

©Copyright 2021

Christine Saeyoung Kang

**Hierarchical design of metal-mediated peptide materials**

**Christine Saeyoung Kang**

A dissertation

submitted in partial fulfillment of the  
requirements for the degree of

Doctor of Philosophy

University of Washington

2021

Reading Committee:

David Baker, Chair

Julie Kovacs

Cole DeForest

Stéphanie Valteau

Program Authorized to Offer Degree:

Department of Chemical Engineering

University of Washington

**Abstract**

Hierarchical design of metal-mediated peptide materials

Christine Saeyoung Kang

Chair of Supervisory Committee:  
David Baker  
Department of Biochemistry

Metal–organic frameworks (MOFs) are a subset of coordination polymers often made up of an organic linker, usually a rigid small molecule, and either a metal ion or metal cluster. Despite key advances in the development and functionalization of MOFs, their prediction and rational design has remained a challenge. New methodologies have allowed for limited computational prediction of MOFs made from highly ordered metal clusters and short, very rigid linkers, however, the computational design of MOFs with more flexible linkers like peptides has not yet been realized. Furthermore, few MOFs discovered that use these more flexible but highly functional peptide components exist, making the *de novo* design of such structures a new and exciting endeavor.

To help address this challenge, I worked towards designing MOF-inspired materials from *de novo* cyclic peptides. In the second chapter of this thesis, I will describe the design of

symmetric cyclic peptides for use as structured modular precursors. Here, I designed and characterized 94 peptides in a variety of  $C_n$  symmetries, ranging from 6-24 residues in length.

In the third chapter of this thesis, I report our efforts towards developing two different methods for the design and characterization of larger crystalline structures from these ordered cyclic peptides. The first was a bottom-up approach, developing computational methods for the design of new peptides that when coordinated to metal in a defined fashion can be geometrically compatible with four cubic space groups. The second was a top-down experimental approach, developing new ways to screen and test crystal growth assembly conditions in a high-throughput manner. Here, we selected 42 peptides for crystallization and solved the structures of three crystals that incorporated metal ions into their structure. Although these three observed crystals deviate from our computational design models, they improved our understanding of different features of the crystallization processes, including local metal coordination geometries, side chain configurations, and crystal packing density. All of these lessons will be incorporated into our computational methodology for improved crystal prediction and design. Overall, although the development of computational methods to design MOF-inspired peptide materials is still in its infancy, the methods outlined here provide a basic framework upon which future efforts can build and expand.

**Dedication**

To my 엄마 and 아빠.

### **Acknowledgments**

Thank you to Drs. Parisa Hosseinzadeh and Vikram Mulligan for teaching me everything I know about designing cyclic peptides and the Rosetta Software Suite, and how to be a better mentor and scientist. Thank you to Meerit Said, and Drs. Shunzhi Wang and Will Sheffler for your many insights, discussions, and fruitful collaboration and teamwork throughout my final few years of graduate school. Thank you also to Dr. Hua Bai for coding help, especially on the “drape\_and\_score” script. Thank you to the Peptiders subgroup for making our lab feel a little bit smaller. Thank you Drs. Tim Craven, Patrick Salveson, Garauv Bhardwaj, Kris Deibler, Jacob O’Connor, and Adam Moyer for helping me practice my talks and hone my slides throughout graduate school. Thank you to the IPD Core, especially Xinting Li, RYanne Ballard, Alex Kang, Hannah Nguyen, and Drs. Paul Levine, Asim Bera, and Lance Stewart, for help with peptide synthesis and crystallization. I am also grateful for the opportunities to collaborate with the labs of Drs. Todd Yeates, Brandi Cossairt, and Tamir Gonen during my time in graduate school.

Thank you to the SCI-STEM, JUPITER, and #equity organizing teams for giving me opportunities to learn about and work at the interface of scientific research and STEM equity. Thank you to UAW 4121 and its members for continuing to fight for the needs of academic student employees and postdoctoral fellows at the UW.

Thank you to my committee, including Drs. Julie Kovacs, Cole DeForest, Stéphanie Valteau, and Shaoyi Jiang for providing valuable scientific insights and advice on my research work. Thank you to Dr. David Baker for always providing support, guidance, and direction for the projects I worked on in graduate school, and for many fruitful discussions about next steps.

Thank you to Drs. Ron Zuckermann, Juliet A. Greenwood, and Wassana Yantasee for providing opportunities for me to experience academic research during my time as an undergraduate, and especially to Drs. Ellen Robertson and Joshua Cardiel for helping me realize that I wanted to go to graduate school and for encouraging me to apply to the UW.

Thank you to my family – my mother, father, and brothers – for tirelessly supporting and encouraging me to pursue everything that I want to throughout my life. Thank you especially to DMs Jacob O'Connor, Tim Huddy, Kandise VanWormer, Harley Pyles, and Chad Miller, whose DnD characters and NPCs inspired many of my peptides' nicknames over the years. Finally, thank you to my partner, Kamin Beyer, and friends who've supported and commiserated with me, and always had something to say to make me laugh on my most terrible days when the equipment breaks down and all of my experiments have failed.

For the original manuscript titled "Computational design of mixed chirality peptide macrocycles with internal symmetry" ([doi.org/10.1002/pro.3974](https://doi.org/10.1002/pro.3974)); C.S.K., M.R.S., and V.K.M did data curation. C.S.K., M.R.S., S.R., X.L., I.A., T.Y., and V.K.M did formal analysis. C.S.K, and V.K.M. completed the investigation. C.S.K., M.R.S., S.R., X.L., I.A., A.W., F.D., T.Y., V.K.M, and D.B. developed the methodology. M.R.S. and I.A. collected crystal structure data and solved the structures. T.W.C, A.W., J.L., F.D., and V.K.M contributed to the software. C.S.K., M.R.S, S.R., T.Y., V.K.M., and D.B. wrote, reviewed, and edited the manuscript.

The manuscript was supported by the Howard Hughes Medical Institute (D.B.), Washington Research Foundation (T.W.C.), Open Philanthropy (D.B.), DTRA grant HDTRA1-19-1-0003 (D.B. and C.S.K.), and the Simons Foundation (V.K.M.). Crystallographic diffraction data were collected at the Advanced Photon Source, a U.S. Department of Energy

(DOE) Office of Science User Facility operated for the DOE Office of Science by Argonne National Laboratory under Contract No. DEAC02-06CH11357. We thank Michael Collazo and Duilio Cascio of the UCLADOE X-ray Crystallography and Structure Determination Core (supported by U.S. Department of Energy award DE-FC02-02ER63421) for their assistance with crystallization and data collection. We thank Swetha Sundaram for assistance in crystal mounting. We also thank Andrew Leaver-Fay, Sergey Lyskov, and Rocco Moretti for Rosetta support, and Alexander Ford for fruitful algorithmic discussions. Conformational sampling was performed with the Rosetta `simple_cycpep_prediction` application and peptide design was carried out with the `rosetta_scripts` application. Modifications to both to make this work possible are now included in public releases of the Rosetta software suite, which is available free of charge to academic, not-for-profit, and government users, and can be downloaded from <http://www.rosettacommons.org>.

## Table of Contents

|                                                                                 |    |
|---------------------------------------------------------------------------------|----|
| Chapter 1: Introduction                                                         | 11 |
| Chapter 2: <i>De Novo</i> Design of Structured Symmetric Cyclic Peptides        | 14 |
| 2.1: Development of a Computational Framework for Designing Peptide Macrocycles | 14 |
| 2.1.1: Generating Scaffolds                                                     | 14 |
| 2.1.2: Sequence Assignment - Peptide “Design”                                   | 15 |
| 2.1.3: Computational Structure Validation                                       | 15 |
| 2.2: Experimental Methods and Results                                           | 15 |
| 2.2.1: Symmetric Cyclic Peptides can be Designed with <1Å RMSD Accuracy         | 15 |
| 2.2.2: Additional Peptides                                                      | 16 |
| 2.3: Discussion                                                                 | 18 |
| 2.4: Ongoing Investigations and Future Directions                               | 21 |
| Chapter 3: Design of Crystalline Peptide Materials                              | 25 |
| 3.1 Development of a Computational Framework for Designing Peptide Materials    | 25 |
| 3.2: Development of an Experimental Pipeline for High-Throughput Crystal Growth | 27 |
| 3.2.1: Results and Discussion                                                   | 28 |
| 3.3: Ongoing Investigations and Future Directions                               | 31 |
| Works Cited:                                                                    | 36 |
| Appendix I: Full Description of Computational Protocols                         | 40 |
| A1.1: Basic Peptide Design Protocol                                             | 40 |
| A1.1.1: Enumeration of Symmetric Cyclic Peptide Scaffolds                       | 40 |
| A1.1.2: Computational Design                                                    | 40 |

|                                                                        |    |
|------------------------------------------------------------------------|----|
| A1.1.3: Computational Structure Validation                             | 41 |
| A1.2: Peptide MOF Design Protocol                                      | 42 |
| A1.2.1: Geometric “Matching”                                           | 42 |
| A1.2.2: Computational Peptide Crystal Design                           | 44 |
| A1.2.3: Computational Structure Validation and Design Selection        | 44 |
| A1.3: RosettaScripts XMLs and Supporting Scripts                       | 45 |
| A1.3.1: Peptide Design                                                 | 45 |
| A1.3.2: Crystal Design                                                 | 46 |
| A1.3.3: Computational Structure Validation - Peptide “Forward Folding” | 49 |
| A1.4: PyRosetta Scripts                                                | 49 |
| A1.4.1: Geometric “Matching”                                           | 49 |
| A1.4.2: Computational Structure Validation - Density Calculations      | 50 |
| A1.4.3: Computational Structure Validation - Crystal “Forward Folding” | 53 |
| Appendix II: Peptide Synthesis and Purification                        | 61 |
| A2.1: Peptide Synthesis                                                | 61 |
| A2.1.1: Synthesis of Peptides < 12 Residues in Length                  | 61 |
| A2.1.2: Synthesis of Peptides > 12 Residues in Length                  | 61 |
| A2.2: Peptide Purification                                             | 62 |
| A2.3: Peptide Characterization                                         | 62 |
| A2.3.1: Symmetry Screening <i>via</i> NMR                              | 62 |
| A2.3.2: Structural Characterization <i>via</i> Racemic Crystallography | 62 |
| Appendix III: Materials Crystallization                                | 65 |

|                                                                   |    |
|-------------------------------------------------------------------|----|
| A3.1: Custom HT Crystal Screens                                   | 65 |
| A3.2: HT “Organic Solvent” Crystal Screen Chart                   | 65 |
| A3.3: HT “Aqueous” Crystal Screen Chart                           | 66 |
| A3.4: Crystal Optimization Procedure                              | 66 |
| A3.5: Materials Characterization <i>via</i> X-ray Crystallography | 66 |

## Chapter 1: Introduction

Metal–organic frameworks (MOFs) are a subset of coordination polymers often made up of an organic linker, usually a rigid small molecule, and either a metal ion or metal cluster.<sup>1,2</sup> Interest in MOF materials and other coordination polymers has increased dramatically since the 1990s.<sup>3-5</sup> This interest appears to stem from two key features of these materials: 1) their tunability, as possible combinations of organic linkers and metal ions or metal clusters are seemingly limitless, and 2) their porosity, which makes this class of materials applicable to many strategic areas of research including catalysis, separation, and delivery.<sup>6-8</sup>

Although many early MOFs were made using carboxylate-containing small molecule linkers like 1,4-benzenedicarboxylate (BDC), relatively few MOFs assembled out of non-small molecule precursors like nucleobases, proteins, and short peptides have been discovered so far.<sup>5,9-13</sup> Moreover, many MOFs are discovered – not designed. Indeed, most peptide-based MOFs in the literature today are made up of only short di- or tri- peptide sequences and are discovered after screening large numbers of assembly conditions modulating the ratios of solvents used, metal salt types, pH, presence of counterions, temperature, and time, to list a few. Two interesting examples include an alanine-glycine dipeptide which forms an adaptable material when reacted with zinc(II) nitrate, and a divaline-glutamate tripeptide that forms a porous framework when reacted with copper(II) or calcium nitrate.<sup>9,10</sup> Still, although these short peptide-based MOFs have been found to exhibit interesting emergent properties like guest uptake, their extremely short sequences limit their potential for structural tunability.<sup>14</sup>

Another approach has been to use structured proteins as the “organic” linker portions of MOFs. By modifying native protein crystals to incorporate metal binding amino acids at key

interfaces, modular protein-based materials can be built by using a protein's crystallographic contacts as a template.<sup>12,15</sup> However, this approach still requires high quality, symmetric protein crystal structures as an input, and in this case, also required the use of a small molecule linker to bridge the interactions between proteins.

To date, the longest known peptide organic linker in a MOF is a linear hexaproline peptide.<sup>16</sup> Part of the reason for this may be because short linear peptides are generally easier to synthesize in higher yields than their longer and/or cyclic counterparts. However, linear peptides are generally less conformationally constrained compared to their cyclic counterparts and are often more likely to adopt many low energy conformations, making the rational design of crystalline 3-dimensional materials from linear peptides considerably more challenging.<sup>17</sup>

In order to better support the development of these new materials, additional methods to design and predict the structures and functional properties of MOFs must be developed and built upon. Currently, most computational efforts in this area model the performance of MOFs towards applications like catalysis or guest uptake in the pores of existing materials or otherwise work to understand the impact of a MOF's structure on its material properties.<sup>18,19</sup> In contrast, efforts to predict the structures of MOFs from the geometry of their precursor components are still in development, due at least in part to our limited understanding of the mechanism and factors affecting the crystallization of MOFs.<sup>20-22</sup> Of these, a few popular ways to predict MOF structure from precursor ligands and SBUs *ab initio* include the Automated Assembly of Secondary Building Units (AASBU), direct enumeration method, and reverse topological approach. In the first, the AASBU method uses predefined secondary building units (SBUs) with interactions potentials between them to auto-assemble 3D frameworks through a combination of simulated

annealing and energy minimization procedures. However, while this method has been successful at directly simulating the assembly of various MOFs, it is known to be rather computationally expensive due to the repeated simulated annealing and energy minimizations required by the method. In comparison, the direct enumeration method decorates mathematical nets with a variety of SBUs, which then must also be “relaxed” through quantum chemistry calculations to evaluate their relative energy. In the reverse topological approach, an SBU and a large number of organic linkers is selected then systematically combined into four different proposed MOF “topologies”. Still, in each of these three cases, the prediction of crystalline structures hinge on the idea that the material is made up of short linkers and metal cluster SBUs, and does not appear to have been applied towards the prediction or design of coordination polymers made from peptides and metal ions yet.

In recent years, the Rosetta macromolecular modeling suite has been expanded to include cyclic peptide design.<sup>23-25</sup> Previous lab members have also developed computational frameworks towards modeling and designing a variety of higher-ordered symmetric structures from *de novo* protein building blocks. These include cage-like protein polyhedra, 2-dimensional protein arrays, and other 3-dimensional crystalline materials.<sup>26-29</sup>

Inspired by these newer advances in the the computational design of cyclic peptides and protein materials, updates to the Rosetta score function, and improvements to Rosetta’s ability to model and design symmetric molecules, I developed a framework to design, model, and test the assembly of MOF-inspired cyclic peptide-based coordination polymers from *de novo* symmetric cyclic peptides.<sup>24,29-32</sup>

## Chapter 2: *De Novo* Design of Structured Symmetric Cyclic Peptides

*This chapter contains additional background for (and limited reproduction of) content previously published as:* Mulligan, V. K., Kang, C. S., Sawaya, M. R., Rettie, S., Li, X., Antselovich, I., Craven, T. W., Watkins, A. M., Labonte, J. W., DiMaio, F., Yeates, T. O., & Baker, D. (2020) Computational design of mixed chirality peptide macrocycles with internal symmetry. *Protein Science*. 2020; 29: 2433– 2445. <https://doi.org/10.1002/pro.3974>.

### Development of a Computational Framework for Designing Peptide Macrocycles:

We chose to design cyclic peptides between 6 and 24 residues in length, for point group symmetries  $C_2$ ,  $C_3$ ,  $C_4$ ,  $C_5$ , and  $S_2$ . We also designed and synthesized one  $S_4$  symmetric peptide. Our general strategy for designing these peptide macrocycles was to: 1) enumerate a large set of symmetric poly-glycine peptide backbones of various lengths and symmetries, 2) design these scaffolds using a variety of L- and D- amino acids and using conformationally constrained amino acids to ensure that the peptide backbone maintains its designed structure, 3) select peptide “designs” for synthesis based on their total Rosetta score and  $\Delta G_{\text{folding}}$ , and 4) experimentally validate the structures of the designs using 1D  $^1\text{H}$  NMR and x-ray crystallography.

Summary-level descriptions of these methods can be found below, and more specific information on the computational methods can be found in Appendix I.

