De Novo Design of Protein Nanoparticles with Programmable and Tunable Function

Erin Chi Yang

A dissertation

submitted in partial fulfillment of the

requirements for the degree of

Doctor of Philosophy

University of Washington

2023

Reading Committee:

David Baker, Co-chair

Neil P. King, Co-chair

Phil Bradley

Program Authorized to Offer Degree:

Biochemistry

University of Washington

Abstract

De Novo Design of Protein Nanoparticles with Programmable and Tunable Function

Erin Chi Yang

Co-Chairs of the Supervisory Committee:

David Baker, Department of Biochemistry

Neil P. King, Department of Biochemistry

Assembly of designed symmetric protein nanoparticles from multiple copies of one or more distinct protein subunits warrants unique shapes, size diversity, structural regularity, and polyvalency. Current efforts of protein nanoparticles are aimed toward engineering natural and *de novo* nanoparticles for a myriad of applications including multivalent display of target proteins, packaging of macromolecular materials, and scaffolding for molecular imaging techniques (Divine et al. 2020; Bale et al. 2016; Butterfield et al. 2017; McConnell et al. 2020; Liu et al. 2018; Liu, Huynh, and Yeates 2019; Vulovic et al. 2020; Boyoglu-Barnum et al. 2020; Ueda et al. 2020; Marcandalli et al. 2019; Walls et al. 2020). In this thesis, I present the development of an accessible and versatile software application for the arrangement of protein scaffolds into asymmetric and symmetric nanomaterials–including nanoparticles, using this application to design the first three component, non-porous, pH-responsive antibody nanoparticles (O432) as a platform for targeted delivery, and extension of all one- and two-component nanoparticles docked and designed with this application toward developing an undergraduate research curriculum. Additional stories are presented, which describe assays used to explore the O432 nanoparticles as tools for targeted delivery and general design rules that govern interfaces of designed protein nanoparticles.

Table of Contents

Table of Contents **5**
Acknowledgements **10**
Preface **12**
References **14**
List of scientific papers 18
List of late stage scientific papers not included in this thesis 19
Chapter 1: Fast and versatile sequence- independent protein docking for nanomaterials design using RPXDock **20**
Abstract 21
Author Summary 21
Introduction 22
Methods 23
Overview of RPXDock general methodology 23
Figure 1.1: General software structure of RPXDock 24
Inputs and bodies 24
Bounding volume hierarchy (BVH) 25
Defining degrees of freedom (DOFs) 25
Symmetric Architectures 25
Table 1.1: Keywords associated with each currently supported architecture 26
Input preparation 26
Figure 1.2: Example inputs and docking output architectures currently supported by RPXDock 29
Defining the search space 29
Restricting additional DOFs 29
Sampling the search space 30
Global and hierarchical search 31
Figure 1.3: Schematic representation of hierarchical sampling 32
Specifying termini direction and accessibility 33
Evaluating docked configurations 34
Residue-Pair Transform (RPX) Scoring 34
Motif-enriched docking 34
Restricting regions for scoring 35
Score only SS 35
Masking (Allowed residues) 35
Ranking dock quality (score functions) 36
SASA weighted (sasa_priority) score function 36
Figure 1.4: Interface size bias by the sasa_priority score function 37
Other score functions 38
Table 1.2: List of additional score functions 38

Filtering docks....38
Clustering....38
Additional Optional Filters....39
Result....39
Setup and installation....40
Experimental characterization of one- and two-component polyhedral self-assembling proteins from RPXDock....40
Figure 1.5: Docking and characterization of one- and two-component polyhedral assemblies using RPXDock....42
Discussion....43
Supplemental Information....44
Bodies....44
Search....44
Score....45
Filters....46
Supplemental Figures....47
Figure S1.1: Hierarchical docking performance test....47
Figure S1.2: NContact and SASA are highly correlated....48
Figure S1.3: Derivation of the ncontact score term as a function of interface size, SASA....49
Figure S1.4: Empirical derivation of the ncontact score term weight....50
Figure S1.5: nsEM and CryoEM data and associated plots of one- and two-component polyhedral self-assembling proteins from RPXDock....51
Experimental Material and Methods....51
Computational design....51
Protein expression and purification....52
Negative-Stain Electron Microscopy (nsEM)....52
Negative Stain Electron Microscopy image processing....53
CryoEM sample preparation and data collection....53
CryoEM data processing....53
CryoEM model building and validation....54
Table S1.1: All RPXDock command line options....54
Available filter options....60
Table S1.2: SASA estimate filter parameters....60
Table S1.3: SScount filter parameters....60
Table S1.4: Design construct renaming and input pdb files....61
Table S1.5. CryoEM data collection and refinement statistics....61
Supplementary Note 1.1: Protein Sequences....62
References....64
Chapter 2: Computational design of non-porous, pH-responsive antibody nanoparticles... 68
Abstract....69

Introduction....70
Results....70
Computational design....70
Figure 2.1: Design of symmetry-matched plugs to fill empty symmetry axes in protein nanoparticles....72
Structural Characterization....72
Figure 2.2: Mixing independently purified components enables stable, efficient assembly with both Fc and IgG....74
Cryo-EM structural characterization....74
Figure 2.3: Cryo-EM analysis of 72-subunit nanoparticles composed of three distinct structural components....76
Design of O432-17 cargo loading variants....76
Precise engineering of tunable pH-responsiveness in O432-17 nanoparticles....77
Figure 2.4: Plugged antibody nanoparticles electrostatically package cargoes and disassemble in response to acidification....79
Receptor mediated cellular uptake of O432-17 nanoparticles....80
Figure 2.5: Targeted receptor-mediated uptake of O432 nanoparticles....81
Discussion....81
Methods....83
Extension of pH-responsive trimeric building blocks....83
Computational docking of plugged antibody nanoparticles....84
Computational design of plugged antibody nanoparticles....85
Small scale bicistronic bacterial protein expression for trimeric plug and tetrameric designs for sfGFP-Fc lysate assembly....85
Production of Fc and Fc-fusions....85
Large scale expression and purification of O432-17 components....86
In vitro assembly of O432-17 nanoparticles....86
Dynamic light scattering....86
Negative Stain Electron Microscopy preparation and data collection of O432-17 nanoparticles and variants....86
Negative Stain Electron Microscopy data analysis of O432-17 nanoparticles and variants 87
Cryo-Electron Microscopy preparation and data collection of O432-17 nanoparticles....87
Cryo-Electron Microscopy data analysis of O432-17 nanoparticles....87
Computational design of O432-17 electrostatically charged variants....88
Expression and purification of electrostatically charged trimeric plug variants....88
Gel electrophoresis....88
Rational design of O432-17 pH-responsive variants....89
AF647 conjugation to O432-17 trimeric plug variants....89
Flow-cytometry based pH-titration....89
Cells....90
Immunostaining....90

Nanoparticle uptake confocal microscopy (A431 and HeLa EGFR KO Cells).....90
Nanoparticle uptake image acquisition (HeLa WT Cells).....90
Delivery assays.....91
Gal8 visualization of endosomal disruption.....91
Prime Editing Frameshift Assay.....91
Nanoluciferase reconstitution.....92
Delivery assays with functional protein cargo.....93
Production of Diphtheria Toxin variants..... 93
Assembly of O432-17 variants with Diphtheria Toxin variants..... 93
Analysis of cargo packaging with Diphtheria Toxin variants..... 93
Supplementary Figures.....94
Figure S2.1: Example experimental screen of O432 nanoparticles.....96
Figure S2.2: Mixing two of the three individual components does not result in nanoparticle assembly.....96
Figure S2.3: Cryo-EM processing shows newly designed trimeric plug occupying all 3-fold symmetry axes of the nanoparticle octahedral architecture..... 97
Figure S2.4: Comparison of design models of O42.1 and O432-17 with models relaxed into Cryo-EM density..... 98
Figure S2.5: Negative stain electron microscopy of O432-17 nanoparticle assembly in the presence of RNA.....98
Figure S2.6: Flow cytometry analysis of O432-17(-) and O432-17(+) nanoparticle variants.....100
Figure S2.7: Targeted receptor-mediated uptake of O432-17-CTX nanoparticles in WT and EGFR KO HeLa cells after 3 hours.....100
Supplementary Tables.....101
Table 2.1: Cryo-EM data collection and refinement statistics.....101
Table 2.2: Amino acid sequences of assembling O432-17 designs.....101
Table 2.3: Amino acid sequences of hIgG1-Fc, Fc-fusions, and IgGs produced and used to assemble O432-17 designs.....103
Table 2.4: Amino acid sequences of molecular cargoes used by O432-17 designs 104
Table 2.5: Details on EM data acquisition on different O432-17 samples.....104
Table 2.6: Details on EM data processing on different O432-17 samples..... 105
Table 2.7: Statistical information for pH titration experiments..... 105
Table 2.8: Amino acid sequences of Diphtheria-Toxin variants.....106
Data Availability.....107
Code Availability.....107
References..... 108
Chapter 3: Increasing computational protein design literacy through cohort-based learning for undergraduate students.....112
Abstract.....114
Introduction..... 115
Program Overview..... 116

Figure 3.1. Overview of the JUPITER model and program..... 118
Research overview.....118
Project background..... 118
Phase 1 - Introduction to the pSUFER protocol..... 119
Figure 3.2. JUPITER's first phase of research focused on studying individual previously designed successful nanoparticles..... 120
Phase 2 - Experimental design and results..... 120
Figure 3.3. JUPITER's second phase had students create and test their own hypotheses.....122
Table 3.1. Statistical information for comparing position sensitivity by assembly competency.....123
Figure 3.4. JUPITER students further refined their hypotheses and developed new tests.....124
Discussion..... 124
References..... 127
Chapter 4: Using Rosetta scoring metrics to separate between properly assembled and non-assembled de novo designed nanocages.....131
Introduction..... 132
Table 4.1: Distribution of assembled and not assembled nanoparticles by architecture..... 132
Data collection..... 133
Results.....134
Table 4.2: Working cage ddG are generally in a narrower range than non-working cages.....134
Figure 4.1: Optimal Rosetta Metrics differs, depending on symmetry..... 134
Figure 4.2: Distribution of ddG among working and not working nanoparticles by symmetry.....135
Table 4.3: Mean, median, and standard deviation of ddG among working and not working nanoparticles by symmetry.....135
Figure 4.3: Correlation of ddG and SASA (left) and distribution of SASA (right) among working and not working nanoparticles by symmetry.....136
Table 4.4: Mean, median, and standard deviation of SASA among working and not working nanoparticles by symmetry.....136
Figure 4.4: Distribution of shape complementarity among working and not working nanoparticles by symmetry.....137
Table 4.4: Mean, median, and standard deviation of shape complementarity among working and not working nanoparticles by symmetry.....137
Figure 4.5: High residue counts (HPcCount, AroCount, AlaCount, MetCount) are generally found more often in nonworking cages.....138
Data Availability..... 139
References..... 140

Acknowledgements

I feel extremely fortunate to have had the experience that I did over the last five years. I am indebted to a lot of people, but especially to my co-PIs and mentors, David and Neil. Both are incredibly attentive and generous with their time, despite the size of their labs. There were several turning points in my PhD where I've been exceptionally appreciative of the freedom and guidance both David and Neil were able to provide for me. From their unwavering optimism for accepting me as a student–coming in with an economic consulting background–(especially Neil who took me on in my third year), to their kindness by allowing me to switch projects in the middle of my PhD, to their advice for troubleshooting all kinds of code and experimental data. I will never forget David's face when I showed him the first 2D class average of the first three-component plugged antibody nanoparticle on my phone, while we were walking by Pagliacci's as our 1:1 was taking place outside, and that time I threw down 20 pages of paper on the floor in Neil's office with every single SEC and DLS trace I ever took of the plugged nanoparticles, trying to decipher how to get them fully plugged.

I also want to thank my committee: Phil Bradley, Justin Kollman, and Jesse Zalatan, who's nontraditional questions further strengthened my work. Your perspectives have been invaluable at every committee meeting.

To my funding sources who supported not only myself, but also my colleagues and the projects we worked on together: The NSF GRFP fellowship program under the grant number DGE-1762114, NSF Grant CHE-1629214 (N.P.K. and D.B.), Defense Threat Reduction Agency Grants HDTRA1-18-1-0001 and HDTRA1-19-1-0003 (N.P.K. and D.B.), the grant DE-SC0018940 funded by the U.S. Department of Energy, Office of Science (D.B.), National Institutes of Health's National Institute on Aging grants R01AG063845 and 1R01CA240339 (N.P.K. and D.B.), the Bill and Melinda Gates Foundation #INV-010680 (N.P.K. and D.B.), the Audacious Project at the Institute for Protein Design (N.P.K. and D.B.), and the Howard Hughes Medical Institute (D.B.). And ChatGPT as *inspiration* for the original poems throughout this thesis.

To the BPSD program: Chip Asbury, Kelly Lee, Dustin Maly, and of course, Erin Kirschner. This program wouldn't run without any of you. Thank you for putting in the time to provide feedback for my student seminar every year and to Erin K for managing all of the program logistics, along with the million of other responsibilities she has.

The IPD staff: Luki Goldschmidt and Patrick Vecchiato for your instrumental help in making sure things run smoothly in the dry lab, Lauren Carter and Kandise Van Wormer for all of your help in the wet lab, and Lance Stewart, Ian Haydon, Ratika Krishnamurty, and Kristina Herrera for your unconditional support.

I'd also like to thank all members of the Baker and King labs, past and present, for their support, ideas, and friendship for so many years and years to come. To Una Nattermann, Alena Khmelinskaia, Quinton Dowling, Yang Hsia, and Robby Divine, thank you so much for your

unwavering advice in and out of the lab and being such strong role models in science and also great friends.

To my other collaborators including but not limited to Marcos Miranda, Will Sheffler, Jason Zhang, Amijai Saragovi, Mohamad Abedi, Mark Langowski, Marisa Brandys, Christine Kang, Hugh Haddox, Brian Koepnick, Jacob O'Connor, Karla-Luise Herpoldt, Nicolas Goldbach, Zhe Li, Jorge Fallas, George Ueda, James Lazarovitz, Joshua Lubner, Masaharu Somiya, Christian Richardson, Ryan Kibler, Will White, and Scott Boyken. The IPD Cores and Lab Management: Lauren Carter, Luki Goldschmidt, Lance Stewart, Kandise Van Wormer, Andrew Borst, Justin Decarreau, Stacey Gerben, Maggie Ahlrichs, Craig Dobbins, Alexis Hand, Suna Cheng, Mila Lamb, Paul M. Levine, Sidney Chan, and Rebecca Skotheim. To Ashley Vater, Justin Siegel, Dina Listov, and Sarel Fleishman for invaluable discussions and advice throughout the JUPITER program. And lastly, the JUPITER undergrads: Sidney Chan, Elijah Arenas, Zoe Subol, Peter Tinker, Hayden Manninen, Alicia Feichtenbiner, Talal Mustafa, Julia Hallowell, and Isiac Orr. Thanks for keeping the lights on literally and figuratively. My work would not have been possible without your intensive support and hard work.

I want to thank my friends, especially the ones I have made through the BPSD and Biochem program and at the IPD: Audrey Olshefsky, Chelsea Fries, Chloe Adams, Madison Kennedy, Kaitlyn Rutter, Thomas Schlichthärle, Natasha Edman, Xinru Wang, Indrek Kalvet, Naveen Jasti, Anna Lauko, David Juergens, Florian Praetorius, Phil Leung, Nate Bennett, Caci Dishman, Jeremiah Sims, and Michael Goldberg. Thanks for making my time here filled with so many wonderful memories before, during, and after the pandemic. I have been so fortunate to be surrounded by such smart and interesting people.

To Mala Radhakrishnan, Nolan Flynn, and David Haines. The impression you all made on me as an 18-year-old led me to UW and the IPD. Thanks for believing in me and for encouraging me to do this.

Finally, I want to thank my family–my parents, Christine, and my husband, Daniel. Thanks for hearing the worst of it, for reminding me of what else in life is out there, and for your endless support. The four of you are so important to me and in what I achieved here.

Preface

Self-assembling proteins have a strong presence in natural systems, making them a growing field of study in materials science (Pieters et al. 2016). The simple encoding of a few genes in living organisms allows production of hundreds of subunits, which self-assemble into highly specific and functional macromolecular architectures ranging from unbounded 1-dimensional fibers, 2-dimensional arrays, and 3-dimensional crystalline lattices, to bounded cyclic, dihedral, and polyhedral architectures (David S. Goodsell 2019; D. S. Goodsell and Olson 2000). Of these, bounded polyhedral architectures are ideal model systems for repurposing as targeted delivery vehicles because of their highly symmetric geometric features, which enable precise engineering of the interior lumens for compartmentalization, the exteriors for display of moieties to interact with the surrounding environment, and inter-subunit interactions, which define the precise assembly and disassembly processes of each subunit (Khmelinskaia, Wargacki, and King 2021).

Rational engineering of protein containers, viruses, and non-protein formulations have enabled tremendous advancements in developing biomedical drug delivery systems (Mitchell et al. 2020; Czapar and Steinmetz 2017; Waterhouse et al. 2001; Polack et al. 2020; Baden et al. 2021). Despite these advancements, these systems are fundamentally still limited by several clinical barriers which include delivery efficiency and immunogenicity. Overcoming these barriers will require precise engineering of self-assembling protein nanoparticles through *de novo* design and directed evolution such that each protein nanoparticle drug carrier is tailored to satisfy specific therapeutic and functional requirements.

Biomedical drug carriers require the ability to overcome multiple biological barriers, where they must (1) localize from the route of administration to organ of interest (2) transport to target cells, and (3) reach the target organelle and produce the desired therapeutic effect (Poon et al. 2020). One such barrier is the ability for nanoparticles to evade lysosomal degradation. Many polymeric, liposomal, and virus-like nanoparticles have shown significant potential to address this issue (Smith et al. 2019; Banskota et al. 2022; Edwardson, Mori, and Hilvert 2018; Pei and Buyanova 2019; Degors et al. 2019). Here, nanoparticles targeted to the endosome undergo conformational change due to the drop in pH inside the vesicle during the endosomal maturation cycle (Huotari and Helenius 2011), leading to the disruption of the endosomal membrane and evasion of lysosomal degradation (Brock et al. 2019). Inspired by these mechanisms, *de novo* protein design of a protein nanoparticle that could target and enter specific cells and disassemble in response to environmental pH appeared to exhibit potential to establish a novel biological delivery platform that would avoid the high immunogenicity of natural systems. To accomplish this, we required a protein nanoparticle that would be designed to exhibit these features.

The first step in the *de novo* design of protein nanoparticle drug carriers is generating and arranging protein secondary structure into desired configurations. There are three widely used approaches for generating such materials: generation of backbone arrangements using parametric equations or machine learning algorithms (Grigoryan and Degrado 2011; Rhys et al. 2019; Huang et al. 2014); rigid fusion of cyclic protein oligomers with their internal symmetry

axes aligned with those of a desired symmetric architecture (Padilla, Colovos, and Yeates 2001; Lai, King, and Yeates 2012; Laniado, Meador, and Yeates 2021; Hsia et al. 2021; Divine et al. 2021), and sequence-independent rigid body docking of monomers or cyclic oligomers such that their internal symmetry axes are aligned with those of a desired architecture (Fallas et al. 2017; Sahasrabuddhe et al. 2018; King et al. 2014; Hsia et al. 2016; Bale et al. 2016; Shen et al. 2018; Gonen et al. 2015; Ben-Sasson et al. 2021). One challenge these methods face is that the majority of these methods are not well-documented, computationally inefficient, and often siloed as independent applications specific to each architecture as a result of highly specific scoring methods per architecture. Further, incorporating desired functions into protein nanoparticle materials such as targeting nanoparticles to certain cell types, programming pH-induced disassembly, and packaging different molecular payloads, have been nascent design strategies, where no single nanoparticle has been able to perform all three functions at once.

The work I describe in this thesis provides advancements in our understanding of nanomaterial docking and design, and how design enables programmable and tunable function into protein nanoparticles. This information is important for development and refinement of protein nanoparticles for tailored functions, such as efficient targeted delivery vehicles. In one project, I describe the software structure of a fast and versatile sequence-independent computational docking application for generating protein nanomaterials of a variety of symmetric and asymmetric architectures. The manuscript provides a practical guide for its use and describes the variety of available functionalities and tools such as scoring and filtering docks that can be used to guide and refine docking results towards desired configurations.

In a second project, I used this docking application to generate computationally designed non-porous, pH-responsive antibody nanoparticles. I used an existing pH-responsive trimeric helical bundle and two-component antibody nanoparticles as scaffolds, and designed a plug made from this trimeric helical bundle to occupy the apertures of the antibody nanoparticle. The resulting nanoparticle accurately assembles from three independently purified components, each with a distinct function: the trimeric plug enables electrostatic cargo loading and pH-driven disassembly of the nanoparticle and release of loaded cargo from the nanoparticle, the dimeric antibody enables targeting of the nanoparticle to extracellular surface receptors, and the *de novo* tetramer drives the assembly of all three components into an octahedral architecture. I demonstrate tunable pH-drive disassembly of the nanoparticle with a customized flow cytometry based assay, electrostatic cargo loading of both protein and nucleic acid, and targeted cellular entry.

The third project applies our understanding of nanoparticle docking and design to a course designed for undergraduate research. We curated a library of 828 docked and designed one- and two-component nanoparticles of tetrahedral, octahedral, or icosahedral architecture and with a cohort of 9 undergraduate students, analyzed the interfaces of each nanoparticle. Collectively, the students learned how to conduct hypothesis driven computational research. Specifically, the students tested the likelihood that interface mutations would stabilize or destabilize a nanoparticle interface and compared whether trends in these interface mutations were different across all properly assembling and improperly assembling nanoparticles. This

course demonstrates that undergraduates can learn and apply computational protein design to nanomaterials problems within an academic quarter in a cohort-based remote research environment and establishes the importance of analyzing protein-protein interfaces in the context of nanoparticle assemblies.

Two other projects are described here that relate to the study of designing nanoparticles as vehicles for targeted intracellular delivery. One describes the strategies I used to assess for targeted intracellular delivery using the O432 nanoparticle. This undertaking stressed the need for highly sensitive and accurate readout assays involving either the nanoparticle or delivered payload that can be executed in medium to high throughput. The second project involved evaluating docked and designed nanoparticle scaffolds with computational metrics stemming from biophysical principles. This work defined general design principles for the design of protein-protein interfaces in docked and designed nanoparticle assemblies. Specifically, this study found that the boundaries set by computational metrics generally used to filter designs needed to be set differently, depending on target architecture.

Beyond the described work, my research involved several additional projects that will not be described here in detail. I contributed to efforts to design highly geometrically specific nanoparticles to be used for hierarchical self-assembly of 3D crystalline protein materials. Crystal space groups in three dimensions are generated from the combination of crystallographic point groups and Bravais lattices (Albert Cotton 1991). In this work, I helped refine the hierarchical assembly pipeline, inspired by geometric rules for designing symmetry combination materials (Laniado and Yeates 2020). This pipeline first required designed interfaces directing assembly of protein monomers into assemblies with point group symmetry followed by the introduction of a third interface that constrained the translation of these assemblies to generate the desired Bravais lattice. We focused on crystal space groups containing high symmetry point groups (tetrahedral and octahedral) related by dihedral centers, which requires three orthogonal interfaces on a single protein component of varying strengths to maximize assembly cooperativity. I worked on designing five tetrahedral nanoparticles and developed strategies to make all interfaces orthogonal by developing a workflow to implement hydrogen bonding networks at the dihedral centers and implementing methods to restrict sampling to properly orient point group nanoparticles into properly assembling Bravais lattices (Li et al. 2022). I also helped establish geometric requirements to enable design of icosahedra nanoparticles of increasing T number (Dowling et al., *in preparation*, Lee et al., *in preparation*, (Caspar and Klug 1962; Twarock and Luque 2019; Mannige and Brooks 2010)) and linearly extendable protein nanoparticles of point group symmetries using regularized building blocks (Huddy et al., *in review*). Finally, I developed design protocols, assays, and pipelines for characterization of pH-responsiveness in designed proteins, which can allow for unequivocal determination of conformational changes *in vitro* due to changes in environmental pH.

References

Albert Cotton, F. 1991. *Chemical Applications of Group Theory*. John Wiley & Sons.

Baden, Lindsey R., Hana M. El Sahly, Brandon Essink, Karen Kotloff, Sharon Frey, Rick Novak, David Diemert, et al. 2021. “Efficacy and Safety of the mRNA-1273 SARS-CoV-2 Vaccine.” *The New England Journal of Medicine* 384 (5): 403–16.

Bale, Jacob B., Shane Gonen, Yuxi Liu, William Sheffler, Daniel Ellis, Chantz Thomas, Duilio Cascio, et al. 2016. “Accurate Design of Megadalton-Scale Two-Component Icosahedral Protein Complexes.” *Science* 353 (6297): 389–94.

Banskota, Samagya, Aditya Raguram, Susie Suh, Samuel W. Du, Jessie R. Davis, Elliot H. Choi, Xiao Wang, et al. 2022. “Engineered Virus-like Particles for Efficient in Vivo Delivery of Therapeutic Proteins.” *Cell* 185 (2): 250–65.e16.

Ben-Sasson, Ariel J., Joseph L. Watson, William Sheffler, Matthew Camp Johnson, Alice Bittleston, Logeshwaran Somasundaram, Justin Decarreau, et al. 2021. “Design of Biologically Active Binary Protein 2D Materials.” *Nature* 589 (7842): 468–73.

Brock, Dakota J., Helena M. Kondow-McConaghy, Elizabeth C. Hager, and Jean-Philippe Pellois. 2019. “Endosomal Escape and Cytosolic Penetration of Macromolecules Mediated by Synthetic Delivery Agents.” *Bioconjugate Chemistry* 30 (2): 293–304.

Caspar, D. L., and A. Klug. 1962. “Physical Principles in the Construction of Regular Viruses.” *Cold Spring Harbor Symposia on Quantitative Biology* 27: 1–24.

Czapar, Anna E., and Nicole F. Steinmetz. 2017. “Plant Viruses and Bacteriophages for Drug Delivery in Medicine and Biotechnology.” *Current Opinion in Chemical Biology* 38 (June): 108–16.

Degors, Isabelle M. S., Cuifeng Wang, Zia Ur Rehman, and Inge S. Zuhorn. 2019. “Carriers Break Barriers in Drug Delivery: Endocytosis and Endosomal Escape of Gene Delivery Vectors.” *Accounts of Chemical Research* 52 (7): 1750–60.

Divine, Robby, Ha V. Dang, George Ueda, Jorge A. Fallas, Ivan Vulovic, William Sheffler, Shally Saini, et al. 2021. “Designed Proteins Assemble Antibodies into Modular Nanocages.” *Science* 372 (6537). https://doi.org/10.1126/science.abd9994.

Edwardson, Thomas G. W., Takahiro Mori, and Donald Hilvert. 2018. “Rational Engineering of a Designed Protein Cage for siRNA Delivery.” *Journal of the American Chemical Society* 140 (33): 10439–42.

Fallas, Jorge A., George Ueda, William Sheffler, Vanessa Nguyen, Dan E. McNamara, Banumathi Sankaran, Jose Henrique Pereira, et al. 2017. “Computational Design of Self-Assembling Cyclic Protein Homo-Oligomers.” *Nature Chemistry* 9 (4): 353–60.

Gonen, Shane, Frank DiMaio, Tamir Gonen, and David Baker. 2015. “Design of Ordered Two-Dimensional Arrays Mediated by Noncovalent Protein-Protein Interfaces.” *Science* 348 (6241): 1365–68.

Goodsell, David S. 2019. “Symmetry at the Cellular Mesoscale.” *Symmetry* 11 (9): 1170.

Goodsell, D. S., and A. J. Olson. 2000. “Structural Symmetry and Protein Function.” *Annual Review of Biophysics and Biomolecular Structure* 29: 105–53.

Grigoryan, Gevorg, and William F. Degrado. 2011. “Probing Designability via a Generalized Model of Helical Bundle Geometry.” *Journal of Molecular Biology* 405 (4): 1079–1100.

Hsia, Yang, Jacob B. Bale, Shane Gonen, Dan Shi, William Sheffler, Kimberly K. Fong, Una Nattermann, et al. 2016. “Design of a Hyperstable 60-Subunit Protein Icosahedron.” *Nature*. https://doi.org/10.1038/nature18010.

Hsia, Yang, Rubul Mout, William Sheffler, Natasha I. Edman, Ivan Vulovic, Young-Jun Park, Rachel L. Redler, et al. 2021. “Design of Multi-Scale Protein Complexes by Hierarchical Building Block Fusion.” *Nature Communications* 12 (1): 2294.

Huang, Po-Ssu, Gustav Oberdorfer, Chunfu Xu, Xue Y. Pei, Brent L. Nannenga, Joseph M.

Rogers, Frank DiMaio, Tamir Gonen, Ben Luisi, and David Baker. 2014. “High Thermodynamic Stability of Parametrically Designed Helical Bundles.” *Science* 346 (6208): 481–85.
Huotari, Jatta, and Ari Helenius. 2011. “Endosome Maturation.” *The EMBO Journal* 30 (17): 3481–3500.
Khmelinskaia, Alena, Adam Wargacki, and Neil P. King. 2021. “Structure-Based Design of Novel Polyhedral Protein Nanomaterials.” *Current Opinion in Microbiology* 61 (June): 51–57.
King, Neil P., Jacob B. Bale, William Sheffler, Dan E. McNamara, Shane Gonen, Tamir Gonen, Todd O. Yeates, and David Baker. 2014. “Accurate Design of Co-Assembling Multi-Component Protein Nanomaterials.” *Nature* 510 (7503): 103–8.
Lai, Yen-Ting, Neil P. King, and Todd O. Yeates. 2012. “Principles for Designing Ordered Protein Assemblies.” *Trends in Cell Biology* 22 (12): 653–61.
Laniado, Joshua, Kyle Meador, and Todd O. Yeates. 2021. “A Fragment-Based Protein Interface Design Algorithm for Symmetric Assemblies.” *Protein Engineering, Design & Selection: PEDS* 34: 1–34.
Laniado, Joshua, and Todd O. Yeates. 2020. “A Complete Rule Set for Designing Symmetry Combination Materials from Protein Molecules.” *Proceedings of the National Academy of Sciences of the United States of America* 117 (50): 31817–23.
Li, Zhe, Shunzhi Wang, Una Nattermann, Asim K. Bera, Andrew J. Borst, Matthew J. Bick, Erin C. Yang, et al. 2022. “Accurate Computational Design of 3D Protein Crystals.” *bioRxiv*. https://doi.org/10.1101/2022.11.18.517014.
Mannige, Ranjan V., and Charles L. Brooks 3rd. 2010. “Periodic Table of Virus Capsids: Implications for Natural Selection and Design.” *PloS One* 5 (3): e9423.
Mitchell, Michael J., Margaret M. Billingsley, Rebecca M. Haley, Marissa E. Wechsler, Nicholas A. Peppas, and Robert Langer. 2020. “Engineering Precision Nanoparticles for Drug Delivery.” *Nature Reviews. Drug Discovery*. https://doi.org/10.1038/s41573-020-0090-8.
Padilla, J. E., C. Colovos, and T. O. Yeates. 2001. “Nanohedra: Using Symmetry to Design Self Assembling Protein Cages, Layers, Crystals, and Filaments.” *Proceedings of the National Academy of Sciences of the United States of America* 98 (5): 2217–21.
Pei, Dehua, and Marina Buyanova. 2019. “Overcoming Endosomal Entrapment in Drug Delivery.” *Bioconjugate Chemistry* 30 (2): 273–83.
Pieters, Bas J. G. E., Mark B. van Eldijk, Roeland J. M. Nolte, and Jasmin Mecinović. 2016. “Natural Supramolecular Protein Assemblies.” *Chemical Society Reviews* 45 (1): 24–39.
Polack, Fernando P., Stephen J. Thomas, Nicholas Kitchin, Judith Absalon, Alejandra Gurtman, Stephen Lockhart, John L. Perez, et al. 2020. “Safety and Efficacy of the BNT162b2 mRNA Covid-19 Vaccine.” *The New England Journal of Medicine* 383 (27): 2603–15.
Poon, Wilson, Benjamin R. Kingston, Ben Ouyang, Wayne Ngo, and Warren C. W. Chan. 2020. “A Framework for Designing Delivery Systems.” *Nature Nanotechnology* 15 (10): 819–29.
Rhys, Guto G., Christopher W. Wood, Joseph L. Beesley, Nathan R. Zaccai, Antony J. Burton, R. Leo Brady, Andrew R. Thomson, and Derek N. Woolfson. 2019. “Navigating the Structural Landscape of De Novo α-Helical Bundles.” *Journal of the American Chemical Society* 141 (22): 8787–97.
Sahasrabuddhe, Aniruddha, Yang Hsia, Florian Busch, William Sheffler, Neil P. King, David Baker, and Vicki H. Wysocki. 2018. “Confirmation of Intersubunit Connectivity and Topology of Designed Protein Complexes by Native MS.” *Proceedings of the National Academy of Sciences of the United States of America* 115 (6): 1268–73.
Shen, Hao, Jorge A. Fallas, Eric Lynch, William Sheffler, Bradley Parry, Nicholas Jannetty, Justin Decarreau, et al. 2018. “De Novo Design of Self-Assembling Helical Protein Filaments.” *Science* 362 (6415): 705–9.
Smith, Samuel A., Laura I. Selby, Angus P. R. Johnston, and Georgina K. Such. 2019. “The Endosomal Escape of Nanoparticles: Toward More Efficient Cellular Delivery.”

Bioconjugate Chemistry 30 (2): 263–72.
Twarock, Reidun, and Antoni Luque. 2019. “Structural Puzzles in Virology Solved with an Overarching Icosahedral Design Principle.” *Nature Communications* 10 (1): 4414.
Waterhouse, D. N., P. G. Tardi, L. D. Mayer, and M. B. Bally. 2001. “A Comparison of Liposomal Formulations of Doxorubicin with Drug Administered in Free Form: Changing Toxicity Profiles.” *Drug Safety: An International Journal of Medical Toxicology and Drug Experience* 24 (12): 903–20.

List of scientific papers

1. **Yang EC**, Divine R, Miranda M, Borst AJ, Sheffler W, Zhang JZ, Choi H, Decarreau J, Goldbach N, Chan S, Alhrichs M, Li X, Dobbins C, Lubner J, Hendel SJ, Somiya M, Khmelinskaia A, King NP, Baker D. (*submitted*) Computational design of non-porous, pH-responsive antibody nanoparticles.
2. Sheffler W*, **Yang EC***, Dowling Q*, Hsia Y*, Fries CN, Stanislaw J, Langowski M, Brandys M, Khmelinskaia A, King NP, Baker D (2022) Fast and versatile sequence-independent protein docking for nanomaterials design using RPXDock. *bioRxiv* 2022.10.25.513641.
3. **Yang EC***, Divine R*, Kang CS, Chan S, Arenas E, Subol Z, Tinker P, Manninen H, Feichtenbiner A, Mustafa T, Hallowell J, Orr I, Haddox H, Koepnick B, O'Connor J, Haydon IC, Herpoldt K, Van Wormer K, Abell C, Baker D, Khmelinskaia A, King NP (2022) Increasing Computational Protein Design Literacy through Cohort-Based Learning for Undergraduate Students. *Journal of Chemical Education* **99 (9)**, 3177-3186. [Cover]

List of late stage scientific papers not included in this thesis

1. Li Z*, Wang S*, Nattermann U*, Bera AK, Borst AJ, Bick MJ, **Yang EC**, Sheffler W, Lee B, Seifert S, Nguyen H, Kang A, Dalal R, Lubner JM, Hsia Y, Haddox H, Courbet A, Dowling Q, Miranda M, Favor A, Etemadi A, Edman NI, Yang W, Sankaran B, Negahdari B, Baker D (2022) Accurate Computational Design of 3D Protein Crystals. *bioRxiv* 2022.11.18.517014
2. Dowling Q, Park YJ, Gerstenmaier N, **Yang EC**, Wargacki A, Hsia Y, Fries CN, Ravichandran R, Walkey C, Burrell A, Veesler D, Baker D, King NP. (*submitted*) Hierarchical design of pseudosymmetric protein nanoparticles.
3. Huddy TF*, Hsia Y, Xu J, Kibler R, Bethel N, Nagarajan D, Redler R, Leung P, Borst AJ, **Yang EC**, Bera AK, Kang A, Coudray N, Calise SJ, Han HL, Li Z, McHugh R, Coventry B, Brunette T, Liu Y, Dauparas J, Kollman JM, Bhabha G, Ekiert D, Baker D. (*submitted*) Geometrically programmable nanomaterial construction using regularized protein building blocks.
4. Goldbach N, Yousif I, Wicky B, Croft J, Li X, Carter L, **Yang EC**, Lee K, Baker D., (*in preparation*) Rational de novo design of helical bundles for pH-controlled membrane lysis.

Chapter 1: Fast and versatile sequence-independent protein docking for nanomaterials design using RPXDock

Adapted from:
Sheffler, William, Erin C. Yang, Quinton Dowling, Yang Hsia, Chelsea N. Fries, Jenna Stanislaw, Mark Langowski, et al. 2022. “Fast and Versatile Sequence-Independent Protein Docking for Nanomaterials Design Using RPXDock.” *bioRxiv*. https://doi.org/10.1101/2022.10.25.513641.

Protein assemblies in symmetric architectures, we all admire,
A central technique for docking these scaffolds is what we desire.
But existing docking methods are hard to modify,
Tailored for specific architectures, they make us cry.

Enter RPXDock, fast and flexible,
A software package made to be extensible.
Residue-pair transform scoring and hierarchical search,
Guiding docking results toward the desired perch.

Generality and efficiency is ingrained at its core,
Due to this, new applications are easier to explore.
Protein nanomaterials are now a reality,
Due to RPXDock, driving the new age of practicality.

Abstract

Computationally designed multi-subunit assemblies have shown considerable promise for a variety of applications, including a new generation of potent vaccines. One of the major routes to such materials is rigid body sequence-independent docking of cyclic oligomers into architectures with point group or lattice symmetries. Current methods for docking and designing such assemblies are tailored to specific classes of symmetry and are difficult to modify for novel applications. Here we describe RPXDock, a fast, flexible, and modular software package for sequence-independent rigid-body protein docking across a wide range of symmetric architectures that is easily customizable for further development. RPXDock uses an efficient hierarchical search and a residue-pair transform (*RPX*) scoring method to rapidly search through multidimensional docking space. We describe the structure of the software, provide practical guidelines for its use, and describe the available functionalities including a variety of score functions and filtering tools that can be used to guide and refine docking results towards desired configurations.

Author Summary

Protein design methodologies are now able to generate, through a stepwise approach, a wide variety of self-assembling protein structures that begin to rival the structural complexity of naturally occurring protein nanomachines. Efficient methods for docking oligomeric protein building blocks in user-defined target symmetries are central to these techniques. We developed RPXDock as a fast and versatile method to systematically dock pre-existing proteins together into a multitude of asymmetrical and symmetrical architectures. RPXdock is also readily extendable to future applications through the addition of new symmetries, score functions, and filtering criteria.

Introduction

There has been considerable progress in the design of symmetric protein assemblies ranging from relatively small, cyclically symmetric proteins, to megadalton structures containing more than 100 subunits (Ueda et al. 2020; King et al. 2012; Hsia et al. 2016; Bale et al. 2016; Woolfson 2005; Laniado, Meador, and Yeates 2021; Lai, King, and Yeates 2012; Padilla, Colovos, and Yeates 2001; Golub et al. 2020; Kakkis et al. 2020; Lin et al. 2017). There are three widely used approaches for generating such materials: generation of backbone arrangements using parametric equations (primarily applied to helical bundles with cyclic symmetries such as coiled coils) (Grigoryan and Degrado 2011; Rhys et al. 2019; Huang et al. 2014); rigid fusion of cyclic protein oligomers with their internal symmetry axes aligned with those of a desired symmetric architecture (Padilla, Colovos, and Yeates 2001; Lai, King, and Yeates 2012; Laniado, Meador, and Yeates 2021; Hsia et al. 2021; Divine et al. 2021), and sequence-independent rigid body docking of cyclic oligomers such that their internal symmetry axes are aligned with those of a desired architecture followed by combinatorial sequence optimization at the newly generated protein-protein interface to drive assembly (Fallas et al. 2017; Sahasrabuddhe et al. 2018; King et al. 2014; Hsia et al. 2016; Bale et al. 2016; Shen et al. 2018; Gonen et al. 2015; Ben-Sasson et al. 2021). The third approach has the advantage of considerable generality since cyclic building blocks can be combined in a very wide variety of docked arrangements independent of the constraint of chain fusion accessibility. However, while many sequence-dependent docking methods exist for protein-protein interaction prediction (Yan, Tao, and Huang 2018; Schneidman-Duhovny et al. 2005; Park et al. 2019; Lyskov and Gray 2008; Chauhan and Pantazes 2022), software for sequence-independent docking for protein design remains relatively underdeveloped. One challenge such methods face is that in the absence of sequence information, scoring of different docked arrangements is not straightforward. Fast Fourier Transform (FFT) docking methods can be used without sequence information for design applications, but the interatomic interactions are blurred out, and the results are generally not rotationally invariant (Padhorny et al. 2016). The “slide-into-contact” `tc_dock` method (King et al. 2014) and derivatives thereof, which use a residue-pair transform (*RPX*) hashing method to approximate residue-residue interaction energies prior to explicit sequence design (Fallas et al. 2017), have proven useful in the design of a wide variety of symmetric protein nanomaterials including cyclic homooligomers (Fallas et al. 2017), dihedral assemblies (Sahasrabuddhe et al. 2018), multi-component symmetric protein nanocages (King et al. 2014; Hsia et al. 2016; Bale et al. 2016; King et al. 2012; Ueda et al. 2020), one-dimensional fibers (Shen et al. 2018), two-dimensional layers (Gonen et al. 2015; Ben-Sasson et al. 2021), and three-dimensional crystals (Li et al. submitted). However, these methods have not been thoroughly documented, are computationally inefficient, and are difficult to modify for new applications.

We set out to develop a computationally efficient and readily customizable method for rigid body sequence-independent docking capable of pruning unproductive regions of the available search space to reduce time spent in computationally expensive downstream sequence design calculations. Here we describe the RPXDock software, which improves on the earlier `tc_dock` software in three major areas:

1. *Generalizability*: RPXDock unifies previous docking methods specific to particular architectures under a single framework that globally searches rigid body space, sampling the relevant rigid body degrees of freedom (DOFs) across multiple classes of symmetric and asymmetric architectures.
2. *Extensibility*: All the computationally expensive operations in RPXDock are written in C++ that the user interfaces via python. The lower-level libraries are interoperable and thoroughly covered by tests. The codebase is structured to encourage development of new user-defined constraints such that the top outputs are the highest quality docks that satisfy a given set of criteria. For example, newly implemented features allow biasing of the results towards particular interface sizes and protein termini geometry. Adding new docking architectures, score functions, or filters requires minimal updates to existing code.
3. *Speed*: RPXDock utilizes hierarchical decomposition of the underlying degrees of freedom paired with a matching hierarchy of *RPX* score functions to rapidly scan a full docking space at lower resolution; discard large, low-quality regions of the space; and refine docks in progressively higher-quality regions. As a result, RPXDock is very fast and computationally inexpensive, capable of explicitly evaluating millions of docked configurations in minutes. A typical docking trajectory involving two building blocks finishes in seconds to minutes, including overhead.

Prior to publication of this manuscript, RPXDock was used to successfully design cyclic oligomers (Gerben et al. 2023), one-component nanocages (Wang et al. 2022), two-component nanocages (Li et al., *submitted*; Huddy et al., *submitted*; Dosey et al., *in preparation*), and even larger pseudo-symmetric nanomaterials (Dowling et al., *in preparation*; Lee et al., *in preparation*), establishing its utility and generality. Here we provide a guide to using RPXDock to produce rigid body docks, prior to sequence design (Leaver-Fay et al. 2011; Dauparas et al. 2022). Additional technical descriptions of individual modules in the software are provided in the supplementary materials.

Methods

Overview of RPXDock general methodology

A visual outline of the software structure is provided in **Figure 1.1**. Users pass options into the `dock.py` application, which include required inputs such as Protein Data Bank (.pdb) files and the desired docking architecture, as well as other optional docking parameters described in detail in subsequent sections. A full list of command line options can be found in **Table S1.1** and can be retrieved interactively using `--help`. The `dock.py` application interprets user-defined options and drives the machinery behind the docking algorithm. Input .pdb files are loaded using PyRosetta (Chaudhury, Lyskov, and Gray 2010) as poses, then converted by the *Body* class into body objects. Various structural data are compiled from the input .pdb files, including transformable Bounding Volume Hierarchies (BVH) that index atomic coordinates. The *Spec* and *Sampler* classes define the DOFs of the target architecture and how they are to be broken down hierarchically. This space is traversed in the *Search* class, using a hierarchical search

algorithm similar to branch-and-bound search (Larsson and Akenine-Möller 2006). During each iteration of the hierarchical search, each docked configuration, or transform, is evaluated by a residue-pair motif score (Fallas et al. 2017) matched to the resolution of the search step, and then by a user-selected score function. Residue-pair motifs are identified by interacting pairs of backbone positions determined via the BVH data structures. Once the hierarchical search algorithm reaches its final resolution, the remaining docked configurations can be filtered with optional user-defined metrics. The filtered docked configurations are clustered based on redundancy among docked transforms and stored by the *Result* class as transformation matrices and scores in an xarray dataframe. The *Result* class can subsequently be used to re-apply a transformation matrix to the stored input pose, yielding a full-atom .pdb file.

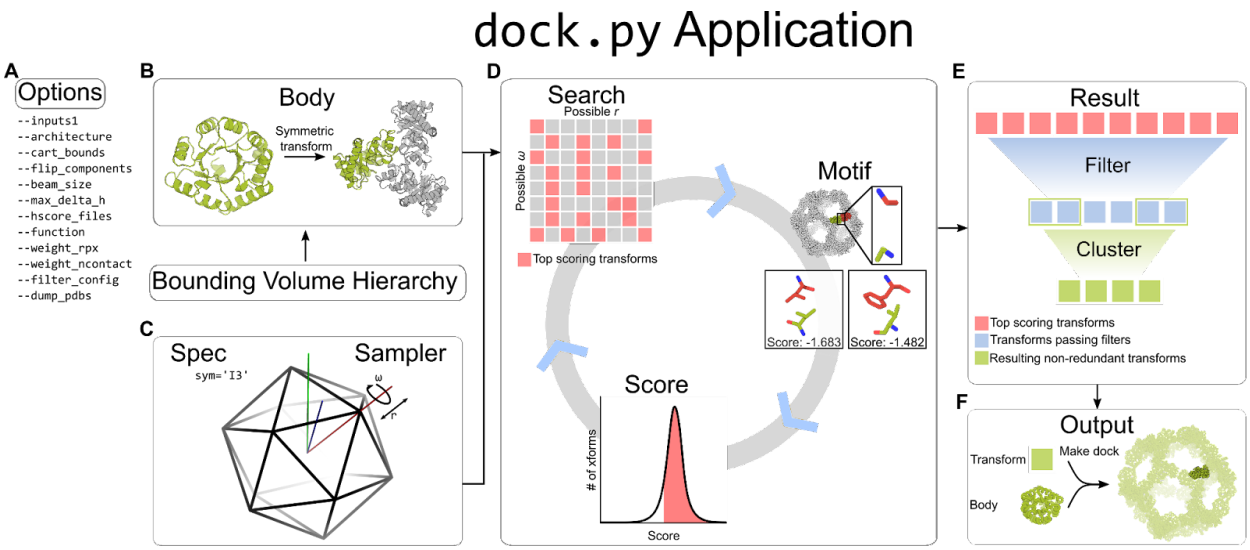


Figure 1.1: General software structure of RPXDock.
A. User-defined inputs are given as options to the `dock.py` application. **B.** Within the application, input .pdb files are stored in the Body object as a PyRosetta pose. The *Body* class implements a Bounding Volume Hierarchy (BVH) for rapid operations on coordinates. **C.** The *Spec* and *Sampler* classes define the rigid-body DOFs the Body object is allowed to sample. **D.** Within the *Search* class, the Body object receives the DOFs as rigid body transforms (indicated as grid squares). Each transform is evaluated by the *Motif* and *Score* classes, which ranks the quality of residue-pair motifs at a given interface of a dock (Fallas et al. 2017) and subsequently summarizes the residue-pair motif scores with additional interface quality metrics through a user-selected score function. The top scoring transforms are searched iteratively with higher resolution sampling and scoring in a hierarchical search algorithm. **E.** The final top scoring transforms from the search are fed into the *Result* class, which prunes the results using filter metrics and clusters the transforms based on backbone redundancy. **F.** The results are stored and output as transforms, which can be re-applied to the input Body object to generate a full-atom .pdb file of the resulting docked configuration.

