric protein building blocks [26]. While I designed, experimentally identified, and characterized a few self-assembling protein crystals, they formed off-target lattice architectures. These unexpected results highlighted a need for increased specificity in the designed interfaces, removal of extraneous exposed hydrophobic residues, and custom-made building block components.

To address these challenges, I developed a new computational framework using the Rosetta macromolecular modeling software [27] that is truly hierarchical and takes advantage of many recent advances in protein design. Broadly speaking, instead of starting with pre-curated databases of either naturally occurring or *de novo* proteins, the building blocks used were custom-made to be particularly amenable for the crystal design strategy. Specifically, we sought to design cubic crystal space groups that contain dihedral and polyhedral centers. Because of

this, I developed methods to quickly design dihedral structures from pre-made cyclic protein assemblies and to geometrically constrain the design of two-component nanocages of which one component is the cyclic-version of the new dihedral protein complexes.

During this work, I pursued designing a two-component $P2_13$ lattice by designing a protein-protein interface between two C3 oligomers, and observed the spontaneous assembly of one of the designed C3 oligomers into that architecture on its own (Chapter 1.3). In addition, when designing a dihedral interface between a two-component cage and itself, the mutations results in the two building blocks self-assembling when mixed into an off-target and more compact architecture using the introduced mutations (Chapter 1.6). To achieve better control of assembly and reduce the possibility of off-target assembly, I then developed a hierarchical crystal design framework. First, I designed dihedral oligomers (Chapter 2.2) of which I have four experimentally and/or structurally verified ones comprised of a naturally occurring protein (Chapter 2.2.1) and nine and counting comprised of *de novo* proteins (Chapter 2.2.2). Second, I constrained these dihedral oligomers' orientation while using them as a building block in nanocage design (Chapter 2.3), and then reverted the dihedral-forming mutations back to the cyclic-forming amino acid identities to allow for testing the designed nanocage interface on its own. By structurally verifying the dihedral and nanocage interfaces of the crystal independent of one another, I reduce the chance of off-target protein-protein interactions occurring. For cases in which a nanocage is successfully created, the dihedral interface mutations are re-introduced to generate the crystal-forming construct (Chapter 2.4). Currently, I am troubleshooting the experimental difficulties associated with introducing many surface-exposed surface hydrophobic residues to protein building blocks as well as regulating the kinetics of the crystal assembly

process. In the future, we plan to use the HBNet protocol [4] to introduce specific hydrogen bonding interactions at the designed protein-protein interfaces.

With the ever-increasing accuracy of *de novo* protein design, this new hierarchical method allows for the independent characterization of the different working parts of a designed protein crystal and for a better understanding of the kinetics of designed crystallization with regards to affinity and specificity.

Chapter 1. Crystal Design Using Pre-Existing Protein Building Blocks

1.1 Development of a Computational Framework for Designing Protein Crystals

Building off of the successes of other designed materials and utilizing both native oligomers from the Protein Data Bank (PDB) and *de novo* oligomers designed by members of the Baker group [4,5], I decided to develop a computational method using the Rosetta macromolecular modeling suite [27] capable of yielding crystalline structures constructed from complex protein scaffolds with atomic-level accuracy. We have elucidated all of the theoretical combinations of point group symmetries that result in three-dimensional arrays (Table 1.1) [26] and a corresponding database of known native and *de novo* oligomeric protein structures has been curated for use as design building blocks. We chose to focus on crystal lattices that had lower cycle connectivities (Fig. 1.1), meaning that it takes a fewer number of subunit-subunit interactions to create a bounded structure within the overall architecture. Using the symmetry modeling framework in Rosetta [28], protein oligomers with cyclic, dihedral, and/or polyhedral symmetries will first be aligned along corresponding crystal symmetry axes. Next, the identified degrees of freedom of rotation or translation between crystal building blocks will be sampled within the constraints of the crystal lattice to generate starting points for design. Finally, the identities of the residues constituting the interface between components will be computationally optimized to drive assembly into specific three-dimensional lattices.

The design protocol takes advantage of the theory of hierarchical assembly [14] in which the combination of multiple non-covalent interactions between macromolecules drive assembly. We hypothesize that through the design of symmetrically related, weak, non-covalent interactions we can promote very precise orientations of protein-protein interactions [15] which will allow for the highly accurate self-assembly of components into a pre- determined crystal lattice. Using symmetric buildings blocks allows for the ability to control several protein-protein interactions by designing a single interface, which minimizes the number of mutations needed per building block to drive assembly.

First, a scaffold library of point symmetric building blocks was curated from existing structures in the PDB (Chapter 1.2.1) and from *de novo* protein oligomer designs from the Baker group (Chapter 1.2.2). Having more oligomer building blocks increases the sampling we can perform, which increases the likelihood of a successful design. Because of this, efforts are also being made to create a high-throughput and accurate method for the design of *de novo* protein oligomers (Chapter 1.2.3, 1.2.4, 2.2). All of the computational protocols described in this thesis were performed using the RosettaScripts interface [30] to the Rosetta macromolecular modeling software [27].

The scaffold library is used in a protein-protein docking step that acts within the geometric constraints of the crystal symmetry being designed. The Rosetta codebase already contained some software for modeling symmetric protein assemblies [28]; however, three-dimensional arrays had not yet been modeled. Because of this, I modified the existing code to include new attributes making it compatible with modeling three-dimensional crystals and

properly constraining assemblies in a way that is consistent with the desired lattice architecture (Appendix Methods 1.1). Sampling is allowed along identified degrees of freedom, which vary depending on the symmetries of the building blocks being used and the desired crystalline architecture. For example, when docking octahedral and D4 dihedral oligomers into an P432 crystal space group, there is only one full degree of freedom (DOF) in which the octahedral oligomer and D4 oligomer slide into contact with each other along their four-fold symmetry axes. There is also a rotational DOF of the D4 oligomers if the octahedron is kept static; however, this DOF only has two discrete options: 0° and 45°. In another scenario, a C3 cyclic oligomer along with another C3 cyclic oligomer can be combined to form a $P2_13$ crystal space group (Fig. 1.6). In this case, both C3 oligomers can rotate around their three-fold axes, can translate in relation to each other, and also slide into contact. However, the dihedral angle between the two oligomer's three-fold axes must remain constant during all of this sampling. These two examples demonstrate the flexibility that the computational framework must have to accurately generate and model different combinations of oligomers in various crystal architectures.

Notably, the modifications I made to the Rosetta symmetry modeling code are sufficient for designing cubic space group architectures because I added changes that make it easy to sample the unit cell dimensions related to spacing in a symmetric fashion while restricting the angles of unit cell to 90°, as is consistent with cubic space groups. To expand this code to allow for properly sampling the DOFs of non-cubic space groups, one would have to further modify the Rosetta symmetry modeling code.

Once the possibilities of docked proteins are generated, they are ranked by their "motif score," which is a value that correlates to how many pairwise favorable amino acid interactions can fit at the interface [5]. These interactions can be modulated to contain all hydrophobic residues, only aliphatic residues, or any combination of pairwise motifs that the designer wants. At the docking stage, focus is put on potential protein-protein interfaces that can accomodate favorable hydrophobic interactions, and hydrophilic amino acids are assigned to the protein-protein interfaces during the following design stage. The docked architectures with the highest motif scores are taken on to the next step of Rosetta interface design, the first step of which is re-generating the symmetric architecture. Currently, I am able to manually generate symmetry definition files that represent the architectures and DOFs allowed for a specific combination of oligomers. While this method works, it is not easily amenable to fully searching the crystal design space or changing the number of building blocks being used to generate the protein crystal. Next, using Rosetta design, the protein-protein interfaces between building blocks are re-engineered to contain mutations that promote binding between the protein components. After filtering designs based on Rosetta metrics such as interface size, shape complementarity of the components, energy of binding, the number of mutations, and unsatisfied hydrogen bonds, genes are ordered for the various proteins at which point experimental characterization of the designed constructs begins.

After computationally designing constructs, I used multiple approaches to experimentally characterize them in the lab. Below, I describe the typical experimental workflow that was tested. The designs are tested as both polycistronically expressed genes as well as genes

expressed independently and mixed after purification. After expression in bacterial cell culture, designs are analyzed for expression, solubility, and binding by SDS-PAGE and affinity chromatography pull-down assays. The first step is to ensure that the designed mutations have not disrupted the stability and foldability of the protein scaffolds. Small-scale expression tests monitored by SDS-PAGE gels of the insoluble and soluble fractions of cell lysate help filter out designs in which the proteins are not expressing well. For designs that are primarily insoluble, attempts are made to wash the insoluble fraction to screen for *in vivo* crystal formation by negative-stain electron microscopy. For those that are soluble, larger-scale expression and purification are necessary. In these two- component systems, only one building block has a hexahistidine tag added. Because of this, immobilized metal affinity chromatography (IMAC) is used to determine whether the two components are interacting with each other. SDS-PAGE gels of elution fractions are run and designs in which two bands of the correct molecular weight can be seen are further investigated. Mass spectrometry (MS) is used to determine whether the proteins present in the elution are indeed of the correct molecular weight and sequence of the desired interacting proteins.

Macroscopic crystal formation is monitored by light microscopy; and for nanoscale nucleation, I utilized negative-stain transmission electron microscopy (EM) to screen samples for crystal formation and/or binding of the components. If crystals do not form upon mixing of the components, I set up crystallization screens using vapor diffusion to see if the proteins crystallize artificially into the desired lattice architecture, as this could be important for future design methods. If the designed crystals do form, I try titrations of the various components, as well as

vary the pH of the buffer solutions of the components, to try regulating the crystal growth. The extent of 3D lattice formation for designs for which the two components are found to interact is investigated by small angle X-ray scattering (SAXS), electron microscopy, and X-ray crystallography for the most successful outcomes. Understanding the physical and chemical characteristics of successful protein crystal designs using biophysical techniques guides further improvements in our design methods and research into potential functional applications. Because of the iterative nature of the design process, my efforts were split between computational and experimental work. Designing new crystals while characterizing ones whose genes have already been ordered allowed me to incorporate experimental results back into the design methods to improve the process with each new round of designs.

1.2 Materials Design Building Block Database

For all the work described, we used protein oligomer building blocks with cyclic and dihedral symmetries as well as one- and two-component nanocages with tetrahedral or octahedral symmetries. We used naturally-occurring (Chapter 1.2.1) and *de novo* protein homo-oligomers (Chapter 1.2.2), and even designed new cyclic (Chapter 1.2.3) and polyhedral (Chapter 1.2.4) building blocks to expand the database.

1.2.1 *Curating naturally-occurring proteins from the Protein Data Bank*

Similar to how a database of homo-oligomers with cyclic symmetries were curated for nanocage design [7,8,9,10], a new database of dihedral oligomers was curated from the PDB for this project. Specifically, we curated crystal structures of proteins with identified symmetries that

were D2, D3, or D4, as those are the only dihedral symmetries found in two-component crystal space groups. In addition, we decided to use only structures that were at a sub-3.0 Å resolution, proteins where the monomer was between 75 and 500 amino acids in length, and from deposited structures in which the protein had been purified from *Escherichia coli*. After downloading these files from the PDB, scripts were written to determine whether the PDB structure itself has the correct number of chains given its tagged protein symmetry and whether the authors of the deposited structure had actually verified the protein as the tagged protein symmetry, versus this being an artifact of the crystallization conditions. All of these steps were taken to curate a database of dihedral protein building blocks that would be easy to work with computationally and experimentally (Appendix Table 1.1).

1.2.2 *Curating de novo designed proteins*

In addition to naturally-occurring proteins that had high-resolution structures in the PDB, we also curated computationally-designed homo-oligomers with structures that have been experimentally validated with either X-ray crystallography, or other lower-resolution methods (Appendix Table 1.1). Many of these *de novo* homo-oligomers have been described [4,5]; however, some are unpublished as of yet. While it is difficult to solve a high resolution crystal structure for all of these homo-oligomers, we have adequate biochemical and structural characterization suggesting that they are forming the designed homo-oligomeric structure. For those *de novo* designed proteins for which we do not have high-resolution crystal structures, I used the computational models of the designs. For example, we require that the *de novo* designed proteins are found to be the correct oligomeric molecular weight by MALS measurements and have

collected SAXS traces that match calculated traces. In some cases, we also have EM images or class averages that support the assembly of the oligomers into the correct architecture.

1.2.3 *Cyclic fusion protein design*

Researchers demonstrated the utility of computational protein design in creating homo-oligomers with cyclic symmetries by both engineering symmetric coiled-coil helical bundle (HB) proteins [4] as well as symmetrically docking and designing monomeric *de novo* helical repeat (DHR) proteins [29]. To further expand the database of cyclic homo-oligomers, I married these two prior successes by docking monomeric DHRs onto the HB homo-oligomers to generate computationally designed genetic fusions. The goal of this process was to quickly create new homo-oligomer building blocks by combining highly stable *de novo* proteins that provide different characteristics. The DHRs provided extensibility and diverse topologies while the HBs provided a homo-oligomerization hub.

Here, I describe a high-throughput computational pipeline for the design of new cyclically symmetric protein oligomers made from DHR rods – rigid proteins with tunable twist and curvature – and HB nodes – symmetric protein interfaces, which can then be connected to generate a wide range of nanostructures. Using these rods and nodes, we can generate cyclic assemblies of the rods by rigid fusion to the nodes, which are parametrically generated helical bundles with cyclic symmetries [4]. The helical bundles will act as symmetric nodes to direct the assembly of the rods into cyclic protein homooligomers (Fig. 1.2a). This will be done using a computational design protocol that optimizes the interface sequence between the bundle and repeat protein and adds a short loop fragment between them at the point of genetic fusion (Fig.

1.2b,c). With additional design to extend or shorten the repeat protein rods and to generate terminal interfaces , we will be able to design self-assembling nanostructures of a very wide variety of shapes and sizes (Fig. 1.2d).

Designs with high shape complementarity, low-energy interfaces, and unstrained loops will be experimentally characterized. Genes encoding hexahistidine-tagged designed cyclic homo- oligomers will be transformed into *E. coli*, and the proteins purified by IMAC. The assembly states of the purified proteins will be characterized using size exclusion chromatography (SEC), multi-angle light scattering (MALS), and SAXS. The designs found to adopt the desired oligomerization state will be submitted for crystal screens to hopefully obtain high-resolution structures using X-ray crystallography.

Preliminary SEC-MALS and SAXS results suggest that some of the first round of designs are working (Fig. 1.3). The data suggest that they are forming the correct oligomeric complex and overall structure. These are good candidates for full structural characterization by X-ray crystallography.

1.2.4 *Nanocage design*

As the tetrahedral [7,8] and octahedral [7] nanocage databases are relatively limited, I sought to expand these databases by designing new nanocages that specifically contain building blocks capable of interacting along their cyclic symmetry axes with themselves to form dihedral interfaces. To this extent, I successfully designed a new octahedral nanocage consisting of tetramer and trimer building blocks. Following nanocage docking and design protocols described previously [7,8,9,10] (Fig. 1.4), a bicistronically-expressed design was identified in which a tetramer

(PDB ID: 1e4c) formed an octahedral nanocage by interacting with a trimer (PDB ID: 1wa3). This result was verified by MS, SEC-MALS, dynamic light scattering (DLS), and negative-stain EM (Fig. 1.5a,b). Furthermore, when the two components were expressed and purified separately, they were found to assemble *in vitro* into the same octahedral nanocage structure (Fig.1.5c,d). The experimental SAXS data of the *in vitro* assembled octahedron overlays over the calculated SAXS data from the design model (Fig. 1.5e) indicating that the subunits are assembling when mixed into the designed nanocage architecture. This was the first described two-component octahedral nanocage that readily assembled both *in vivo* and *in vitro*. In that it contains a tetrameric subunit, this opens doors to functionalization of the nanocage for display of tetramic antigens such as neuraminidase, which is a tetramic protein on influenza virus (collaboration with Neil King's group).

1.3 Crystal Design Using Cyclic Protein Building Blocks

Using cyclically symmetric protein oligomers as building blocks in crystal design is an attractive avenue because there are simply more structures of proteins with cyclic symmetries in the PDB than with higher symmetries. Additionally, while some crystal architectures that can be engineered by combining two cyclic oligomers together (Table 1.1) have cycle connectivities of twelve or higher, there was one cyclic-by-cyclic architecture that seemed amenable to crystal design. Combining two different proteins with C3 symmetry together to form a $P2_13$ space group has a low cycle connectivity of ten (Fig. 1.6b).

To pursue $P2_13$_C3-C3, a hierarchical docking software was developed (unpublished data) that aligns the two C3 symmetric proteins on their symmetry axes in the crystal space

group and allows them to sample their degrees of freedom (DOFs) (Fig. 1.6a). The top docked structures were then carried forward for symmetric Rosetta interface design (Fig. 1.6c). The framework for allowing the correct sampling of crystal DOFs was not yet in the Rosetta code base, so we added parameters to the SymDofMover to allow for translational and rotational sampling that did not go against the geometric constraints of the $P2_13$ architecture (Appendix Methods 1.1).

DNA encoding the chosen designs were ordered as bicistronic genes in which the gene for one of the protein building blocks had a hexahistidine affinity tagged added on to it. Proteins were tested for expression and solubility, followed by a pull-down assay in which the two proteins were purified over an immobilized metal affinity column (IMAC) with nickel-coated beads. If the two proteins interact with each other, this is visible on a SDS-PAGE gel of the IMAC eluate. While there were a few promising design candidates for which either both proteins were found in a one-to-one ratio in the insoluble fraction or were both soluble, none exhibited the pull-down phenotype.

Interestingly, one of the individual proteins that was purified by IMAC without it's designed binding partner (Fig. 1.7a), spontaneously crystallized with itself during storage in IMAC elution buffer (Fig. 1.7b). We were able to solve a crystal structure of this construct (Fig. 1.7c). From comparing the off-target interface in the solved one-component crystal structure to the designed protein-protein interface between the two components, it is apparent that the off-target crystal is using some of the designed residues in the interface driving its spontaneous self-assembly. The most conspicuous residue at the off-target interface was a tryptophan that was native to the building block that, in the off-target crystal, flipped to occupy its other rotamer

state, forming favorable hydrophobic contacts as well as a hydrogen bond. This observation that a small number of added surface exposed hydrophobic residues caused this C3 oligomer to spontaneous self-assembly support a hypothesis put forward by Levy and colleagues [31] that proteins, and especially protein oligomers, have evolved to be on the edge of a higher order assembly state.

This observed off-target spontaneously self-assembling one-component crystal highlights the need to integrate protocols into the crystal design scheme that avoid non-specific interactions from occurring. Excitingly, this off-target crystal also proves that it is feasible to achieve these types of protein-based nanomaterials.

1.4 Crystal Design Using Dihedral Protein Building Blocks

Another interesting low cycle connectivity crystal space group was the F222 lattice made of two building blocks with D2 symmetry, which has a cycle connectivity of six to create a bounded nuclei. The same hierarchical docking scheme described above was used; however, because the input building blocks had D2 symmetry, in this case, we had to add more options to the SymDofMover to account for the unique positional variants of objects with D2 symmetry (Appendix Methods 1.1).

Essentially, while in other symmetries, components have a complete rotational DOF, D2 proteins have six unique positions they can occupy. For example, comparing an object with D2 symmetry to one with D3 symmetry, both have C_n symmetry along the Z-axis as well as two two-fold axes along the X-Y plane. In the case of the D3, the C3 axis must always stay constant when aligning it with D3's in the crystal context. However, as a D2 essentially has three

two-folds, and mathematically, any of them can be aligned to the Z-axis while the other two remain on the X-Y plane, at each of these alignments, there are two rotations around the Z-axis that display the D2 in different ways. Notably, this F222 lattice was the only non-cubic lattice I attempted to design. F222 is an orthorhombic space group meaning that its unit cell dimensions related to cell spacing is not equivalent over all three of its dimensionalities. I was able to get by without addressing this, but likely lost some of the DOFs related to the unit cell spacing. This would be a smart addition to the Rosetta symmetry code in the future.

After designing the F222_D2-D2 docks, chosen designs were again ordered as bicistronic constructs in a single plasmid where only one of the proteins contained a hexahistidine affinity tag. Expressed and soluble proteins were analyzed after IMAC by SDS-PAGE for pull-down of the non-His-tagged protein by the His-tagged one. In most cases, both proteins were not expressed solubly. While the insoluble fractions of the lysed cells were analyzed by EM for crystalline assemblies, no promising assemblies were immediately noticeable. A few designs, UWN_0157 and UWN_0163 (Appendix Table 1.2), that showed initial signs of pull-down occurring (Appendix Fig 1.1a,b). , but need to be followed up on in terms of reproducing those results, MS analysis, and further experimental characterization. Notably, UWN_0163 showed some interesting potential for forming some self-assembling higher order complexes (Appendix Fig. 1.1c), but I was not able to collect definitive data on this.

While nothing immediately promising came out of this work, it is recommended that the F222_D2-D2 designs are revisited. The main obstacle of this research was that individual building blocks became insoluble when the hydrophobic interface residues were added to their surfaces. With the design of new D2 oligomers (Chapter 2.2) as well as other *de novo* dihedral

oligomers being designed in the Baker Group (unpublished data), this problem may be avoidable.

1.5 Crystal Design Using Dihedral and Nanocage Protein Building Blocks

Taking advantage of the, at the time recently published, two-component tetrahedral nanocages [7,8], and later also the new designed nanocage (unpublished data, Section 1.2.4) our first thought was to dock those tetrahedral nanocages with homo-oligomers with D3 symmetry from the Protein Data Bank (PDB) (Chapter 1.2.1) to generate an $F4_132$ lattice architecture. Because of the geometric constraints of the lattice, the input building blocks had to be accurately aligned such that their internal symmetry elements matched the desired space group. Symmetry definition files were created that modeled a subsection of the crystal unit cell in which two cages were separated by a dihedral protein.

Similar to the work discussed above, one of the main challenges encountered was the difficulty in being able to solubly purify all of the components of the system. This was particularly difficult for this lattice architecture and its constituent components because I docked a two-component cage to a dihedral oligomers meaning that in this situation, there were three components versus the previous design schemes that had two components. These encountered challenges were part of the inspiration for the hierarchical crystal design approach described in Chapter 2.

1.6 Crystal Design Using Nanocage Protein Building Blocks

Using all of the existing tetrahedral and octahedral nanocages designed by various members of the Baker Lab, some nanocage-only symmetries were pursued. The F23_T-T and $F4_132$_T-D3 space groups were pursued because we have 13 tetrahedral cages [7,8] (unpublished data) designed by various members of the Baker Lab, whereas, at the time, we only had one octahedral cage [7]. In the former, two different cages were docked together in the F23 lattice architecture to see if a decent protein-protein interface could be formed. In the latter, tetrahedral cages were docked against themselves in a $F4_132$ lattice architecture such that they were forming a D3 interface using one of their C3 facets.

The most promising candidate here was the previously described T33-15 nanocage [8], which is a tetrahedral nanocage formed by the co-assembly of two trimeric building blocks (Fig. 1.8b). When T33-15 was first designed, it was noted that the two components of the cage could be solubly expressed separately from one another and then mixed *in vitro* to successfully form the same architecture. The cage was docked against itself (Fig. 1.8a) and a new protein-protein interface was designed (Fig. 1.8c) to promote the self-assembly of cages with each other into the $F4_132$ crystal.

Three dihedral interface constructs were ordered, and one of them, UWN_0222, was able to be expressed solubly. When IMAC eluates of both components, UWN_0222 and UWN_0223, were mixed, crystals visible by light microscopy formed within minutes (Fig. 1.9c). Following crystal formation, I sequentially pelleted and washed the crystals. SDS-PAGE gels of the individual components, the mixture, the wash steps, and the isolated crystals (Appendix Fig. 1.2)

showed that both components were indeed within the assembled crystals. Interestingly, when we further purified their components by SEC and analyzed them by MALS, they were both found to have oligomeric molecular weights matching their cyclic states (Fig. 1.9 a,b) despite the dihedral interface designed in UWN_0222. The crystals that formed were further analyzed by negative-stain EM, and clear packing of the T33-15 was observed (Fig. 1.9d). Because of the ability of these two proteins to assemble when mixed *in vitro*, I was able to run static light scattering experiments to gain preliminary insight into the salt (Appendix Fig. 1.3a) and protein (Appendix Fig. 1.3b) concentration dependence of the spontaneous crystallization process.

A crystal structure of the spontaneously assembling two-component crystals was solved at 2.23 Å resolution. This structure showed that the crystals that had formed were occupying a different space group, H32 (Fig. 1.10b), instead of $F4_132$ (Fig. 1.10a), and that assembly was driven by off-target protein-protein interfaces (Fig. 1.10d) instead of the designed interface (Fig. 1.10c). In response to this unfortunate turn of events, I created a second round of designs based on UWN_0222, but with mutations that I predicted would interfere with the off-target interface observed in the crystal structure, but not with the intended designed interface (Fig. 1.11 a,b,c,d). Most notably, there were two surface-exposed hydrophobic residues on the building-block protein itself that I mutated to polar residues: Ile18Asp and Ala16Lys. There were also two polar residues forming favorable hydrogen bonds in the observed crystal interface, which I mutated to other polar residues incapable of forming those same hydrogen bonds: Gln64Asp and d Arg13Glu. Additionally, I made a few minor mutations at the interface to alter its strength. All of these UWN_0222 variants expressed solubly and I was able to purify them robustly (Fig. 1.11 e,f,g,h). When mixed with UWN_0223, none of the variants spontaneously crystallized, which

suggests that I successfully knocked out the off-target interactions. This result also suggests that the designed interface is unable to drive spontaneous crystallization, which was my initial goal. However, promisingly, vapor diffusion crystallization screens of the new mixtures (Fig. 12) showed some promising crystal hits, suggesting that the designed interface may drive crystallization under certain conditions. I am currently conducting further experiments to structurally characterize these hits to determine whether they crystalize via the designed interface. Overall, while this particular design strategy did not yield the expected results, it did lend important insight into the research project at large: namely, it revealed the possibility of off-target crystal formation, and the importance of devising design strategies that both promote the desired assembly and avoid off-target assembly. This lesson inspired the hierarchical approach described in Chapter 2.