### Generating Scaffolds:

To design and characterize a diverse set of structured symmetric cyclic peptide macrocycles representing a large number of different symmetric point groups, I first enumerated a large scaffold (poly-glycine backbone) set of symmetric cyclic peptides using methods previously developed in our lab.<sup>23-25</sup>

### Sequence Assignment - “Design”:

I wrote several protocols to assign sequences to each scaffold using RosettaScripts, an XML-based scripting interface for the Rosetta macromolecular modeling suite.<sup>33</sup>

### Computational Structure Validation:

Designed peptides were selected for synthesis based on their total Rosetta score and  $\Delta G_{\text{folding}}$  (in kcal/mol) as evaluated *in silico* using the SimpleCycpepPredict application.<sup>31,34-37</sup>

### Experimental Methods and Results:

1D  $^1\text{H}$  NMR was used to evaluate whether the peptides are structured and symmetric in solution, and crystals were grown using racemic mixtures of the design peptides and their mirror sequences. Racemic mixtures of the designed peptides were crystallized in commercially available protein crystallization screens and grown using either sitting drop or hanging drop diffusion. Crystals were looped and screened in-house, and sent to the Advanced Photon Source (APS) at Argonne National Lab for data collection. More specific information on the experimental methods can be found in Appendix II.

*Symmetric Cyclic Peptides can be Designed with  $<1\text{\AA}$  RMSD Accuracy:*

**Table 2.1:** Peptides published in Mulligan, *et al.*\*\*<sup>24</sup>

| Peptide Name     | Symmetry | Sequence of the Asymmetric Unit* | RMSD to Design (in $\text{\AA}$ ) |
|------------------|----------|----------------------------------|-----------------------------------|
| C2-1 “Zappy”     | $C_2$    | sEXSL                            | 1.032                             |
| C3-1 “Sporty”    | $C_3$    | PRvD                             | 2.7                               |
| C3-2 “Baby”      | $C_3$    | PRD                              | 0.487                             |
| C3-3 “PDD”       | $C_3$    | PDD                              | 0.299                             |
| S2-1 “Wednesday” | $S_2$    | pEvK                             | 0.593                             |

|                      |                |        |       |
|----------------------|----------------|--------|-------|
| S2-2 “Morticia”      | S <sub>2</sub> | PkVe   | 0.390 |
| S2-3 “Lurch”         | S <sub>2</sub> | qTRP   | 0.609 |
| S2-4 “Pugsley”       | S <sub>2</sub> | EppKv  | 0.531 |
| S2-5 “Uncle Fester”  | S <sub>2</sub> | pSEiR  | 0.495 |
| S2-6 “London Bridge” | S <sub>2</sub> | aNkhPe | 0.433 |
| S4-1 “Tim”           | S <sub>4</sub> | KLqeXH | 0.308 |

\*X = AIB, 2-aminoisobutyric acid

\*\*All X-ray crystal structures above were deposited in the Protein Data Bank (peptide C2-1: PDB IDs 6UFU and 6UG2; peptide C3-1: PDB IDs 6UG3 and 6UG6; peptide C3-2: PDB ID 6UGB; peptide C3-3: PDB ID 6UGC; peptide S2-1: PDB ID 6UCX; peptide S2-2: PDB ID 6UD9; peptide S2-3: PDB IDs 6UDR and 6UDW; peptide S2-4: PDB IDs 6UDZ and 6UF4; peptide S2-5: PDB ID 6UF7; peptide S2-6: PDB ID 6UF8; peptide S4-2(holo): PDB ID 6UFA; peptide S4-1(apo): PDB ID 6UF9).

#### *Additional Peptides:*

Table 2.2 below contains information about additional cyclic peptides I designed and characterized that were not included in the above paper.

**Table 2.2:** Synthesized peptides that were not published, but were symmetric *via* NMR

| Peptide Name  | Symmetry       | Sequence of the Asymmetric Unit* | Status                                          |
|---------------|----------------|----------------------------------|-------------------------------------------------|
| C2-2 “Bert”   | C <sub>2</sub> | ypDQS                            | Symmetric <i>via</i> NMR; unable to crystallize |
| C2-3 “Ernie”  | C <sub>2</sub> | DXRLq                            | Symmetric <i>via</i> NMR; unable to crystallize |
| C2-4 “Batman” | C <sub>2</sub> | PTeeyQ                           | Symmetric <i>via</i> NMR; unable to crystallize |
| C3-1a “Sage”  | C <sub>3</sub> | PRd <i>N-methyl</i> -E           | Crystallized; 0.72 Å all-atom RMSD to design    |
| C3-1b “Shiso” | C <sub>3</sub> | <i>N-methyl</i> -EVRd            | Crystal didn’t diffract well enough to solve    |

|                  |                |        |                                                 |
|------------------|----------------|--------|-------------------------------------------------|
| C3-4 “Londrilun” | C <sub>3</sub> | DLk    | Crystallized; 2.2 Å RMSD to design              |
| C3-5 “Utre”      | C <sub>3</sub> | DpYk   | Symmetric <i>via</i> NMR; unable to crystallize |
| C3-6 “Veche”     | C <sub>3</sub> | DpKy   | Symmetric <i>via</i> NMR; unable to crystallize |
| C3-7 “Pouluno”   | C <sub>3</sub> | Dvpk   | Symmetric <i>via</i> NMR; unable to crystallize |
| C3-8 “Merric”    | C <sub>3</sub> | EPK    | Symmetric <i>via</i> NMR; unable to crystallize |
| C3-9 “Dusk”      | C <sub>3</sub> | DPrY   | Symmetric <i>via</i> NMR; unable to crystallize |
| C3-10 “Ice”      | C <sub>3</sub> | EkAP   | Symmetric <i>via</i> NMR; unable to crystallize |
| C3-11 “Paul”     | C <sub>3</sub> | KdpNL  | Symmetric <i>via</i> NMR; unable to crystallize |
| C3-12 “Berb”     | C <sub>3</sub> | AvdQK  | Symmetric <i>via</i> NMR; unable to crystallize |
| C3-13 “Rane”     | C <sub>3</sub> | dlqkP  | Symmetric <i>via</i> NMR; unable to crystallize |
| C3-14 “Derrick”  | C <sub>3</sub> | Dlk    | Symmetric <i>via</i> NMR; unable to crystallize |
| C3-15 “Chat”     | C <sub>3</sub> | pvrD   | Symmetric <i>via</i> NMR; unable to crystallize |
| C3-16 “Morty”    | C <sub>3</sub> | RePI   | Symmetric <i>via</i> NMR; unable to crystallize |
| C3-17 “H010”     | C <sub>3</sub> | KPVD   | Symmetric <i>via</i> NMR; unable to crystallize |
| C3-18 “Dewey”    | C <sub>3</sub> | DKTL   | Symmetric <i>via</i> NMR; unable to crystallize |
| C3-19 “Veni”     | C <sub>3</sub> | dPyR   | Symmetric <i>via</i> NMR; unable to crystallize |
| C3-20 “Darla”    | C <sub>3</sub> | DKvY   | Symmetric <i>via</i> NMR; unable to crystallize |
| C3-21 “Jean”     | C <sub>3</sub> | DIRs   | Symmetric <i>via</i> NMR; unable to crystallize |
| C3-22 “Black”    | C <sub>3</sub> | PeR    | Symmetric <i>via</i> NMR; unable to crystallize |
| C3-23 “Leto”     | C <sub>3</sub> | PQkDv  | Symmetric <i>via</i> NMR; unable to crystallize |
| C3-24 “Melange”  | C <sub>3</sub> | lpEASK | Symmetric <i>via</i> NMR; unable to crystallize |
| C3-25 “Caterpie” | C <sub>3</sub> | iKep   | Symmetric <i>via</i> NMR; unable to crystallize |
| C3-26 “Ekans”    | C <sub>3</sub> | pLeka  | Symmetric <i>via</i> NMR; unable to crystallize |
| C3-27 “Nidoran”  | C <sub>3</sub> | PTKDv  | Symmetric <i>via</i> NMR; unable to crystallize |
| C4-1 “Tenzin”    | C <sub>4</sub> | dXyK   | Symmetric <i>via</i> NMR; unable to crystallize |
| C4-2 “Pasang”    | C <sub>4</sub> | etPK   | Symmetric <i>via</i> NMR; unable to crystallize |
| C5-1 “Vega”      | C <sub>5</sub> | PKD    | Symmetric <i>via</i> NMR; unable to crystallize |

|                   |                |       |                                                 |
|-------------------|----------------|-------|-------------------------------------------------|
| C5-2 “Cassiopeia” | C <sub>5</sub> | pRd   | Symmetric <i>via</i> NMR; unable to crystallize |
| C5-3 “Capella”    | C <sub>5</sub> | HKD   | Symmetric <i>via</i> NMR; unable to crystallize |
| C5-4 “Scorpius”   | C <sub>5</sub> | dIR   | Symmetric <i>via</i> NMR; unable to crystallize |
| C5-5 “Lyra”       | C <sub>5</sub> | vRd   | Symmetric <i>via</i> NMR; unable to crystallize |
| C5-6 “Deneb”      | C <sub>5</sub> | pKeT  | Symmetric <i>via</i> NMR; unable to crystallize |
| C5-7 “Golden”     | C <sub>5</sub> | FTApA | Symmetric <i>via</i> NMR; unable to crystallize |

\*X = AIB, 2-aminoisobutyric acid

### Discussion:

In addition to these peptides listed above, 38 peptides were able to be synthesized but did not appear to be symmetric in solution *via* NMR and 15 peptides were also attempted to be synthesized but failed. In both cases, generally, as peptides become longer, their linear form can become more flexible and can often adopt many more conformations. To compensate for this, I increased the number of constrained amino acids like AIB and Prolines in each longer design. However, it is also well known that constrained amino acids tend to be harder to cyclize on-to due to a variety of reasons, primarily steric hindrance.<sup>38</sup> As a result, some designs could not be cyclized in high enough yields to test (or at all). Many longer sequences, especially those >15 residues in length were found to not be structured or symmetric *via* 1D <sup>1</sup>H NMR. These sequences can be found in **Appendix Table 2.1**.

Still, it is possible that some of these symmetric peptides may adopt many conformations in solution, but will still crystallize into a singular conformation. One notable example of this is S4-1, which in the presence of Zn(II) crystallizes into a conformation with an RMSD of 0.308Å to the design, but did not appear to be structured *via* NMR even in the presence of Zn(II). This 24-residue peptide is clearly somewhat conformationally flexible as in the absence of Zn(II), it

crystallizes in an “inside-out” conformation, S4-2.<sup>24</sup> However, the crystal structure of this peptide does confirm that the addition of structural motifs like metal ions can be used similarly to existing chemical crosslinkers to stabilize even longer, more flexible cyclic peptides. These motifs allow the peptides to be structured enough to be used in the design of peptide-based materials.<sup>34</sup>

Previous work in the lab designing structured *de novo* peptide macrocycles have found that torsionally-constrained amino acids like L- and D- prolines play key roles in structure designation, as do backbone-backbone and side chain-backbone hydrogen bonds.<sup>23</sup> Other similarly constrained non-canonical amino acid residues like  $\alpha$ -aminoisobutyric acid (AIB) can also help in the design of cyclic peptides with a strong preference for the designed conformation. Hydrogen bonding also plays an important role in the design of structured macrocycles. Unsatisfied buried hydrogen bond donors (unbonded backbone amide nitrogens) can change the predicted conformation of a peptide macrocycle in order to hydrogen bond with an acceptor (carbonyl oxygen). This is illustrated in C3-1, which crystallized in a different conformation than that of the design (**Figure 2.1**). In this case, buried unsatisfied hydrogen bond donor atoms on the aspartate residues of the peptide design became satisfied by a neighboring peptide in the context of the crystal lattice by adopting this new conformation. Rosetta’s scorefunction still seems to have a slight preference for the designed conformation over that of the crystal structure. So, I investigated which sequences could be used to help satisfy the currently unsatisfied amide nitrogen. I redesigned the crystal structure scaffold to look for new sequences that both select for the new design when computationally modeled and recapitulate the design state when crystallized.

I selected four new variants of C3-1 for synthesis and methylated the unsatisfied amide nitrogen in each of these designs to ensure that its structure matches that of the crystal structure of C3-1. Of these, only C3-1a “Sage” and C3-1b “Shiso” were able to be synthesized. Both C3-1a and C3-1b appeared to be C3 symmetric *via* 1D  $^1\text{H}$  NMR, and both were able to be crystallized, but only C3-1a diffracted at a high enough resolution to be solved. As expected, C3-1a matched the crystal structure of C3-1 with an RMSD of 0.72 Å (**Figure 2.2**).

C3-2 and C3-3 matched that of their designs very closely (**Figure 2.3A**). The first of these, C3-2, contained two distinct peptides in the asymmetric unit of the crystal structure. Interestingly, three  $\text{Na}^+$  ions were found to coordinate along the axis of symmetry for the peptides, with two ions binding to the peptide’s backbone carbonyl oxygens, and one ion appearing to mediate the interaction between the two peptides in the asymmetric unit. This crystal structure also revealed that the arginine side chains were taking on different conformations even within the same peptide. To investigate this further, I mutated the arginines to every other L- amino acid or glycine to determine whether the arginines in this peptide design were necessary to its structure. The SimpleCycpepPredict application was used to evaluate the effects of each of these mutations *in silico*. With the exception of L- proline, I found that mutating the L- arginines in this peptide to any other L- amino acid had no effect on the predicted low-energy backbone conformation of the peptide, meaning that I could conceivably use any non-canonical L-  $\alpha$ -amino acid in this position without changing the structure of the peptide itself. L- proline and glycine were exceptions to this case for very different reasons. L- proline’s Ramachandran ( $\psi$  and  $\phi$ ) preferences are much more restrictive than other L- amino acid side chains due to its aliphatic ring structure. In contrast, glycine is achiral and its van der

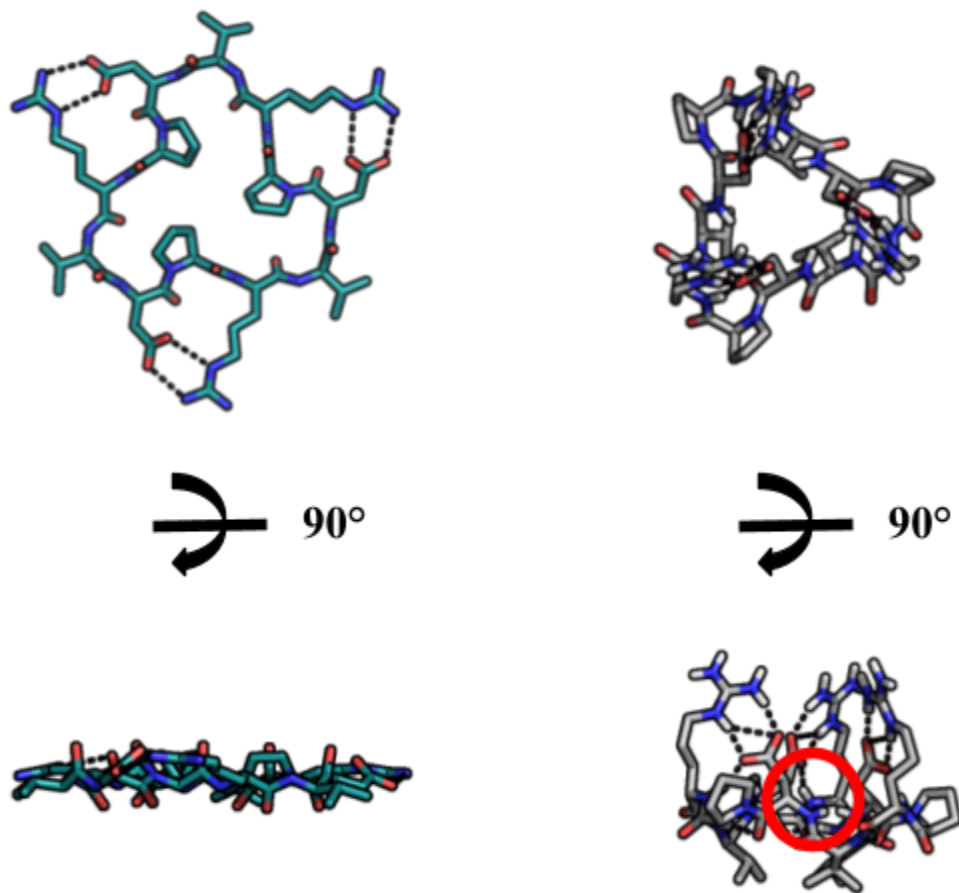
Waals radius is much smaller, so its Ramachandran preferences are much more flexible than that of other amino acids. Overall, these results demonstrate that although these arginines can be mutated to many other amino acid side chains, the chirality and flexibility of the substituted group matters.

To demonstrate this, I synthesized a variant of the original peptide design where I mutated all of the L- arginines to L- aspartates. **Figure 2.3B** shows clearly that, indeed, this residue position can be mutated fairly freely within the constraints listed above. In fact, this peptide, called C3-3, appears to match that of the design even more closely with an RMSD of 0.299Å!