Inputs and bodies

RPXDock uses the PyRosetta (Chaudhury, Lyskov, and Gray 2010) *pose* module to load the atomic coordinates of input .pdb files and make secondary structure assignments via Define Secondary Structure of Proteins (DSSP) (Touw et al. 2015; Kabsch and Sander 1983). The PyRosetta pose is stored in the *Body* class as a Body object. Input .pdb files are provided to the

`dock.py` application using the `--inputs1` option. The input can be a path to a single .pdb (e.g., example.pdb) file, or a path with a wildcard (e.g., /path/to/files/*.pdb) can be supplied for multiple inputs. For multicomponent docking, additional inputs can be provided using the `--inputs2` and `--inputs3` option as necessary. For trajectories with multiple input lists provided to `--inputs[n]`, each object in the list will be sampled against every other object in a partner list. The results for list inputs are batched and ranked together against one another. Thus, the "top" results may not include representatives from every input .pdb. If results from every input are desired, the user can either analyze the entire output list or execute each input or pair of inputs in separate RPXDock trajectories.

Bounding volume hierarchy (BVH)

The *Body* class implements a Bounding Volume Hierarchy (BVH) representation for efficient contact, sliding, clash checking, and determination of contacts for scoring (Gu et al. 2013). As time taken for these operations scales with interface size, valuable compute time is saved by our implementation of BVH, which utilizes spheres rather than traditional bounding boxes for rotational invariance, allowing rigid body motions without recalculation. The BVH is first used to check for contacts, rapidly discarding configurations where bodies do not interact. During docking, clashing docks where the BVH intersect are removed. Lastly, BVH identifies all interacting pairs of residue stubs during scoring so that only interacting residues are evaluated. These operations are adjusted conservatively based on the resolution of the sampling, such that even large regions of the search space can be discarded as unlikely to contain favorable configurations.

Defining degrees of freedom (DOFs)

Sampling configurations of bodies is performed through a composable set of primitive samplers, including 1D, 2D, and 3D cartesian grids, 1D rotations, 2D directions, and 3D orientations. The space of orientations is modeled as the equivalent space of quaternions on a 3-sphere, and sampling is performed by subdividing the cells of a bitruncated 24-cell, a uniform 4D polytope that divides the 3-sphere uniformly into roughly cubic regions. This approach avoids the pitfalls of using Euler angles to represent 3D rotations. Streamlined combinations of these samplers are provided, such as rotation and translation on a symmetry axis, or a full 6D rigid body transformation, as well as a simple framework to create user-defined compositions and products of sampling spaces. All of these samplers and their combinations provide configurable resolutions, bounds, a hierarchy of nested sampling grids, and the ability to map indices between higher and lower resolutions.

Symmetric Architectures

In symmetrical systems, the "architecture" defines the connectivity and allowed rigid-body kinematics, or movements, of the building blocks. RPXDock currently has built-in support for asymmetric, cyclic, stacking, dihedral, wallpaper (2D), and polyhedral group architectures. While the current release of `dock.py` does not support helical (1D) and crystal (3D) architectures, the

components necessary for these protocols are available, and we plan to implement these in future builds of RPXDock. The desired architecture is specified per trajectory with the `--architecture` option using a keyword (**Table 1**).

Table 1.1: Keywords associated with each currently supported architecture.

--architecture	**Number of unique protein components supported**	**Keyword(s)**
Asymmetric	2	`"ASYM"`
Cyclic	1	`"C[n]"` where [n] = 1, 2, 3, …, n
Stacking	2	`"AXLE_[n]"` where [n] = 1, 2, 3, …, n, or `"AXLE_1_[m]_[n]"` where [m] and [n] correspond to the cyclic symmetries of the inputs and [n] != [m]. Currently supports up to [m] = 5 and [n] = 6.
Dihedral	1	`"DX_X"`, where X is the cyclic symmetry perpendicular to the dihedral plane and the oligomeric state of the input scaffold
	1	`"DX_2"`, same as above, but the input oligomer is a dimer aligned to the dihedral plane
Polyhedral group	1	`"T2", "T3", "O2", "O3", "O4", "I2", "I3", "I5"`
	2	`"T32", "T33", "O32", "O42", "O43", "I32", "I52", "I53"`
	3	`"T332", "O432", "I532"`
Wallpaper	2, 3	`P6_632, P6_63, P6_62, P6_32, P3_33, P4_42, P4_44` where `"Px"` describes the lattice symmetry and cyclic oligomer symmetries are listed after the underscore

Input preparation

To dock two distinct monomers asymmetrically or to form cyclic oligomers, monomeric building blocks should have their center of mass at [0,0,0] (**Figure 1.2A-B**). RPXDock will not center the inputs by default, but the `--recenter_input` option can be passed to translate a monomeric building block such that its center of mass is at [0,0,0]. The final transform values reported are

relative to the recentered pose, so it is recommended that inputs are pre-centered if the user plans to use these values.

To form dihedral, stacking, wallpaper, and polyhedral group symmetries such as tetrahedral, octahedral, and icosahedral architectures, the input building blocks must be cyclic oligomers. The input .pdb files must be pre-aligned such that their internal rotational symmetry axes are aligned to the Z axis and the center of mass of the oligomer should be centered at [0,0,0] (**Figure 1.2C-D**). It is important to note that the input .pdb files should only contain the asymmetric unit (asu) of the cyclic oligomer rather than the full symmetric building block, as RPXDock will generate the symmetry-related chains. Currently, dihedral docking only supports one-component (i.e., homomeric) architectures; stacking supports two-component architectures; polyhedral group docking supports one-, two-, and three-component architectures; and wallpaper docking supports two- and three-component architectures.

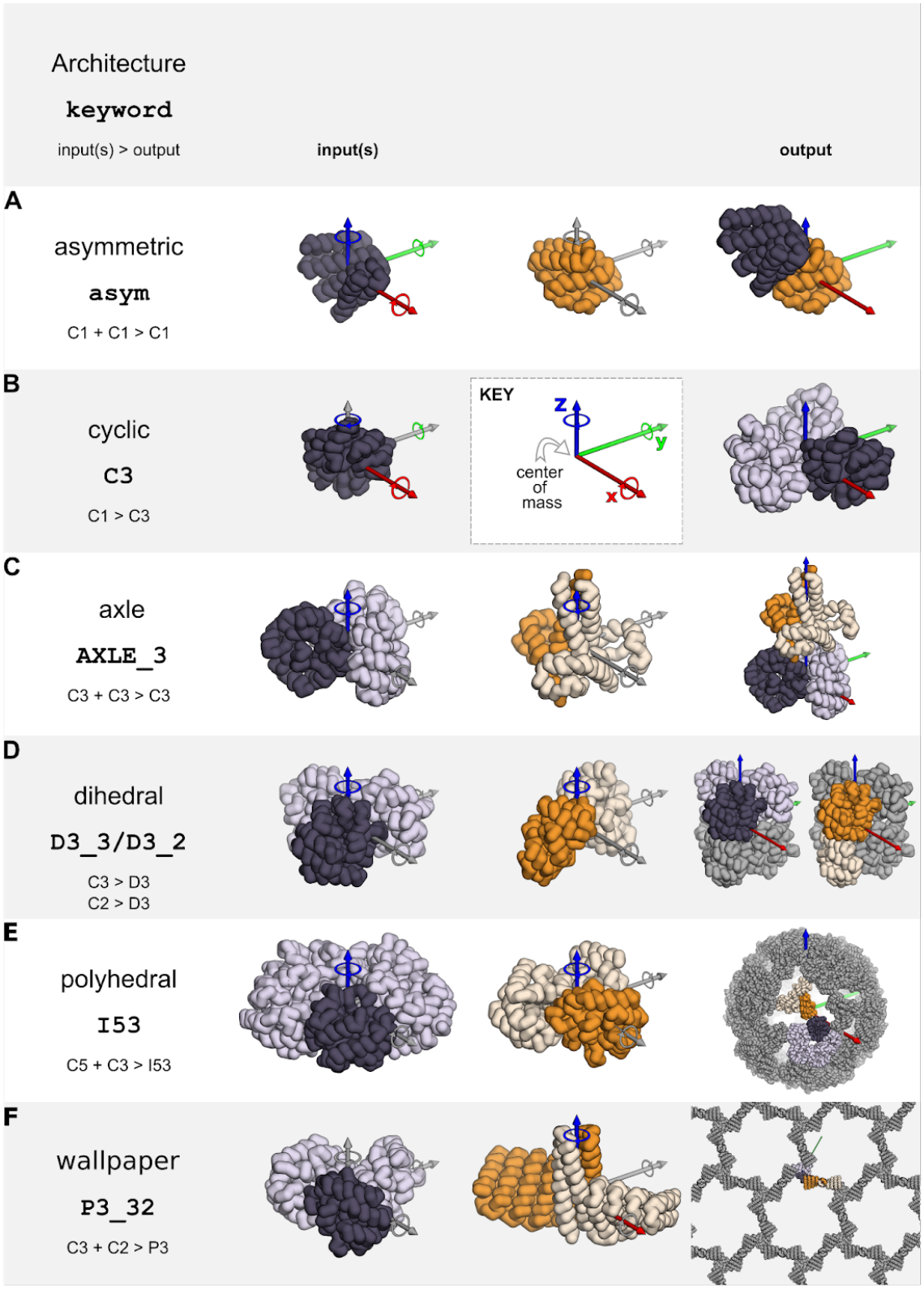

Architecture
keyword
input(s) > output
input(s)
output
A
asymmetric
asym
C1 + C1 > C1
B
cyclic
C3
C1 > C3
KEY
z
y
x
center
of
mass
C
axle
AXLE_3
C3 + C3 > C3
D
dihedral
D3_3/D3_2
C3 > D3
C2 > D3
E
polyhedral
I53
C5 + C3 > I53
F
wallpaper
P3_32
C3 + C2 > P3

Figure 1.2: Example inputs and docking output architectures currently supported by RPXDock. X/Y/Z cartesian axes are shown in red, green, and blue respectively. Corresponding translational and rotational DOFs are sampled along and around these axes. Axes where DOFs are not sampled for an architecture are colored gray. **A.** Asymmetric docking samples 6 DOFs belonging to the first of two input monomers. **B.** Cyclic docking samples four DOFs belonging to an input monomer to generate a cyclic structure with its cyclic axis aligned to the Z axis. **C-F.** Oligomeric input structures must have their cyclic axis aligned to the Z axis and the input .pdb should only contain the asu (dark). Stacking, dihedral, polyhedral group, and wallpaper docking samples the rotational and translational DOF along the Z axis of the input cyclic oligomer, which is aligned during docking to the relevant rotational symmetry axes in the target architecture.

Defining the search space

The search spaces for the supported architectures in RPXDock are either one-, two-, or three-body problems and the number of allowed DOFs sampled depends on the kinematics defined by the specified architecture. Two-body asymmetric docking technically allows all three rotational DOFs and all three translational DOFs (X, Y, and Z) per component, but in practice it is sufficient to hold one component static while sampling the other component against it (**Figure 1.2A**). Cyclic docking allows sampling of all three rotational DOFs but only one translational DOF (the radius), as sampling the remaining two cartesian DOFs results in identical final structures (**Figure 1.2B**). Each oligomeric component in stacking, dihedral, polyhedral group, wallpaper, and crystal architectures is aligned to a single rotational symmetry axis in the target architecture (the Z axis in the input .pdb) and is therefore limited to sampling one rotational and one translational DOF along that axis (**Figure 1.2C-F**).

Each translational or rotational DOF is set by bounds in cartesian or angular space. Cartesian bounds can be set by `--cart_bounds d1 d2` where the lower (d1) and upper (d2) bounds are distances in Ångstroms. The default values of `d1` and `d2` for symmetrical architectures are `0` and `500`, limiting the search to only the positive direction of the space, as the reverse translational degrees of freedom are redundant when combined with the `--flip_components` option (see below). For asymmetrical docking scenarios, however, the default values are `-500` and `500`, allowing search in both directions. The larger this range is set, the longer the runtime and memory required. Thus, if the user has an idea of the desired search size, these values should be reduced as appropriate. Angular bounds are defined by the cyclic symmetry of the input component by default. For example, the angular bounds of a C3 input component are `0` and `120°`. The final search space is defined by combining the DOF assignments and boundaries.

Restricting additional DOFs

For some docking problems, a user may want to restrict either one or all of the rotational or translational DOFs of their inputs during the search; for example, some docking problems require specific building blocks to be aligned to additional symmetry axes (Huddy et al., *in preparation*, Dowling et al., *in preparation*, Dosey et al., *in preparation*, Li et al., *submitted*). The rotational and/or translational DOFs can be turned off (`--fixed_rot`, `--fixed_trans`,

`--fixed_components`) or restricted to a user-defined range (`--fixed_wiggle`). These are activated by listing which inputs should be fixed (0-delimited; e.g., `--fixed_rot 1` to restrict the rotation DOF of all .pdb files provided in `--inputs2`, or `--fixed_rot 0 1` to restrict the rotation DOF of all .pdb files provided in `--inputs1` and `--inputs2`).

- `--fixed_rot`: fix the rotational DOF for desired input component
- `--fixed_trans`: fix the translational DOF for desired input component
- `--fixed_components`: fix both the translational and rotational DOFs for desired input component
- `--fixed_wiggle`: limit the translation and rotational DOFs to a certain range from the starting position. Additionally, specifications for the upper and lower bounds (ub and lb) of translation and rotation are required (`--fw_rot_lb`, `--fw_rot_ub`, `--fw_trans_lb`, `--fw_trans_ub`), where the upper and lower bounds are not equal.

The `--flip_components` option can be used to specify which cyclic components proceed to DOF sampling both before and after rotating the input .pdb 180° along the X axis (“flipping”). For example, a C3 oligomer can sample along the Z axis 0 to 120° and also 0 to 120° after flipping. This is effectively identical to sampling “negative” translations in dihedral, polyhedral group, and stacking architectures, and is required to fully search the available docking space in most symmetries. This option takes a list of boolean values corresponding to each input and defaults to *true* for all components (e.g., `--flip_components 1 1` for a trajectory with two inputs).

Sampling the search space

RPXDock samples the defined search space via the modular sampler objects previously discussed and stores transformation matrices for each component of a set of sampled docked configurations. Each transform is applied to the respective input(s), resulting in a single docked configuration that was sampled during a docking trajectory, and is subsequently used to check for clashes and in some cases “flatness” at each iteration of the search. Clashing is evaluated by the BVH as described above. The “flatness” of a docked configuration is calculated during cyclic and multi-component docking (i.e., polyhedral group, stacking, wallpaper). During cyclic docking, flatness refers to the orientation of the longest physical axis of the input .pdb, as defined by principal component analysis, relative to the cyclic symmetry axis. The “flatness” of a cyclic dock can be constrained using the `--max_longaxis_dot_z` option, which restricts the orientation of the input .pdb relative to the cyclic symmetry axis (conventionally aligned along Z) by calculating the cosine between this axis and the longest axis of the input .pdb. Docks that exceed the cosine value given by the `--max_longaxis_dot_z` option are removed from the next stage of the search. This option can be set to any value between 0 and 1 (inclusive), where 1 allows the input .pdb to adopt any configuration relative to the cyclic symmetry axis, while 0 constrains the long axis of the input .pdb to perfect alignment, or perpendicularity, to the cyclic symmetry axis. During multi-component docking, flatness refers to differences in the translation of each component along its respective symmetry axis. In this case, the `--max_delta_h`

option can be used to set an upper bound on the maximum allowable difference in offset between components.

Global and hierarchical search

In the asymmetric two-body docking problem, there are six DOFs: three translational and three rotational, where one body samples all six DOFs while the other remains static. As all six DOFs are sampled explicitly, the total number of transforms to evaluate equals the number of top-level samples (which is determined by the type and resolution of each DOF) multiplied by the number of subdivisions of that DOF raised to the 6th power. For example, if a typical top-level search space with six dimensions comprised of 10,000,000 total samples is used, sampling a single transform in a 16 Å space at a resolution of 16 Å for each dimension would result in 10,000,000 * 1^6 = 10,000,000 total transforms across the entire search space. Sampling at this 16 Å space at 1 Å resolution for each dimension (16 transforms per dimension) would result in 10,000,000 * 16^6 = 167 trillion total transforms to sample the entire search space. Enumerative sampling, even with some dimensionality reduction as implemented in previous iterations of "slide into contact" docking, is prohibitive at a reasonable resolution in architectures with a high number of DOFs (Fallas et al. 2017; King et al. 2014; Brouwer et al. 2019; Marcandalli et al. 2019).

To enable efficient exploration of the search space in such architectures, we implemented an iterative hierarchical search that prunes away areas of the search space unlikely to contain good solutions (Morrison et al. 2016; Ibaraki 1976). In this sampling and evaluation scheme, the search begins at low resolution and is repeated with increasing resolution at each iteration. Only the top-scoring regions of the search space are kept for further exploration in the next iteration (**Figure 1.3A-B**). This reduces the number of samples that must be evaluated at each stage such that the total number of transforms evaluated no longer grows exponentially with dimension. For the simple 2D illustration in **Figure 1.3A**, the total number of samples is reduced from 256 to 24. Efficiency gains are roughly exponential with dimensionality and are thus much higher for less constrained docking problems. At each resolution, configurations are scored as an implicit ensemble (**Figure 1.3B**) through the use of residue-pair motifs (see "Score functions and motifs" section), tuned to provide an approximation of the best possible score within a corresponding ensemble of residue pair positions (**Figure 1.3C**). By evaluating the best possible score within an ensemble, as opposed to an average score, an entire region of docking space can be discarded without missing desirable docks.

Due to the reduction in search space, the hierarchical search and scoring of a typical system with the default search space parameters takes approximately 30 seconds on a 4-core cpu. Further reductions or expansions in the number of transforms sampled at each stage of the hierarchical search protocol can be implemented using the `--beam_size` option, which defines the maximum number of sampled docks taken to the next stage of a hierarchical search protocol (default 100,000). The `beam_size` excludes the first, most coarse stage, which samples the entire search space at the lowest assigned resolution, as defined by `--ori_resl` (default 30°) and `--cart_resl` (default 10 Å).

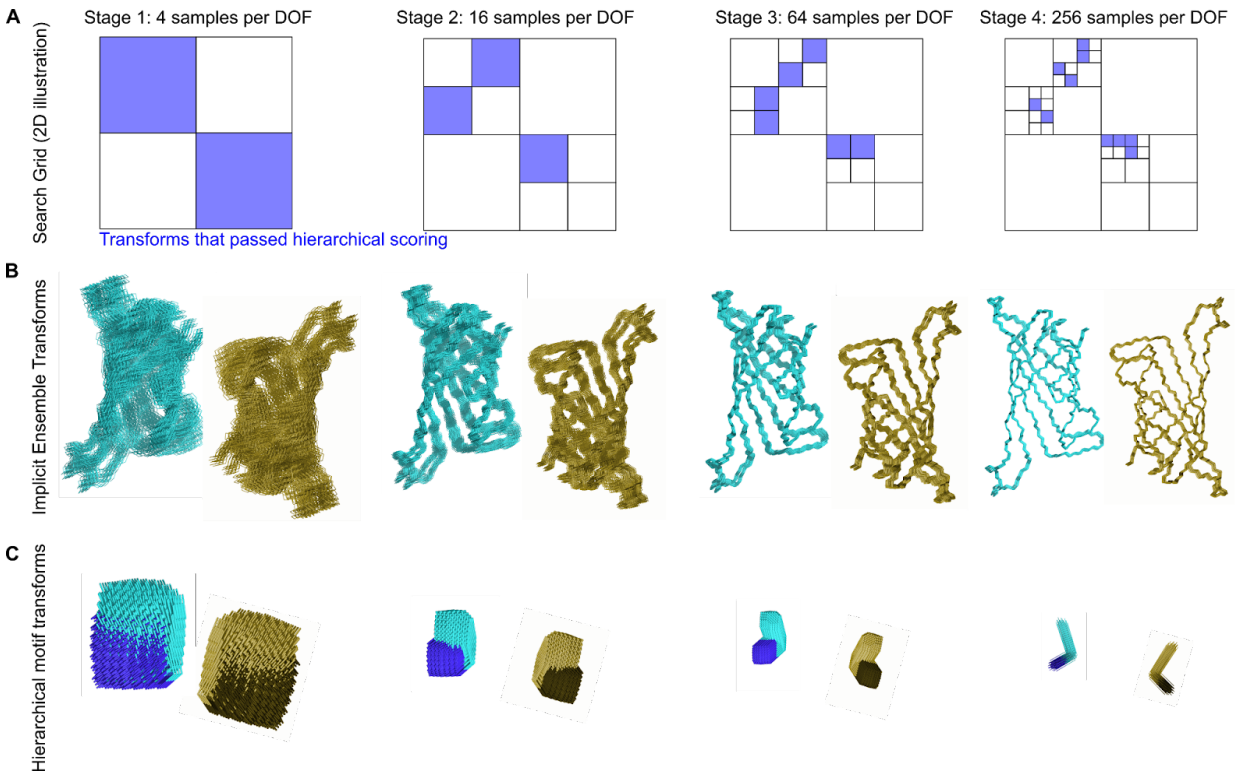


Figure 1.3: Schematic representation of hierarchical sampling.
A. Schematic of a search grid for a single DOF keeping only the transforms that passed hierarchical scoring (blue) at each stage of search. This reduces the space searched at later stages where the search grid is subdivided at increasing resolution. **B**. A schematic depiction of protein backbones sampled with increasing resolution. The backbones shown would correspond to a single blue box at each stage of the search depicted in panel A; a cloud of such backbones would be sampled for each of the distinct docked configurations corresponding to each blue box. **C**. Residue-pair motifs are also evaluated at increasing resolution during each iteration of the search.

To reduce low-resolution artifacts, we take the upper bound of scores within each grid square in the hierarchical search rather than its potentially low scoring and non-representative low resolution center (average). This effectively gives each grid square the “benefit of the doubt” during iteration, so that poor scoring regions can be discarded with confidence. We found during empirical testing that the hierarchical search approach did not over-prune substantial numbers of “good” candidates (**Figure S1.1**). Specifically, we compared how efficiently the hierarchical sampling method recovered the top docks identified by enumerative grid sampling (~10^8 total asymmetric docks) (**Figure S1.1A**). The hierarchical search method recovered the top dock in this test set by searching less than $1/10^5$ of the total search space and the top 10 docks by searching less than $1/10^2$ (**Figure S1.1B**). To find the top 100 and top 1000 docks, the hierarchical method searched nearly the entire space, although it recovered 80% of the top 100 docks within $1/10^4$ of the total search space. As identifying ~10 top docks per input or input pair is reasonable for most docking problems in practice, the reduction in search space and the consequent reduction in time that the hierarchical search method requires to find top-scoring docks is most likely an acceptable tradeoff.

While a major advantage of RPXDock is utilizing the hierarchical search method (`--docking_method hier`), it is possible to globally search the conformation space (`--docking_method grid`). This option may be appropriate for one-component dihedral or polyhedral docking problems that have two or fewer DOFs available. As the global search space is sampled at a single resolution, the user should specify different search resolutions in translational and rotational DOFs (`--grid_resolution_cart_angstroms` and `--grid_resolution_ori_degrees`) across multiple independent trajectories. Nevertheless, grid search is implemented mainly for debugging purposes and is not recommended for production runs.

Specifying termini direction and accessibility

For polyhedral group architectures, the orientation and accessibility of the components' termini can be important for downstream applications such as multivalent antigen display via genetic fusion (Marcandalli et al. 2019; Ueda et al. 2020; Boyoglu-Barnum et al. 2021; Walls et al. 2020; Brouwer et al. 2019). Two options are implemented for polyhedral group architectures to (1) restrict the sampling space to docks with termini in the desired orientation (`--termini_dir[n]`) and (2) evaluate the accessibility of the termini residues (`--term_access[n]`). The `--termini_dir[n]` and `--term_access[n]` options mirror the syntax of the `--inputs[n]` options, where `--termini_dir1` and `--term_access1` refer to the termini direction and accessibility for `--inputs1, --termini_dir2` and `--term_access2` for `--inputs2`, etc. Both options operate by aligning an ideal 21-residue alpha helix to 7 residues at the user-specified termini.

The `--termini_dir[n]` option evaluates the helix orientation by calculating the Z direction of the vector defined by the center of mass of the first and last three residues of the aligned ideal helix (e.g. residues 1-3 to 19-21 for N termini, and inversely for C termini). The option then picks the desired orientation from the required `--flip_components` option and disables sampling of the other. The aligned ideal helix is removed before sampling docked transforms.

The `--term_access[n]` option evaluates the accessibility of user-defined termini during sampling by adding the aligned ideal helix to the *Body* class for the BVH to use for clash checking at each step of the search. The aligned helix is omitted in the *Score* and *Result* classes for *RPX* scoring and output. The option syntax is as follows:

- `--termini_dir1` [`--termini_dir2`, `--termini_dir3`]: Accepts a desired orientation as "in", "out", or "None", for the amino terminus followed by the carboxyl terminus (space-delimited) for each corresponding input (`--inputs1` for `--termini_dir1`, `--inputs2` for `--termini_dir2`, etc.). "In" restricts sampling to configurations in which the specified terminus points towards the architecture's center of mass, while "out" restricts sampling to the opposite. This option alternatively accepts a space-delimited pair of booleans where "in" is *True*, and "out" is *False*. The option(s) default to "None". The `--flip_components` option must be set to *True*.

- `--term_access1` [`--term_access2`, `--term_access3`]: Accepts a space-delimited pair of boolean values to enable evaluation of terminus accessibility at the amino and carboxyl termini of each component, respectively. (e.g. `--term_access1 0 1` evaluates accessibility of the carboxyl termini for input .pdb files passed through `--inputs1`)

Evaluating docked configurations

Residue-Pair Transform (*RPX*) Scoring

We employ a 6D implicit side-chain methodology when evaluating residue-pair interactions in a sequence-independent manner. The interaction between two residues is represented by the full 6D rigid-body transformation between their respective backbone N, Cα, and C atoms (Fallas et al. 2017). Transforms are binned into six dimensional body-centered cubic lattices, with three dimensions each for translation and rotation. The curved space of rotations is divided into 24 relatively flat cells, with one lattice in each cell. A pre-compiled residue-pair transform, or *hscore*, database of all residue-pair interactions for each amino acid found in structures from the Protein Data Bank (PDB) was binned based on this method and scored using the Rosetta full-atom energy function (Alford et al. 2017). During docking, pairs of residues across a docked interface are assigned an *RPX score*, which is the lowest pre-calculated Rosetta full-atom energy found in the relevant spatial transformation bin of the database. The top-scoring residue pair scores across the interface are evaluated based on a user-defined RPXDock score function (see Ranking dock quality (score functions)). This score was previously found to be more predictive of the interface energy from full-atom sequence-design calculation than the Rosetta centroid energy function or other “coarse-grained” scoring models (Fallas et al. 2017).

Motif-enriched docking

A user may want to diversify or restrict the motifs and secondary structure elements used to score RPXDocked configurations. This can be done using the `--hscore_files` and `--hscore_data_dir` options. The path suffix in `--hscore_files` will be appended to `--hscore_data_dir`, which is the default path to search for *hscore* files. These *hscore* files can be read in as a tarball zipped .txz format that is slow to load but Python version-agnostic, or in a .pickle format that is fast but Python version-dependent. The `--generate_hscore_pickle_files` option can be passed to generate .pickle versions from the .txz file, which can then simply be moved to the corresponding hscore folder before use. Each category of *hscore* files contains scores for a subset of the full residue-pair motif database, restricted to certain amino acid identities and secondary structure elements. By restricting the database, transforms with no motifs found among the chosen subset result in a score of zero, and are thrown out when proceeding to the next phase of the search, thus biasing against the unselected amino acids and secondary structures (H α-helices, E β-sheets, and L loops). Note that RPXDock is sequence-agnostic, meaning the residue identities of the input .pdb are ignored when placing motif pairs. The default motif set only includes pairs involving isoleucine, leucine, and valine; and only in α-helices. The following *hscore* files are pre-compiled

and provided in the Institute for Protein Design public repository at https://files.ipd.uw.edu/pub/rpxdock/hscores.zip:

- ILV_H (default; isoleucine, leucine, valine; helices only)
- AILV_H (alanine, isoleucine, leucine, valine; helices only)
- AFILMV_EHL (all hydrophobic amino acids; all secondary structures: sheets, helices, and loops)

Restricting regions for scoring

Score only SS

Scoring can be restricted to only certain secondary structure elements using the `--score_only_ss` option (any non-delimited combination of 'EHL' for sheets, helices, and loops). When active, only residue pairs where at least one of the two motif pairs reside on the desired secondary structure types will be scored. To additionally restrict such that both motif pairs must reside on the designated secondary structure types, the `--score_only_sspair` option can be used. Conceptually this results in a similar effect as providing *hscore* files for only the desired secondary structure types and will enrich for these motifs. Users should note that these restrictions do not explicitly remove or penalize contacts, which contribute to the docking score independently of motifs, at positions on non-desired secondary structure elements.

Masking (Allowed residues)

To bias the search towards generating interfaces focused on a specific region of the input structure(s), a list of residue positions can be provided using the `--allowed_residues[n]` option. Specifying positions in this way does not prevent other regions of the protein from forming contacts, nor does it affect clash checking. Instead, regions of the protein structure not included as allowed residues simply do not contribute to the score of the docked configuration, thus biasing the search. The `--allowed_residues[n]` option mirrors the syntax of the `--inputs[n]` options, where `--allowed_residues1` refers to the list of allowed residues for `--inputs1`, `--allowed_residues2` for `--inputs2`, and so forth. The `--allowed_residues[n]` options can either be left blank (default), take a single file which applies to all corresponding component inputs, or take a list of files which must have the same length as the list of inputs. The files themselves must contain a whitespace- and/or newline-delimited list of either numbers and/or ranges using Python syntax. For example, a three-lined file:

```
1 2 3 4 5

7:12

80:-1
```

will result in specifying residues 1 2 3 4 5 (first line), 7 8 9 10 11 12 (second line), and 80 through the last residue (third line) as "allowed" for all of the corresponding list of inputs.

Residue numbering starts from one and numeric gaps in the input .pdb files are ignored and renumbered sequentially. Multi-chain inputs will be concatenated into a single chain by default. It is recommended that users sanitize input .pdb files to these standards prior to using RPXDock to prevent unexpected results.

Ranking dock quality (score functions)

RPXDock evaluates dock quality with a score function that summarizes the number of contacting residue-pairs at an interface ("contacts") and the *RPX* score, derived from motif pairs as described above. The *RPX* score is evaluated for each pair of residues in the docked interface within a maximum distance of each other, as defined by the `--max_pair_dist` option (default 8.0 Å at the highest search resolution), which scales with the resolution during the hierarchical search. Afterwards, all the relevant *RPX* scores are combined according to the score function definition, controlled by the `--function` option. The default score function (`stnd`) is defined as:

$$score = a * RPX + b * ncontact$$

where *a* and *b* are coefficients set by `--weight_rpx` and `--weight_ncontact` (default 1 and 0.001, respectively), *RPX* is the sum of the maximum *RPX* scores for each pair of contacting residues (i) in the interface ($\sum_{i=1}^{n} max(motif\_score_i)$), and *ncontact* is the number of pairwise contacts in the interface. In this standard score function, *RPX* is highly covariate with *ncontact*, and thus it is also highly correlated with the total score. As a result, because RPXDock seeks to maximize the score, the standard algorithm will tend to find the largest possible interfaces.

SASA weighted (sasa_priority) score function

It is likely that there is an optimum interface size for each docking architecture and the subtypes within them, due to the apparent relationship between interface size and interface strength of symmetrical assemblies, the latter of which can be a critical determinant of the fidelity of the assembly process (Wargacki et al. 2021; Asor et al. 2019). Therefore, the user may wish to bias docked conformations toward a particular interface size. This can be achieved by taking advantage of the correlation between *ncontact* and interface size, as measured by buried solvent accessible surface area (SASA) (Durham et al. 2009) (**Figure S1.2**). The `sasa_priority` score function seeks to find the best docked configuration for a target interface size as measured by the average motif quality $\overline{X}_{RPX}$ across all residue-pair combinations. For each residue pair, the maximum motif score is considered in this average. Thus, the `sasa_priority` score function is defined as:

$$score = a * \overline{X}_{RPX} + b * ln\, N\left(\mu, \sigma^2\right)$$

where *a* is set by `--weight_rpx` (default 1) and *b* by `--weight_ncontact`. Note that while the default value of `--weight_ncontact` in the standard score function is 0.001, a value of 5 is recommended for the `sasa_priority` score function. `N` is the number of contacting residues in the interface, scored based on a log normal distribution with a mean, μ, set by `--weight_sasa` (default 1152 Å^2) and a tolerance level, σ, set by `--weight_error` (default 4). The resultant top-scoring configurations are biased towards the mean (**Figure 1.4A**), such that the buried SASA of the top docks at or very close to the `weight_sasa`, should such docks exist (**Figure 1.4B**). An artifact of this score function is that at higher target interface sizes, a set of high-scoring docks with small SASA estimate values emerge as a result of very small interfaces with high average *RPX* score; these outliers can be removed by the `filter_sasa` (see Additional Optional Filters below).

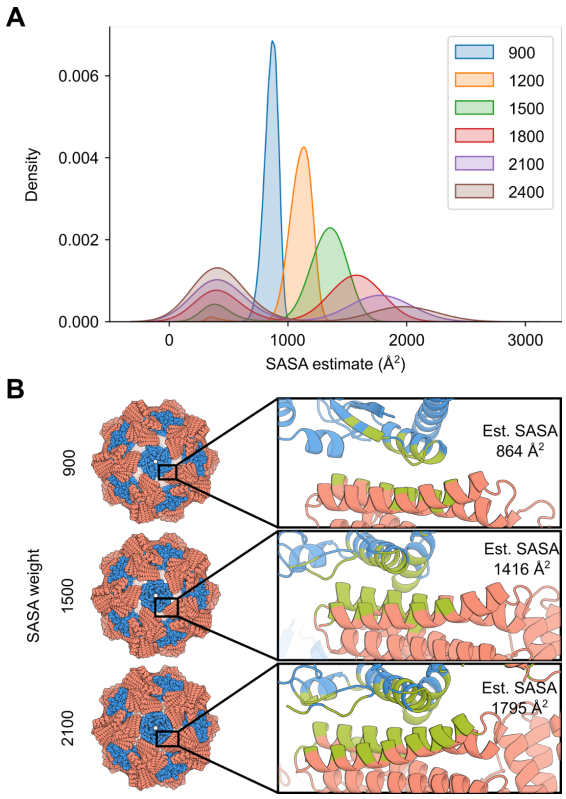


Figure 1.4: Interface size bias by the `sasa_priority` score function.
A. 572 pairs of inputs were docked in a two-component icosahedral architecture at a `--weight_sasa` value of 900, 1200, 1500, 1800, 2100, and 2400, with total area under each curve normalized to 1. **B.** The interface of the top-scoring docked configuration for `--weight_sasa` value of 900, 1500, and 2100 is highlighted (green). Estimated buried SASA calculated using Rosetta for these docks are 864, 1416, and 1795 Å^2, respectively.

The `--weight_sasa` parameter may need to be modified depending on the docking problem. For example, cyclic docking might require a different `--weight_sasa` parameter than one- or two-component polyhedral group docking. The optimal `--weight_sasa` may be determined

empirically for each architecture or docking problem by comparing independent docking trajectories and visually inspecting the results. Note that if the value is set to improbably high values (e.g., 99999), the search will fail rather than finding the largest interface, as docks near that SASA value do not exist. Note that this score function was fit using two-component polyhedral group architectures, so other architectures may need additional optimization of the variables. The development and optimization of this score function is described further in the supplemental information.

Other score functions

Additional variants of the standard score function are available, by replacing the *sum* of the maximum *RPX* scores at each residue pair considered in the *stnd* score function with the *mean* or *median* (**Table 1.2**). These two score functions partially remove the correlation between *ncontact* and total *RPX* score. Finally, two more functions were used in development of the `sasa_priority` score function that empirically estimated the relationship between *RPX* and *ncontact* with either a linear or exponential fit.

Table 1.2: List of additional score functions

--function	***Description***
stnd	score = a * *RPX* + b * *ncontact*, where *RPX* is the sum of the max(motif score) across all residue pairs in a docked interface
sasa_priority	Function developed to bias interfaces to a certain size given user requirements. The `-weight_sasa` (default=1152), `-weight_ncontact` (default=0.01 but a value of 5 provides optimum scaling for this score function), and `-weight_error` (default=4) flags must also be specified.
mean	Takes the mean of max(motif score), instead of sum() in the standard score function
median	Takes the median of max(motif score), instead of sum() in the standard score function
exp	scores = *RPX* - 4.6679 * $ncontact^{0.588}$
lin	scores = *RPX* - 0.7514 * *ncontact*

Filtering docks

Clustering

After docked configurations are scored, the results are clustered through redundancy filters. Redundancy checking is performed by the `filter_redundancy()` function, which performs a distance check on the transformed bodies (approximating an unaligned RMSD calculation) and

discards similar transforms with distances below a user-defined cutoff set by the `--max_bb_redundancy` option (default, 3 Å). Cluster size can be controlled by the `--max_cluster` option (default, no limit), which specifies the maximum number of clusters the docked transforms can be sorted into. As docks are sorted by score, only the highest-scoring dock from each cluster is kept. The redundancy filter returns an array of indices corresponding to the docked configurations that pass this filter.

Additional Optional Filters

We have developed a set of modular filters that can be applied post-docking to remove docks that do not meet certain requirements or to provide more information about the results. Currently available filters are:

- `filter_sscount`: Removes docks below a specified number of secondary structure (SS) elements in the docked interface
- `filter_sasa`: Removes docks outside a specified interface SASA using a similar method to the *sasa_priority* score function

New filters can be added without having to modify code in the *search* or *scoring* modules. At the time of publication, filtering is possible for architectures of the cyclic, dihedral, stacking, wallpaper, and polyhedral groups.

Filter behavior is controlled by a .yaml configuration file passed through the `--filter_config` option. This allows facile stacking of an arbitrary number of filters, including multiple instances of the same filter configured in different ways. Filters are defined with a key, or filter label, that can be any arbitrary string without whitespace. All filters have standard and filter-specific parameters. The standard parameters are a “type” parameter and a “confidence” parameter. The “type” parameter must exactly match the name of the filter in the RPXDock main code. The “confidence” parameter is a boolean that controls whether or not the filter will remove docks from the result. If confidence is *False*, the result will report values for all docks, including those that would have been removed had the confidence been set to *True*. Note that if confidence is *True*, a filter can potentially remove all of the results if none of them meet the thresholds, resulting in an empty result object. **Table S1.2-S1.3** provides a list of all available filter-specific parameters.

Result

After RPXDock has been executed, the result class outputs a zipped tarball .txz file and a .pickle file that stores i) the initial body object along with ii) the transforms and iii) associated score and filter values of each docked configuration in a concatenated xarray format. While the .pickle file is faster to access, it is Python version-dependent, so the .txz format is also returned as a version-agnostic output. Each of these output formats can be turned on or off using their respective options: `--save_results_as_tarball` and `--save_results_as_pickle`, which both default to *True*. With the the `dump_pdb()` function, the result object can output the resulting dock in the form of a .pdb file for any given model number, corresponding to the rank of

the desired docked configuration by score. We have included an example Python script in the GitHub repository under `tools/dump_pdb_from_output.py` that demonstrates how to access score and filter information and regenerate docked configurations as .pdb files for any desired dock configuration from either file format. The `--overwrite_existing_results` option, which defaults to *False*, can be passed to overwrite existing outputs for file management purposes.

The top-scoring transforms can be directly output in .pdb format using the `--dump_pdbs` option. When used in combination with `--nout_top N`, which defaults to 10, the top-scoring N transforms can be output in .pdb format from the RPXDock result object. The user may be interested in saving disk space or for other reasons only saving the asymmetric unit (asu) of the resulting dock; this behavior can be set with the option `--output_asym_only`. The `--output_closest_subunits` option can be used in combination, which outputs a .pdb file containing the asu chains in positions that exhibit the highest motif contact count from the symmetric result (eg. the asu chains that are closest to each other in space) instead of the default asu chains in positions defined by each symmetry. This can be useful for visualization and for generating inputs for downstream steps in design pipelines.

Setup and installation

At the time of publication, RPXDock has been verified to compile and function correctly on Linux-based operating systems. To set it up, a user must first clone the public repository of the full source code, which can be found at https://github.com/willsheffler/rpxdock, and set up a proper conda environment using the `environment.yml` file. Note that a user must obtain a pyrosetta license (free for non-profit users) and update the username and password fields for their pyrosetta license in the `environment.yml` file before creating the environment. Users may need to also install additional packages in their conda environment such as `pyyaml` to properly build the application. To build and compile the codebase with the newly created conda environment, a user may simply run the *pytest* command using a gcc>9-compatible compiler.

To verify that the code compiled properly, execute `rpxdock/app/dock.py --help` in the new conda environment. The output should provide a list of options that are relevant for docking (**Table S1**). Note that several options are still experimental in nature and therefore are not described fully in this publication. For a template of how to set up a simple RPXDock run, please refer to the available example provided in `tools/dock.sh` in the GitHub repository.

Experimental characterization of one- and two-component polyhedral self-assembling proteins from RPXDock

We set out to experimentally evaluate symmetric one- and two-component structures with polyhedral group symmetry generated using RPXDock. Given a set of prevalidated homomeric scaffolds with cyclic symmetry, we generated docks using RPXDock, and the resulting interfaces were sequence-optimized via Rosetta sequence design (Leman et al. 2020). Two

one-component designs (T3-rpxdock-02, I3-rpxdock-71) and two two-component designs (O43-rpxdock-15, O43-rpxdock-HO11) with tetrahedral, octahedral, or icosahedral symmetry were examined by negative-stain electron microscopy and found to adopt the intended architecture (**Figure 1.5A-D, Figure S1.5A**). I3-rpxdock-71, while completely independently sampled and designed, resembles a dock previously sampled by RPXDock's predecessor, tcdock, indicating that the similar top results are identified by the new search algorithm (Hsia et al. 2016). We obtained a 3.7 Å resolution single-particle reconstruction of the two-component octahedral assembly O43-rpxdock-EK1 (PDB: 8FWD, EMD-29502) using cryogenic electron microscopy and found that it assembles to the intended structure with high accuracy (4 Å Cα root mean square deviation between all 48 chains of the original dock and cryoEM structure; **Figure 1.5E, Figure S1.5B-F, Table S1.5**). Together, these data confirm that docks generated using RPXDock can be designed to assemble in the intended configurations without disrupting the integrity of the starting scaffolds. Input .pdb files, docking and design scripts, and design models are provided in the `tools/` directory available on the RPXDock GitHub page at https://github.com/willsheffler/rpxdock.

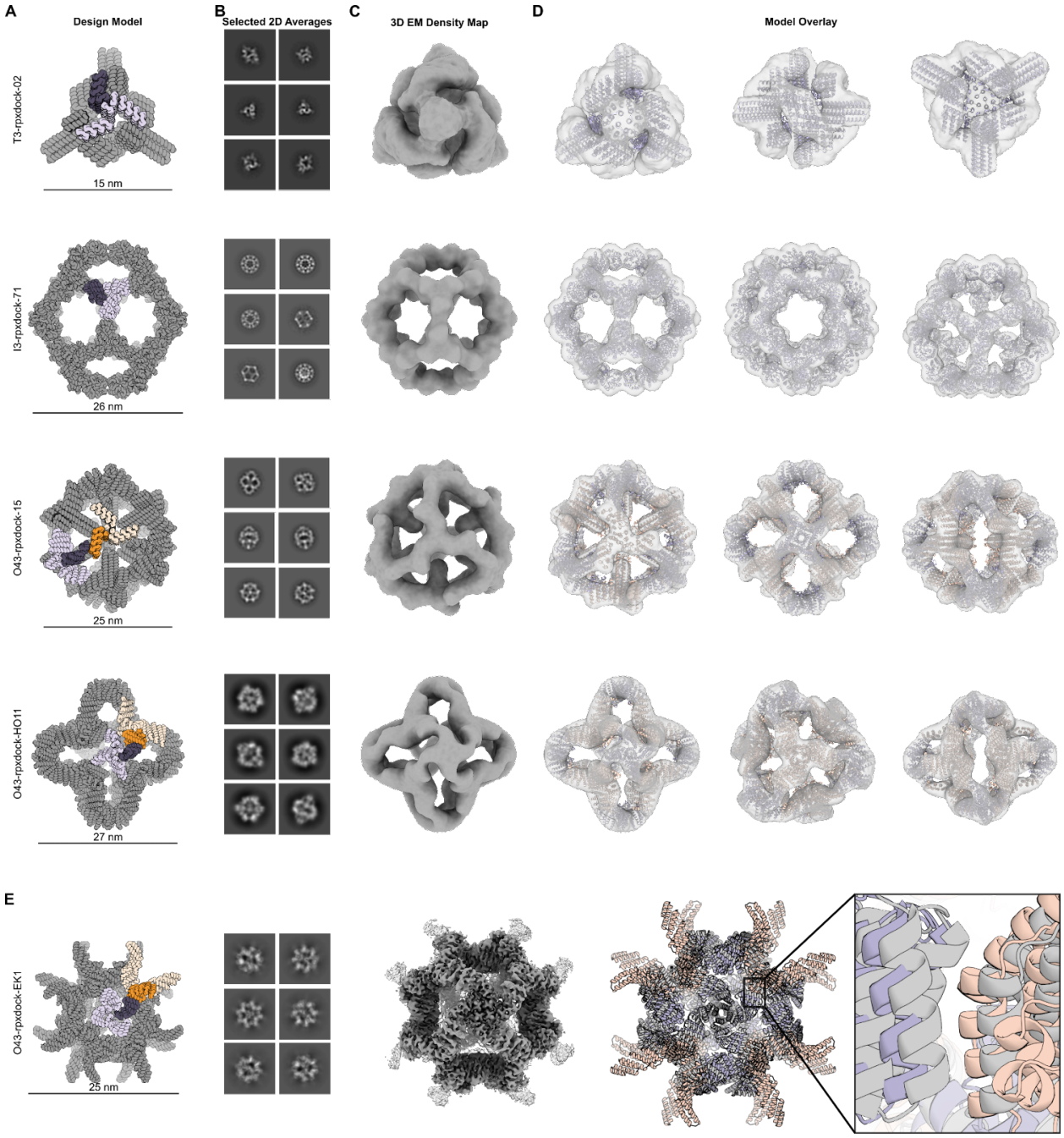


Figure 1.5: Docking and characterization of one- and two-component polyhedral assemblies using RPXDock.

A. Models of one- and two-component docked polyhedral assemblies with the oligomeric building blocks in purple and orange. The asymmetric unit of each assembly, comprising one subunit of each building block, is colored dark purple and dark orange. **B.** Reference-free 2D class averages from negative stain electron microscopy. Each assembly is viewed along several axes of symmetry. **C.** 3D density maps reconstructed from selected 2D class averages. **D.** Overlays of each design model fit into its 3D density map, confirming that each design assembles to the architecture identified by RPXDock. **E.** Characterization of the two-component octahedral assembly O43-rpxdock-EK1 by cryogenic electron

microscopy. The design model is colored as in A). To the right are representative 2D class averages showing different axes of symmetry and a reconstructed 3D map at 3.7 Å resolution. The overlay of the original dock (orange and purple) with the model built from the 3D reconstruction (gray) shows 4 Å Cα root mean square deviation between the original dock and cryoEM structure over 48 chains.

Discussion

RPXdock provides a powerful and general route to modeling, sampling, and scoring symmetric protein complexes across multiple symmetric architectures. Docking monomeric and oligomeric building blocks into higher-order symmetric complexes followed by protein-protein interface design is an established and successful paradigm for accurately creating novel self-assembling protein nanomaterials (King et al. 2014, 2012; Fallas et al. 2017; Shen et al. 2018; Ben-Sasson et al. 2021; Bale et al. 2016; Gonen et al. 2015). While deep learning-based generative models have recently proven successful in designing de novo oligomers (Wicky et al. 2022) and small nanocages (Lutz et al. 2022), the ability of RPXdock to use experimentally verified or previously designed scaffolds in a stepwise manner enables the use of specific building blocks that have optimal features for specific applications (Marcandalli et al. 2019; Boyoglu-Barnum et al. 2021; Ueda et al. 2020; Walls et al. 2020; Brouwer et al. 2019). The RPXdock code can accommodate specific user requirements for complex docking problems, and the efficiency at which high-quality docks are found has been greatly improved compared to its predecessors (tcdock; sicdock; sicaxel (King et al. 2014; Fallas et al. 2017; Brouwer et al. 2019; Marcandalli et al. 2019)) due to the hierarchical search and scoring algorithms. While new capabilities are continuously under development, the core software structure is complete and robust, and has already been successfully applied to a number of symmetric docking and design problems in addition to the structures presented here ((Wang et al. 2022; Li et al. 2022; Gerben et al. 2023); Huddy et al., *in preparation*, Dowling et al., *in preparation*; Lee et al., *in preparation*; Dosey et al., *in preparation*). Any future modifications and new modules added to the RPXDock application will be updated via the GitHub repository: https://github.com/willsheffler/rpxdock.