1.7 Discussion and Future Directions of Crystal Design Using Pre-Existing Protein Building Blocks

This chapter highlights a bulk of my dissertation work on using pre-existing protein building blocks to design spontaneously assembling three-dimensional crystal lattices. This work involved curating large libraries of protein building blocks and then systematically exploring the ability of individual proteins and combinations of proteins to satisfy geometric constraints required for crystal formation. To perform this latter step, I developed a new computational framework in Rosetta that is highly generalizable for designing defined assemblies with any protein building block. Following computational design, I experimentally tested a large number of designs in the lab. Notably, a few examples of spontaneous crystal formation were observed

and could be useful as materials for chaperoning crystallization [32] of small peptides or patterning metal nanoparticles [33, 34, 35]. This work demonstrated that it is indeed possible to form spontaneously self-assembling one- and multi-component protein crystals. However, to control the final architecture of the protein crystals, several modeling and strategic difficulties need to be resolved.

Throughout this work of designing self-assembling protein crystals with pre-existing proteins, it became evident that challenges existed in: i) ensuring the stability of input building blocks with regards to their tolerance for the introduction of surface exposed hydrophobic amino acids and ii) finding pairs of building blocks that would dock to form reasonable protein-protein interfaces given the stringent geometric constraints. Additionally, as highlighted by the examples of self-crystallization of building blocks (Fig. 1.7) as well as off-target crystal lattice formation by mixtures of building blocks (Fig. 1.9), it is crucial to prevent off-target intra- and inter-protein interactions. To address these challenges, I switched focus from using pre-existing building blocks, either naturally-occurring or *de novo* designed proteins, to custom-making new symmetric building blocks that are more stable because of their constituent monomers and are more likely to form favorable protein-protein interfaces because of the hierarchical design strategy that was employed.

Figures

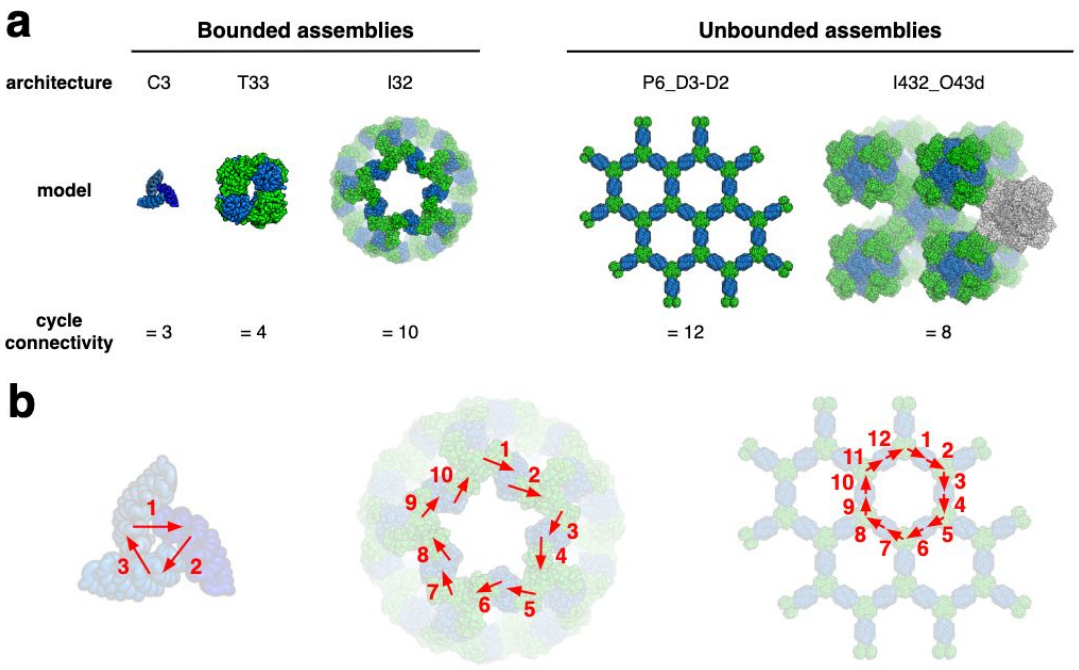


Figure 1.1. The cycle connectivities for various bounded and unbounded assemblies. Cycle connectivity is described as the number of unique subunit-subunit interactions required to create a bounded nuclei for an architecture. (A) Models of and cycle connectivities for various architectures are displayed with the constituent subunits colored as either shades of the same color for one-component architectures and as different colors for two-component architectures. For the I432_O43d architecture, the unit cell of the crystal space group is shown in colors while an extra polyhedron is shown in gray as it is needed to calculate the cycle connectivity of the system. (B) Examples of how the cycle connectivity subunit-subunit interactions are counted for C3, I32, and P6_D3-D2 architectures are detailed to show how bounded nuclei are created for the assemblies.

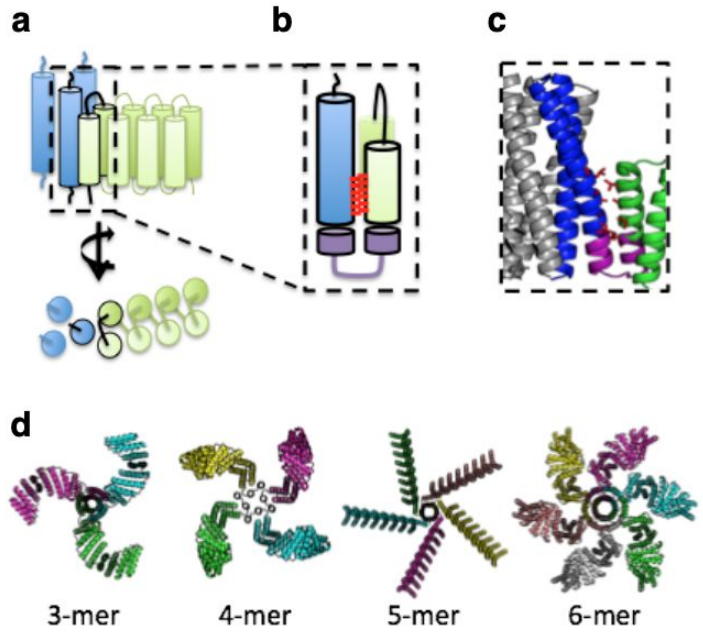


Figure 1.2. Cyclic fusion protein design. (a) Cartoons of a blue helical bundle node and green rod docked together and (b) a new protein core (red) is designed as well as a loop (purple) to genetically fuse the two pieces together. (c) An example of a design and (d) the many homo-oligomeric architectures that can be formed.

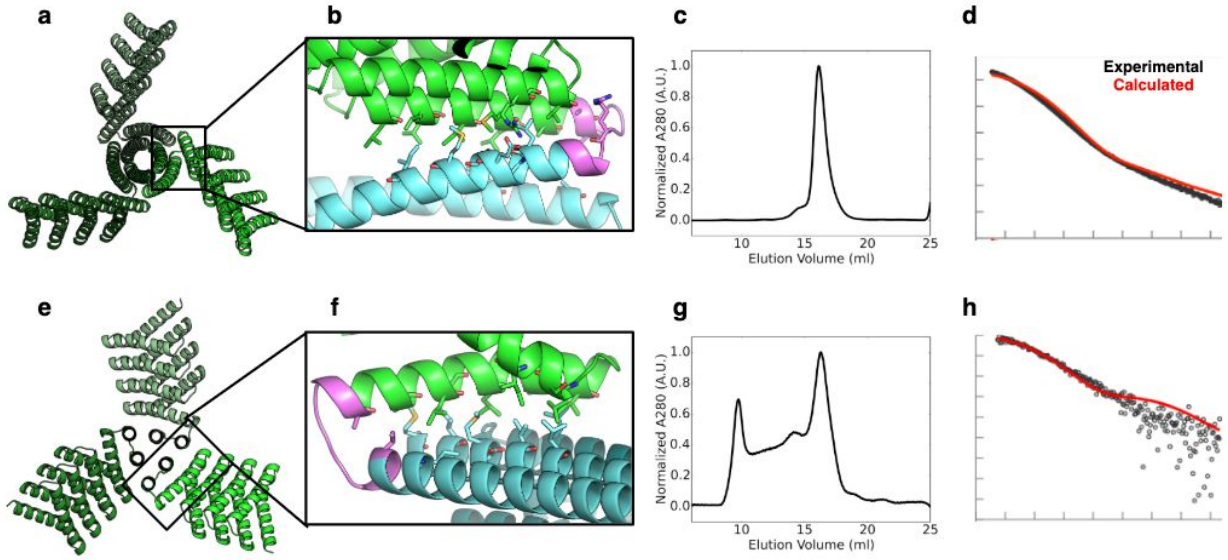


Figure 1.3. Experimental characterization of designed cyclic fusion proteins. Designed homo-oligomeric models (a, e), designed interfaces (b, f), size exclusion chromatography traces (c, g), and small-angle X-ray scattering data (d, h) where the experimental data are shown in black and the calculated data are shown in red for the two working trimeric protein complexes.

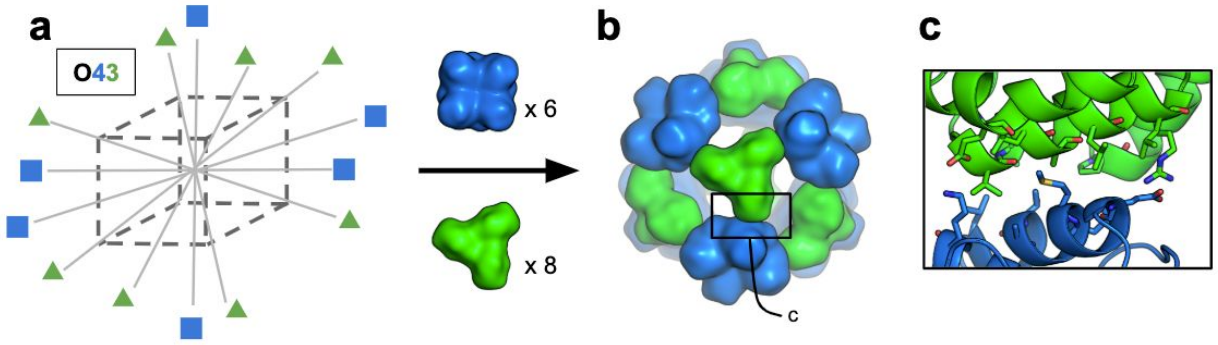


Figure 1.4. Octahedral nanocage design. (a) The geometric scheme for an octahedral protein nanocage made of tetramers (blue squares) and trimers (green triangles). For this architecture six tetramers and eight trimers can be docked along the depicted symmetry axes to form an octahedral nanocage (b), for which a protein-protein interface can be designed using Rosetta (c).

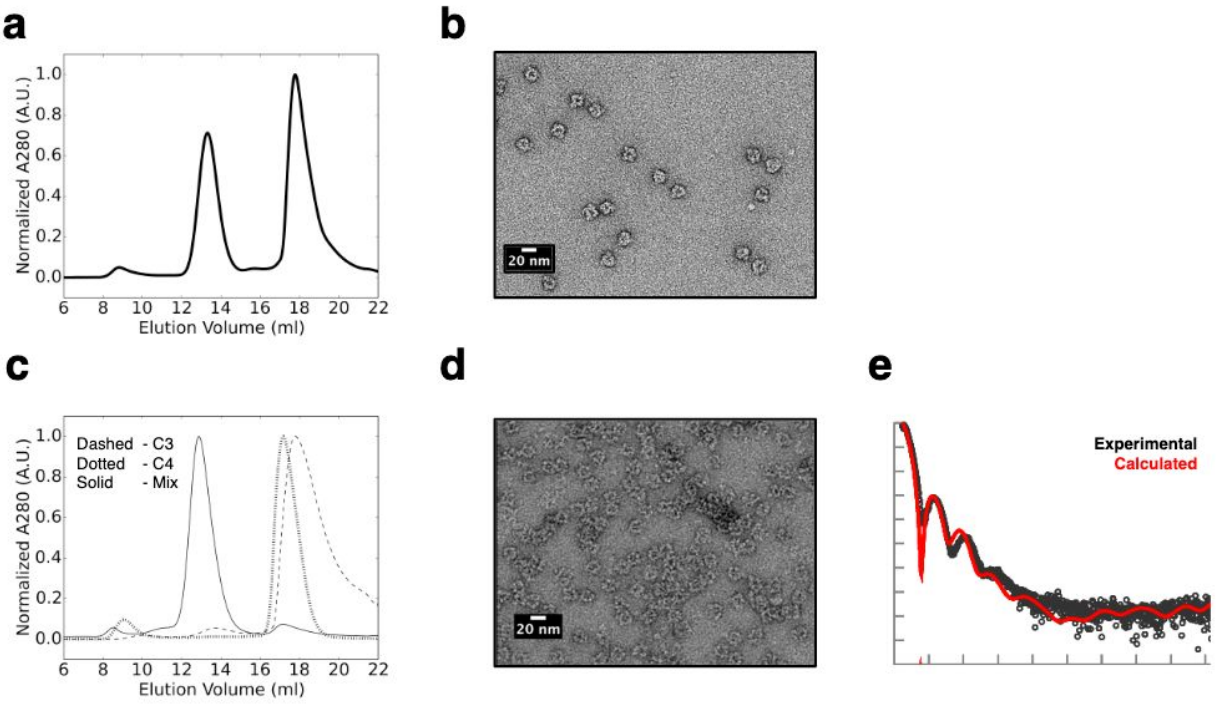


Figure 1.5. Experimental characterization of octahedral nanocage, O43-38. (a) Size exclusion chromatography data for the two-component nanocage in which both building blocks were co-expressed. (b) The peak at the predicted elution volume of ~13mL was visualized by negative-stain electron microscopy. (c) The two building blocks are able to be expressed solubly and purified separately (dashed and dotted lines) and they assemble when mixed *in vitro* (solid line). (d) The *in vitro* mixture was visualized by negative-stain electron microscopy, (e) and also characterized using small-angle X-ray scattering.

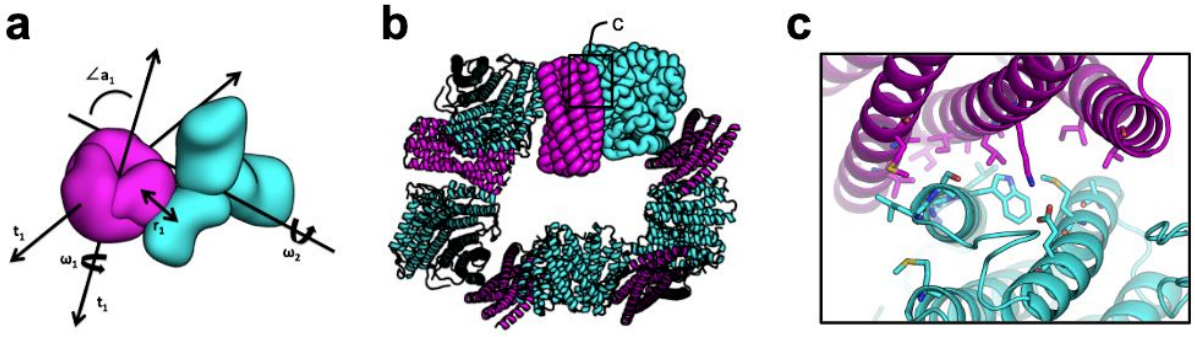


Figure 1.6. Scheme for combining two trimeric proteins together to form a $P2_13$ lattice. (a) The five degrees of freedom (DOFs) that are available when docking two trimeric proteins together to form a $P2_13$ lattice: t_1, t_2, w_1, w_2, and r_1. The a_1 angle needs to be kept constant while sampling these DOFs. (b) An example docked architecture in which the cycle connectivity of ten can be visualized. (c) An example of a Rosetta designed protein-protein interface between the two building block proteins.

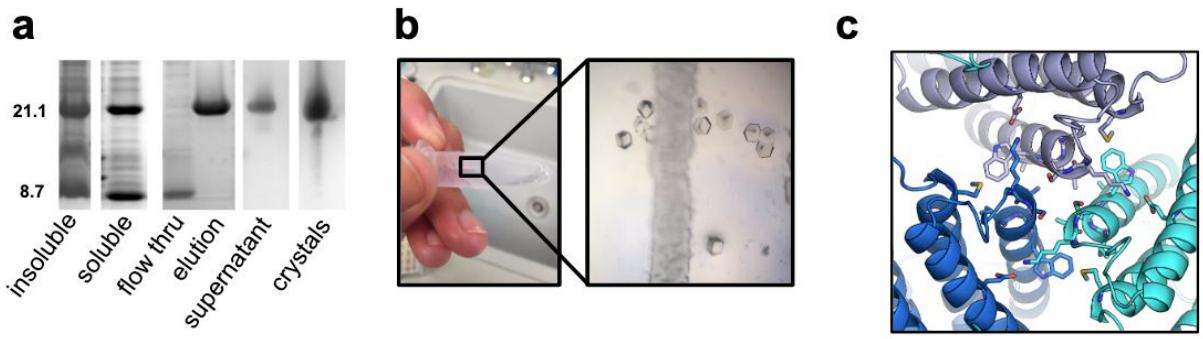


Figure 1.7. Experimental characterization of a single component spontaneously self-assembling into an off-target crystal. (a) SDS-PAGE gel indicating the lack of pull-down of the smaller molecular weight non-affinity-tagged protein by the larger molecular weight affinity-tagged protein. (b) The elution fraction of the single component on its own spontaneously self-assembled into cubic crystal visible by light microscopy. (c) An X-ray crystallography structure was solved of this off-target assembly. Seen is the new protein-protein interface the single-component formed with other copies of itself to spontaneously self-assemble.

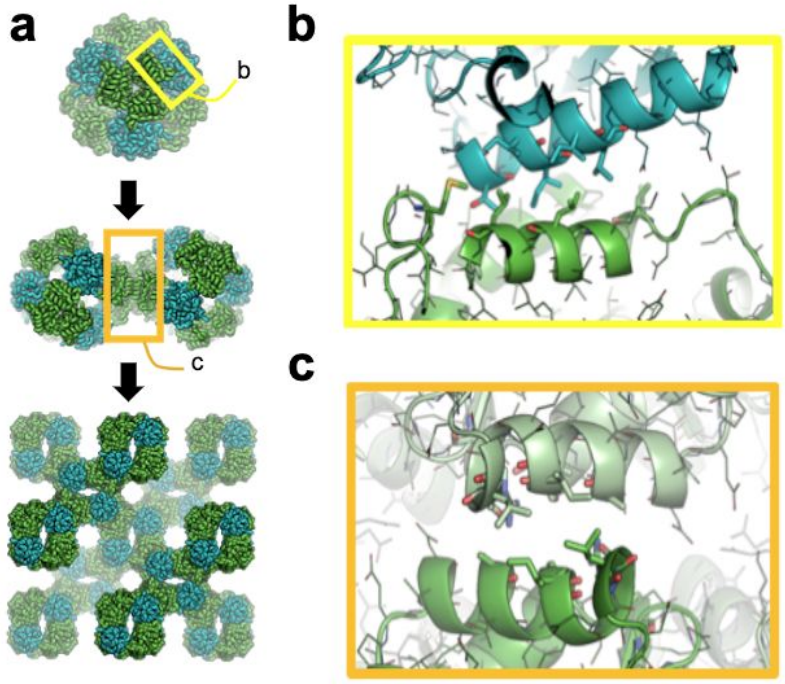


Figure 1.8. Designing a tetrahedron to interact with itself to form a F4$_1$32 lattice. (a) A pre-existing T33 nanocage is docked to form a dihedral interface along one of its three-fold axes to propagate into an F4$_1$32 lattice. (b) The original nanocage protein-protein interface and the (c) designed dihedral crystal interface are shown.

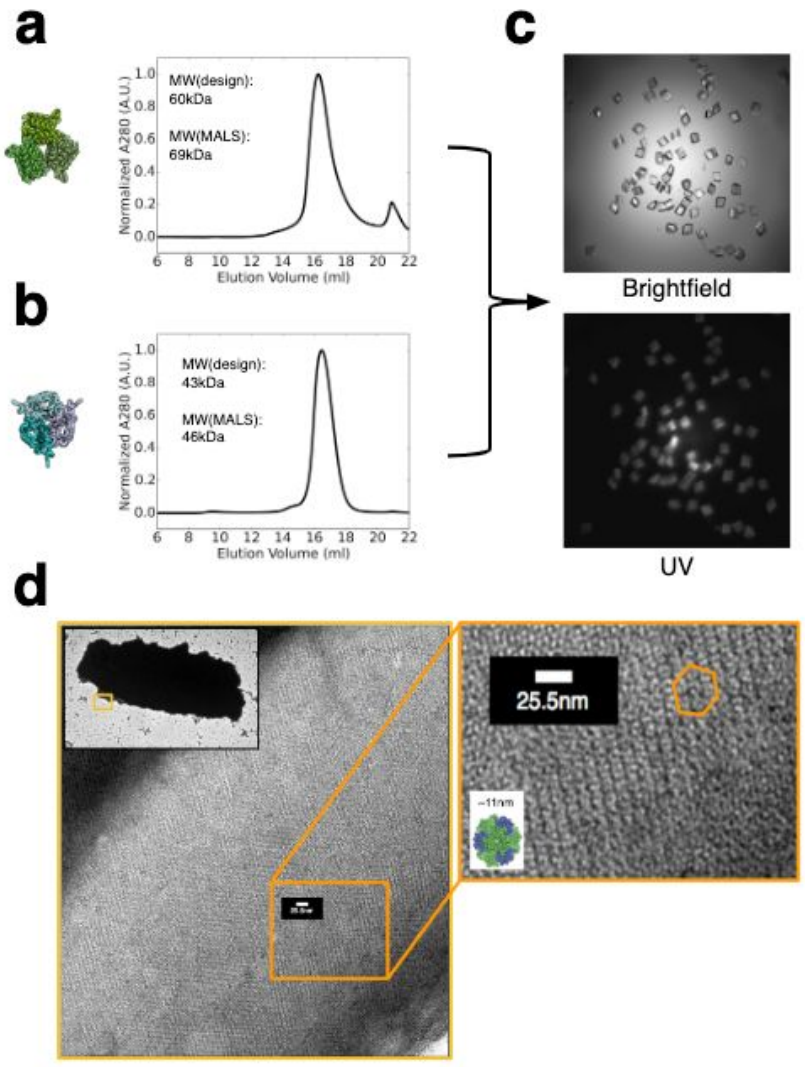


Figure 1.9. Purification and *in vitro* assembly of UWN_0222 and UWN_0223 when mixed. SEC traces and MALS data for (a) UWN_0223 and (b) UWN_0222. When mixed, these samples readily assembled into crystals as visualized by (c) both brightfield and UV light microscopy. (d) Negative-stain EM micrographs of the crystals with sequential magnifications of the field of view.

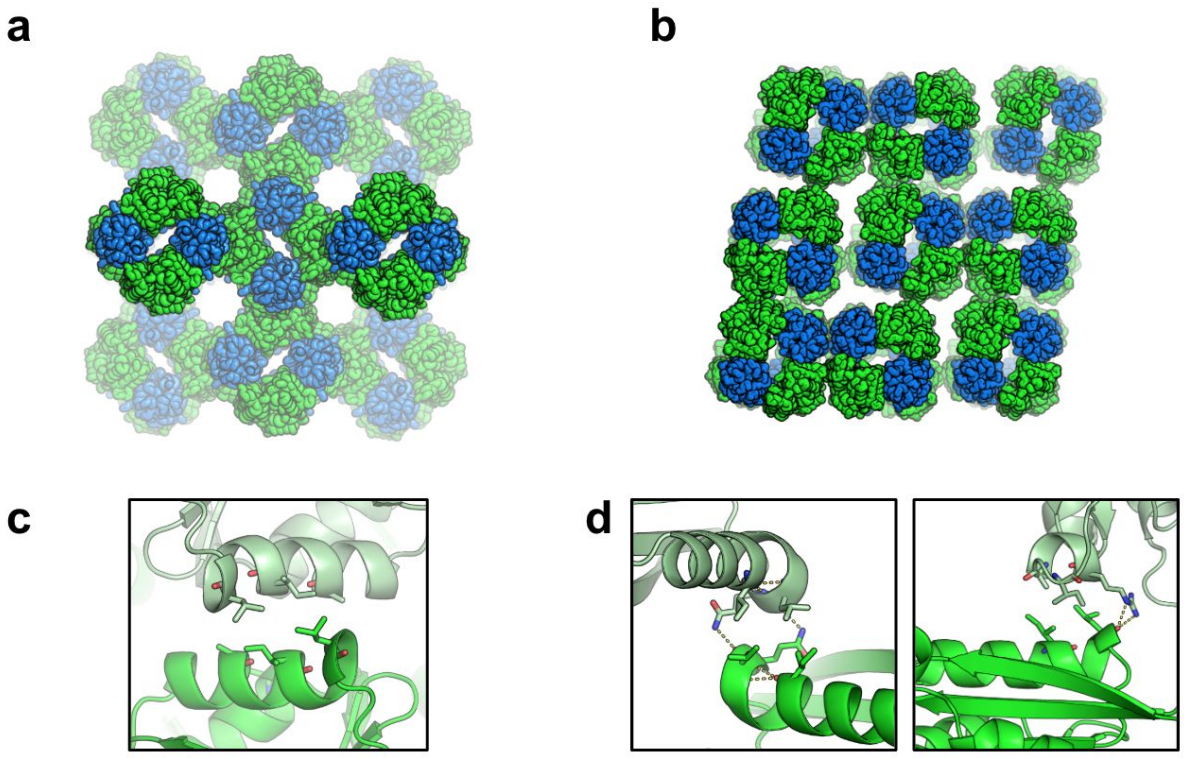


Figure 1.10. The observed crystal packing of the UWN_0222 and UWN_0223 mixture. (a) The designed $F4_132$ crystal lattice packing and (c) the designed dihedral interface that should drive the self-assembly process. (b) The observed H32 crystal lattice packing and (d) the two off-target protein-protein interfaces that ended up driving the self-assembly process.