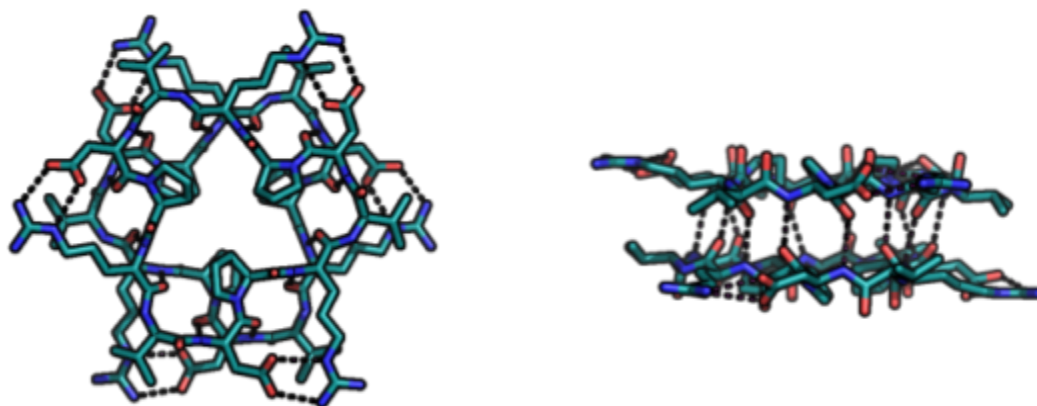
#### Ongoing Investigations and Future Directions:

Now that I have shown that we can computationally design and characterize symmetric cyclic peptides, I am working with Dr. Brandi Cossairt's lab to investigate the impact of these peptides on templating InP magic-sized clusters (MSCs). Since the crystal structures for C3-2 and C3-3 have demonstrated that R->D mutations do not impact the overall structure of the peptide, and in particular do not impact the conformation of the conserved L- aspartic acid side chains, we have begun testing whether the presence of more hydrophobic variants of this peptide during InP particle synthesis have an impact on the size of the resulting MSCs. Our hypothesis is that since carboxylic acid-containing ligands like oleic acid (as well as phosphonate- and amine-containing ligands) are commonly used to passivate InP MSCs during their synthesis, structured carboxylic acid-containing peptides like C3-3 and its more hydrophobic variants may also be useful in modulating the overall size and structures of InP MSCs as well.<sup>40</sup>

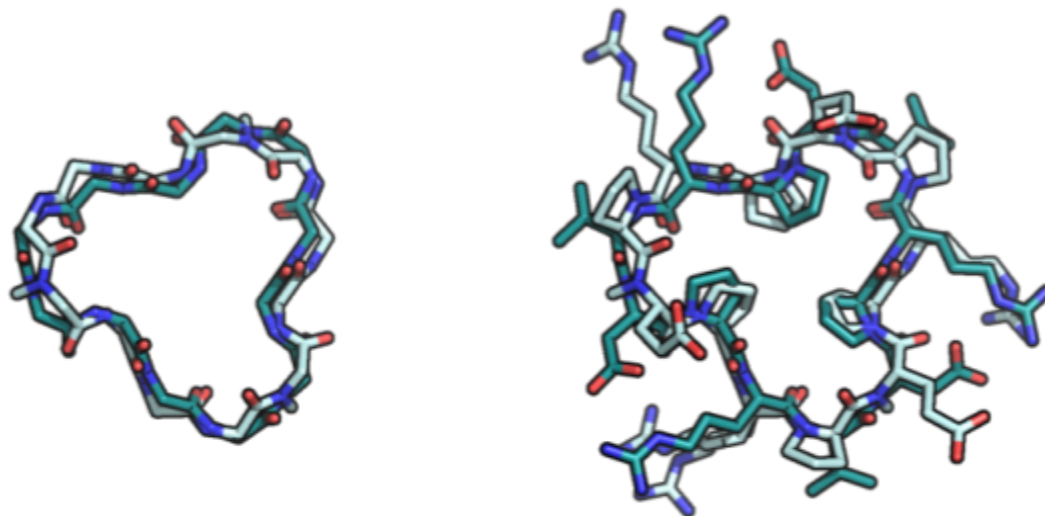
A)



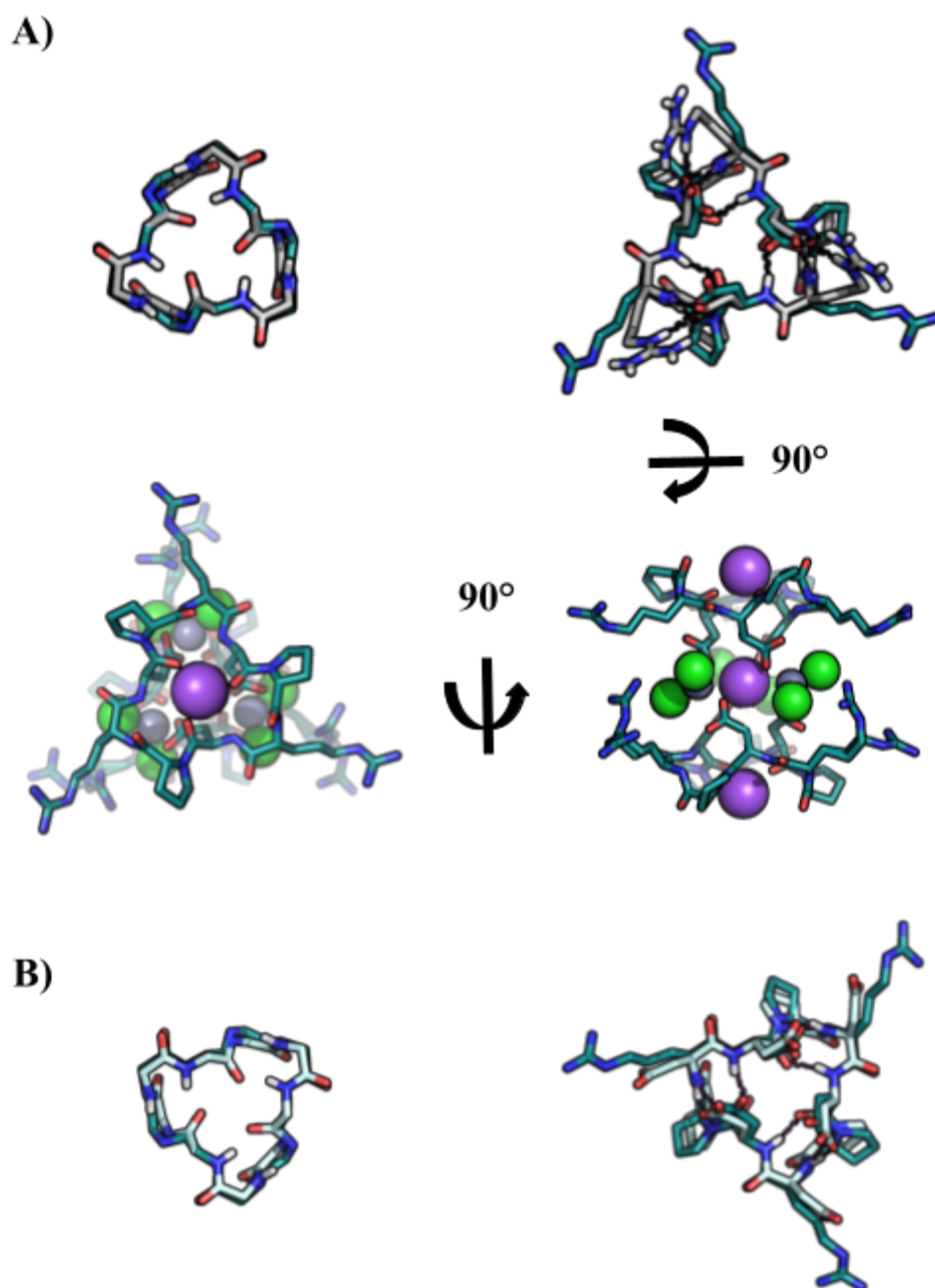
B)



**Figure 2.1:** A) The crystal structure (left) of C3-1 shows that it takes on a significantly different conformation than that of the design (right) with a backbone RMSD of 2.7 Å. This peptide diffracted to 1.4 Å resolution. B) This new conformation appears to be a result of a buried unsatisfied hydrogen bond donor atom on the aspartate residues, which become satisfied (left) by a neighboring peptide in the context of the crystal lattice causing the peptides to “stack” (right).



**Figure 2.2:** The crystal structure of C3-1a (light green) matched the crystal structure of C3-1 (teal) with an RMSD of 0.72 Å. Comparing the backbones of the two designs, it is clear that there is good agreement between both.



**Figure 2.3:** The crystal structures of C3-2 (A) and C3-3 (B) matched that of their designs very closely. The design was rendered in light gray and the crystal structures were rendered in teal. Hydrogen bonds are colored in purple. A) Comparing the backbones of the design and crystal structures for C3-2, it is clear that there is good agreement between both. There are two peptides in the asymmetric unit of the crystal. Three Na<sup>+</sup> ions were found to coordinate to the backbone of this peptide, two of which were coordinated to carbonyl oxygens along the axis of symmetry for this peptide. The backbone RMSD to design for this crystal structure was determined to be 0.487 Å, and the all-atom RMSD to design for this crystal structure was determined to be 0.829 Å. C3-2 diffracted to 1.0 Å resolution. B) The crystal structure for C3-3 also matched its design with very good agreement. The backbone RMSD to design for C3-3 was 0.299 Å and the all-atom RMSD to design was 0.674 Å.

### **Chapter 3: Computational Design and Exploration of Crystalline Peptide Materials**

In order to increase our understanding of key factors involved in the design of MOF-inspired, peptide-based materials, we developed two different pipelines for computationally designing and experimentally discovering new structures. The first is a bottom-up computational approach inspired by existing reverse topological methods used in the prediction of MOF crystal structures. This involves the development of methods to create design models for peptide crystals in various space groups, assign amino acid sequences to these models to enforce their rigidity, and select the modeled structures for synthesis by evaluating their relative energies compared to other “off-target” forms. The second is a top-down experimental exploration of different assembly conditions and their impacts on the growth of metal-mediated crystalline materials. Our efforts towards both of these frameworks are outlined below.

#### Development of a Computational Framework for Designing Peptide-based Materials:

Over the past several years, Frank DiMaio, PhD and Una Nattermann, PhD developed an algorithm to identify all of the possible combinations of different symmetric point groups that can be used to define a crystallographic space group. Using this information, I used PyRosetta, a Python interface for the Rosetta software suite, to write a pilot protocol that could determine the compatibility of a symmetric scaffold for any given crystal lattice space group based on geometry. From there, Will Sheffler, PhD wrote a much more comprehensive script that would not only determine the compatibility of a symmetric scaffold for a small subset of predefined crystal lattice space groups based on geometry, but also calculate crystal unit cell dimensions ( $a$ ,  $b$ ,  $c$ , and  $\alpha$ ,  $\beta$ ,  $\gamma$ ). This allows the rest of the peptide sequence to be redesigned in the context of the crystal at a later stage using symmetry machinery built by DiMaio. To start, this script was

only able to identify geometric “matches” for four cubic lattices  $I2_13$ ,  $P4_132$ ,  $P4_332$ , and  $P23$ .<sup>41</sup>

The first three can be visualized as a threefold symmetric peptide with a twofold symmetric metal binding site. More work has been done to expand this method to include additional space groups, including  $P23$ , another Primitive cubic lattice, made from a  $C_3$  symmetric peptide and modeling the metal ion as a  $D_2$  symmetric site. Efforts to include additional space groups are underway, however in this report, I will be focusing only on our efforts towards designing peptide materials targeting these first four space groups -  $I2_13$ ,  $P4_132$ ,  $P4_332$ , and  $P23$ .

Many metal ions are well known to be rather promiscuous with their binding geometry, despite having some geometric preferences.<sup>42</sup> However, since the Rosetta software suite is not yet able to model metal ions, to start, we chose to explicitly define the geometry of the metal binding site using a system of distance, angle, and dihedral constraint using existing literature values for Zn(II).<sup>42</sup> Since it is also well known that peptide side chains can often be somewhat flexible in nature, in order to help make the interaction through the metal binding sidechain more rigid, our initial strategy focused on finding matches with a built-in bidentate metal binding site consisting of two metal binding residues (*e.g.* aspartic acid, glutamic acid, histidine, and/or cystine) on each asymmetric unit of the cyclic peptide backbone. More information on these protocols can be found in Appendix I.

From there, Rosetta was used to assign (“design”) amino acids to the remaining residue positions on the asymmetric unit of the peptide in the context of the matched crystal lattice.<sup>43,44</sup> For example, a  $C_3$  symmetric 9mer would have one residue per asymmetric unit available for assignment in this way. Larger peptides would have more residue positions available for assignment as well.

Once these peptide crystal designs were generated, we used several different mechanisms to select a small subset of them for synthesis and experimentation. The first was to filter designs based on their atomic density. Although MOFs are generally known for their porosity, we chose to select more densely packed crystal designs because we understand that our cyclic peptide nodes, while relatively structured, are still not as rigid as the kinds of small molecule linkers widely explored and used in MOF synthesis, and therefore may be less likely to form structures with large void space. The second was to computationally evaluate each peptide's likelihood of folding into the design conformation using the `simple_cycpep_predict` application. This was done in order to identify which peptide designs have large  $\Delta G_{\text{folding}}$ . Finally, designs were evaluated on whether their sequence "prefers" that scaffold in that crystal space group over others. Additional information describing each of these computational methods can be found in Appendix I.

#### Development of an Experimental Pipeline for High-Throughput Crystal Growth:

Although many MOFs in the literature are often grown at tens or hundreds of milliliters scale with high concentrations of ligands and metal ions, because our peptides are typically synthesized at milligram scale, we needed to develop new crystallization methods that would allow us to conserve the amount of peptide used in any one experiment and maximize the number of conditions we could test with a single synthesis. To do this, we took inspiration from methods commonly used to grow protein crystals, and designed two custom crystal screens in a 96-well format to quickly identify conditions for crystal growth. The first focused on mostly organic solvents commonly used in MOF growth like N,N-dimethylformamide (DMF), N,N-diethylformamide (DEF), acetonitrile (ACN), and various alcohols. The second screen

focused on screening various buffer conditions at a large range of pHs (from pH 5.5 to 8).

Additional information on the conditions included in these custom screens can be found in Appendix III.

#### *Results and Discussion:*

Of the 42 peptides selected for synthesis, 28 were able to be crystallized, and 3 were able to be solved. None of the solved structures matched the space group they were designed for, indicating that we still have a long way to go to understand the key principles for designing a cyclic peptide based MOF. The vast majority of the crystals that grew had either a needle-like or “urchin”-like morphology, making them difficult to loop and collect XRD data on.

(“Urchin”-like crystals look like balls of many needle crystals growing out from what appears to be a single nucleation site.) In addition, many of the crystals that grew did not incorporate any metal ions in their structure, despite having metal binding residues in the peptide’s sequence and being grown in the presence of metal. In some cases, even crystals that grew into non-needle-like structures could not be solved as they dissolved outside of their mother liquor or diffracted poorly, if at all.

Our first crystal structure was for a crystal design called “1.3\_176” where the design crystallized in an equimolar amount of peptide and  $\text{Zn}(\text{NO}_3)_2$  in MES buffer at pH 6 after being heated to 80°C for 24 hours. Although this peptide’s backbone crystallized in a C3 symmetric conformation, neither the conformation of the peptide nor the space group of the crystal matched the design. Furthermore, although the crystal structure was designed to coordinate to the metal ion with one histidine and one aspartic acid per asymmetric unit — to form a C<sub>2</sub> symmetric metal binding site with two histidines and two aspartic acids coordinating the metal in a tetrahedral

geometry — in the crystal structure, each Zn(II) ion was internally coordinated by three histidines of one peptide and an aspartic acid from another in a tetrahedral fashion (**Figure 3.1**). We believe that this happened for two reasons: 1) the actual crystal structure is packed much more densely and has more favorable peptide-peptide interactions than the original design, and 2) we found that our computational protocol had accidentally placed a d- histidine in a residue position with a negative phi value. Based on the Ramachandran plot, an l- histidine should have been placed there instead. The bug that caused this error has been found and fixed, but this demonstrates again how important the identity and chirality of each amino acid in a sequence is in determining the resulting structure of these peptides.

Because of the difficulty in crystallizing even just this first peptide, in order to increase our chances of successfully crystallizing more metal binding peptides, we also tried to crystallize  $S_2$  symmetric peptides with the high-throughput screens we had built using all of the different metal ions we had available to us. (See Appendix III for screen conditions.) This allows us to more quickly generate additional crystal structure data with which we can iterate on our computational design process. We specifically chose to crystallize  $S_2$  symmetric peptides in the presence of metals specifically because unlike peptides with  $C_n$  symmetry, S-symmetric peptides are centrosymmetric, meaning they contain an inversion center as one of their symmetry elements. In other words, S-symmetric peptides are their own mirrors and are therefore achiral. Racemic crystallography is well known to increase the possibility of a crystallization hit because chiral molecules (like  $C_n$  symmetric peptides) can only crystallize in just 65 of the 230 different possible space groups. In contrast, achiral, centrosymmetric molecules (or mirrored pairs of chiral molecules) can crystallize in up to 165 different packing arrangements due to their added

inversion center.<sup>45</sup> Given that racemic crystallography is a well known strategy for crystallizing short peptides, we believed that S-symmetric peptides might be easier experimentally to crystallize as well.