Supplemental Information

Bodies

The body class uses pyrosetta to access the pose and initial coordinates of the input .pdb files for a particular docking trajectory. From the pyrosetta pose, the body class stores chain, sequence, secondary structure, and backbone positional information of the asymmetric unit. In this class, any user inputs to allow only certain portions of the pose for docking are also stored (`--allowed_residues`, `--term_access`. Only the initial coordinates of the pyrosetta pose are stored in the body class, while the transformation matrix generated by the search is applied to the initial coordinates of the starting pose. Backbone positional information derived from the transforms are stored as clouds of points during the hierarchical search. The body class checks for clashes between transformed backbones by looking for intersections between these clouds of points at each level of the bounding volume (`intersect_range`, `intersect`, and `clash_ok`). At lower resolutions, the clouds of points are smoother and larger, and at higher resolutions, the clouds of points are smaller. Pairs of contacting positions and secondary structure elements (`contact_pairs`) and counts of contacting pairs (`contact_count`) are also evaluated in the body class.

Search

The search module contains the core code controlling the search process. It contains the two fundamental search methods, hierarchical and grid, the geometric specifications (*spec*) for each architecture, the allowed degrees of freedom (*sampler*), and a module for each docking application depending on the architecture and number of bodies. Which module is used depends on the specific architecture and is called by `dock.py`. They are *asym*, *cyclic*, *onecomp*, and *multicomp*. There are also the special-use search applications for one-dimensional (*helix*), and stacking (*axle*) architectures. Finally, the *search* module contains a *result* object which defines the *result* class and associated functions.

Each module type (*asym*, *cyclic*, *onecomp*, *multicomp,* etc.) has a *make function*, e.g., `make_multicomp()` and one or more *evaluator functions*. The *make function* takes as required arguments a body or bodies, a spec, a motif-score hash-table (hscore), a search method (default `hier_search`), and a sampler (default None). The sampler is `hier_multi_axis_sampler()` for multicomp and `hier_axis_sampler()` for onecomp. *Make functions* manage the execution of the docking process by setting up the evaluator function, which performs a redundancy check and calls any specified filters. The evaluator function, in conjunction with the search method, will evaluate the transforms and scores from the sampler at each search resolution and return the top-scoring transforms to be expanded in the next level of search resolution, affecting the docking trajectory. Finally, at the end of docking, the *evaluator function* generates and returns a result object. Result objects are described in further detail in the main text.

Score

The `sasa_priority` score function takes into consideration both the quality of the motifs in the interface, and also how far the interface is from a desired size. To develop this function, we generated a predictive model by docking a set of oligomeric scaffolds in all two-component polyhedral group architectures using the `stnd` score function and designed the novel protein-protein interfaces of a random selection of docks using the Rosetta software suite to obtain buried SASA scores (Bale et al. 2016). The resulting model fit to the relationship is $SASA = 29.1 * ncontact + 282,\ R^2 = 0.634$. Because `ncontact` correlates strongly with the computationally measured interface size, `SASA`, we parameterized an `ncontact` score term with respect to `SASA` over a range of plausible interface sizes and standard deviations (**Figure S1.2**).

We fit the correlation between the distribution mean and mode of computationally predicted SASA with a linear regression, and the relationship between the standard deviation and the slope of the correlation between the mean and mode of the distribution with a Gaussian decay function. The resulting log-normal distributions have a maximum score at the input SASA, and the score is invariant with respect to the standard deviation (**Fig S1.3**).

The final score function contains an *RPX* score term and an ncontact score term: $score = a * \overline{X}_{RPX} + b * ln\,N\left(\mu, \sigma^2\right)$. The *RPX* score term consists of a scalar multiplier of $\overline{X}_{RPX}$, which refers to the average of the best motifs found across all residue pairs and is used as an approximation of interface quality (Default 1.0). The ncontact score term scores the number of unique contacting pairs (`N`) based on a log-normal distribution set by μ, the desired SASA set by the user, and σ, a user-defined tolerance value that is a scalar multiplier of the standard deviation of the fit error for the correlation between SASA and ncontact, i.e., the accuracy of the prediction for the desired SASA (default 4). Since the *RPX* score term tends to bias towards interfaces that have few but very high quality motif pairs, the weight for the ncontact score term, *b*, needs to be scaled appropriately to overcome the $\overline{X}_{RPX}$ tendency towards small interfaces.

To determine the appropriate default value for *b*, we systematically varied it from 0 to 13 and docked a standard set of scaffolds. The top-, middle-, and bottom-ranked docks were designed using `tools/cage_design.xml`, included in the GitHub repository. Increasing the value of *b* made the interface-size bias to the total score more pronounced, but somewhat surprisingly decreased the maximum interface size observed in all docks (**Fig S1.4A**). The weighting also had an unexpected effect on the average RPX score: The $\overline{X}_{RPX}$ also decreased around the desired SASA as *b* increased (**Fig S1.4B**). Because $\overline{X}_{RPX}$ is calculated using the mean() gather function, as opposed to a sum() as in the `stnd` score function, this result can be interpreted as the ncontact score term weight having a negative impact on the interface quality for a given interface size. This effect converges above an ncontact weighting of 5, although convergence depends on user-defined interface size (**Fig S1.4C**).

We also evaluated the 960 top-, middle-, and bottom-scoring docks for each weighting of the ncontact score term and a target SASA of 1125 $Å^2$ against Rosetta design computational filters including ddG < -20 and SASA between 850 $Å^2$ and 1200 $Å^2$ (included in `cage_design.xml`). Despite the apparent decrease in interface quality as a function of increasing ncontact weight, weights of 5 and above resulted in a higher percentage of top-scoring docks passing Rosetta design filters (**Figure S1.4D**). There was also almost no difference in computationally estimated ddG or SASA for top docks above an ncontact weight of 3 (**Figure S1.4E-F**). In fact, no statistically significant difference between weights could be detected for any computational design metric. Qualitatively the designs from each weighting look similar, with the top dock for each weight converging after weight = 5 (**Fig S1.4G**). Therefore, we selected a default ncontact weight of 5 as the most conservative weighting that also maximizes the design success rate.

Filters

The filter module serves two purposes. The first is to filter redundant docks during the search process, described in the Clustering section. The second is to use the `filter()` function to execute an arbitrary number of filters, defined by the user in a filter config file (in .yaml format). This function takes in the body object and transforms, and parses the config file set with the `--filter_config` argument, calling any filters defined in the config file. For all filters, the `filter()` function returns an array of indices for docks passing all filters if the “confidence” configuration is set to *True*. The function also returns any extra data provided by the filters. Available filters are `filter_sasa()` and `filter_sscount()`.

The SASA filter attempts to estimate the Solvent Accessible Surface Area buried by the formation of a protein-protein interface. The estimate is based on a linear fit of SASA, calculated by the SASA filter in Rosetta, as a function of the number of unique residues in a docked interface. The `filter_sasa()` function takes as arguments transforms and bodies, as well as parsed keyword arguments from the config file. A full list of options for the SASA filter can be found in **Table S1.2**.

The sscount filter attempts to estimate the number of secondary elements in contact at a protein-protein interface. The filter uses the `secondary_structure_map` class, which maps secondary structure elements (either Helix, Sheet, or Loop), based on user-definitions of each secondary structure element, onto the body object. Secondary structure elements are recorded for a particular body object if a consecutive stretch of identified secondary structure types exceed a given minimum residue length controlled by `min_helix_length`, `min_sheet_length`, and `min_loop_length`. Given the number of unique pairs of residue contacts at a protein-protein interface, the `filter_sscount()` function estimates the number of each secondary structure element type that is in contact at the interface. A full list of options for the sscount filter can be found in **Table S1.3**.

Supplemental Figures

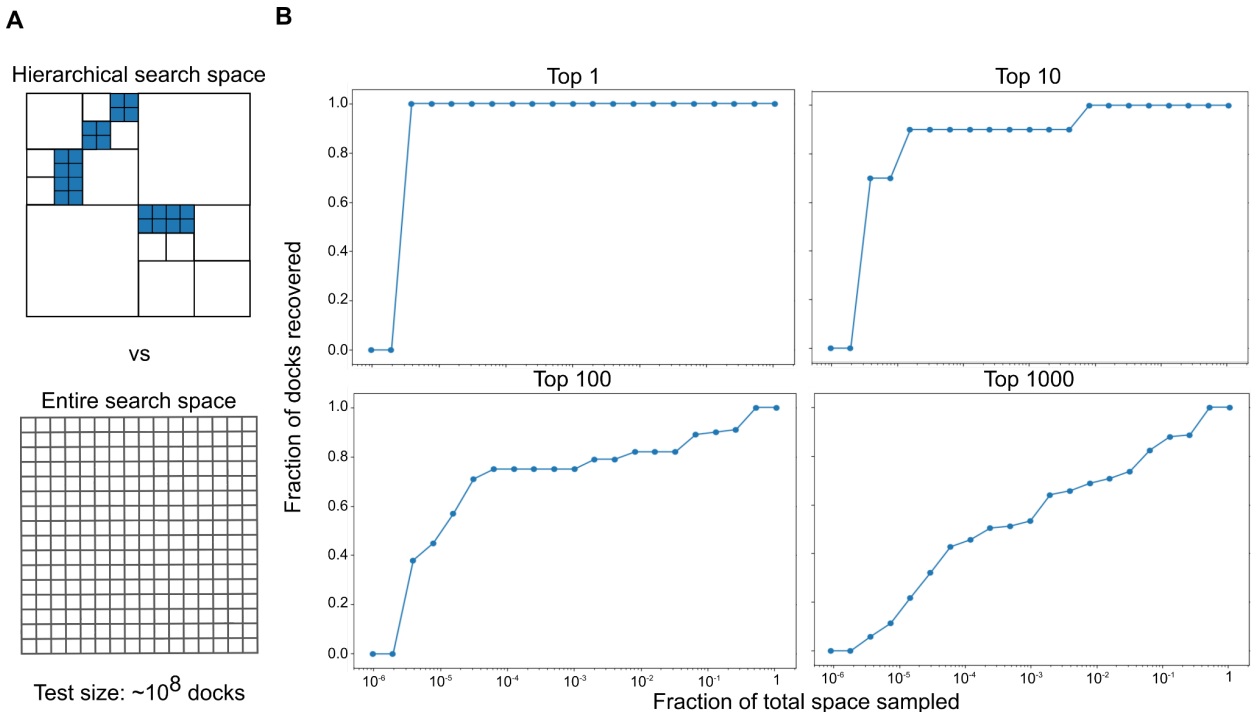


Figure S1.1: Hierarchical docking performance test.
A. A 2-dimensional illustration of a hierarchical search grid with samples searched at the highest resolution in blue vs. a complete search grid at the same resolution. In this test dataset, ~10^8 total docks were sampled. **B**. A cumulative distribution of the fraction of the total search space that needs to be sampled in order to recover the top 1, 10, 100, and 1000 docks from this dataset.

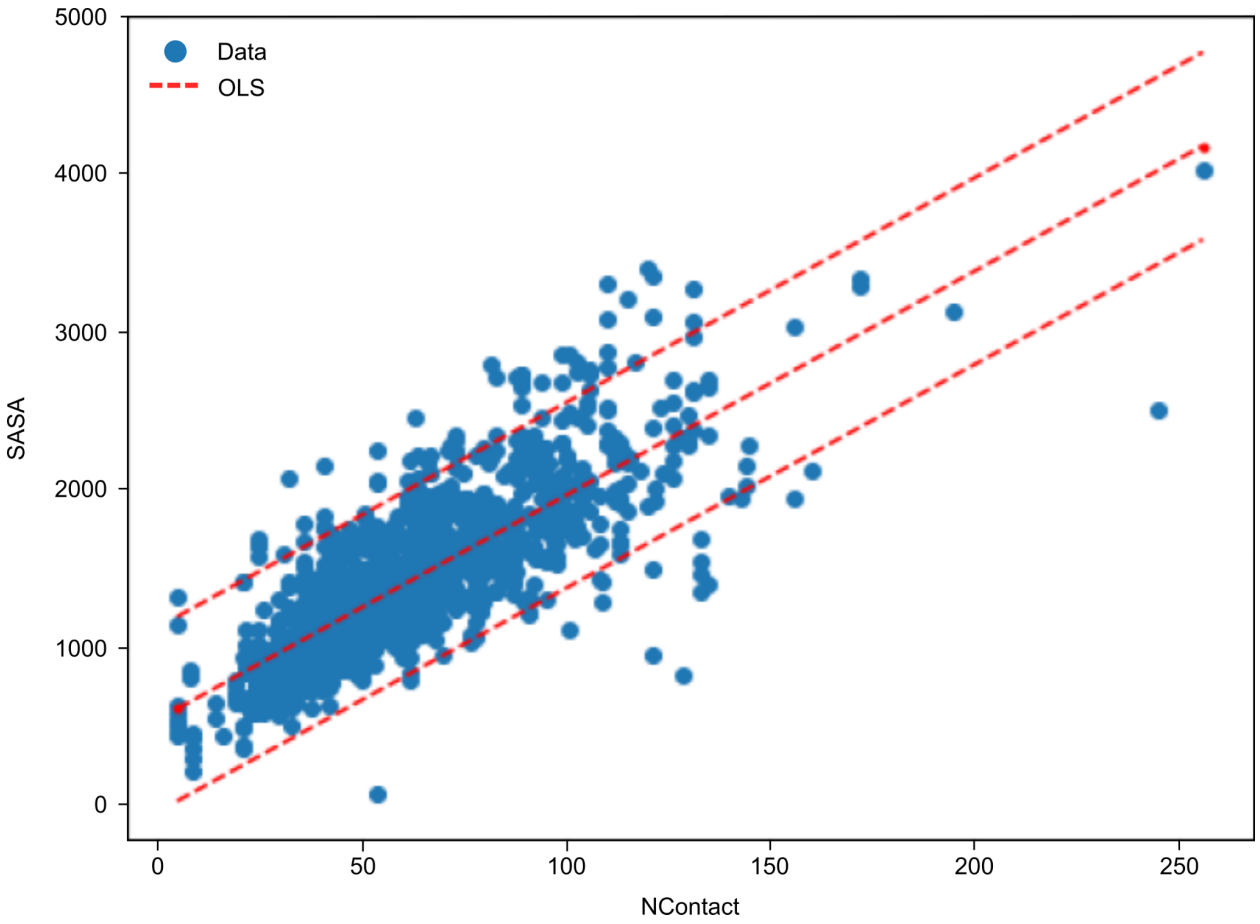


Figure S1.2: NContact and SASA are highly correlated.
As such, we parameterized an ncontact score term with respect to computationally measured interface size, SASA.

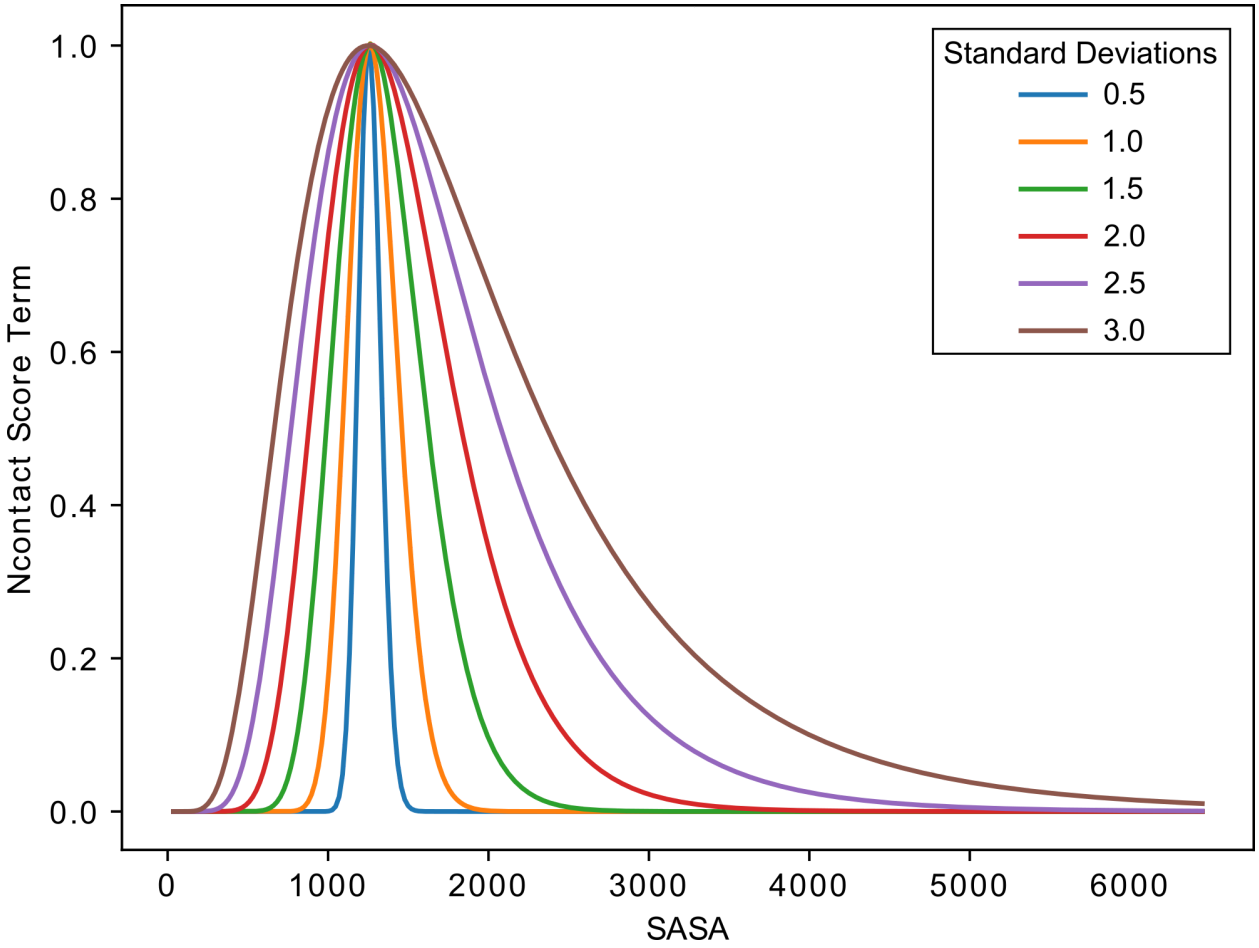


Figure S1.3: Derivation of the ncontact score term as a function of interface size, SASA.
Parameterization of an ncontact score term as a function of interface size, SASA, results in a log-normal distribution with a maximum ncontact score term at a user-input SASA regardless of standard deviation.

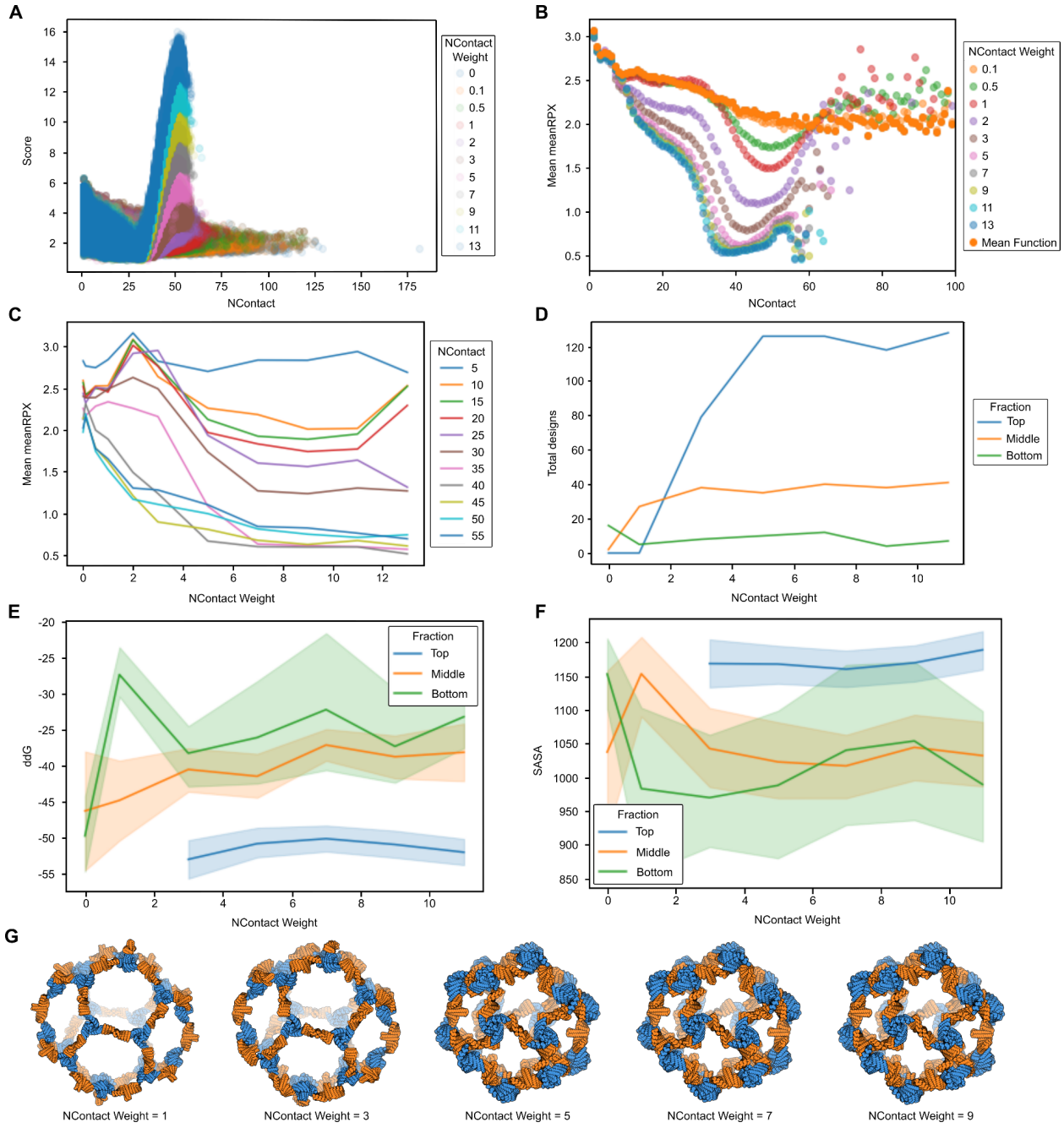


Figure S1.4: Empirical derivation of the ncontact score term weight.
A. Score as a function of ncontact across various ncontact weights. **B.** Mean *RPX* as a function of ncontact. **C.** Mean *RPX* as a function of ncontact weighting plotted for interface sizes from Number of unique contacts = 5-55. **D.** Total number of passing designs out of 960 docks for each weighting and fraction. **E-F.** Computational design metrics as a function of ncontact weight for top-, middle-, and bottom-ranked designs for **E.** ddG, and **F.** SASA. **G.** The top dock with I32 icosahedral symmetry for, left to right, ncontact weight 1, 3, 5, 7, 9.

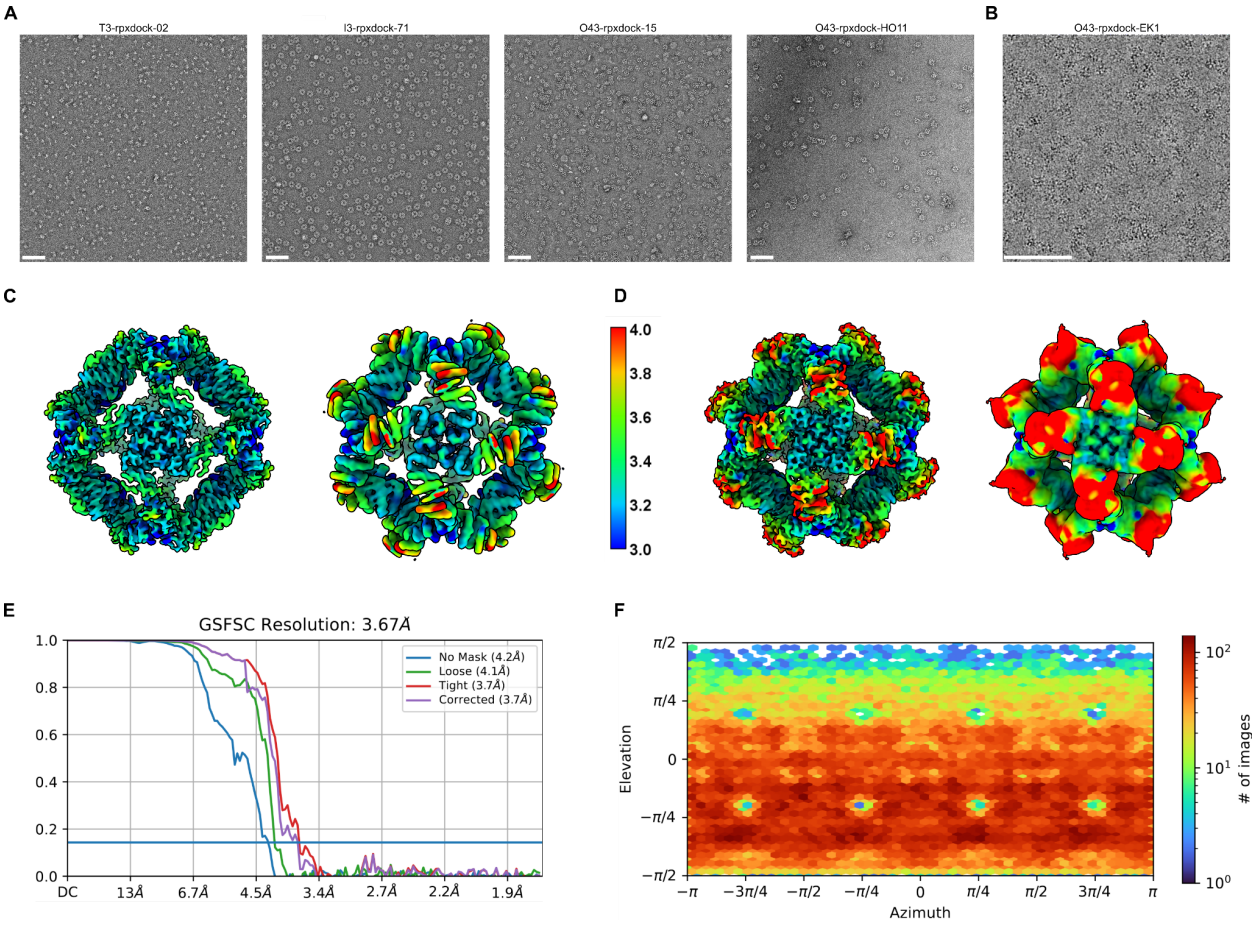


Figure S1.5: nsEM and CryoEM data and associated plots of one- and two-component polyhedral self-assembling proteins from RPXDock.
A. Representative raw nsEM micrographs of one- and two-component polyhedral self-assembling proteins from RPXDock. Scale bar = 100 nm **B.** Representative raw CryoEM micrograph showing good particle distribution and contrast of (Scale Bar = 100 nm). **C.** CryoEM local resolution map of O43-rpxdock-EK1, with the sharpened map at two different contour levels, using a tight mask, and calculated using an FSC value of 0.143. **D.** Local resolution estimates of the unsharpened map, also at two different contour levels (FSC = 0.143). The protruding arms of the designed cage only start to become visible at very low contour levels. Local resolution estimates range from ~3.2 Å at the core to >4.0 Å along the periphery of the extended arms due to a high degree of flexibility within this region. **E.** Global resolution estimation plot. **F.** Orientational distribution plot demonstrating near-complete angular sampling.

Experimental Material and Methods

Computational design

As inputs to RPXDock we used one native scaffold (PDB ID: 1wa3) and cyclic oligomers of either C3 or C4 symmetry generated via rigid helical fusion from validated oligomeric scaffolds and de novo helical repeat proteins as input building blocks for RPXDock (Fallas et al. 2017;

Brunette et al. 2015; Hsia et al. 2021; Boyken et al. 2016). Table S4 describes the input pdb files used to generate each dock, and the asymmetric units of each input pdb file are provided in the `tools/inputs/scaffolds/` directory of the RPXDock GitHub page (https://github.com/willsheffler/rpxdock). Docks were generated using the tools/dock.sh file, also provided on GitHub. We used the sasa_priority score function, providing a value of 1500 or 1125 for the `--weight_sasa` option for one- and two-component docking problems, respectively. Docks were allowed to sample a Cartesian bound space between 0 and 300 Å with the `ailv_h` motif settings. The sequences of the interfaces for the top 10 docks for each scaffold (one-component) or scaffold pair (two-component) were optimized symmetrically using Rosetta sequence design (Leman et al. 2020) with the `tools/rpxdock_to_design.xml` file provided on GitHub. Designable residues at the docked interfaces were selected using Rosetta-based interface selection task operations. The designable residues were split into core, boundary, and surface layers with residue selectors and designed via layer design followed by side chain minimization (Bale et al. 2016; Hsia et al. 2016). The number of side-chain dependent clashes, interface size, and the predicted binding energy of the complexes (ddG) were then calculated for each sequence using Rosetta-based filters.

Protein expression and purification

Synthetic genes were optimized for *E. coli* expression and purchased from IDT (Integrated DNA Technologies) as plasmids in the pET29b expression vector encoding a C-terminal hexahistidine affinity tag. Bicistronic genes encoding two components were joined together by a short intergenic region containing a ribosome binding site (gene sequence: TAAAGAAGGAGATATCATATG) and the hexahistidine tag was included on only one of the two components. Expression plasmids were transformed into BL21(DE3) *E. coli* competent cells (Invitrogen). Single colonies from agar plate with 100 mg/L kanamycin were inoculated in 50 mL of Studier autoinduction media (Studier and William Studier 2005), and the expression continued at 37 °C for over 24 hours. Cells were harvested by centrifugation at 4000 g for 10 min, and resuspended in 2.66 - 35 mL lysis buffer of 300 mM NaCl, 25 mM Tris pH 8.0, 1 mM PMSF, 0.25 mg/mL lysozyme, and 0.25 mg/mL DNase. After lysis by sonication and centrifugation at 14000 g for 45 min, the supernatant was purified by Ni^{2+} immobilized metal affinity chromatography (IMAC) with Ni-NTA Superflow resin (Qiagen). Resins with bound cell lysate were washed with 10 mL (bed volume 1 mL) of washing buffer (300 mM NaCl, 25 mM Tris pH 8.0, 60 mM imidazole) and eluted with 5 mL of elution buffer (300 mM NaCl, 25 mM Tris pH 8.0, 300 mM imidazole). Concentrated samples were purified by SEC in 300 mM NaCl, 25 mM Tris pH 8.0 on a Superose 6 Increase 10/300 gel filtration column (Cytiva).

Negative-Stain Electron Microscopy (nsEM)

Assemblies were diluted to ~0.1 mg/mL (asymmetric unit concentration) and applied onto glow discharged, carbon supported 300-mesh copper grids (Ted Pella, Inc.), followed by 2× application of 4 µl 2% uranyl formate stain. Micrographs were recorded using EPU software (Thermo Fisher) on a 120 kV Talos L120C transmission electron microscope (Thermo Scientific) at a pixel size of 2.47 Å per pixel and a defocus range of 1.0 to 2.5 µm.

Negative Stain Electron Microscopy image processing

nsEM datasets were processed by CryoSPARC software v4.0.3 (Punjani et al. 2017). Micrographs were imported into the CryoSparc software. Around 200 particles were manually picked, 2D classified and selected classes were used as templates for particle picking in all images. All the picked particles were 2D classified for 40 iterations into 50 classes. Particles from selected classes were used for building three asymmetric ab-initio initial models. Each model was then homogeneously refined using C1 and the corresponding T/O/I symmetry independently.

CryoEM sample preparation and data collection

3 µl of O43-EK1 sample at 0.8 mg/mL in 25 mM Tris pH 8.0, 150 mM NaCl was applied onto C-flat 1.2/1.3 holey carbon grids overlaid with a thin layer of continuous carbon. Grids were then plunge-frozen into liquid ethane and cooled with liquid nitrogen using a ThermoFisher Vitrobot Mk IV with 0.5 s blotting time, a 5 second wait time, and 0 blot force. The blotting process took place inside the vitrobot chamber at 22°C and 100% humidity. Data acquisition was performed with SerialEM on a ThermoFisher Glacios electron microscope operating at 200 kV using a K3 Summit direct electron detector operating in super-resolution mode. The nominal magnification for data collection was 45000× with a calculated pixel size of 0.4425 Å/pixel, with a final dose of 50 e-/A2 for 1638 movies.

CryoEM data processing

The raw micrographs were collected on the ThermoFisher Glacios electron microscope using SerialEM and were processed in CryoSPARC v3.0.0, v4.0.2, and v4.0.3 (Punjani et al. 2017). The 1638 raw movies with a raw pixel size of 0.4425Å, total exposure dose of 50 e/Å^2, and spherical aberration of 2.7 mm at an accelerating voltage of 200 kV were imported into the CryoSPARC software package. The imported micrographs were motion corrected using Patch Motion Correction with the maximum alignment resolution of 5, F-crop output factor of 0.5, and B-factor of 500. The motion corrected micrographs were CTF corrected using Patch CTF with amplitude contrast of 0.1, a resolution range between 4 and 21 Å, search defocus range between 1000 and 40000 Å, and search phase shift range between 0 and π radians. 4000 peaks per micrograph were chosen using Blob Picker, with particle diameter set between 175 and 325 Å and a low pass filter of 20 Å applied to both templates and micrographs. Particle picks were manually curated to exclude ice and noise by adjusting NCC and local power thresholds using Inspect Picks. Selected particles were extracted at a box size of 564 pixels and F-cropped to 282 pixels. Particles were sorted into 100 2D class averages over 40 iterations of classification with a batch size of 200. 2D classes showing assembled cages and clear secondary structural features were selected as templates for a second round of particle picking. The template picked particles were iteratively sorted by 2D classifications using varying parameters, and the selected particles were used to generate a single 3D *ab initio* volume using imposed octahedral symmetry, with minimum and maximum resolutions of 34 Å and 12 Å, respectively. The 3D *ab initio* volume was used as the initial reference for a non-uniform refinement that reached a GSFC of 3.73 Å. The unsharpened volume generated from the non-uniform refinement was used to create a mask around the rigid region of the cage,

excluding the flexible arms, lowpass filtered by 15 Å, with a dilation radius of 3 Å and soft padding of 5 Å. The selected particles were re-extracted at a box size of 380 pixels, without F-cropping, and were run through Exposure Group Utilities, splitting the input particles into smaller subsets using a blob/path result field, string_spl token creation strategy, split group index of 5, and fail combine strategy. The Exposure Group Utilities output particles were used to create a non-uniform refinement with imposed octahedral symmetry, initial lowpass resolution of 12 Å, and GSFSC split resolution of 20 with optimized per-particle defocus and optimized per-group CTF parameters. The output volume and particles were used with the previously generated mask to perform a local refinement with window inner and outer radii of 0.65 and 0.7, 20 degree rotation search extent, 10 Å shift search extent, 0.2 degree maximum alignment resolution, imposed octahedral symmetry, 12 Å initial lowpass resolution, non-uniform filter order 2, batchsize epsilon of 0.001, batch size snr factor of 50, and a 34 Å GSFSC split resolution using a symmetric noise model and noise initial sigma scale of 3. The local refinement yielded a final 3.67 Å map. For comparison, an identical local refinement was performed without any symmetry imposed on the particles at any stage, resulting in a 5.37 Å map.

CryoEM model building and validation

A Gaussian version of the unsharpened local refinement map was created in UCSF ChimeraX (Pettersen et al. 2021) and the design model was iteratively relaxed into the Gaussian, unsharpened, and sharpened versions of the map. Each iteration used the previous output structure as the input structure for the density guided relaxation tool, Namdinator (Kidmose et al. 2019). The final Namdinator output model was aligned with the sharpened map. ISOLDE (Croll 2018) was used to run a global simulation of molecular dynamics based on the AMBER force field model, followed by smaller simulations on each chain individually and the regions containing interfaces and by local simulations and manual adjustments of Ramachandran outliers, rotamer outliers, and steric clashes above 0.5 Å ngstroms, alternating with periodic global simulations to avoid local minima. The manually adjusted ISOLDE output model was real space refined in Phenix to correct for unfavorable bond lengths and angles. The remaining Ramachandran outliers, rotamer outliers, and clashes were conservatively adjusted in ISOLDE before a single global minimization macrocycle in Phenix. Finally, side chains and sections of the backbone that were not clearly supported by density were deleted from the final solved structure and the final structure was evaluated in Phenix, showing a clash score of 1.5, molprobity score of 0.89, 0% Ramachandran outliers, 0% CBeta outliers, 0.05% rotamer outliers, 0% twisted prolines, 0.14% CaBLAM outliers, bond length RMSD of 0.003 Å, bond angle RMSD of 0.522 (0) degrees, and whole, helix, and loop Ramachandran plot Z scores (RMSDs) of 1.38 (.11), 1.01 (0.07), and 0.94 (0.32), respectively. The final coordinates and cryoEM maps for O43-rpxdoc-EK1 were deposited in the Protein Data Bank and Electron Microscopy Data Bank under accession numbers PDB: 8FWD and EMD-29502, respectively.

Table S1.1: All RPXDock command line options

Option	Default	Description

`-h, --help`		Show a full list of RPXDock options
BVH and Bodies		
`--architecture`	None	architecture to be produced by docking. Can be Cx for cyclic, Dx_y for dihedral, where x is the dihedral symmetry and y is the symmetry of the scaffold, y=2 or y=x, or polyhedral group where the larger axis of symmetry is listed first. Axle docking requires AXLE_X where X is the symmetry of the two input scaffolds. For symmetric axle docking of two scaffolds of different symmetry, use AXLE_1_X_Y where X and Y are the cyclic symmetries of each scaffold. (e.g. AXLE_1_2_3 would dock a dimer against a trimer). Px_yz for layer architectures where Px describes the lattice symmetry and y and z are the scaffold cyclic symmetries.
`--inputs1`	None	input structures for single component protocols or first component for 2+ protocols. Can be inputted as a string or list of strings
`--inputs2`	None	input structures for the second component for 2+ component protocols.
`--inputs3`	None	input structures for the third component for 3+ component protocols.
`--max_longaxis_dot_z`	1.000001	maximum dot product of the longest input axis (as determined by PCA) with the main symmetry axis (the cosine of the angle between the two axes). Can be used to force cyclic docks to lay flat.
`--max_delta_h`	9999	maximum difference between cartesian component offsets for multicomponent symmetry axis aligned docking such as cages and layers.
Defining the Search Space		
`--cart_bounds`	None	cartesian bounds for various protocols. no default as it's protocol specific
`--flip_components`	*True*	list of boolean value(s) specifying if and which components should be allowed to flip in axis aligned docking protocols.
`--fixed_rot`	None	list of components (0,1,2,etc) which should be fixed from rotating in hierarchical docking
`--fixed_trans`	None	list of components (0,1,2,etc) which should be fixed from translating in hierarchical

		docking
`--fixed_components`	None	list of components (0,1,2,etc) which should be fixed from rotating *and* translating in hierarchical docking
`--fixed_wiggle`	None	Similar to fixed_components (input as list 0,1,2,etc) but allows user-inputted translation and rotation wiggling about orientation axis in hierarchical docking
`--fw_cartlb`	-5	Lower bound for fixed_wiggle translation (in Angstroms)
`--fw_cartub`	5	Upper bound for fixed_wiggle translation (in Angstroms)
`--fw_rotlb`	-5	Lower bound for fixed_wiggle rotation (in degrees)
`--fw_rotub`	5	Upper bound for fixed_wiggle rotation (in degrees)
`--termini_dir1`	None	Restrict sampling for termini orientation of inputs1 either in (*True*) or out (*False*) for each amino or carboxyl termini specified
`--termini_dir2`	None	Restrict sampling for termini orientation of inputs2 either in (*True*) or out (*False*) for each amino or carboxyl termini specified
`--termini_dir3`	None	Restrict sampling for termini orientation of inputs3 either in (*True*) or out (*False*) for each amino or carboxyl termini specified
`--term_access1`	None	Sample docks that pass termini accessibility compliance for inputs1 at either the amino or carboxyl termini, respectively.
`--term_access2`	None	Sample docks that pass termini accessibility compliance for inputs2 at either the amino or carboxyl termini, respectively.
`--term_access3`	None	Sample docks that pass termini accessibility compliance for inputs3 at either the amino or carboxyl termini, respectively.
Sampling the Search Space		
`--docking_method`	hier	search method to use in docking. available methods may include "hier" for hierarchical search (probably best), “grid” for a flat grid search, and "slide" for a lower dimension grid search using slide moves. Not all options available for all protocols (grid is not available for multicomp docking).

--nresl	None (All stages)	number of hierarchical stages to do for hierarchical searches. probably use only for debugging
--cart_resl	10	resolution of top level cartesian, sometimes ignored, and resolution is taken from hscore data instead
--ori_resl	-30	resolution of top level orientation, sometimes ignored, and resolution is taken from hscore data instead
--grid_resolution_cart_angstroms	1	cartesian resolution in Angstroms during grid search
--grid_resolution_ori_degrees	1	rotation orientation resolution in degrees during grid search
--beam_size	100000	Maximum number of samples for each stage of a hierarchical search protocol (except the first, coarsest stage, which must sample all available positions). This is the most important parameter for determining runtime (aside from number of allowed residues list)
Scoring		
--function	stnd	score function to use for scoring. Default is stnd score function. Example: stnd, sasa_priority, mean, exp, median. Full list is defined in score/scorefunctions.py
--weight_rpx	1	score weight of the main *RPX* score component.
--weight_ncontact	0.01	score weight of each contact (pair of centroids within --max_pair_dist)
--weight_sasa	1152	Desired SASA used to weight dock scoring for sasa_priority score function
--weight_error	4	Standard deviation used to calculate the distribution of SASA weighting for sasa_priority score function
--hscore_files	'ilv_h'	*RPX* score files used in scoring for most protocols. defaults to pairs involving only ILV and only in helices. Can be only a path-suffix, which will be appended to --hscore_data_dir. Can be a list of files.
--hscore_data_dir	hscore	default path to search for hcores_files
--score_only_ss	EHL	only consider residues of the specified secondary structure type when scoring
--score_only_sspair	None	only consider pairs with the specified

		secondary structure types when scoring. may not work in all protocols
`--allowed_residues1`	None	allowed residues list for single component protocols or first component of 2+ component protocols or the monomeric plug for plug protocol. Takes either nothing (if you leave them out), a single file which applies to all the corresponding inputs, or a list of files which must have the same length as the list of inputs. The files themselves must contain a whitespace separated list of either numbers or ranges.
`--allowed_residues2`	None	allowed residues list for the second component for 2+ component protocols or the hole for the plug protocol.
`--allowed_residues3`	None	allowed residues for the third component for 3+ component protocols.
`--max_pair_dist`	8	maximum distance between centroids for a pair of residues to be considered interacting. In hierarchical protocols, coarser stages will add appropriate amounts to this distance
`--ignored_aas`	CGP	Amino acids to ignore in scoring
`--primary_iface_cut`	None	score cut for helix primary interface
`--score_self`	*False*	score each interface separately and dump in output pickle
Filtering and Clustering		
`--clashdis`	3.5	minimum distance allowed between heavy atoms
`--max_bb_redundancy`	3	minimum distance between outputs from a single docking run. is more-or-less a non-aligned backbone RMSD
`--max_cluster`	0	maximum number of results to cluster (filter redundancy via max_bb_redundancy) for each dock
`--filter_config`	None	Path to a *.yaml file containing the configurations for filters. NOTE: filters only work for cyclic, onecomp, and multicomp docking (ie. not for stacking or asymmetric docking).
Result		

--nout_debug	0	Specify number of pdb outputs for individual protocols to output for each search. This is not the preferred way to get pdb outputs, use --nout_top and --nout_each unless you have a reason not to
--nout_top	10	total number of top scoring output structures across all docks. only happens if --dump_pdbs is also specified
--nout_each	1	number of top scoring output structures for each individual dock. only happens if --dump_pdbs is also specified
--dump_pdbs	*False*	activate output of *.pdb files.
--output_asym_only	*False*	dump only asu to *.pdbs. Must be used with `--dump_pdbs` also activated.
--output_closest_subunits	*False*	for two component stuff, output subunit 2 most contacting subunit 1. Must be used with `--dump_pdbs` also activated.
--suppress_dump_results	*False*	suppress the output of results files
--iface_summary	min	method to use for summarizing multiple created interfaces into a single score. For example, a three component cage could have 3 interfaces A/B B/C and C/A, or a monomer-based cage plug will have an oligomer interface and an oligomer / cage interface. default is min. e.g. to take the overall score as the worst of the multiple interfaces
--output_prefix	rpxdock	output file prefix. will output pickles for a base ResPairScore plus --hierarchy_depth hier XMaps
--dont_store_body_in_results	*False*	reduce result output size and maybe runtime by not including structure information in results objects. will not be able to rescore or output pdbs from results objects.
--loglevel	INFO	select log level from CRITICAL, ERROR, WARNING, INFO or DEBUG
--use_orig_coords	*False*	remember and output the original sidechains from the input structures
--symframe_num_helix_repeats	10	number of helix repeat frames to dump
Compute		
--ncpu	All cores or cores	number of cpu cores to use. defaults to all cores or cores available according to slurm

	available	allocation
`--nthread`	Single thread and/or ncpu	number of threads to use in threaded protocols
`--nprocess`	`--ncpu`	number of processes to use for multiprocess protocols
`--trial_run`	*False*	reduce runtime by using minimal samples, smaller score files, etc.
`--debug`	*False*	Enable potentially expensive debugging checks

Available filter options

Table S1.2: SASA estimate filter parameters

Parameter	Required Setting	Default	Description
type	filter_sasa	NA	Must be the required setting exactly.
confidence	No	*False*	Should the filter remove failing docks from the result table?
min_sasa	Yes	750	Min interface size.
max_sasa	Yes	1500	Max interface size.
max_dist:	Yes	8	Maximum distance between residue pair centroids to count as part of the interface.
apply	No	*True*	Return sasa?
ncont	No	*False*	Return unique ncontact (instead of normal ncontact). Note if apply and ncont are both false this filter essentially returns ncontact.

Table S1.3: SScount filter parameters

Parameter	Required Setting	Default	Description
type	filter_sscount	NA	Must be the required setting exactly.
confidence	No	*False*	Should the filter remove failing docks from the result table?
min_helix_length	Yes	4	Min resis in helix to count as ss element.
min_sheet_length	Yes	3	Min resis in sheet to count as ss element.
min_loop_length	Yes	1	Min resis in loop to count as ss element.
max_dist	Yes	8	Maximum distance between residues to include in SS

			count.
min_element_resis	Yes	3	Min interface resis in ss_element to include in ss count.
sstype	Yes	“EH”	Types of secondary structure to include in count.
min_ss_count	Yes	3	If sscount_confidence set, minimum number of ss elements to pass the filter.
strict	No	*False*	Require that both pairs of residues in the interface are in an SS element meeting the set criteria. (This is not recommended for standard docking problems).