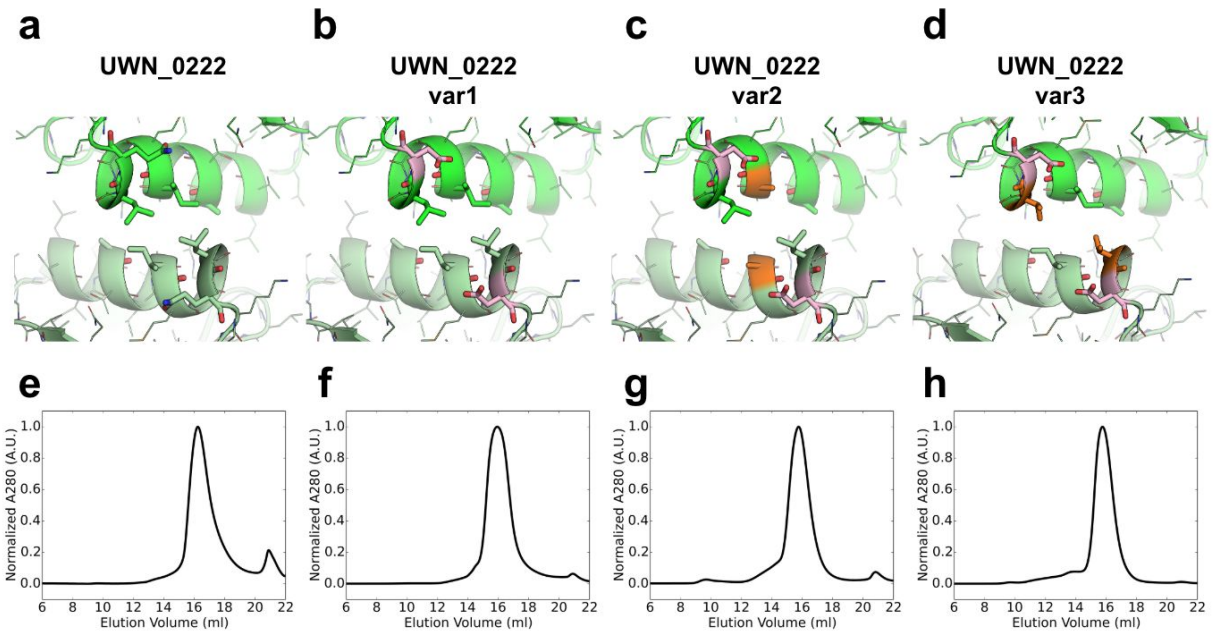


Figure 1.11. The dihedral interface variants and their purification. (a) Models of the original designed dihedral interface as well as (b, c, d) three negative design interface variants. In the images, sticks represent the designed interface residues, and pink and orange sticks represent mutations made at the interface. The mutations for var1 (b) are Gln64Asp (pink), Ile18Asp, Ala16Lys, and Arg13Glu. The mutations for var2 (c) are the same as var1 plus Ile59Ala (orange). The mutations for var3 (d) are the same as var1 plus Leu63Val (orange). (e, f, g, h) SEC traces of all of the variants show that they all elute from the SEC column at the elution volume, indicating that they are all forming a cyclic variants of the design.

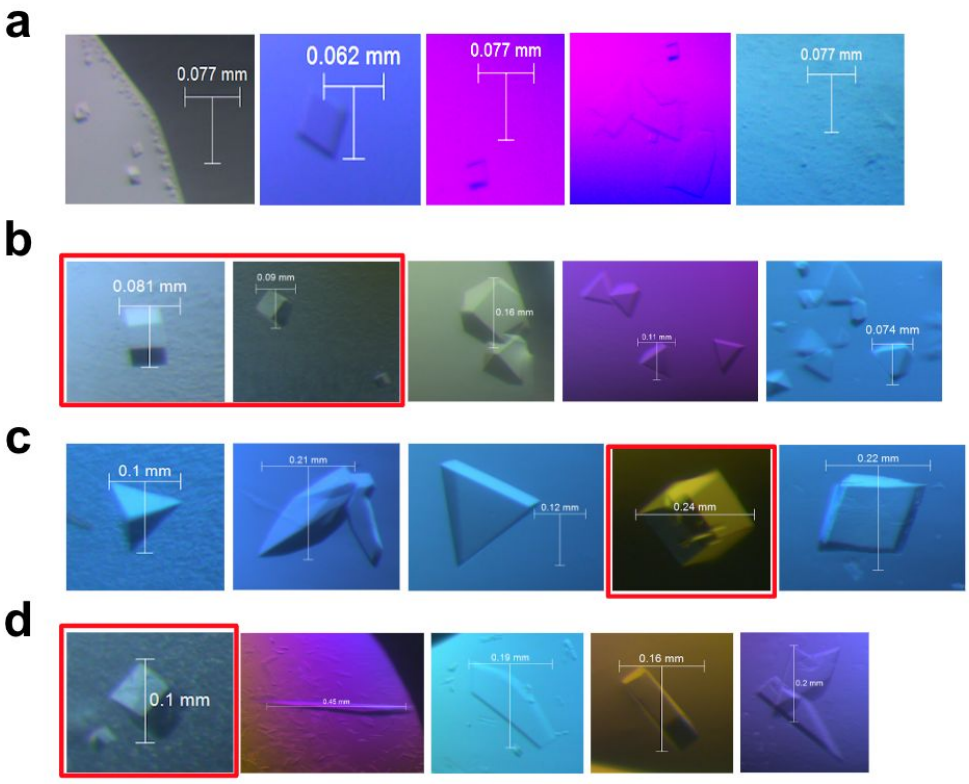


Figure 1.12. Selected crystals from crystallization screens of UWN_0223 mixed with UWN_0222 and its weaker interface variants. Vapor diffusion crystal screens of UWN_0223 plus (a) UWN_0222, (b) UWN_0222 var1, (c) UWN_0222 var 2, and (d) UWN_0222 var3. Images boxed in red look cubic and indicate the most promising positions to follow up on.

Tables

Component 2 \ Component 1	C3	C4	D2	D3	D4	D6	T	O
C2	$I2_13$ $P4_132$	I432	$I4_122$ $P6_222$ I432 $I4_132$	R32 $P6_322$ $F4_132$ $I4_132$ I432 $P4_132$	I422 P432 I432	P622	P23 F23 $F4_132$	P432 F432 I432
C3	**$P2_13$**	F432	P23 F432 $I4_132$	$P4_132$	P432		F23	F432
C4		P432	I432 F432	I432	P432		F432	P432
D2			**F222** $P4_222$ $P6_222$ $P4_232$ $I4_132$	P622 $P4_232$ $I4_132$	P422 I422 I432	P622	**P23** F432 $P4_232$	**F432** I432
D3				$P3_12$ $P6_122$ $P4_232$ $P_{41}32$	I432	P622	**$F4_132$**	**I432**
D4					P422 P432			**P432**
T							**F23**	F432
O								P432 F432

Table 1.1. Theoretical two-component crystal design space. The x-axis (blue) and the y-axis (green) show the internal symmetries of component 1 and component 2, respectively. Boxes at intersections of the two components indicate the possible three-dimensional crystal lattice architectures made possible by combining the two symmetric architectures together in some geometry consistent with the crystal space group. Boxes that are black are redundant and boxes in gray do not have solutions. Space group architectures that are bolded and highlighted in yellow are ones that are described in this thesis. This table is adapted from Yeates et al. 2016. Curr Op Struct Biol.

Table 1.2. Data collection and refinement statistics for C3_P2$_1$3 crystal.

	C3C3_13_NiE
Data Collection	
Space group	*P2$_1$3*
Cell parameters, a,b,c (Å) α, β, γ (°)	91.47 91.47 91.47 90.00 90.00 90.00
Resolution (highest shell) (Å)	50.00 - 1.98 (2.01 - 1.98)
No. of unique reflections	17946
Redundancy (last shell)	17.9 (18.1)
Completeness (last shell) (%)	99.8 (100.0)
I/σ(I) (last shell)	28.35 (1.23)
R_{merge} (last shell)	0.89 (2.48)
$CC_{1/2}$	0.89
CC*	0.92
Refinement Statistics	
Resolution range (Å)	40.91 - 1.98
No. of reflections used	17819
No. of protein atoms	1327
No. of water molecules	70
R_{work}	0.18

R_{free}	0.20
rmsd bond lengths (Å)	0.003
rmsd bond angles (°)	0.51
Ramachandran Favored/allowed Outlier (%)	 100.00/0.00 00.00
Average $B_{factors}$ (Å^2) protein water	 63.11 70.14

Table 3. Data collection and refinement statistics for T33d-15 crystal.

	T33d-15
Data Collection	
Space group	R 3 2 :H
Cell parameters, a,b,c (Å) α, β, γ (°)	147.90 147.90 466.68 90.00 90.00 120.00
Resolution (highest shell) (Å)	123.52 - 2.23 (2.31 - 2.23)
No. of unique reflections	96094
Redundancy (last shell)	20.4 (18.9)
Completeness (last shell) (%)	99.26 (92.86)
I/σ(I) (last shell)	19.67 (0.72)
R_{merge} (last shell)	0.115 (4.233)
$CC_{1/2}$	0.97 (0.429)
CC*	0.99 (0.775)
Refinement Statistics	
Resolution range (Å)	123.52 – 2.23
No. of reflections used	95400
No. of protein atoms	8930
No. of water molecules	275
R_{work}	0.18
R_{free}	0.22

rmsd bond lengths (Å)	0.003
rmsd bond angles (°)	0.56
Ramachandran Favored/allowed Outlier (%)	 98.20/1.80 00.00
Average $B_{factors}$ ($Å^2$) protein water	 79.78 70.72

Chapter 2. Crystal Design Using Custom-Made Protein Building Blocks

2.1 Custom-Engineering Protein Building Blocks For Crystal Design

Most of our nanomaterials design work to date, at the time, have employed naturally occurring scaffolds as building blocks. As demonstrated by previous successes [7,8,9,10], this approach is quite powerful in generating new protein-based materials. Nonetheless, repurposing proteins that evolved for functions unrelated to that intended for the designed material is less desirable than using custom designed building blocks. For example, a major source of failure in previous efforts has been insoluble protein expression. It is well known that naturally occurring proteins are on the edge of stability [31], and that the many mutations required to introduce the interfaces directing self-assembly can unintentionally destabilize the proteins or cause them to non-specifically aggregate. Custom designed building blocks can be much more stable than naturally occurring proteins [5] and, moreover, can be built with shapes [29] optimal for nanomaterial construction.

Computational protein design has come a long way since the design of the first *de novo* protein, Top7, in 2003. Researchers have optimized the Rosetta energy function [36] and improved design algorithms, allowing the production of a variety of novel protein folds as well as cyclic oligomers made from these *de novo* monomers. Additionally, the field has progressed greatly in terms of creating frameworks for the design of biomaterials such as nanocages with diverse 0D point symmetries [4,5,6,7,8,9,10], 1D helical filaments [11], and 2D sheets [12]. However, designing proteins

with higher symmetries and complexities requires addressing some unsolved technical and experimental challenges. Notably, when designing 0D symmetry groups such as cyclic oligomers and nanocages, there are still several degrees of freedom to vary in the system. However, as designs increase in symmetry, the degrees of freedom become more and more constrained, and it becomes further necessary to have building blocks that fit "just right" to end up with a successfully assembling material. To address these challenges, I have developed a computational and experimental workflow for the custom design and characterization of new cyclic and dihedral oligomer building blocks made from already-experimentally-verified *de novo* proteins. This allows for the control of both the stability of the input protein building blocks as well as their overall topologies.

I sought to design self-assembling protein crystals that consist of two protein building blocks. The overall goal is to design constructs similar to that which was described in Chapter 1.6 in that the desired crystal architectures are easily imagined as polyhedral nanocages stacked with themselves along one of their symmetry axes. The hierarchical trick is to design a cyclic oligomer to form a dihedral oligomer first, thus verifying, at the beginning, the robustness of the dihedral interface in the final crystalline assembly (Chapter 2.2). Next, that new dihedral oligomer is docked as one of the building blocks into a polyhedral nanocage (Chapter 2.3), similar to as described in Chapter 1.2.4. Since the dihedral interface can be reverted back to its cyclic-forming state, the nanocages can be characterized as soluble 0D assemblies. If the nanocage interface is validated through experimental and structural techniques, then the dihedra interface can be reintroduced resulting in the crystal-forming construct (Chapter 2.4).

2.2 Design Of Dihedral Protein Complexes From Cyclic Oligomers

The first step in this hierarchical crystal design process is the design of dihedral oligomers from pre-existing cyclic oligomers. To design this first component, I focused on input proteins with C2, C3, and C4 symmetries; and designed them to assemble into D2, D3, and D4 symmetries, respectively. The motivation for this project was multi-faceted: (1) we wanted to expand the database of *de novo* dihedral oligomers that could be used in downstream design processes as a building block, (2) we chose to use *de novo* cyclic oligomers as our input proteins as they have already been shown to have high thermal and chemical stability, and therefore, potentially be more mutation tolerant, (3) the *de novo* cyclic oligomers we chose have the ability to form dihedral interfaces and afterwards still have remaining exposed secondary structure elements useful for the creation of a second protein-protein interface in downstream design, and (4) it is very useful to have confidence in the structures in both the cyclic and dihedral states to allow for easy characterization of soluble proteins in downstream design.

2.2.1 *Turning a highly stable naturally-occurring cyclically symmetric protein complex into a dihedral complex*

One of the main challenges of protein materials design is to design stable building blocks. That is because mutations that promote self-assembly can often destabilize the protein. Because we are using symmetric proteins as our input structures, they already have a protein-protein interface driving the homo-oligomeric self-assembly of the protein into its base cyclic state. On top of that, our design scheme introduces at least one, if not two, more protein-protein interfaces,

thus adding exposed hydrophobic residues on the surface of the protein. Undoubtedly, this can interfere with both the folding of the monomer as well as the solubility of the protein in solution.

To address these challenges, I chose to design an additional dihedral interface onto a naturally-occurring cyclic oligomer. This starting homo-trimer (PDB ID: 1wa3) [37] was chosen because it is naturally highly stable -- it is derived from a thermophilic bacterium -- and it has been amenable to design into multiple nanocages in the past. The tilted TIM-barrel oligomeric architecture of the protein is ideal for this purpose because it has exposed helices both parallel and perpendicular to its cyclic symmetry axis, meaning that it is reasonable to design both a dihedral and nanocage interface without clashes.

To create additional interfaces with 1wa3, I first docked the 1wa3 trimer into D3 architectures to identify potential interfaces with high shape complementarity (Fig. 2.1a). Next, I used Rosetta to re-design the amino-acid sequence on either side of the interface to promote binding (Fig. 2.1a). Of the five structures that I ordered and tested, four formed the expected dihedral architectures as confirmed by SEC-MALS (Fig. 2.1b), while one remained in its cyclic state. We were able to solve crystal structures for two of these cyclic-to-dihedral architectures (Fig. 2.2). Excitingly, I observed accurate matching of the experimental structures to the design models.

The high success rate of this project demonstrates the advantage of using highly stable proteins as the building blocks for further design purposes. This project inspired the following description (Chapter 2.3.2) of utilizing an analogous cyclic-to-dihedral design framework on stable *de novo* cyclic oligomers with topologies amenable to both dihedral and nanocage design.

2.2.2 *Turning de novo cyclically symmetric protein complexes into dihedral complexes*

Because of the high success of turning the cyclic 1wa3 oligomer into a dihedral oligomer (Chapter 2.2.1), I sought to apply the methods I developed to *de novo* cyclic oligomers to further expand our database of oligomers that through mutations could be controlled to be in either a stable cyclic or stable dihedral state. De novo cyclic oligomers were curated from the Baker lab(unpublished data) (Fig. 2.3a), and particular focus was put on those made of *de novo* helical repeat (DHR) [5, 29] proteins and other alpha helical motifs (unpublished data). To expand the database of input cyclic oligomers, I trimmed each oligomer one helix at a time (Fig. 2.3b), and treated the trimmed PDBs as their own unique input structures. Depending on the architecture of the cyclic oligomers, they could be trimmed on the N- and/or C- terminus. Docking was performed by using the tcdock protocol [6], and the D22, D32, and D42 architectures with an empty PDB input protein in place of the dimer (second component) of the architecture (Fig. 2.3c). At the top 100 docked positions, Rosetta design was performed (Fig. 2.3d) as previously described [10]. I curated the resulting design pool by removing designs with buried unsatisfied H-bond donor or acceptor atoms, designs with large clashes, and designs that contained protein-protein interfaces that primarily consisted of interactions along loops, as opposed to along secondary structure elements,. Afterwards, I removed designs that, by eye, were highly redundant.

Next, I experimentally characterized several designs in the lab. To do so, I first ordered DNA encoding the dihedral genes with an N- or C-terminal hexahistidine affinity tag. As a first

high-throughput screen, all the ordered designs were expressed in 2ml of culture, lysed, and purified in a 96-well plate format. IMAC-purified protein was eluted in 210ul and concentrated down to 50ul, filtered with an 0.22um filter, and then measured for absorbance at 280nm. Corrected concentrations (uM) were calculated for each design, and these samples were characterized by native mass spectrometry [6] in collaboration with the Wysocki group. The observed molecular weights from the native mass spectrometry experiment were evaluated for whether the dihedral molecular weights were observed, as opposed to the cyclic molecular weight or that of another species. Those designs in which the dihedral molecular weight was observed were scaled up for further biochemical characterization (Fig. 2.4). These designs were grown in 0.5 liter cultures, and the IMAC-purified proteins were then applied to the SEC for further purification. Size exclusion chromatograms indicated the monodispersity of the samples and also suggested correlation of the bulk protein with the dihedral molecular weight or the cyclic molecular weight of the designs. The fractions of the major peaks of the SEC runs were collected and submitted for SAXS data collection (Fig. 2.4). All designs for which the SAXS experimental data matched the calculated data were screened for structure determination by X-ray crystallography. Many of these structures have crystallized in screens, and one of the dihedrals has diffracted to reasonable resolutions. Excitingly, I observed good agreement of the crystal structure of the dihedrals to its design model (Fig. 2.5). Although I do not yet have crystal structures for the other dihedrals, there is good agreement between the designed structure and other sources of experimental data (nMS, SEC-MALS, and SAXS). This level of agreement is on par with that seen for the crystallized structures, supporting the hypothesis that these structures are accurate as well.

2.3 Computational Design of Geometrically-Constrained Nanocages

The end goal of this project is to design self-assembling protein crystals. Above, I described my approach for designing dihedral oligomers. In this section, I will describe how I incorporated those dihedral oligomers as a building block of a geometrically-constrained nanocage. I designed these nanocages to not overlap with the dihedral interface and so that the dihedral interface would be exposed to solvent in a geometry that would allow the nanocage to self-assemble into a three-dimensional crystal. Before generating the crystal-forming construct, I engineered a mutant version of the dihedral oligomer with mutations that knock out the dihedral interface, allowing me to verify the nanocage interface by standard experimental and structural techniques, which are not amenable to full three-dimensional crystals. I chose to focus on dihedral oligomers containing polyhedral centers within the crystal lattice because those architectures tend to have a lower cycle connectivity, or number of unique protein-protein interactions required to close the system, on par with some of the more porous polyhedra themselves. Inspired by previous work illustrating the use of Rosetta in the computational design of self-assembling polyhedral protein complexes [7,8,9,10], we were able to custom-make polyhedra with geometries that allow them to propagate into crystal lattices with the introduction of the pre-designed dihedral interface.

My approach for designing nanocages involved two main steps: docking individual components to find interfaces with high shape complementarity and then designing amino-acid sequences that promote binding at the interface. I performed these steps as described above, with a few changes. Dihedral subunits were used as input building blocks for one of the components

in the two-component nanocage design process. The dihedral subunits were pre-oriented with respect to their internal symmetries and the final desired crystal lattice context. I did so by rotating all designs around the z-axis to line up the 2-folds of the dihedral subunits to be coincident with 2-folds in the final desired crystal lattice context. Finally, I reverted the amino-acid sequence of the dihedral oligomer to its cyclic state, and then re-aligned its center of mass in the cyclic context (Fig. 2.6c). These dihedral-to-cyclic oligomers are then docked pairwise against a database of *de novo* cyclic oligomers into various polyhedral nanocage architectures that can fit into different crystal contexts.

During two-component nanocage docking, there are typically four degrees of freedom (DOFs) corresponding to the rotational and translational movements for each of the building blocks. Because the orientation of the dihedral-to-cyclic component needs to be constrained to fit the final desired crystal context, one DOF -- the rotation of the dihedral-to-cyclic component -- is removed from the process (Fig. 2.6a). Still, with three DOFs remaining, it is possible to find optimal docking configurations (Fig.2.6b). After docking all pairwise components, I used Rosetta to design amino-acid sequences to promote binding at each interface. I did this for the top 100 docked configurations per pairwise combination per architecture of dihedral-to-cyclic component plus *de novo* cyclic oligomer. Designs passing automated and manual filtering based on Rosetta metrics had genes for their expression in *Escherichia coli* ordered, and results are described below.

One octahedral nanocages , UWN_0453 (Fig. 2.7a), showed stoichiometric pull-down of its non-affinity-tagged building block by its affinity-tagged building block (Fig. 2.7b). Additionally, it migrated as a higher band on a Native-PAGE gel (Fig. 2.7c) indicating that the

purified protein sample is forming a larger assembly state as shown is characteristic of components being designed into nanocages [7,8,9,10]. UWN_0453 was purified by SEC and MALS confirmed that the monodisperse peak contained assemblies at the same molecular weight of the designed structure (Fig. 2.7d). This sample was applied to EM grids and visualized by negative-stain EM (Fig. 2.7e). Particles that look like the design model were observed. Finally, I submitted this sample to SAXS to collect more structural information on the nanocage. Further confirming the accuracy of this design, the collected SAXS data overlays very well with the calculated trace that was generated using the UWN_0453 design model (Fig. 2.7f).

2.4 Crystal Construct Generation and Structural Characterization

From the above work, I identified a designed nanocage that successfully assembled when tested in the lab (Fig. 2.7) and was constructed with a component that can be easily modified to a dihedral-forming state (Fig. 2.2). My next goal was to use these designs to generate nanocages that assemble into higher-order 3D lattices. To do so, I sought to modify the building blocks of these nanocages to generate their corresponding crystal construct. Because of the strategies employed, this is the easy process of combining the dihedral and nanocage interface mutations onto the same genetic construct. This process of combining exposed hydrophobic mutations on the surface of a protein can disrupt the folding and stability of the building block, which we observed. While the cyclic version of the trimer, 1wa3, with the cage interface and the dihedral version of 1wa3 were both highly soluble when expressed independently, the protein became very insoluble when both the cage- and dihedral-forming interfaces were combined. Additionally, the tetrameric component, C4_tpr-pm3 [5] , which takes on no additional mutations

between the cage and crystal construct has low solubility when expressed on its own versus when it is co-expressed with its cage-forming trimeric partner.

To troubleshoot these experimental difficulties, common chemical solubilization additives to the purification buffers were tested for solubilizing both the crystal-construct D3 dihedral and tetramer. In the end, the tetramer was able to be purified in high yields with the addition of 0.75% CHAPS into all solutions used in the purification process (Fig. 2.8 a,f). I also made a mutational variant of the tetramer to reduce exposed hydrophobics on its surface that were unrelated to the nanocage interface. This variant, A-C4_3mut, was still easiest to purify in high yields with 0.75% CHAPS buffer solutions (Fig. 2.8 b,g). The dihedral, B-D3, could be refolded and solubilized with urea, and then purified (Fig. 2.8 c,h); however, I could only isolate low yields of the protein. To potentially help with solubilization as well as alter the kinetics of crystal formation, three weaker dihedral interface variants were ordered and tested. Two of them expressed solubly and were able to be purified in high yields in normal buffer conditions (Fig. 2.8 d, e, i, j).

Mixtures of the CHAPS-solubilized tetramer were made with the original D3 dihedral in 6M urea and the two weaker interface variants of the D3 dihedral. These mixtures were analyzed by negative-stain electron microscopy. From the resulting images, it was clear that nanocage-like particles were forming for all of the mixtures. It was also clear that many of the cages were clustering. This was most apparent for the mixture between A-C4 and B-D3_weak2 (Fig. 2.8k). There were observed particles that seemed to be the UWN_0453 nanocage with dihedral turrets sticking off of its three-fold axes. A few particles were chosen that closely resembled views of

the nanocage when looking down the two-fold (Fig. 2.8l), four-fold (Fig. 2.8m), or three-fold (Fig. 2.8n) axis of the nanocage.

While I have not yet observed crystal formation, these observations are incredibly promising. I have successfully designed a dihedral oligomer, D3_1wa3-44 (Chapter 2.2.2) and integrated its cyclic-forming-state into an octahedral nanocage, UWN_0453 (Chapter 2.3). With the current state of experimental and structural characterization of their combined crystal construct, the nanocage architecture can still be assembled *in vitro* with signs of the dihedral interface forming off of the nanocage's three-fold axes. Now that we have an understanding of each of the interfaces in the crystal construct, we can start to modulate them to achieve the *in vitro* crystallization into the designed lattice architecture.

2.5 Discussion of Crystal Design Using Custom-Made Building Blocks

This chapter illustrates how the accuracy and versatility of protein design [3] can now be leveraged to custom-make proteins for specific functions. Here, I described a three-step process to hierarchically design protein crystals from the bottom up. Step one is to make dihedral oligomers from starting points consisting of either naturally-occurring or *de novo* cyclic protein oligomers (Chapter 2.2), which I was able to do with success, lending many starting points for the next step. Step two is to incorporate the cyclic version of the dihedral protein into a nanocage in a geometry that allows for multiple nanocages to use this interface to self-assemble in a three-dimensional array. (Chapter 2.3). Having only started to thoroughly explore this space, it is exciting that I have one successful protein nanocage, UWN_0453, that has been structurally verified so far. The final step is to combine the dihedral- and cage-forming interfaces together on

a single construct to generate the crystal-forming construct (Chapter 2.4). The results demonstrate that in the crystal construct both interfaces are still capable of forming; however, not in a way that has resulted in the spontaneous *in vitro* assembly of the two building blocks into a crystal lattice.

Why is the crystal-forming construct not spontaneously self assembling, as expected? In the off-target crystal formed by the two-component nanocage, T33-15 (Chapter 1.6), it was clear that the designed dihedral interface was too weak to out-compete off-target crystal contacts. For this design, the dihedral interface is definitely strong enough to form, but could actually be too strong. Additionally, there could be more complicated kinetics at play that we could modulate by varying buffer conditions (salt, pH) or varying the ratios of the two building blocks in the mixture.

In addition to the clear paths forward to troubleshoot and learn more about the UWN_0453 and D3_1wa3-44 based crystal construct, this work also demonstrates that this hierarchical approach has the potential to yield promising results with other constructs. The crystal construct generated in this work only represents a small snapshot of the hierarchical design space. This construct was made from an octahedral protein interacting with a dihedral interface along its three-fold axis. For three-fold oligomers alone, there are two options of space groups that integrate polyhedral and dihedral centers (Fig. 2.9). Therefore, with the dihedral oligomers described in Chapter 2.2.1 alone, there are three other polyhedral plus dihedral architectures that can be pursued.