The second and third crystal structures were grown from  $S_2$  symmetric peptides. The second peptide crystal was grown from S2-6 and crystallized with equimolar  $ZnCl_2$  and peptide in Isopropanol. This peptide crystallized after just 24 hours at room temperature. The structure of this peptide and its crystal matched that of the original, published peptide-only crystal structure with very good agreement. Interestingly, although the sequence of S2-6 contains both histidines and glutamic acids, none of these residues are bound to metal ions in the crystal structure. Instead the structure appears to be mediated mostly by backbone to backbone or sidechain to backbone peptide-peptide interactions. In addition, all of the metal binding atoms of these charged residues appear to be bound to water. As a result, despite the presence of metal binding residues, the  $Zn(II)$  ions simply entered into the water-accessible pores of the crystal structure and did not bind the peptides (**Figure 3.2**). This is also interesting because in the higher pH aqueous conditions we tested, this peptide failed to crystallize, and even the original published crystal structure, S2-6, grew in slightly acidic conditions at pH 5.0. This suggests that at higher pH, the crystal contacts may be broken as the peptides coordinate to the metal.

The third crystal structure was grown from S2-4 and was crystallized with equimolar  $Cu(NO_3)_2$  and peptide in 1:1 water:DMF after being heated to 80°C for 24 hours. This structure is mediated by metal binding interactions between the  $Cu(II)$  ion and lysine and glutamic acid residues from peptides across a layer (so that each metal ion is bound to two lysines and two glutamic acids). Backbone-backbone interactions between adjacent neighboring peptides, and a

string of water molecules down the symmetry axis of the peptides to which backbone carbonyl oxygens, amide nitrogens, and other water molecules hydrogen bond mediate the other axes of the crystal. This effectively forms a network linking one half of each peptide to one half of another resulting in the aforementioned “water molecule axis of symmetry” down the centers of these peptides in the crystal structure (**Figure 3.3**). More information about each crystal that was able to be grown and solved can be found in **Appendix Table 3.3**.

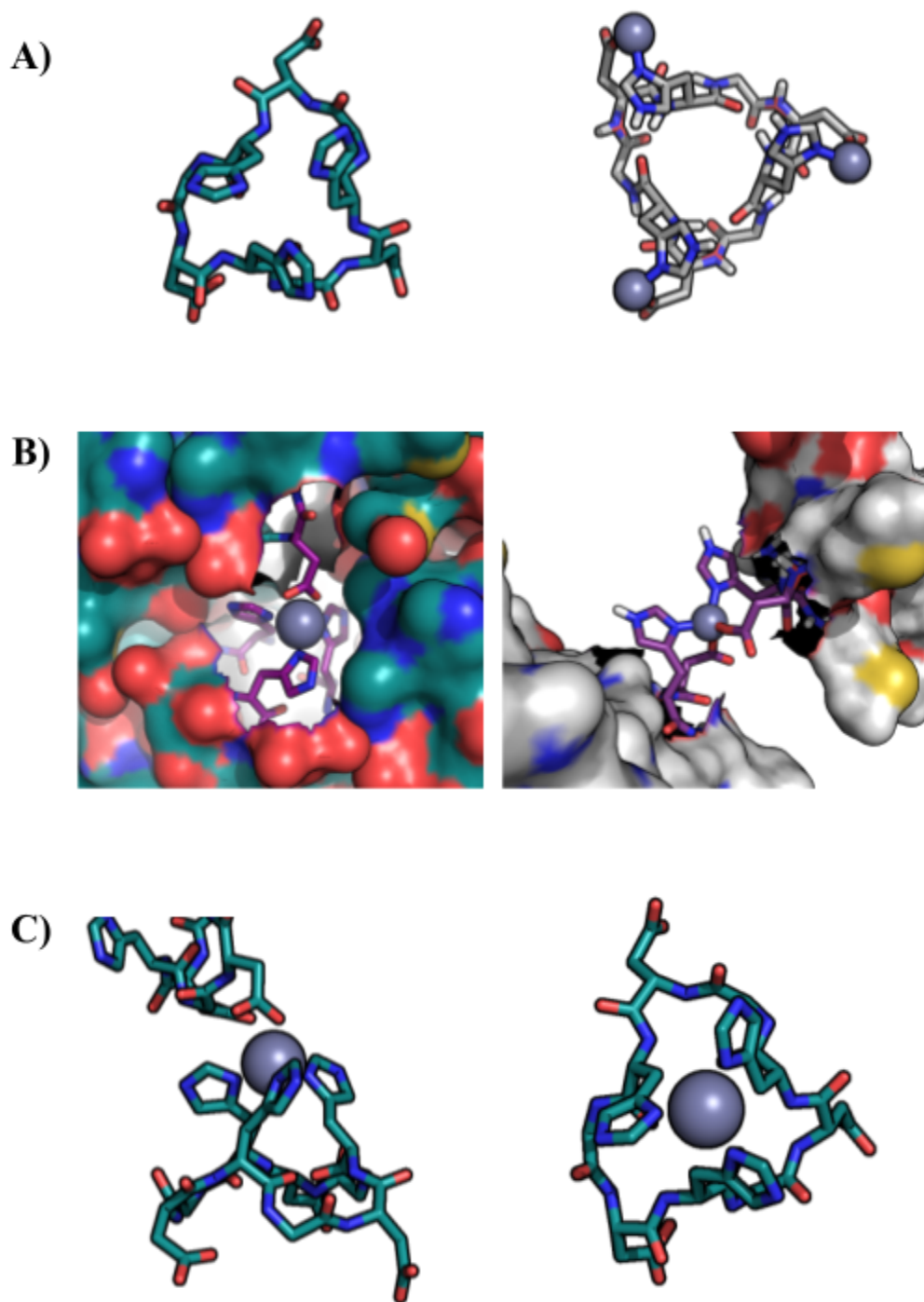
In the end, although we have yet to design a peptide that crystallizes into the designed crystal space group, we were able to grow a peptide MOF with what is now known to be the longest peptide sequence — 10 residues! The previous record, published earlier this year, utilized a linear hexaproline peptide.<sup>16</sup> Furthermore, through this process, we developed a robust experimental pipeline for quickly screening growth conditions and characterizing peptide crystals. Finally, we developed an overarching framework for the computational design of peptide-based crystals, which can be expanded upon in the future to increase the number of space groups peptides can be designed to crystallize into and improved to better model metal ions in future designs.

#### Ongoing Investigations and Future Directions:

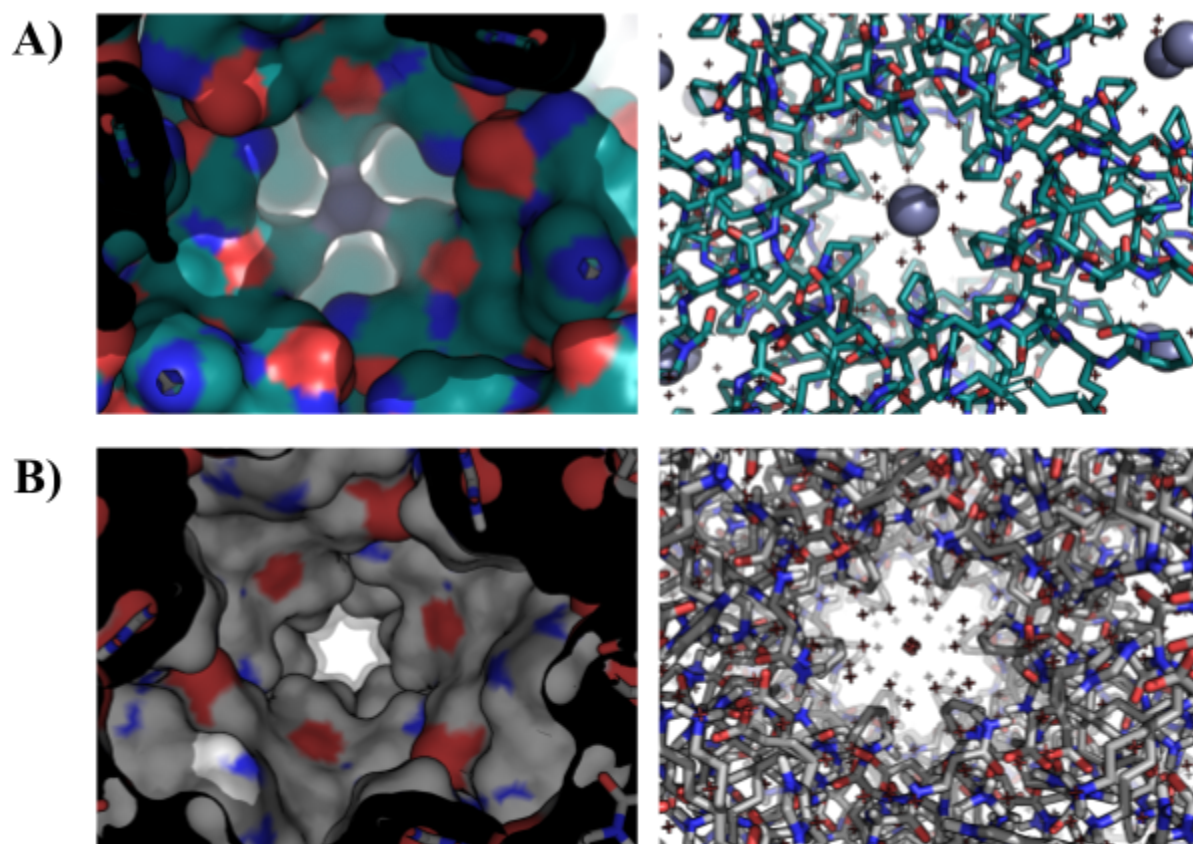
One current area of investigation is our collaboration with Tamir Gonen’s lab to use microED to shoot and solve the structures of crystals that grow with a needle-like morphology. Investigating the use of noncanonical, metal-binding amino acids is another promising direction for future MOF-inspired cyclic peptide-based materials.<sup>46,47</sup> Meerit Said, another graduate student in the Baker lab is currently working to design new crystals using a variety of noncanonical amino acids. Replacing metal ions with metal-oxo clusters is another avenue for

future investigation. Since metal-oxo clusters have very well-defined geometry, this may be a good strategy to increase the rigidity of the metal node component of future peptide-based materials.<sup>48</sup> Increasing our understanding of and ability to model the metal itself using QM or MM-based methods could also be yet another way to improve our existing computational matching and design protocol.

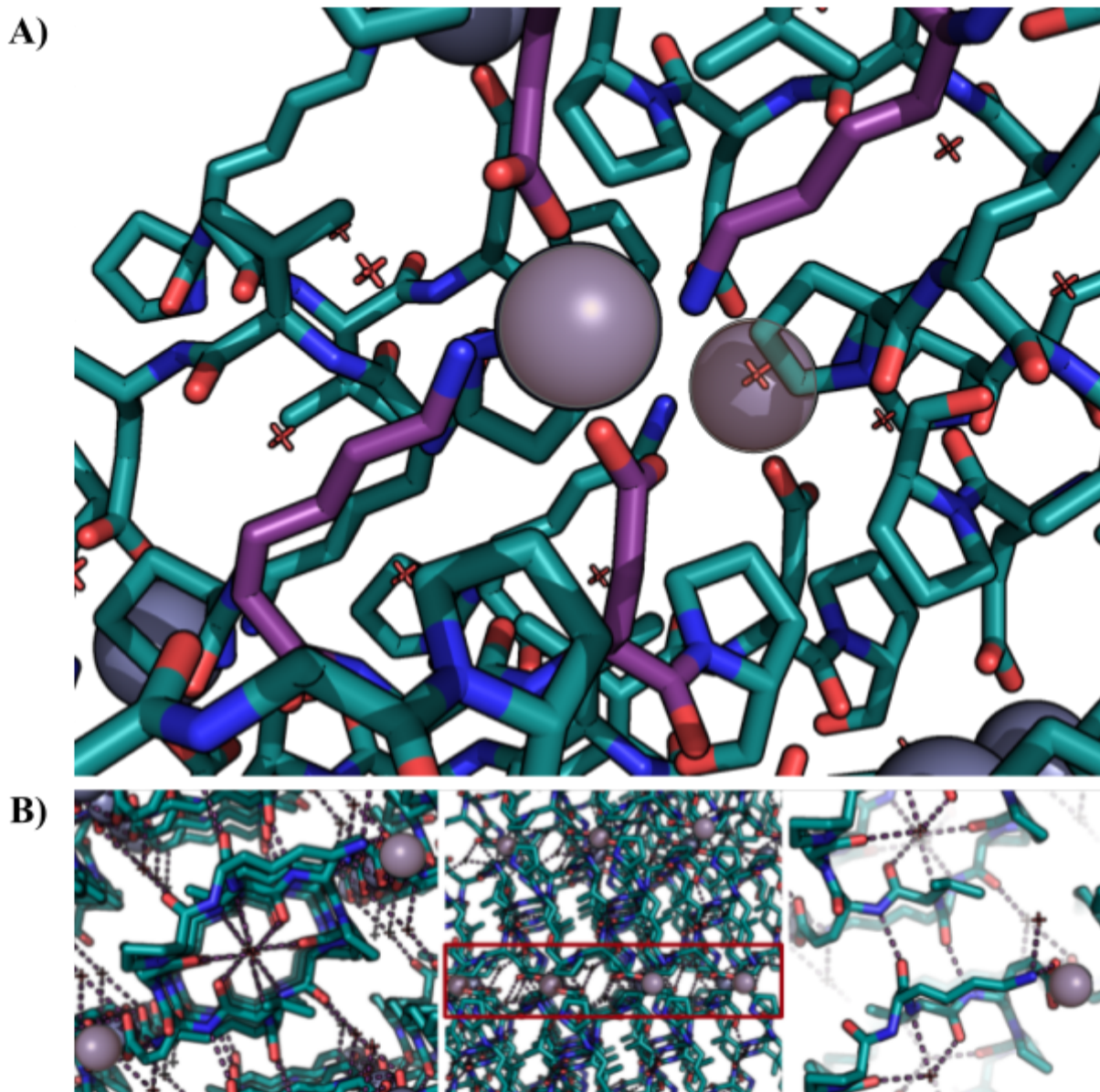
On the computational side, it would also be useful to expand the number of space groups we can design in beyond the four discussed here because it would increase the amount of information with which we can evaluate whether any given combination of peptide, metal binding residue placement, and metal ion or cluster is likely to assemble into the desired crystal space group. It would also be interesting to try and redesign the three solved crystal structures to build in new functions and better evaluate the extent of their modularity (since this is the case with some small molecule and protein-based MOFs), as well as increase the scale of their synthesis and test them for any potential emergent properties.<sup>15,49</sup>



**Figure 3.1:** A) The crystal structure (left) shows that the peptide takes on a significantly different conformation than that of the design (right) due to the placement of a d- histidine in l- Ramachandran space. The design was rendered in light gray and the crystal structure was rendered in teal. B) The crystal structure (left) is also significantly more densely packed than the original design, and the metal binding residues in the peptide are different than in the original design. C) Additional views of the metal binding site.



**Figure 3.2:** S2-6 crystallized into the same space group in both IPA (A) and a buffered pH 5 solution (B). A) In the presence of metal however, the metal simply entered into a pore in the crystal, and does not bind to any of the metal binding residues in the peptide.



**Figure 3.3:** A) Lysine and glutamic acid residues in S2-4 bind to Cu(II) ions in a tetrahedral geometry. B) This structure is mediated by (left) hydrogen bonding to a string of water molecules down the peptides' symmetry axes, (center - red box) metal binding interactions between the Cu(II) ion and lysine and glutamic acid residues from peptides across a layer, and (right) backbone-backbone interactions between adjacent neighboring peptides.