Table S1.4: Design construct renaming and input pdb files

Published name	Original name	Inputs1	Inputs2
T3-rpxdock-02	cage_twtls-02	C3_hfuse_twtls_003_4x_asu.pdb	
I3-rpxdock-71	I3-71_M3I	C3_1wa3_asu.pdb	
O43-rpxdock-15	cage_twtls-15	C4_171-7_asu.pdb	C3_hfuse_twtls_003_4x_asu.pdb
O43-rpxdock-HO11	O43-HO11	C4_171-7_asu.pdb	C3_HO10_asu.pdb
O43-rpxdock-EK1	O43-EK1	C4_tpr1C4-pm3_asu.pdb	C3_1na0HFuse_015_asu.pdb

Table S1.5. CryoEM data collection and refinement statistics

	O43-rpxdock-EK1
Microscope	**Glacios**
Voltage (kV)	**200**
Detector	**Gatan K3 Summit**
Recording mode	**Super-resolution**
Magnification	**45,000×**
Movie micrograph pixel size (Å)	**0.4425**
Dose rate (e−/Å2/s)	**10**
No. of frames per movie micrograph	**100**
Frame exposure time (ms)	**5**
Movie micrograph exposure time (s)	**5**
Total dose (e−/Å2)	**50**
Under focus range (μm)	**1.0 - 2.0**
Number of movie micrographs	**1,638**

	O43-rpxdock-EK1
Microscope	**Glacios**
Total number of picked particles	**771,134**
Particles in the final reconstruction	**130,778**
Map symmetry	**O**
Map resolution (GS-FSC)	**3.67 Å**
B-factor	**-238.2**
EMDB ID	EMD-29502

Supplementary Note 1.1: Protein Sequences

>T3-rpxdock-02

MGSVELLAVAALQELNIELARALLEAVARLQELNIDLVRKTSELTDEKTIREEIRKVKEESKRIVEEAEELIRLA
KLASEAIARMAEVAARGAPPELLIELLERLLKKAQEAGMSPEIIHLLLELALAIVEARGVPPEQLAEFAERLVE
ILREAGGSPELVFELLKRIMEIIERRGAPPELLIELLERLLELAREAGLSPEQITKLLILALVIVMRRGVPPEQL
AEFAEKLKEILREAGGSPELQRELKILIKLIEDLRGAGGSlehhhhhh

>I3-rpxdock-71

MKIEELFKKHKIVAVLRANSVEEAKKKALAVFLGGVHLIEITFTVPDADTVIKELSKLKEDGAIIGAGTVTSVE
QCRKAVESGAEFIVSPHLDEEISQFCKEKGVFYMPGVMTPTELVKAMKLGHTILKLFPGEVVGPQFVKAM
KGPFPNVKFVPTGGVNLDNVCEWFKAGVLAVGVGSALVKGEPVEVAEKAKAFVEKIRGCTElehhhhhh

>O43-rpxdock-15A

MGLEELLAKAAKDALSPDPEDLKEAVRLAEEVVRERPGSEAAKKALRIIQLAAELLKKSPDPEAIIAAARALL
KIAATTGDNEAAKQAIEAASKAAQLAEQRGDDELVCEALALLIAAQVLLLKQQGVPMLEVAIHVAETILQILQ
RLKRKGASEEVRKECLKRILREIAEALQRSGVPEEEIALIMLLIILLLMML

>O43-rpxdock-15B

MGSVELLAVAALQELNIELARALLEAVARLQELNIDLVRKTSELTDEKTIREEIRKVKEESKRIVEEAEELIRLA
KLASEAIARMAEVAARGAPPELLIELLERLLKKAQEAGMSPEIIHLLLELALAIVEARGVPPEQLAEFAERLVE
ILIRAGGSPELVFELLKRIMEIIERRGAPPELLIELLLNLLVLAVIAGLSPEQIHKLLEEALKIVERRGVPPEQLA
EFAEQLKLILKLAGGSPELQKELKKEIEEIEQRRGAGGSGGSWGGlehhhhhh

>O43-rpxdock-HO11A

MSEELIREAVEAAKRFEEARKRFEEAKERGDEKEAREALKEMLRAIEELARVATELNDSRLVKAAAKLAIKL
AEEALRFSDPEAAREAVRAALEIIRLMEKLAKKSNSEEIVELAARAAVELAAVAFQVGSSETARQAIETAARL
IALLVELLKRRGTSEDEIAEIVARLISEIIRILKEANAEYKFICKAVAIVVAAIVEALKRSGTSEDEIAEIVARVISE

VIRTLKESGSDYLIICVCVAIIVAEIVEALKRSGTSEDEIAEIVARVISEVIRTLKESGSSYEVIKECVQIIVLAIILA
LMKSGTEVEEILLILLRVKTEVRRTLKESGSWSG

>O43-rpxdock-HO11B

MLEMKLAVAELAAKSPDPELLKEAVKLAEEVVRERPGSEAAKKALEIIQEAAEKLKKSPDPEAIIAAARALLK
IAATTGDNEAAKQAIEAASKAAQLAEQRGDDELVCEALALLIAAQVLLLKQQGVPMLEVAIHVAETILQILQR
LKRKGASEEVRKECLKRILREIAEALQRSGVPEEEIALIMLLIILLLMMLGSWSGLEHHHHHH

>O43-rpxdock-EK1A

MALAYVMLGLLLSLLNRLSLAAEAYKKAIELDPNDALAWLLLGSVLEKLKRLDEAAEAYKKAIELKPNDASA
WKELGKVLEKLGRLDEAAEAYLIAIMLDPEDAEAAKELGKVLEKLGELEMAEEAYKLAIKLDPND

>O43-rpxdock-EK1B

MEEAELAYLLGELAYKLGEYRIAIRAYRIALKRDPNNAEAWYNLGNAYTKQGDYDEAIEYYLRALVLDPNNA
EAATNLGQAYMNQGDKDRAKLMLLLALKLDPNNDSARVILGVAKVGIEELAKLASQAQQEGDSEKQKAIE
LAAEAARVAQEVGDPELEKLALEAARRGDSEKAKAILLAAEAARVAKEVGDPELIKLALEAARRGDSEKAR
AILEAAERAREAKERGDPEQIKKARELAKRLEHHHHHH

References

Alford, Rebecca F., Andrew Leaver-Fay, Jeliazko R. Jeliazkov, Matthew J. O'Meara, Frank P. DiMaio, Hahnbeom Park, Maxim V. Shapovalov, et al. 2017. "The Rosetta All-Atom Energy Function for Macromolecular Modeling and Design." *Journal of Chemical Theory and Computation* 13 (6): 3031–48.

Asor, Roi, Lisa Selzer, Christopher John Schlicksup, Zhongchao Zhao, Adam Zlotnick, and Uri Raviv. 2019. "Assembly Reactions of Hepatitis B Capsid Protein into Capsid Nanoparticles Follow a Narrow Path through a Complex Reaction Landscape." *ACS Nano* 13 (7): 7610–26.

Bale, Jacob B., Shane Gonen, Yuxi Liu, William Sheffler, Daniel Ellis, Chantz Thomas, Duilio Cascio, et al. 2016. "Accurate Design of Megadalton-Scale Two-Component Icosahedral Protein Complexes." *Science* 353 (6297): 389–94.

Ben-Sasson, Ariel J., Joseph L. Watson, William Sheffler, Matthew Camp Johnson, Alice Bittleston, Logeshwaran Somasundaram, Justin Decarreau, et al. 2021. "Design of Biologically Active Binary Protein 2D Materials." *Nature* 589 (7842): 468–73.

Boyken, Scott E., Zibo Chen, Benjamin Groves, Robert A. Langan, Gustav Oberdorfer, Alex Ford, Jason M. Gilmore, et al. 2016. "De Novo Design of Protein Homo-Oligomers with Modular Hydrogen-Bond Network-Mediated Specificity." *Science* 352 (6286): 680–87.

Boyoglu-Barnum, Seyhan, Daniel Ellis, Rebecca A. Gillespie, Geoffrey B. Hutchinson, Young-Jun Park, Syed M. Moin, Oliver J. Acton, et al. 2021. "Quadrivalent Influenza Nanoparticle Vaccines Induce Broad Protection." *Nature* 592 (7855): 623–28.

Brouwer, Philip J. M., Aleksandar Antanasijevic, Zachary Berndsen, Anila Yasmeen, Brooke Fiala, Tom P. L. Bijl, Ilja Bontjer, et al. 2019. "Enhancing and Shaping the Immunogenicity of Native-like HIV-1 Envelope Trimers with a Two-Component Protein Nanoparticle." *Nature Communications* 10 (1). https://doi.org/10.1038/s41467-019-12080-1.

Brunette, T. J., Fabio Parmeggiani, Po-Ssu Huang, Gira Bhabha, Damian C. Ekiert, Susan E. Tsutakawa, Greg L. Hura, John A. Tainer, and David Baker. 2015. "Exploring the Repeat Protein Universe through Computational Protein Design." *Nature* 528 (7583): 580–84.

Chaudhury, Sidhartha, Sergey Lyskov, and Jeffrey J. Gray. 2010. "PyRosetta: A Script-Based Interface for Implementing Molecular Modeling Algorithms Using Rosetta." *Bioinformatics* 26 (5): 689–91.

Chauhan, Varun M., and Robert J. Pantazes. 2022. "MutDock: A Computational Docking Approach for Fixed-Backbone Protein Scaffold Design." *Frontiers in Molecular Biosciences* 9 (August): 933400.

Croll, Tristan Ian. 2018. "ISOLDE: A Physically Realistic Environment for Model Building into Low-Resolution Electron-Density Maps." *Acta Crystallographica. Section D, Structural Biology* 74 (Pt 6): 519–30.

Dauparas, J., I. Anishchenko, N. Bennett, H. Bai, R. J. Ragotte, L. F. Milles, B. I. M. Wicky, et al. 2022. "Robust Deep Learning-Based Protein Sequence Design Using ProteinMPNN." *Science*, September, eadd2187.

Divine, Robby, Ha V. Dang, George Ueda, Jorge A. Fallas, Ivan Vulovic, William Sheffler, Shally Saini, et al. 2021. "Designed Proteins Assemble Antibodies into Modular Nanocages." *Science* 372 (6537). https://doi.org/10.1126/science.abd9994.

Durham, Elizabeth, Brent Dorr, Nils Woetzel, René Staritzbichler, and Jens Meiler. 2009. "Solvent Accessible Surface Area Approximations for Rapid and Accurate Protein Structure Prediction." *Journal of Molecular Modeling* 15 (9): 1093–1108.

Fallas, Jorge A., George Ueda, William Sheffler, Vanessa Nguyen, Dan E. McNamara, Banumathi Sankaran, Jose Henrique Pereira, et al. 2017. "Computational Design of Self-Assembling Cyclic Protein Homo-Oligomers." *Nature Chemistry* 9 (4): 353–60.

Gerben, Stacey R., Andrew J. Borst, Derrick R. Hicks, Isabelle Moczygemba, David Feldman, Brian Coventry, Wei Yang, et al. 2023. “Design of Diverse Asymmetric Pockets in De Novo Homo-Oligomeric Proteins.” *Biochemistry* 62 (2): 358–68.

Golub, Eyal, Rohit H. Subramanian, Julian Esselborn, Robert G. Alberstein, Jake B. Bailey, Jerika A. Chiong, Xiaodong Yan, Timothy Booth, Timothy S. Baker, and F. Akif Tezcan. 2020. “Constructing Protein Polyhedra via Orthogonal Chemical Interactions.” *Nature* 578 (7793): 172–76.

Gonen, Shane, Frank DiMaio, Tamir Gonen, and David Baker. 2015. “Design of Ordered Two-Dimensional Arrays Mediated by Noncovalent Protein-Protein Interfaces.” *Science* 348 (6241): 1365–68.

Grigoryan, Gevorg, and William F. Degrado. 2011. “Probing Designability via a Generalized Model of Helical Bundle Geometry.” *Journal of Molecular Biology* 405 (4): 1079–1100.

Gu, Yan, Yong He, Kayvon Fatahalian, and Guy Blelloch. 2013. “Efficient BVH Construction via Approximate Agglomerative Clustering.” In *Proceedings of the 5th High-Performance Graphics Conference*, 81–88. HPG ’13. New York, NY, USA: Association for Computing Machinery.

Hsia, Yang, Jacob B. Bale, Shane Gonen, Dan Shi, William Sheffler, Kimberly K. Fong, Una Nattermann, et al. 2016. “Design of a Hyperstable 60-Subunit Protein Icosahedron.” *Nature*. https://doi.org/10.1038/nature18010.

Hsia, Yang, Rubul Mout, William Sheffler, Natasha I. Edman, Ivan Vulovic, Young-Jun Park, Rachel L. Redler, et al. 2021. “Design of Multi-Scale Protein Complexes by Hierarchical Building Block Fusion.” *Nature Communications* 12 (1): 2294.

Huang, Po-Ssu, Gustav Oberdorfer, Chunfu Xu, Xue Y. Pei, Brent L. Nannenga, Joseph M. Rogers, Frank DiMaio, Tamir Gonen, Ben Luisi, and David Baker. 2014. “High Thermodynamic Stability of Parametrically Designed Helical Bundles.” *Science* 346 (6208): 481–85.

Ibaraki, Toshihide. 1976. “Theoretical Comparisons of Search Strategies in Branch-and-Bound Algorithms.” *International Journal of Computer & Information Sciences* 5 (4): 315–44.

Kabsch, W., and C. Sander. 1983. “Dictionary of Protein Secondary Structure: Pattern Recognition of Hydrogen-Bonded and Geometrical Features.” *Biopolymers* 22 (12): 2577–2637.

Kakkis, Albert, Derek Gagnon, Julian Esselborn, R. David Britt, and F. Akif Tezcan. 2020. “Metal-Templated Design of Chemically Switchable Protein Assemblies with High-Affinity Coordination Sites.” *Angewandte Chemie* 59 (49): 21940–44.

Kidmose, Rune Thomas, Jonathan Juhl, Poul Nissen, Thomas Boesen, Jesper Lykkegaard Karlsen, and Bjørn Panyella Pedersen. 2019. “Namdinator - Automatic Molecular Dynamics Flexible Fitting of Structural Models into Cryo-EM and Crystallography Experimental Maps.” *IUCrJ* 6 (Pt 4): 526–31.

King, Neil P., Jacob B. Bale, William Sheffler, Dan E. McNamara, Shane Gonen, Tamir Gonen, Todd O. Yeates, and David Baker. 2014. “Accurate Design of Co-Assembling Multi-Component Protein Nanomaterials.” *Nature* 510 (7503): 103–8.

King, Neil P., William Sheffler, Michael R. Sawaya, Breanna S. Vollmar, John P. Sumida, Ingemar André, Tamir Gonen, Todd O. Yeates, and David Baker. 2012. “Computational Design of Self-Assembling Protein Nanomaterials with Atomic Level Accuracy.” *Science* 336 (6085): 1171–74.

Lai, Yen-Ting, Neil P. King, and Todd O. Yeates. 2012. “Principles for Designing Ordered Protein Assemblies.” *Trends in Cell Biology* 22 (12): 653–61.

Laniado, Joshua, Kyle Meador, and Todd O. Yeates. 2021. “A Fragment-Based Protein Interface Design Algorithm for Symmetric Assemblies.” *Protein Engineering, Design & Selection: PEDS* 34: 1–34.

Larsson, Thomas, and Tomas Akenine-Möller. 2006. “A Dynamic Bounding Volume Hierarchy

for Generalized Collision Detection." *Computers & Graphics* 30 (3): 450–59.
Leaver-Fay, Andrew, Michael Tyka, Steven M. Lewis, Oliver F. Lange, James Thompson, Ron Jacak, Kristian Kaufman, et al. 2011. "ROSETTA3: An Object-Oriented Software Suite for the Simulation and Design of Macromolecules." *Methods in Enzymology* 487: 545–74.
Leman, Julia Koehler, Brian D. Weitzner, Steven M. Lewis, Jared Adolf-Bryfogle, Nawsad Alam, Rebecca F. Alford, Melanie Aprahamian, et al. 2020. "Macromolecular Modeling and Design in Rosetta: Recent Methods and Frameworks." *Nature Methods* 17 (7): 665–80.
Lin, Yu-Ru, Nobuyasu Koga, Sergey M. Vorobiev, and David Baker. 2017. "Cyclic Oligomer Design with de Novo αβ-Proteins." *Protein Science: A Publication of the Protein Society* 26 (11): 2187–94.
Li, Zhe, Shunzhi Wang, Una Nattermann, Asim K. Bera, Andrew J. Borst, Matthew J. Bick, Erin C. Yang, et al. 2022. "Accurate Computational Design of 3D Protein Crystals." *bioRxiv*. https://doi.org/10.1101/2022.11.18.517014.
Lutz, Isaac D., Shunzhi Wang, Christoffer Norn, Andrew J. Borst, Yan Ting Zhao, Annie Dosey, Longxing Cao, et al. 2022. "Top-down Design of Protein Nanomaterials with Reinforcement Learning." *bioRxiv*. https://doi.org/10.1101/2022.09.25.509419.
Lyskov, Sergey, and Jeffrey J. Gray. 2008. "The RosettaDock Server for Local Protein-Protein Docking." *Nucleic Acids Research* 36 (Web Server issue): W233–38.
Marcandalli, Jessica, Brooke Fiala, Sebastian Ols, Michela Perotti, Willem de van der Schueren, Joost Snijder, Edgar Hodge, et al. 2019. "Induction of Potent Neutralizing Antibody Responses by a Designed Protein Nanoparticle Vaccine for Respiratory Syncytial Virus." *Cell* 176 (6): 1420–31.e17.
Morrison, David R., Sheldon H. Jacobson, Jason J. Sauppe, and Edward C. Sewell. 2016. "Branch-and-Bound Algorithms: A Survey of Recent Advances in Searching, Branching, and Pruning." *Discrete Optimization* 19 (February): 79–102.
Padhorny, Dzmitry, Andrey Kazennov, Brandon S. Zerbe, Kathryn A. Porter, Bing Xia, Scott E. Mottarella, Yaroslav Kholodov, David W. Ritchie, Sandor Vajda, and Dima Kozakov. 2016. "Protein-Protein Docking by Fast Generalized Fourier Transforms on 5D Rotational Manifolds." *Proceedings of the National Academy of Sciences of the United States of America* 113 (30): E4286–93.
Padilla, J. E., C. Colovos, and T. O. Yeates. 2001. "Nanohedra: Using Symmetry to Design Self Assembling Protein Cages, Layers, Crystals, and Filaments." *Proceedings of the National Academy of Sciences of the United States of America* 98 (5): 2217–21.
Park, Taeyong, Minkyung Baek, Hasup Lee, and Chaok Seok. 2019. "GalaxyTongDock: Symmetric and Asymmetric Ab Initio Protein-Protein Docking Web Server with Improved Energy Parameters." *Journal of Computational Chemistry* 40 (27): 2413–17.
Pettersen, Eric F., Thomas D. Goddard, Conrad C. Huang, Elaine C. Meng, Gregory S. Couch, Tristan I. Croll, John H. Morris, and Thomas E. Ferrin. 2021. "UCSF ChimeraX: Structure Visualization for Researchers, Educators, and Developers." *Protein Science: A Publication of the Protein Society* 30 (1): 70–82.
Punjani, Ali, John L. Rubinstein, David J. Fleet, and Marcus A. Brubaker. 2017. "cryoSPARC: Algorithms for Rapid Unsupervised Cryo-EM Structure Determination." *Nature Methods* 14 (3): 290–96.
Rhys, Guto G., Christopher W. Wood, Joseph L. Beesley, Nathan R. Zaccai, Antony J. Burton, R. Leo Brady, Andrew R. Thomson, and Derek N. Woolfson. 2019. "Navigating the Structural Landscape of De Novo α-Helical Bundles." *Journal of the American Chemical Society* 141 (22): 8787–97.
Sahasrabuddhe, Aniruddha, Yang Hsia, Florian Busch, William Sheffler, Neil P. King, David Baker, and Vicki H. Wysocki. 2018. "Confirmation of Intersubunit Connectivity and Topology of Designed Protein Complexes by Native MS." *Proceedings of the National Academy of Sciences of the United States of America* 115 (6): 1268–73.

Schneidman-Duhovny, Dina, Yuval Inbar, Ruth Nussinov, and Haim J. Wolfson. 2005. “PatchDock and SymmDock: Servers for Rigid and Symmetric Docking.” *Nucleic Acids Research* 33 (Web Server issue): W363–67.

Shen, Hao, Jorge A. Fallas, Eric Lynch, William Sheffler, Bradley Parry, Nicholas Jannetty, Justin Decarreau, et al. 2018. “De Novo Design of Self-Assembling Helical Protein Filaments.” *Science* 362 (6415): 705–9.

Touw, Wouter G., Coos Baakman, Jon Black, Tim A. H. te Beek, E. Krieger, Robbie P. Joosten, and Gert Vriend. 2015. “A Series of PDB-Related Databanks for Everyday Needs.” *Nucleic Acids Research*. https://doi.org/10.1093/nar/gku1028.

Ueda, George, Aleksandar Antanasijevic, Jorge A. Fallas, William Sheffler, Jeffrey Copps, Daniel Ellis, Geoffrey B. Hutchinson, et al. 2020. “Tailored Design of Protein Nanoparticle Scaffolds for Multivalent Presentation of Viral Glycoprotein Antigens.” *eLife* 9 (August). https://doi.org/10.7554/eLife.57659.

Walls, Alexandra C., Brooke Fiala, Alexandra Schäfer, Samuel Wrenn, Minh N. Pham, Michael Murphy, Longping V. Tse, et al. 2020. “Elicitation of Potent Neutralizing Antibody Responses by Designed Protein Nanoparticle Vaccines for SARS-CoV-2.” *Cell* 183 (5): 1367–82.e17.

Wang, Jing Yang (john), Alena Khmelinskaia, William Sheffler, Marcos C. Miranda, Aleksandar Antanasijevic, Andrew J. Borst, Susana Vazquez Torres, et al. 2022. “Improving the Secretion of Designed Protein Assemblies through Negative Design of Cryptic Transmembrane Domains.” *bioRxiv*. https://doi.org/10.1101/2022.08.04.502842.

Wargacki, Adam J., Tobias P. Wörner, Michiel van de Waterbeemd, Daniel Ellis, Albert J. R. Heck, and Neil P. King. 2021. “Complete and Cooperative in Vitro Assembly of Computationally Designed Self-Assembling Protein Nanomaterials.” *Nature Communications* 12 (1): 883.

Wicky, B. I. M., L. F. Milles, A. Courbet, R. J. Ragotte, J. Dauparas, E. Kinfu, S. Tipps, et al. 2022. “Hallucinating Symmetric Protein Assemblies.” *Science*, September, eadd1964.

Woolfson, Derek N. 2005. “The Design of Coiled-Coil Structures and Assemblies.” *Advances in Protein Chemistry* 70: 79–112.

Yan, Yumeng, Huanyu Tao, and Sheng-You Huang. 2018. “HSYMDOCK: A Docking Web Server for Predicting the Structure of Protein Homo-Oligomers with Cn or Dn Symmetry.” *Nucleic Acids Research* 46 (W1): W423–31.

Chapter 2: Computational design of non-porous, pH-responsive antibody nanoparticles

Adapted from:
Yang, Erin C., Robby Divine, Marcos C. Miranda, Andrew J. Borst, Will Sheffler, Jason Z. Zhang, Justin Decarreau, et al. 2023. “Computational Design of Non-Porous, pH-Responsive Antibody Nanoparticles.” *bioRxiv*. https://doi.org/10.1101/2023.04.17.537263.

Designing nanoparticles made of protein,
Forge a path to targeted delivery, it should seem.
We start with an octahedral, quite porous in kind,
Leaving an unoccupied symmetry axis, a pore that we find.

We fill the pore with proteins, distinct and fine,
Tetramer, antibody, and trimer, all in line.
Assembling cooperatively, indeed they do,
Cryo-EM confirmed the design to be true.
Packaging cargo and targeting cells too,
And disassembling at pH 5.9-6.7 range, together all new.

Each component performs its own role,
Occupying a distinct symmetry axis and achieving different goals.
Antibodies direct the nanoparticle’s way,
Trimers disassemble to release the array,
Tetramers’ interface makes the assembly stay,
Unlocking targeted delivery potential, this system conveys.

Abstract

Programming protein nanomaterials to respond to changes in environmental conditions is a current challenge for protein design and important for targeted delivery of biologics. We describe the design of octahedral non-porous nanoparticles with the three symmetry axes (four-fold, three-fold, and two-fold) occupied by three distinct protein homooligomers: a *de novo* designed tetramer, an antibody of interest, and a designed trimer programmed to disassemble below a tunable pH transition point. The nanoparticles assemble cooperatively from independently purified components, and a cryo-EM density map reveals that the structure is very close to the computational design model. The designed nanoparticles can package a variety of molecular payloads, are endocytosed following antibody-mediated targeting of cell surface receptors, and undergo tunable pH-dependent disassembly at pH values ranging between to 5.9-6.7. To our knowledge, these are the first designed nanoparticles with more than two structural components and with finely tunable environmental sensitivity, and they provide new routes to antibody-directed targeted delivery.

Introduction

There is considerable interest in tailoring nanoparticle platforms for targeted delivery of therapeutic molecules. Effective targeted delivery nanoparticle platforms require assembly and encapsulation of molecules outside the target cell, followed by target recognition, triggered nanoparticle disassembly, and controlled cargo release once inside the cell (Edwardson, Mori, and Hilvert 2018; Mitchell et al. 2020; Banskota et al. 2022; D. Wang, Tai, and Gao 2019; Hou et al. 2021; Douglas and Young 2006; Seo et al. 2021; Van de Steen et al. 2021; Azuma, Edwardson, and Hilvert 2018). Cellular uptake of extracellular molecules such as protein-based nanoparticles via endocytosis involves traversal of various membrane-bound organelles, including the low-pH endosome and lysosome (Dimitrov 2004; Martens et al. 2014; Lönn et al. 2016; Czapar and Steinmetz 2017). While a number of self-assembling protein nanoparticles with customized structures have been designed, they are composed of just one or two unique, static building blocks and efforts to adapt them for cargo packaging and delivery applications are still in their infancy (Cannon et al. 2020; Lai, King, and Yeates 2012; King et al. 2014; Butterfield et al. 2017; Votteler et al. 2016; Tetter et al. 2021; Levasseur et al. 2021; Edwardson, Mori, and Hilvert 2018). Particularly attractive for delivery applications are antibody-incorporating nanoparticles, where one component is a designed homooligomer that, upon mixing with any antibody of interest, generates a bounded, multivalent, polyhedral assembly (Divine et al. 2021) (**Figure 2.1A**). While such antibody nanoparticles can enhance cell signaling activity and have demonstrated a broad utility for selectively targeting cell surface receptors, they are quite porous, which complicates the packaging and retention of molecular cargoes.

Results

Computational design

To enable packaging and pH-dependent release of molecular cargoes, we sought to close off the apertures in the previously designed antibody nanoparticles with the addition of computationally designed pH-responsive proteins. We focused on octahedral antibody nanoparticles (O42.1) constructed from a C4-symmetric designed tetramer and C2-symmetric IgG dimers, which are aligned along the C4 and C2 symmetry axes of the octahedral architecture (**Figure 2.1A**). On the open C3 axis, we aimed to incorporate a designed pH-dependent C3 trimer (Boyken et al. 2019) and tune it to disassemble at pH values corresponding to the native environment of the endosome (**Figure 2.1B**). We reasoned that such three-component nanoparticles could 1) selectively enter target cells, 2) encapsulate molecular cargoes without leakage, and 3) disassemble in the acidic environment of the endosome.

The previously designed pH-dependent trimer is much smaller than the aperture along the C3 axis of the octahedral nanoparticle (**Figure 2.1B**), and hence filling the C3 axis of the octahedral nanoparticles with the pH-dependent trimer required extending the backbone of the trimer such that it could contact and make shape-complementary interactions with the designed tetramer. To

enable this, we combined helical fusion (Hsia et al. 2021; Padilla, Colovos, and Yeates 2001) and protein docking (Sheffler et al. 2022) approaches into a single design pipeline. We extended the pH trimer by fusing combinations of helical repeat protein building blocks onto each subunit to generate a total of 80,000 distinct C3 fusions with helical repeats of variable geometry extending outwards from the C3 axis (**Figure 2.1B**, see methods). The resultant diverse set of C3 building blocks were docked into the three-fold pore by aligning both C3 axes and sampling translational and rotational degrees of freedom along this axis (Sheffler et al. 2022) (**Figure 2.1C**). Variable-length truncations of the terminal helices of the repeat protein arms were evaluated to optimize the docked interface between the new C3 building block and the C4 subunits of the octahedral assembly (see methods). The resulting “plugged” octahedral assembly (O432) contains twelve IgG1-Fc domains along the octahedral two-fold axes, six tetramers along the octahedral four-fold axes, and eight trimeric plugs along the octahedral three-fold axes (**Figure 2.1D**).

The newly generated interfaces between the pH trimer fusions and the octahedral assembly were evaluated for designability using a combination of the residue pair transform (rpx) score—a prediction of interaction energy following sequence design (Fallas et al. 2017; Sheffler et al. 2022)—and overall shape complementarity at the interface (Lawrence and Colman 1993). For 6000 docks that were predicted to have high designability and shape complementarity, the amino acid sequence at the newly formed fusion junctions and at the interface between the trimer and antibody nanoparticle were optimized using Rosetta sequence design calculations (Leman et al. 2020). This design step introduced mutations on both the trimer and the tetramer subunits. Designed interfaces were evaluated for secondary structure contacts and chemical complementarity, and 45 designed trimeric plug and nanoparticle tetramer pairs were selected for experimental characterization. The designed interfaces intentionally spanned a broad range of buried solvent accessible surface area (SASA; 1500–3000 $Å^2$), as the optimal interface size for this type of symmetric assembly was unknown.

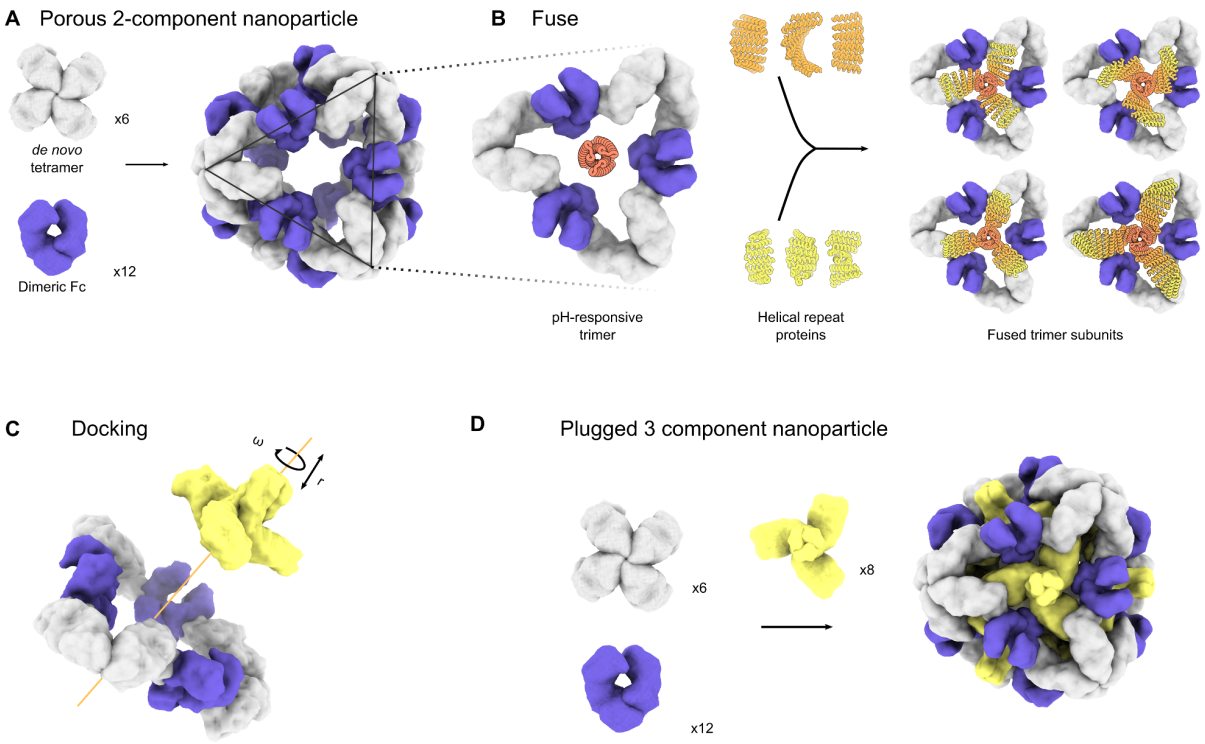


Figure 2.1: Design of symmetry-matched plugs to fill empty symmetry axes in protein nanoparticles.
A. 6 *de novo* tetramers (gray) and 12 dimeric Fc domains (purple) assemble into a porous octahedral O42 nanoparticle. The tetramers are aligned along the 4-fold symmetry axis and the Fc domains along the 2-fold symmetry axis. **B.** Combinations of helical repeat proteins were fused to each other and to the pH trimer subunit at regions of high backbone overlap between pairs of helices to generate fused trimer subunits large enough to fully occupy the void along the 3-fold axis in the original nanoparticle. **C.** These pH-dependent trimers were then docked into the nanoparticle by rotating and translating along its 3-fold axis. **D.** The resulting three-component nanoparticle has 8 new trimeric subunits (yellow) which occupy the three-fold symmetry axis of the octahedral architecture.

Structural Characterization

Designed trimers and tetramers with C-terminal 6×-histidine tags on the tetramer were expressed bicistronically in *E. coli* and subject to immobilized metal affinity chromatography (IMAC) purification. SDS-polyacrylamide gel electrophoresis (SDS-PAGE) identified 16 of 45 designs where the designed trimer co-eluted with the tetramer. A sfGFP-Fc fusion protein (Divine et al. 2021) was added to the clarified lysates of these co-expressed trimers and tetramers, and the resulting assemblies were purified via IMAC. SDS-PAGE and native PAGE were used to determine which designs formed three-component assemblies (**Figure S2.1A-B**). For the 5 out of 16 designs in which all three components associated, genes expressing the trimer and tetramer were subcloned into separate expression vectors, and the independently expressed oligomers were purified separately by size exclusion chromatography (SEC) on a Superdex 200 10/300 GL column (**Figure S2.1C**). Three-component assemblies were prepared

by mixing together the three purified proteins—trimeric plug, tetramer, and the Fc of human IgG1—in a 1:1:1 stoichiometric ratio, followed by overnight incubation at 25°C and purification by SEC using a Superdex 6 10/300 GL column (**Figure S2.1D**). Due to tetramer insolubility, insufficient material was produced to prepare a three-component assembly reaction for one design, O432-43.

Since the addition of the trimeric plug should not affect the diameter of the octahedral antibody nanoparticle, we next compared the SEC elution volumes of all three-component mixtures to the elution profile of the original O42.1 antibody nanoparticle (Divine et al. 2021). Mixing all three components in a 1:1:1 stoichiometric ratio yielded elution peaks in the void volume with shoulders between 9-11 mL (**Figure S2.1D**), where the shoulder peaks matched the elution profile of the original O42.1 antibody nanoparticle (Divine et al. 2021) (**Figure S2.1E**). Non-reducing SDS-PAGE of both the void and shoulder peaks showed that only one assembly (O432-17) contained all three protein components in the elution fraction (**Figure S2.1F-G**). Optimization of the stoichiometric ratios of each protomer present during *in vitro* assembly resulted in the O432-17 peak shifting from the void volume toward the expected elution volume during SEC, which matched the profile of the original O42.1 antibody nanoparticle (**Figure 2.2B**). The optimal assembly ratio per protomer of purified trimeric plug, tetramer, and Fc was determined to be 1.1:1.1:1. Dynamic light scattering (DLS) of the main O432 peak indicated a hydrodynamic diameter of 34 nm and a polydispersity index (PDI) of 0.05 (**Figure 2.2C**) and negative-stain electron microscopy (NS-EM) revealed monodisperse nanoparticles (**Figure 2.2D**). Two-dimensional class averages of negatively stained micrographs revealed plug-like density in the 3-fold views compared to the original two-component antibody nanoparticle.

We found that assembly of the designed nanoparticle was cooperative and required all three components: mixing any two of the three O432-17 components stoichiometrically did not result in nanoparticles (**Figure S2.2A**). As expected, mixing the trimeric plug and Fc resulted in no assembly, as there is no designed interface between those two components. Mixing the tetramer with Fc resulted in visible aggregate and partially formed nanoparticles (**Figure S2.2B**), likely due to the formation of off-target interactions by the newly designed hydrophobic plug interface. Mixing the tetramer and trimeric plug did not result in association even at high concentration (400 μM per monomer) (**Figure S2.2C**). This cooperativity simplifies preparation of the three component nanoparticles as it prevents incomplete assembly of nanoparticles containing two out of the three components, and thus eliminates the need for additional purification steps to separate these species from the intended three-component assembly.

For downstream delivery applications, we tested whether the O432-17 nanoparticle would assemble when the designed trimer and tetramer were co-incubated with full-length IgG antibodies containing both Fc and Fab domains (**Figure 2.2B-D**). The O432-17 design eluted in the void volume, owing to the increased diameter from the additional Fab domains (**Figure 2.2B**). DLS of this void volume peak revealed a monodisperse hydrodynamic diameter of 40 nm and PDI of 0.07, in line with the expected diameter of the IgG-containing O432-17 assembly (**Figure 2.2C**). NS-EM micrographs and two-dimensional class averages of this peak fraction exhibited plug-like density in three-fold views following 2D classification as well as Fab-like

density in two-fold, three-fold, and four-fold views (**Figure 2.2D**). Despite the clear presence of additional density corresponding to the IgG Fab domains, due to the inherent marked flexibility between the IgG Fc and Fab domains (Roux, Strelets, and Michaelsen 1997), the Fab domains are largely averaged out in the 2D averages of this complex (**Figure 2.2D**).

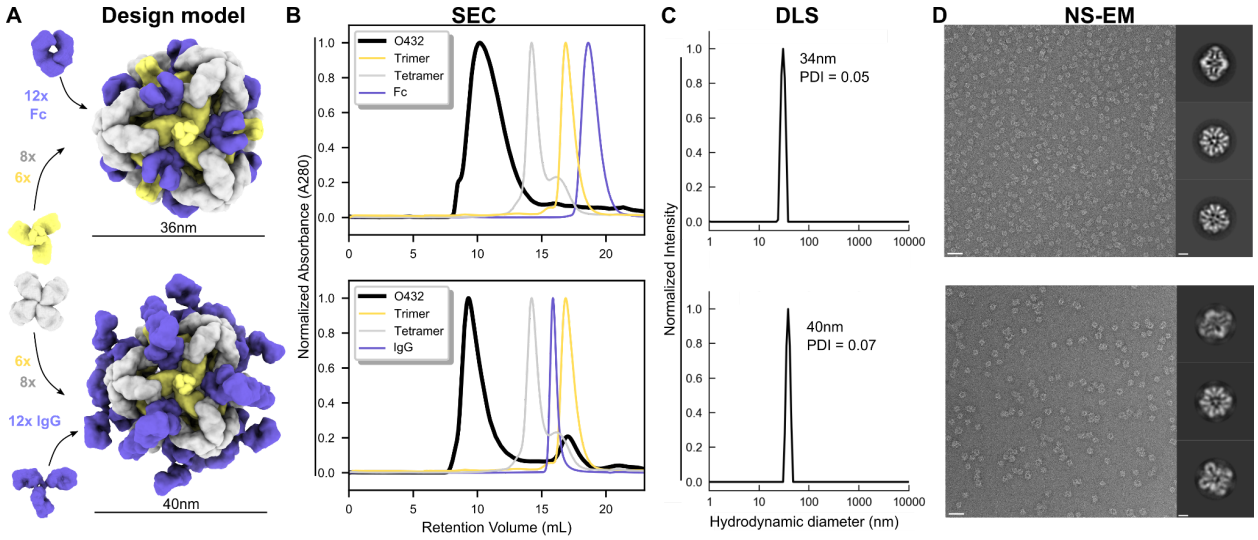


Figure 2.2: Mixing independently purified components enables stable, efficient assembly with both Fc and IgG.
A. Design models with Fc or IgG (purple), designed nanoparticle-forming tetramers (gray), and pH-dependent plug-forming trimers (yellow). **B.** Overlay of representative SEC traces of full assembly formed by mixing designed tetramers, trimers, and Fc or IgG (black) with those of the single components in gray (tetramer), yellow (trimeric plug), or purple (Fc or IgG). **C.** Representative DLS of fractions collected from the O432 assembly peak shows average hydrodynamic diameters of 34 nm (polydispersity index or PDI: 0.05) and 40 nm (PDI: 0.05) for the O432 assemblies with Fc and full-length IgG, respectively. **D.** Negatively stained electron micrographs with reference-free 2D class averages along each axis of symmetry in inset; electron microscopy images were collected prior to SEC purification. Scale bars, 100 nm and 10 nm for the micrograph and 2D averages, respectively.

Cryo-EM structural characterization

We next sought to determine the completeness of the three-component Fc-containing octahedral nanoparticle assembly and the accuracy of the newly designed interface between the trimer and tetramer using single-particle cryo-EM (**Figure 2.3A-D, Figure S2.3E**). Following data collection and preprocessing of raw micrographs, a subset of the best selected two-dimensional averages with fully assembled O432-17 nanoparticles (**Figure S2.3A-B**) were used to generate an *ab initio* 3D reconstruction in the absence of applied symmetry (**Figure S2.3C**). This initial map and the corresponding set of particles were next subclassified into four distinct classes following 3D heterogeneous refinement (**Figure S2.3D**), also in the absence of any applied symmetry operator. All four subclasses revealed the presence of fully plugged antibody nanoparticles, demonstrating complete O432 assembly for this system at the concentrations used. We observe density occupying all symmetry axes of the octahedral

architecture with six designed tetramers at the four-fold octahedral symmetry axes forming interfaces with eight designed trimers at the three-fold symmetry axes and twelve dimeric Fc fragments at the two-fold symmetry axis. Upon confirming the completeness of this nanoparticle assembly, a subsequent 3D refinement containing particles from all four of the aforementioned classes was generated after applying octahedral symmetry, resulting in a final map with an estimated global resolution of 7 Å (**Figure S2.3C**).

We estimated the accuracy of our design protocol by relaxing the original computational O432-17 design model in the experimentally determined cryo-EM map. The relaxed model of O432-17 containing the plug, tetramer, and Fc is in close agreement with the original design (**Figure 2.3D**). We observed cryo-EM density corresponding to the helices of the tetramers at the four-fold vertices, where each helical repeat extends from the four-fold axis along the edges of the octahedral architecture to bind Fc. The cryo-EM density also shows the helices of the trimeric components along the faces of the octahedral architecture, where the helical repeats extend to form an interface with the tetramers at the edges. The addition of the trimeric plug reduces porosity of the antibody nanoparticle as intended: the largest diameter pore is 3 nm compared to 13 nm in the original O42.1 nanoparticle (Divine et al. 2021). While there is some structural distortion in the helical repeat regions of the trimeric plug and tetramer, suggesting slight structural flexibility of the helical repeat domains or fusions between helical repeat regions, the helices forming the trimer and tetramer interface are largely consistent between the design and relaxed model (**Figure S2.3F**). The Fc domain exhibited very little secondary structure deviation between the design model and relaxed model.

Comparing the relaxed model of our plugged O432-17 nanoparticle to that of the original O42.1 antibody nanoparticle design model (Divine et al. 2021) showed that the relaxed cryo-EM model is significantly closer to the original O42.1 design model than the previously determined O42.1 cryo-EM structure. The Cα RMSDs of the asymmetric units are 1.6 Å between the O432-17 relaxed model and its design model, 1.9 Å between the O432-17 design model and O42.1 relaxed model, and 4.2 Å between the the O42.1 relaxed model and its design model. These results suggest that the addition of the trimeric plug buttresses the tetramer into a conformation that more closely matches the original design model, and may also reduce the overall flexibility of the system, as compared to the original O42.1 design (**Figure S2.4A-C**).

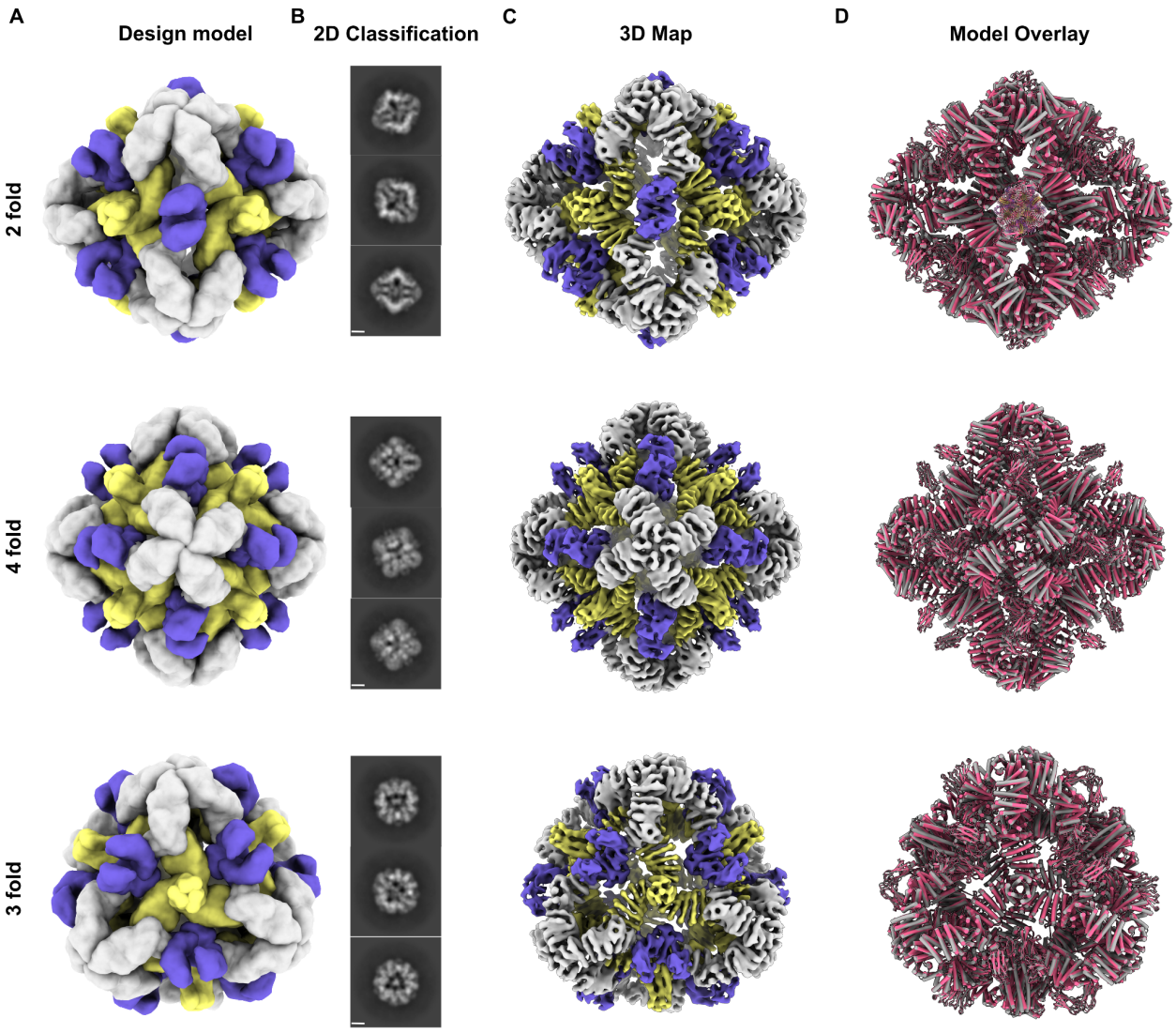


Figure 2.3: Cryo-EM analysis of 72-subunit nanoparticles composed of three distinct structural components.
A. The O432 assembly with Fc before SEC purification was characterized by cryogenic electron microscopy. Computational design models viewed along each axis of symmetry of the octahedral architecture are shown. **B.** Representative 2D class averages along each axis of symmetry. Scale bar, 10 nm. **C.** 7 Å 3D EM density map reconstructed from the collected dataset. **D.** Overlay of the design model (gray) and the design model relaxed into the 3D reconstruction (pink) showing high agreement.

Design of O432-17 cargo loading variants

Encouraged by these structural results, we next set out to redesign the nanoparticles to have either highly positively or highly negatively charged interiors to enable packaging of molecular cargoes via electrostatic interactions (Bale et al. 2016; Butterfield et al. 2017; Edwardson, Levasseur, and Hilvert 2021). We focused our redesign on selected interior surface residues of

the trimeric plug, favorably weighting mutations to amino acids with the desired charge (or no charge), and unfavorably weighting mutations to amino acids with the undesired charge. We screened for packaging of positively charged GFP (pos36GFP) (Zuris et al. 2015) by mixing negatively charged trimeric plug variants exhibiting different magnitudes of interior surface charge with tetramer, Fc, and pos36GFP (**Figure 2.4A**). One variant interacted with supercharged GFP in these conditions, as shown by co-elution of the 488 nm signal and the 280 nm signal during SEC (**Figure 2.4B**), and will be referred to subsequently as O432-17(-). Including 1 M NaCl in the buffers used during packaging and SEC prevented packaging (**Figure 2.4C**), suggesting that GFP packaging was largely driven by electrostatic interactions between the cargo and nanoparticle interior.

Trimeric plug variants of varying interior positive charge magnitudes were then screened for packaging of a 154-nt prime editing guide RNA (pegRNA) cargo (Hsu et al. 2021; Anzalone et al. 2019; Nelson et al. 2022). We mixed the positively charged trimer variants, tetramer, and the α-EGFR antibody Cetuximab (CTX) with the pegRNA (**Figure 2.4A**), carried out non-denaturing electrophoresis, and stained with SYBR Gold (RNA) and Coomassie (protein) with and without Benzonase treatment. For one variant, which will be referred to as O432-17(+), comigration of nucleic acid and protein with and without Benzonase treatment indicated successful packaging of nucleic acid (**Figure 2.4D-E**). Excess nucleic acid that did not comigrate with the protein was degraded in the Benzonase-treated sample, demonstrating nanoparticle protection of the nucleic acid that comigrated with O432-17(+). NS-EM micrographs and two-dimensional class averages confirmed three-component nanoparticle assembly of O432-17(+) in the presence of RNA (**Figure S2.5A-B**). Thus, our designed three-component nanoparticles efficiently encapsulate and protect molecular cargoes.