This protocol can also be expanded to use with D2 oligomers along two-fold axes and D4 oligomers along four-fold axes to generate the crystal space groups shown in Table 1.1. The next

steps forward will be to apply this protocol using the currently structurally verified *de novo* cyclic-to-dihedral oligomers described in Chapter 2.2.2 as well as other dihedral oligomers designed in the Baker Lab (Appendix Table 1.1). Combining ongoing developments in the Rosetta software and its energy function with this hierarchical custom-protein design strategy, more complex architecture such as three-dimensional crystals will soon be designable.

Figures

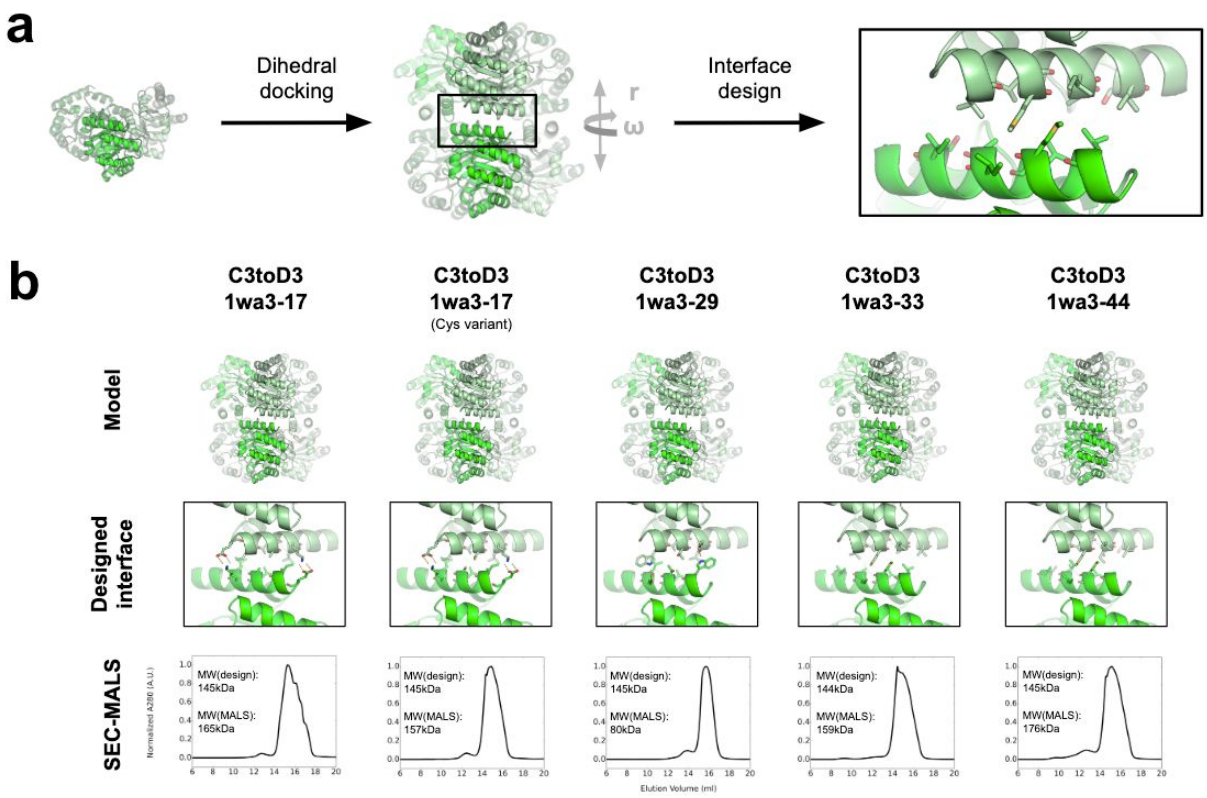


Figure 2.1. Cyclic-to-dihedral design of native homo-trimer, 1wa3. (a) The computational protocol for generating dihedral versions of 1wa3 included a docking step in which the trimer was kept stationary, and after applying dihedral symmetry, two degrees of freedom were sampled, r and ω, which correlate to translations between the upper and lower trimer and rotations around the cyclic axis, respectively. This docking step was followed by Rosetta interface design. (b) Models and close-ups of the new designed interfaces of the five ordered cyclic-to-dihedral 1wa3 variants are shown along with SEC-MALS data indicating their homogeneity and observed molecular weights.

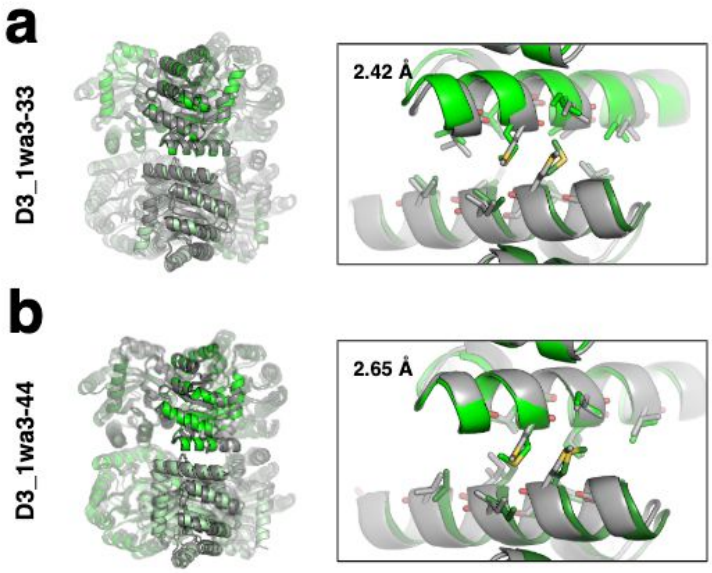


Figure 2.2. Comparison between experimentally determined crystal structures and corresponding design models. Crystal structures are shown in green and models in gray. Left column, full model and crystal structure superposition. Right column, superposition showing hydrophobic side chains at the designed interface. (a), D3_1wa3-33 (b), D3_1wa3-44.

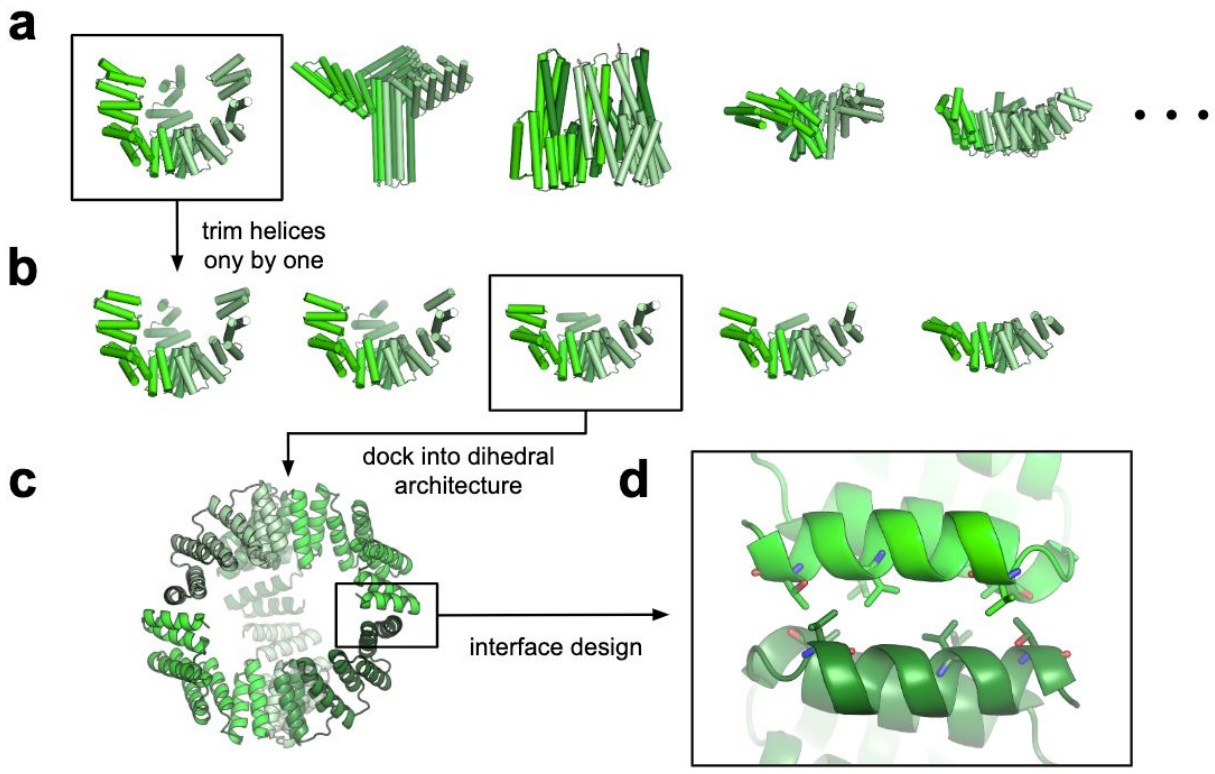


Figure 2.3. Computational protocol for designing dihedral oligomers from *de novo* cyclic oligomers. (a) A databased of *de novo* oligomers primarily comprised of alpha helices is curated. (b) For those oligomers where it is possible, helices are trimmed off one by one to generate a larger database of input structures. (c) All of these input structures are docked into dihedral architectures and then (d) Rosetta interface design is performed.

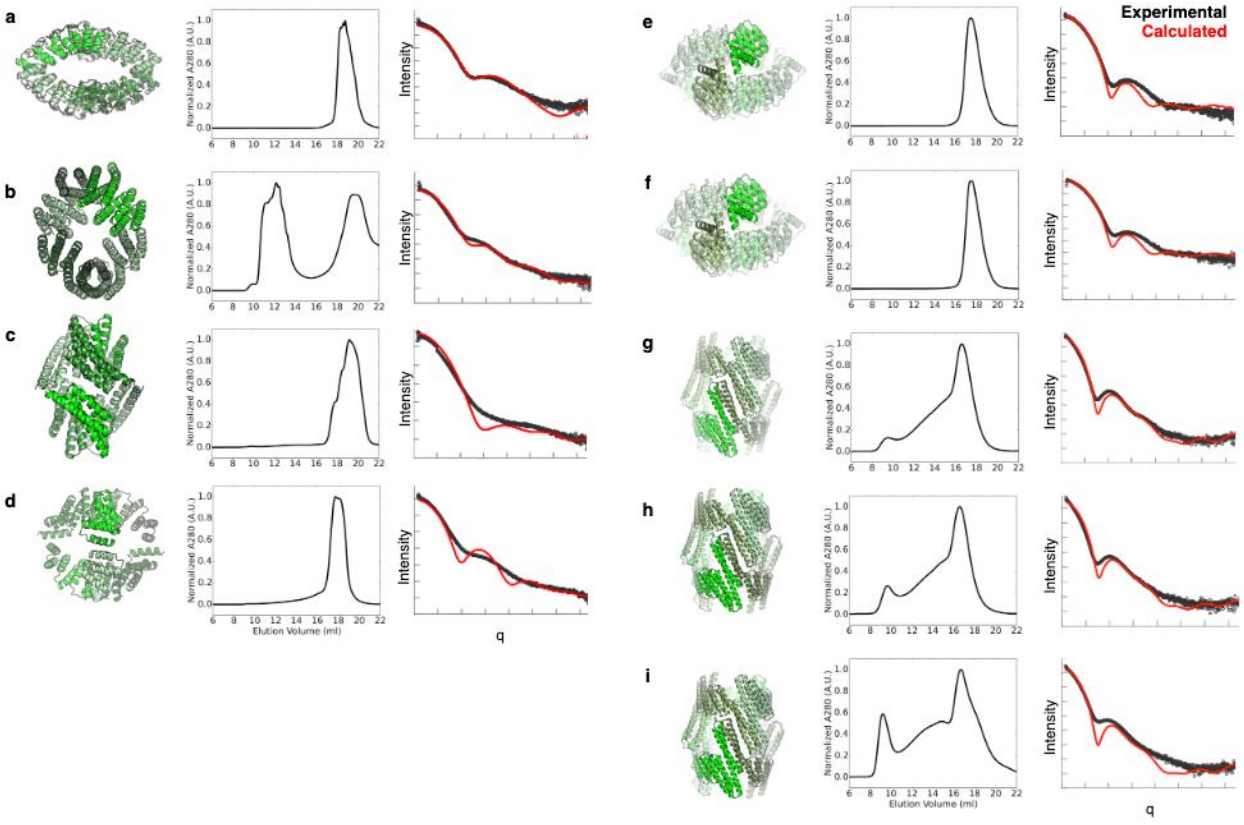


Figure 2.4. Experimental data on nine cyclic-to-dihedral *de novo* designed oligomers. Left column, design models. Middle column, size exclusion chromatography traces. Right column, small-angle X-ray scattering data showing experimental data in black and calculated data for the design model dihedral oligomers in red. Data is shown for (a) C2toD2_558, (b) C2toD2_595, (c) C2toD2_615, (d) C3toD3_455, (e) C3toD3_485, (f) C3toD3_486, (g) C3toD3_501, (h) C3toD3_502, and (i) C3toD3_504.

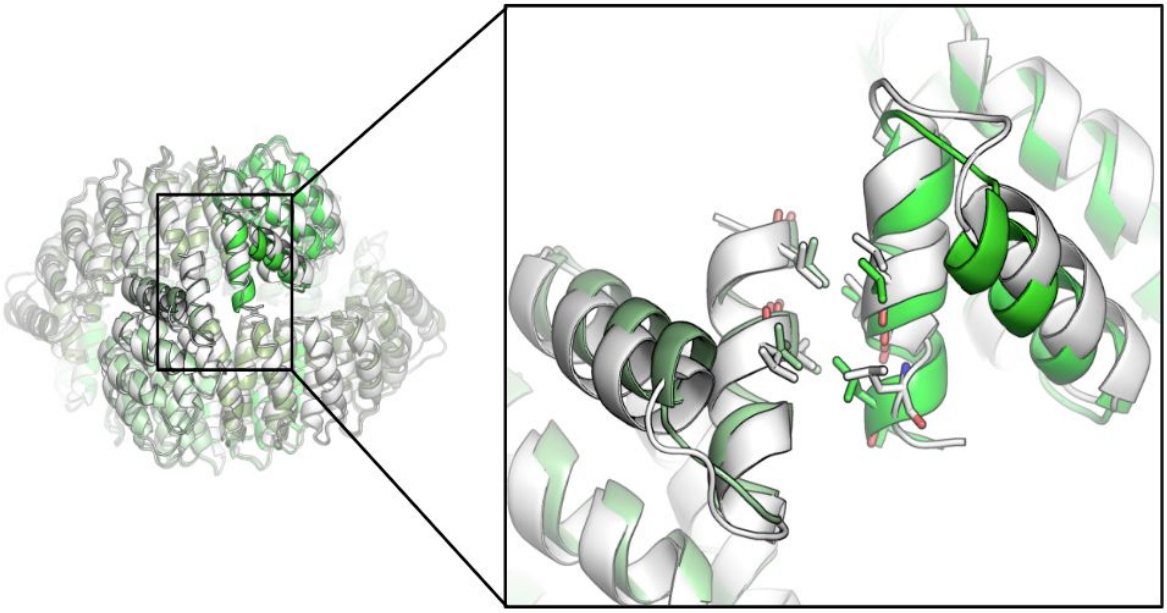


Figure 2.5. Crystal structure of C3toD3_486. The crystal structure shown in white overlays well with the design model shown in green, with structural agreement down to the rotamer positions at the interface.

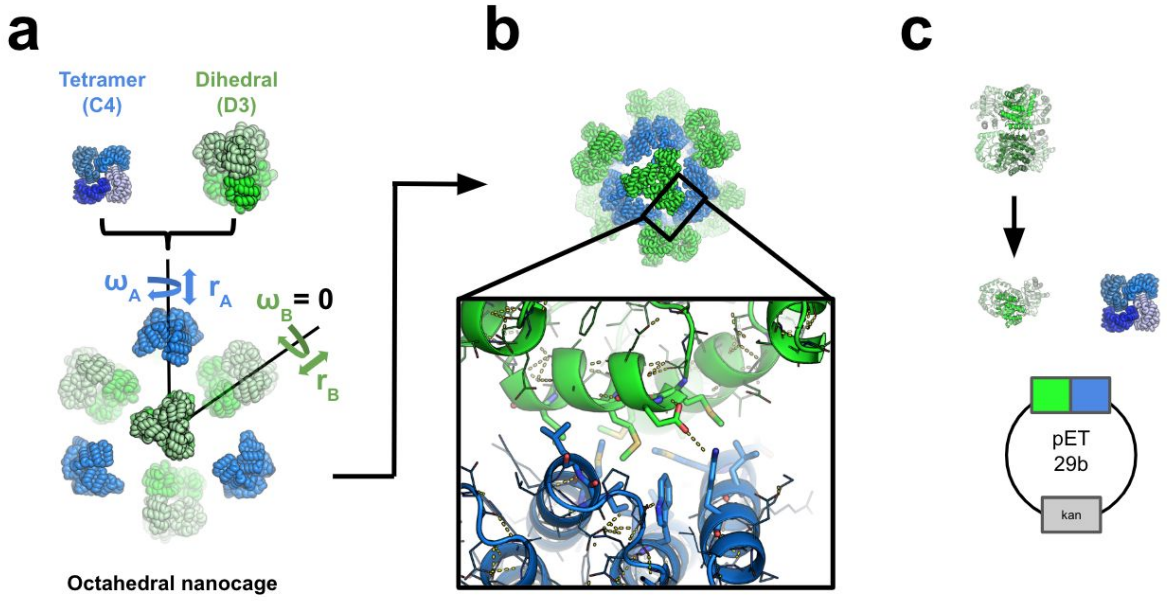


Figure 2.6. Computational protocol for geometrically-constrained nanocage design. (a) In the docking protocol, the C4 and D3 component are aligned with respect to the O43 architecture and three degrees of freedom are sampled while the rotation of the D3 protein is disallowed. (b) Rosetta interface design is used to generate a promising nanocage interface between the two subunits. (c) The dihedral interface is reverted back to its cyclic-oligomer-forming state, and then genes encoding the cage-forming C3 and C4 are ordered bicistronically.

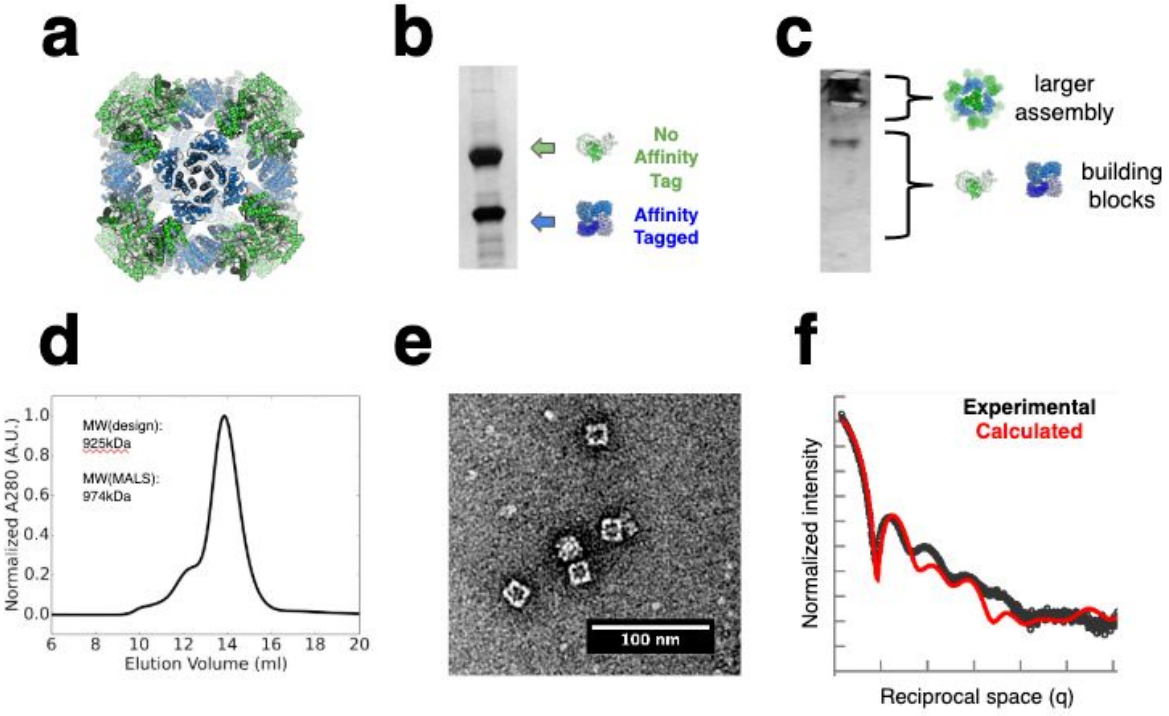


Figure 2.7. Experimental characterization of O43-453. (a) The design model of nanocage O43-453, (b) an SDS-PAGE gel of the pull-down of the non-affinity tagged trimer (green) by the affinity-tagged tetramer (blue), (c) a Native-PAGE gel of an assembly assay, (d) an SEC trace of the sample and the SEC-MALS observed molecular weight compared to the expected nanocage molecular weight, (e) negative-stain electron microscopy of the SEC-purified sample, and (f) an overlay of the calculated and experimental SAXS profiles of the nanocage.

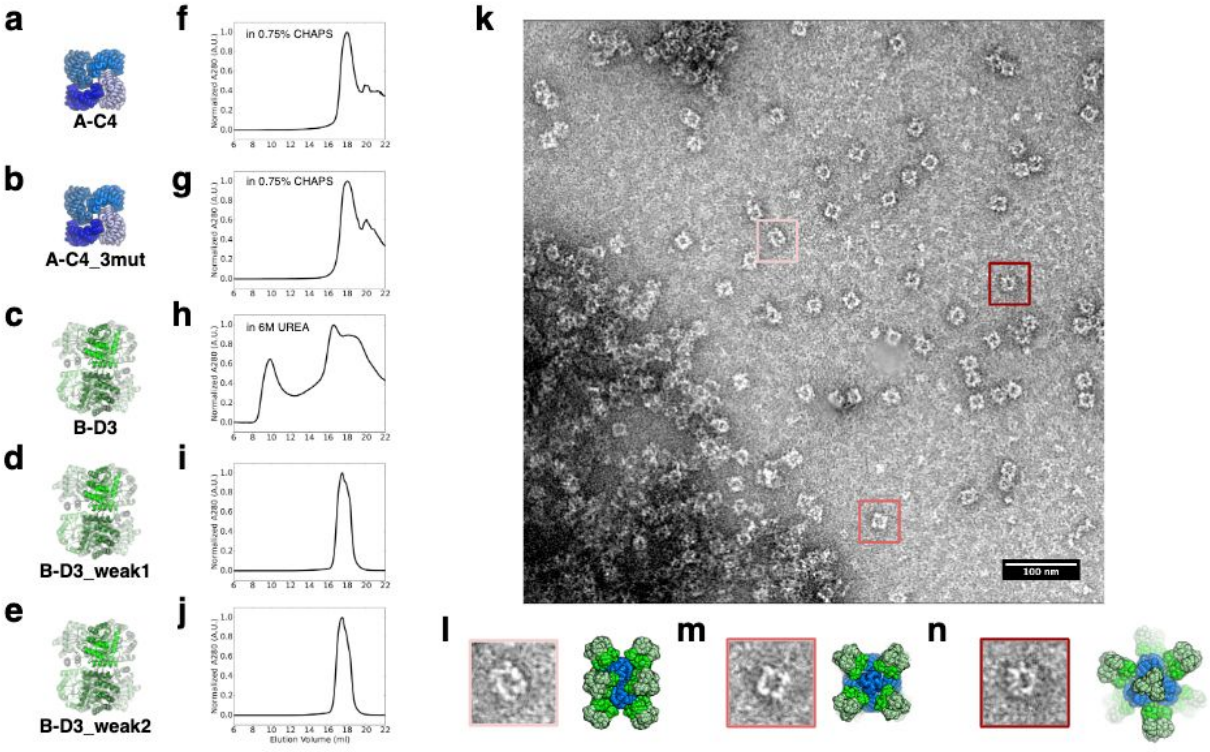


Figure 2.8. Purification and preliminary characterization of crystal construct UWN_453 with D3_1wa3-44 mutations. (a, b, c, d, e) Design models of the various A and B components constructs along with SEC traces from their purification (f, g, h, i, j). (k) A 1:1 mixture of A-C4 (a) and B-D3_weak2 (e) visualized by negative-stain EM. Individual particles (l, m, n) showcase the various views of the nanocage plus dihedral vertices.

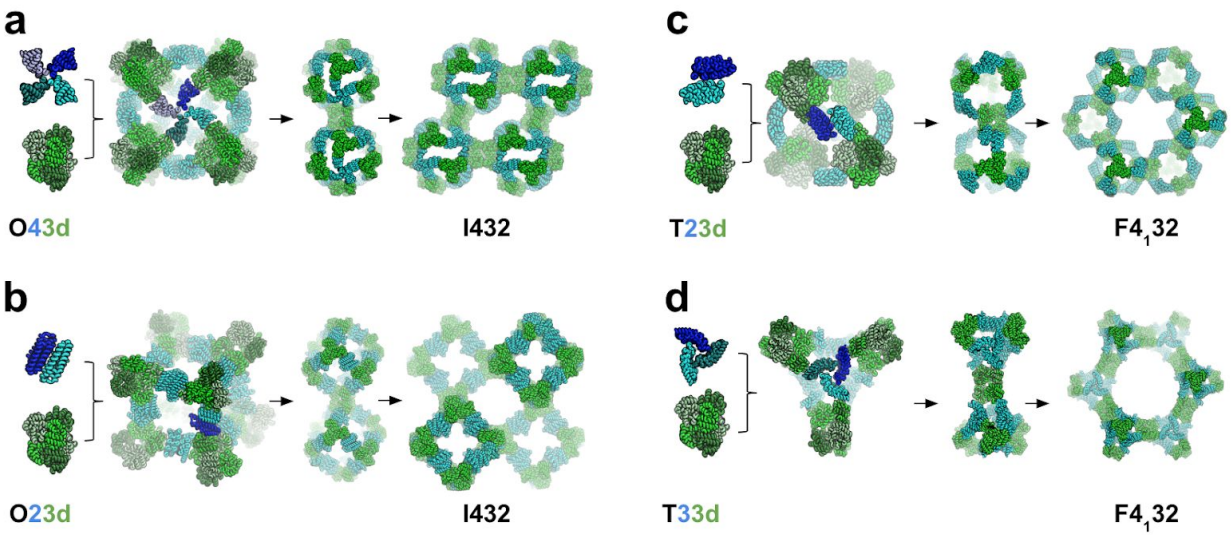


Figure 2.9. The crystal space groups that can be generated by combining D3 dihedral and polyhedral architectures. (a,b) The I432 space group can be generated by combining octahedra, O43 or O23, with D3 oligomers along their three-fold interface. (c,d) The F4$_1$32 space group can be generated by combining tetrahedra, T23 or T33, with D3 oligomers along their three-fold interface.