### Works Cited:

1. Furukawa H, Cordova, K.E., O’Keeffe, M., Yaghi, O.M. The Chemistry and Applications of Metal-Organic Frameworks. (2013). *Science*. 341, 1230444. <http://dx.doi.org/10.1126/science.1230444>
2. Batten, S.R., Champness, N.R., Chen, X., Garcia-Martinez, J., Kitagawa, S., Öhrström, L., O’Keeffe, M., Suh, M.P., Reedijk, J. Terminology of metal–organic frameworks and coordination polymers (IUPAC Recommendations 2013)\*. (2013). *Pure Appl. Chem.* 85, 8, 1715-1724. <http://dx.doi.org/10.1351/PAC-REC-12-11-20>
3. Hoskins, B.F., Robson, R. Infinite polymeric frameworks consisting of three dimensionally linked rod-like segments. (1989). *J. Am. Chem. Soc.* 111, 15, 5962-5964. <https://doi.org/10.1021/ja00197a079>
4. O’Keeffe, M., Yaghi, O.M. Deconstructing the Crystal Structures of Metal–Organic Frameworks and Related Materials into Their Underlying Nets. (2012). *Chem. Rev.* 112, 2, 675-702. <https://doi.org/10.1021/cr200205j>
5. Li, H., Eddaoudi, M., O’Keeffe, M., Yaghi, OM. Design and synthesis of an exceptionally stable and highly porous metal-organic framework. (1999). *Nature*. 402, 276-279. <https://doi.org/10.1038/46248>
6. Baek, J., Rungtaweeworanit, B., Pei, X., Park, M., Fakra, S.C., Liu, Y., Matheu, R., Alshimiri, S.A., Alshehri, S., Trickett, C.A., Somorjai, G.A., Yaghi, O.M. Bioinspired Metal-Organic Framework Catalysts for Selective Methane Oxidation to Methanol. (2008). *J. Am. Chem. Soc.* 140, 51, 18208-18216. <https://doi.org/10.1021/jacs.8b11525>
7. Li, J., Sculley, J., Zhou, H. Metal-Organic Frameworks for Separations. (2012). *Chem. Rev.* 112, 2, 869-932. <https://doi.org/10.1021/cr200190s>
8. Keskin, S., Kızılel, S. Biomedical Applications of Metal Organic Frameworks. (2011). *Ind. Eng. Chem. Res.* 50, 4, 1799-1812. <https://doi.org/10.1021/ie101312k>
9. Rabone, J., Yue, Y.F., Chong, S.Y., Stylianou, K.C., Bacsá, J., Bradshaw, D., Darling, G.R., Berry, N.G., Khimyak, Y.Z., Ganin, A.Y., Wiper, P., Claridge, J.B., Rosseinsky, M.J. An Adaptable Peptide-Based Porous Material. (2010). *Science*. 329, 5995, 1053-1057. <https://doi.org/10.1126/science.1190672>
10. Manton, A., Massüger, L., Rabu, P., Palivan, C., McCusker, L.B., Taubert, A. Metal-Peptide Frameworks (MPFs): “Bioinspired” Metal Organic Frameworks. (2008). *J. Am. Chem. Soc.* 130, 8, 2517-2526. <https://doi.org/10.1021/ja0762588>
11. Horcajada, P., Gref, R., Baati, T., Allan, P.K., Maurin, G., Couvreur, P., Férey, G., Morris, R.E., Serre, C. Metal-Organic Frameworks in Biomedicine. (2012). *Chem. Rev.* 112, 2, 1232-1268. <https://doi.org/10.1021/cr200256v>
12. Suzuki, Y., Cardone, G., Restrepo, D., Zavattieri, P.D., Baker, T.S., Tezcan, F.A. Self-assembly of coherently dynamic, auxetic, two-dimensional protein crystals. (2016). *Nature*. 533, 369-373. <https://doi.org/10.1038/nature17633>
13. Martí-Gastaldo, C., Antypov, D., Warren, J.E., Briggs, M.E., Chater, P.A., Wiper, P.V., Miller, G.J., Khimyak, Y.Z., Darling, G.R., Berry, N.G., Rosseinsky, M.J. Side-chain control of porosity closure in single- and multiple-peptide-based porous materials by

- cooperative folding. (2014). *Nat. Chem.* 6, 343-351.  
<https://doi-org.offcampus.lib.washington.edu/10.1038/nchem.1871>
14. Katsoulidis, A.P., Antypov, D., Whitehead, G.F.S., Carrington, E.J., Adams, D.J., Berry, N.G., Darling, G.R., Dyer, M.S., Rosseinsky, M.J. Chemical control of structure and guest uptake by a conformationally mobile porous material. (2019). *Nature*. 565, 213-217. <https://doi-org.offcampus.lib.washington.edu/10.1038/s41586-018-0820-9>
  15. Bailey, J.B., Zhang, L., Chiong, J.A., Ahn, S., Tezcan, F.A. Synthetic Modularity of Protein–Metal–Organic Frameworks. (2017). *J. Am. Chem. Soc.* 139, 8160-8166.  
<https://doi.org/10.1021/jacs.7b01202>
  16. Schnitzer, T., Paenurk, E., Trapp, N., Gershoni-Poranne, R., Wennemers, H. Peptide–Metal Frameworks with Metal Strings Guided by Dispersion Interactions. (2021). *J. Am. Chem. Soc.* 143, 2, 644-648. <https://doi.org/10.1021/jacs.0c11793>
  17. Empting, M. “An Introduction to Cyclic Peptides.” *Cyclic Peptides: From Bioorganic Synthesis to Applications*, edited by Koehnke, J., Naismith, J., van der Donk, W.A. Royal Society of Chemistry. 2017. <https://doi.org/10.1039/9781788010153-00001>
  18. Gomez-Gualdrón, D.A., Gutov, O.V., Krungleviciute, V., Borah, B., Mondloch, J.E., Hupp, J.T., Yildirim, T., Farha, O.K., Snurr, R.Q. Computational Design of Metal–Organic Frameworks Based on Stable Zirconium Building Units for Storage and Delivery of Methane. (2014). *Chem. Mater.* 26, 5632-5639.  
<https://doi.org/10.1021/cm502304e>
  19. Bernales, V., Ortuño, M.A., Truhlar, D.G., Cramer, C.J., Gagliardi, L. Computational Design of Functionalized Metal–Organic Framework Nodes for Catalysis. (2018). *ACS Cent. Sci.* 4(1): 5-19. <https://doi.org/10.1021/acscentsci.7b00500>
  20. Dighe, A.V., Nemade, R.Y., Singh, M.R. Modeling and Simulation of Crystallization of Metal–Organic Frameworks. (2019). *Processes*. 7(8), 527.  
<https://doi.org/10.3390/pr7080527>
  21. Coudert, F., Fuchs, A.H. Computational characterization and prediction of metal–organic framework properties. (2015). *Coord. Chem. Rev.* 307, 2, 211-236.  
<https://doi.org/10.1016/j.ccr.2015.08.001>
  22. Luo, Y., Ahmad, M., Schug, A., Tsotsalas, M. Rising Up: Hierarchical Metal–Organic Frameworks in Experiments and Simulations. (2019). *Adv. Mater.* 31.  
<https://doi.org/10.1002/adma.201901744>
  23. Hosseinzadeh, P., Bhardwaj, G., Mulligan, V.K., Shortridge, M.D., Craven, T.W., Pardo-Avila, F., Rettie, S.A., Kim, D.E., Silva, D., Ibrahim, Y.M., Webb, I.K., Cort, J.R., Adkins, J.N., Varani, G., Baker, D. Comprehensive computational design of ordered peptide macrocycles. (2017). *Science*. 358, 6369, 1461-1466.  
<http://doi.org/10.1126/science.aap7577>
  24. Mulligan, V. K., Kang, C. S., Sawaya, M. R., Rettie, S., Li, X., Antselovich, I., Craven, T. W., Watkins, A. M., Labonte, J. W., DiMaio, F., Yeates, T. O., & Baker, D. Computational design of mixed chirality peptide macrocycles with internal symmetry. (2020). *Protein Science*. 2020; 29: 2433– 2445. <https://doi.org/10.1002/pro.3974>
  25. Bhardwaj, G., Mulligan, V.K., Bahl, C.D., Gilmore, J.M., Harvey, P.J., Cheneval, O., Buchko, G.W., Pulavarti, S.V., Kaas, Q., Eletsky, A., Huang, P.S., Johnsen, W.A., Greisen, P.J., Rocklin, G.J., Song, Y., Linsky, T.W., Watkins, A., Rettie, S.A., Xu, X.,

- Carter, L.P., Bonneau, R., Olson, J.M., Coutsiyas, E., Correnti, C.E., Szyperski, T., Craik, D.J., Baker, D. Accurate de novo design of hyperstable constrained peptides. (2016). *Nature*. 538(7625):329-335. <https://doi.org/10.1038/nature19791>
26. King, N.P., Bale, J.B., Sheffler, W., McNamara, D.E., Gonen, S., Gonen, T., Yeates, T.O., Baker, D. Accurate design of co-assembling multi-component protein nanomaterials. (2014). *Nature*. 510, 103-108. <https://doi.org/10.1038/nature13404>
  27. 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., Davis, T.N., Gonen, T., King, N.P., Baker, D. Design of a hyperstable 60-subunit protein icosahedron. (2016). *Nature*. 535, 136-139. <https://doi.org/10.1038/nature18010>
  28. Bale, J.B., Gonen, S., Liu, Y., Sheffler, W., Ellis, D., Thomas, C., Cascio, D., Yeates, T.O., Gonen, T., King, N.P., Baker, D. Accurate design of megadalton-scale two-component icosahedral protein complexes. (2016). *Science*. 353, 6297, 389-394. <http://doi.org/10.1126/science.aaf8818>
  29. Nattermann, U.W. 2019, "A Hierarchical Approach to the Design of Protein Crystals", PhD thesis, University of Washington, Seattle. <http://hdl.handle.net/1773/45110>
  30. André, I., Bradley, P., Wang, C., Baker, D. Prediction of the structure of symmetrical protein assemblies. (2007). *PNAS*. 104 (45), 17656-17661. <https://doi.org/10.1073/pnas.0702626104>
  31. Alford, R.F., Leaver-Fay, A., Jeliazkov, 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., Kortemme, T., Gray, J.J. The Rosetta All-Atom Energy Function for Macromolecular Modeling and Design. (2017). *J. Chem. Theory Comput.* 13, 6, 3031-3048. <https://doi.org/10.1021/acs.jctc.7b00125>
  32. DiMaio, F., Leaver-Fay, A., Bradley, P., Baker, D., André, I. Modeling Symmetric Macromolecular Structures in Rosetta3. (2011). *PLoS ONE*. 6, 6: e20450. <https://doi.org/10.1371/journal.pone.0020450>
  33. Fleishman, S.J., Leaver-Fay, A., Corn, J.E., Strauch, E.-M., Khare, S.D., Koga, N., Ashworth, J., Murphy, P., Richter, F., Lemmon, G., Meiler, J., Baker, D. (2011). RosettaScripts: A Scripting Language Interface to the Rosetta Macromolecular Modeling Suite. *PLoS ONE*. 6(6): e20161. <https://doi.org/10.1371/journal.pone.0020161>
  34. Mandell, D.J., Coutsiyas, E.A., Kortemme, T. Sub-angstrom accuracy in protein loop reconstruction by robotics-inspired conformational sampling. (2009). *Nat. Methods*. 6(8):551-2. <https://doi.org/10.1038/nmeth0809-551>
  35. Coutsiyas, E.A., Seok, C., Jacobson, M.P., Dill, K.A. A kinematic view of loop closure. (2004). *J. Comput. Chem.* 25(4):510-28. <https://doi.org/10.1002/jcc.10416>
  36. Mulligan, V.K. Generalized kinematic closure (GeneralizedKIC) tutorial series [Internet]. 2017 [cited 2021 Jul 1]. Available from: [https://www.rosettacommons.org/demos/latest/tutorials/GeneralizedKIC/generalized\\_kinematic\\_closure\\_1](https://www.rosettacommons.org/demos/latest/tutorials/GeneralizedKIC/generalized_kinematic_closure_1)
  37. Mulligan, V.K. Simple cyclic peptide prediction (simple\_cycpep\_predict) application [Internet]. 2015 [cited 2021 Jul 1]. Available from: [https://www.rosettacommons.org/docs/latest/structure\\_prediction/simple\\_cycpep\\_predict](https://www.rosettacommons.org/docs/latest/structure_prediction/simple_cycpep_predict)

38. Thakkar, A., Trinh, T.B., Pei, D. Global Analysis of Peptide Cyclization Efficiency. (2013). *ACS Comb. Sci.* 15(2): 120-129. <https://doi.org/10.1021/co300136j>
39. Choi, J., Joo, S.H. Recent Trends in Cyclic Peptides as Therapeutic Agents and Biochemical Tools. (2020). *Biomol Ther (Seoul)*. 28(1): 18-24. <https://doi.org/10.4062/biomolther.2019.082>
40. Gary, D.C., Petrone, A., Li, X., Cossairt, B.M. Investigating the role of amine in InP nanocrystal synthesis: destabilizing cluster intermediates by Z-type ligand displacement. (2016). *Chem. Comm.* 53, 161-164. <https://doi.org/10.1039/C6CC07952K>
41. Laniado, J., Yeates, T.O. A complete rule set for designing symmetry combination materials from protein molecules. (2020). *PNAS*. 117, 50: 31817-31823. <https://doi.org/10.1073/pnas.2015183117>
42. Rulísek, L., Vondrásek, J. Coordination geometries of selected transition metal ions (Co<sup>2+</sup>, Ni<sup>2+</sup>, Cu<sup>2+</sup>, Zn<sup>2+</sup>, Cd<sup>2+</sup>, and Hg<sup>2+</sup>) in metalloproteins. (1998). *J. Inorg. Biochem.* 71(3-4): 115-127. [https://doi.org/10.1016/S0162-0134\(98\)10042-9](https://doi.org/10.1016/S0162-0134(98)10042-9)
43. Rämisch, S., Mulligan, V.K. Symmetry Tutorial [Internet]. 2017. [cited 2021 Jul 1]. Available from: <https://www.rosettacommons.org/demos/latest/tutorials/Symmetry/Symmetry>
44. André, I., DiMaio, F. Symmetry User's Guide [Internet]. 2019. [cited 2021 Jul 1]. Available from: [http://new.rosettacommons.org/docs/latest/rosetta\\_basics/structural\\_concepts/symmetry](http://new.rosettacommons.org/docs/latest/rosetta_basics/structural_concepts/symmetry)
45. Matthews, B.W. Racemic crystallography—Easy crystals and easy structures: What's not to like? (2009). *Protein Science*. 18(6): 1135-1138. <https://doi.org/10.1002/pro.125>
46. Renfrew, P.D., Choi, E.J., Bonneau, R., Kuhlman, B. Incorporation of Noncanonical Amino Acids into Rosetta and Use in Computational Protein-Peptide Interface Design. (2012). *PLoS ONE*. 7(3):e32637. <https://doi.org/10.1371/journal.pone.0032637>
47. O'Connor, J.J. 2021, "De novo design of passively permeable cyclic peptides and incorporation of noncanonical amino acids to improve predicted binding affinity", PhD thesis, University of Washington, Seattle.
48. Kalmutzki, M.J., Hanikel, N., Yaghi, O.M. Secondary building units as the turning point in the development of the reticular chemistry of MOFs. (2018). *Sci. Adv.* 4, 10. <https://doi.org/10.1126/sciadv.aat9180>
49. Deng, H., Grunder, S., Cordova, K.E., Valente, C., Furukawa, H., Hmadeh, M., Gándara, F., Whalley, A.C., Liu, Z., Asahina, S., Kazumori, H., O'Keeffe, M., Terasaki, O., Stoddart, J.F., Yaghi, O.M. Large-Pore Apertures in a Series of Metal-Organic Frameworks. (2012). *Science*. 336, 6084: 1018-1023. <https://doi.org/10.1126/science.1220131>
50. Dunbrack R.L., Karplus M. Backbone-dependent Rotamer Library for Proteins Application to Side-chain Prediction. (1993). *J. Mol. Biol.* 230(2):543–574. <https://doi.org/10.1006/jmbi.1993.1170>
51. Chaudhury S., Lyskov S., Gray, J.J. PyRosetta: a script-based interface for implementing molecular modeling algorithms using Rosetta. (2010). *Bioinformatics*. 26:689–691. <https://doi.org/10.1093/bioinformatics/btq007>

## Appendix I: Full Description of Computational Protocols

### Basic Peptide Design Protocol:

My computational peptide design protocol involved three key steps. 1) Enumeration of symmetric cyclic peptide scaffolds, 2) Computational design, and 3) Computational structure validation.

### *Enumeration of Symmetric Cyclic Peptide Scaffolds:*

Exhaustive enumeration of short cyclic poly-glycine peptides (called "scaffolds") was achieved using kinematic closure (KIC)-based methods. KIC and its more generalized version, GeneralizedKIC, is a robotics-inspired analytical algorithm that allows for the conformational sampling and closure of a covalently-connected chain of atoms.<sup>35,36</sup> Peptide scaffolds were enumerated using poly-glycine backbones because the glycine molecule is achiral, which allows the generated scaffolds to include positions suitable for the assignment of both L- and D- amino acids during design.