Precise engineering of tunable pH-responsiveness in O432-17 nanoparticles

We next explored whether O432-17(-) and O432-17(+) disassembled below pH 6.0 using a medium-throughput, low-concentration flow cytometry-based assay. We conjugated AlexaFluor647 (AF647) via maleimide chemistry to a C-terminal cysteine on the O432-17(-) trimeric plug variant. We assembled O432-17 nanoparticles with the tetramer, AF647-conjugated trimeric plug, and a 50/50 mixture of α-myc mouse mAb (Cell Signaling Technologies 9B11) and sfGFP-Fc, the latter of which acted as a marker for a non-plug nanoparticle component. Assembled nanoparticles were incubated with 3 μm polystyrene beads coated with biotinylated myc-tag for one hour to allow efficient loading of the α-myc containing nanoparticles onto the beads. The beads were then split into an excess volume of titrated citrate-phosphate buffers ranging from pH 4.2 to pH 7.5 and were incubated at 25°C for 30 minutes. After 30 minutes, the beads were resuspended and brought back to pH 7.5 to normalize sfGFP fluorescence, and then analyzed for AF647 and sfGFP signal by flow cytometry (**Figure 2.4F**). After isolating singlet beads, we normalized the mean fluorescence intensity of the AF647 on the trimer and sfGFP on the Fc at each pH (**Figure S2.6A-D**) to the minimum and maximum values observed across the pH titration. The O432-17(-) plug design (named O432-17(-)_2HIS), which was based on a pH-responsive helical bundle with two

histidine hydrogen bond networks that dissociated into monomers at pH 4.9 (Boyken et al. 2019), did not show reduction in AF647 or sfGFP fluorescence until below pH 5.1, similar to a negative control trimer variant with all histidines substituted with asparagine (named O432-17(-)_0HIS; **Figure S2.6E-F**). This result indicated that the sensitivity of the trimeric plug to pH was dampened relative to the original pH-responsive helical bundle, likely due to stabilization afforded by nanoparticle assembly (Wargacki et al. 2021).

To improve the pH sensitivity of the O432-17(-) trimeric plug design, we introduced a third histidine hydrogen bond network (3HIS) and point mutations at two residues involved in hydrophobic packing interactions at the trimeric interface within each protomer, I56V and L74A (named O432-17(-)_3HIS_I56V and O432-17(-)_3HIS_L74A). O432-17(-)_3HIS_I56V and O432-17(-)_3HIS_L74A each showed clear pH-dependent release of the pH trimer (**Figure 2.4G**). The apparent pKas of trimer dissociation (the pH where the AF647 fluorescence was 50% of the maximum signal) were pH 6.1 and pH 5.9 for O432-17(-)_3HIS_I56V and O432-17(-)_3HIS_L74A, respectively (**Figure S2.6E**). The apparent pKa of sfGFP fluorescence for these variants also shifted to more basic pH (pH 5.3 and pH 4.7, respectively) relative to the negative control trimer variant (named O432-17(-) 0 HIS networks), but less so than the AF647 apparent pKa, suggesting that nanoparticle disassembly was less pH-sensitive than plug dissociation from the nanoparticle (**Figure S2.6F**). O432-17(-)_3HIS_I56V_L74A, containing both point mutations per protomer, had an AF647-based apparent pKa of pH 6.7 and sfGFP-based apparent pKa of pH 5, indicating a synergistic effect of the two point mutations (**Figure 2.4G-H**). These pH-responsive, O432-17(-) variants did not encapsulate pos36GFP, suggesting that weakening the trimeric interface with an additional histidine hydrogen-bond network and up to two hydrophobic packing mutations per protomer reduces the assembly efficiency of the fully plugged nanoparticle in the presence of molecular cargo, even when the histidine network and point mutations were not designed at the interface between the trimeric plug and the tetramer.

We also introduced a third histidine hydrogen-bond network and the same two hydrophobic packing mutations in the O432-17(+) trimeric plug design. We generated four variants, two containing the third histidine hydrogen-bond network and either the I56V or L74A point mutation (O432-17(+)_3HIS_I56V and O432-17(+)_3HIS_L74A), one containing the third histidine hydrogen-bond network and both I56V and L74A point mutations (O432-17(+)_3HIS_ I56V_L74A), and one negative control containing zero histidine hydrogen bond networks (O432-17_0HIS), where all histidines were mutated to asparagine. We compared the apparent pKa, calculated from the AF647 and sfGFP fluorescence signals, of each positively charged plug variant over the pH titration relative to the negative control. All pH-responsive O432-17(+) trimeric plug variants showed an apparent pKa of pH 6.1 for AF647 and pH 5.8 for sfGFP fluorescence, respectively. Unlike the negatively charged variants, we did not observe a synergistic effect when combining the I56V and L74A mutations within the positively charged plug variants (**Figure S2.6G-H**). The apparent pKa of AF647 fluorescence for O432-17(+) variants containing histidine hydrogen-bond networks was more basic than the negative control (**Figure S2.6G-H**). The apparent pKa of sfGFP fluorescence for each O432-17(+) trimeric plug variant containing three histidine hydrogen-bond networks was only slightly more acidic than the

apparent pKa for AF647 fluorescence, suggesting that nanoparticle dissociation and plug dissociation from the nanoparticle were equally pH-sensitive for the O432-17(+) nanoparticles.

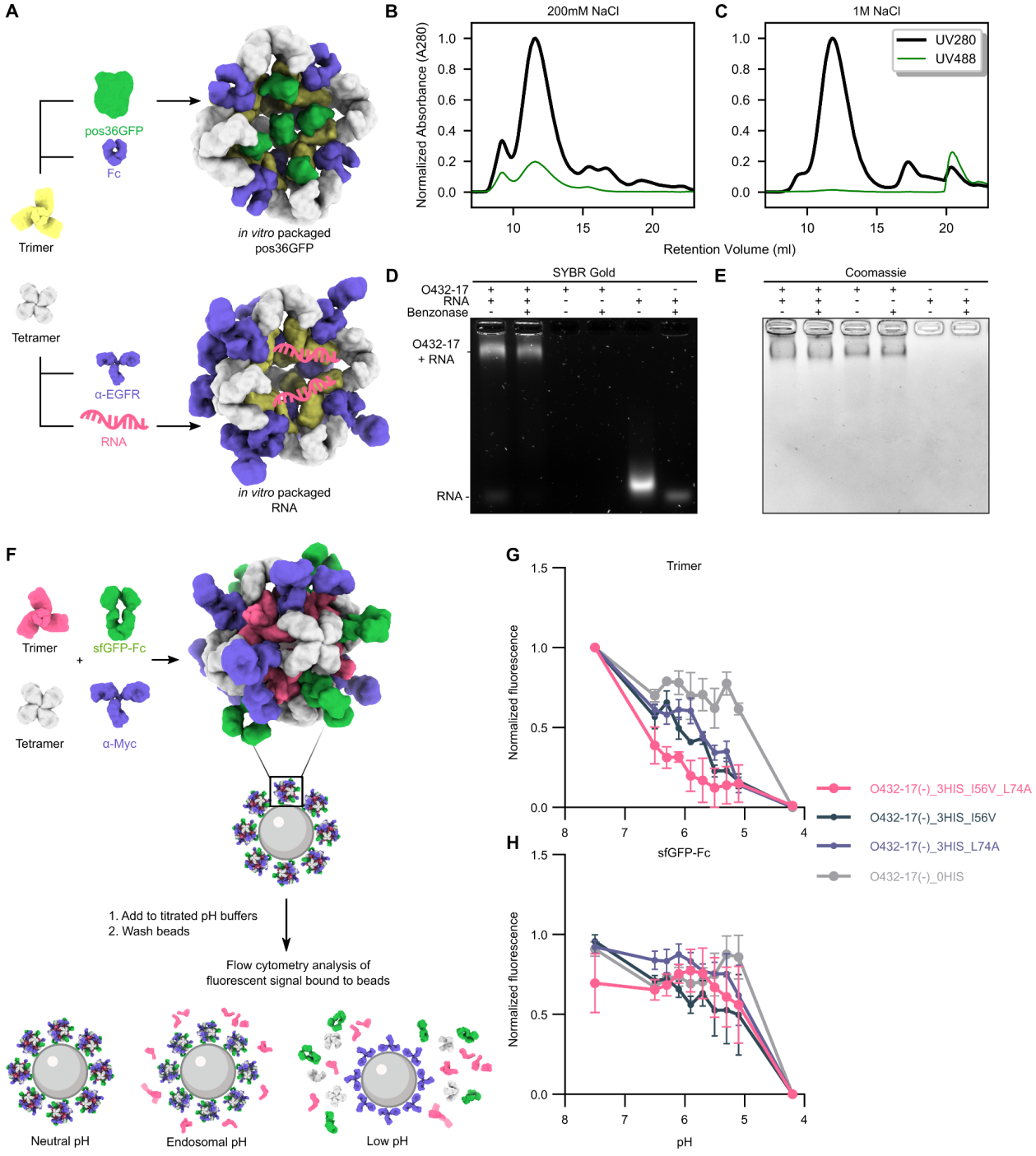


Figure 2.4: Plugged antibody nanoparticles electrostatically package cargoes and disassemble in response to acidification.

A. Positively or negatively charged variants of the designed trimers and designed tetramers were either assembled with pos36GFP and human Fc (top) or RNA and α-EGFR mAb (bottom). **B.** SEC chromatograms of *in vitro* packaging reactions with O432-17(-) were performed in either 200 mM NaCl or **C.** 1 M NaCl. Absorbance was monitored at 280 nm (black) and 488 nm (green). **D.** O432-17(+)

assembled *in vitro* with RNA was treated with Benzonase, electrophoresed on non-denaturing 0.8% agarose gels, and stained with SYBR gold (nucleic acid) and **E.** Coomassie (protein). **F.** pH titration experimental design. O432-17 nanoparticles were assembled with AF647-conjugated trimeric plug variants, designed tetramer, sfGFP-Fc, and α-Myc antibody. Nanoparticles were incubated with Myc peptide-coated beads and split into titrated pH buffers. Beads were washed in TBS pH 7.5 and the remaining trimer and sfGFP-Fc fluorescence remaining on the beads was analyzed by flow cytometry. **G.** AF647 and **H.** sfGFP fluorescence were normalized to the minimum and maximum values across the titration and analyzed as a function of pH for each nanoparticle variant.

Receptor mediated cellular uptake of O432-17 nanoparticles

We next tested the ability of the O432-17 nanoparticles to enter cells through receptor-mediated endocytosis. We assembled targeted nanoparticles by incubating a 1:1 stoichiometric mixture of CTX and Fc fused to mRuby2 (mRuby2-Fc) with the tetramer and a pH-responsive trimeric plug variant labeled with AF647 (named O432-17-CTX). As negative controls, we assembled a non-EGFR targeting nanoparticle by mixing the tetramer and trimer-AF647 with mRuby2-Fc (**Figure 2.5A**) (named O432-17-Fc). Assembled O432-17-CTX and O432-17-Fc were incubated at a final concentration of 10 nM per monomer in serum-free media for up to 16 hours with either A431 (high EGFR expression) wild-type (WT), HeLa WT (moderate EGFR expression), or HeLa EGFR knockout (KO) cells. After three hours of nanoparticle treatment, cells were fixed and immunostained with LAMP2A (late endosome and lysosomal marker) and imaged with confocal microscopy. We observed mRuby2 and AF647 fluorescence in A431 cells incubated with O432-17-CTX but little to no mRuby2 and AF647 fluorescence in cells incubated with O432-17-Fc, suggesting that the O432-17(-) nanoparticles can enter cells via EGFR-dependent endocytosis (**Figure 2.5B**). We also observed mRuby2 and AF647 fluorescence in HeLa WT cells incubated with O432-17-CTX (**Figure S2.7**). O432-17-CTX nanoparticles target specifically to EGFR-expressing cells, as pH-dependent O432-17-CTX nanoparticles did not accumulate in HeLa EGFR KO cells (**Figure S2.7**).

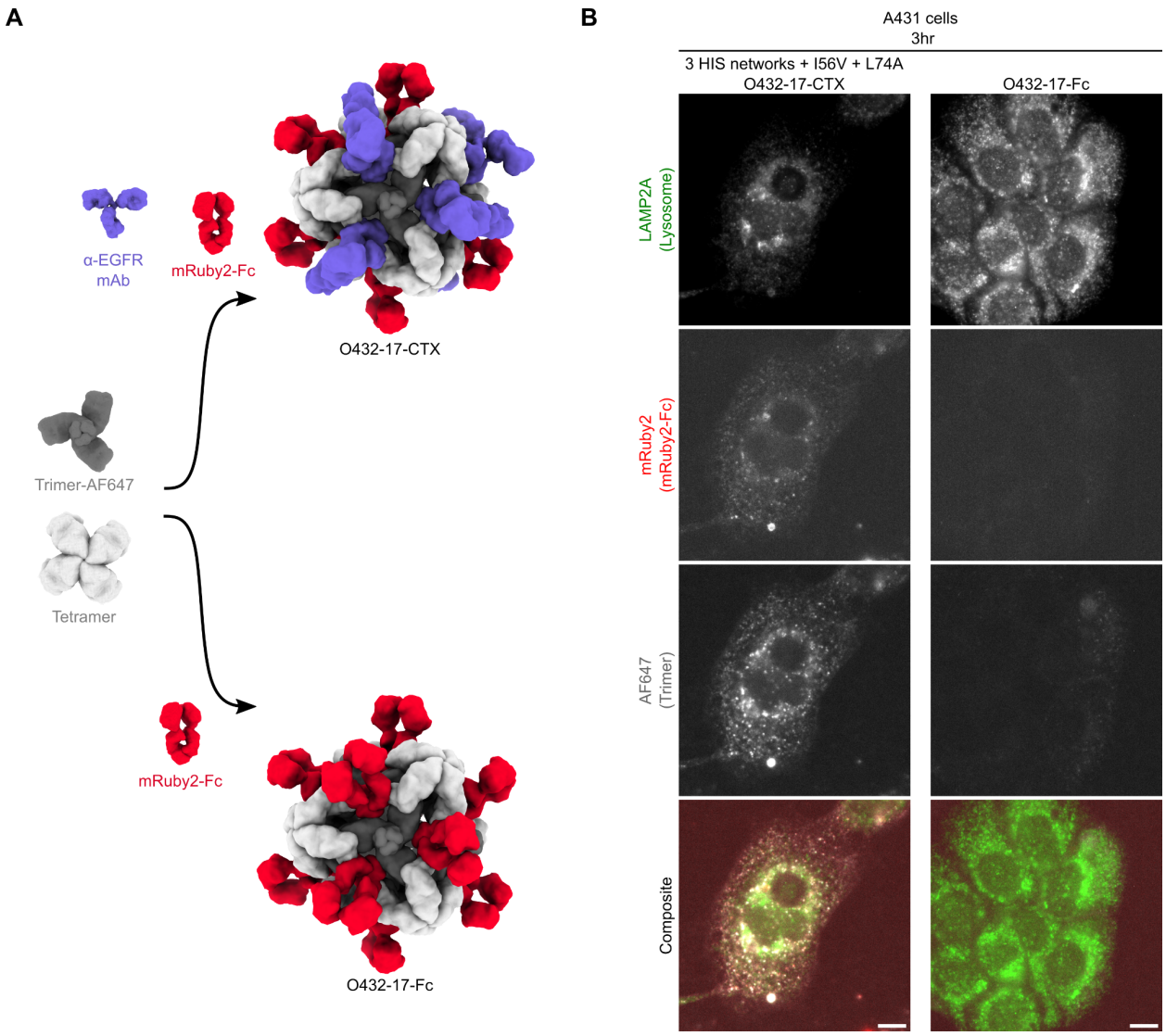


Figure 2.5: Targeted receptor-mediated uptake of O432 nanoparticles.
A. Targeted and non-targeted variants of the O432-17(-) nanoparticle were assembled *in vitro* with trimeric plug conjugated with AF647, designed tetramer, and either premixed mRuby2-Fc and α-EGFR mAb (top) or mRuby2-Fc alone (bottom). **B.** Cellular uptake of O432-17-CTX and O432-17-Fc nanoparticles was measured by the AF647 and mRuby2 fluorescence within the cell area, identified by lysosome membrane immunostaining using LAMP2A antibodies, after 3 hours of incubation in A431 cells. Images correspond to single confocal planes, and grayscale panels correspond to each channel of the composite image showing lysosomal membranes (green), mRuby2-Fc (red), and Trimer-AF647 (gray). Scale bar, 10 μm.

Discussion

We describe a general approach for reducing the porosity of protein nanomaterials by designing custom symmetric plugs that fill pores present along unoccupied symmetry axes. Using this approach, we generated the first designed protein nanoparticles with distinct structural

components on three different symmetry axes. These designed nanoparticles can modularly package molecular cargoes, undergo pH-dependent disassembly, and selectively enter cells in a programmable, antibody dependent manner. In contrast to functionalized homomeric and designed two-component protein nanoparticles (Sun et al. 2021; Edwardson, Mori, and Hilvert 2018; Sutter et al. 2008; Van de Steen et al. 2021; Steinmetz, Lim, and Sainsbury 2020; Cannon et al. 2020; Divine et al. 2021; Butterfield et al. 2017; Bale et al. 2016; Hsia et al. 2016; Votteler et al. 2016), each of the three components in our designs has a specific functional role. The trimer is responsible for pH-responsive disassembly and cargo packaging, the antibody provides cell targeting functionality, and the tetramer drives assembly of the three-component nanoparticle by making an interface with both the trimer and Fc domain. This division of labor makes our designed system highly modular: targeting specificity can be altered simply by switching the antibody component to target cells of interest, and the pH of disassembly and cargo packaging specificity can be programmed by choosing the appropriate trimer. Furthermore, assembly of the nanoparticles *in vitro* from independently purified components enables facile inclusion of multiple variants of each component, such as multiple distinct antibodies.

Induced nanoparticle disassembly at biologically relevant pH is a critical challenge for engineering drug delivery platforms, and our results represent a test of our understanding of protein disassembly and assembly dynamics (Lavelle, Michel, and Gingery 2007; H. Chen et al. 2016; Dalmau, Lim, and Wang 2009; Kim et al. 2011). We were able to finely tune the pH of disassembly of our O432 system through a combination of histidine hydrogen bonding networks and cavity-introducing mutations that weaken hydrophobic interactions at the trimeric component's oligomeric interface. Through this approach, we are able to raise the apparent pKa of disassembly to a remarkably high pH of 6.7, well above the pH of the endosome and in the range of many tumor microenvironments, and close to the maximum value achievable given the pKa of histidine.

Our O432 nanoparticle system is capable of packaging and protecting both protein and nucleic acid cargoes, disassembles at biologically relevant pH with precise tunability, incorporates a wide variety of targeting moieties, and is readily internalized by target cells. The tunable pH dependence makes this system a particularly attractive platform for engineering release and delivery of drugs during early stages of endosomal maturation. However, to be a broadly useful intracellular biologics delivery system, it will be necessary to incorporate endosomal escape machinery in future designs. The nanoparticles also provide a route to conditional delivery of drugs into the tumor microenvironment. Tumor-killing or -modulating cargoes could be packaged within the nanoparticles and directed to the tumor through targeting with tumor-specific antibodies; the pH-dependent release of cargo could minimize off-tumor toxicity and systemic exposure as compared to classic direct antibody conjugation approaches by providing an additional checkpoint on proper localization. The pH-dependent disassembly, programmability, and versatility of the O432 platform provides multiple exciting paths forward for biologics delivery.

Methods

Extension of pH-responsive trimeric building blocks

We used the WORMS helical fusion software (Hsia et al. 2021) to generate fusions between the base pH-trimer helical bundle and pairs of 82 helical repeat proteins (Brunette et al. 2015), designed to cover a broad range of axial and radial displacement between repeat units. Scaffold information was prepared in a json file, which is used as an input parameter in the WORMS software through the `--dbfiles` option. For each scaffold, the pdb file path (`"file"`), a geometric class specification (`"class"`), and a connections dictionary (`"connections"`) specifying the chain (`"chain"`), terminus direction (`"direction"`), and residue positions that were available for fusion (`"residues"`) were included in the json file. Geometric class specification for oligomeric scaffolds required an N or C either following the oligomeric architecture, which defined the terminus available for subsequent fusions. A connections dictionary specifying each chain, terminus direction, and residue positions allowed for fusion was required for each chain and terminus direction. The oligomeric scaffold asymmetric unit only required one connections dictionary, specifying fusion to only one terminus and one chain. As monomers, helical repeat proteins require a list of two connections dictionaries as they allow fusions to either N or C terminus. An example json file example is shown below where the connections dictionary specifies allowable fusions on the C terminus of the pH trimer from residue 50 and over, and anywhere along the helical repeat proteins:

```
{
{"file": "/path/to/ph_trimer.pdb",
  "class": ["C3_C"],
  "connections":[
    {"chain": 1, "direction": "C", "residues":["50:"]}
 ]
},
{"file": "/path/to/helicalrepeatprotein1.pdb",
  "class": ["Monomer"],
  "connections": [
   {"chain": 1, "direction": "N", "residues":[":210"]},
   {"chain": 1, "direction": "C", "residues":["-210:"]}
  ]
},
{"file": "/path/to/helicalrepeatprotein2.pdb",
  "class": ["Monomer"],
  "connections": [
   {"chain": 1, "direction": "N", "residues":[":160"]},
   {"chain": 1, "direction": "C", "residues":["-160:"]}
  ]
}
}
```

To prepare the WORMS software for fusions, we also included a map for backbone connections (`--bbconn`), which specifies the order for which scaffolds of each class could be fused. We used the following backbone connection map, which specifies that the N terminus of the fusion of two helical repeat proteins in the monomer class be fused to the C terminus of an oligomer with C3 symmetry:

```
        --bbconn C3 C3_C   \
                 NC Monomer \
                 N_ Monomer \
```

Finally, we specified a `--geometry NullCriteria()` option, which allowed generating arbitrary fusions of varying radius and length to the C terminus of the pH trimer.

Computational docking of plugged antibody nanoparticles

The generated fusions served as docking inputs into the antibody nanoparticle through the (`--inputs1` and `--inputs2` options). To prepare the antibody nanoparticle for axle docking along its C3 axis, we generated a context structure pdb file by isolating the chains forming the C3 symmetric aperture. We fed this context structure and each trimer fusion separately into the RPXDock application (Sheffler et al. 2022) using the axle docking protocol (`--architecture AXLE_C3`). To enable trimming, we set a maximum number of trimmed residues (`--max_trim`) and specified the chains representing the fusion to be trimmed (`--trimmable_components`). Finally, to enable greater sampling, we allowed the RPXDock application to sample the entire search space (`--docking_method`) at a cartesian and angular resolution of 0.25 angstroms and degrees, respectively (`--grid_resolution_cart_angstroms` and `--grid_resolution_ori_degrees`). An example axle docking executable is shown below.

```
PYTHONPATH=/path/to/rpxdock/site-packages /path/to/python/environment
/path/to/rpxdock/dock.py \
                --architecture AXLE_C3 \
                --inputs1 /path/to/plugfusion.pdb \
                --inputs2 /path/to/contextstructure.pdb \
                --max_trim 400 \
                --cart_bounds -300 300 \
                --docking_method grid \
                --grid_resolution_cart_angstroms 0.25 \
                --grid_resolution_ori_degrees 0.25 \
                --hscore_files ailv_h \
                --hscore_data_dir /path/to/rpxdock/hscore \
                --trimmable_components “A” \
                --function stnd \
```

We evaluated the docked interfaces between each fusion and antibody nanoparticle using the RPX score and NContact metrics, which are predictive of geometric complementarity and

designability. Empirically, we deemed docks with a RPX score greater than 50 and greater than one residue pair contact to be suitable for sequence optimization.

Computational design of plugged antibody nanoparticles

We used the Rosetta Macromolecular Modeling suite to optimize the sequence residues contributing to either the fusion domains or interface between plug and nanoparticle. With residue selectors, we identified clashing residue positions as a result of the WORMS fusion and residues contributing to the interface between the plug and antibody nanoparticle. The sequence of the selected residues were optimized based on the local positions of each residue defined by secondary structure, solvent accessible surface area (SASA) of the backbone and of the Cβ atom, and number of amino acid side chains in a cone extending along the CA-CB vector side chain neighbors. An example of the fusion, docking, and design of O432-17 is included at https://github.com/erincyang/plug_design.git.

Small scale bicistronic bacterial protein expression for trimeric plug and tetrameric designs for sfGFP-Fc lysate assembly

Synthetic genes bicistronically encoding designed trimeric plug and tetramer sequences were purchased from Genscript in pET29b+ vectors and connected by an intergenic region (‘TAAAGAAGGAGATATCATATG') and a C-terminal 6xHistidine tag on the tetramer. Expression plasmids were transformed into BL21(DE3) *e. Coli* cells and grown in LB medium supplemented with 50 mg/L Kanamycin at 37°C overnight. The overnight culture was diluted into autoinduction media and incubated 37°C overnight. Cells were lysed chemically in BugBuster supplemented with 1mM PMSF and 20 mM imidazole, and cleared by centrifugation. Clarified lysates were incubated with 3uM final concentration of sfGFP-Fc and purified by immobilized metal affinity chromatography (IMAC) with Ni-NTA magnetic beads. The soluble fractions were washed with 25 mM Tris pH 8.0, 300 mM NaCl, 60 mM imidazole before eluting with 25 mM Tris pH 8.0, 300 mM NaCl, 300 mM Imidazole. Elution fractions from bicistronic expression plasmids were subsequently subjected to Native PAGE and SDS-PAGE to identify slow migrating species that coeluted three proteins of different molecular weights indicating assembly of three-component species.

Production of Fc and Fc-fusions

Fc, sfGFP-Fc, and Cetuximab IgG were cloned into CMVR and transfected into in ExpiHEK293F cells and purified by IMAC with 50 mM Tris pH8.0, 300 mM NaCl, 500 mM imidazole elution buffer and purified further via size exclusion chromatography over a Superdex 200 10/300 GL FPLC column (Cytiva) into 50 mM Tris, pH 8.0, 300 mM NaCl, 0.75% CHAPS (Fc) or 50 mM Tris, pH 8.0, 300 mM NaCl, 0.05% glycerol (Cetuximab IgG, sfGFP-Fc). Stocks were frozen at -80 for subsequent analyses.

Large scale expression and purification of O432-17 components

Designs appearing to co-purify and yielding slowly migrating species by native PAGE were subsequently subcloned into pet29b+ vectors each encoding either a trimeric plug or a tetramer variant–both with C terminal hexahistidine tags–and expressed at larger scale (1 to 12 liters of culture). Cells were lysed by microfluidization in 25 mM Tris pH 8.0, 300 mM NaCl, 1 mM DTT, 1mM PMSF, 0.1mg/ml DNAse and cleared by centrifugation. Clarified lysates were filtered through 0.7um filters and purified by IMAC via gravity columns with nickel-NTA resin or HisTrap HP columns (Cytiva) using 25 mM Tris pH 8.0, 300 mM NaCl, 60 mM Imidazole wash buffer and 25 mM Tris pH 8.0, 300 mM NaCl, 300 mM Imidazole elution buffer. Elution fractions containing pure proteins were concentrated using centrifugal filter devices (Millipore) and further purified on a Superdex 200 10/300 GL (for large scale purification) or Superose 6 10/300 GL (for comparison to assembly) gel filtration column (Cytiva) using 25 mM Tris pH 8.0, 150 mM NaCl, 0.75% CHAPS. Gel filtration fractions containing pure protein in the desired oligomeric state were pooled, concentrated and frozen in aliquots at -80°C for subsequent analyses.

In vitro assembly of O432-17 nanoparticles

Purified tetramer and Fc or IgG from gel filtration fractions were assembled in a 1:1 molar ratio and with 1.1x excess trimeric plug. Molar ratios were calculated based on the monomeric extinction coefficient. Assemblies were assembled between 100-500 uL between 5-50 uM, dialyzed overnight at 25°C into 25 mM Tris pH 8.0, 150 mM NaCl, and purified on a Superose 6 10/300 GL gel filtration column (Cytiva) using 25 mM Tris pH 8.0, 150 mM NaCl as the running buffer.

In vitro assembly of O432-17 nanoparticles with molecular cargoes were assembled in the same molar ratio of purified tetramer, Fc or Cetuximab IgG, and trimeric plug from gel filtration fractions with 1x purified pos36GFP per monomer of component or 3x pegRNA per nanoparticle to conserve RNA material. Nanoparticles were screened for in vitro packaging by SEC on a Superose 6 10/300 GL column (Cytiva) in either low or high salt Tris buffer (25 mM Tris, 200 mM NaCl pH 8.0, 25 mM Tris, 1 M NaCl, pH 8.0).

Dynamic light scattering

DLS measurements were performed using the default Sizing and Polydispersity method on the UNcle (Unchained Labs). O432-17 variants (8.8 µl) were pipetted into the provided glass cuvettes. DLS measurements were run in triplicate at 25°C with an incubation time of 1 s; results were averaged across 10 runs and plotted using Python3.7.

Negative Stain Electron Microscopy preparation and data collection of O432-17 nanoparticles and variants

Pre- or post-SEC O432-17 assemblies between 0.1 and 0.2 mg/ml in 25 mM Tris pH 8.0, 150 mM NaCl were applied onto 400- or 200-mesh carbon-coated copper grids and glow discharged for 20 s, followed by 3x application of 3.04 µl 2% nano-W or Uranyless stain.

Micrographs were recorded using EPU software (Thermo Fisher) on a 120 kV Talos L120C transmission electron microscopy (Thermo Fisher) at 45,000 nominal magnification (pixel size: 3.156 Å per pixel) at a defocus range of 1.0 to 2.5 µm.

Negative Stain Electron Microscopy data analysis of O432-17 nanoparticles and variants

Negative Stain Electron Microscopy datasets were processed by Relion3.0 software (Zivanov et al. 2018). Micrographs were imported into the Relion3.0 software and a contrast transfer function was estimated using GCTF (Zhang 2016). Around 500 particles were manually picked, 3D classified, and selected classes were used as templates for particle picking in all images. Approximately 100k picked particles were 2D classified for 25 iterations into 50 classes.

Cryo-Electron Microscopy preparation and data collection of O432-17 nanoparticles

2 ul of pre-SEC purified O432-17-Fc sample at 0.5 mg/ml in 25 mM Tris pH 8.0, 150 mM NaCl was applied onto C-flat 1.2/1.3 holey carbon grids. Grids were then plunge-frozen into liquid ethane and cooled with liquid nitrogen using a ThermoFisher Vitrobot Mk IV with 6.5-s blotting time and 0 blot force. The blotting process took place inside the vitrobot chamber at 22°C and 100% humidity. Data acquisition was performed with Leginon on a Titan Krios electron microscope operating at 300 kV using a K3 summit direct electron detector equipped with an energy filter and operating in super-resolution mode. The nominal magnification for data collection was 105000X with a calculated pixel size of 0.42 Å/pixel, with a final dose of 63.775 e^-/A^2 for 2223 movies.

Cryo-Electron Microscopy data analysis of O432-17 nanoparticles

All data processing was carried out in CryoSPARC v3.0.0 (Punjani et al. 2017). Alignment of movie frames was performed using Patch Motion with an estimated B-factor of 500 $Å^2$, with a maximum alignment resolution set to 5. During alignment, all movies were fourier cropped by ½. Defocus and astigmatism values were estimated using Patch CTF with default parameters. 1,391 nanoparticle particles were initially manually picked and extracted with a box size of 576 pixels. This was followed by a round of 2D classification and subsequent template-picking using the best 2D class averages low-pass filtered to 20 Å. Particles were next picked with Template Picker and were manually inspected before extracting with a box size of 660 pixels, and further fourier cropped to a final box size of 330 pixels, for a total of 66,904 particles. A round of reference-free 2D classification was next performed in CryoSPARC with a maximum alignment resolution of 6 Å. The best classes which revealed visibly assembled nanoparticles were used for 3D ab initio determination using the C1 symmetry operator. This was followed by a round of 3D heterogeneous refinement using C1 symmetry and sorting into 4 distinct classes, all of which revealed complete plugging of the octahedral 3-component nanoparticle. Thus, all 50,017 of the best particles selected from 2D classification were subjected to non-uniform 3D refinement with octahedral symmetry applied, yielding a final map with an estimated global resolution of 7.06 Å, following per-particle defocus refinement. The final maps were deposited in the EMDB under

accession number EMD-29602. Relaxed models were generated by rigid-body docking followed by relaxing the design model into the final cryo-EM density map in Rosetta (Tyka et al. 2011).

Computational design of O432-17 electrostatically charged variants

A consensus design approach was used to first identify interior surface positions predicted to be the most robust to surface mutations. These positions were divided into three tiers based on the predicted enhancement to stability and / solubility. Using the Rosetta modeling suite, the trimeric plug design model was redesigned by allowing optimization of the identities of interior surface residues that did not contribute to the interface between plug monomers or between the trimeric plug and antibody nanoparticle. We deployed three tiers of sequence optimization strategies per charged variant. The first strategy enabled optimization of residue positions toward only charged amino acid identities (arginine or lysine for the positively charged variants, glutamate and aspartate for the negatively charged variants). The second strategy included the charged amino acid identities but also polar, non-charged amino acid identities such as asparagine, glutamine, and alanine to maintain helical propensity. The third strategy included the residue identities in the first and second strategy, but also enabled reversion back to the original identity in the trimeric plug design model. Mutations that resulted in losses of significant atomic packing interactions or side chain-side chain or side chain-backbone hydrogen bonds were discarded. The best scoring design for each design strategy and surface position tier were selected for inclusion as variant proteins.

Expression and purification of electrostatically charged trimeric plug variants

Plasmids encoding electrostatically charged trimeric plug variants and containing a C terminal hexahistidine tag were cloned into pet29b+ vectors and expressed overnight at 37°C in autoinduction media (Studier 2005). Cells were lysed by microfluidization or sonication in 25 mM Tris pH 8.0, 500 mM NaCl, 1 mM DTT, 1mM PMSF, 0.1mg/ml DNAse and cleared by centrifugation. Clarified lysates were filtered through 0.7um filters and purified by IMAC via gravity columns with nickel-NTA resin or HisTrap HP columns (Cytiva) using 25 mM Tris pH 8.0, 500 mM NaCl, 60 mM Imidazole wash buffer and 25 mM Tris pH 8.0, 500 mM NaCl, 300 mM Imidazole elution buffer. Elution fractions containing pure proteins were concentrated using centrifugal filter devices (Millipore) and further purified on a Superdex 200 10/300 GL (for large scale purification) or Superose 6 10/300 GL (for comparison to assembly) gel filtration column (Cytiva) using 25 mM Tris pH 8.0, 500 mM NaCl, 0.75% CHAPS. Gel filtration fractions containing pure protein in the desired oligomeric state were pooled, concentrated and frozen in aliquots at -80°C for subsequent analyses.

Gel electrophoresis

Native agarose gels were prepared using 0.8% ultrapure agarose (Invitrogen) in TAE buffer (Thermo Fisher) containing SYBR Gold (Invitrogen). 9 µL of O432-17 nucleocapsids were treated with 1 µL of Benzonase (Invitrogen) at 25 °C for 30 min, followed by mixing with 2uL 6x

loading dye (NEB, no SDS), and electrophoresed at 120 V for 30 min. Gels were imaged for RNA and subsequently stained by Gelcode Blue for protein (ThermoFisher).

Protein SDS-PAGE gels were performed using anyKD polyacrylamide gels (Bio-Rad) in Tris-glycine buffer.

Rational design of O432-17 pH-responsive variants

Bulky hydrophobic residues such as isoleucine and leucine amino acid positions within the pH-responsive trimeric interface were selected for optimization to either alanine or valine using the Rosetta Software Suite. The point mutation with the best scoring interface energy (ddG) at each position was selected as a trimeric plug variant. Combinatorial pairs of each point mutation variant were also included as trimeric plug variants.

AF647 conjugation to O432-17 trimeric plug variants

Trimeric plug variants containing a T359C mutation were generated by site directed mutagenesis PCR. Alexa Fluor™ 647 C2 Maleimide (Thermo Fisher Scientific) dissolved in DMSO and trimeric plug variants containing 10x TCEP were incubated in a 5:1 molar ratio with respect to the trimeric plug monomer for 16 hours (overnight) at 4°C in PBS + 0.75% CHAPS titrated to pH 7.2. The maleimide reaction was quenched with 1mM DTT and buffer exchanged into 25 mM Tris pH 8.0, 150 mM NaCl using PD-10 desalting columns with Sephadex-25 resin (Cytiva) according to manufacturer's protocol and dialyzed for 3-4 days in 25 mM Tris pH 8.0, 150 mM NaCl, 0.75% CHAPS at 4°C. Positively charged trimeric plug variants were buffer exchanged and dialyzed into 25 mM Tris pH 8.0, 500 mM NaCl, 0.75% CHAPS.

Flow-cytometry based pH-titration

Linear myc peptide with an N terminal lysine side chain and 3x glycine linker (KGGGEQKLISEEDL) was produced via solid-phase peptide synthesis and biotinylated via amide formation. The resulting biotinylated myc peptide was purified by RP-HPLC and quality checked for the proper molecular weight via LC-MS, lyophilized, and dissolved in 100% DMSO for long term storage. 3.0-3.4 μm streptavidin coated polystyrene particles (Spherotech) were incubated with biotinylated myc peptide diluted in PBS to 5% DMSO for 20 minutes at 25°C. Following two washes in PBS + 3% BSA by pelletting at 3000G for 5 minutes, coated polystyrene particles were incubated with α-myc-O432-17 nanoparticle variants for one hour at 25°C. The coated particles were washed in PBS + 3% BSA twice and split equally into pH-titrated citrate-phosphate buffers for 30 minutes at 25°C. Coated particles were washed twice with PBS + 3% BSA and resuspended for flow-cytometry.

All flow-cytometry experiments were performed on a LSR II Flow Cytometer (BD Biosciences). Lasers were calibrated with coated particles stained with either FITC Anti-Myc tag antibody (Abcam 9E10), APC Anti-Myc tag antibody (Abcam 9E10), or no antibody. 10,000 events were collected per sample on three biological replicates. All flow-cytometry results were analyzed using the FlowJo, LLC software. Normalization to the minimum and maximum fluorescence of

each channel with each titration sample was performed in Python3.7, and the apparent pKa of AF647 and sfGFP fluorescence was estimated with a four parameter nonlinear logistic regression fit in GraphPad Prism version 9.3.1 for Windows, GraphPad Software, San Diego, California USA, www.graphpad.com.

Cells

WT HeLa (ATCC CCL-2), EGFR KO HeLa (Abcam ab255385), and A431 cells (ATCC CRL-1555) were cultured at 37 °C with 5% CO2 in flasks with Dulbecco's modified Eagle medium (DMEM) (Gibco) supplemented with 1 mM L-glutamine (Gibco), 4.5 g/liter D-glucose (Gibco), 10% fetal bovine serum (FBS) (Hyclone) and 1% penicillin-streptomycin (PenStrep) (Gibco). WT HeLa and EGFR KO HeLa also cultured with 1× nonessential amino acids (Gibco) supplemented into the media. Cells were passaged twice per week. To passage, cells were dissociated using 0.05% trypsin EDTA (Gibco) and split 1:5 or 1:10 into a new tissue culture (TC)–treated T75 flask (Thermo Scientific ref 156499).

Immunostaining

35 mm glass bottom dishes seeded at a density of 20k cells / dish. A final monomeric concentration of 10 nM of O432-17-CTX or O432-17-Fc nanoparticles were incubated with cultured cells in serum-free DMEM. Cells were fixed 4% paraformaldehyde, permeabilized with 100% methanol, and blocked with PBS + 1% BSA. Cells were immunostained with Anti-LAMP2A antibody (Abcam ab18528) followed by goat anti-rabbit- IgG Alexa Fluor™ 488 secondary antibody (Thermo Fisher A-11034) and 4′,6-diamidino-2-phenylindole (DAPI) (Thermo Fisher D1306) and stored in the dark at 4°C until imaging.

Nanoparticle uptake confocal microscopy (A431 and HeLa EGFR KO Cells)

Cells were washed twice with FluoroBrite DMEM imaging media and subsequently imaged in the same media in the dark at room temperature. Epifluorescence imaging was performed on a Yokogawa CSU-X1 spinning dish confocal microscope with either a Lumencor Celesta light engine with 7 laser lines (408, 445, 473, 518, 545, 635, 750 nm) or a Nikon LUN-F XL laser launch with 4 solid state lasers (405, 488, 561, 640 nm), 40x/0.95 NA objective and a Hamamatsu ORCA-Fusion scientific CMOS camera, both controlled by NIS Elements software (Nikon). The following excitation/emission filter combinations (center/bandwidth in nm) were used: BFP: EX408, EM443/38, GFP: EX473 EM525/36, RFP: EX545 EM605/52, Far Red: EX635 EM705/72. Exposure times were 100 ms for the acceptor direct channel and 500ms for all other channels, with no EM gain set and no ND filter added. All epifluorescence experiments were subsequently analyzed using Image J.

Nanoparticle uptake image acquisition (HeLa WT Cells)

4-color, 3D images were acquired with a commercial OMX-SR system (GE Healthcare). Toptica diode lasers with excitation at 405nm, 488nm, and 640nm were used. Emission was collected on three separate PCO.edge sCMOS cameras using an Olympus 60× 1.42NA PlanApochromat oil immersion lens. 1024×1024 images (pixel size 6.5 µm) were captured with no binning.

Acquisition was controlled with AcquireSR Acquisition control software. Z-stacks were collected with a step size of 125 nm. Images were deconvolved in SoftWoRx 7.0.0 (GE Healthcare) using the enhanced ratio method and 200 nm noise filtering. Images from different color channels were registered in SoftWoRx using parameters generated from a gold grid registration slide (GE Healthcare).

Delivery assays

Gal8 visualization of endosomal disruption

Galectin 8 (Gal8) is a cytosolically dispersed protein that redistributes by binding to interior-facing glycans present in endosomal membranes. As a result, the redistribution of Gal8 fluorescent fusion protein (Gal8-YFP) from the cytosol into endosomes can be tracked real-time and in live cells, where diffuse distribution of Gal8-YFP fusion indicates no endosomal disruption or glycan binding and Gal8-YFP puncta would correspond to endosomal disruption and access to binding glycosylation moieties selectively located on the inner face of endosomal membranes. HEK293T or HeLa cells stabling expressing Gal8-YFP are cultured in DMEM supplemented with 10% fetal bovine serum (FBS) and 1% puromycin. The puromycin is a selection antibiotic for YFP expressing cells. Stable cells were obtained from the Duvall group (Kilchrist et al. 2019).

Cells were plated at 15k cells per well in Corning 96 Well Half-Area High-Content Imaging Glass-Bottom Microplates (Corning™ 4580) and allowed to proliferate for 24 hours before substrate treatment. Substrates were added to each well in a serial dilution with final concentration of 10nM to 10uM and a final volume of 100uL per well. Substrates were incubated with cells overnight (16-18 hours) before imaging.

Before imaging, cells were washed with PBS and the media was replaced with: DMEM Fluorobright + 25 mM HEPES + 10% fetal bovine serum (FBS) + hoechst 33342 (nuclear marker) + Propidium iodide (PI, live/dead cell marker) and screened via IN Cell Analyzer 2000 Bioimager (GE HEALTHCARE) for YFP puncta as an indication of endosomal disruption.

Prime Editing Frameshift Assay

This cell-based assay reports the delivery of a guide RNA via a change in fluorescence and antibiotic resistance. Through prime editing, which is a modification of the CRISPR system, a mutant Cas9 editor (Cas9-H840A) fused to a modified reverse transcriptase from the MMLV virus creates a single-stranded nick (rather than a double stranded break) (Anzalone et al. 2019). The precise location of this nick is encoded in a modified single guide RNA (sgRNA), fused to a reverse transcriptase template encoding the desired edit, called a prime-editing guide RNA (pegRNA) (“PE Resources - Liu Group” 2020, “pegLIT” n.d., “pegRNA Design GuideFinal_022620.pdf” n.d.; Nelson et al. 2022; P. J. Chen et al. 2021; Scholefield and Harrison 2021).

The reporter plasmid encodes for mCherry and is resistant to blasticidin in its “off state”. Upon delivery of the pegRNA, a gene editing event occurs which inserts a guanine into position 2934 and results in the expression of eGFP and confers puromycin resistance and disables expression of mCherry and blasticidin resistance.

WT HEK293T and WT HeLa cells were transduced with the reporter plasmid via lentiviral transduction and selected with blasticidin. Per approximately 1.4 million cells in a T25 flask, cells were transfected using Lipofectamine 3000 (Thermo) with 3 µg PEmax, 1 µg PE3b, and 1 µg mTagBFP plasmid DNA and allowed to proliferate for 24 hours in DMEM + 10% fetal bovine serum + 1% blasticidin. Transfected cells were then seeded at a density of 50k cells / well in sterile, tissue-culture treated, clear bottom, 24-well microplates and allowed to proliferate for 24 hours in DMEM + 10% fetal bovine serum + 1% blasticidin.

Prior to adding O432-17(+) nanoparticles containing encapsidated pegRNA (sequence below), cells were washed with PBS and moved into media containing DMEM + 10% fetal bovine serum + 1% Pen/Strep. O432-17(+) trimeric plug variants were mixed with equimolar amounts of O432-17 tetramer and Cetuximab and 3 pegRNAs per nanoparticle to assemble the full O432-17 nanoparticles (O432-17(+)-CTX). The mixture containing all components was dialyzed overnight to a final buffer of 25 mM Tris, 150 mM NaCl, pH 8.0. O432-17(+)-CTX nanoparticles were added at a final concentration of 10 nM per well. As a positive control, 25 pmol of pegRNA was transfected per well via Xfect™ RNA Transfection Reagent according to manufacturer’s instructions (Takara Bio). Cells were washed, trypsinized, and analyzed after 24 hours for expression of eGFP and mCherry via flow cytometry without puromycin selection and after 72 hours after puromycin selection. Selected cells were expanded into 6 well plates prior to selection.

pegRNA sequence:
mC*mC*mA*rGrGrCrUrUrCrCrGrGrGrUrCrArUrCrCrCrGrUrUrUrUrArGrArGrCrUrArGrArArArUrArGrCrArArGrUrUrArArArArArUrArArGrGrCrUrArGrUrCrCrGrUrUrArUrCrArArCrUrUrGrArArArArArGrUrGrGrCrArCrCrGrArGrUrCrGrGrUrGrCrArCrCrUrGrGrUrGrUrArUrGrArCrCrCrGrGrArCrGrCrGrGrUrUrCrUrArUrCrUrArGrUrUrArCrGrGrUrGrUrUrArArArArCrCrArArCrUrA*mG*mA*mA

Nanoluciferase reconstitution

Stable HeLa cells expressing PuroR-EGFP fusion protein and LgGiT protein were seeded at 20k cells/well in a 96-well white bottom plate (Corning) and selected by puromycin. O432-17 trimeric plugs were expressed with an N terminal HIS tag and C terminal fusion to HiBiT peptide. O432-17 trimeric plugs with terminal HiBiT fusions were mixed with equimolar amounts of O432-17 tetramer and Cetuximab to assemble the full O432-17 nanoparticles (O432-17-CTX). The mixture containing all components was dialyzed overnight to a final buffer of 25 mM Tris, 150 mM NaCl, pH 8.0. The full O432-17-CTX nanoparticle variants or trimeric plug variants were incubated with WT HeLa cells expressing LgBiT-EGFP for 24 hours at a final concentration of 10 nM per monomer in the presence of 10 µM of DrkBiT peptide (Somiya and Kuroda 2021). After the incubation with O432-17-CTX, cells were mixed with live cell luciferase substrate (Promega) and luminescence of split nanoBiT reconstitution was read with a Synergy Neo2 plate reader (BioTek). Positive control cells were cotransfected with plasmids expressing HiBiT peptide and LgBiT protein, while negative control cells were only transfected with LgBiT protein.

Delivery assays with functional protein cargo

Production of Diphtheria Toxin variants

Diphtheria Toxin is a three-domain bacteria toxin with implications for disrupting cell proliferation. The receptor binding domain (DTR) enables the toxin to enter specific cell types via receptor mediated endocytosis. The translocase domain (DTT) enables the toxin to form a large pore in the endosomal membrane in response to endosomal acidification and shuttles the toxin into the cytosol, and the domain carrying the active site for ADP ribosylation of EF-2 (DTA) inhibits ribosomal activity upon binding to cytosolic ribosomes (Kagan, Finkelstein, and Colombini 1981; J. Wang and London 2009; Senzel et al. 2000).

DTA-DTT variants (**Table S2.8**) were codon optimized for e. coli expression. Each construct contained an N terminal HIS tag, followed by a SUMO tag. DTA-DTT were separated by a furin cleavage site and each gene was terminated by a thrombin cleavage site, followed by a strep II tag. Subsequent payloads such as zinc finger nucleases could also be constructed with the following format: 6xHIS tag - SUMO tag - **Gene of interest** - thrombin cleavage site - strep II tag. Variants were transformed in BL21DE3 competent cells and cultured in LB media overnight. 100 mL of LB media was inoculated with 2 mL of overnight culture and shaken at 200 rpm at 37°C. At OD = 0.6, cultures were induced with 1 mM IPTG and shaken for 4 hours at 20°C. CUltures were pelleted and lysed using sonication and purified with immobilized metal affinity chromatography (IMAC). IMAC eluates were buffer exchanged either by desalting columns or size exclusion chromatography into low-imidazole containing buffer (25 mM Tris, 150 mM NaCl, pH 8.0) and incubated with SUMO protease at 4°C overnight. Following SUMO cleavage, DTA-DTT variants were purified via size exclusion chromatography on a Superdex 200 10/300 GL column (Cytiva).