Tables

Table 2.1. Data collection and refinement statistics for cyclic-to-dihedral 1wa3 crystals.

	1wa3_33	1wa3_44
Wavelength	0.9598	0.9598
Resolution range	48.76 - 2.419 (2.506 - 2.419)	47.58 - 2.646 (2.74 - 2.646)
Space group	P 2 21 21	P 1 21 1
Unit cell	87.192 130.317 136.64 90 90 90	79.129 127.358 79.871 90 116.329 90
Total reflections	442305 (42924)	156421 (15490)
Unique reflections	60109 (5373)	41099 (3350)
Multiplicity	7.4 (7.3)	3.8 (3.8)
Completeness (%)	89.48 (66.05)	83.33 (54.32)
Mean I/sigma(I)	12.31 (0.76)	10.23 (0.78)
Wilson B-factor	46.15	54.10
R-merge	0.1313 (2.46)	0.1187 (1.823)
R-meas	0.1413 (2.648)	0.1382 (2.121)
R-pim	0.05188 (0.972)	0.0705 (1.08)
CC1/2	0.999 (0.346)	0.997 (0.303)
CC*	1 (0.717)	0.999 (0.682)
Reflections used in refinement	59581 (3884)	39273 (2236)
Reflections used for R-free	1773 (132)	1719 (115)
R-work	0.2038 (0.3253)	0.2124 (0.4056)
R-free	0.2375 (0.3532)	0.2403 (0.4148)

CC(work)	0.961 (0.667)	0.955 (0.474)
CC(free)	0.942 (0.591)	0.931 (0.261)
Number of non-hydrogen atoms	9340	8994
macromolecules	9174	8967
solvent	166	27
Protein residues	1246	1241
RMS(bonds)	0.005	0.003
RMS(angles)	0.74	0.69
Ramachandran favored (%)	98.38	98.37
Ramachandran allowed (%)	1.62	1.63
Ramachandran outliers (%)	0.00	0.00
Rotamer outliers (%)	0.00	0.35
Clashscore	5.07	7.64
Average B-factor	53.37	60.65
macromolecules	53.44	60.67
solvent	49.10	53.67

Statistics for the highest-resolution shell are shown in parentheses.

Table 2.2. Data collection and refinement statistics for *de novo* cycic-to-dihedral UWN_486 crystal.

	UWN_486
Data Collection	
Space group	$P2_1$
Cell parameters, a,b,c (Å) α, β, γ (°)	84.42 97.66 102.32 90.00 114.04 90.00
Resolution (highest shell) (Å)	93.45 – 3.50 (3.63 – 3.50)
No. of unique reflections	19073
Redundancy (last shell)	4.0 (4.0)
Completeness (last shell) (%)	98.78 (92.97)
I/σ(I) (last shell)	7.58 (2.63)
R_{merge} (last shell)	0.083 (0.637)
$CC_{1/2}$	0.998
CC*	1.000
Refinement Statistics	
Resolution range (Å)	93.45 – 3.50
No. of reflections used	19048
No. of protein atoms	7546
No. of water molecules	32
R_{work}	0.21
R_{free}	0.26
rmsd bond lengths (Å)	0.001
rmsd bond angles (°)	0.41
Ramachandran Favored/allowed Outlier (%)	 96.16/3.74 00.09

Average $B_{factors}$ (Å^2)	
protein	117.81
water	83.96

REFERENCES

[1] Wegst, U.G.K., Bai, H., Saiz, E., Tomsia, A.P., and Ritchie, R.O. (2015). Bioinspired structural materials. *Nat Mater* 14: 23-36.

[2] Hong, F., Zhang, F., Liu, Y., and Yan, H. (2017). DNA Origami: Scaffolds for Creating Higher Order Structures. *Chem Rev* 117: 12584-640.

[3] Huang, P.S., Boyken, S.E., and Baker, D. (2016). The coming of age of de novo protein design. *Nature* 537: 320-7.

[4] Boyken, S.E., Chen, Z., Groves, B., Langan, R.A., Oberdorfer, G., Ford, A., Gilmore, J.M., Xu, C., DiMaio, F., Pereira, J.H., Sankaran, B., Seelig, G., Zwart, P.H., and Baker D. (2016). De novo design of protein homo-oligomers with modular hydrogen-bond network-mediated specificity. *Science* 352: 680-7.

[5] Fallas, J.A., Ueda, G., Sheffler, W., Nguyen, V., McNamara, D.E., Sankaran, B., Pereira, J.H., Parmeggiani, F., Brunette, T.J., Cascio, D., Yeates, T.R., Zwart, P., and Baker, D. (2017). Computational design of self-assembling cyclic protein homo-oligomers. *Nat Chem* 9: 353-60.

[6] Sahasrabuddhe, A., Hsia, Y., Busch, F., Sheffler, W., King, N.P., Baker, D., and Wysocki, V.H. (2018). Confirmation of intersubunit connectivity and topology of designed protein complexes by native MS. *Proc Natl Acad Sci USA* 115:1268-73.

[7] King, N.P., Sheffler, W., Sawaya, M.R., Vollmar, B.S., Sumida, J.P., Andre, I., Gonen, T., Yeates, T.O., and Baker, D. Computational design of self-assembling protein nanomaterials with atomic-level accuracy. *Science*. (2012) 336:1171-4.

[8] King, N.P., Bale, J.B., Sheffler, W., McNamara, D.E., Gonen, S., Gonen, T., Yeates, T.O., and Baker, D. Accurate design of co-assembling multi-component protein nanomaterials. *Nature*. (2014) 510:103-8.

[9] Hsia, Y., Bale, J.B., Gonen, S., Shi, D., Sheffler, W., Fong, K.K., Nattermann, U., Xu, C., Huang, P.S., Ravichandran, R., Yi, S., David, T.N., Gonen, T., King, N.P., and Baker, D. (2016). Design of a hyperstable 60-subunit protein dodecahedron. *Nature* 535:136-9.

[10] Bale, J.B., Gonen, S., Liu, Y., Sheffler, W., Ellis, D., Thomas, C., Cascio, D., Yeates, T.O., Gonen, T., King, N.P., and Baker, D. (2016). Accurate design of megadalton-scale two-component icosahedral protein complexes. *Science* 353:389-94.

[11] Shen, H. Fallas, J.A, Lynch, E., Sheffler, W., Parry, B., Jannetty, N., Decarreau, J., Wagenbach, M., Vincente, J.J., Chen, J., Wang, L., Dowling, W., Oberdorfer, G., Stewart, L., Wordeman, K., DeYoreo, J., Jacobs-Wagner, C., Kollman, J., and Baker, B. (2018) De novo design of self-assembling helical protein filaments. *Science* 362: 705-9.

[12] Gonen, S., DiMaio, F., Gonen, T., and Baker, D. (2015). Design of ordered two-dimensional arrays mediated by noncovalent protein-protein interfaces. *Science* 348: 1365-8.

[13] Chen, Z., Johnson, M.C., Chen, J., Bick, M.J., Boyken, S.E., Lin, B., DeYoreo, J.J., Kollman, J.M., Baker, D., and DiMaio, D. (2019). Self-Assembling 2D Arrays with de Novo Protein Building Blocks. *J Am Chem Soc* 141: 8891-5.

[14] Elemans, J.A.A.W., Rowan, A.E., Noelte, R.J.M. Mastering Molecular Matter. Supramolecular architectures by hierarchical self-assembly. *Journal of Mat Chem*. (2003) 13:2661-70.

[15] Grueninger, D., Treiber, N., Ziegler, M.O., Koetter, J.W., Schulze, M.S., and Schultz, G.E. Designed protein-protein association. *Science*. (2008) 319:206-9.

[16] Zheng, J., Birktoft, J.J., Chen, Y., Wang, T., Sha, R., Constantinou, P.E., Ginell, S.L., Mao, C., and Seeman, N.C. (2009). From molecular to macroscopic via the rational design of a self-assembling 3D DNA crystal. *Nature* 461: 74-77.

[17] Koide, S. (2009).Engineering of recombinant crystallization chaperones. *Curr Opin Struct Biol.* 19: 449-57.

[18] Idan, O. and Hess, H. (2013). Engineering enzymatic cascades on nanoscale scaffolds. *Curr Opin Biotech.* 24: 606-11.

[19] Lijestrom, V., Mikkila, J., and Kostiainen, M.A. (2014). Self-assembly and modular functionalization of three-dimensional crystals from oppositely charged proteins. *Nat Comm* 5: 4445.

[20] Brodin, J.D., Ambroggio, X.I., Tang, C., Parent, K.N., Baker, T.S., and Tezcan, F.A. (2012). Metal-directed, chemically tunable assembly of one-, two- and three-dimensional crystalline protein arrays. *Nat Chem.* 4: 375-82.

[21] Sontz, P.A., Bailey, J.B., Ahn, S., and Tezcan, F.A. (2015). A metal organic framework with spherical protein nodes: rational chemical design of 3D protein crystals. *J Am Chem Soc.* 137:11598-601.

[22] Brodin, J.D., Auyeung, E., and Mirkin, C.A. (2015). DNA-mediate engineering of multicomponent enzyme crystals. *Proc. Natl. Acad. Sci. USA.* 112:4564-9.

[23] Kostiainen, M.A., Hiekkataipale, P., Laiho, A., Lemieux, V. Seitsonen, J., Ruokolainen, J., and Ceci, P. (2013). Electrostatic assembly of binary nanoparticle superlattices using protein cages. *Nat Nanotech* 8: 52-6.

[24] Zhang, L., Bailey, J.B., Subramanian, R.H., Groismna, A., and Tezcan, F.A. (2018). Hyperexpandable, self-healing macromolecular crystal with integrated polymer networks. *Nature* 557: 86-91.

[25] Lanci, C.J., MacDermaid, C.M., Kang, S., Acharya, R., North, B., Yang, X., Qiu, X.J., DeGrado, W.F., and Saven, J.G. (2012). Computational design of a protein crystal. *Proc. Natl. Acad. Sci.* 109: 7304-09.

[26] Yeates, T.O., Liu, Y., and Laniado, J. (2016). The design of symmetric protein nanomaterials comes of age in theory and practice. *Curr Opin Struct Biol* 39: 134-43.

[27] Leaver-Fay, A., et al. (2011). ROSETTA3: an object-oriented software suite for the simulation and design of macromolecules. *Methods Enzymol*. 487:545-74.

[28] DiMaio, F., Leaver-Fay, A., Bradley, P., Baker, D., and Andre, I. (2011). Modeling Symmetric Macromolecular Structures in Rosetta3. *PLoS One* 6:e20450.

[29] Brunette, T.J., Parmeggiani, F., Huang, P.S., Bhabha, G., Ekiert, D.C., Tsutakawa, S.E., Hura, G.L., Tainer, J.A., and Baker, D. (2015). Exploring the repeat protein universe through computational protein design. *Nature* 528: 580-4.

[30] Fleishman SJ, Leaver-Fay A, Corn JE, Strauch EM, Khare SD, et al. (2011). RosettaScripts: A Scripting Language Interface to the Rosetta Macromolecular Modeling Suite. *PLoS One* 6: e20161.

[31] Garcia-Seisdedos, H., Empereur-Mot, C., Elad, N., and Levy, E.D. (2017). Proteins evolve on the edge of supramolecular self-assembly. *Nature* 548: 244-7.

[32] Koide, S. (2009). Engineering recombinant crystallization chaperones. *Curr Opin Struct Biol* 19:449-57.

[33] Bhaskar, S., and Lim, S. (2017). Engineering protein nanocages as carriers for biomedical applications. *NPG Asia Mater* 9: e371.

[34] Zhao, Z, Jacovetty, E.L., Liu, Y., and Yan, H. (2011). Encapsulation of gold nanoparticles in a DNA origami cage. *Angew Chem Int Ed Engl* 50: 2041-4.

[35] Zang, J., Zheng, B., Zhang, X., Arosio, P., and Zhao, G. (2019). Design and site-directed compartmentalization of gold nanoclusters within the intrasubunit interfaces of ferritin nanocage. *J Nanobiotech* 17: 79.

[36] Alford, R.F., Leaver-Fay, A., Jealiazkov, J.R., O'Meara, M.J., DiMaio, F.P., Park, H., Shapovalov, M.V., Renfrew, P.D., Mulligan, V.K., Kappel, K., Labonte, J.W., Pacella, M.S., Bonneau, R., Bradley P., Dunbrack, Jr., R.L., Das, R., Baker, D., Kuhlman, B., Koretemme, T., and Gray, J.J. (2017). The Rosetta All-Atom Energy Function for Macromolecular Modeling and Design. *J Chem Theory Comput* 13: 3031-48.

[37] Fullerton, S.W.B., Griffiths, J.S., Merkel, A.B., Cheriyan, M., Wymer, N.J., Hutchins, M.J., Fierke, C.A., Toone, E.J., and Naismith, J.H. (2006). Mechanism of the Class I KDPG aldolase. *Bioorgan Med Chem* 14: 3002-10.

APPENDIX

APPENDIX METHODS

Appendix Methods 1.1. Modifications to the SymDofMover to improve the modeling of crystalline materials.

The primary symmetry modeling component of the Rosetta macromolecular modeling software that was used for the projects described in this thesis is called the SymDofMover. This tool is used to set up symmetric systems in which the input structures are aligned along the x, y, or z axis and allow for sampling movements -- rotations and translations -- around those axes. While the SymDofMover already contained all of the necessary attributes for modeling cyclic, dihedral, and polyhedral symmetries, it was not yet capable of modeling crystalline symmetries. To address this, I added new attributes that could be optionally used when modeling crystals. The following RosettaScript options, and their default settings in parentheses, were added:

```
cell_dof_(""),
cell_dof_value_(0),
cell_dof_delta_step_(0),
cell_dof_delta_min_(0),
cell_dof_delta_max_(0),
cell_dof_keep_input_coords_(false),
d2position_(),
d3position_(),
d4position_(),
d5position_(),
d6position_(),
```

The C++ code added for the cell_dof_value option is shown below:

```
SymDofMover::get_cell_dof_value() {
        // for now, only one cell dof supported
        Real cell_dof_value;
        if(sampling_mode_ == "grid" ) {
```

```
                // offset based on absolute magnitude
                if( cell_dof_value_ > 0.0 ){
                        cell_dof_value = cell_dof_value_ +
SymDofMoverSampler::get_instance()->get_cell_dof_offset();
                } else {
                        cell_dof_value = cell_dof_value_ -
SymDofMoverSampler::get_instance()->get_cell_dof_offset();
                }
        } else {
                utility_exit_with_message("cell dof sampling is grid only at this point.");
        }
        // if( set_sampler_ ) {
        //
SymDofMoverSampler::get_instance().set_cell_dof_offset(cell_dof_value);
        // }
        return cell_dof_value;
}
```

The C++ code added for the cell_dof options that allow for grid sampling along the unit cell dimensions is shown below:

```
if(""!=slide_dof_)
slide_building_blocks(slide_dof_,pose,slide_chunk_one_,slide_chunk_two_);

                if( cell_dof_.size() ){
                        Size sym_aware_jump_id =
core::pose::symmetry::sym_dof_jump_num( pose, cell_dof_ );
                        core::kinematics::Jump j( pose.jump(sym_aware_jump_id) );
                        Vec t = j.get_translation();
                        if( sampling_mode_ == "grid" ) {
                                if( slide_dof_.size()){
                                        TR << "set cell spacing after slide: " <<
j.get_translation() << " + " << get_cell_dof_value() << std::endl;
                                        if( j.get_translation() > 0){
                                                j.set_translation( j.get_translation() + Vec(
get_cell_dof_value() ,0,0) );
                                        } else {
                                                j.set_translation( j.get_translation() + Vec(
-get_cell_dof_value() ,0,0) );
                                        }
                                } else if( fabs(cell_dof_value_) > 0.0001 ) {
                                        j.set_translation( Vec( get_cell_dof_value() ,0,0) );
```

```
                                        } else {
                                                utility_exit_with_message("if cell_dof specified,
you must either slide or specify cell_dof_value");
                                        }
                                } else {
                                        Real val=0, rand=numeric::random::rg().gaussian();
                                        if( ""!=slide_dof_ ){
                                                if( t.x() > 0.0 ) rand += 1.0;
                                                else              rand -= 1.0;
                                                TR << "cell spacing offset by 1.0 because slide
used" << std::endl;
                                        }
                                        if( fabs(cell_dof_value_) > 0.001 ){
                                                val = cell_dof_value_ + cell_dof_delta_step_*rand;
                                                TR << "setting cell spacing dof " << cell_dof_ << "
" << cell_dof_value_ << " " << val << " " << rand << std::endl;
                                        } else {
                                                val = t.x() + rand;
                                                TR << "purturbing existing cell spacing dof " <<
cell_dof_ << " " << cell_dof_value_ << " " << val << " " << rand << std::endl;
                                        }
                                        // runtime_assert_msg( cell_dof_value_ != 0.0,
"cell_dof_value must be set to non-zero value if cel_dof is specified." );
                                        runtime_assert_msg( fabs(t.y()) < 0.0001 && fabs(t.z()) <
0.0001, "cell_dof translation should only be along x." );
                                        j.set_translation( Vec( val ,0,0) );
                                }
        // cell dof sampling
                        cell_dof_delta_step_ = tag->getOption<core::Real>(
"cell_dof_delta_step", 0.0 );
                        cell_dof_delta_min_  = tag->getOption<core::Real>(
"cell_dof_delta_min", 0.0 );
                        cell_dof_delta_max_  = tag->getOption<core::Real>(
"cell_dof_delta_max", 0.0 );
                        SymDofMoverSampler::get_instance()->set_cell_dof_range(
cell_dof_delta_min_, cell_dof_delta_max_, cell_dof_delta_step_ );
```

The C++ code added for the cell_dof_keep_input_coords option is shown below:

```
                                // store protein coords
                                ///std::cout << "copy pose tmp" << std::endl;
                                Pose tmp(pose); // could be made more efficient usint ony asym
        unit
```

```
                        ///pose.dump_pdb("cell_dof_jump_set_before_copy.pdb");
                        pose.set_jump(sym_aware_jump_id,j);

                        if( cell_dof_keep_input_coords_ == true ) {
                                ///std::cout << "restore asym unit coords" << std::endl;
                                core::conformation::symmetry::SymmetryInfoCOP
sym_info = core::pose::symmetry::symmetry_info(pose);
                                utility::vector1< bool > resi_is_indep =
sym_info->independent_residues();
                                // restore protein coords, would be better to use only asym
unit
                                for(core::Size ir = 1; ir <= pose.size(); ++ir){
                                        if( resi_is_indep[ir] && !
pose.residue(ir).is_virtual_residue() ){
                                                for(core::Size ia = 1; ia <=
pose.residue_type(ir).natoms(); ++ia){
                                                        pose.set_xyz(
core::id::AtomID(ia,ir), tmp.residue(ir).xyz(ia) );
                                                }
                                        }
                                }
                                ///std::cout << "restore asym coords done" << std::endl;

                                ///pose.dump_pdb("cell_dof_jump_set.pdb");
                        }
                }

                if (basic::options::option[
basic::options::OptionKeys::symmetry::detect_bonds ]){
                        pose.conformation().detect_bonds();
                        pose.conformation().detect_disulfides();
                }
```

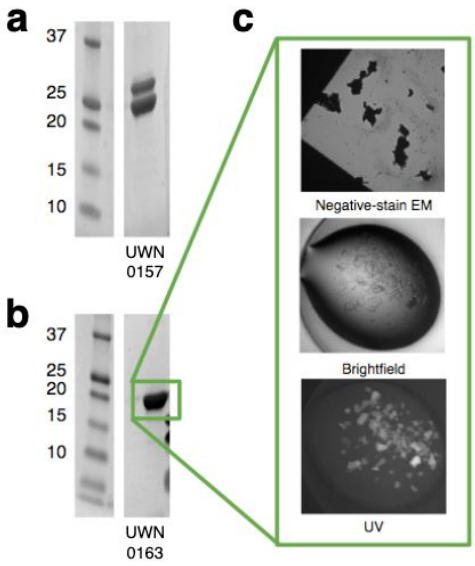
The C++ code added for the options that allow one to sample discrete dihedral positions is shown below:

```
if( d2position_.size() ){

runtime_assert_msg(d2position_.size()==2,"d2position must be length 2");
                                        if( d2position_[i] != "0" )       TR << "set D2
position on component " << i << " to " << d2position_[i] << std::endl;
                                        if(     d2position_[i] == "0" )  ;
```

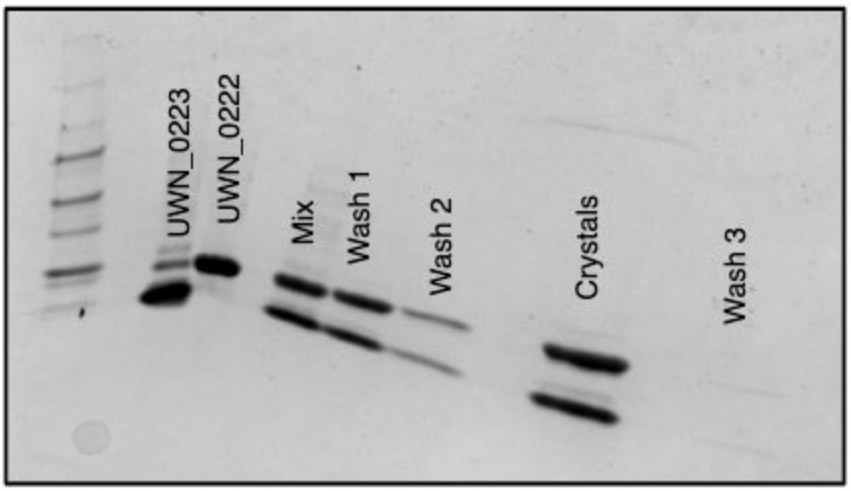
```
                                        else if( d2position_[i] == "1" )  ;
                                        else if( d2position_[i] == "2" )
rot_pose(pose,Vec(1,0,0), 90.0,sub_start,sub_end);
                                        else if( d2position_[i] == "3" )
rot_pose(pose,Vec(0,1,0), 90.0,sub_start,sub_end);
                                        else if( d2position_[i] == "4" )
rot_pose(pose,Vec(0,0,1), 90.0,sub_start,sub_end);
                                        else if( d2position_[i] == "5" )
rot_pose(pose,Vec(1,1,1),120.0,sub_start,sub_end);
                                        else if( d2position_[i] == "6" )
rot_pose(pose,Vec(1,1,1),240.0,sub_start,sub_end);
                                        else utility_exit_with_message("unknown
d2position: "+d2position_[i]+", must be 1-6, or 0");
                                }
                                if( d3position_.size() ){

runtime_assert_msg(d3position_.size()==2,"d3position must be length 2");
                                        if( d3position_[i] != "0" )       TR << "set D3
position on component " << i << " to " << d3position_[i] << std::endl;
                                        if(     d3position_[i] == "0" )  ;
                                        else if( d3position_[i] == "1" )  ;
                                        else if( d3position_[i] == "2" )
rot_pose(pose,Vec(0,0,1), 60.0,sub_start,sub_end);
                                        else utility_exit_with_message("unknown
d3position: "+d3position_[i]+", must be 1-2, or 0");
                                }
                                if( d4position_.size() ){

runtime_assert_msg(d4position_.size()==2,"d4position must be length 2");
                                        if( d4position_[i] != "0" )       TR << "set D4
position on component " << i << " to " << d4position_[i] << std::endl;
                                        if(     d4position_[i] == "0" )  ;
                                        else if( d4position_[i] == "1" )  ;
                                        else if( d4position_[i] == "2" )
rot_pose(pose,Vec(0,0,1), 45.0,sub_start,sub_end);
                                        else utility_exit_with_message("unknown
d4position: "+d4position_[i]+", must be 1-2, or 0");
                                }
                                if( d5position_.size() ){

runtime_assert_msg(d5position_.size()==2,"d5position must be length 2");
                                        if( d5position_[i] != "0" )       TR << "set D5
position on component " << i << " to " << d5position_[i] << std::endl;
                                        if(     d5position_[i] == "0" )  ;
```

```
                                        else if( d5position_[i] == "1" )  ;
                                        else if( d5position_[i] == "2" )
rot_pose(pose,Vec(0,0,1), 36.0,sub_start,sub_end);
                                        else utility_exit_with_message("unknown
d5position: "+d5position_[i]+", must be 1-2, or 0");
                                    }
                                    if( d6position_.size() ){

runtime_assert_msg(d6position_.size()==2,"d6position must be length 2");
                                        if( d6position_[i] != "0" )       TR << "set D6
position on component " << i << " to " << d6position_[i] << std::endl;
                                        if(      d6position_[i] == "0" )  ;
                                        else if( d6position_[i] == "1" )  ;
                                        else if( d6position_[i] == "2" )
rot_pose(pose,Vec(0,0,1), 30,sub_start,sub_end);
                                        else utility_exit_with_message("unknown
d6position: "+d6position_[i]+", must be 1-2, or 0");
                                    }
```

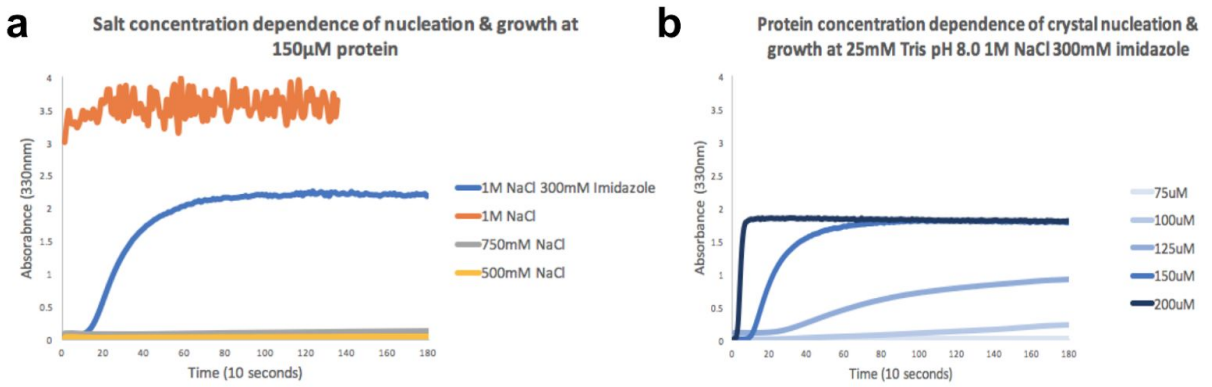
Appendix Figures



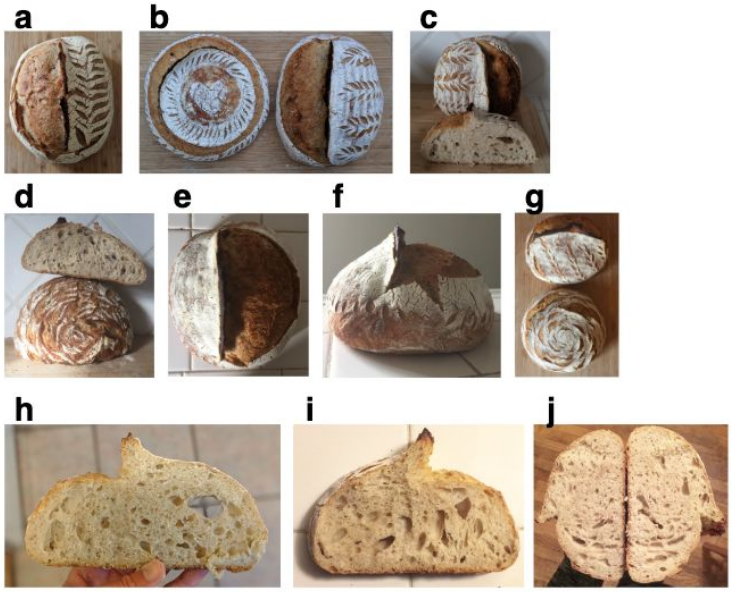
Appendix Figure 1.1. Promising F222_D2-D2 design candidates for follow up experimental and structural studies. SDS-PAGE gels of IMAC eluate for two-component designs (a) UWN_0157 and (b) UWN_0163. The two bands corresponding to the expected molecular weights for UWN_0157 are apparent; and the molecular weights of the components are similar enough for UWN_0163 that they overlay. (c) Negative-stain EM, brightfield, and UV light microscopy images of assemblies from the UWN_0163 IMAC eluate.