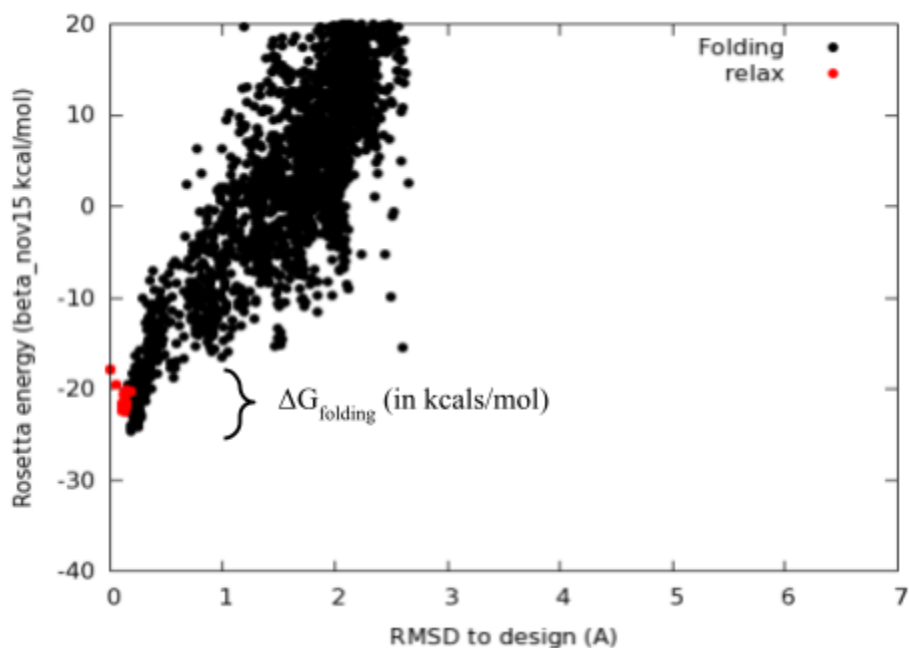
### *Computational Design:*

The full set of GenKIC closed scaffolds was separated ("clustered") into different bins based on RMSD to ensure that only unique scaffolds were designed upon. Rosetta was then used to evaluate which amino acids could be assigned in each residue position. Sequence assignments were also biased using the `aa_composition` feature to promote the assignment of certain key amino acids for ease of synthesis or influence the conformational rigidity of the design. For example, penalties were used to ensure that designs contained at least one Aspartic Acid or Glutamic Acid residue for on-bead cyclization, and weights were used to heavily favor the

assignment of Proline or  $\alpha$ -aminoisobutyric acid (AIB) residues to stabilize the peptide backbone and increase the molecule's likelihood of adopting the designed conformation.

*Computational Structure Validation:*

To ensure that designs with diverse backbone conformations were selected and synthesized, all designed peptides went through a secondary clustering step to sort them based on their Ramachandran “ABEGO” torsion classifications. From here, all peptides with total Rosetta scores of better than 0 kcal/mol were evaluated using the fragment-independent SimpleCycpepPredict application, which rapidly samples closed conformations of small cyclic peptide using the GenKIC algorithm.<sup>15</sup> SimpleCycpepPredict is used to computationally predict (via “forward folding”) whether the designed sequence is in its lowest energy conformation. **Appendix Figure 1.1** shows an ideal forward-folding funnel where each point in the plot represents an independent simulation. Structure prediction plots are evaluated qualitatively by looking at the “energy gap” (y-axis, in kcal/mol) between simulated structures  $>1$  Å RMSD to the designed structure and the Rosetta energy of the designed structure itself. The larger the energy gap, the more likely it is that a given sequence will fold into the desired designed conformation when synthesized. Only peptides that pass this “forward-folding” step were selected to be synthesized.



**Appendix Figure 1.1:** An example of a “good” structure prediction plot. Each point represents an independent simulation. The energy difference (in kcal/mol) between the score of the lowest simulated “folding” structure (in black) at 1 Å compared to that of the design (in red) corresponds to the  $\Delta G_{\text{folding}}$ .

#### Peptide MOF Design Protocol:

The MOF design protocol involved several steps: 1) Geometric “matching”, 2) Computational peptide design, 3) Computational validation and design selection.

#### *Geometric “Matching”:*

Several different strategies were employed in our attempt to design peptide MOF materials from the bottom up. Previously enumerated peptide backbones were screened to identify whether they could accommodate metal binding amino acids in geometries that are compatible with one of a handful of predefined crystal space groups as described in Chapter 3.

These enumerated scaffolds were screened using a PyRosetta script to find geometric “matches” from the existing set of scaffolds, checking each amino acid rotamer at each residue position.<sup>50</sup>

The program takes as input a large set of symmetric peptide scaffolds, and information defining the geometric relationships of symmetry elements (point groups) needed to make a 3-dimensional space group. It then identifies residues positions on the peptide scaffold where pairs of metal binding amino acids like l- or d- Histidine, Aspartic Acid, Glutamic Acid, and Cysteine can be installed in order to bind a metal ion like Zn(II) in a bidentate fashion. The program also checks that the relationship between the axes of symmetry through the peptide and the metal are consistent with what is defined for the space group it is finding “matches” for. If a match is found, the program then calculates the matched crystal’s unit cell dimensions (a, b, c, and  $\alpha$ ,  $\beta$ ,  $\gamma$  - written out in the output .pdb file as a CRYST1 line) and outputs a peptide model with metal binding amino acids already installed.

The relationship between the two symmetry axes (peptide and metal) is defined by the placement of metal binding amino acids along the peptide backbone and their rotamers. In this case, our initial computational design strategy focused on finding pairs of metal binding amino acids that can bind to the target metal in a bidentate fashion. So, the program generated matrices of metal coordinate frames representing the finely sampled rotamer clouds of each metal binding amino acid at each residue position on the peptide, and found places where the metal coordinate frame of one metal binding amino acid not only intersect with one another, but also allow for the axis of symmetry through the resulting metal ion to form the correct dihedral with the axis of symmetry through the peptide macrocycle.

*Computational Peptide Crystal Design:*

Once peptide scaffolds that are geometrically compatible with crystal space groups of interest were identified, similarly to the design of cyclic peptide monomers described above, a Rosetta XML script was written to evaluate which amino acids could be assigned in each available (non-metal binding) residue position. These amino acids were selected mostly to influence and constrain the conformational rigidity of the design in order to increase the likelihood that the “matched” metal binding residues can maintain their rotamer as well as increase the overall density of the resulting designed crystal. Frank DiMaio’s make\_lattice mover uses the CRYST1 line calculated in the previous step to set up and enforce distance, angle, and torsion constraints were used to hold the residues in place and penalize the assignment of amino acids to the open residue positions that would cause the metal binding residues to deviate from their “matched” orientation.<sup>43,44</sup>

*Computational Structure Validation and Design Selection:*

To identify the best structures from the output of the peptide crystal design script, we developed three separate “filtering” steps to ensure that: 1) the peptide crystal design isn’t so sparse (low density) that it might collapse/not form properly, 2) the peptide sequence “prefers” the crystal it is designed for more than the other crystal space groups we’ve designed for so far, and 3) the peptide monomer itself is likely to remain structured/as-designed.

To quickly screen the thousands of crystal designs and eliminate very low density structures, a PyRosetta script was written to calculate the density of each MOF design.<sup>51</sup> This was accomplished by calculating the mass of each peptide design’s asymmetric unit and using each design .pdb’s CRYST1 line to calculate the unit cell volume from the calculated unit cell

dimensions. Each design's density was calculated in this way, and only crystal designs with a density above 0.3 atoms/Å<sup>3</sup> were selected for consideration.

Simultaneously, to identify which design sequences “preferred” the crystal space group it is designed for over the other selected space groups, another PyRosetta script was written to “drape” each permutation of each sequence of each peptide design onto every other scaffold in that same space group, “relax” each of these new structures, and then calculate their Rosetta energy scores. If the original design sequence, in its original permutation, “prefers” the crystal structure it is designed for, then it should be calculated to have the lowest energy. If not, then that means that another permutation of the sequence or an entirely different sequence “prefers” the crystal structure more than the sequence it was designed to have, and therefore those designs were not selected to be synthesized.

Finally, as described previously, in order to ensure that the newly designed peptide monomer itself is likely to remain structured/as designed, all peptides with total Rosetta scores of better than 0 kcal/mol were evaluated using the fragment-independent `simple_cycpep_predict` application. Very few designs made it past this filter, so we chose to synthesize designs that at least sampled the designed conformation.

RosettaScripts XMLs and Supporting Scripts.<sup>33</sup>

#### *Peptide Design:*

Instructions and inputs for running the peptide design script can be found in the Supplementary Information of: Mulligan, V. K., Kang, C. S., Sawaya, M. R., Rettie, S., Li, X., Antselovich, I., Craven, T. W., Watkins, A. M., Labonte, J. W., DiMaio, F., Yeates, T. O., & Baker, D. (2020)

Computational design of mixed chirality peptide macrocycles with internal symmetry. *Protein Science*. 2020; 29: 2433– 2445. <https://doi.org/10.1002/pro.3974>.

### *Crystal Design:*

Additional inputs (resfiles) for the below script can be found in the Supplementary Information of: Mulligan, V. K., Kang, C. S., Sawaya, M. R., Rettie, S., Li, X., Antselovich, I., Craven, T. W., Watkins, A. M., Labonte, J. W., DiMaio, F., Yeates, T. O., & Baker, D. (2020) Computational design of mixed chirality peptide macrocycles with internal symmetry. *Protein Science*. 2020; 29: 2433– 2445. <https://doi.org/10.1002/pro.3974>.

These scripts were written by Christine S. Kang:

### Example Execution Script - “run\_9res\_P1432.sh”:

```
#!/bin/sh
#SBATCH -p medium
#SBATCH --mem=4g
#SBATCH -o log

../../../../Rosetta/main/source/bin/rosetta_scripts.default.linuxgccrelease
@inputs/working_peptide_xtal_design.flags
```

### Example Rosetta Options File - “working\_peptide\_xtal\_design.flags”:

```
-l inputs/all.list
-nstruct 1
-parser:protocol inputs/working_peptide_xtal_design.xml
-crystal_refine
-ex1
-ex2
-out:file:scorefile score.sc
-matdes::num_subs_building_block 2
-no_his_his_pairE
-per_chain_renumbering
-out:path:all output/
-detect_disulf false
-symmetry_definition CRYST1
-overwrite
-out:path:pdb outputs
```

### Example Rosetta XML Crystal Design Script - “working\_peptide\_xtal\_design.xml”:

```

<ROSETTASCRIPTS> #This script will be used for designing peptide+metal ion
xtals - last run 11/23/2020 CSK
  <SCOREFXNS>
    <ScoreFunction name="beta" weights="ref2015" symmetric="1" />
    <ScoreFunction name="beta_ncs" weights="ref2015" symmetric="1" >
      <Reweight scoretype="dihedral_constraint" weight="1.0" />
    </ScoreFunction>
    <ScoreFunction name="beta_soft" weights="ref2015" symmetric="1" >
      <Reweight scoretype="fa_rep" weight="0.05" />
    </ScoreFunction>
    <ScoreFunction name="beta_cst_highhbond" weights="ref2015_cst.wts"
symmetric="true" >
      <Reweight scoretype="chainbreak" weight="25.0" />
      <Reweight scoretype="hbond_sr_bb" weight="10.0" />
      <Reweight scoretype="hbond_lr_bb" weight="10.0" />
    </ScoreFunction>

    <ScoreFunction name="beta_cst_highhbond_comp" weights="ref2015_cst.wts"
symmetric="true" >
      <Reweight scoretype="chainbreak" weight="25.0" />
      <Reweight scoretype="hbond_sr_bb" weight="10.0" />
      <Reweight scoretype="hbond_lr_bb" weight="10.0" />
      <Reweight scoretype="aa_composition" weight="1.0" />
      <Reweight scoretype="aspartimide_penalty" weight="1.0" />
    </ScoreFunction>
  </SCOREFXNS>

  <RESIDUE_SELECTORS>
    <ResidueName name="metal_binding"
residue_name3="ASP, GLU, HIS, DAS, DGU, DHI" />
    <Not name="designable" selector="metal_binding" />
    <Phi name="pos_phi" select_positive_phi="true"
ignore_unconnected_upper="false"/>
    <Not name="neg_phi" selector="pos_phi" />
    <And name="designable_posphi" selectors="designable,pos_phi" />
    <And name="designable_negphi" selectors="designable,neg_phi" />
    <Index name="asym_unit" resnums="1-3" />
    <Index name="select_first_res" resnums="1" />
    <Index name="select_last_res" resnums="3" />
    <And name="designable_asym_unit" selectors="designable,asym_unit" />
  </RESIDUE_SELECTORS>

  <PACKER_PALETTES>
    <CustomBaseTypePackerPalette name="palette"
additional_residue_types="AIB, DALA, DPHE, DILE, DLYS, DLEU, DMET, DASN, DPRO, DGLN, DARG
, DSER, DTHR, DVAL, DTRP, DTYR"/>
  </PACKER_PALETTES>

  <TASKOPERATIONS>
    <ExtraRotamersGeneric name="extrarot" ex1="true" ex2="true"
ex3="false" ex4="false" extrachi_cutoff="3" />

```

```

        <ReadResfile name="laa" filename="inputs/laa.resfile"
selector="designable_negphi" />
        <ReadResfile name="daa" filename="inputs/daa.resfile"
selector="designable_posphi" />
        <OperateOnResidueSubset name="no_design_metal_binding_positions"
selector="metal_binding" >
            <PreventRepackingRLT/>
        </OperateOnResidueSubset>
    </TASKOPERATIONS>

    <FILTERS>
        <ShapeComplementarity name="sc_filt" jump="1" verbose="1" min_sc="0.2"
write_int_area="1" confidence="0"/>
        <ScoreType name="fa_rep" score_type="fa_rep" threshold="4000"
scorefxn="beta" confidence="1" />
        <ScoreType name="fa_rep_reporter" score_type="fa_rep" threshold="2000"
scorefxn="beta" confidence="0" />
        <ScoreType name="total_score_reporter" score_type="total_score"
threshold="2000" scorefxn="beta" confidence="0" />

        <OversaturatedHbondAcceptorFilter name="oversat"
max_allowed_oversaturated="0" hbond_energy_cutoff="-0.1"
consider_mainchain_only="true" scorefxn="beta" />
    </FILTERS>

    <MOVERS>
        <MakeLatticeMover name="make_lattice" contact_dist="20" />
        <ModifyVariantType name="upper_cutpoints" add_type="CUTPOINT_UPPER"
residue_selector="select_first_res" />
        <ModifyVariantType name="lower_cutpoints" add_type="CUTPOINT_LOWER"
residue_selector="select_last_res" />
        <DeclareBond name="bond1" res1="9" res2="1" atom1="C" atom2="N"
add_termini="false" />
        <Small name="small_perturbation" angle_max="5.0" scorefxn="beta" />
        <FastRelax name="frlx1" repeats="1" scorefxn="beta_cst_highhbond"
min_type="dfpmin">
            <MoveMap name="frlx_mm1" >
                <Span begin="1" end="999" bb="false" chi="false" />
            </MoveMap>
        </FastRelax>
        <FastRelax name="frlx2" repeats="1" scorefxn="beta_cst_highhbond"
min_type="dfpmin">
            <MoveMap name="frlx_mm2" >
                <Span begin="1" end="999" bb="true" chi="true" />
            </MoveMap>
        </FastRelax>
        <AddCompositionConstraintMover name="aacomp_all"
filename="inputs/all.comp" selector="designable_asym_unit" />
        <PackRotamersMover name="fdes1" scorefxn="beta_cst_highhbond_comp"
task_operations="laa,no_design_metal_binding_positions"
packer_palette="palette" />

```

```

    <FastRelax name="frlx3" repeats="1" scorefxn="beta" min_type="dfpmin"
task_operations="extrarot" >
    <MoveMap name="frlx_mm3" >
        <Span begin="1" end="999" bb="true" chi="true" />
    </MoveMap>
</FastRelax>

</MOVERS>
<APPLY_TO_POSE>
</APPLY_TO_POSE>
<PROTOCOLS>
    <Add mover="make_lattice" />
    <Add filter="sc_filt" />
    <Add filter="fa_rep" />
    <Add filter="fa_rep_reporter" />
    <Add filter="total_score_reporter" />
    <Add mover="small_perturbation" />
        <Add mover="aacomp_all" />
        <Add mover="fdes1" />
    <Add filter="sc_filt" />
    <Add filter="fa_rep" />
</PROTOCOLS>
</ROSETTASCRIPTS>

```

### *Computational Structure Validation - Peptide “Forward Folding”:*

Instructions and inputs for running computational structure validation can be found in the Supplementary Information of: Mulligan, V. K., Kang, C. S., Sawaya, M. R., Rettie, S., Li, X., Antselovich, I., Craven, T. W., Watkins, A. M., Labonte, J. W., DiMaio, F., Yeates, T. O., & Baker, D. (2020) Computational design of mixed chirality peptide macrocycles with internal symmetry. *Protein Science*. 2020; 29: 2433– 2445. <https://doi.org/10.1002/pro.3974>.