Assembly of O432-17 variants with Diphtheria Toxin variants

O432-17(-) and O432-17(+) trimeric plugs were mixed with equimolar amounts of O432-17 tetramer and Cetuximab and 0.25✕ DTA-DTT variants to assemble the full O432-17 nanoparticles (O432-17-CTX). The mixture containing all components was dialyzed overnight to a final buffer of 25 mM Tris, 150 mM NaCl, pH 8.0. DTA-DTT variants genetically fused to O432-17 trimeric plugs were assembled with tetramer and Cetuximab in a premixed mixture of 25% DTA-DTT fused to trimeric plug and 75% O432-17(-) or 75% O432-17(+) trimeric plug, followed by dialysis.

Analysis of cargo packaging with Diphtheria Toxin variants

Assembled nanoparticles containing DTA-DTT payload were purified via size exclusion chromatography via a Superose 6 10/300 GL column (Cytiva) in 25 mM Tris, 150 mM NaCl, pH 8.0. The nanoparticle and component peak fractions were analyzed via SDS-PAGE and Western Blot, staining with an α-strep II HRP antibody (IBA #2-1509-001) and an α-HIS HRP antibody (Invitrogen MA1-21315-HRP) as a staining control.

Supplementary Figures

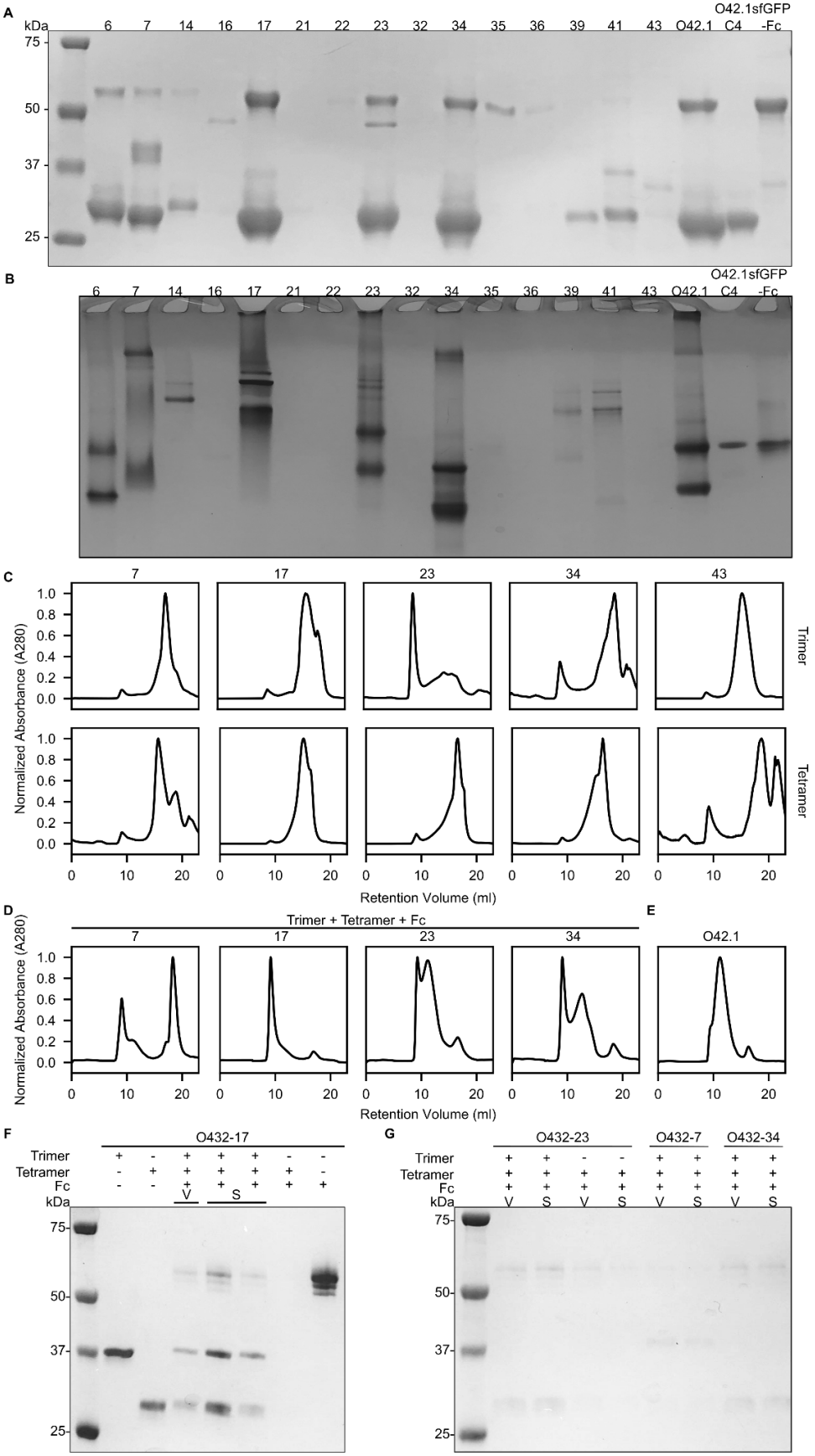

A
O42.1sfGFP
kDa
6 7 14 16 17 21 22 23 32 34 35 36 39 41 43 O42.1 C4 -Fc
75
50
37
25
B
O42.1sfGFP
6 7 14 16 17 21 22 23 32 34 35 36 39 41 43 O42.1 C4 -Fc
C
7 17 23 34 43
Normalized Absorbance (A280)
Trimer
Tetramer
Retention Volume (ml)
D
Trimer + Tetramer + Fc
7 17 23 34
Normalized Absorbance (A280)
Retention Volume (ml)
E
O42.1
F
O432-17
Trimer
Tetramer
Fc
kDa
V S
75
50
37
25
G
O432-23 O432-7 O432-34
Trimer
Tetramer
Fc
kDa
V S V S V S V S
75
50
37
25

Figure S2.1: Example experimental screen of O432 nanoparticles.
A. Clarified lysates of 16 designs where the designed trimer co-eluted with the tetramer were supplemented with a purified sfGFP-Fc fusion protein, purified by IMAC, and subject to reducing SDS-PAGE, **B.** followed by non-denaturing native PAGE gel electrophoresis. All designs were compared against the previously designed, two-component antibody nanoparticle (O42.1) (Divine et al. 2021)**,** O42.1 tetrameric component and sfGFP-Fc. **C.** Subcloned components from putative three-component assemblies were purified separately, **D.** and, material permitting, after stoichiometric *in vitro* assembly by SEC. **E.** Non-reducing SDS-PAGE of the void (V) and shoulder (S) peaks from SEC of O432-17 with purified O432-17 tetramer + Fc assembly and trimer, tetramer, and Fc components as controls, and **F.** of the void (V) and shoulder (S) peaks from SEC purification of O432 assemblies.

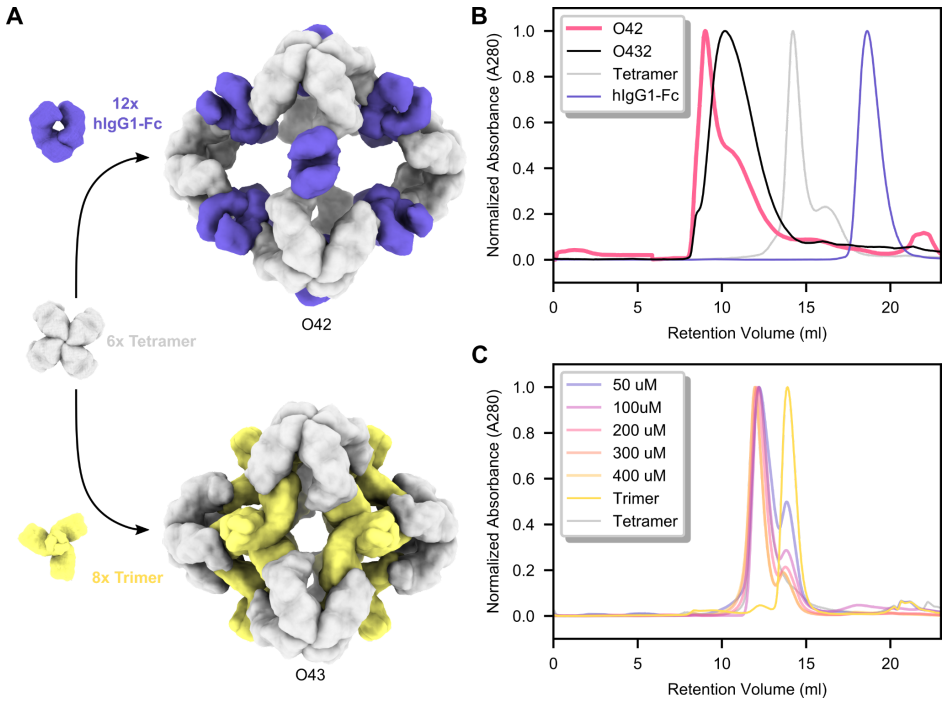


Figure S2.2: Mixing two of the three individual components does not result in nanoparticle assembly.
A. 6× designed tetramers form protein-protein interfaces with both 12× Fc and 8× designed trimers. A schematic depicts a hypothetical nanoparticle assembly from the designed tetramer with only the Fc (top) or only the trimer (bottom). **B.** Representative SEC traces on the Superose 6 10/300 GL of assembly reactions containing only the designed tetramer and Fc (pink), compared to the full three-component assembly (black) and the individual components (gray and purple). **C.** Representative SEC traces on the Superdex 200 10/300 GL of assembly reactions containing only the designed tetramer and trimer at concentrations of 50 μM, 100 μM, 200 μM, 300 μM, and 400 μM, compared to the individual components (gray and yellow).

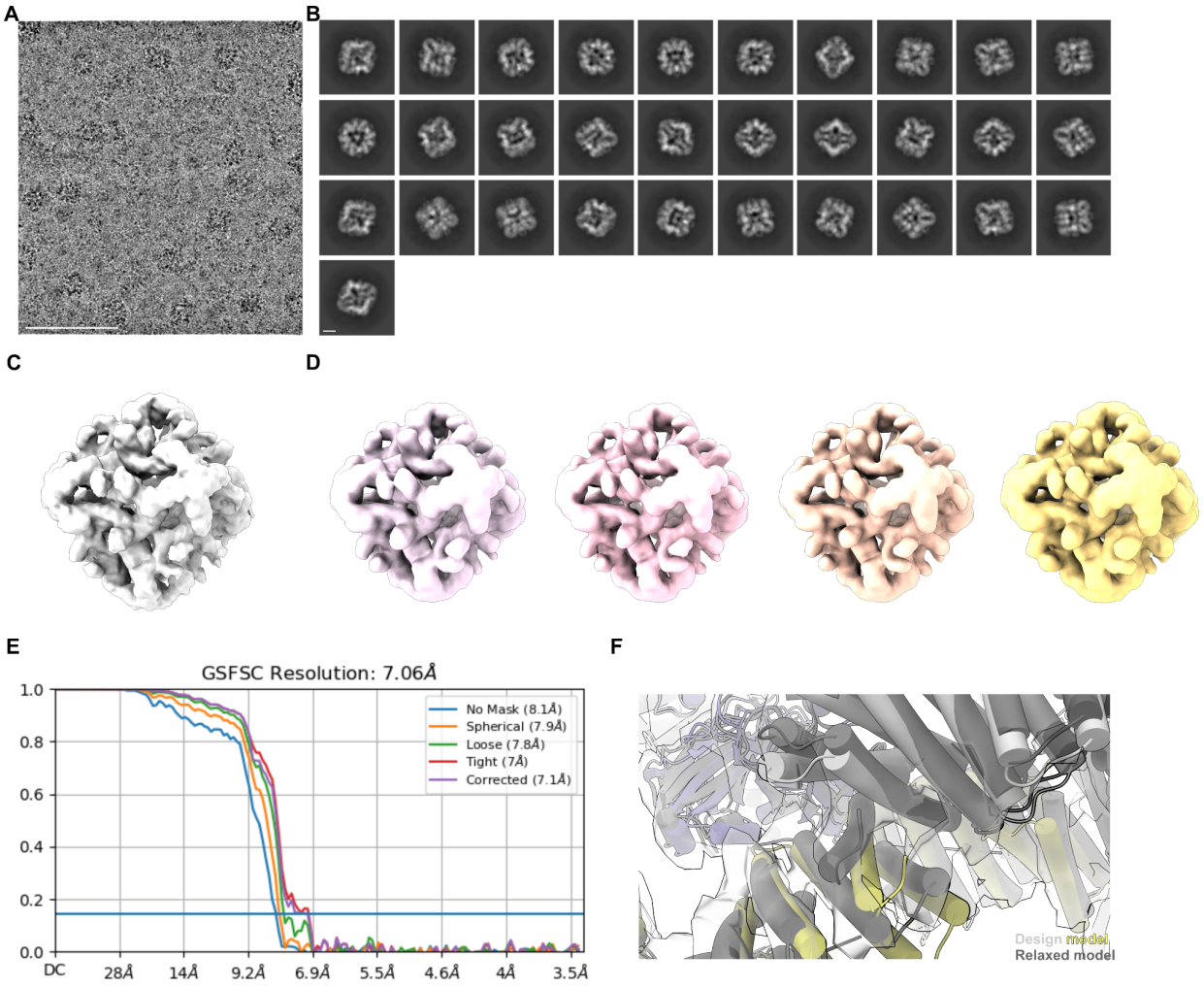


Figure S2.3: Cryo-EM processing shows newly designed trimeric plug occupying all 3-fold symmetry axes of the nanoparticle octahedral architecture.
A. Representative micrograph of cryo-EM sample. Scale bar, 100 nm. **B.** Reference-free two-dimensional class averages. Scale bar, 10 nm. **C.** *Ab initio* three-dimensional reconstruction without applied octahedral symmetry. **D.** 3D reconstructions generated following a heterogeneous refinement in the absence of applied octahedral symmetry. All four classes show trimeric plugs occupying all facets of the designed nanoparticle. **E.** Gold-standard Fourier shell correlation curves for the O432-Fc EM density map with octahedral symmetry applied. **F.** Close up of plug interface between design model (light gray and yellow) and model built from the 3D reconstruction (dark gray).

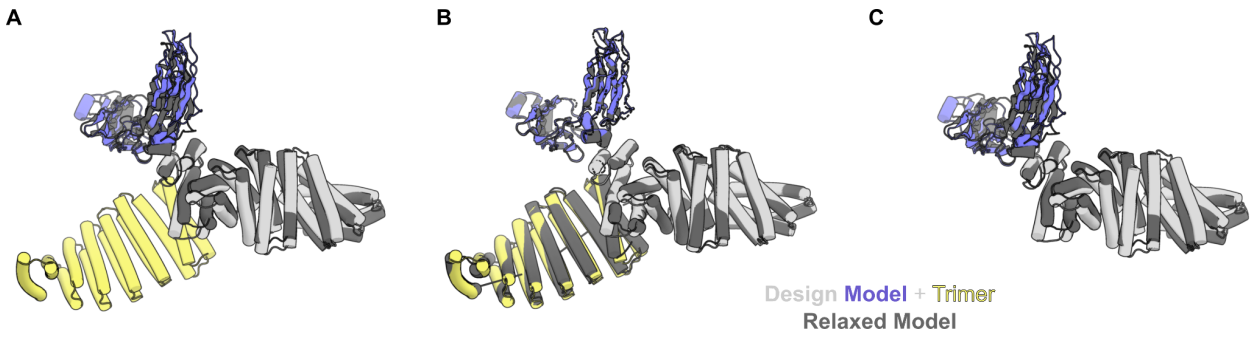


Figure S2.4: Comparison of design models of O42.1 and O432-17 with models relaxed into Cryo-EM density.
A. The three-component O432-17 design model (light gray, purple, and yellow) overlaid on the two-component O42.1 relaxed model (dark gray) shows an RMSD of 1.9 Å. **B.** The O432-17 design model (light gray, purple, and yellow) deviates from its relaxed model (dark gray) by 1.6 Å. **C.** The O42.1 design model deviates from its relaxed model (dark gray) by 4.2 Å.

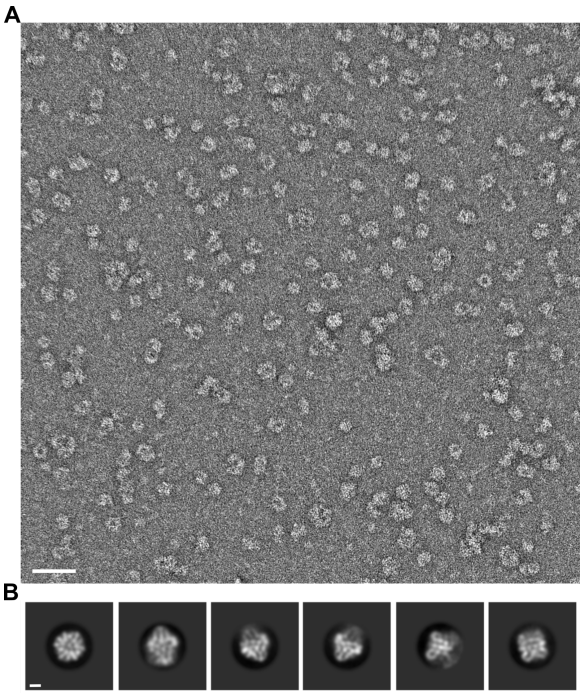


Figure S2.5: Negative stain electron microscopy of O432-17 nanoparticle assembly in the presence of RNA.
A. Representative negatively stained electron micrograph of O432-17(+) in the presence of RNA. Scale bar, 100 nm. **B.** Reference-free two-dimensional class averages showing multiple views of O432-17 nanoparticles. Scale bar, 10 nm.

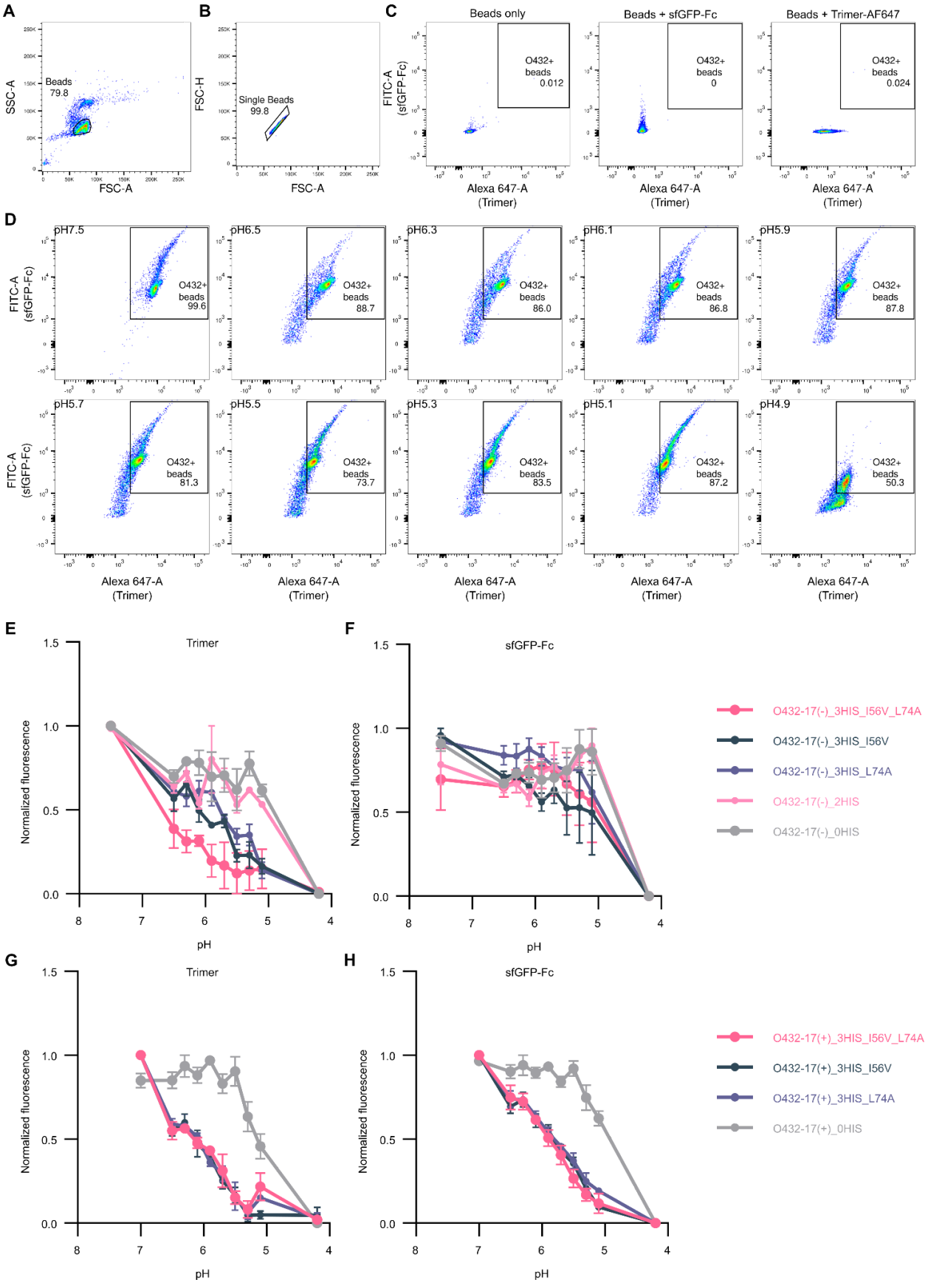

A
SSC-A
FSC-A
Beads
79.8
B
FSC-H
FSC-A
Single Beads
99.8
C
Beads only
Beads + sfGFP-Fc
Beads + Trimer-AF647
FITC-A
(sfGFP-Fc)
Alexa 647-A
(Trimer)
O432+
beads
0.012
O432+
beads
0
O432+
beads
0.024
D
pH7.5
O432+
beads
99.6
pH6.5
O432+
beads
88.7
pH6.3
O432+
beads
86.0
pH6.1
O432+
beads
86.8
pH5.9
O432+
beads
87.8
pH5.7
O432+
beads
81.3
pH5.5
O432+
beads
73.7
pH5.3
O432+
beads
83.5
pH5.1
O432+
beads
87.2
pH4.9
O432+
beads
50.3
E
Trimer
F
sfGFP-Fc
Normalized fluorescence
pH
O432-17(-)_3HIS_I56V_L74A
O432-17(-)_3HIS_I56V
O432-17(-)_3HIS_L74A
O432-17(-)_2HIS
O432-17(-)_0HIS
G
Trimer
H
sfGFP-Fc
O432-17(+)_3HIS_I56V_L74A
O432-17(+)_3HIS_I56V
O432-17(+)_3HIS_L74A
O432-17(+)_0HIS

Figure S2.6: Flow cytometry analysis of O432-17(-) and O432-17(+) nanoparticle variants.
A. Forward and side scattering intensity isolated all beads. **B.** Isolated beads were selected for singlet scattering. **C.** Singlet beads were gated for both AF647 and sfGFP-Fc signal with reference to negative controls: beads only (left), beads incubated with sfGFP-Fc (middle), and beads incubated with AF647-labeled trimeric plug (right). **D.** The mean fluorescence intensity of beads positive for trimer and sfGFP-Fc signal was taken at each pH. This gating strategy was applied across all trimeric plug variants: 0HIS, 2HIS, 3HIS_I56V, 3HIS_L74A, 3HIS_I56V_L74A. Mean fluorescence intensity of O432-17(-) nanoparticles was measured as a function of pH for the **E.** trimeric plug variants and **F.** sfGFP-Fc. Mean fluorescence intensity of O432-17(+) nanoparticles was measured as a function of pH for the **G.** trimeric plug variants and **H.** sfGFP-Fc.

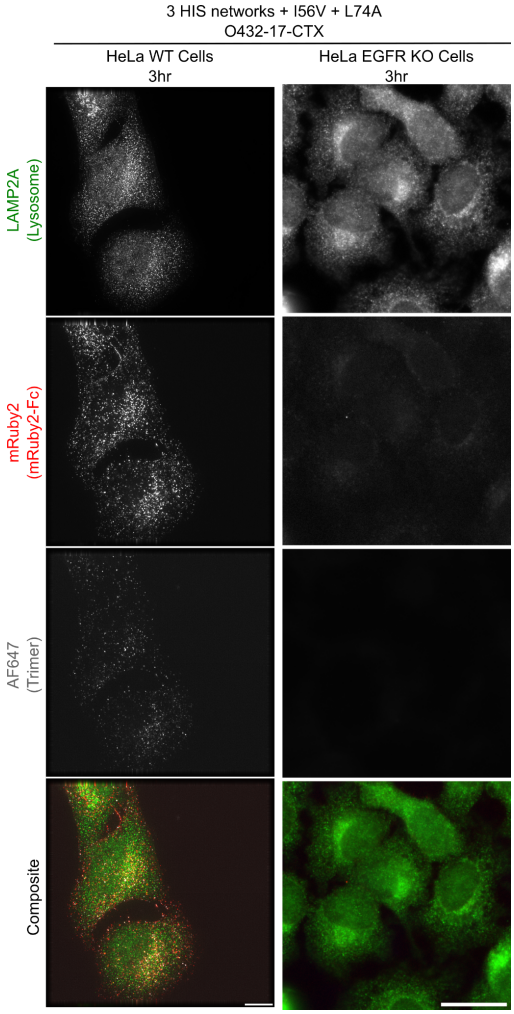


Figure S2.7: Targeted receptor-mediated uptake of O432-17-CTX nanoparticles in WT and EGFR KO HeLa cells after 3 hours.
Scale bar, 10 μm.

Supplementary Tables

Table 2.1: Cryo-EM data collection and refinement statistics

	O432-17
Microscope	Krios
Voltage (kV)	300
Detector	Gatan K3 Summit
Recording mode	Super-resolution
Magnification	105,000 X
Movie micrograph pixel size (Å)	0.42
Dose rate (e^{-}/Å^2/s)	20
No. of frames per movie micrograph	75
Frame exposure time (ms)	40
Movie micrograph exposure time (s)	3
Total dose (e^{-}/Å^2)	63.775
Under focus range (μm)	0.8 - 1.7
Number of movie micrographs	2,223
Total number of picked particles	88,357
Particles in the final reconstruction	50,017
Map symmetry	O
Map resolution (GS-FSC)	7.05 Å
B-factor	-230.0
EMDB ID	EMD-29602

Table 2.2: Amino acid sequences of assembling O432-17 designs

Design name	**Sequence**
O432-17-C3	MSEEKIEKLLEELTASTAELKRATASLRAITEELKKNPSEDALVEHNRAIVEHNAIIV ENNRIIATVLLAIVAAIATNEATLAADKAKEAGASEVAKLAKKVLEEAEELAKENDS EEALKVVKAIADAAKAAAEAAREGKTEVAKLALKVLEEAIELAKENRSEEALKVV REIARAALAAAQAAEEGKTEVAKLALKVLEEAIELAKENRSEEALKVVREIARAAL AAAQAAEEEGKTEVAKLALEVLEQAIEAAKLQRSERALEMVREIARAALEAARN AEGGRSDRARAILASLKVSIIVVKLKSSGTSEEEILRIVLKIIKELRKTAKESGQSAS YIATMEAEIVKAIDYALDLSGTSGSWSGLEHHHHHH
O432-17-C4	MFNKDQQSAFYEILNMPNLNEALRNGFIQLLKDDPSKSEVILTAALIAAKLSEDIR TLKESGSSYEEIAERVARAVALLVALLKTNGVSEDEIALAVALIISAVIQTLKESGSS YEVIAEIVARIVAEIVEALKRSGTSEDEIAEIVARVISEVIRTLKESGSSYEVIAEIVAR IVAEIVEALKRSGTSEDEIAKIVARVIAEVLRTLKESGSSEEVIKEIVARIITEIKEALK

	RSGTSEDEIELITLMIEAALEIAKLKSSGSEYEEIAEDVARRIAELVEKLKRDGTSA VEIAKIVAAIISAVIAMLKASGSSYEVIAEIVARIVAEIVEALKRSGTSAAIIALIVALVIS EVIRTLKESGSSFEVILEIVIRIVLEIIEALKRSGTSEQDVMLIVMAVLLVVLATLQLS GSGSWSGLEHHHHHH
Negatively charged variants	
O432-17(-)_2HIS-C3	MSEEKIEKLLEELTASTAELKRATASLRAITEELKKNPSEDALVEHNRAIVEHNAIIV ENNRIIATVLLAIVAAIATNEATLAADKAKEAGASEVAELAKEVLEEAEELAKENDS EEALKVVKAIADAAKAAAEAAREGKTEVAELALKVLEEAIELAKENRSEEALKVV REIARAALAAAQAAEEGKTEVAELALEVLEEAIELAKENRSEEALKVVREIARAAL AAAQAAEEGKTEVAELALEVLEQAIEAAKLQRSERALEMVREIARAALEAARNA EGGRSDRARAILASLKVSIIVVKLKSSGTSEEEILRIVLKIIKELRKEAKEEGQSAS YIATMEAEIVKAIDYALDLSGTSGSWSGLEHHHHHH*
O432-17(-)_3HIS_I56V_L74A-C3	MMSEEKIEKLLEELTAATAELKRATASLRAITEELKKNPSEDALVEHNRAIVEHNAI VVEHNRIIATVLLAIVAAAATNEATLAADKAKEAGASEVAELAKEVLEEAEELAKE NDSEEALKVVKAIADAAKAAAEAAREGKTEVAELALKVLEEAIELAKENRSEEAL KVVREIARAALAAAQAAEEGKTEVAELALEVLEEAIELAKENRSEEALKVVREIAR AALAAAQAAEEGKTEVAELALEVLEQAIEAAKLQRSERALEMVREIARAALEAAR NAEGGRSDRARAILASLKVSIIVVKLKSSGTSEEEILRIVLKIIKELRKEAKEEGQS ASYIATMEAEIVKAIDYALDLSGCSGSWSGLEHHHHHH
O432-17(-)_0HIS-C3	MMSEEKIEKLLEELTASTAELKRSTASLRASTEELKKNPSEDALVENNRLIVENNA IIVENNRIIATVLLAIVAAIATNEATLAADKAKEAGASEVAELAKEVLEEAEELAKEN DSEEALKVVKAIADAAKAAAEAAREGKTEVAELALKVLEEAIELAKENRSEEALK VVREIARAALAAAQAAEEGKTEVAELALEVLEEAIELAKENRSEEALKVVREIARA ALAAAQAAEEGKTEVAELALEVLEQAIEAAKLQRSERALEMVREIARAALEAARN AEGGRSDRARAILASLKVSIIVVKLKSSGTSEEEILRIVLKIIKELRKEAKEEGQSA SYIATMEAEIVKAIDYALDLSGCSGSWSGLEHHHHHH
O432-17(-)_3HIS_I56V-C3	MSEEKIEKLLEELTAATAELKRATASLRAITEELKKNPSEDALVEHNRAIVEHNAIV VEHNRIIATVLLAIVAAIATNEATLAADKAKEAGASEVAELAKEVLEEAEELAKEND SEEALKVVKAIADAAKAAAEAAREGKTEVAELALKVLEEAIELAKENRSEEALKV VREIARAALAAAQAAEEGKTEVAELALEVLEEAIELAKENRSEEALKVVREIARAA LAAAQAAEEGKTEVAELALEVLEQAIEAAKLQRSERALEMVREIARAALEAARNA EGGRSDRARAILASLKVSIIVVKLKSSGTSEEEILRIVLKIIKELRKEAKEEGQSAS YIATMEAEIVKAIDYALDLSGCSGSWSGLEHHHHHH
O432-17(-)_3HIS_L74A-C3	MSEEKIEKLLEELTAATAELKRATASLRAITEELKKNPSEDALVEHNRAIVEHNAIIV EHNRIIATVLLAIVAAAATNEATLAADKAKEAGASEVAELAKEVLEEAEELAKEND SEEALKVVKAIADAAKAAAEAAREGKTEVAELALKVLEEAIELAKENRSEEALKV VREIARAALAAAQAAEEGKTEVAELALEVLEEAIELAKENRSEEALKVVREIARAA LAAAQAAEEGKTEVAELALEVLEQAIEAAKLQRSERALEMVREIARAALEAARNA EGGRSDRARAILASLKVSIIVVKLKSSGTSEEEILRIVLKIIKELRKEAKEEGQSAS YIATMEAEIVKAIDYALDLSGCSGSWSGLEHHHHHH*
Positively charged variants	
O432-17(+)_2HIS-C3	MSEEKIEKLLEELTASTAELKRATASLRAITEELKKNPSEDALVEHNRAIVEHNAIIV ENNRIIATVLLAIVAAIATNEATLAADKAKEAGASEVAKLAKKVLKQAEQLAKENDS EEALKVVKAIADAAKAAAEAAREGKTEVAKLALKVLANAIKLAKENRSEEALKVV REIARAALAAAQAAEEGKTEVARLALKVLQNAIQLAKENRSEEALKVVREIARAA

	LAAAQAAEEGKTEVAKRALKVLQQAIQAAKLQRSERALEMVREIARAALEAARN AEGGRSDRARAILASLQVSIIVVKLKSSGTSEEEILRKVLKIIKELRKKAKEQGQS ASYIATMEAEIVKAIDYALDLSGTSGSWSGLEHHHHHH*
O432-17(+)_3HIS_I56V-C3	MSEEKIEKLLEELTAATAELKRATASLRAITEELKKNPSEDALVEHNRAIVEHNAIV VEHNRIIATVLLAIVAAIATNEATLAADKAKEAGASEVAKLAKKVLKQAEQLAKEND SEEALKVVKAIADAAKAAAEAAREGKTEVAKLALKVLANAIKLAKENRSEEALKV VREIARAALAAAQAAEEGKTEVARLALKVLQNAIQLAKENRSEEALKVVREIARA ALAAAQAAEEGKTEVAKRALKVLQQAIQAAKLQRSERALEMVREIARAALEAAR NAEGGRSDRARAILASLQVSIIVVKLKSSGTSEEEILRKVLKIIKELRKKAKEQGQ SASYIATMEAEIVKAIDYALDLSGCSGSWSGLEHHHHHH
O432-17(+)_3HIS_L74A-C3	MSEEKIEKLLEELTAATAELKRATASLRAITEELKKNPSEDALVEHNRAIVEHNAIIV EHNRIIATVLLAIVAAAATNEATLAADKAKEAGASEVAKLAKKVLKQAEQLAKEND SEEALKVVKAIADAAKAAAEAAREGKTEVAKLALKVLANAIKLAKENRSEEALKV VREIARAALAAAQAAEEGKTEVARLALKVLQNAIQLAKENRSEEALKVVREIARA ALAAAQAAEEGKTEVAKRALKVLQQAIQAAKLQRSERALEMVREIARAALEAAR NAEGGRSDRARAILASLQVSIIVVKLKSSGTSEEEILRKVLKIIKELRKKAKEQGQ SASYIATMEAEIVKAIDYALDLSGCSGSWSGLEHHHHHH
O432-17(+)_0HIS-C3	MSEEKIEKLLEELTASTAELKRSTASLRASTEELKKNPSEDALVENNRLIVENNAII VENNRIIATVLLAIVAAIATNEATLAADKAKEAGASEVAKLAKKVLKQAEQLAKEND SEEALKVVKAIADAAKAAAEAAREGKTEVAKLALKVLANAIKLAKENRSEEALKV VREIARAALAAAQAAEEGKTEVARLALKVLQNAIQLAKENRSEEALKVVREIARA ALAAAQAAEEGKTEVAKRALKVLQQAIQAAKLQRSERALEMVREIARAALEAAR NAEGGRSDRARAILASLQVSIIVVKLKSSGTSEEEILRKVLKIIKELRKKAKEQGQ SASYIATMEAEIVKAIDYALDLSGCSGSWSGLEHHHHHH

Table 2.3: Amino acid sequences of hIgG1-Fc, Fc-fusions, and IgGs produced and used to assemble O432-17 designs

Design name	Sequence
hIgG1-Fc	EPKSSDKTHTCPPCPAPELLGGPSVFLFPPKPKDTLMISRTPEVTCVVVDVSHE DPEVKFNWYVDGVEVHNAKTKPREEQYNSTYRVVSVLTVLHQDWLNGKEYKC KVSNKALPAPIEKTISKAKGQPREPQVYTLPPSRDELTKNQVSLTCLVKGFYPSDI AVEWESNGQPENNYKTTPPVLDSDGSFFLYSKLTVDKSRWQQGNVFSCSVMH EALHNHYTQKSLSLSPGK
sfGFP-Fc	SRATMETDTLLLWVLLLWVPGSTGHHHHHHGGSENLYFQGGSSKGEELFTGVV PILVELDGDVNGHKFSVRGEGEGDATNGKLTLKFICTTGKLPVPWPTLVTTLTYG VQCFSRYPDHMKRHDFFKSAMPEGYVQERTISFKDDGTYKTRAEVKFEGDTLV NRIELKGIDFKEDGNILGHKLEYNFNSHNVYITADKQKNGIKANFKIRHNVEDGSV QLADHYQQNTPIGDGPVLLPDNHYLSTQSVLSKDPNEKRDHMVLLEFVTAAGIT HGMDELYKGGSGSEPKSSDKTHTCPPCPAPELLGGPSVFLFPPKPKDTLMISRT PEVTCVVVDVSHEDPEVKFNWYVDGVEVHNAKTKPREEQYNSTYRVVSVLTVL HQDWLNGKEYKCKVSNKALPAPIEKTISKAKGQPREPQVYTLPPSRDELTKNQV SLTCLVKGFYPSDIAVEWESNGQPENNYKTTPPVLDSDGSFFLYSKLTVDKSRW QQGNVFSCSVMHEALHNHYTQKSLSLSPGK

mRuby2-Fc	SRATMETDTLLLWVLLLWVPGSTGHHHHHHGGSENLYFQGGSVSKGEELIKEN MRMKVVMEGSVNGHQFKCTGEGEGNPYMGTQTMRIKVIEGGPLPFAFDILATS FMYGSRTFIKYPKGIPDFFKQSFPEGFTWERVTRYEDGGVVTVMQDTSLEDGC LVYHVQVRGVNFPSNGPVMQKKTKGWEPNTEMMYPADGGLRGYTHMALKVD GGGHLSCSFVTTYRSKKTVGNIKMPGIHAVDHRLERLEESDNEMFVVQREHAV AKFAGLGGGMDELYKGGSGSEPKSSDKTHTCPPCPAPELLGGPSVFLFPPKPK DTLMISRTPEVTCVVVDVSHEDPEVKFNWYVDGVEVHNAKTKPREEQYNSTYR VVSVLTVLHQDWLNGKEYKCKVSNKALPAPIEKTISKAKGQPREPQVYTLPPSR DELTKNQVSLTCLVKGFYPSDIAVEWESNGQPENNYKTTPPVLDSDGSFFLYSK LTVDKSRWQQGNVFSCSVMHEALHNHYTQKSLSLSPGK
Cetuximab light chain	MELGLSWIFLLAILKGVQCDILLTQSPVILSVSPGERVSFSCRASQSIGTNIHWYQ QRTNGSPRLLIKYASESISGIPSRFSGSGSGTDFTLSINSVESEDIADYYCQQNN NWPTTFGAGTKLELKRTVAAPSVFIFPPSDEQLKSGTASVVCLLNNFYPREAKV QWKVDNALQSGNSQESVTEQDSKDSTYSLSSTLTLSKADYEKHKVYACEVTHQ GLSSPVTKSFNRGEC
Cetuximab heavy chain	MELGLSWIFLLAILKGVQCQVQLKQSGPGLVQPSQSLSITCTVSGFSLTNYGVH WVRQSPGKGLEWLGVIWSGGNTDYNTPFTSRLSINKDNSKSQVFFKMNSLQS NDTAIYYCARALTYYDYEFAYWGQGTLVTVSAASTKGPSVFPLAPSSKSTSGGT AALGCLVKDYFPEPVTVSWNSGALTSGVHTFPAVLQSSGLYSLSSVVTVPSSSL GTQTYICNVNHKPSNTKVDKRVEPKSCDKTHTCPPCPAPELLGGPSVFLFPPKP KDTLMISRTPEVTCVVVDVSHEDPEVKFNWYVDGVEVHNAKTKPREEQYNSTY RVVSVLTVLHQDWLNGKEYKCKVSNKALPAPIEKTISKAKGQPREPQVYTLPPS REEMTKNQVSLTCLVKGFYPSDIAVEWESNGQPENNYKTTPPVLDSDGSFFLYS KLTVDKSRWQQGNVFSCSVMHEALHNHYTQKSLSLSPGKGSGHHHHHH

Table 2.4: Amino acid sequences of molecular cargoes used by O432-17 designs

Design name	Sequence
pos36-GFP	MGHHHHHHGGASKGERLFRGKVPILVELKGDVNGHKFSVRGKGKGDATRGKL TLKFICTTGKLPVPWPTLVTTLTYGVQCFSRYPKHMKRHDFFKSAMPKGYVQER TISFKKDGKYKTRAEVKFEGRTLVNRIKLKGRDFKEKGNILGHKLRYNFNSHKVY ITADKRKNGIKAKFKIRHNVKDGSVQLADHYQQNTPIGRGPVLLPRNHYLSTRSK LSKDPKEKRDHMVLLEFVTAAGIKHGRDERYK
pegRNA	mC*mC*mA*rGrGrCrUrUrCrCrGrGrGrUrCrArUrCrCrCrGrUrUrUrUrArGrArGrCrUr ArGrArArArUrArGrCrArArGrUrUrArArArArUrArArGrGrCrUrArGrUrCrCrGrUrUrAr UrCrArArCrUrUrGrArArArArGrUrGrGrCrArCrCrGrArGrUrCrGrGrUrGrCrGrCrArC CrCrUrGrGrUrGrUrArUrGrArCrCrCrGrGrArCrGrCrGrGrUrUrCrUrArUrCrUrArGrU rUrArCrGrCrGrUrUrArArArArCrCrArArCrUrA*mG*mA*mA

Table 2.5: Details on EM data acquisition on different O432-17 samples

Sample name	Stain	Magnification	Pixel size (Å/pixel)	# Micrographs
O432-17 Fc	Uranyless	45,000	3.156	289
O432-17 CTX	Uranyless	45,000	3.156	249

O432-17(+) RNA CTX	Uranyless	45,000	3.156	169

Table 2.6: Details on EM data processing on different O432-17 samples

Sample name	Particle picking	CTF estimation	2D class averages	# particles in final selected 2D classes/total picked particles
O432-17 Fc	CisTEM	CTFFIND4 within Relion	Relion	84107/112334
O432-17 CTX	Relion template picking	CTFFIND4 within Relion	Relion	23676/84107
O432-17(+) RNA CTX	CryoSPARC template picking	CTFFIND4 within CryoSPARC	CryoSPARC	18204/79682

Table 2.7: Statistical information for pH titration experiments.
All analyses were performed using Graphpad Prism version 9.3.1 Software.

Experiment (Fig.)	Dunnett's multiple comparisons test	Mean Diff.	95.00% CI of diff.	Below threshold?	Summary	Adjusted P Value
Fig 4G/S6D	O432-17(-)_0HIS vs. O432-17(-)_2HIS	0.06277	-0.2448 to 0.3703	No	ns	0.9598
	O432-17(-)_0HIS vs. O432-17(-)_3HIS_I56V	0.25	-0.05761 to 0.5575	No	ns	0.1407
	O432-17(-)_0HIS vs. O432-17(-)_3HIS_I56V_L74A	0.388	0.08047 to 0.6956	Yes	**	0.0092
	O432-17(-)_0HIS vs. O432-17(-)_3HIS_74A	0.1996	-0.1080 to 0.5071	No	ns	0.3019
Fig 4H/S6E	O432-17(-)_0HIS vs. O432-17(-)_2HIS	0.02994	-0.2523 to 0.3122	No	ns	0.9964
	O432-17(-)_0HIS vs. O432-17(-)_3HIS_I56V	0.1138	-0.1684 to 0.3960	No	ns	0.6984
	O432-17(-)_0HIS vs. O432-17(-)_3HIS_I56V_L74A	0.07698	-0.2052 to 0.3592	No	ns	0.8962
	O432-17(-)_0HIS vs. O432-17(-)_3HIS_L74A	-0.02741	-0.3096 to 0.2548	No	ns	0.9974
Fig S6F	O432-17(+)_0HIS vs. O432-17(+)_3HIS_I56V	0.3731	0.04387 to 0.7024	Yes	*	0.0231

	O432-17(+)_0HIS vs. O432-17(+)_3HIS_I56V_L74A	0.3502	0.02095 to 0.6794	Yes	*	0.0349
	O432-17(+)_0HIS vs. O432-17(+)_3HIS_L74A	0.3564	0.02720 to 0.6857	Yes	*	0.0312
Fig S6G	O432-17(+)_0HIS vs. O432-17(+)_3HIS_I56V	0.3094	-0.02647 to 0.6454	No	ns	0.0766
	O432-17(+)_0HIS vs. O432-17(+)_3HIS_I56V_L74A	0.3221	-0.01386 to 0.6580	No	ns	0.0627
	O432-17(+)_0HIS vs. O432-17(+)_3HIS_L74A	0.2852	-0.05070 to 0.6211	No	ns	0.1109

Table 2.8: Amino acid sequences of Diphtheria-Toxin variants.

Design name	Sequence
DTA-DTT	MGSSHHHHHHGSGLVPRGSASMSDSEVNQEAKPEVKPEVKPETHINLKVSD GSSEIFFKIKKTTPLRRLMEAFAKRQGKEMDSLRFLYDGIRIQADQTPEDLDME DNDIIEAHREQIGGGADDVVDSSKSFVMENFSSYHGTKPGYVDSIQKGIQKPK SGTQGNYDDDWKGFYSTDNKYDAAGYSVDNENPLSGKAGGVVKVTYPGLT KVLALKVDNAETIKKELGLSLTEPLMEQVGTEEFIKRFGDGASRVVLSLPFAEG SSSVEYINNWEQAKALSVELEINFETRGKRGQDAMYEYMAQACAGNRVRRS VGSSLSCINLDWDVIRDKTKTKIESLKEHGPIKNKMSESPNKTVSEEKAKQYL EEFHQTALEHPELSELKTVTGTNPVFAGANYAAWAVNVAQVIDSETADNLEKT TAALSILPGIGSVMGIADGAVHHNTEEIVAQSIALSSLMVAQAIPLVGELVDIGFA AYNFVESIINLFQVVHNSYNRPAYSPGHKTQPFALVPRGSAWSHPQFEK
DTA-DTT 9xLys	MGSSHHHHHHGSGLVPRGSASMSDSEVNQEAKPEVKPEVKPETHINLKVSD GSSEIFFKIKKTTPLRRLMEAFAKRQGKEMDSLRFLYDGIRIQADQTPEDLDME DNDIIEAHREQIGGGADDVVDSSKSFVMENFSSYHGTKPGYVDSIQKGIQKPK SGTQGNYDDDWKGFYSTDNKYDAAGYSVDNENPLSGKAGGVVKVTYPGLT KVLALKVDNAETIKKELGLSLTEPLMEQVGTEEFIKRFGDGASRVVLSLPFAEG SSSVEYINNWEQAKALSVELEINFETRGKRGQDAMYEYMAQACAGNRVRRS VGSSLSCINLDWDVIRDKTKTKIESLKEHGPIKNKMSESPNKTVSEEKAKQYL EEFHQTALEHPELSELKTVTGTNPVFAGANYAAWAVNVAQVIDSETADNLEKT TAALSILPGIGSVMGIADGAVHHNTEEIVAQSIALSSLMVAQAIPLVGELVDIGFA AYNFVESIINLFQVVHNSYNRPAYSPGHKTQPFKKKKKKKKKALVPRGSAWSH PQFEK
DTA-DTT-(G4S)3-Furin-O432-17-C3(-)_3HIS_I56V_L74A	MGSSHHHHHHGSGLVPRGSASMSDSEVNQEAKPEVKPEVKPETHINLKVSD GSSEIFFKIKKTTPLRRLMEAFAKRQGKEMDSLRFLYDGIRIQADQTPEDLDME DNDIIEAHREQIGGGADDVVDSSKSFVMENFSSYHGTKPGYVDSIQKGIQKPK SGTQGNYDDDWKGFYSTDNKYDAAGYSVDNENPLSGKAGGVVKVTYPGLT KVLALKVDNAETIKKELGLSLTEPLMEQVGTEEFIKRFGDGASRVVLSLPFAEG SSSVEYINNWEQAKALSVELEINFETRGKRGQDAMYEYMAQACAGNRVRRS VGSSLSCINLDWDVIRDKTKTKIESLKEHGPIKNKMSESPNKTVSEEKAKQYL EEFHQTALEHPELSELKTVTGTNPVFAGANYAAWAVNVAQVIDSETADNLEKT TAALSILPGIGSVMGIADGAVHHNTEEIVAQSIALSSLMVAQAIPLVGELVDIGFA AYNFVESIINLFQVVHNSYNRPAYSPGHKTQPFKKKKKKKKKALVPRGSAWSH PQFEK
DTA-DTT-(G4S)3-O4	MGSSHHHHHHGSGLVPRGSASMSDSEVNQEAKPEVKPEVKPETHINLKVSD

32-17-C3(-)_3HIS_I56V_L74A	GSSEIFFKIKKTTPLRRLMEAFAKRQGKEMDSLRFLYDGIRIQADQTPEDLDME DNDIIEAHREQIGGGADDVVDSSKSFVMENFSSYHGTKPGYVDSIQKGIQKPK SGTQGNYDDDWKGFYSTDNKYDAAGYSVDNENPLSGKAGGVVKVTYPGLT KVLALKVDNAETIKKELGLSLTEPLMEQVGTEEFIKRFGDGASRVVLSLPFAEG SSSVEYINNWEQAKALSVELEINFETRGKRGQDAMYEYMAQACAGNRVRRS VGSSLSCINLDWDVIRDKTKTKIESLKEHGPIKNKMSESPNKTVSEEKAKQYL EEFHQTALEHPELSELKTVTGTNPVFAGANYAAWAVNVAQVIDSETADNLEKT TAALSILPGIGSVMGIADGAVHHNTEEIVAQSIALSSLMVAQAIPLVGELVDIGFA AYNFVESIINLFQVVHNSYNRPAYSPGHKTQPFGGGGSGGGGSGGGGSSEE KIEKLLEELTAATAELKRATASLRAITEELKKNPSEDALVEHNRAIVEHNAIVVEH NRIIATVLLAIVAAAATNEATLAADKAKEAGASEVAELAKEVLEEAEELAKENDS EEALKVVKAIADAAKAAAEAAREGKTEVAELALKVLEEAIELAKENRSEEALKV VREIARAALAAAQAAEEGKTEVAELALEVLEEAIELAKENRSEEALKVVREIAR AALAAAQAAEEGKTEVAELALEVLEQAIEAAKLQRSERALEMVREIARAALEA ARNAEGGRSDRARAILASLKVSIIVVKLKSSGTSEEEILRIVLKIIKELRKEAKEE GQSASYIATMEAEIVKAIDYALDLSGTS

Data Availability

All images and data were generated and analyzed by the authors, and will be made available by the corresponding authors (D.B., N.P.K.) upon reasonable request. Density maps have been deposited in the Electron Microscopy Data Bank under the accession number EMD-29602. Source data are provided with this manuscript.