Appendix Figure 1.2. SDS-PAGE gel of the components and the T33-15 self-assembling crystal. SDS-PAGE gels showing the individual components, UWN_0223 and UWN_0222, the mixtures itself, and then sequential washes of the mixture with new buffer after pelleting and resuspending the crystals. The lane labelled 'Crystals' clearly shows two bands, while the final 'Wash 3' has very little protein left in it.



Appendix Figure 1.3. Static light scattering measurements of the crystal assembly process. Preliminary static light scattering measurements at 330nm showing (a) the change in the assembly kinetics from changing the salt concentration of the buffer solution while keeping the protein concentration of the mixtures constant and (b) the change in the assembly kinetics from changing the protein concentration while keeping the buffer solution conditions constant.



Appendix Figure 3.1. The sourdough bread loaves baked during the writing of this doctoral thesis. The seven loaves that were baked (a, b, d, e, g) as well as cross-section crumb shots of several of the loaves (c, d, h, i, j).

APPENDIX TABLES

Appendix Table 1.1. Oligomeric proteins with cyclic, dihedral ,and polyhedral symmetries curated from the Protein Data Bank and from the Baker Lab database.

	C2	**C3**	**C4**	**D2**	**D3**	**D4**	**T**	**O**
PDB	11,433	2,411	461	628	520	63	--	--
de novo	21	29	6	12	22	6	13	3
This work	--	2	--	3	10	--	--	1

Appendix Table 1.2. List of all protein designs tested and the locations of their plasmids.

Design ID	Architecture	Design Name	Plasmid Location
UWN_0001	$F4_132$_T-D3	T33-15_1_2fli_0054	UNPlasmidBox00
UWN_0002	$F4_132$_T-D3	T33-15_1_3d19_0008	UNPlasmidBox00
UWN_0003	$F4_132$_T-D3	T33-15_1_3d19_0017	UNPlasmidBox00
UWN_0004	$F4_132$_T-D3	T33-15_1_3d19_f0040	UNPlasmidBox00
UWN_0005	$F4_132$_T-D3	T33-21_1_3p5m_0041	UNPlasmidBox00
UWN_0006	$F4_132$_T-D3	T33-28_1_2v5j_0052	UNPlasmidBox00
UWN_0007	$F4_132$_T-D3	T33-28_2_2bty_0098	UNPlasmidBox00
UWN_0008	$F4_132$_T-D3	T33-28_2_2ef4_0098	UNPlasmidBox00
UWN_0009	$F4_132$_T-D3	T33-28_2_2ijc_0090	UNPlasmidBox00
UWN_0010	$F4_132$_T-D3	T33-28_2_3qvl_0023	UNPlasmidBox00
UWN_0011	$F4_132$_T-D3	T33-28_2_3qvl_0063	UNPlasmidBox00
UWN_0012	$F4_132$_T-D3	T33-28_2_4lhb_0021	UNPlasmidBox00
UWN_0013	$F4_132$_T-D3	T32-28_1_2brx_0042	UNPlasmidBox00
UWN_0014	P23_T-D2	TD2_01	UNPlasmidBox00
UWN_0015	P23_T-D2	TD2_02	UNPlasmidBox00
UWN_0016	P23_T-D2	TD2_03	UNPlasmidBox00
UWN_0017	P23_T-D2	TD2_04	UNPlasmidBox00
UWN_0018	P23_T-D2	TD2_06	UNPlasmidBox00
UWN_0019	P23_T-D2	TD2_07	UNPlasmidBox00
UWN_0020	P23_T-D2	TD2_08	UNPlasmidBox00
UWN_0021	P23_T-D2	TD2_09	UNPlasmidBox00
UWN_0022	P23_T-D2	TD2_10	UNPlasmidBox00
UWN_0023	P23_T-D2	TD2_11	UNPlasmidBox00
UWN_0024	P23_T-D2	TD2_12	UNPlasmidBox00
UWN_0025	P23_T-D2	TD2_13	UNPlasmidBox00
UWN_0026	P23_T-D2	TD2_14	UNPlasmidBox00
UWN_0027	P23_T-D2	TD2_15	UNPlasmidBox00
UWN_0028	P23_T-D2	TD2_16	UNPlasmidBox00
UWN_0029	P23_T-D2	TD2_17	UNPlasmidBox00
UWN_0030	P23_T-D2	TD2_18	UNPlasmidBox00
UWN_0031	P23_T-D2	TD2_19	UNPlasmidBox00
UWN_0032	P23_T-D2	TD2_20	UNPlasmidBox00
UWN_0033	P23_T-D2	TD2_21	UNPlasmidBox00
UWN_0034	P23_T-D2	TD2_22	UNPlasmidBox00
UWN_0035	P23_T-D2	TD2_23	UNPlasmidBox00
UWN_0036	P23_T-D2	TD2_24	UNPlasmidBox00

UWN_0037	P23_T-D2	TD2_25	UNPlasmidBox00
UWN_0038	P23_T-D2	TD2_26	UNPlasmidBox00
UWN_0039	P23_T-D2	TD2_27	UNPlasmidBox00
UWN_0040	P23_T-D2	TD2_28	UNPlasmidBox00
UWN_0041	P23_T-D2	TD2_29	UNPlasmidBox00
UWN_0042	P23_T-D2	TD2_30	UNPlasmidBox00
UWN_0043	Cn	1BH_01	UNPlasmidBox00
UWN_0044	Cn	1BH_02	UNPlasmidBox00
UWN_0045	Cn	1BH_03	UNPlasmidBox00
UWN_0046	Cn	1BH_04	UNPlasmidBox00
UWN_0047	Cn	1BH_05	UNPlasmidBox00
UWN_0048	Cn	1BH_06	UNPlasmidBox00
UWN_0049	Cn	1BH_08	UNPlasmidBox00
UWN_0050	Cn	1BH_09	UNPlasmidBox00
UWN_0051	Cn	1BH_10	UNPlasmidBox00
UWN_0052	Cn	1BH_11	UNPlasmidBox00
UWN_0053	Cn	1BH_12	UNPlasmidBox00
UWN_0054	Cn	1BH_13	UNPlasmidBox00
UWN_0055	Cn	1BH_14	UNPlasmidBox00
UWN_0056	Cn	1BH_15	UNPlasmidBox00
UWN_0057	Cn	1BH_16	UNPlasmidBox00
UWN_0058	Cn	1BH_17	UNPlasmidBox00
UWN_0059	Cn	1BH_18	UNPlasmidBox00
UWN_0060	Cn	1BH_19	UNPlasmidBox00
UWN_0061	Cn	1BH_20	UNPlasmidBox00
UWN_0062	Cn	1BH_21	UNPlasmidBox00
UWN_0063	Cn	1BH_22	UNPlasmidBox00
UWN_0064	Cn	1BH_23	UNPlasmidBox00
UWN_0065	Cn	NM01	UNPlasmidBox00
UWN_0066	Cn	NM02	UNPlasmidBox00
UWN_0067	Cn	NM03	UNPlasmidBox00
UWN_0068	Cn	NM04	UNPlasmidBox00
UWN_0069	Cn	NM05	UNPlasmidBox00
UWN_0070	Cn	NM06	UNPlasmidBox00
UWN_0071	Cn	NM07	UNPlasmidBox00
UWN_0072	Cn	NM08	UNPlasmidBox00
UWN_0073	Cn	NM09	UNPlasmidBox00
UWN_0074	Cn	NM10	UNPlasmidBox00
UWN_0075	Cn	NM11	UNPlasmidBox00
UWN_0076	Cn	NM12	UNPlasmidBox00

UWN_0077	Cn	NM14	UNPlasmidBox00
UWN_0078	Cn	NM15	UNPlasmidBox00
UWN_0079	Cn	NM16	UNPlasmidBox00
UWN_0080	Cn	NM17	UNPlasmidBox00
UWN_0081	Cn	NM18	UNPlasmidBox00
UWN_0082	Cn	NM19	UNPlasmidBox00
UWN_0083	Cn	NM20	UNPlasmidBox00
UWN_0084	Cn	NM21	UNPlasmidBox00
UWN_0085	Cn	NM23	UNPlasmidBox00
UWN_0086	Cn	NM24	UNPlasmidBox00
UWN_0087	Cn	NM25	UNPlasmidBox00
UWN_0088	P213_C3-C3	C3C3P213_01	UNPlasmidBox00
UWN_0089	P213_C3-C3	C3C3P213_02	UNPlasmidBox00
UWN_0090	P213_C3-C3	C3C3P213_03	UNPlasmidBox00
UWN_0091	P213_C3-C3	C3C3P213_04	UNPlasmidBox00
UWN_0092	P213_C3-C3	C3C3P213_05	UNPlasmidBox00
UWN_0093	P213_C3-C3	C3C3P213_06	UNPlasmidBox00
UWN_0094	P213_C3-C3	C3C3P213_08	UNPlasmidBox00
UWN_0095	P213_C3-C3	C3C3P213_09	UNPlasmidBox00
UWN_0096	P213_C3-C3	C3C3P213_10	UNPlasmidBox00
UWN_0097	P213_C3-C3	C3C3P213_11	UNPlasmidBox00
UWN_0098	P213_C3-C3	C3C3P213_12	UNPlasmidBox00
UWN_0099	P213_C3-C3	C3C3P213_13	UNPlasmidBox00
UWN_0100	P213_C3-C3	C3C3P213_14	UNPlasmidBox00
UWN_0101	P213_C3-C3	C3C3P213_15	UNPlasmidBox00
UWN_0102	P213_C3-C3	C3C3P213_16	UNPlasmidBox00
UWN_0103	P213_C3-C3	C3C3P213_17	UNPlasmidBox00
UWN_0104	P213_C3-C3	C3C3P213_18	UNPlasmidBox00
UWN_0105	P213_C3-C3	C3C3P213_19	UNPlasmidBox00
UWN_0106	P213_C3-C3	C3C3P213_20	UNPlasmidBox00
UWN_0107	P213_C3-C3	C3C3P213_21	UNPlasmidBox00
UWN_0108	P213_C3-C3	C3C3P213_22	UNPlasmidBox00
UWN_0109	P213_C3-C3	C3C3P213_23	UNPlasmidBox00
UWN_0110	P213_C3-C3	C3C3P213_24	UNPlasmidBox00
UWN_0111	P213_C3-C3	C3C3P213_25	UNPlasmidBox00
UWN_0112	P213_C3-C3	C3C3P213_27	UNPlasmidBox00
UWN_0113	P213_C3-C3	C3C3P213_28	UNPlasmidBox00
UWN_0114	P213_C3-C3	C3C3P213_29	UNPlasmidBox00
UWN_0115	P213_C3-C3	C3C3P213_30	UNPlasmidBox00
UWN_0116	P213_C3-C3	C3C3P213_31	UNPlasmidBox00

UWN_0117	P213_C3-C3	C3C3P213_32	UNPlasmidBox00
UWN_0118	P213_C3-C3	C3C3P213_33	UNPlasmidBox01
UWN_0119	P213_C3-C3	C3C3P213_34	UNPlasmidBox01
UWN_0120	P213_C3-C3	C3C3P213_35	UNPlasmidBox01
UWN_0121	P213_C3-C3	C3C3P213_36	UNPlasmidBox01
UWN_0122	P213_C3-C3	C3C3P213_37	UNPlasmidBox01
UWN_0123	P213_C3-C3	C3C3P213_38	UNPlasmidBox01
UWN_0124	P213_C3-C3	C3C3P213_39	UNPlasmidBox01
UWN_0125	P213_C3-C3	C3C3P213_40	UNPlasmidBox01
UWN_0126	P213_C3-C3	C3C3P213_41	UNPlasmidBox01
UWN_0127	P213_C3-C3	C3C3P213_42	UNPlasmidBox01
UWN_0128	P213_C3-C3	C3C3P213_43	UNPlasmidBox01
UWN_0129	P213_C3-C3	C3C3P213_44	UNPlasmidBox01
UWN_0130	P213_C3-C3	C3C3P213_46	UNPlasmidBox01
UWN_0131	P213_C3-C3	C3C3P213_48	UNPlasmidBox01
UWN_0132	P213_C3-C3	C3C3P213_49	UNPlasmidBox01
UWN_0133	P213_C3-C3	C3C3P213_50	UNPlasmidBox01
UWN_0134		HBlinker_01	UNPlasmidBox01
UWN_0135	O	SL_Ocages_01	UNPlasmidBox01
UWN_0136	O	SL_Ocages_02	UNPlasmidBox01
UWN_0137	O	SL_Ocages_03	UNPlasmidBox01
UWN_0138	O	SL_Ocages_04	UNPlasmidBox01
UWN_0139	O	SL_Ocages_05	UNPlasmidBox01
UWN_0140	O	SL_Ocages_06	UNPlasmidBox01
UWN_0141	O	SL_Ocages_07	UNPlasmidBox01
UWN_0142	T	HT_Tcages_01	UNPlasmidBox01
UWN_0143	T	HT_Tcages_02	UNPlasmidBox01
UWN_0144	T	HT_Tcages_03	UNPlasmidBox01
UWN_0145	T	HT_Tcages_04	UNPlasmidBox01
UWN_0146	T	HT_Tcages_05	UNPlasmidBox01
UWN_0147	T	HT_Tcages_06	UNPlasmidBox01
UWN_0148	T	HT_Tcages_07	UNPlasmidBox01
UWN_0149	T	HT_Tcages_08	UNPlasmidBox01
UWN_0150	F222_D2-D2	SL_D2D2_01	UNPlasmidBox01
UWN_0151	F222_D2-D2	SL_D2D2_02	UNPlasmidBox01
UWN_0152	F222_D2-D2	SL_D2D2_03	UNPlasmidBox01
UWN_0153	F222_D2-D2	SL_D2D2_04	UNPlasmidBox01
UWN_0154	F222_D2-D2	SL_D2D2_05	UNPlasmidBox01
UWN_0155	F222_D2-D2	SL_D2D2_06	UNPlasmidBox01
UWN_0156	F222_D2-D2	SL_D2D2_07	UNPlasmidBox01

UWN_0157	F222_D2-D2	SL_D2D2_08	UNPlasmidBox01
UWN_0158	F222_D2-D2	SL_D2D2_09	UNPlasmidBox01
UWN_0159	F222_D2-D2	SL_D2D2_10	UNPlasmidBox01
UWN_0160	F222_D2-D2	SL_D2D2_11	UNPlasmidBox01
UWN_0161	F222_D2-D2	SL_D2D2_12	UNPlasmidBox01
UWN_0162	F222_D2-D2	SL_D2D2_13	UNPlasmidBox01
UWN_0163	F222_D2-D2	SL_D2D2_14	UNPlasmidBox01
UWN_0164	F23_T-T	T-T_01	UNPlasmidBox01
UWN_0165	F23_T-T	T-T_02	UNPlasmidBox01
UWN_0166	F23_T-T	T-T_03	UNPlasmidBox01
UWN_0167	F23_T-T	T-T_04	UNPlasmidBox01
UWN_0168	F23_T-T	T-T_05	UNPlasmidBox01
UWN_0169	F23_T-T	T-T_06	UNPlasmidBox01
UWN_0170	F23_T-T	T-T_07	UNPlasmidBox01
UWN_0171	F23_T-T	T-T_08	UNPlasmidBox01
UWN_0172	F23_T-T	T-T_09	UNPlasmidBox01
UWN_0173	F23_T-T	T-T_10	UNPlasmidBox01
UWN_0174	F23_T-T	T-T_11	UNPlasmidBox01
UWN_0175	F23_T-T	T-T_12	UNPlasmidBox01
UWN_0176	P432_O-D4_C3	UN_O43-D01_C3	UNPlasmidBox01
UWN_0177	P432_O-D4_D4	UN_O43-D01_D4	UNPlasmidBox01
UWN_0178	P432_O-D4_C3	UN_O43-D02_C3	UNPlasmidBox01
UWN_0179	P432_O-D4_D4	UN_O43-D02_D4	UNPlasmidBox01
UWN_0180	P432_O-D4_C3	UN_O43-D03_C3	UNPlasmidBox01
UWN_0181	P432_O-D4_D4	UN_O43-D03_D4	UNPlasmidBox01
UWN_0182	P432_O-D4_C3	UN_O43-D04_C3	UNPlasmidBox01
UWN_0183	P432_O-D4_D4	UN_O43-D04_D4	UNPlasmidBox01
UWN_0184	P432_O-D4_C3	UN_O43-D05_C3	UNPlasmidBox01
UWN_0185	P432_O-D4_D4	UN_O43-D05_D4	UNPlasmidBox01
UWN_0186	P432_O-D4_C3	UN_O43-D06_C3	UNPlasmidBox01
UWN_0187	P432_O-D4_D4	UN_O43-D06_D4	UNPlasmidBox01
UWN_0188	P432_O-D4_C3	UN_O43-D07_C3	UNPlasmidBox01
UWN_0189	P432_O-D4_D4	UN_O43-D07_D4	UNPlasmidBox01
UWN_0190	P432_O-D4_C3	UN_O43-D08_C3	UNPlasmidBox01
UWN_0191	P432_O-D4_D4	UN_O43-D08_D4	UNPlasmidBox01
UWN_0192	P432_O-D4_C3	UN_O43-D09_C3	UNPlasmidBox01
UWN_0193	P432_O-D4_D4	UN_O43-D09_D4	UNPlasmidBox01
UWN_0194	P432_O-D4_C3	UN_O43-D10_C3	UNPlasmidBox01
UWN_0195	P432_O-D4_D4	UN_O43-D10_D4	UNPlasmidBox01
UWN_0196	P432_O-D4_C3	UN_O43-D11_C3	UNPlasmidBox01

UWN_0197	P432_O-D4_D4	UN_O43-D11_D4	UNPlasmidBox01
UWN_0198	P432_O-D4_C3	UN_O43-D12_C3	UNPlasmidBox01
UWN_0199	P432_O-D4_D4	UN_O43-D12_D4	UNPlasmidBox01
UWN_0200	P432_O-D4_C3	UN_O43-D13_C3	UNPlasmidBox01
UWN_0201	P432_O-D4_C3	UN_O43-D14_C3	UNPlasmidBox01
UWN_0202	P432_O-D4_D4	UN_O43-D14_D4	UNPlasmidBox01
UWN_0203	P432_O-D4_C3	UN_O43-D15_C3	UNPlasmidBox01
UWN_0204	P432_O-D4_D4	UN_O43-D15_D4	UNPlasmidBox01
UWN_0205	P432_O-D4_C3	UN_O43-D16_C3	UNPlasmidBox01
UWN_0206	P432_O-D4_D4	UN_O43-D16_D4	UNPlasmidBox01
UWN_0207	P432_O-D4_C3	UN_O43-D18_C3	UNPlasmidBox01
UWN_0208	P432_O-D4_D4	UN_O43-D18_D4	UNPlasmidBox01
UWN_0209	P432_O-D4_C3	UN_O43-D19_C3	UNPlasmidBox01
UWN_0210	P432_O-D4_C3	UN_O43-D20_C3	UNPlasmidBox01
UWN_0211	P432_O-D4_D4	UN_O43-D20_D4	UNPlasmidBox01
UWN_0212	P432_O-D4_C3	UN_O43-D21_C3	UNPlasmidBox01
UWN_0213	P432_O-D4_D4	UN_O43-D21_D4	UNPlasmidBox01
UWN_0214	P432_O-D4_C3	UN_O43-D22_C3	UNPlasmidBox01
UWN_0215	P432_O-D4_D4	UN_O43-D22_D4	UNPlasmidBox02
UWN_0216	P432_O-D4_C3	UN_O43-D23_C3	UNPlasmidBox02
UWN_0217	P432_O-D4_D4	UN_O43-D23_D4	UNPlasmidBox02
UWN_0218	P432_O-D4_D4	UN_O43-D13_D4	UNPlasmidBox02
UWN_0219	P432_O-D4_D4	UN_O43-D19_D4	UNPlasmidBox02
UWN_0220	$F4_132$_T-D3_D3	T33-15_01	UNPlasmidBox02
UWN_0221	$F4_132$_T-D3_D3	T33-15_02	UNPlasmidBox02
UWN_0222	$F4_132$_T-D3_D3	T33-15_03	UNPlasmidBox02
UWN_0223	$F4_132$_T-D3_C3	T33-15_CB	UNPlasmidBox02
UWN_0224	?432_O-D2_C3	UN_O32_3c2d_01_C3comp	UNPlasmidBox02
UWN_0225	?432_O-D2_D2	UN_O32_3c2d_01_D2comp	UNPlasmidBox02
UWN_0226	?432_O-D2_C3	UN_O32_3c2d_02_C3comp	UNPlasmidBox02
UWN_0227	?432_O-D2_D2	UN_O32_3c2d_02_D2comp	UNPlasmidBox02
UWN_0228	?432_O-D2_C3	UN_O32_3c2d_03_C3comp	UNPlasmidBox02
UWN_0229	?432_O-D2_C3	UN_O32_3c2d_04_C3comp	UNPlasmidBox02
UWN_0230	?432_O-D2_D2	UN_O32_3c2d_04_D2comp	UNPlasmidBox02
UWN_0231	I432_O-D3_C2	UN_O32_3d2c_01_C2comp	UNPlasmidBox02
UWN_0232	I432_O-D3_D3	UN_O32_3d2c_01_D3comp	UNPlasmidBox02
UWN_0233	?432_O-D2_D2	UN_O32_3c2d_03_D2comp	UNPlasmidBox02
UWN_0234	I432_O-D3	OD3_01	
UWN_0235	I432_O-D3	OD3_02	
UWN_0236	I432_O-D3	OD3_03	

UWN_0237	I432_O-D3	OD3_04	
UWN_0238	I432_O-D3	OD3_05	
UWN_0239	I432_O-D3	OD3_06	
UWN_0240	I432_O-D3	OD3_07	
UWN_0241	I432_O-D3	OD3_08	
UWN_0242	I432_O-D3	OD3_09	
UWN_0243	I432_O-D3	OD3_10	
UWN_0244	I432_O-D3	OD3_11	
UWN_0245	I432_O-D3	OD3_12	
UWN_0246	I432_O-D3	O43-38_1wa3dihedral_01	
UWN_0247	I432_O-D3	O43-38_1wa3dihedral_02	
UWN_0248	I432_O-D3	O43-38_1wa3dihedral_03	
UWN_0249	I432_O-D3	O43-38_1wa3dihedral_04	
UWN_0250	I432_O-D3	O43-38_1wa3dihedral_05	
UWN_0251	$F4_i32$_T-D3	T3-AM_1wa3dihedral_01	
UWN_0252	$F4_i32$_T-D3	T3-AM_1wa3dihedral_02	
UWN_0253	$F4_i32$_T-D3	T3-AM_1wa3dihedral_03	
UWN_0254	$F4_i32$_T-D3	T3-AM_1wa3dihedral_04	
UWN_0255	$F4_i32$_T-D3	T3-AM_1wa3dihedral_05	
UWN_0256	$F4_i32$_T-D3	T3-AM_1wa3dihedral_06	
UWN_0257	I432_O-D3	O43-38_D3_01	
UWN_0258	I432_O-D3	O43-38_D3_02	
UWN_0259	I432_O-D3	O43-38_D3_03	
UWN_0260	I432_O-D3	O43-38_D3_04	
UWN_0261	I432_O-D3	O43-38_D3_05	
UWN_0262	I432_O-D3	O43-38_D3_06	
UWN_0263	I432_O-D3	O43-38_D3_07	
UWN_0264	I432_O-D3	O43-38_D3_08	
UWN_0265	I432_O-D3	O43-38_D3_09	
UWN_0266	I432_O-D3	O43-38_D3_10	
UWN_0267	I432_O-D3	O43-38_D3_11	
UWN_0268	I432_O-D3	O43-38_D3_12	
UWN_0269	I432_O-D3	O43-38_D3_13	
UWN_0270	I432_O-D3	O43-38_D3_14	
UWN_0271	I432_O-D3	O43-38_D3_15	
UWN_0272	I432_O-D3	O43-38_D3_16	
UWN_0273	I432_O-D3	O43-38_D3_17	
UWN_0274	I432_O-D3	O43-38_D3_18	
UWN_0275	I432_O-D3	O43-38_D3_19	
UWN_0276	I432_O-D3	O43-38_D3_20	