### PyRosetta Scripts.<sup>51</sup>

#### *Geometric Matching:*

This program was written by Will Sheffler, PhD. Information on the most up-to-date version of the geometric matching script can be found at: [github.com/willsheffler/mof](https://github.com/willsheffler/mof)

*Computational Structure Validation - Density Calculations:*

This script was written by Meerit Y. Said and is shared here with her permission:

```
## This script takes a list of pdb names without a path or a '.pdb' at the end
and determines their atomic density.
## The unit cell volume is calculated based on CRYST1 line

## Assumptions:
## 1. each unit cell contains one asymmetric unit. This is based on Will S.
original peptide crystal "matches".
## 2. The unit cell is cubic. This is true for now but will have to change
for different space groups.

from pyrosetta import *
import re
import argparse
import glob
import sys
import math
from pathlib import Path
from astropy.table import Table
from astropy.io import ascii

#takes a list of pdb names
def parse_arguments(argv):
    parser = argparse.ArgumentParser(description="takes list of peptide names")
    parser.add_argument('PEP_LIST', nargs = 1, type = str, help = 'List of pdb
names')
    args = parser.parse_args()
    return args

#calculates the cubic volume of the unit cell. Takes in the file path to a pdb
def unit_cell_volume (pdb_file_path):
    file_path = Path(pdb_file_path)
    with open (file_path) as pdb:
        #read Cryst1 line
        cryst1 = []
        for line in pdb:
            if line.startswith("CRYST1"):
                cryst1 = (line).split()
        print("CRYST1! LINE IS: ", cryst1)
        #extract x,y,z, and angles from cryst1 based on digit patterns
        x_y_z = [float(i) for i in cryst1[1:7]]

        #determine crystal system
        if x_y_z[0] == x_y_z[1] == x_y_z[2] and x_y_z[3] == x_y_z[4] ==
x_y_z[5] == 90.00:
            crystal_system = "cubic"
        elif x_y_z[0] == x_y_z[1] != x_y_z[2] and x_y_z[3] == x_y_z[4] == 90.00
and x_y_z[5] == 120.00:
            crystal_system = "hexagonal"
```

```

        elif x_y_z[0] == x_y_z[1] != x_y_z[2] and x_y_z[3] == x_y_z[4] ==
x_y_z[5] == 90.00:
            crystal_system = "tetragonal"
        elif x_y_z[0] == x_y_z[1] == x_y_z[2] and x_y_z[3] == x_y_z[4] ==
x_y_z[5] != 90.00:
            crystal_system = "rhombohedral"
        elif x_y_z[0] != x_y_z[1] != x_y_z[2] and x_y_z[3] == x_y_z[4] ==
x_y_z[5] == 90.00:
            crystal_system = "orthorhombic"
        elif x_y_z[0] != x_y_z[2] and x_y_z[3] == x_y_z[4] == 90.00 and
x_y_z[5] != 90.00:
            crystal_system = "monoclinic"
        elif x_y_z[0] != x_y_z[1] != x_y_z[2] and x_y_z[3] != x_y_z[4] !=
x_y_z[5]:
            crystal_system = "triclinic"

#determine lattice centering and number of asu in a unit cell
if crystl[7] == "P":
    lattice_centering = "primitive"
    N_asu = "1"
elif crystl[7] == "I":
    lattice_centering = "body centered"
    N_asu = "2"
elif crystl[7] == "F":
    lattice_centering = "face centered"
    N_asu = "4"
elif crystl[7] == "A" or crystl[7] == "B" or crystl[7] == "C":
    lattice_centering = "base centered"
    N_asu = "2"
elif crystl[7] == "R":
    lattice_centering = "D centered"
    N_asu = "1"

#calculate cubic volume of unit cell
if crystal_system == "cubic":
    volume = x_y_z[0] * x_y_z[1] * x_y_z[2]
    return volume
elif crystal_system == "hexagonal":
    volume = 0.866 * x_y_z[0] * x_y_z[0] * x_y_z[2]
    return volume
elif crystal_system == "tetragonal":
    volume = x_y_z[0] * x_y_z[0] * x_y_z[2]
    return volume
elif crystal_system == "rhombohedral":
    volume = [sqrt(1- 3 * (math.cos(x_y_z[3]))**2 + 2 *
(math.cos(x_y_z[3]))**3)] * (x_y_z[0] ** 3)
    return volume
elif crystal_system == "orthorhombic":
    volume = x_y_z[0] * x_y_z[1] * x_y_z[2]
    return volume
elif crystal_system == "monoclinic":
    volume = x_y_z[0] * x_y_z[1] * x_y_z[2] * math.sin(x_y_z[4])

```

```

        return volume
    elif crystal_system == "triclinic":
        volume = x_y_z[0] * x_y_z[1] * x_y_z[2]
    return volume

#calculates the energy of one chain. Takes in the file path to a pdb and the
length of the asymmetric unit.
def get_mass_asym_unit (pdb_file_path):
    file_path = Path(pdb_file_path)
    with open (file_path) as pdb:
        #read Cryst1 line
        cryst1=[]
        for line in pdb:
            if line.startswith("CRYST1"):
                cryst1 = (line).split()

        #extract x,y,z, and angles from cryst1 based on digit patterns
        x_y_z = cryst1[1]

    #use ref 2015 score function
    sfxn = get_score_function()

    #creates pose from one chain as the asymmetric unit
    pose = pose_from_pdb(pdb_file_path)

    #make a list of atom types in a pose and count the number of atoms in that
    list.
    Atoms = []
    for res in pose:
        for i in range(1, len(res.atoms())+1):
            Atoms.append(res.atom_name(i))

    #calculate mass of asym unit pose in grams/pose not moles
    mass = pyrosetta.rosetta.core.pose.mass(1,3,pose) / (6.022*10**23)

    #returns number of atoms and energy of the unit cell
    return [mass, sfxn(pose)]

if __name__ == '__main__':
    pyrosetta.init('-crystal_refine -in:file:extra_res_fa
/projects/peptides/mofs/params_files/HDZ.params' )
    args = parse_arguments(sys.argv)

    #takes in a list of pdb names. ex tmp.list
    input_file = args.PEP_LIST[0]
    data_rows = []

    with open (input_file) as f:
        for lines in f:
            #adds the path of the designed pdb to the name. This way we can
            keep the designed peptides as pairs in the loop.
            designed_name = 'input/' + lines.rstrip()

```

```

#calculates the volume and density of each asymmetric unit
vol = (unit_cell_volume(design_name))

#####REMEMBER TO CHANGE BASED ON SPACE GROUP#####
#####density of P23 space group using multi = 12 according to Frank#####
#####
density =
round((get_mass_asym_unit(design_name)[0]*12)/(vol*10**-24), 6)
data_rows.append((lines.rstrip(), vol, density) )

#saves data in a table format
data = Table(rows=data_rows, names= ('PDB', 'UNIT_CELL_VOLUME',
'Density_g/cm^3'))

#export data as energy_data.list so it can be sorted and filtered later.
ascii.write(data, 'energy_data_g_cm.list', overwrite=True,
format='fixed_width' )

```

### *Computational Structure Validation - Crystal “Forward Folding” AKA “Drape\_and\_Score”:*

This script was written by Christine S. Kang with help from Hua Bai, PhD:

```

import pyrosetta
from pyrosetta import rosetta
import numpy as np

pyrosetta.init('-extra_res_fa /home/csykang/scripts/Params/HZD.params
-in:file:fullatom -symmetry_definition CRYST1')
path_to_scaffolds =
str("/projects/peptides/mofs/mof_design/xtal_matches_WS_20201119/12mers/I213/al
1/")

# This script grabs the sequences from the designed pdbs and determine the
extent to which a given sequence is favored by a given peptide xtal lattice
# This is done by draping each sequence and circular permutations of the
sequence over a series of lattices, then scoring each draped sequence to see
which peptide + scaffold combinations work the best.

_DEBUG = False

### defs ###
def read_in_pdb():
    # Reads a pdbs.list file to get the paths to all of the pdbs I want to
test, then opens the pdbs with import_pose.pose_from_file
    # Returns a list, raw_pose_list, that contains all of the poses
    raw_pose_path_list = []
    raw_pose_list = []
    pdbslist = open("pdbs1.list", "r")
    lines = pdbslist.read().splitlines()
    for line in lines:
        raw_pose_path_list.append(line)

```

```

for path in raw_pose_path_list:
    raw_pose = rosetta.core.import_pose.pose_from_file(path)
    raw_pose_list.append(raw_pose)
return raw_pose_list

def cyc_align(pose):
    # Takes in a pose, cyclizes it, then aligns it to the z-axis
    # Returns a pose
    pcm = rosetta.protocols.cyclic_peptide.PeptideCyclizeMover()
    ata=rosetta.protocols.cyclic_peptide.SymmetricCycpepAlign()
    pcm.apply(pose)
    ata.apply(pose)
    return pose

def mutate_residues(t,r,p):
    mut = rosetta.protocols.simple_moves.MutateResidue()
    mut.set_target(t)
    mut.set_res_name(r)
    mut.set_preserve_atom_coords(False)
    mut.apply(p)
    return p

def fix_HZD_termini(s):
    substring = ":"
    if s == "HZD":
        return "HIS"
    elif substring in s:
        return s.split(':')[0]
    else:
        return s

def output_data(seq_pdb, scaff_score_dictionary):
    seq_name = seq_pdb.split(".pdb")[0]

    file_name = 'drape_seq_summaries/sequence_{}_scores.txt'.format(seq_name)
    o = open(file_name, "w")
    o.write("DESIGNED SEQ PDB NAME: %s\n" % seq_pdb)

    for scaff in scaff_score_dictionary:
        o.write("SCAFFOLD: %s\t\t" % scaff)
        o.write("SEQ_PERMUTATION, SCORE: %s\n" % scaff_score_dictionary[scaff])

def minimize(p,sfxn):
    movemap = rosetta.core.kinematics.MoveMap()
    movemap.set_chi(True)
    movemap.set_bb(False)
    movemap.set_jump(False)
    minmover =
pyrosetta.rosetta.protocols.minimization_packing.MinMover(movemap, sfxn,
'linmin', 0.001, True)
    for i in range(5):
        minmover.apply(p)

```

```

def cyclic_perm(list_of_seq):
    n = len(list_of_seq)
    result = []
    for j in range(n):
        def f(l, k=j):
            return list(map(lambda i: l[i - k], range(n)))
        result.append(f)
    return result

def check_and_remove_variant(input_pose):
    # util function to remove the terminal variant.
    # Calculates the distances between fragment pairs and find the correct atom
    (for example, "N", "C") to declare bond and determine connecting direction
    for in_idx in range(1, input_pose.size()+1):
        upper =
input_pose.residue(in_idx).has_variant_type(pyrosetta.rosetta.core.chemical.VariantType.UPPER_TERMINUS_VARIANT)
        if upper:
            pyrosetta.rosetta.core.pose.remove_variant_type_from_pose_residue(
input_pose, pyrosetta.rosetta.core.chemical.UPPER_TERMINUS_VARIANT, in_idx)

        lower =
input_pose.residue(in_idx).has_variant_type(pyrosetta.rosetta.core.chemical.VariantType.LOWER_TERMINUS_VARIANT)
        if lower:
            pyrosetta.rosetta.core.pose.remove_variant_type_from_pose_residue(
input_pose, pyrosetta.rosetta.core.chemical.LOWER_TERMINUS_VARIANT, in_idx)

def create_xyz_array(input_pose):
    # go through all the heavyatoms of the pose, extract the xyz coordinate,
    and put it into a huge xyz numpy array. Use a dictionary to keep track the
    residue and atom index for each numpy array entry
    xyz_list = []
    idx_info_dict = {}
    count = 0
    for res_i in range(1, input_pose.size()+1):
        this_res = input_pose.residue(res_i)
        if this_res.is_virtual_residue(): continue
        this_aa3 = this_res.name3().strip()
        if this_aa3 == "ZN":continue
        for atom_i in range(1, 5):
            this_atom_name = this_res.atom_name(atom_i).strip()
            this_xyz = this_res.atom(this_atom_name).xyz()
            this_array = [this_xyz.x, this_xyz.y, this_xyz.z]
            xyz_list.append(this_array)
            idx_info_dict[count] = [res_i, atom_i, this_atom_name]
            count += 1
    return np.array(xyz_list), idx_info_dict

```

```

def connect_downstream_frag(input_lattice, ref_frag_key = 1, dist_cutoff =
2.0):
    from scipy.spatial.distance import cdist
    # This function will find the downstream fragment of a specified fragment
    in a lattice and declare a bond between this fragment and the downstream one.
    ## step1: extract all the fragments, and clean up by removing all the
    terminal variants ##
    fragments = input_lattice.split_by_chain()
    for frag_key, each_frag in enumerate(fragments, start=1):
        check_and_remove_variant(each_frag)
    # Go through fragments, create a dictionary storing the fragment
    information
    # The fragment key is actually also the chain index of the lattice, 1-based
    index
    frag_dict = {}
    for frag_key, each_frag in enumerate(fragments, start=1):
        xyz_array, idx_info_dict = create_xyz_array(each_frag)
        frag_dict[frag_key] = (each_frag, xyz_array, idx_info_dict)

    # Go through fragment and find the fragment corresponding to the specified
    fragment (reference fragment, find its neighbor fragments
    # It only need to connect the downstream one, due to the symmetry
    connections = {}
    for frag_key, frag_value in frag_dict.items():
        # skip the specified fragment
        if frag_key == ref_frag_key: continue

        # collect the query and ref information
        query_xyz      = frag_value[1]
        query_info_dict = frag_value[2]

        ref_xyz      = frag_dict[ref_frag_key][1]
        ref_info_dict = frag_dict[ref_frag_key][2]

        if query_xyz.size > 0:
            pair_dist = cdist(query_xyz, ref_xyz)
            query_indices, ref_indices = np.where(pair_dist < dist_cutoff)
            for query_idx, ref_idx in zip(query_indices, ref_indices):
                query_atom = query_info_dict[query_idx][2]
                ref_atom   = ref_info_dict[ ref_idx][2]

                query_resnum = query_info_dict[query_idx][0]
                ref_resnum   = ref_info_dict[ ref_idx][0]

                if f"{ref_atom}{query_atom}" == "CN":
                    if ref_frag_key == 1:
                        res1 = ref_resnum
                    else:
                        res1 = input_lattice.chain_end(ref_frag_key-1) +
ref_resnum

                atom1 = ref_atom
                if frag_key == 1:

```

```

        res2 = query_resnum
    else:
        res2 = input_lattice.chain_end(frag_key-1) +
query_resnum
        atom2 = query_atom
        connections["down"] = ((ref_frag_key, frag_key), (res1,
atom1, res2, atom2))

    declare_bond = pyrosetta.rosetta.protocols.cyclic_peptide.DeclareBond()
    print(f'Declaring bond of the specified fragment
{connections["down"][0][0]} to the downstream fragment:
{connections["down"][0][1]}')
    declare_bond.set(connections["down"][1][0], connections["down"][1][1],
connections["down"][1][2], connections["down"][1][3], add termini = False)
    declare_bond.apply(input_lattice)
    return None

def connect_downstream_frag_simplified(input_lattice, ref_frag_key = 1,
dist_cutoff = 2.0):
    from scipy.spatial.distance import cdist
    # This function will find the downstream fragment of a specified fragment
in a lattice, and declare a bond between this fragment and the downstream one.
    ## step1 extract all the fragments ##
    fragments = input_lattice.split_by_chain()

    ref_frag = fragments[ref_frag_key]
    ref_indices = [x for x in range(1, ref_frag.size()+1) if
ref_frag.residue(x).is_protein()]
    if not ref_indices:
        print("please choose another ref fragment")
        return None
    ref_frag.sequence()
    ref_C_res = ref_frag.residue(ref_indices[-1])
    ref_C_xyz = ref_C_res.atom("C").xyz()
    ## Go through fragments and create a list storing the connections between N
and C
    connections = []
    for frag_key, each_frag in enumerate(fragments, start=1):
        if frag_key == ref_frag_key: continue
        frag_indices = [x for x in range(1, each_frag.size()+1) if
each_frag.residue(x).is_protein()]
        if not frag_indices: continue
        N_res = each_frag.residue(frag_indices[0])
        N_xyz = N_res.atom("N").xyz()
        CN_dist = np.linalg.norm(ref_C_xyz - N_xyz)
        if CN_dist < dist_cutoff:
            if ref_frag_key == 1:
                res1 = ref_indices[-1]
            else:
                res1 = input_lattice.chain_end(ref_frag_key-1) +
ref_indices[-1]
            atom1 = "C"