Code Availability

Source code for the fusion, docking, and design of non-porous pH-responsive antibody nanoparticles is made available at https://github.com/erincyang/plug_design. The protocol requires compilation of the worms and rpxdock repositories, which have been made available at https://github.com/willsheffler/worms and https://github.com/willsheffler/rpxdock, respectively. Source code for generating the figures in this manuscript were written by the authors and provided in the supplementary materials.

References

Anzalone, Andrew V., Peyton B. Randolph, Jessie R. Davis, Alexander A. Sousa, Luke W. Koblan, Jonathan M. Levy, Peter J. Chen, et al. 2019. “Search-and-Replace Genome Editing without Double-Strand Breaks or Donor DNA.” *Nature* 576 (7785): 149–57.

Azuma, Yusuke, Thomas G. W. Edwardson, and Donald Hilvert. 2018. “Tailoring Lumazine Synthase Assemblies for Bionanotechnology.” *Chemical Society Reviews* 47 (10): 3543–57.

Bale, Jacob B., Shane Gonen, Yuxi Liu, William Sheffler, Daniel Ellis, Chantz Thomas, Duilio Cascio, et al. 2016. “Accurate Design of Megadalton-Scale Two-Component Icosahedral Protein Complexes.” *Science* 353 (6297): 389–94.

Banskota, Samagya, Aditya Raguram, Susie Suh, Samuel W. Du, Jessie R. Davis, Elliot H. Choi, Xiao Wang, et al. 2022. “Engineered Virus-like Particles for Efficient in Vivo Delivery of Therapeutic Proteins.” *Cell* 185 (2): 250–65.e16.

Boyken, Scott E., Mark A. Benhaim, Florian Busch, Mengxuan Jia, Matthew J. Bick, Heejun Choi, Jason C. Klima, et al. 2019. “De Novo Design of Tunable, pH-Driven Conformational Changes.” *Science* 364 (6442): 658–64.

Brunette, T. J., Fabio Parmeggiani, Po-Ssu Huang, Gira Bhabha, Damian C. Ekiert, Susan E. Tsutakawa, Greg L. Hura, John A. Tainer, and David Baker. 2015. “Exploring the Repeat Protein Universe through Computational Protein Design.” *Nature* 528 (7583): 580–84.

Butterfield, Gabriel L., Marc J. Lajoie, Heather H. Gustafson, Drew L. Sellers, Una Nattermann, Daniel Ellis, Jacob B. Bale, et al. 2017. “Evolution of a Designed Protein Assembly Encapsulating Its Own RNA Genome.” *Nature* 552 (7685): 415–20.

Cannon, Kevin A., Vy N. Nguyen, Christian Morgan, and Todd O. Yeates. 2020. “Design and Characterization of an Icosahedral Protein Cage Formed by a Double-Fusion Protein Containing Three Distinct Symmetry Elements.” *ACS Synthetic Biology* 9 (3): 517–24.

Chen, H., S. Zhang, C. Xu, and G. Zhao. 2016. “Engineering Protein Interfaces Yields Ferritin Disassembly and Reassembly under Benign Experimental Conditions.” *Chemical Communications* 52 (46): 7402–5.

Chen, Peter J., Jeffrey A. Hussmann, Jun Yan, Friederike Knipping, Purnima Ravisankar, Pin-Fang Chen, Cidi Chen, et al. 2021. “Enhanced Prime Editing Systems by Manipulating Cellular Determinants of Editing Outcomes.” *Cell* 184 (22): 5635–52.e29.

Czapar, Anna E., and Nicole F. Steinmetz. 2017. “Plant Viruses and Bacteriophages for Drug Delivery in Medicine and Biotechnology.” *Current Opinion in Chemical Biology* 38 (June): 108–16.

Dalmau, Mercè, Sierin Lim, and Szu-Wen Wang. 2009. “pH-Triggered Disassembly in a Caged Protein Complex.” *Biomacromolecules* 10 (12): 3199–3206.

Dimitrov, Dimiter S. 2004. “Virus Entry: Molecular Mechanisms and Biomedical Applications.” *Nature Reviews. Microbiology* 2 (2): 109–22.

Divine, Robby, Ha V. Dang, George Ueda, Jorge A. Fallas, Ivan Vulovic, William Sheffler, Shally Saini, et al. 2021. “Designed Proteins Assemble Antibodies into Modular Nanocages.” *Science* 372 (6537). https://doi.org/10.1126/science.abd9994.

Douglas, Trevor, and Mark Young. 2006. “Viruses: Making Friends with Old Foes.” *Science* 312 (5775): 873–75.

Edwardson, Thomas G. W., Mikail D. Levasseur, and Donald Hilvert. 2021. “The OP Protein Cage: A Versatile Molecular Delivery Platform.” *Chimia* 75 (4): 323–28.

Edwardson, Thomas G. W., Takahiro Mori, and Donald Hilvert. 2018. “Rational Engineering of a Designed Protein Cage for siRNA Delivery.” *Journal of the American Chemical Society* 140 (33): 10439–42.

Fallas, Jorge A., George Ueda, William Sheffler, Vanessa Nguyen, Dan E. McNamara,

Banumathi Sankaran, Jose Henrique Pereira, et al. 2017. “Computational Design of Self-Assembling Cyclic Protein Homo-Oligomers.” *Nature Chemistry* 9 (4): 353–60.
Hou, Xucheng, Tal Zaks, Robert Langer, and Yizhou Dong. 2021. “Lipid Nanoparticles for mRNA Delivery.” *Nature Reviews. Materials* 6 (12): 1078–94.
Hsia, Yang, Jacob B. Bale, Shane Gonen, Dan Shi, William Sheffler, Kimberly K. Fong, Una Nattermann, et al. 2016. “Design of a Hyperstable 60-Subunit Protein Icosahedron.” *Nature*. https://doi.org/10.1038/nature18010.
Hsia, Yang, Rubul Mout, William Sheffler, Natasha I. Edman, Ivan Vulovic, Young-Jun Park, Rachel L. Redler, et al. 2021. “Design of Multi-Scale Protein Complexes by Hierarchical Building Block Fusion.” *Nature Communications* 12 (1): 2294.
Hsu, Jonathan Y., Julian Grünewald, Regan Szalay, Justine Shih, Andrew V. Anzalone, Kin Chung Lam, Max W. Shen, et al. 2021. “PrimeDesign Software for Rapid and Simplified Design of Prime Editing Guide RNAs.” *Nature Communications* 12 (1): 1034.
Kagan, B. L., A. Finkelstein, and M. Colombini. 1981. “Diphtheria Toxin Fragment Forms Large Pores in Phospholipid Bilayer Membranes.” *Proceedings of the National Academy of Sciences of the United States of America* 78 (8): 4950–54.
Kilchrist, Kameron V., Somtochukwu C. Dimobi, Meredith A. Jackson, Brian C. Evans, Thomas A. Werfel, Eric A. Dailing, Sean K. Bedingfield, Isom B. Kelly, and Craig L. Duvall. 2019. “Gal8 Visualization of Endosome Disruption Predicts Carrier-Mediated Biologic Drug Intracellular Bioavailability.” *ACS Nano* 13 (2): 1136–52.
Kim, Mihee, Yecheol Rho, Kyeong Sik Jin, Byungcheol Ahn, Sungmin Jung, Heesoo Kim, and Moonhor Ree. 2011. “pH-Dependent Structures of Ferritin and Apoferritin in Solution: Disassembly and Reassembly.” *Biomacromolecules* 12 (5): 1629–40.
King, Neil P., Jacob B. Bale, William Sheffler, Dan E. McNamara, Shane Gonen, Tamir Gonen, Todd O. Yeates, and David Baker. 2014. “Accurate Design of Co-Assembling Multi-Component Protein Nanomaterials.” *Nature* 510 (7503): 103–8.
Lai, Yen-Ting, Neil P. King, and Todd O. Yeates. 2012. “Principles for Designing Ordered Protein Assemblies.” *Trends in Cell Biology* 22 (12): 653–61.
Lavelle, Laurence, Jean-Philippe Michel, and Mari Gingery. 2007. “The Disassembly, Reassembly and Stability of CCMV Protein Capsids.” *Journal of Virological Methods* 146 (1-2): 311–16.
Lawrence, M. C., and P. M. Colman. 1993. “Shape Complementarity at Protein/protein Interfaces.” *Journal of Molecular Biology* 234 (4): 946–50.
Leman, Julia Koehler, Brian D. Weitzner, Steven M. Lewis, Jared Adolf-Bryfogle, Nawsad Alam, Rebecca F. Alford, Melanie Aprahamian, et al. 2020. “Macromolecular Modeling and Design in Rosetta: Recent Methods and Frameworks.” *Nature Methods* 17 (7): 665–80.
Levasseur, Mikail D., Shiksha Mantri, Takahiro Hayashi, Maria Reichenbach, Svenja Hehn, Ying Waeckerle-Men, Pål Johansen, and Donald Hilvert. 2021. “Cell-Specific Delivery Using an Engineered Protein Nanocage.” *ACS Chemical Biology* 16 (5): 838–43.
Lönn, Peter, Apollo D. Kacsinta, Xian-Shu Cui, Alexander S. Hamil, Manuel Kaulich, Khirud Gogoi, and Steven F. Dowdy. 2016. “Enhancing Endosomal Escape for Intracellular Delivery of Macromolecular Biologic Therapeutics.” *Scientific Reports* 6 (September): 32301.
Martens, Thomas F., Katrien Remaut, Jo Demeester, Stefaan C. De Smedt, and Kevin Braeckmans. 2014. “Intracellular Delivery of Nanomaterials: How to Catch Endosomal Escape in the Act.” *Nano Today* 9 (3): 344–64.
Mitchell, Michael J., Margaret M. Billingsley, Rebecca M. Haley, Marissa E. Wechsler, Nicholas A. Peppas, and Robert Langer. 2020. “Engineering Precision Nanoparticles for Drug Delivery.” *Nature Reviews. Drug Discovery*. https://doi.org/10.1038/s41573-020-0090-8.
Nelson, James W., Peyton B. Randolph, Simon P. Shen, Kelcee A. Everette, Peter J. Chen, Andrew V. Anzalone, Meirui An, et al. 2022. “Engineered pegRNAs Improve Prime Editing

Efficiency." *Nature Biotechnology* 40 (3): 402–10.
Padilla, J. E., C. Colovos, and T. O. Yeates. 2001. "Nanohedra: Using Symmetry to Design Self Assembling Protein Cages, Layers, Crystals, and Filaments." *Proceedings of the National Academy of Sciences of the United States of America* 98 (5): 2217–21.
"pegLIT." n.d. Accessed April 24, 2023. https://peglit.liugroup.us/.
"pegRNA Design GuideFinal_022620.pdf." n.d. Google Docs. Accessed April 24, 2023. https://drive.google.com/file/d/1xHWhscwIsyxUUdyGO-BZ8kleYUVmpBJA/view.
"PE Resources - Liu Group." 2020. Liu Group -. Liu Group. February 7, 2020. https://www.liugroup.us/pe-resources/.
Punjani, Ali, John L. Rubinstein, David J. Fleet, and Marcus A. Brubaker. 2017. "cryoSPARC: Algorithms for Rapid Unsupervised Cryo-EM Structure Determination." *Nature Methods* 14 (3): 290–96.
Roux, K. H., L. Strelets, and T. E. Michaelsen. 1997. "Flexibility of Human IgG Subclasses." *Journal of Immunology* 159 (7): 3372–82.
Scholefield, Janine, and Patrick T. Harrison. 2021. "Prime Editing - an Update on the Field." *Gene Therapy* 28 (7-8): 396–401.
Senzel, L., M. Gordon, R. O. Blaustein, K. J. Oh, R. J. Collier, and A. Finkelstein. 2000. "Topography of Diphtheria Toxin's T Domain in the Open Channel State." *The Journal of General Physiology* 115 (4): 421–34.
Seo, Junyoung, Jae Do Yoo, Minseong Kim, Gayong Shim, Yu-Kyoung Oh, Rang-Woon Park, Byungheon Lee, In-San Kim, and Soyoun Kim. 2021. "Fibrinolytic Nanocages Dissolve Clots in the Tumor Microenvironment, Improving the Distribution and Therapeutic Efficacy of Anticancer Drugs." *Experimental & Molecular Medicine* 53 (10): 1592–1601.
Sheffler, William, Erin C. Yang, Quinton Dowling, Yang Hsia, Chelsea N. Fries, Jenna Stanislaw, Mark Langowski, et al. 2022. "Fast and Versatile Sequence-Independent Protein Docking for Nanomaterials Design Using RPXDock." *bioRxiv*. https://doi.org/10.1101/2022.10.25.513641.
Somiya, Masaharu, and Shun 'ichi Kuroda. 2021. "Real-Time Luminescence Assay for Cytoplasmic Cargo Delivery of Extracellular Vesicles." *Analytical Chemistry* 93 (13): 5612–20.
Steinmetz, Nicole F., Sierin Lim, and Frank Sainsbury. 2020. "Protein Cages and Virus-like Particles: From Fundamental Insight to Biomimetic Therapeutics." *Biomaterials Science* 8 (10): 2771–77.
Studier, F. William. 2005. "Protein Production by Auto-Induction in High Density Shaking Cultures." *Protein Expression and Purification* 41 (1): 207–34.
Sun, Xuanrong, Yulu Hong, Yubei Gong, Shanshan Zheng, and Dehui Xie. 2021. "Bioengineered Ferritin Nanocarriers for Cancer Therapy." *International Journal of Molecular Sciences* 22 (13). https://doi.org/10.3390/ijms22137023.
Sutter, Markus, Daniel Boehringer, Sascha Gutmann, Susanne Günther, David Prangishvili, Martin J. Loessner, Karl O. Stetter, Eilika Weber-Ban, and Nenad Ban. 2008. "Structural Basis of Enzyme Encapsulation into a Bacterial Nanocompartment." *Nature Structural & Molecular Biology* 15 (9): 939–47.
Tetter, Stephan, Naohiro Terasaka, Angela Steinauer, Richard J. Bingham, Sam Clark, Andrew J. P. Scott, Nikesh Patel, et al. 2021. "Evolution of a Virus-like Architecture and Packaging Mechanism in a Repurposed Bacterial Protein." *Science* 372 (6547): 1220–24.
Tyka, Michael D., Daniel A. Keedy, Ingemar André, Frank Dimaio, Yifan Song, David C. Richardson, Jane S. Richardson, and David Baker. 2011. "Alternate States of Proteins Revealed by Detailed Energy Landscape Mapping." *Journal of Molecular Biology* 405 (2): 607–18.
Van de Steen, Alexander, Rana Khalife, Noelle Colant, Hasan Mustafa Khan, Matas Deveikis, Saverio Charalambous, Clare M. Robinson, et al. 2021. "Bioengineering Bacterial

Encapsulin Nanocompartments as Targeted Drug Delivery System." *Synthetic and Systems Biotechnology* 6 (3): 231–41.
Votteler, Jörg, Cassandra Ogohara, Sue Yi, Yang Hsia, Una Nattermann, David M. Belnap, Neil P. King, and Wesley I. Sundquist. 2016. "Designed Proteins Induce the Formation of Nanocage-Containing Extracellular Vesicles." *Nature*. https://doi.org/10.1038/nature20607.
Wang, Dan, Phillip W. L. Tai, and Guangping Gao. 2019. "Adeno-Associated Virus Vector as a Platform for Gene Therapy Delivery." *Nature Reviews. Drug Discovery* 18 (5): 358–78.
Wang, Jie, and Erwin London. 2009. "The Membrane Topography of the Diphtheria Toxin T Domain Linked to the a Chain Reveals a Transient Transmembrane Hairpin and Potential Translocation Mechanisms." *Biochemistry* 48 (43): 10446–56.
Wargacki, Adam J., Tobias P. Wörner, Michiel van de Waterbeemd, Daniel Ellis, Albert J. R. Heck, and Neil P. King. 2021. "Complete and Cooperative in Vitro Assembly of Computationally Designed Self-Assembling Protein Nanomaterials." *Nature Communications* 12 (1): 883.
Zhang, Kai. 2016. "Gctf: Real-Time CTF Determination and Correction." *Journal of Structural Biology* 193 (1): 1–12.
Zivanov, Jasenko, Takanori Nakane, Björn O. Forsberg, Dari Kimanius, Wim Jh Hagen, Erik Lindahl, and Sjors Hw Scheres. 2018. "New Tools for Automated High-Resolution Cryo-EM Structure Determination in RELION-3." *eLife* 7 (November). https://doi.org/10.7554/eLife.42166.
Zuris, John A., David B. Thompson, Yilai Shu, John P. Guilinger, Jeffrey L. Bessen, Johnny H. Hu, Morgan L. Maeder, J. Keith Joung, Zheng Yi Chen, and David R. Liu. 2015. "Cationic Lipid-Mediated Delivery of Proteins Enables Efficient Protein-Based Genome Editing in Vitro and in Vivo." *Nature Biotechnology* 33 (1): 73–80.

Chapter 3: Increasing computational protein design literacy through cohort-based learning for undergraduate students

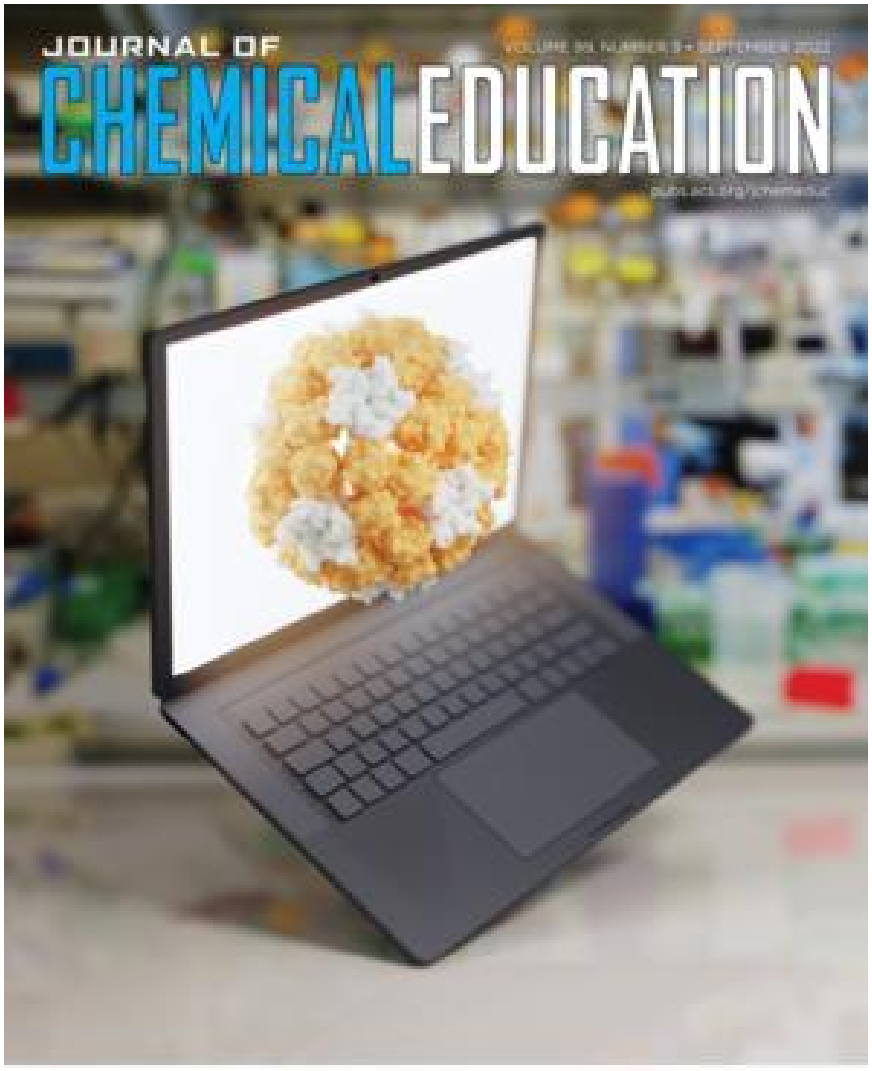
Adapted from:
Yang, Erin C., Robby Divine, Christine S. Kang, Sidney Chan, Elijah Arenas, Zoe Subol, Peter Tinker, et al. 2022. “Increasing Computational Protein Design Literacy through Cohort-Based Learning for Undergraduate Students.” *Journal of Chemical Education* 99 (9): 3177–86.

This is an undergraduate research program piloted at IPD,
Fully remote, cohort-based, and filled with protein biochemistry.
From COVID-19, the university had closed,
Leaving a hindrance in undergraduate research, a problem posed.

We designed a research program (JUPITER) based on protein design,
And employed undergraduate students, a total of nine.
We looked at nanoparticles and how they behaved,
Whether some assemble or don’t, or how they could be saved.

We meet weekly to learn Rosetta, Python, and Bash,
Made interface mutations–but watched for a clash.
They analyzed nanoparticles on a single and global scale,
And learned to work together–as a team, they would not fail.

Their curiosity led to hypothesis testing, a critical skill they learned,
Within a year, they secured jobs, research, and graduate school, well-deserved and well-earned.
A special thanks to Justin Siegel & Co, for their inspiration and time,
We sought advice from their experience with a program designing enzymes.
JUPITER spanned 2020-2021, making 9/9/22 a very special day,
As our cover art had been selected for display.


JOURNAL OF
CHEMICAL EDUCATION
pubs.acs.org/jchemed
ACS Publications
Most Trusted. Most Cited. Most Read.
www.divched.org www.acs.org

Abstract

Undergraduate research experiences can improve student success in graduate education and STEM careers. During the COVID-19 pandemic, undergraduate researchers at our institution and many others lost their work-study research positions due to interruption of in-person research activities. This imposed a financial burden on the students and eliminated an important learning opportunity. To address these challenges, we created a paid, fully-remote, cohort-based research curriculum in computational protein design. Our curriculum used existing protein design methods as a platform to first educate and train undergraduate students and then to test research hypotheses. In the first phase, students learned computational methods to assess the stability of designed protein assemblies. In the second phase, students used a larger dataset to identify factors that could improve the accuracy of current protein design algorithms. This cohort-based program created valuable new research opportunities for undergraduates at our institute and enhanced the undergraduates' feeling of connection with the lab. Students learned transferable and useful skills such as literature review, programming basics, data analysis, hypothesis testing, and scientific communication. Our program provides a model of structured computational research training opportunities for undergraduate researchers in any field for organizations looking to expand educational access.

Introduction

Participation in hands-on research fuels students' interest in pursuing STEM graduate education and careers(Russell, Hancock, and McCullough 2007; Bauer and Bennett 2003; Hathaway, Nagda, and Gregerman 2002) and also heightens graduate education performance in relevant research skills.(Gilmore et al. 2015) Several studies have correlated research experiences with increased retention,(Nagda et al. 1998) improved academic performance,(Scott-Johnson et al. 2008) and persistence in STEM courses.(A. E. L. Barlow and Villarejo 2004) Through undergraduate research experiences, students are exposed to laboratory techniques, analytical thinking, autonomous problem solving, and collaboration. Students' motivation to learn is also enhanced by research experiences.(Seymour et al. 2004; Hunter, Laursen, and Seymour 2007; Lopatto 2007) Exposure to non-classroom science, guidance from mentors in the lab environment, and peer relationships during their research experience are particularly valuable for students learning about STEM careers at minority-serving institutions.(Carpi et al. 2017) However, all of these benefits are limited by mentor and resource availability.

Prior to the COVID-19 pandemic, undergraduate research at the Institute for Protein Design at the University of Washington was based on an apprenticeship model.(Russell, Hancock, and McCullough 2007; Junge et al. 2010; Hunter et al. 2010; Lopatto 2007, 2017) Undergraduates spent two quarters learning about the organization of the Institute and providing general lab support to researchers. After these initial two quarters, undergraduates were granted the opportunity to conduct an independent research project under the mentorship of a graduate student or postdoctoral researcher at the Institute. Projects ranged from solely computational to solely experimental, depending on the project scope decided between the mentor and trainee. Each project was tailored to the undergraduate, who was encouraged to present their work at a university-wide Undergraduate Research Symposium held at the end of each academic year.("Undergraduate Research Symposium" n.d.) While the apprenticeship model is a well-established approach, it is limited by mentor availability, research ideas with an appropriate scope for an undergraduate research project, and the mentee's graduation timeline. The apprenticeship model is particularly difficult at institutions where research is not central to the institutional mission, especially for students who do not actively seek out opportunities.(Wei and Woodin 2011) Alternatives to the apprenticeship model that still enable student creativity and independence can help provide diverse, high-quality research opportunities to a wider variety of students.

Due to the COVID-19 pandemic, many undergraduates, including those at the University of Washington, spent nearly a full year learning remotely, and undergraduate researchers lost their on-campus work-study research positions.(Forrester 2021; "As We Face Challenges Ahead, the UW Will Put People First" n.d.) We set out to re-employ the work-study undergraduate researchers at our Institute by offering a virtual research opportunity, which have been reported beneficial and valuable for students in other undergraduate research programs.(Erickson et al. 2022; Cohen et al. 2021) However, we needed a different approach than our previous apprenticeship model, as the frequent 1:1 contact central to that model was impractical to implement remotely and too few mentors were free to individually work with each

undergraduate at our Institute. Previous cohort-based research courses in computational protein design, in which multiple students are taught by a smaller number of mentors, enabled them to accommodate greater numbers of students and have shown significant success in increasing students' enthusiasm for biochemistry and computational biology.(Tantillo et al. 2019; Le et al. 2021; Vater et al. 2021) We adopted this cohort-based model and curated a completely remote computational curriculum that allowed students to take ownership of their portion of a team research project while learning from each other. We called this program *Jobs for Undergraduates in PandemIc Times Emergency Response* (JUPITER). Through this pilot program, we provided tools for undergraduates to understand basic protein structure concepts, coding conventions, and data analysis, while developing transferable skills such as critical thinking, troubleshooting, collaboration, communication, and motivation. The JUPITER program was designed in two phases: in the first phase, students were taught the concepts behind the protein design process; in the second phase, students reinforced their learning by developing and implementing these concepts. We believe that they will use both the technical and transferable skills they learned from this course in their professional careers.

Program Overview

Nine undergraduate students participated in JUPITER. Among these students, 4 identified as non-White, 3 identified as members of the LGBTQIA+ community, 5 identified as non-male, and 5 were enrolled in some form of need-based financial aid with the university. All students were majoring in STEM fields at the University of Washington, on track to graduate between the years of 2020 and 2022, and had previously held jobs at the Institute for Protein Design which they lost due to the COVID-19 pandemic and associated University policies. To accommodate for their lost pay, quarterly stipends were provided for students' work and time with the expectation that they work 10 hours per week on average.

In our cohort-based model (**Figure 3.1A**), the students learned from each other in a collaborative environment while maintaining parallel research directions, following a teaching and research program with two phases. In the first phase, the students were taught how to read and understand a scientific paper and received an introduction to protein structure and design terminology (week 1);(Alford et al. 2017; King et al. 2012, 2014; Bale et al. 2016; Hsia et al. 2016) were trained to use protein visualization and design software such as PyMOL and FoldIt (weeks 2-3);(Kleffner et al. 2017; Schrödinger, LLC 2015) and practiced computational research approaches including accessing remote computational resources, performing a protein design and analysis protocol on a single individually assigned protein nanoparticle (described in detail in the following section),(King et al. 2014; Hsia et al. 2016; Bale et al. 2016; Ueda et al. 2020) and conducting basic data analysis (weeks 4-7). This phase ended with individual presentations to the other program participants (weeks 7-10, **Figure 3.1B**). The undergraduates met with three regular advisors and occasional guest lecturers via Zoom for a one hour-long lecture and additional optional office hours to troubleshoot homework assignments. Assignments were designed to help the undergraduates learn how to evaluate nanoparticle interfaces over the course of the quarter. Following these initial 10 weeks, one student secured a full-time research position in another computational biology group at the University of Washington, citing the

JUPITER program as a key driver of their motivation to pursue research. Given their ability to understand and apply the phase one material, as qualitatively assessed by the mentors, the remaining students were invited to continue onto a second phase of research.

In the second phase, we embarked on a longer-term research project with our undergraduate cohort, training them in critical thinking. Students worked together in small teams to generate hypotheses on what influenced predicted assembly phenotype, and in the process generated their own datasets, created data analysis pipelines, and presented their work to an audience of over 90 members of the Institute for Protein Design (**Figure 3.1C**). A major advantage of the JUPITER phase 2 program was that students could probe different questions regarding the mutational effects of nanoparticle interfaces using a single dataset they collectively generated. The undergraduates used Jupyter Notebooks to write Python scripts for data analyses and visualize their results.(Kluyver et al. 2016) Teams met weekly with two mentors to troubleshoot and uncover alternative analysis methods and global trends across the library of nanoparticles. The students' final Institute-wide presentations clearly demonstrated their greater understanding of protein structure, Rosetta scripting, Python and Bash coding, and broad scientific thinking. Among the 9 undergraduates that had started in our pilot program, 6 secured full-time research/higher education positions in both computational and wet lab projects, 1 applied to medical school, and 2 decided to pursue alternative opportunities in STEM.

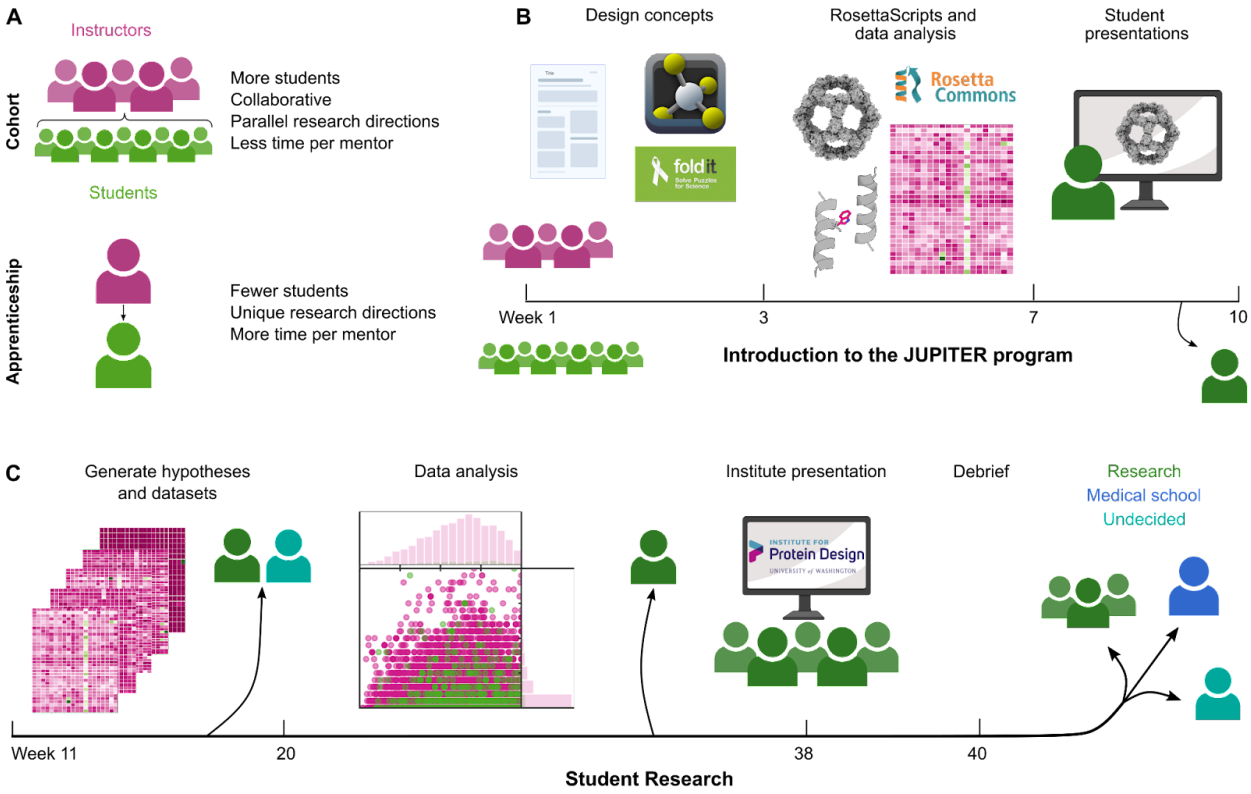


Figure 3.1. Overview of the JUPITER model and program.
A. The cohort-based model allows mentors to instruct more students. This model encourages more collaboration between students, compared to a one-on-one apprenticeship model. **B.** In the first 10 weeks of JUPITER, students first learned design concepts and computational approaches. These approaches were then applied to a case study that each student presented to the rest of the cohort. **C.** Over the next several months, students applied their knowledge to generate data on a large design set, proposed their own hypotheses and data processing pipelines required to approach them, and presented their results in an Institute-wide seminar.

Research overview

Project background

We designed the JUPITER curriculum to address an ongoing scientific challenge: improving the success rate of current computational methods for the *de novo* design of self-assembling protein complexes using the Rosetta software suite.(Fallas et al. 2017; Ueda et al. 2020; Leaver-Fay et al. 2011; Gonen et al. 2015; Ben-Sasson et al. 2021; Shen et al. 2019; Sahasrabuddhe et al. 2018; King et al. 2012, 2014; Hsia et al. 2016; Bale et al. 2016; Wargacki et al. 2021; Khmelinskaia, Wargacki, and King 2021) These methods focus on docking oligomeric protein building blocks into target symmetric architectures (e.g., icosahedral point group symmetry), followed by protein-protein interface design to generate low-energy interfaces between the building blocks that drive assembly specifically to the target structure. While such assemblies are well-suited for molecular encapsulation or display applications such as cellular delivery and vaccine development,(Butterfield et al. 2017; Marcandalli et al. 2019; Walls et al. 2020; Boyoglu-Barnum et al. 2021; Divine et al. 2021; Khmelinskaia, Wargacki, and King 2021) current computational models do not allow us to confidently predict the success or stability of design models. Indeed, many designs aggregate or fail to assemble, due to misfolding or suboptimal assembly conditions.(Wargacki et al. 2021; Magliery 2015) Protein designers often test tens to hundreds of designs to find a handful of nanoparticles that assemble to the desired structure.(Khmelinskaia, Wargacki, and King 2021; Lai, King, and Yeates 2012; Cannon et al. 2020)

To better understand the pitfalls of current nanoparticle design methods, we aimed to computationally distinguish between successful and unsuccessful protein nanoparticle designs. While several published efforts to stabilize monomeric proteins demonstrated high success rates based on approaches using phylogenetic analysis, structure-based rational design, or sequence-based design,(Lehmann et al. 2000; Guerois and de la Paz 2006; Borgo and Havranek 2012; Jacak, Leaver-Fay, and Kuhlman 2012; Lawrence, Phillips, and Liu 2007; Goldenzweig et al. 2016; Listov et al. 2021) none of these approaches addressed the stability of interfaces between multiple interacting proteins and its effect on design success.(K. A. Barlow et al. 2018) We thus sought to test a recently developed computational pipeline—protein Strain Unsatisfactoriness and Frustration findER (pSUFER)—for local sequence optimality evaluation on previously designed protein assemblies of known assembly phenotype.(Listov et al. 2021;

pSUFER: Protein Strain, Unsatisfactoriness, and Frustration findER n.d., *LSO: Local Sequence Optimality* n.d.) pSUFER is a Rosetta software suite-based protocol that estimates the effect of point mutations at a particular position in a protein. We tested the pSUFER protocol on a set of 809 designed protein nanoparticle models in order to: (1) determine the effect of point mutations at the previously designed protein-protein interfaces on free energy; and (2) to identify key residue positions predicted to weaken/strengthen the protein-protein interfaces.

Phase 1 - Introduction to the pSUFER protocol

After becoming familiar with protein design in general and nanoparticle design concepts specifically, each student applied the pSUFER protocol to one assigned nanoparticle. For this, students used PyMOL to extract the asymmetric unit of their nanoparticle and performed design model minimization with the Rosetta all-atom energy function (**Figure 3.2A**). Students then applied the FilterScan Mover(Whitehead et al. 2012), which scans all possible mutations at a user-defined residue position and computes the difference in a user-defined metric. More specifically, the students computed the difference in Gibbs free energy (dG) between the wild-type amino acid and every possible point mutant (change in free energy, or ddG) at every residue position they identified as belonging to their nanoparticle interface (**Figure 3.2B**).(Alford et al. 2017; Whitehead et al. 2012; Schrödinger, LLC 2015) A decrease in free energy was used to define mutations that were more favorable than wild-type (ddG < 0 Rosetta Energy Units, REU), and an increase in free energy was used to define mutations that were less favorable than wild-type (ddG > 0 REU) (**Figure 3.2C**). By graphing the calculated ddG for each mutation at each position in a barplot, the students comprehensively identified amino acid identities that were predicted to stabilize or destabilize the protein-protein interface (**Figure 3.2D**). Residue positions with more than two favorable mutations were deemed “frustrated” (**Figure 3.2E**), as these positions were more mutable than positions with fewer favorable mutations and thus not yet fully optimized. By contrast, residue positions with up to two favorable mutations were deemed “sensitive” positions. Global analysis of the effects of interface mutations through ddG heatmaps allowed the students to identify specific amino acid identities that were unfavorable at many positions or single positions with many unfavorable mutations (**Figure 3.2F**). Through phase one, students were taught literature review, programming basics, computational protein analysis, and technical communication, using a model that could easily be extended to teach similar skills in other research contexts. All teaching materials from phase 1 and example scripts to execute the pSUFER protocol on a single protein nanoparticle and analyze the resulting data can be downloaded at the following link: https://files.ipd.uw.edu/pub/JUPITER/JUPITER.tar.gz.

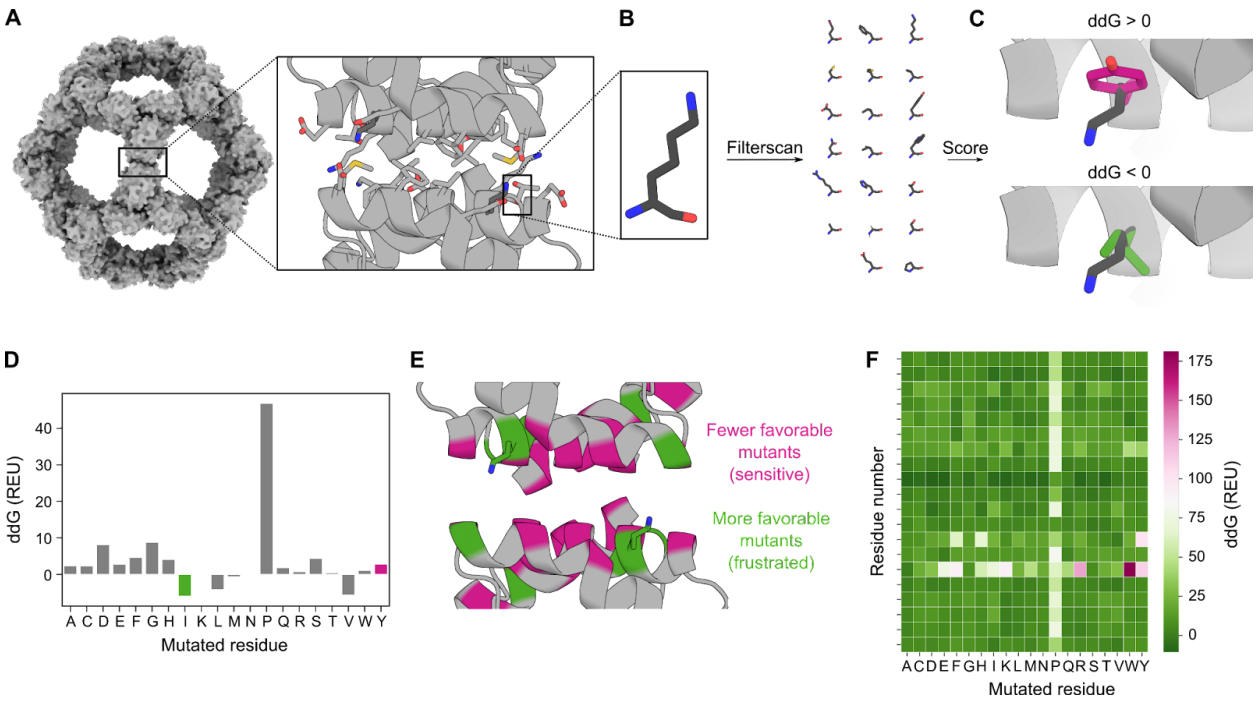


Figure 3.2. JUPITER's first phase of research focused on studying individual previously designed successful nanoparticles.
Analysis of I3-01(Hsia et al. 2016) through the pSUFER protocol. **A-C.** pSUFER overview: **A.** The interface between two nanoparticle subunits of a relaxed design model, here with side chains highlighted in stick representation, is selected using ResidueSelectors; **B.** each interface residue (here, K23) is computationally mutated to all possible amino acids; and **C.** the free energy difference compared to the wild-type (ddG) is calculated. **D-F.** Data visualization by the students. **D.** A representative barplot showing the energy difference for each possible mutation at a given sequence position (K32), highlighting a favorable mutation (K23I, green) and an unfavorable mutation (K23Y, pink). **E.** PyMOL visualization of pSUFER scores for the whole interface by differential coloring: *frustrated* residue positions (more than two favorable mutations) are in green while *sensitive* positions (up to two favorable mutations) are in pink. **F.** A heatmap compiling the results: columns represent each possible mutation, and rows represent each interface residue position.

Phase 2 - Experimental design and results

In phase 2, the undergraduate cohort began to expand the pSUFER protocol they learned in phase 1 to all the nanoparticle designs ever experimentally tested at the Institute for Protein Design between 2012 and July 2020, totalling 809 nanoparticles across 11 icosahedral, octahedral, or tetrahedral architectures. Of these 809 designs, 64 have been experimentally confirmed to adopt the target architecture ("working" designs), while the other 745 apparently failed to assemble ("non-working" designs). Using the pSUFER protocol, the undergraduates generated a single dataset for further analysis, aiming (1) to identify trends that could provide insight into why certain designs may or may not assemble as designed and (2) to predict residue positions where mutations could be introduced to substantially weaken or strengthen protein-protein interfaces across all nanoparticles.

In groups, the students developed computational experiments to probe three hypotheses from a collectively generated pSUFER dataset. One team hypothesized that working nanoparticles would have larger losses in predicted binding energy upon mutation of interface residues compared to non-working nanoparticles (**Figure 3.3A, left**). This group determined the number of mutations at each position of every nanoparticle interface with favorable ddG scores (ddG < 0). Next, the undergraduates compared the fraction of interface positions per nanoparticle that were sensitive or frustrated between working and non-working nanoparticles. They observed no significant difference in the fraction of sensitive and frustrated interface positions between working and non-working nanoparticles (**Figure 3.3A, right**). Similar analyses were performed on each individual symmetric architecture with a similar outcome.

The second group hypothesized that positions closer to the center of mass of a nanoparticle interface, by being buried in the designed hydrophobic core, contribute most directly to the protein-protein interface and therefore would be more sensitive (have fewer favorable mutations) compared to positions further from the interface center of mass (**Figure 3.3B, left**). This group used the pSUFER dataset and the atomic coordinates of each interface position in a nanoparticle to estimate the center of mass of each nanoparticle interface and calculated the number of favorable mutations for each position relative to the position's Cβ distance from the center of mass of the interface. The group corrected for different interface sizes by normalizing all absolute distances to the largest distance within each nanoparticle interface. Following normalization, the students observed that positions approximately halfway between the center of the interface and the farthest residue from the interface tended to be more frustrated, i.e. to have the most number of favorable mutations per nanoparticle (**Figure 3.3B, right**).

Finally, the third team hypothesized that amino acid mutations would score more favorably if they were mutated to amino acids with properties—such as size or charge—similar to those of the wild-type residue, as opposed to amino acids with different properties (**Figure 3.3C, left**). Consistent with this hypothesis, the number of leucine positions with favorable ddG scores upon mutation to isoleucine were higher than the number of leucine positions with favorable ddG scores upon mutation to asparagine (**Figure 3.3C, right**). Even though leucine, isoleucine, and asparagine are similar in molecular weight, leucine and isoleucine are similar in property as they are both nonpolar residues while asparagine is a polar residue.

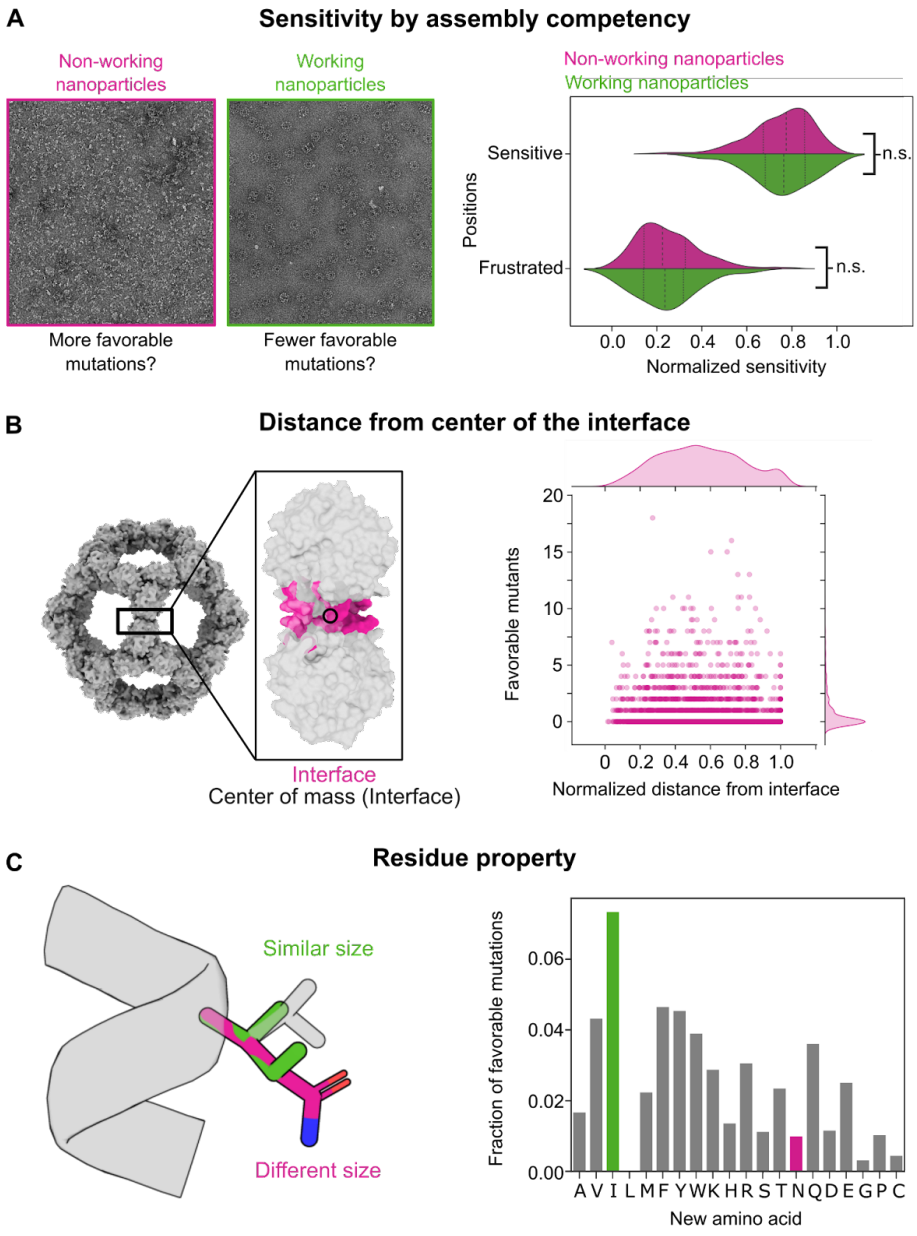


Figure 3.3. JUPITER's second phase had students create and test their own hypotheses.