UWN_0277	I432_O-D3	O43-38_D3_21	
UWN_0278	I432_O-D3	O43-38_D3_22	
UWN_0279	I432_O-D3	O43-38_D3_23	
UWN_0280	I432_O-D3	O43-38_D3_24	
UWN_0281	I432_O-D3	O43-38_D3_25	
UWN_0282	I432_O-D3	O43-38_D3_26	
UWN_0283	I432_O-D3	O43-38_D3_27	
UWN_0284	I432_O-D3	O43-38_D3_28	
UWN_0285	I432_O-D3	O43-38_D3_29	
UWN_0286	I432_O-D3	O43-38_D3_30	
UWN_0287	I432_O-D3	O43-38_D3_31	
UWN_0288	I432_O-D3	O43-38_D3_32	
UWN_0289	I432_O-D3	O43-38_D3_33	
UWN_0290	I432_O-D3	O43-38_D3_34	
UWN_0291	I432_O-D3	O43-38_D3_35	
UWN_0292	I432_O-D3	O43-38_D3_36	
UWN_0293	I432_O-D3	O43-38_D3_37	
UWN_0294	I432_O-D3	O43-38_D3_38	
UWN_0295	I432_O-D3	O43-38_D3_39	
UWN_0296	I432_O-D3	O43-38_D3_40	
UWN_0297	I432_O-D3	O43-38_D3_41	
UWN_0298	I432_O-D3	O43-38_D3_42	
UWN_0299	I432_O-D3	O43-38_D3_43	
UWN_0300	I432_O-D3	O43-38_D3_44	
UWN_0301	I432_O-D3	O43-38_D3_45	
UWN_0302	I432_O-D3	O43-38_D3_46	
UWN_0303	I432_O-D3	O43-38_D3_47	
UWN_0304	I432_O-D3	O43-38_D3_48	
UWN_0305	I432_O-D3	O43-38_D3_49	
UWN_0306	I432_O-D3	O43-38_D3_50	
UWN_0307	I432_O-D3	O43-38_D3_51	
UWN_0308	I432_O-D3	O43-38_D3_52	
UWN_0309	I432_O-D3	O43-38_D3_53	
UWN_0310	I432_O-D3	O43-38_D3_54	
UWN_0311	I432_O-D3	O43-38_D3_55	
UWN_0312	I432_O-D3	O43-38_D3_56	
UWN_0313	I432_O-D3	O43-38_D3_57	
UWN_0314	I432_O-D3	O43-38_D3_58	
UWN_0315	I432_O-D3	O43-38_D3_59	
UWN_0316	I432_O-D3	O43-38_D3_60	

UWN_0317	C3toD3	1wa3dihedral_01	
UWN_0318	C3toD3	1wa3dihedral_02	
UWN_0319	C3toD3	1wa3dihedral_03	
UWN_0320	C3toD3	1wa3dihedral_04	
UWN_0321	C3toD3	1wa3dihedral_05	
UWN_0322	O43	O43-01	
UWN_0323	O43	O43-02	
UWN_0324	O43	O43-03	
UWN_0325	O43	O43-04	
UWN_0326	O43	O43-05	
UWN_0327	O43	O43-06	
UWN_0328	O43	O43-07	
UWN_0329	O43	O43-08	
UWN_0330	O43	O43-09	
UWN_0331	O43	O43-10	
UWN_0332	O43	O43-11	
UWN_0333	O43	O43-12	
UWN_0334	O43	O43-13	
UWN_0335	O43	O43-14	
UWN_0336	O43	O43-15	
UWN_0337	O43	O43-16	
UWN_0338	O43	O43-17	
UWN_0339	O43	O43-18	
UWN_0340	O43	O43-19	
UWN_0341	O43	O43-20	
UWN_0342	O43	O43-21	
UWN_0343	O43	O43-22	
UWN_0344	O43	O43-23	
UWN_0345	O43	O43-24	
UWN_0346	O43	O43-25	
UWN_0347	O43	O43-26	
UWN_0348	O43	O43-27	
UWN_0349	O43	O43-28	
UWN_0350	O43	O43-29	
UWN_0351	O43	O43-30	
UWN_0352	O43	O43-31	
UWN_0353	O43	O43-32	
UWN_0354	O43	O43-33	
UWN_0355	O43	O43-34	
UWN_0356	O43	O43-35	

UWN_0357	O43	O43-36	
UWN_0358	O43	O43-37	
UWN_0359	O43	O43-38	
UWN_0360	O43	O43-39	
UWN_0361	O43	O43-40	
UWN_0362	O43	O43-41	
UWN_0363	O43	O43-42	
UWN_0364	O43	O43-43	
UWN_0365	O43	O43-44	
UWN_0366	O43	O43-45	
UWN_0367	O43	O43-46	
UWN_0368	O43	O43-47	
UWN_0369	O43	O43-48	
UWN_0370	O43	O43-49	
UWN_0371	O43	O43-50	
UWN_0372	O43	O43-51	
UWN_0373	O43	O43-52	
UWN_0374	O43	O43-53	
UWN_0375	O43	O43-54	
UWN_0376	O43	O43-55	
UWN_0377	O43	O43-56	
UWN_0378	O43	O43-57	
UWN_0379	O43	O43-58	
UWN_0380	O43	O43-59	
UWN_0381	O43	O43-60	
UWN_0382	O43	O43-61	
UWN_0383	O43	O43-62	
UWN_0384	O43	O43-63	
UWN_0385	O43	O43-64	
UWN_0386	O43	O43-65	
UWN_0387	O43	O43-66	
UWN_0388	O43	O43-67	
UWN_0389	O43	O43-68	
UWN_0390	O43	O43-69	
UWN_0391	O43	O43-70	
UWN_0392	O43	O43-71	
UWN_0393	O43	O43-72	
UWN_0394	O43	O43-73	
UWN_0395	O43	O43-74	
UWN_0396	O43	O43-75	

UWN_0397	F4$_i$32_T-D3	D3_AL01_1_77C3_noAb_chA_T33F_11_1_0072	UNPlasmidBox03
UWN_0398	F4$_i$32_T-D3	D3_AL01_1_78C3_noAb_chA_T33F_96_1_0017	UNPlasmidBox03
UWN_0399	F4$_i$32_T-D3	D3_AL01_1_C3_3ltjC3_1_T33F_73_1_0053	UNPlasmidBox03
UWN_0400	F4$_i$32_T-D3	D3_AL01_1_C3_un1BH19_1_T33R_50_1_0009	UNPlasmidBox03
UWN_0401	F4$_i$32_T-D3	D3_AL05_1_1BH_69_chA_T33R_29_1_0059	UNPlasmidBox03
UWN_0402	F4$_i$32_T-D3	D3_AL05_1_77C3_noAb_chA_T33R_59_1_0065	UNPlasmidBox03
UWN_0403	F4$_i$32_T-D3	D3_AL05_1_C3_1na0-1_1_T33R_16_1_0087	UNPlasmidBox03
UWN_0404	F4$_i$32_T-D3	D3_AL05_1_C3_1na0-G1_1_T33R_10_1_0005	UNPlasmidBox03
UWN_0405	F4$_i$32_T-D3	D3_AL05_1_C3_1na0-int2_1_T33R_12_1_0009	UNPlasmidBox03
UWN_0406	F4$_i$32_T-D3	D3_AL05_1_C3_3ltjC3_1_T33F_26_1_0059	UNPlasmidBox03
UWN_0407	F4$_i$32_T-D3	D3_AL05_1_C3_3ltjC3_1_T33F_65_1_0089	UNPlasmidBox03
UWN_0408	F4$_i$32_T-D3	D3_AL05_1_C3_yh1BH05_1_T33R_86_1_0050	UNPlasmidBox03
UWN_0409	F4$_i$32_T-D3	D3_AL05_1_rTc3-01_chA_T33R_53_1_0052	UNPlasmidBox03
UWN_0410	F4$_i$32_T-D3	D3_AL08_1_1BH_69_chA_T33R_1_1_0058	UNPlasmidBox03
UWN_0411	F4$_i$32_T-D3	D3_AL08_1_C3_hetC2HFuse-05_chA_T33F_57_1_0085	UNPlasmidBox03
UWN_0412	F4$_i$32_T-D3	D3_AL08_1_C3_TJ41-pm1v2_1_T33F_44_1_0009	UNPlasmidBox03
UWN_0413	F4$_i$32_T-D3	D3_AL08_1_C3_un1BH19_1_T33R_5_1_0004	UNPlasmidBox03
UWN_0414	F4$_i$32_T-D3	D3_AL08_1_rTc3-01_chA_T33R_78_1_0072	UNPlasmidBox03
UWN_0415	F4$_i$32_T-D3	D3_AL08_1_rTc3-09_chA_T33R_53_1_0017	UNPlasmidBox03
UWN_0416	F4$_i$32_T-D3	D3_AL08_1_rTc3-11_chA_T33R_24_1_0010	UNPlasmidBox03
UWN_0417	F4$_i$32_T-D3	D3_AL09_1_C3_1na0-6_1_T33R_35_1_0047	UNPlasmidBox03
UWN_0418	F4$_i$32_T-D3	D3_AL09_1_C3_1na0-int2_1_T33R_59_1_0080	UNPlasmidBox03
UWN_0419	F4$_i$32_T-D3	D3_AL09_1_C3_tj08C3_1_T33R_8_1_0007	UNPlasmidBox03
UWN_0420	F4$_i$32_T-D3	D3_AL09_1_C3_tj18C3_1_T33R_8_1_0083	UNPlasmidBox03
UWN_0421	F4$_i$32_T-D3	D3_AL09_1_C3_yh1BH03_1_T33R_2_1_0016	UNPlasmidBox03
UWN_0422	F4$_i$32_T-D3	D3_AL10_1_77C3_noAb_chA_T33R_18_1_0021	UNPlasmidBox03
UWN_0423	F4$_i$32_T-D3	D3_AL10_1_C3_1na0-3_1_T33R_95_1_0001	UNPlasmidBox03
UWN_0424	F4$_i$32_T-D3	D3_AL10_1_C3_1na0-6_1_T33R_24_1_0039	UNPlasmidBox03
UWN_0425	F4$_i$32_T-D3	D3_AL10_1_C3_1na0-G1_1_T33R_38_1_0027	UNPlasmidBox03
UWN_0426	F4$_i$32_T-D3	D3_AL10_1_C3_1na0-G1_1_T33R_83_1_0045	UNPlasmidBox03
UWN_0427	F4$_i$32_T-D3	D3_AL10_1_C3_TJ41-pm1v2_1_T33F_46_1_0020	UNPlasmidBox03
UWN_0428	F4$_i$32_T-D3	D3_AL10_1_C3_tj71C3-pdb_1_T33F_65_1_0027	UNPlasmidBox03
UWN_0429	F4$_i$32_T-D3	D3_AL11_1_1BH_69_chA_T33R_2_1_0033	UNPlasmidBox03
UWN_0430	F4$_i$32_T-D3	D3_AL11_1_C3_yh1BH05_1_T33R_60_1_0032	UNPlasmidBox03
UWN_0431	F4$_i$32_T-D3	D3_AL13_1_1BH_69_chA_T33R_1_1_0048	UNPlasmidBox03
UWN_0432	F4$_i$32_T-D3	D3_AL13_1_C3_1na0-3_1_T33R_4_1_0095	UNPlasmidBox03
UWN_0433	F4$_i$32_T-D3	D3_AL13_1_C3_1na0-6_1_T33F_81_1_0074	UNPlasmidBox03
UWN_0434	F4$_i$32_T-D3	D3_AL13_1_rTc3-09_chA_T33R_7_1_0042	UNPlasmidBox03
UWN_0435	F4$_i$32_T-D3	D3_AL14_1_C3_1na0-3_1_T33R_85_1_0070	UNPlasmidBox03
UWN_0436	F4$_i$32_T-D3	D3_AL14_1_C3_un1BH19_1_T33R_28_1_0083	UNPlasmidBox03

UWN_0437	I432_O43d	C4_1na0-G1_1_D3_1wa3-33_rot15_O43dF_10_1_0036	UNPlasmidBox03
UWN_0438	I432_O43d	C4_1na0-G1_1_D3_1wa3-33_rot15_O43dF_20_1_0027	UNPlasmidBox03
UWN_0439	I432_O43d	C4_1na0-G1_1_D3_1wa3-33_rot15_O43dF_46_1_0054	UNPlasmidBox03
UWN_0440	I432_O43d	C4_1na0-G1_1_D3_1wa3-44_rot15_O43dF_70_1_0018	UNPlasmidBox03
UWN_0441	I432_O43d	C4_2L8H-12_1_D3_1wa3-33_rot15_O43dF_20_1_0007	UNPlasmidBox03
UWN_0442	I432_O43d	C4_2L8H-12_1_D3_1wa3-33_rot15_O43dF_61_1_0010	UNPlasmidBox03
UWN_0443	I432_O43d	C4_2L8H-12_1_D3_1wa3-33_rot15_O43dF_66_1_0009	UNPlasmidBox03
UWN_0444	I432_O43d	C4_2L8H-12_1_D3_1wa3-33_rot15_O43dR_87_1_0049	UNPlasmidBox03
UWN_0445	I432_O43d	C4_2L8H-12_1_D3_1wa3-44_rot15_O43dF_25_1_0058	UNPlasmidBox03
UWN_0446	I432_O43d	C4_2L8H-12_1_D3_1wa3-44_rot15_O43dF_72_1_0045	
UWN_0447	I432_O43d	C4_5L8H-6_1_D3_1wa3-33_rot15_O43dF_2_1_0048	
UWN_0448	I432_O43d	C4_5L8H-6_1_D3_1wa3-33_rot15_O43dF_37_1_0038	
UWN_0449	I432_O43d	C4_5L8H-6_1_D3_1wa3-33_rot15_O43dF_55_1_0068	
UWN_0450	I432_O43d	C4_5L8H-6_1_D3_1wa3-33_rot15_O43dR_43_1_0052	
UWN_0451	I432_O43d	C4_5L8H-6_1_D3_1wa3-44_rot15_O43dF_16_1_0070	
UWN_0452	I432_O43d	C4_5L8H-6_1_D3_1wa3-44_rot15_O43dF_2_1_0026	
UWN_0453	I432_O43d	C4_5L8H-6_1_D3_1wa3-44_rot15_O43dF_21_1_0063	
UWN_0454	I432_O43d	C4_5L8H-6_1_D3_1wa3-44_rot15_O43dR_44_1_0094	UNPlasmidBox03
UWN_0455	I432_O43d	C4_5L8H-6_1_D3_1wa3-44_rot15_O43dR_83_1_0022	UNPlasmidBox03
UWN_0456	I432_O43d	C4_tj10-G1_1_D3_1wa3-33_rot15_O43dF_8_1_0026	UNPlasmidBox03
UWN_0457	I432_O43d	C4_tj10-G1_1_D3_1wa3-44_rot15_O43dF_18_1_0001	UNPlasmidBox03
UWN_0458	I432_O43d	C4_tj10-G1_1_D3_1wa3-44_rot15_O43dF_26_1_0049	UNPlasmidBox03
UWN_0459	I432_O43d	C4_tj10-G1_1_D3_1wa3-44_rot15_O43dF_32_1_0064	UNPlasmidBox03
UWN_0460	I432_O43d	C4_tj10-G1_1_D3_1wa3-44_rot15_O43dR_7_1_0080	UNPlasmidBox03
UWN_0461	I432_O43d	C4_tpr1-pm3_1_D3_1wa3-33_rot15_O43dF_31_1_0027	UNPlasmidBox03
UWN_0462	I432_O43d	C4_tpr1-pm3_1_D3_1wa3-33_rot15_O43dF_74_1_0021	UNPlasmidBox03
UWN_0463	I432_O43d	C4_tpr1-pm3_1_D3_1wa3-33_rot15_O43dF_91_1_0012	UNPlasmidBox03
UWN_0464	I432_O43d	C4_tpr1-pm3_1_D3_1wa3-44_rot15_O43dF_79_1_0007	UNPlasmidBox03
UWN_0465	C3toD3	C3_1na0-1_m2_chA_C2_dummy_asu_D32R_37_1_0062	UNPlasmidBox03
UWN_0466	C3toD3	C3_1na0-6_m0_chA_C2_dummy_asu_D32F_16_1_0028	UNPlasmidBox03
UWN_0467	C3toD3	C3_1na0-6_m0_chA_C2_dummy_asu_D32F_40_1_0025	UNPlasmidBox03
UWN_0468	C3toD3	C3_1na0-6_m0_chA_C2_dummy_asu_D32F_47_1_0083	UNPlasmidBox03
UWN_0469	C3toD3	C3_1na0-6_m0_chA_C2_dummy_asu_D32R_13_1_0057	UNPlasmidBox03
UWN_0470	C3toD3	C3_1na0-G1_m0_chA_C2_dummy_asu_D32F_26_1_0005	UNPlasmidBox03
UWN_0471	C3toD3	C3_1na0-G1_m0_chA_C2_dummy_asu_D32F_29_1_0078	UNPlasmidBox03
UWN_0472	C3toD3	C3_1na0-G1_m0_chA_C2_dummy_asu_D32F_53_1_0069	UNPlasmidBox03
UWN_0473	C3toD3	C3_1na0-G1_m0_chA_C2_dummy_asu_D32F_83_1_0023	UNPlasmidBox03
UWN_0474	C3toD3	C3_1na0-G1_m0_chA_C2_dummy_asu_D32F_84_1_0046	UNPlasmidBox03
UWN_0475	C3toD3	C3_1na0-G1_m0_chA_C2_dummy_asu_D32F_84_1_0070	UNPlasmidBox03
UWN_0476	C3toD3	C3_1na0-G1_m0_chA_C2_dummy_asu_D32F_92_1_0086	UNPlasmidBox03

UWN_0477	C3toD3	C3_1na0-G1_m0_chA_C2_dummy_asu_D32R_4_1_0064	UNPlasmidBox03
UWN_0478	C3toD3	C3_1na0-G1_m0_chA_C2_dummy_asu_D32R_4_1_0075	UNPlasmidBox03
UWN_0479	C3toD3	C3_1na0-G1_m0_chA_C2_dummy_asu_D32R_5_1_0027	UNPlasmidBox03
UWN_0480	C3toD3	C3_1na0-G1_m0_chA_C2_dummy_asu_D32R_53_1_0063	UNPlasmidBox03
UWN_0481	C3toD3	C3_1na0-G1_m2_chA_C2_dummy_asu_D32R_25_1_0037	
UWN_0482	C3toD3	C3_1na0-int2-r4_m5_chA_C2_dummy_asu_D32F_25_1_0014	UNPlasmidBox04
UWN_0483	C3toD3	C3_1na0-int2-r4_m5_chA_C2_dummy_asu_D32F_34_1_0023	UNPlasmidBox04
UWN_0484	C3toD3	C3_1na0-int2-r4_m6_chA_C2_dummy_asu_D32F_21_1_0052	UNPlasmidBox04
UWN_0485	C3toD3	C3_3ltj-int1_m0_chA_C2_dummy_asu_D32R_52_1_0090	UNPlasmidBox04
UWN_0486	C3toD3	C3_3ltj-int1_m0_chA_C2_dummy_asu_D32R_7_1_0029	UNPlasmidBox04
UWN_0487	C3toD3	C3_3ltj-int1_m1_chA_C2_dummy_asu_D32F_21_1_0064	UNPlasmidBox04
UWN_0488	C3toD3	C3_3ltj-int1_m1_chA_C2_dummy_asu_D32R_86_1_0052	UNPlasmidBox04
UWN_0489	C3toD3	C3_3ltj-int1_m2_chA_C2_dummy_asu_D32R_40_1_0024	UNPlasmidBox04
UWN_0490	C3toD3	C3_hetC2HFuse-04_m0_chA_C2_dummy_asu_D32F_2_1_0003	UNPlasmidBox04
UWN_0491	C3toD3	C3_hetC2HFuse-04_m0_chA_C2_dummy_asu_D32F_46_1_0055	UNPlasmidBox04
UWN_0492	C3toD3	C3_hetC2HFuse-04_m0_chA_C2_dummy_asu_D32R_41_1_0016	UNPlasmidBox04
UWN_0493	C3toD3	C3_hetC2HFuse-04_m0_chA_C2_dummy_asu_D32R_42_1_0042	UNPlasmidBox04
UWN_0494	C3toD3	C3_hetC2HFuse-04_m0_chA_C2_dummy_asu_D32R_77_1_0004	UNPlasmidBox04
UWN_0495	C3toD3	C3_hetC2HFuse-05_m0_chA_C2_dummy_asu_D32F_10_1_0010	UNPlasmidBox04
UWN_0496	C3toD3	C3_hetC2HFuse-05_m0_chA_C2_dummy_asu_D32F_37_1_0024	UNPlasmidBox04
UWN_0497	C3toD3	C3_hetC2HFuse-05_m0_chA_C2_dummy_asu_D32F_38_1_0023	UNPlasmidBox04
UWN_0498	C3toD3	C3_hetC2HFuse-05_m0_chA_C2_dummy_asu_D32F_47_1_0024	UNPlasmidBox04
UWN_0499	C3toD3	C3_hetC2HFuse-05_m0_chA_C2_dummy_asu_D32F_47_1_0041	UNPlasmidBox04
UWN_0500	C3toD3	C3_hetC2HFuse-05_m0_chA_C2_dummy_asu_D32F_75_1_0029	UNPlasmidBox04
UWN_0501	C3toD3	C3_hetC2HFuse-06_m0_chA_C2_dummy_asu_D32F_19_1_0071	UNPlasmidBox04
UWN_0502	C3toD3	C3_hetC2HFuse-06_m0_chA_C2_dummy_asu_D32R_56_1_0017	UNPlasmidBox04
UWN_0503	C3toD3	C3_hetC2HFuse-06_m0_chA_C2_dummy_asu_D32R_57_1_0005	UNPlasmidBox04
UWN_0504	C3toD3	C3_hetC2HFuse-06_m0_chA_C2_dummy_asu_D32R_80_1_0041	UNPlasmidBox04
UWN_0505	C3toD3	C3_hetC2HFuse-06_m0_chA_C2_dummy_asu_D32R_93_1_0076	UNPlasmidBox04
UWN_0506	C3toD3	C3_hetC2HFuse-07_m0_chA_C2_dummy_asu_D32F_66_1_0027	UNPlasmidBox04
UWN_0507	C3toD3	C3_hetC2HFuse-07_m0_chA_C2_dummy_asu_D32F_75_1_0037	UNPlasmidBox04
UWN_0508	C3toD3	C3_hetC2HFuse-07_m0_chA_C2_dummy_asu_D32F_76_1_0036	UNPlasmidBox04
UWN_0509	C3toD3	C3_hetC2HFuse-07_m0_chA_C2_dummy_asu_D32F_76_1_0042	UNPlasmidBox04

UWN_0510	C3toD3	C3_hetC2HFuse-07_m0_chA_C2_dummy_asu_D32R_85_1_0031	UNPlasmidBox04
UWN_0511	C3toD3	C3_hetC2HFuse-07_m0_chA_C2_dummy_asu_D32R_85_1_0046	UNPlasmidBox04
UWN_0512	C3toD3	C3_tj07-V03_m2_chA_C2_dummy_asu_D32R_17_1_0089	UNPlasmidBox04
UWN_0513	C3toD3	C3_tj07-V03_m2_chA_C2_dummy_asu_D32R_6_1_0044	UNPlasmidBox04
UWN_0514	C3toD3	C3_tj08-V13_m0_chA_C2_dummy_asu_D32F_12_1_0029	UNPlasmidBox04
UWN_0515	C3toD3	C3_tj08-V13_m0_chA_C2_dummy_asu_D32F_36_1_0038	UNPlasmidBox04
UWN_0516	C3toD3	C3_tj08-V13_m0_chA_C2_dummy_asu_D32F_39_1_0012	UNPlasmidBox04
UWN_0517	C3toD3	C3_tj08-V13_m0_chA_C2_dummy_asu_D32R_65_1_0033	UNPlasmidBox04
UWN_0518	C4toD4	C4_1na0-G1_m0_chA_C2_dummy_asu_D42F_58_1_0091	UNPlasmidBox04
UWN_0519	C4toD4	C4_1na0-G1_m0_chA_C2_dummy_asu_D42R_67_1_0072	UNPlasmidBox04
UWN_0520	C4toD4	C4_1na0-G1_m0_chA_C2_dummy_asu_D42R_75_1_0081	UNPlasmidBox04
UWN_0521	C4toD4	C4_1na0-G1_m1_chA_C2_dummy_asu_D42F_12_1_0035	UNPlasmidBox04
UWN_0522	C4toD4	C4_1na0-G1_m1_chA_C2_dummy_asu_D42F_47_1_0004	UNPlasmidBox04
UWN_0523	C4toD4	C4_1na0-G1_m2_chA_C2_dummy_asu_D42F_16_1_0049	UNPlasmidBox04
UWN_0524	C4toD4	C4_1na0-G1_m2_chA_C2_dummy_asu_D42F_32_1_0001	UNPlasmidBox04
UWN_0525	C4toD4	C4_ank4-2_m0_chA_C2_dummy_asu_D42F_40_1_0003	UNPlasmidBox04
UWN_0526	C4toD4	C4_ank4-2_m0_chA_C2_dummy_asu_D42F_51_1_0001	UNPlasmidBox04
UWN_0527	C4toD4	C4_ank4-2_m0_chA_C2_dummy_asu_D42F_53_1_0051	UNPlasmidBox04
UWN_0528	C4toD4	C4_ank4-2_m0_chA_C2_dummy_asu_D42F_67_1_0097	UNPlasmidBox04
UWN_0529	C4toD4	C4_ank4-2_m0_chA_C2_dummy_asu_D42F_89_1_0013	UNPlasmidBox04
UWN_0530	C4toD4	C4_ank4-2_m0_chA_C2_dummy_asu_D42R_21_1_0010	UNPlasmidBox04
UWN_0531	C4toD4	C4_ank4-2_m0_chA_C2_dummy_asu_D42R_35_1_0015	UNPlasmidBox04
UWN_0532	C4toD4	C4_ank4-2_m0_chA_C2_dummy_asu_D42R_71_1_0028	UNPlasmidBox04
UWN_0533	C4toD4	C4_ank4-2_m0_chA_C2_dummy_asu_D42R_77_1_0029	UNPlasmidBox04
UWN_0534	C4toD4	C4_ank4-2_m2_chA_C2_dummy_asu_D42F_30_1_0022	UNPlasmidBox04
UWN_0535	C4toD4	C4_ank4-2_m2_chA_C2_dummy_asu_D42R_37_1_0009	UNPlasmidBox04
UWN_0536	C4toD4	C4_ank4-2_m2_chA_C2_dummy_asu_D42R_42_1_0055	UNPlasmidBox04
UWN_0537	C4toD4	C4_ank4-2_m2_chA_C2_dummy_asu_D42R_43_1_0007	UNPlasmidBox04
UWN_0538	C4toD4	C4_ank4-2_m2_chA_C2_dummy_asu_D42R_47_1_0026	UNPlasmidBox04
UWN_0539	C4toD4	C4_ank4-2_m2_chA_C2_dummy_asu_D42R_53_1_0013	UNPlasmidBox04
UWN_0540	C4toD4	C4_ank4-2_m2_chA_C2_dummy_asu_D42R_72_1_0033	UNPlasmidBox04
UWN_0541	C4toD4	C4_ank4-2_m2_chA_C2_dummy_asu_D42R_73_1_0013	UNPlasmidBox04
UWN_0542	C4toD4	C4_tpr1-pm3_m0_chA_C2_dummy_asu_D42F_19_1_0016	UNPlasmidBox04
UWN_0543	C4toD4	C4_tpr1-pm3_m0_chA_C2_dummy_asu_D42F_21_1_0005	UNPlasmidBox04
UWN_0544	C4toD4	C4_tpr1-pm3_m0_chA_C2_dummy_asu_D42R_20_1_0006	UNPlasmidBox04
UWN_0545	C4toD4	C4_tpr1-pm3_m0_chA_C2_dummy_asu_D42R_21_1_0060	UNPlasmidBox04
UWN_0546	C4toD4	C4_tpr1-pm3_m0_chA_C2_dummy_asu_D42R_21_1_0072	UNPlasmidBox04
UWN_0547	C4toD4	C4_tpr1-pm3_m0_chA_C2_dummy_asu_D42R_8_1_0024	UNPlasmidBox04