```

```

        res2 = input_lattice.chain_end(frag_key-1) + frag_indices[0]
        atom2 = "N"
        connections.append( ((ref_frag_key, frag_key), (res1, atom1, res2,
atom2)) )

    declare_bond = pyrosetta.rosetta.protocols.cyclic_peptide.DeclareBond()
    for keys, connection in connections:
        print(f'Declaring bond of the specified fragment {keys[0]} to the
downstream fragment: {keys[1]}')
        declare_bond.set(connection[0], connection[1], connection[2],
connection[3], add termini = False)
        declare_bond.apply(input_lattice)
    return None

# Read in designed pdbs
designed_pdb = read_in_pdb() # imports poses - requires a "pdb.list" file
containing paths
sfxn =
rosetta.core.scoring.ScoreFunctionFactory.create_score_function('ref2015')

# Make xml objs for specific Movers
xml = rosetta.protocols.rosetta_scripts.XmlObjects.create_from_string("""
<RESIDUE_SELECTORS>
    <Index name="peptide" resnums="1-12" />
</RESIDUE_SELECTORS>
<MOVERS>
    <MakeLatticeMover name="mlm" contact_dist="20" />
</MOVERS>
""")
mlm = xml.get_mover("mlm")

# Put designed sequences into a dictionary
dictionary_designed = {}
for pdb in designed_pdb:
    pdb_name = pdb.pdb_info().name().split('/')[1] # gets the pdb name for
outputs later
    res_index = list(range(1,rosetta.core.pose.chain_end_res(pdb,1)+1))

    seq_list = []
    for res in range(1, rosetta.core.pose.chain_end_res(pdb, 1)+1):
        seq_list.append(fix_HZD_termini(pdb.residue(res).name()))

    seq_nozn_list = [x for x in seq_list if x != "ZN"]
    cp_seq_list = []
    for seq in cyclic_perm(seq_nozn_list): # cyclic permutations
        cp_seq_list.append(seq(seq_nozn_list))
    print(seq)

    dictionary_designed.update(dict({pdb_name : cp_seq_list}))

# depending on what you want, this could be either all of the scaffolds, or
just the scaffolds that are comparable (same metal binding ligands)

```

```

ext = str(".pdb")
dictionary_scaff = {}

for key in dictionary_designed.keys():
    list_of_tuples = []
    k = key.split('_0001.pdb')[0]
    k_n = "".join((k,ext))
    s_l = "".join((path_to_scaffolds,k_n))
    scaff = rosetta.core.import_pose.pose_from_file(str(s_l))

    for resnum in range(1, int(rosetta.core.pose.chain_end_res(scaff,1))):
        print(resnum, scaff)
        mutate_residues(resnum, 'GLY', scaff)

    mlm.apply(scaff) # apply MakeLatticeMover
    connect_downstream_frag_simplified(scaff, ref_frag_key = 1, dist_cutoff =
2.0)

    polygly_score = round(sfxn(scaff), 2)

    list_of_tuples.append((scaff, polygly_score))
    dictionary_scaff.update(dict({k_n : list_of_tuples}))
    # key = name of the scaffold, value = list of tuples containing (poly gly
lattice pose (called "scaff") and the score of "scaff")

    # Mutate each scaffold to the sequence saved in the dictionary
dictionary_scores = {}
for pep_seq in dictionary_designed:
    dictionary_seq_scaff = {}
    for scaff, info in dictionary_scaff.items():
        for scaff_pdb in info:
            permutation_counter = 1
            list_of_tuples = []
            for seq_permutation in dictionary_designed[pep_seq]:
                for resnum in range(1,
int(rosetta.core.pose.chain_end_res(scaff_pdb[0],1))):
                    mutated_pose =
mutate_residues(resnum, seq_permutation[resnum-1], scaff_pdb[0])

                    minimize(mutated_pose, sfxn) # to fix the wonky side chains
                    mut_score = round(sfxn(mutated_pose), 2) # score the peptide,
round makes it output to 2 decimal places

                    # Subtracting the poly_gly score from the "total_score" value
so that we're only looking at the effects of the designed residues.
                    total_score = mut_score - scaff_pdb[1]

                    if _DEBUG:
                        total_energies = mutated_pose.energies().total_energies()
                        fa_rep =
mutated_pose.energies().total_energies()[pyrosetta.rosetta.core.scoring.ScoreTy
pe.fa_rep]

```

```
mutated_pose.dump_pdb('sequence_{}_permutation_number_{}_in_scaffold_{}'.format
(pep_seq.split(".pdb")[0].split("mofdock_asym_")[1],permutation_counter,
scaff))
        list_of_tuples.append((permutation_counter, total_score))
        dictionary_seq_scaff.update(dict({scaff : list_of_tuples}))
        permutation_counter +=1

# output data to a .txt file.
    output_data(pep_seq, dictionary_seq_scaff)
```

## Appendix II: Peptide Synthesis, Purification, and Characterization

### Peptide Synthesis:

Fmoc-amino acids were purchased from P3 Biosystems. Acetonitrile was purchased from Fisher Chemical. Peptides were synthesized on a 0.1 mmol scale using a CEM Liberty Blue microwave peptide synthesizer. A typical SPPS reaction cycle includes Fmoc deprotection, washing, coupling, and post-coupling washing steps.

#### *Synthesis of Peptides < 12 Residues in Length:*

For sequences with fewer than twelve residues, Fmoc-Glu(Wang resin LL)-ODmab and Fmoc-Asp(Wang resin LL)- ODmab from Novabiochem were used for the synthesis (0.29 mmol/g loading). After synthesis, the peptides' termini were deprotected in 2% Hydrazine in Dimethylformamide (DMF) before the peptide was cyclized on-bead. The cyclized peptides were cleaved off the resin by treatment with 10mL TFA/H<sub>2</sub>O/DODT/TIPS (92.5:0.25:0.25:0.25, v,v,v,v) for 2.5 hours before evaporating off any residual TFA using a rotovap. The crude peptide was then obtained after precipitation in cold diethyl ether.

#### *Synthesis of Peptides > 12 Residues in Length:*

For sequences with more than twelve residues and/or with no aspartic acids or glutamic acids, Cl-TCP(Cl) resin from CEM was used as solid support. The linear peptide was cleaved from the resin using 1% (v/v) TFA in dichloromethane (DCM). The protected peptide in DCM was drained into 1:1 acetonitrile and water, rotovapped off, then lyophilized. The resulting powder was redissolved (to 1mM concentration) in DCM and 2 eq. of (7-Azabenzotriazol-1-yloxy)tripyrrolidinophosphonium hexafluorophosphate (PyAOP) was added directly to the solution. The solution was left to equilibrate on a magnetic stirrer for 30

minutes, then 10 eq. of N,N-diisopropylethylamine (DIEA) was added dropwise. The cyclization reaction was left stirring overnight. The solution was rotovapped down to 1-2 mL, and the cyclic peptide was then deprotected, precipitated and purified as described in Section 3.1.

#### Peptide Purification:

The crude pellet was dried down and dissolved in 50:50 ACN:H<sub>2</sub>O. After the identity of the major product was confirmed by electrospray ionization mass spectrometry (ESI-MS) on a Thermo Scientific TSQ Quantum Access mass spectrometer, peptides were purified using an Agilent Infinity Preparative high-pressure liquid chromatography (HPLC) system on an Agilent Zorbax SB-C18 column. A linear gradient of 1% /min was used (ACN with 0.1% TFA) at a flow rate of 5 mL/min to purify peaks. Elution peaks were monitored and collected using UV-vis detection at an absorbance of 214 nm. Post-purification, peptide samples were lyophilized and the purified powder was weighed in 20 mL glass scintillation vials, sealed, and placed in a -20 freezer for storage.

#### Peptide Characterization:

##### *Symmetry Screening via NMR:*

Purified peptide samples were resuspended in 90:10 DI H<sub>2</sub>O:D<sub>2</sub>O to a final peptide concentration of 2 mM and pipetted into 5 mm 300 MHz NMR tubes. A Bruker AV 500 Nuclear Magnetic Resonance (NMR) instrument with a TBI 1H {13C,X}-Z gradient probe was used to screen purified peptides for symmetry.

##### *Structural Characterization via Racemic Crystallography:*

The mirror forms of NMR-confirmed symmetric peptides were synthesized and purified as described in the previous sections. Racemic mixtures of the designed peptides and mirror

peptides were made at concentrations greater than 20 mg/mL total. Commercially available protein crystallization screens, including Molecular Dimensions' Morpheus® screen, Qiagen's JCSG Core Suite screens (I, II, III, and IV) were used to screen the racemic mixtures of peptides for crystal growth. Crystals were grown using either sitting drop or hanging drop diffusion. Crystals were looped and screened in-house, and sent to the Advanced Photon Source (APS) at Argonne National Lab where collaborators in Todd Yeates' lab at the University of California, Los Angeles (UCLA) shot and solved the crystal structures.

**Appendix Table 2.1:** Synthesized peptides that were not published, and were not symmetric *via* NMR.

| Peptide Name | Symmetry       | Sequence of the Asymmetric Unit* |
|--------------|----------------|----------------------------------|
| “Robin”      | C <sub>2</sub> | yXndqr                           |
| “Jane”       | C <sub>3</sub> | DIYr                             |
| “Dawn”       | C <sub>3</sub> | DnLR                             |
| “Noon”       | C <sub>3</sub> | DSRi                             |
| “Heath”      | C <sub>3</sub> | DQRi                             |
| “Beefy”      | C <sub>3</sub> | DyKp                             |
| “Berry”      | C <sub>3</sub> | Dysk                             |
| “Derendil”   | C <sub>3</sub> | tkdIt                            |
| “Vici”       | C <sub>3</sub> | DrPy                             |
| “Bulbasaur”  | C <sub>3</sub> | pEkL                             |
| “Charmander” | C <sub>3</sub> | kDVS                             |
| “Weedle”     | C <sub>3</sub> | EPVK                             |
| “Pidgey”     | C <sub>3</sub> | KPLE                             |
| “Squirtle”   | C <sub>3</sub> | Vdqr                             |
| “Pikachu”    | C <sub>3</sub> | nVkds                            |

|              |                |       |
|--------------|----------------|-------|
| “Cleairy”    | C <sub>3</sub> | plETK |
| “Jigglypuff” | C <sub>3</sub> | kpPYD |
| “Paras”      | C <sub>3</sub> | plskD |
| “Stool”      | C <sub>4</sub> | DnYR  |
| “Zuko”       | C <sub>4</sub> | eWXRH |
| “Azula”      | C <sub>4</sub> | diYqX |
| “Abbot”      | C <sub>4</sub> | EsXSK |
| “Mako”       | C <sub>4</sub> | DQVtX |
| “Momo”       | C <sub>4</sub> | DQWX  |
| “Gyatso”     | C <sub>4</sub> | ELYX  |
| “Chin”       | C <sub>4</sub> | EXnk  |
| “Amon”       | C <sub>4</sub> | DXnq  |
| “Asami”      | C <sub>4</sub> | DXsk  |
| “Lin”        | C <sub>4</sub> | dXQI  |
| “Korra”      | C <sub>4</sub> | DQXaX |
| “Varrick”    | C <sub>4</sub> | DLXaX |
| “Rohan”      | C <sub>4</sub> | DHlaX |
| “BaSingSe”   | C <sub>4</sub> | daTLr |
| “TyLee”      | C <sub>4</sub> | dXTak |
| “Avatar”     | C <sub>4</sub> | EYpsa |
| “Posh”       | C <sub>5</sub> | PDR   |
| “Altair”     | C <sub>5</sub> | fttPp |
| “Faraday”    | C <sub>5</sub> | pLsrE |

\*X = AIB, 2-aminoisobutyric acid



HT “Aqueous” Crystal Screen Chart:**Appendix Table 3.2:** Aqueous, buffer-based conditions tested in the initial HT screen.

|          | Metal 1               |                      |                        |                        |                          |                        | Metal 2               |                      |                        |                        |                          |                        |
|----------|-----------------------|----------------------|------------------------|------------------------|--------------------------|------------------------|-----------------------|----------------------|------------------------|------------------------|--------------------------|------------------------|
|          | 25mM<br>MES<br>pH 5.5 | 25 mM<br>MES<br>pH 6 | 25 mM<br>MES<br>pH 6.5 | 25 mM<br>HEPES<br>pH 7 | 25 mM<br>HEPES<br>pH 7.5 | 25 mM<br>HEPES<br>pH 8 | 25mM<br>MES<br>pH 5.5 | 25 mM<br>MES<br>pH 6 | 25 mM<br>MES<br>pH 6.5 | 25 mM<br>HEPES<br>pH 7 | 25 mM<br>HEPES<br>pH 7.5 | 25 mM<br>HEPES<br>pH 8 |
|          | 1                     | 2                    | 3                      | 4                      | 5                        | 6                      | 7                     | 8                    | 9                      | 10                     | 11                       | 12                     |
| <b>A</b> | -                     | -                    | -                      | -                      | -                        | -                      | -                     | -                    | -                      | -                      | -                        | -                      |
| <b>B</b> | PEG300 2%             | PEG300 2%            | PEG300 2%              | PEG300 2%              | PEG300 2%                | PEG300 2%              | PEG300 2%             | PEG300 2%            | PEG300 2%              | PEG300 2%              | PEG300 2%                | PEG300 2%              |
| <b>C</b> | PEG1000 2%            | PEG1000 2%           | PEG1000 2%             | PEG1000 2%             | PEG1000 2%               | PEG1000 2%             | PEG1000 2%            | PEG1000 2%           | PEG1000 2%             | PEG1000 2%             | PEG1000 2%               | PEG1000 2%             |
| <b>D</b> | PEG2000 2%            | PEG2000 2%           | PEG2000 2%             | PEG2000 2%             | PEG2000 2%               | PEG2000 2%             | PEG2000 2%            | PEG2000 2%           | PEG2000 2%             | PEG2000 2%             | PEG2000 2%               | PEG2000 2%             |
| <b>E</b> | PEG4000 2%            | PEG4000 2%           | PEG4000 2%             | PEG4000 2%             | PEG4000 2%               | PEG4000 2%             | PEG4000 2%            | PEG4000 2%           | PEG4000 2%             | PEG4000 2%             | PEG4000 2%               | PEG4000 2%             |
| <b>F</b> | PEP 2%                | PEP 2%               | PEP 2%                 | PEP 2%                 | PEP 2%                   | PEP 2%                 | PEP 2%                | PEP 2%               | PEP 2%                 | PEP 2%                 | PEP 2%                   | PEP 2%                 |
| <b>G</b> | EtOH 25%              | EtOH 25%             | EtOH 25%               | EtOH 25%               | EtOH 25%                 | EtOH 25%               | EtOH 25%              | EtOH 25%             | EtOH 25%               | EtOH 25%               | EtOH 25%                 | EtOH 25%               |
| <b>H</b> | MeOH 25%              | MeOH 25%             | MeOH 25%               | MeOH 25%               | MeOH 25%                 | MeOH 25%               | MeOH 25%              | MeOH 25%             | MeOH 25%               | MeOH 25%               | MeOH 25%                 | MeOH 25%               |

Crystal Optimization Procedure:

Any conditions resulting in crystal growth in the above crystal screens were regrown in 0.2 mL PCR tubes at 20x scale (100uL). Additional conditions varying solvent ratios, and metal:peptide ratios, and metal salt types were also tested to optimize crystallization conditions. Promising peptide designs were ordered to be synthesized at a larger scale from WuXi AppTec and tested in the lab for further optimization and characterization.

Materials Characterization *via* X-ray Crystallography:

Crystals were looped and screened in-house by Alex Kang and Hannah Nguyen, and shipped to the Advanced Photon Source (APS) at Argonne National Lab where Asim Bera collected data on the crystals and I solved their structures *via* direct methods (Shelx).

**Appendix Table 3.3:** Information about solved crystals.

| Design Name                | “1.3_176_3*”                                                                                            | “London Bridge_Zn”                                                  | “Pugsley_Cu”                                                                                             |
|----------------------------|---------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------|
| Peptide Sequence (ASU)     | C <sub>3</sub> : Dh <sub>m</sub>                                                                        | S <sub>2</sub> : aNkhPe*                                            | S <sub>2</sub> :EppKv*                                                                                   |
| Designed Space Group       | P4 <sub>3</sub> 32 (Cubic)                                                                              | N/A                                                                 | N/A                                                                                                      |
| Crystallized Space Group   | P4 <sub>3</sub> 2 <sub>1</sub> 2 (Tetragonal)                                                           | R3 (Trigonal)                                                       | P-1 (Triclinic)                                                                                          |
| Crystallization Conditions | 0.6 mM peptide<br>0.6 mM Zn(NO <sub>3</sub> ) <sub>2</sub><br>10 mM MES pH 6<br>Heated to 80°C for 24 h | 5 mM peptide<br>5 mM ZnCl <sub>2</sub><br>IPA<br>Room Temp for 24 h | 1.25 mM peptide<br>1.25 mM Cu(NO <sub>3</sub> ) <sub>2</sub><br>1:1 water:DMF<br>Heated to 80°C for 24 h |
| Diffacted to (in Å)        | 0.879                                                                                                   | 0.946                                                               | 0.878                                                                                                    |
| R-value                    | 0.1074                                                                                                  | 0.1419                                                              | 0.1619                                                                                                   |

\* Both of these peptides are S<sub>2</sub> symmetric, so the second half of the peptide’s sequence is a mirror of the first (*e.g.* an l- proline in the first half becomes a d- proline in the second).