A. The first group hypothesized that working nanoparticles would be more sensitive (larger predicted losses in Gibbs free energy) to mutations than non-working nanoparticles. However, they observed no significant difference between the mutability ranges of all working vs. all non-working nanoparticles across sensitive, moderate, and frustrated positions (**Table 1**). **B.** The second group hypothesized that residues near the center of the interface would be less mutable than those further away; for I3 nanoparticles, they found that positions approximately halfway between the center of mass of the interface and the furthest residue from the interface to be the most mutable. Each circle on the graph represents a single position of a single nanoparticle. **C.** The third group hypothesized that mutations to residues with similar properties (e.g., size, polarity, or charge) would more often be favorable than mutations to residues of differing properties; e.g., for leucines (L) on all nanoparticles, they found that

mutations to hydrophobic residues of similar size and characteristics, such as isoleucine (I), were more favorable than to residues introducing larger changes, such as asparagine (N).

Table 3.1. Statistical information for comparing position sensitivity by assembly competency. Two-tailed unpaired t-tests were used to compare means (**Figure 3A, right**) with $\alpha = 0.05$ in Graphpad Prism Software.

Experiment	Condition	n	Mean	Summary	Adjusted p value	t	df
Sensitive positions	Non-working nanoparticles	745	0.7517	n.s.	0.8444	0.1963	807
	Working nanoparticles	64	0.7553	n.s.			
Frustrated positions	Non-working nanoparticles	745	0.2483	n.s.	0.8444	0.1963	807
	Working nanoparticles	64	0.2447	n.s.			

Following our teaching model, the undergraduates could further easily generate three additional datasets to evaluate their hypotheses using simple tweaks to the pSUFER protocol. First, the undergraduates expanded the pSUFER protocol to residues neighboring but not directly participating in the previously analyzed nanoparticle interface (**Figure 3.4A**). While the number of neighboring residues was lower than that of interface residues, residues in both groups were generally most tolerant of mutations further away from the interface, without many differences observed between groups.

The undergraduates also introduced simple modifications to how ddG scores were evaluated. The score function initially used was an all-atom Rosetta energy score function, dominated by pairwise atomic interactions between protein backbones and side chains, including packing interactions, electrostatic interactions, and implicit solvation(Alford et al. 2017). The undergraduates additionally tested pilot burial- and distance-based score terms added to the original score function (*manuscript in preparation*). The burial score term penalized unfavorable polar interactions based on how deeply atoms were buried in the core or interface of the protein, while the distance-based score term rewarded more favorable electrostatic interactions within an optimal distance (**Figure 3.4B**). The undergraduates added just the burial-dependent score term to generate one dataset and both the burial- and distance-dependent score term to generate the second dataset. By comparing these new datasets to the original one, the undergraduates observed significant differences in the number of favorable mutations across amino acids, especially among polar amino acids.

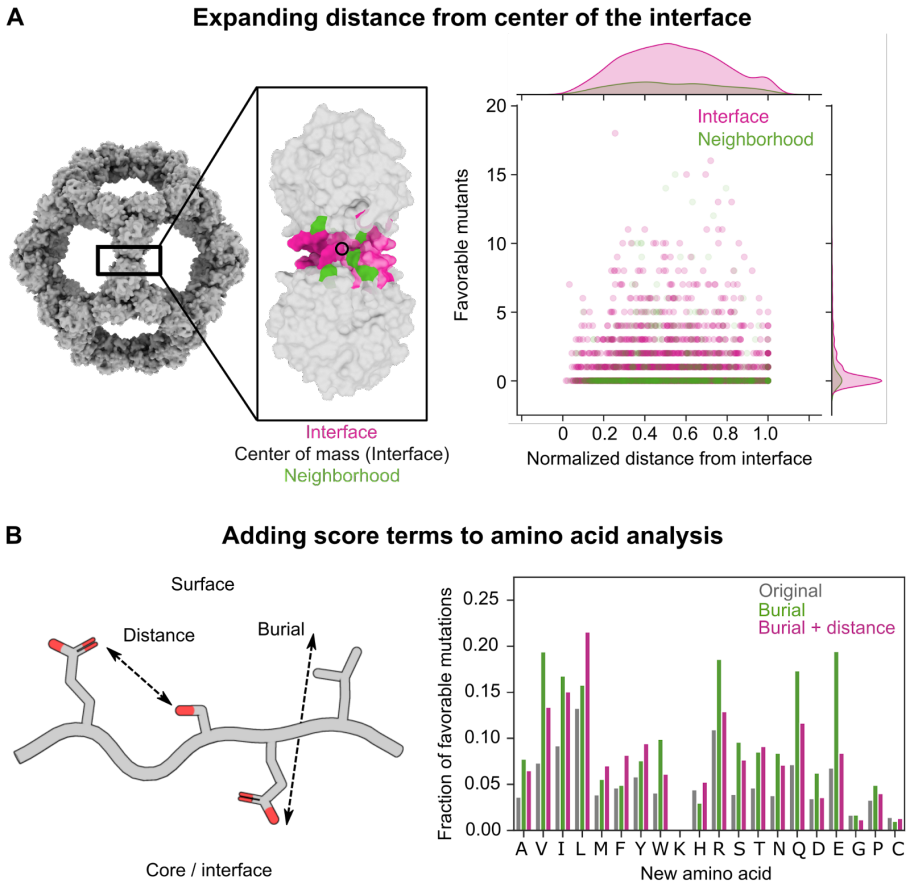


Figure 3.4. JUPITER students further refined their hypotheses and developed new tests. **A.** Students updated their hypotheses to analyze residues neighboring but not directly participating in the previously analyzed interfaces. The students found that residues neighboring the interface had the most favorable mutations halfway between the center of the mass of the interface and the furthest residue from the interface. Each circle on the graph represents a single position of a single nanoparticle. **B.** The Rosetta score function can be refined by adding distance- and burial-dependent score terms to reward favorable electrostatic interactions (distance) or penalize buried polar residues (burial) based on their depth from the protein surface. By applying two different variants of the score function, the students observed that the addition of burial- and distance-dependent score terms shifted the degree to which different residues lead to favorable mutations across all nanoparticles.

Discussion

The JUPITER program successfully re-employed undergraduates who lost research positions due to COVID-19, following a model that could be extended to any research group looking to mentor multiple students at once. The program enabled two to three core advisors to take on a cohort of undergraduate students and invite guests to host one-time lectures and office hours. In this cohort-based learning model, undergraduate researchers can take ownership of one part of a project while working with others to weave their subprojects together into a full story. Our computational study in particular fits well with the remote learning environment imposed by the COVID-19 pandemic. Students successfully learned how to design

computational experiments, maintain and analyze large amounts of data in collaboration, and develop scientific hypotheses. Data collected by students is being used to inform future design protocols and improve models that assess protein nanoparticle assembly competency(Wargacki et al. 2021). As hybrid research environments become more popular, the undergraduates from the JUPITER program are returning to research labs, citing the JUPITER program as a major reason for their interest in continuing research.

The student presentations in both phase 1 and 2 provided opportunities to practice scientific communication through both oral presentations and literature review, enabling students to explore the background behind protein nanoparticle design and to develop an understanding of the energetics and kinematics underlying a protein-protein interface. The lecturers observed that the students' ability to interpret the effects of point mutations at a nanoparticle interface was strengthened as the students developed their own hypotheses and followed up on experiments in phase 2. After the second phase, students reported in a debriefing meeting that their experiences in coding and data interpretation were crucial elements in their desire to further pursue computational research.

The undergraduate researchers provided feedback after both phases of JUPITER through surveys, in which they expressed overwhelming satisfaction. Specifically, out of the 8 responses received on our student surveys, at least 6 students enjoyed each of the weekly lectures in understanding the research topic in phase 1, and 7 students agreed that the JUPITER program gave them a better sense of whether they would like to pursue a career in research. Students expressed that the introductory lectures during phase 1 of the program prepared them well for the large-scale project they carried out in phase 2 and expressed interest in learning more about research topics in protein design. At the end of phase 2, most of the undergraduates expressed appreciation for learning protein design concepts and were excited to learn more about other research projects at our Institute.

There are also opportunities to improve the program for future iterations, which were revealed by the survey and observed by mentors throughout the course. Students experienced difficulty studying conceptual topics about protein structure when they were more focused on learning to code and analyze their data, so a future update to the program should provide a more thorough Python or Bash course prior to the start of the research project. The surveys further highlighted the need for good communication and better data management, as well as a desire for undergraduate participants to interact more with researchers outside of the direct JUPITER program. This feedback could inform future cohort-based research programs at our Institute and elsewhere; for example, future iterations of JUPITER should provide more time and resources for learning new programming languages, and also require the use of a lab notebook.

During this pilot program, we demonstrated that rigorous research can be achieved in a virtual environment. The undergraduate students learned the necessary coding skills while generating novel scientific datasets that each undergraduate subgroup took into different analysis directions. Such a course format can be applied to any protein system—not only nanoparticles—and nearly any cohort size, and we continue to expand cohort-based mentorship programs in our labs using other protein design protocols. While our project was exclusively

computational due to the COVID-19 pandemic and our Institute's focus, the format of cohort-based research approaches can be extended to non-computational projects by adapting the idea of one central research question with individual sub-hypotheses; in fact, future iterations of JUPITER are planned for hybrid computational and wet lab projects. This general cohort-based approach allows multiple students to be trained simultaneously regardless of the host lab's specific expertise. We hope that our model for cohort-based undergraduate research can inspire new research programs across any scientific field, in order to increase the number of highly beneficial research opportunities for aspiring scientists.

References

Alford, Rebecca F., Andrew Leaver-Fay, Jeliazko R. Jeliazkov, Matthew J. O'Meara, Frank P. DiMaio, Hahnbeom Park, Maxim V. Shapovalov, et al. 2017. "The Rosetta All-Atom Energy Function for Macromolecular Modeling and Design." *Journal of Chemical Theory and Computation* 13 (6): 3031–48.

"As We Face Challenges Ahead, the UW Will Put People First." n.d. Novel Coronavirus Information. Accessed December 22, 2021. https://www.washington.edu/coronavirus/2020/04/24/as-we-face-challenges-ahead-the-uw-will-put-people-first/.

Bale, Jacob B., Shane Gonen, Yuxi Liu, William Sheffler, Daniel Ellis, Chantz Thomas, Duilio Cascio, et al. 2016. "Accurate Design of Megadalton-Scale Two-Component Icosahedral Protein Complexes." *Science* 353 (6297): 389–94.

Barlow, Amy E. L., and Merna Villarejo. 2004. "Making a Difference for Minorities: Evaluation of an Educational Enrichment Program." *Journal of Research in Science Teaching* 41 (9): 861–81.

Barlow, Kyle A., Shane Ó Conchúir, Samuel Thompson, Pooja Suresh, James E. Lucas, Markus Heinonen, and Tanja Kortemme. 2018. "Flex ddG: Rosetta Ensemble-Based Estimation of Changes in Protein-Protein Binding Affinity upon Mutation." *The Journal of Physical Chemistry. B* 122 (21): 5389–99.

Bauer, Karen W., and Joan S. Bennett. 2003. "Alumni Perceptions Used to Assess Undergraduate Research Experience." *The Journal of Higher Education*. https://doi.org/10.1353/jhe.2003.0011.

Ben-Sasson, Ariel J., Joseph L. Watson, William Sheffler, Matthew Camp Johnson, Alice Bittleston, Logeshwaran Somasundaram, Justin Decarreau, et al. 2021. "Design of Biologically Active Binary Protein 2D Materials." *Nature* 589 (7842): 468–73.

Borgo, Benjamin, and James J. Havranek. 2012. "Automated Selection of Stabilizing Mutations in Designed and Natural Proteins." *Proceedings of the National Academy of Sciences of the United States of America* 109 (5): 1494–99.

Boyoglu-Barnum, Seyhan, Daniel Ellis, Rebecca A. Gillespie, Geoffrey B. Hutchinson, Young-Jun Park, Syed M. Moin, Oliver J. Acton, et al. 2021. "Quadrivalent Influenza Nanoparticle Vaccines Induce Broad Protection." *Nature* 592 (7855): 623–28.

Butterfield, Gabriel L., Marc J. Lajoie, Heather H. Gustafson, Drew L. Sellers, Una Nattermann, Daniel Ellis, Jacob B. Bale, et al. 2017. "Evolution of a Designed Protein Assembly Encapsulating Its Own RNA Genome." *Nature* 552 (7685): 415–20.

Cannon, Kevin A., Rachel U. Park, Scott E. Boyken, Una Nattermann, Sue Yi, David Baker, Neil P. King, and Todd O. Yeates. 2020. "Design and Structure of Two New Protein Cages Illustrate Successes and Ongoing Challenges in Protein Engineering." *Protein Science: A Publication of the Protein Society* 29 (4): 919–29.

Carpi, Anthony, Darcy M. Ronan, Heather M. Falconer, and Nathan H. Lents. 2017. "Cultivating Minority Scientists: Undergraduate Research Increases Self-Efficacy and Career Ambitions for Underrepresented Students in STEM." *Journal of Research in Science Teaching* 54 (2): 169–94.

Cohen, Susan E., Sara M. Hashmi, A-Andrew D. Jones, Vasiliki Lykourinou, Mary Jo Ondrechen, Srinivas Sridhar, Anne L. van de Ven, Lauren S. Waters, and Penny J. Beuning. 2021. "Adapting Undergraduate Research to Remote Work to Increase Engagement." *The Biophysicist* 2 (2): 28–32.

Divine, Robby, Ha V. Dang, George Ueda, Jorge A. Fallas, Ivan Vulovic, William Sheffler, Shally Saini, et al. 2021. "Designed Proteins Assemble Antibodies into Modular Nanocages." *Science* 372 (6537). https://doi.org/10.1126/science.abd9994.

Erickson, Olivia A., Rebecca B. Cole, Jared M. Isaacs, Silvia Alvarez-Clare, Jonathan Arnold, Allison Augustus-Wallace, Joseph C. Ayoob, et al. 2022. "'How Do We Do This at a Distance?!' A Descriptive Study of Remote Undergraduate Research Programs during COVID-19." *CBE—Life Sciences Education* 21 (1): ar1.

Fallas, Jorge A., George Ueda, William Sheffler, Vanessa Nguyen, Dan E. McNamara, Banumathi Sankaran, Jose Henrique Pereira, et al. 2017. "Computational Design of Self-Assembling Cyclic Protein Homo-Oligomers." *Nature Chemistry* 9 (4): 353–60.

Forrester, Nikki. 2021. "How the Pandemic Is Reshaping Undergraduate Research." *Nature*, May. https://doi.org/10.1038/d41586-021-01209-2.

Gilmore, Joanna, Michelle Vieyra, Briana Timmerman, David Feldon, and Michelle Maher. 2015. "The Relationship between Undergraduate Research Participation and Subsequent Research Performance of Early Career STEM Graduate Students." *The Journal of Higher Education* 86 (6): 834–63.

Goldenzweig, Adi, Moshe Goldsmith, Shannon E. Hill, Or Gertman, Paola Laurino, Yacov Ashani, Orly Dym, et al. 2016. "Automated Structure- and Sequence-Based Design of Proteins for High Bacterial Expression and Stability." *Molecular Cell* 63 (2): 337–46.

Gonen, Shane, Frank DiMaio, Tamir Gonen, and David Baker. 2015. "Design of Ordered Two-Dimensional Arrays Mediated by Noncovalent Protein-Protein Interfaces." *Science* 348 (6241): 1365–68.

Guerois, Raphael, and Manuela López de la Paz. 2006. *Protein Design: Methods and Applications*. Springer Science & Business Media.

Hathaway, Russel S., Biren A. Nagda, and Sandra R. Gregerman. 2002. "The Relationship of Undergraduate Research Participation to Graduate and Professional Education Pursuit: An Empirical Study." *Journal of College Student Development* 43 (5): 614–31.

Hsia, Yang, Jacob B. Bale, Shane Gonen, Dan Shi, William Sheffler, Kimberly K. Fong, Una Nattermann, et al. 2016. "Design of a Hyperstable 60-Subunit Protein Icosahedron." *Nature*. https://doi.org/10.1038/nature18010.

Hunter, Anne-Barrie, Sandra L. Laursen, and Elaine Seymour. 2007. "Becoming a Scientist: The Role of Undergraduate Research in Students' Cognitive, Personal, and Professional Development." *Science Education* 91 (1): 36–74.

Hunter, Anne-Barrie, Elaine Seymour, Sandra Laursen, Heather Thiry, and Ginger Melton. 2010. *Undergraduate Research in the Sciences: Engaging Students in Real Science*. John Wiley & Sons.

Jacak, Ron, Andrew Leaver-Fay, and Brian Kuhlman. 2012. "Computational Protein Design with Explicit Consideration of Surface Hydrophobic Patches." *Proteins* 80 (3): 825–38.

Junge, Benjamin, Catherine Quiñones, Jakub Kakietek, Daniel Teodorescu, and Pat Marsteller. 2010. "Promoting Undergraduate Interest, Preparedness, and Professional Pursuit in the Sciences: An Outcomes Evaluation of the SURE Program at Emory University." *CBE Life Sciences Education* 9 (2): 119–32.

Khmelinskaia, Alena, Adam Wargacki, and Neil P. King. 2021. "Structure-Based Design of Novel Polyhedral Protein Nanomaterials." *Current Opinion in Microbiology* 61 (June): 51–57.

King, Neil P., Jacob B. Bale, William Sheffler, Dan E. McNamara, Shane Gonen, Tamir Gonen, Todd O. Yeates, and David Baker. 2014. "Accurate Design of Co-Assembling Multi-Component Protein Nanomaterials." *Nature* 510 (7503): 103–8.

King, Neil P., William Sheffler, Michael R. Sawaya, Breanna S. Vollmar, John P. Sumida, Ingemar André, Tamir Gonen, Todd O. Yeates, and David Baker. 2012. "Computational Design of Self-Assembling Protein Nanomaterials with Atomic Level Accuracy." *Science* 336 (6085): 1171–74.

Kleffner, Robert, Jeff Flatten, Andrew Leaver-Fay, David Baker, Justin B. Siegel, Firas Khatib, and Seth Cooper. 2017. "Foldit Standalone: A Video Game-Derived Protein Structure Manipulation Interface Using Rosetta." *Bioinformatics* 33 (17): 2765–67.

Kluyver, Thomas, Benjamin Ragan-Kelley, Fernando Pérez, Brian E. Granger, Matthias Bussonnier, Jonathan Frederic, Kyle Kelley, et al. 2016. “Jupyter Notebooks.” a publishing format for reproducible computational workflows. In ELPUB.

Lai, Yen-Ting, Neil P. King, and Todd O. Yeates. 2012. “Principles for Designing Ordered Protein Assemblies.” *Trends in Cell Biology* 22 (12): 653–61.

Lawrence, Michael S., Kevin J. Phillips, and David R. Liu. 2007. “Supercharging Proteins Can Impart Unusual Resilience.” *Journal of the American Chemical Society* 129 (33): 10110–12.

Leaver-Fay, Andrew, Michael Tyka, Steven M. Lewis, Oliver F. Lange, James Thompson, Ron Jacak, Kristian Kaufman, et al. 2011. “ROSETTA3: An Object-Oriented Software Suite for the Simulation and Design of Macromolecules.” *Methods in Enzymology* 487: 545–74.

Lehmann, M., L. Pasamontes, S. F. Lassen, and M. Wyss. 2000. “The Consensus Concept for Thermostability Engineering of Proteins.” *Biochimica et Biophysica Acta* 1543 (2): 408–15.

Le, Kathy H., Jared Adolf-Bryfogle, Jason C. Klima, Sergey Lyskov, Jason W. Labonte, Steven Bertolani, Shourya S. Roy Burman, et al. 2021. “PyRosetta Jupyter Notebooks Teach Biomolecular Structure Prediction and Design.” *The Biophysicist* 2 (1): 108–22.

Listov, Dina, Rosalie Lipsh-Sokolik, Che Yang, Bruno E. Correia, and Sarel J. Fleishman. 2021. “Assessing and Enhancing Foldability in Designed Proteins.” *bioRxiv*. https://doi.org/10.1101/2021.11.09.467863.

Lopatto, David. 2007. “Undergraduate Research Experiences Support Science Career Decisions and Active Learning.” *CBE Life Sciences Education* 6 (4): 297–306.

———. 2017. “Adapting to Change: Studying Undergraduate Research in the Current Education Environment.” *Scholarship and Practice of Undergraduate Research; Washington Volume*. https://doi.org/10.18833/spur/1/1/7.

LSO: Local Sequence Optimality. n.d. Github. Accessed April 13, 2022. https://github.com/Fleishman-Lab/LSO.

Magliery, Thomas J. 2015. “Protein Stability: Computation, Sequence Statistics, and New Experimental Methods.” *Current Opinion in Structural Biology* 33 (August): 161–68.

Marcandalli, Jessica, Brooke Fiala, Sebastian Ols, Michela Perotti, Willem de van der Schueren, Joost Snijder, Edgar Hodge, et al. 2019. “Induction of Potent Neutralizing Antibody Responses by a Designed Protein Nanoparticle Vaccine for Respiratory Syncytial Virus.” *Cell* 176 (6): 1420–31.e17.

Nagda, Biren A., Sandra R. Gregerman, John Jonides, William von Hippel, and Jennifer S. Lerner. 1998. “Undergraduate Student-Faculty Research Partnerships Affect Studen Retention.” *The Review of Higher Education* 22 (1): 55–72.

pSUFER: Protein Strain, Unsatisfactoriness, and Frustration findER. n.d. Github. Accessed April 13, 2022. https://github.com/Fleishman-Lab/pSUFER.

Russell, Susan H., Mary P. Hancock, and James McCullough. 2007. “The Pipeline. Benefits of Undergraduate Research Experiences.” *Science* 316 (5824): 548–49.

Sahasrabuddhe, Aniruddha, Yang Hsia, Florian Busch, William Sheffler, Neil P. King, David Baker, and Vicki H. Wysocki. 2018. “Confirmation of Intersubunit Connectivity and Topology of Designed Protein Complexes by Native MS.” *Proceedings of the National Academy of Sciences of the United States of America* 115 (6): 1268–73.

Schrödinger, LLC. 2015. “The PyMOL Molecular Graphics System, Version 1.8.”

Scott-Johnson, Pamela E., Yvonne Wambaa, Brian Schmitt, and Kendra Macko. 2008. “Impact of Undergraduate Research on Mentoring, Research Efficacy, and Attitudes.” *PsycEXTRA Dataset*. https://doi.org/10.1037/e517482008-001.

Seymour, Elaine, Anne-Barrie Hunter, Sandra L. Laursen, and Tracee DeAntoni. 2004. “Establishing the Benefits of Research Experiences for Undergraduates in the Sciences: First Findings from a Three-Year Study.” *Science Education* 88 (4): 493–534.

Shen, Hui, Qihong Lou, Zhao Quan, Xuewen Li, Yifeng Yang, Xiaolong Chen, Qiurui Li, et al. 2019. “Narrow-Linewidth All-Fiber Amplifier with up to 3.01 kW Output Power Based on

Commercial 20/400 μm Active Fiber and Counterpumped Configuration." *Applied Optics* 58 (12): 3053–58.
Tantillo, Dean J., Justin B. Siegel, Carla M. Saunders, Teresa A. Palazzo, Phillip P. Painter, Terrence E. O'Brien, Nicole N. Nuñez, et al. 2019. "Computer-Aided Drug Design for Undergraduates." *Journal of Chemical Education* 96 (5): 920–25.
Ueda, George, Aleksandar Antanasijevic, Jorge A. Fallas, William Sheffler, Jeffrey Copps, Daniel Ellis, Geoffrey B. Hutchinson, et al. 2020. "Tailored Design of Protein Nanoparticle Scaffolds for Multivalent Presentation of Viral Glycoprotein Antigens." *eLife* 9 (August). https://doi.org/10.7554/eLife.57659.
"Undergraduate Research Symposium." n.d. Undergraduate Research Program. Accessed May 11, 2022. https://www.washington.edu/undergradresearch/symposium/.
Vater, Ashley, Jaime Mayoral, Janelle Nunez-Castilla, Jason W. Labonte, Laura A. Briggs, Jeffrey J. Gray, Irina Makarevitch, Sharif M. Rumjahn, and Justin B. Siegel. 2021. "Development of a Broadly Accessible, Computationally Guided Biochemistry Course-Based Undergraduate Research Experience." *Journal of Chemical Education* 98 (2): 400–409.
Walls, Alexandra C., Brooke Fiala, Alexandra Schäfer, Samuel Wrenn, Minh N. Pham, Michael Murphy, Longping V. Tse, et al. 2020. "Elicitation of Potent Neutralizing Antibody Responses by Designed Protein Nanoparticle Vaccines for SARS-CoV-2." *Cell* 183 (5): 1367–82.e17.
Wargacki, Adam J., Tobias P. Wörner, Michiel van de Waterbeemd, Daniel Ellis, Albert J. R. Heck, and Neil P. King. 2021. "Complete and Cooperative in Vitro Assembly of Computationally Designed Self-Assembling Protein Nanomaterials." *Nature Communications* 12 (1): 883.
Wei, Cynthia A., and Terry Woodin. 2011. "Undergraduate Research Experiences in Biology: Alternatives to the Apprenticeship Model." *CBE Life Sciences Education* 10 (2): 123–31.
Whitehead, Timothy A., Aaron Chevalier, Yifan Song, Cyrille Dreyfus, Sarel J. Fleishman, Cecilia De Mattos, Chris A. Myers, et al. 2012. "Optimization of Affinity, Specificity and Function of Designed Influenza Inhibitors Using Deep Sequencing." *Nature Biotechnology* 30 (6): 543–48.

Chapter 4: Using Rosetta scoring metrics to separate between properly assembled and non-assembled de novo designed nanocages

828 Nanoparticles have been designed by computation,
Docked, characterized, and validated for a desired formation.
T, O, and I architectures have been explored,
With TCdock and RPXdock as methods to score.

Experimental validation was carried out,
65 nanocages assembled, without a doubt.
But cooperativity and completeness remained,
As assembly failure modes couldn't be restrained.

We estimated Rosetta scoring metrics as ways to filter,
Proper assembling future designs made by new protein builders.
But these metrics may overlap, we must stress,
Just because it's in range, it's not a success.

There's an ideal ddG, driven by SASA, we'll say,
And it's symmetry dependent, as of today.
Good shape complementarity is predicted to perform well,
But not discriminatory, with outliers that fell.

When evaluating designs by eye,
Unsatisfied Hydrogen Bonds should not lie.
Big interfaces may look better but can result in poorly behaved parts,
Strong interdigitation, shape complementarity, and cooperative interfaces are still works of art.

Introduction

As of July 2021, at least 828 de novo designed nanocages in T, O, and I architectures have been computationally docked and designed, and characterized in the wet lab. Two docking methods for T, O, and I architectures have been established: tcdock and rpxdock. Published in 2014 (King et al. 2014), tcdock was an application that sampled the rotational and translational degrees of freedom between pairs of protein scaffolds in higher order symmetries. Dock configurations were scored based on interface design compatibility and designed with the symmetric modeling framework in Rosetta using user-provided symmetry definition files. Rpxdock expanded upon tcdock, utilizing a new transform-based objective function, which retains some of the power of full atom force-fields while avoiding a costly and difficult-to-optimize full atom model, and optimization techniques such as hierarchical sampling and scoring and hierarchical packing. Of the 828 nanocages experimentally validated, 65 were reported as assembled. The data collected herein makes no claims about cooperativity or completeness of assembly, small discrepancies between the design model and experimentally validated nanocage, nor alternative failure modes such as scaffold misfolding or insolubility that would prevent an assembly from forming.

This document summarizes the computational metrics estimated from the characterized designs, with the goal of guiding future cage design towards a higher success rate.

Table 4.1: Distribution of assembled and not assembled nanoparticles by architecture.

Architecture	Not assembled	Assembled
I	332 (93.3%)	24 (6.7%)
I3	121 (91.7%)	11 (8.3%)
I32	78 (96.3%)	3 (3.7%)
I52	44 (93.6%)	3 (6.4%)
I53	89 (93.7%)	6 (6.3%)
O	200 (93.5%)	14 (6.5%)
O3	57 (87.7%)	8 (12.3%)
O4	63 (98.4%)	1 (1.6%)
O32	19 (100%)	0 (0%)
O43	59 (92.2%)	5 (7.8%)
T	231 (89.5%)	27 (10.5%)
T3	25 (75.8%)	8 (24.2%)
T32	54 (92.5%)	5 (8.5%)

T33	152 (91.6%)	14 (8.4%)

Overall, our analyses suggest that there is no conclusive way to distinguish between properly assembling and non-assembling nanocages, based solely on computational metrics. We strongly encourage that designs still be visually inspected by the designer. However, we do provide Rosetta scoring metric ranges, grouped by symmetric architecture, that encompass properly assembling nanocages. Rosetta scoring metrics of properly assembling and non-assembling nanocages often overlap -- just because designs are within a certain range for these metrics doesn't always mean they will work.

Data collection

Previously, rigid body docks were designed at the docking interface, while preserving the amino acid identity of the rest of both native and de novo protein building blocks. Designs were likely prefiltered by several interface metrics, depending on the designer:

- Shape complementarity > 0.5
- 1 component SASA within 300-1000; 2 component SASA within 600-1500
- ddG < -0.01

The models and scores for all designs were gathered by crowd-sourcing and scored using one Rosetta script XML. All designs were scored based on all-atom Rosetta Energy score terms and centroid-pair and cbeta score terms, with the latter hypothesized to capture interface "closeness".

Sequence identity compatibility for each designed protein-protein interface was also collected computationally. Designed interfaces of all cages were repacked and minimized with full chi angle and backbone movement, and an SSM was estimated for the residues at and within 10A of the relaxed interface. The SSM data was analyzed for each amino acid identity and residue position relative to the rest of the interface for its mutability and favorability.

Size exclusion, light and x-ray scattering, or electron microscopy techniques were used to experimentally validate correct assembly of each design. Failure of assembly was defined at any stage in cage production and characterization (i.e. protein did not express, insolubility, no assembly, off-target assembly, etc.).

A compilation of RosettaScoring Metrics and experimental characterization status of all designs lives here: /home/erinyang/all_ever_ordered/SCORE_CAGES/processing/plot/all_scores_df.csv

Residue level scoring metrics at each designed interface lives here: /projects/jupiter/erinyang/manuscript/output/*

Results

Table 4.2: Working cage ddG are generally in a narrower range than non-working cages

	Average (StdDev) of ddG	
Symmetry	Non-working Cages	Working Cages
I3	-29.99 (20.07)	-33.7 (8.02)
I32	-55.4 (20.42)	-74.94 (2.25)
I52	-35.23 (110.47)	-63.49 (7.39)
I53	-55.73 (21.22)	-51.32 (6.54)
O3	-17.66 (20.9)	-26.79 (5.33)
O32	-54.21 (20.4)	0 (0)
O4	-23.58 (15.32)	95.61 (0)
O43	-32.39 (31.06)	-57.57 (13.12)
T3	-21.17 (8.25)	-31 (5.13)
T32	-59.18 (16.15)	-53.81 (12.94)

Working cages tend to have ddG in a narrower range than non-working cages across all T, O, and I architectures. The fewer total number of working cages across all architectures could also contribute to the difference in standard deviation between working and not working cages.

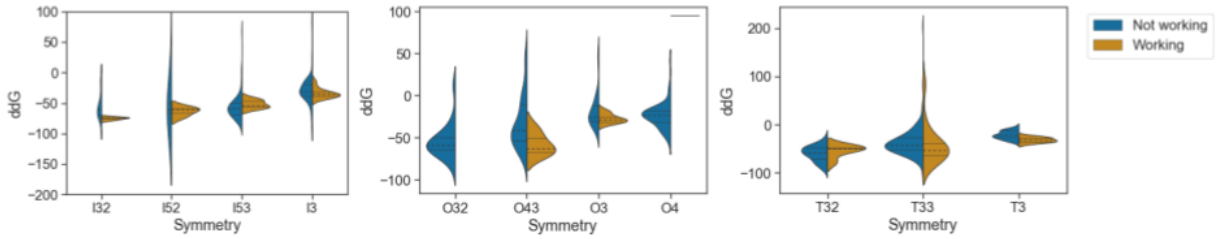


Figure 4.1: Optimal Rosetta Metrics differs, depending on symmetry

ddG:
Generally ddG should be between 0 and -100 REU. If you notice any outliers in ddG, it typically means the initial dock or inputs weren't set up in the proper symmetry orientation, or there is an issue with residue selectors and task operations in the ddG filter. Below is a chart and table summarizing the ddG among working and not working cages by symmetry.

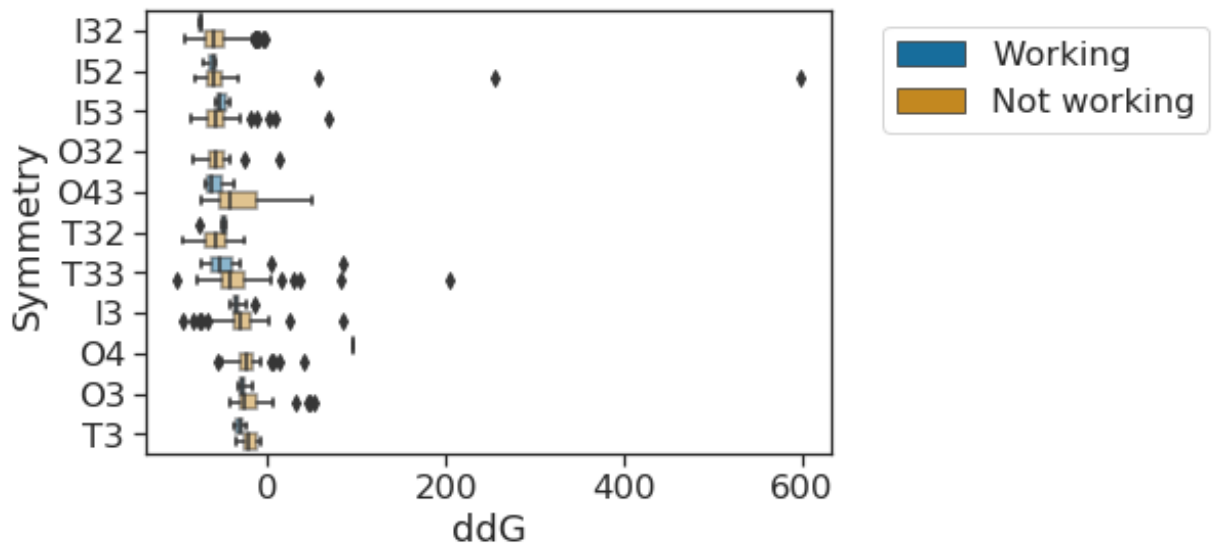


Figure 4.2: Distribution of ddG among working and not working nanoparticles by symmetry

Table 4.3: Mean, median, and standard deviation of ddG among working and not working nanoparticles by symmetry

	Mean		Median		Standard Deviation	
Symmetry	**Not working**	**Working**	**Not working**	**Working**	**Not working**	**Working**
I3	-29.99	-33.70	-30.63	-35.75	20.07	8.02
I32	-55.40	-74.94	-60.93	-73.99	20.42	2.25
I52	-35.23	-63.49	-60.11	-60.46	110.47	7.39
I53	-55.73	-51.32	-58.94	-52.93	21.22	6.54
O3	-17.66	-26.79	-24.77	-27.75	20.90	5.33
O32	-54.21	NaN	-58.49	NaN	20.40	NaN
O4	-23.58	95.61	-22.60	95.61	15.32	NaN
O43	-32.39	-57.57	-41.92	-62.79	31.06	13.12
T3	-21.17	-31.00	-21.31	-31.45	8.25	5.13
T32	-59.18	-53.81	-57.75	-48.64	16.15	12.94
T33	-36.37	-40.32	-42.41	-52.69	30.57	40.82

SASA and ddG are highly correlated

We hypothesize there is an ideal ddG, driven primarily by SASA, for each symmetry (Wargacki et al. 2021). While there isn't a magic sasa value that differentiates working from non-working cages, sasa values typically differ between target symmetries. Specifically, SASA and therefore ddG can be slightly lower for When using the sasa_priority scorefunction in rpxDock, aim to set the sasa to the mean value of your target architecture. Note that one comp sasa is half of the target sasa set using the sasa_priority scorefunction.

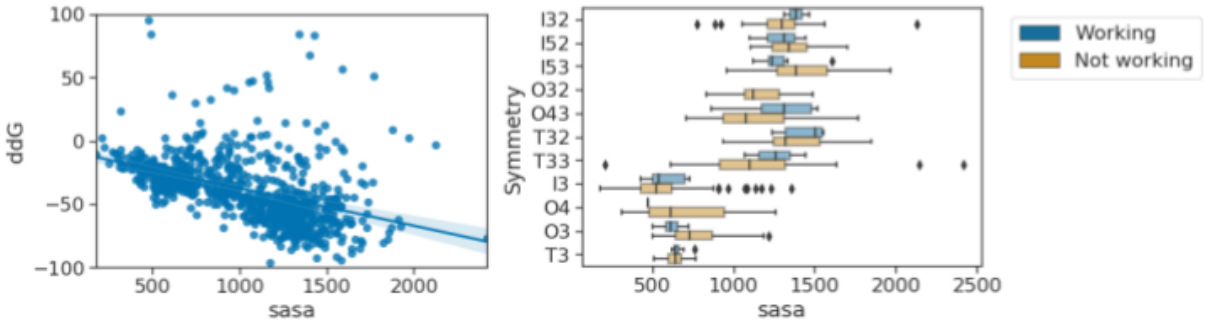


Figure 4.3: Correlation of ddG and SASA (left) and distribution of SASA (right) among working and not working nanoparticles by symmetry

Below is a list of the mean, median, and standard deviation sasa for each symmetry. It is important to note that some of these values stem from very few working cages within a particular symmetry group, and thus should be evaluated carefully.

Table 4.4: Mean, median, and standard deviation of SASA among working and not working nanoparticles by symmetry

	Mean		Median		Standard Deviation	
Symmetry	**Not working**	**Working**	**Not working**	**Working**	**Not working**	**Working**
I3	556.43	575.71	528.70	536.64	203.68	110.74
I32	1297.35	1387.91	1295.61	1384.49	172.94	74.74
I52	1351.56	1287.87	1338.82	1315.03	141.61	173.40
I53	1433.41	1292.27	1384.01	1241.26	220.16	169.49
O3	770.85	618.91	727.56	614.65	178.10	70.49
O32	1164.33	NaN	1120.48	NaN	157.44	NaN
O4	703.44	475.99	612.03	475.99	270.95	NaN
O43	1131.14	1270.21	1073.49	1313.43	239.26	266.06
T3	647.66	659.34	645.77	640.88	73.70	48.66
T32	1356.45	1432.10	1318.66	1503.16	183.86	146.27

T33	1120.61	1255.01	1102.45	1262.37	288.42	121.08

*There are only a handful of working cages in each symmetry--some symmetries only have 1 working cage--these values may not be as standalone as they may appear
**At the time of this analysis there were no working O432 and only one working O4 cage

Because all ordered cages passed visual inspection, we might assume that these cages looked experimentally plausible to someone, and therefore might contain some useful information for the designer.

Shape Complementarity

Generally, interfaces with good shape complementarity are predicted to experimentally perform well but are not discriminatory. Sc1 > 0.5 are typically observed for both working and not working cages, with few non working cage outliers ordered with sc1 < 0.5.

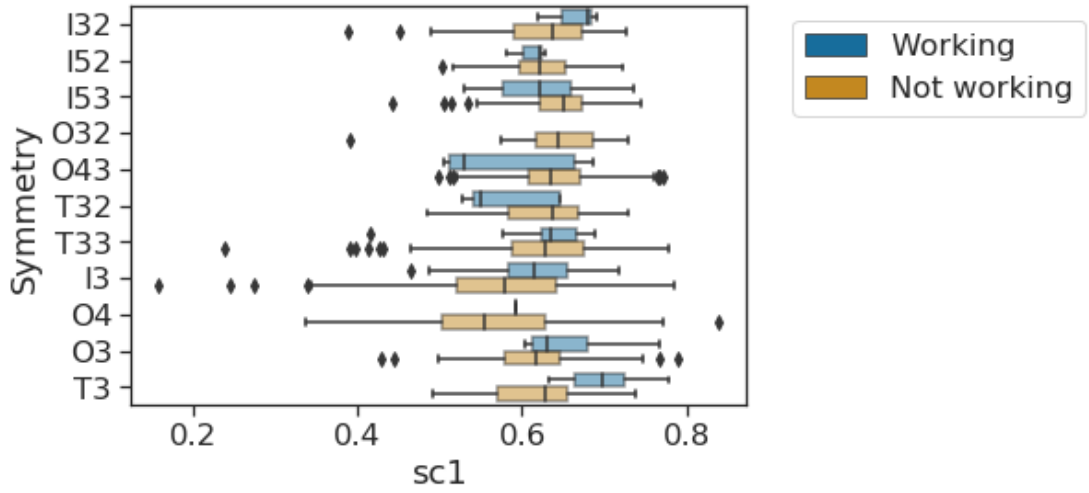


Figure 4.4: Distribution of shape complementarity among working and not working nanoparticles by symmetry

Table 4.4: Mean, median, and standard deviation of shape complementarity among working and not working nanoparticles by symmetry

	Mean		Median		Standard Deviation	
Symmetry	**Not working**	**Working**	**Not working**	**Working**	**Not working**	**Working**
I3	0.57	0.61	0.58	0.61	0.10	0.08
I32	0.62	0.66	0.64	0.68	0.06	0.04
I52	0.62	0.61	0.62	0.62	0.05	0.02
I53	0.64	0.62	0.65	0.62	0.05	0.07

O3	0.61	0.66	0.62	0.63	0.07	0.06
O32	0.64	NaN	0.64	NaN	0.08	NaN
O4	0.56	0.59	0.56	0.59	0.10	NaN
O43	0.64	0.58	0.64	0.53	0.06	0.09
T3	0.61	0.70	0.63	0.70	0.07	0.05
T32	0.62	0.58	0.64	0.55	0.06	0.06
T33	0.62	0.63	0.63	0.63	0.08	0.07

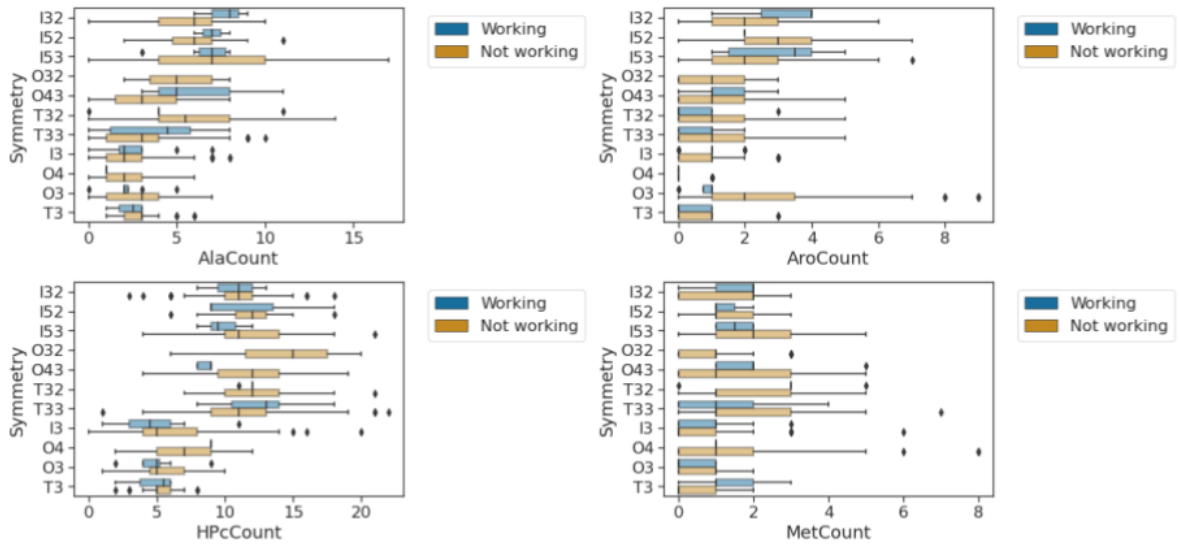


Figure 4.5: High residue counts (HPcCount, AroCount, AlaCount, MetCount) are generally found more often in nonworking cages

Residues closer to the center of the interface generally are less mutable in working cages than in non-working cages

As part of the JUPITER project, undergraduate students took every nanocage and alanine scanned the interface. They noticed that in some architectures, the ***average*** number of favorable mutations (count of alanine scan ddG < 0 per interface residue) was higher in non working cages than working cages closer to the center of the interface. The data suggest that non working cages could use better packing at the interface. However, since not all architectures showed this pattern, the data are inconclusive.

<u>Tips when looking at cage designs by eye:</u>

Avoid Unsatisfied Hydrogen Bonds (UHBs)

UHBs can happen from backbone atoms as well if the terminus of helices get stuffed into interfaces. Brian Coventry has buried_unsat_penalty edits to the Rosetta scorefunction and analysis of how many hydrogen bonds each specific side chain needs: HBond Preferences Of Buried Polars

Big interfaces look better, but result in poorly behaved components
It is easy to fall into a trap of choosing designs with bigger interfaces because they may be more visually appealing. Based on the work of Adam Wargacki, interface size and ddG is highly dependent on architecture--the more interconnected the architecture, the weaker the interface needs to be [Insert figure here]. Large patches of hydrophobic residues often result in misfolding or insolubility of individual components. Ideally, interfaces look weak as monomers but are strong due an avidity effect of an assembly.

Look for strong interdigitation of residues
Shape complementary (SC) scores generally give an indication of how well interfaces are packed. However, a high SC score does not necessarily mean it's good. Two very flat interfaces are also "complementary".

Avoid high AlaCount, AroCount, MetCount, gaps/voids at the interface
Interfaces with purely alanines can be non-specific and/or actually be too close to each other (Rosetta has no other rotamer choice). If you see many large residues (aromatics, methionines), it can be a sign that your interface is too far apart (Rosetta is trying to fill the hole). When you zoom out in Pymol, the interface should not be an obvious gap, as water should not be residing in the hydrophobic core.

Avoid many surface hydrophobic residues
If it can be a polar residue without burying a polar atom (within rotamer reason), it's probably better to be polar. Aromatic residues should be supported on both sides and not just laying on a pad of hydrophobics. Ideally you want a hydrophobic core that is surrounded by polar residues.

Avoid long stretches of hydrophobic residues in primary sequence
This is especially pertinent for protein secretion but can generally result in misfolding or subunit insolubility. If you see designs with this pattern, you may want to consider @jyjwang's Degreaser code.

Minimize clashes
While some minor side chain clashes can be tolerated (the rotamers can flip), backbone clashes should be avoided at all costs.

Data Availability

All scripts used to prepare input pdbs and score cages are uploaded to github at the following link: https://github.com/erincyang/rosetta_scoring_nanoparticles. All nanoparticle scaffolds are located at the following link internally at IPD: /home/erinyang/all_ever_ordered/processed/.

References

King, Neil P., Jacob B. Bale, William Sheffler, Dan E. McNamara, Shane Gonen, Tamir Gonen, Todd O. Yeates, and David Baker. 2014. “Accurate Design of Co-Assembling Multi-Component Protein Nanomaterials.” *Nature* 510 (7503): 103–8.

Wargacki, Adam J., Tobias P. Wörner, Michiel van de Waterbeemd, Daniel Ellis, Albert J. R. Heck, and Neil P. King. 2021. “Complete and Cooperative in Vitro Assembly of Computationally Designed Self-Assembling Protein Nanomaterials.” *Nature Communications* 12 (1): 883.