UWN_0548	C4toD4	C4_tpr1-pm3_m2_chA_C2_dummy_asu_D42R_33_1_0023	UNPlasmidBox04
UWN_0549	C4toD4	C4_tpr1-pm3_m2_chA_C2_dummy_asu_D42R_33_1_0042	UNPlasmidBox04
UWN_0550	C4toD4	C4_tpr1-pm3_m3_chA_C2_dummy_asu_D42F_9_1_0099	UNPlasmidBox04
UWN_0551	C4toD4	C4_tpr1-pm3_m3_chA_C2_dummy_asu_D42R_37_1_0062	UNPlasmidBox04
UWN_0552	C2toD2	C2_ank1-G3-r5_full_m0_chA_C2_dummy_asu_D22F_41_1_0068	UNPlasmidBox04
UWN_0553	C2toD2	C2_ank1-G3-r5_full_m0_chA_C2_dummy_asu_D22F_66_1_0037	UNPlasmidBox04
UWN_0554	C2toD2	C2_ank1-G3-r5_full_m0_chA_C2_dummy_asu_D22R_53_1_0043	UNPlasmidBox04
UWN_0555	C2toD2	C2_ank1-G3-r5_full_m0_chA_C2_dummy_asu_D22R_59_1_0019	UNPlasmidBox04
UWN_0556	C2toD2	C2_ank1-G3-r5_full_m1_chA_C2_dummy_asu_D22R_16_1_0016	UNPlasmidBox04
UWN_0557	C2toD2	C2_ank1-G3-r5_full_m1_chA_C2_dummy_asu_D22R_32_1_0081	UNPlasmidBox04
UWN_0558	C2toD2	C2_ank1-G3-r5_full_m2_chA_C2_dummy_asu_D22F_30_1_0030	UNPlasmidBox04
UWN_0559	C2toD2	C2_ank1-G3-r5_full_m2_chA_C2_dummy_asu_D22R_36_1_0047	UNPlasmidBox04
UWN_0560	C2toD2	C2_ank1-G3-r5_full_m3_chA_C2_dummy_asu_D22F_8_1_0017	UNPlasmidBox04
UWN_0561	C2toD2	C2_ank1-G3-r5_full_m3_chA_C2_dummy_asu_D22R_39_1_0029	UNPlasmidBox04
UWN_0562	C2toD2	C2_bex2-G2_m0_chA_C2_dummy_asu_D22F_17_1_0005	UNPlasmidBox04
UWN_0563	C2toD2	C2_bex2-G2_m0_chA_C2_dummy_asu_D22F_18_1_0085	UNPlasmidBox05
UWN_0564	C2toD2	C2_bex2-G2_m0_chA_C2_dummy_asu_D22F_24_1_0060	UNPlasmidBox05
UWN_0565	C2toD2	C2_bex2-G2_m0_chA_C2_dummy_asu_D22R_2_1_0004	UNPlasmidBox05
UWN_0566	C2toD2	C2_bex2-G2_mC1_chA_C2_dummy_asu_D22F_11_1_0079	UNPlasmidBox05
UWN_0567	C2toD2	C2_bex2-G2_mC1_chA_C2_dummy_asu_D22F_21_1_0056	UNPlasmidBox05
UWN_0568	C2toD2	C2_bex2-G2_mC1_chA_C2_dummy_asu_D22F_22_1_0074	UNPlasmidBox05
UWN_0569	C2toD2	C2_bex2-G2_mC1_chA_C2_dummy_asu_D22R_29_1_0076	UNPlasmidBox05
UWN_0570	C2toD2	C2_bex2-G2_mC1_chA_C2_dummy_asu_D22R_31_1_0089	UNPlasmidBox05
UWN_0571	C2toD2	C2_bex2-G2_mC2_chA_C2_dummy_asu_D22F_11_1_0071	UNPlasmidBox05
UWN_0572	C2toD2	C2_bex2-G2_mC2_chA_C2_dummy_asu_D22F_6_1_0024	UNPlasmidBox05
UWN_0573	C2toD2	C2_bex2-G2_mC3_chA_C2_dummy_asu_D22F_12_1_0050	UNPlasmidBox05
UWN_0574	C2toD2	C2_bex2-G2_mC4_chA_C2_dummy_asu_D22F_2_1_0059	UNPlasmidBox05
UWN_0575	C2toD2	C2_bex2-G2_mC4_chA_C2_dummy_asu_D22F_4_1_0070	UNPlasmidBox05
UWN_0576	C2toD2	C2_bex2-G2_mN1_chA_C2_dummy_asu_D22F_23_1_0034	UNPlasmidBox05
UWN_0577	C2toD2	C2_bex2-G2_mN1_chA_C2_dummy_asu_D22R_15_1_0015	UNPlasmidBox05
UWN_0578	C2toD2	C2_bex2-G2_mN1_chA_C2_dummy_asu_D22R_19_1_0045	UNPlasmidBox05
UWN_0579	C2toD2	C2_bex2-G2_mN2_chA_C2_dummy_asu_D22F_1_1_0034	UNPlasmidBox05
UWN_0580	C2toD2	C2_bex2-G2_mN2_chA_C2_dummy_asu_D22F_19_1_0016	UNPlasmidBox05

UWN_0581	C2toD2	C2_bex2-G2_mN2_chA_C2_dummy_asu_D22F_27_1_0043	UNPlasmidBox05
UWN_0582	C2toD2	C2_bex2-G2_mN2_chA_C2_dummy_asu_D22R_9_1_0036	UNPlasmidBox05
UWN_0583	C2toD2	C2_bex2-G2_mN3_chA_C2_dummy_asu_D22F_7_1_0031	UNPlasmidBox05
UWN_0584	C2toD2	C2_bex2-G2_mN3_chA_C2_dummy_asu_D22R_21_1_0018	UNPlasmidBox05
UWN_0585	C2toD2	C2_bex2-G2_mN3_chA_C2_dummy_asu_D22R_21_1_0084	UNPlasmidBox05
UWN_0586	C2toD2	C2_bex2-G2_mN4_chA_C2_dummy_asu_D22R_46_1_0001	UNPlasmidBox05
UWN_0587	C2toD2	C2_bex2-G2_mN4_chA_C2_dummy_asu_D22R_48_1_0019	UNPlasmidBox05
UWN_0588	C2toD2	C2_4pww_m0_chA_C2_dummy_asu_D22R_11_1_0098	UNPlasmidBox05
UWN_0589	C2toD2	C2_4pww_m0_chA_C2_dummy_asu_D22R_5_1_0081	UNPlasmidBox05
UWN_0590	C2toD2	C2_REFS07_m0_chA_C2_dummy_asu_D22F_14_1_0077	UNPlasmidBox05
UWN_0591	C2toD2	C2_REFS07_m0_chA_C2_dummy_asu_D22R_29_1_0031	UNPlasmidBox05
UWN_0592	C2toD2	C2_REFS07_m0_chA_C2_dummy_asu_D22R_3_1_0019	UNPlasmidBox05
UWN_0593	C2toD2	C2_REFS07_m1_chA_C2_dummy_asu_D22F_20_1_0060	UNPlasmidBox05
UWN_0594	C2toD2	C2_REFS07_m2_chA_C2_dummy_asu_D22F_16_1_0090	UNPlasmidBox05
UWN_0595	C2toD2	C2_REFS07_m2_chA_C2_dummy_asu_D22F_21_1_0042	UNPlasmidBox05
UWN_0596	C2toD2	C2_REFS07_m2_chA_C2_dummy_asu_D22F_29_1_0025	UNPlasmidBox05
UWN_0597	C2toD2	C2_REFS07_m2_chA_C2_dummy_asu_D22F_6_1_0012	UNPlasmidBox05
UWN_0598	C2toD2	C2_REFS07_m2_chA_C2_dummy_asu_D22R_23_1_0022	UNPlasmidBox05
UWN_0599	C2toD2	C2_REFS07_m2_chA_C2_dummy_asu_D22R_45_1_0016	UNPlasmidBox05
UWN_0600	C2toD2	C2_REFS07_m3_chA_C2_dummy_asu_D22F_24_1_0079	UNPlasmidBox05
UWN_0601	C2toD2	C2_REFS07_m3_chA_C2_dummy_asu_D22F_45_1_0044	UNPlasmidBox05
UWN_0602	C2toD2	C2_REFS07_m3_chA_C2_dummy_asu_D22F_5_1_0007	UNPlasmidBox05
UWN_0603	C2toD2	C2_REFS07_m3_chA_C2_dummy_asu_D22R_41_1_0100	UNPlasmidBox05
UWN_0604	C2toD2	C2_REFS07_m4_chA_C2_dummy_asu_D22F_10_1_0010	UNPlasmidBox05
UWN_0605	C2toD2	C2_REFS07_m4_chA_C2_dummy_asu_D22F_26_1_0097	UNPlasmidBox05
UWN_0606	C2toD2	C2_REFS07_m4_chA_C2_dummy_asu_D22F_50_1_0021	UNPlasmidBox05
UWN_0607	C2toD2	C2_REFS07_m4_chA_C2_dummy_asu_D22R_5_1_0044	UNPlasmidBox05
UWN_0608	C2toD2	C2_REFS11_m0_chA_C2_dummy_asu_D22F_36_1_0063	UNPlasmidBox05
UWN_0609	C2toD2	C2_REFS11_m0_chA_C2_dummy_asu_D22F_62_1_0079	UNPlasmidBox05
UWN_0610	C2toD2	C2_REFS11_m0_chA_C2_dummy_asu_D22F_63_1_0081	UNPlasmidBox05
UWN_0611	C3toD3	C3_77-noAb_m0_chA_C2_dummy_asu_D32R_71_1_0067	UNPlasmidBox05
UWN_0612	C3toD3	C3_77-noAb_m2_chA_C2_dummy_asu_D32F_14_1_0099	UNPlasmidBox05
UWN_0613	C3toD3	C3_77-noAb_m4_chA_C2_dummy_asu_D32F_20_1_0054	UNPlasmidBox05
UWN_0614	C3toD3	C3_77-noAb_m4_chA_C2_dummy_asu_D32R_25_1_0007	UNPlasmidBox05
UWN_0615	C3toD3	C3_77-noAb_m4_chA_C2_dummy_asu_D32R_39_1_0009	UNPlasmidBox05
UWN_0616	C3toD3	C3_78-noAb_m0_chA_C2_dummy_asu_D32R_13_1_0068	UNPlasmidBox05
UWN_0617	C3toD3	C3_78-noAb_m4_chA_C2_dummy_asu_D32R_7_1_0024	UNPlasmidBox05
UWN_0618	C3toD3	C3_UN22_m0_chA_C2_dummy_asu_D32F_66_1_0013	UNPlasmidBox05
UWN_0619	C3toD3	C3_UN22_m0_chA_C2_dummy_asu_D32F_67_1_0004	UNPlasmidBox05

UWN_0620	C3toD3	C3_UN22_m0_chA_C2_dummy_asu_D32F_9_1_0033	UNPlasmidBox05
UWN_0621	C3toD3	C3_UN22_m0_chA_C2_dummy_asu_D32R_12_1_0041	UNPlasmidBox05
UWN_0622	C3toD3	C3_UN22_m0_chA_C2_dummy_asu_D32R_32_1_0024	UNPlasmidBox05
UWN_0623	C3toD3	C3_UN22_m0_chA_C2_dummy_asu_D32R_33_1_0060	UNPlasmidBox05
UWN_0624	C3toD3	C3_UN22_m0_chA_C2_dummy_asu_D32R_87_1_0062	UNPlasmidBox05
UWN_0625	C3toD3	C3_UN22_m1_chA_C2_dummy_asu_D32F_15_1_0092	UNPlasmidBox05
UWN_0626	C3toD3	C3_UN22_m1_chA_C2_dummy_asu_D32F_21_1_0018	UNPlasmidBox05
UWN_0627	C3toD3	C3_UN22_m1_chA_C2_dummy_asu_D32F_63_1_0001	UNPlasmidBox05
UWN_0628	C3toD3	C3_UN22_m1_chA_C2_dummy_asu_D32R_7_1_0054	UNPlasmidBox05
UWN_0629	C3toD3	C3_UN22_m1_chA_C2_dummy_asu_D32R_75_1_0036	UNPlasmidBox05
UWN_0630	C3toD3	C3_UN22_m2_chA_C2_dummy_asu_D32F_3_1_0063	UNPlasmidBox05
UWN_0631	C3toD3	C3_UN22_m2_chA_C2_dummy_asu_D32R_22_1_0011	UNPlasmidBox05
UWN_0632	C3toD3	C3_UN22_m2_chA_C2_dummy_asu_D32R_7_1_0007	UNPlasmidBox05
UWN_0633	C3toD3	C3_UN22_m3_chA_C2_dummy_asu_D32R_59_1_0008	UNPlasmidBox05
UWN_0634	C3toD3	C3_UN22_m3_chA_C2_dummy_asu_D32R_63_1_0097	UNPlasmidBox05
UWN_0635	C3toD3	C3_UN22_m4_chA_C2_dummy_asu_D32F_29_1_0100	UNPlasmidBox05
UWN_0636	C3toD3	C3_UN22_m4_chA_C2_dummy_asu_D32R_33_1_0037	UNPlasmidBox05
UWN_0637	C4toD4	C4_RD01_m0_chA_C2_dummy_asu_D42F_80_1_0048	UNPlasmidBox05
UWN_0638	C4toD4	C4_RD01_m0_chA_C2_dummy_asu_D42R_98_1_0009	UNPlasmidBox05
UWN_0639	C4toD4	C4_RD01_m1_chA_C2_dummy_asu_D42F_60_1_0005	UNPlasmidBox05
UWN_0640	C4toD4	C4_RD01_m1_chA_C2_dummy_asu_D42F_99_1_0061	UNPlasmidBox05
UWN_0641	C4toD4	C4_RD01_m1_chA_C2_dummy_asu_D42R_1_1_0044	UNPlasmidBox05
UWN_0642	C4toD4	C4_RD01_m1_chA_C2_dummy_asu_D42R_27_1_0087	UNPlasmidBox05
UWN_0643	C4toD4	C4_RD01_m1_chA_C2_dummy_asu_D42R_74_1_0056	UNPlasmidBox05
UWN_0644	C4toD4	C4_RD01_m2_chA_C2_dummy_asu_D42F_12_1_0073	UNPlasmidBox05
UWN_0645	C4toD4	C4_RD01_m2_chA_C2_dummy_asu_D42F_39_1_0040	UNPlasmidBox05
UWN_0646	C4toD4	C4_RD01_m2_chA_C2_dummy_asu_D42R_36_1_0015	UNPlasmidBox05
UWN_0647	C4toD4	C4_RD01_m3_chA_C2_dummy_asu_D42R_41_1_0015	UNPlasmidBox05
UWN_0648	C4toD4	C4_RD01_m3_chA_C2_dummy_asu_D42R_92_1_0043	UNPlasmidBox05
UWN_0649	C4toD4	C4_RD01_m4_chA_C2_dummy_asu_D42R_13_1_0002	UNPlasmidBox05
UWN_0650	C4toD4	C4_RD01_m4_chA_C2_dummy_asu_D42R_39_1_0076	UNPlasmidBox05
UWN_0651	C4toD4	C4_RD01_m4_chA_C2_dummy_asu_D42R_78_1_0074	UNPlasmidBox05
UWN_0652	C4toD4	C4_RD01_m4_chA_C2_dummy_asu_D42R_92_1_0030	UNPlasmidBox05
UWN_0653	C4toD4	C4_RD01_m5_chA_C2_dummy_asu_D42F_90_1_0016	UNPlasmidBox05
UWN_0654	C4toD4	C4_RD01_m6_chA_C2_dummy_asu_D42F_27_1_0047	UNPlasmidBox05
UWN_0655	C4toD4	C4_RD01_m6_chA_C2_dummy_asu_D42F_64_1_0070	UNPlasmidBox05
UWN_0656	C4toD4	C4_RD01_m6_chA_C2_dummy_asu_D42F_85_1_0094	UNPlasmidBox05
UWN_0657	C4toD4	C4_RD01_m6_chA_C2_dummy_asu_D42R_17_1_0063	UNPlasmidBox05
UWN_0658	C4toD4	C4_RD01_m6_chA_C2_dummy_asu_D42R_82_1_0001	UNPlasmidBox05
UWN_0659	C4toD4	C4_RD01_m7_chA_C2_dummy_asu_D42F_98_1_0008	UNPlasmidBox05

UWN_0660	C4toD4	C4_RD01_m7_chA_C2_dummy_asu_D42R_20_1_0006	UNPlasmidBox05
UWN_0661	C4toD4	C4_RD01_m7_chA_C2_dummy_asu_D42R_31_1_0042	UNPlasmidBox05
UWN_0662	I432_O23d	C2_3L6H-4_1_D3_1wa3-44_rot15_O23dF_33_1_0091	UNPlasmidBox06
UWN_0663	I432_O23d	C2_ank1-G3_1_D3_1wa3-33_rot15_O23dF_26_1_0003	UNPlasmidBox06
UWN_0664	I432_O23d	C2_ank1-G3_1_D3_1wa3-33_rot15_O23dF_60_1_0042	UNPlasmidBox06
UWN_0665	I432_O23d	C2_ank1-G3_1_D3_1wa3-33_rot15_O23dF_76_1_0019	UNPlasmidBox06
UWN_0666	I432_O23d	C2_ank3-1_1_D3_1wa3-33_rot15_O23dF_5_1_0067	
UWN_0667	I432_O23d	C2_ank3-2_1_D3_1wa3-33_rot15_O23dF_38_1_0042	UNPlasmidBox06
UWN_0668	I432_O23d	C2_ank3-2_1_D3_1wa3-44_rot15_O23dR_20_1_0058	UNPlasmidBox06
UWN_0669	I432_O23d	C2_frx-2_1_D3_1wa3-33_rot15_O23dF_1_1_0012	UNPlasmidBox06
UWN_0670	I432_O23d	C2_frx-2_1_D3_1wa3-44_rot15_O23dF_19_1_0007	UNPlasmidBox06
UWN_0671	I432_O23d	C2_frx-2_1_D3_1wa3-44_rot15_O23dF_4_1_0010	UNPlasmidBox06
UWN_0672	I432_O23d	C2_frx-2_1_D3_1wa3-44_rot15_O23dF_64_1_0031	UNPlasmidBox06
UWN_0673	I432_O23d	C2_REFS10_1_D3_1wa3-33_rot15_O23dF_68_1_0017	
UWN_0674	I432_O23d	C2_REFS10_1_D3_1wa3-44_rot15_O23dF_43_1_0004	UNPlasmidBox06
UWN_0675	I432_O23d	C2_REFS10_1_D3_1wa3-44_rot15_O23dF_92_1_0069	UNPlasmidBox06
UWN_0676	I432_O23d	C2_REFS11_1_D3_1wa3-33_rot15_O23dF_31_1_0002	
UWN_0677	I432_O23d	C2_REFS11_1_D3_1wa3-33_rot15_O23dF_54_1_0002	
UWN_0678	I432_O23d	C2_REFS11_1_D3_1wa3-33_rot15_O23dF_94_1_0023	UNPlasmidBox06
UWN_0679	I432_O23d	C2_REFS11_1_D3_1wa3-44_rot15_O23dF_87_1_0013	
UWN_0680	I432_O23d	C2_REFS13_1_D3_1wa3-33_rot15_O23dF_79_1_0001	UNPlasmidBox06
UWN_0681	I432_O23d	C2_REFS13_1_D3_1wa3-33_rot15_O23dR_65_1_0040	UNPlasmidBox06
UWN_0682	I432_O23d	C2_REFS13_1_D3_1wa3-44_rot15_O23dF_73_1_0031	UNPlasmidBox06
UWN_0683	I432_O23d	C2_REFS13_1_D3_1wa3-44_rot15_O23dR_3_1_0017	UNPlasmidBox06
UWN_0684	I432_O23d	C2_rop20_1_D3_1wa3-33_rot15_O23dR_51_1_0061	UNPlasmidBox06
UWN_0685	I432_O23d	C2_rop20_1_D3_1wa3-44_rot15_O23dF_3_1_0044	UNPlasmidBox06
UWN_0686	I432_O23d	C2_rop20_1_D3_1wa3-44_rot15_O23dR_60_1_0042	
UWN_0687	I432_O23d	C2_rop4_1_D3_1wa3-33_rot15_O23dF_7_1_0060	UNPlasmidBox06
UWN_0688	I432_O23d	C2_rop4_1_D3_1wa3-33_rot15_O23dR_68_1_0012	UNPlasmidBox06
UWN_0689	I432_O23d	C2_rop4_1_D3_1wa3-44_rot15_O23dF_13_1_0023	UNPlasmidBox06
UWN_0690	I432_O23d	C2_tj18-V03_1_D3_1wa3-44_rot15_O23dF_63_1_0014	UNPlasmidBox06
UWN_0691	I432_O23d	C2_tj79-V01_1_D3_1wa3-33_rot15_O23dF_29_1_0024	UNPlasmidBox06
UWN_0692	I432_O23d	C2_tj79-V01_1_D3_1wa3-33_rot15_O23dF_78_1_0020	
UWN_0693	I432_O23d	C2_tj81-V39_1_D3_1wa3-44_rot15_O23dF_6_1_0015	UNPlasmidBox06
UWN_0694	I432_O23d	C2_tj81-V39_1_D3_1wa3-44_rot15_O23dF_97_1_0042	
UWN_0695	I432_O23d	C2_ZC14_1_D3_1wa3-33_rot15_O23dF_30_1_0084	UNPlasmidBox06
UWN_0696	I432_O23d	C2_ZC14_1_D3_1wa3-44_rot15_O23dF_61_1_0085	UNPlasmidBox06
UWN_0697	I432_O23d	C2_ZC14_1_D3_1wa3-44_rot15_O23dF_78_1_0090	UNPlasmidBox06
UWN_0698	I432_O23d	C2_ZC14_1_D3_1wa3-44_rot15_O23dR_82_1_0045	UNPlasmidBox06
UWN_0699	I432_O23d	C2_ZC15_1_D3_1wa3-33_rot15_O23dF_50_1_0082	UNPlasmidBox06

UWN_0700	I432_O23d	C2_ZC15_1_D3_1wa3-33_rot15_O23dF_5_1_0057	UNPlasmidBox06
UWN_0701	I432_O23d	C2_ZC15_1_D3_1wa3-44_rot15_O23dF_50_1_0021	UNPlasmidBox06
UWN_0702	I432_O23d	C2_ZC15_1_D3_1wa3-44_rot15_O23dR_51_1_0097	UNPlasmidBox06
UWN_0703	I432_O23d	C2_ZC16_1_D3_1wa3-33_rot15_O23dF_2_1_0067	UNPlasmidBox06
UWN_0704	I432_O23d	C2_ZC16_1_D3_1wa3-33_rot15_O23dF_31_1_0067	UNPlasmidBox06
UWN_0705	I432_O23d	C2_ZC16_1_D3_1wa3-44_rot15_O23dF_4_1_0056	UNPlasmidBox06
UWN_0706	I432_O23d	C2_ZC16_1_D3_1wa3-44_rot15_O23dF_71_1_0096	UNPlasmidBox06
UWN_0707	I432_O23d	C2_ZC17_1_D3_1wa3-33_rot15_O23dF_81_1_0036	UNPlasmidBox06
UWN_0708	I432_O23d	C2_ZC17_1_D3_1wa3-33_rot15_O23dR_90_1_0094	UNPlasmidBox06
UWN_0709	I432_O23d	C2_ZC17_1_D3_1wa3-44_rot15_O23dF_24_1_0072	UNPlasmidBox06
UWN_0710	I432_O23d	C2_ZC17_1_D3_1wa3-44_rot15_O23dF_64_1_0007	UNPlasmidBox06

VITA

Una Nattermann grew up in Bloomington, IN and completed her undergraduate studies at Vanderbilt University in Nashville, TN in 2013. There, she received two degrees in French and Honors in Molecular and Cellular Biology. Una did undergraduate research in the laboratory of Gerald Stubbs studying the structures of amyloid protein fibrils using X-ray fiber diffraction and negative-stain electron microscopy, and received the Departmental Award for Outstanding Research in Biological Sciences. After joining the Biological Physics, Structure, and Design graduate program at the University of Washington, Seattle, she became fascinated by the computational design of protein biomaterials and their prospects for environmental and health applications. Working in the laboratory of David Baker, Professor of Biochemistry and Directory of the Institute for Protein Design, Una is now an expert in computational protein engineering and design with Rosetta macromolecular modeling suite, symmetric protein assembly theory, and diverse experimental and structural characterization methods. Outside of the lab, Una enjoys rock climbing and mountaineering, as well as continuing to manipulate protein materials while baking sourdough bread (Appendix Fig. 3.1).