Machine Learning and Modeling Methods for Protein Engineering

Thesis by

Aiden Joseph Aceves

In Partial Fulfillment of the

Requirements for the degree of

Doctor of Philosophy

CALIFORNIA INSTITUTE OF TECHNOLOGY

Pasadena, California

2022

(Defended August 18, 2021)

ACKNOWLEDGEMENTS

This thesis is dedicated to my family. Without my father's commitment to my education and the stories he told me about Caltech when I was growing up, I would never be here today. My mother has shown me how it is possible to calmly and tirelessly persevere, without ever sacrificing empathy or understanding. I thank my brother and sister for their companionship and counsel as we all navigate our schooling, careers, and personal journeys. Also someday soon (maybe it has already happened), Christine, you will be a better programmer than me, and I will ask you for advice. And to my partner Marlene, I thank you for nearly a decade of support. You have helped me to possess the confidence and connectedness to face many challenges in life.

I would like to thank my advisor Dr. Steve Mayo for being an impeccable role model of professionalism melded with kindness and personability. I am grateful for the advice and opportunities you have given me, and could not have asked for a better advisor.

Abstract

Computation has been an integral part of structural biology, ever since the first protein macromolecular structure was solved via Fourier Synthesis on the EDSAC Mark I electronic computer in 1958 (Kendrew et al., 1958). Throughout my time at Caltech, I have endeavored to develop new methods to apply machine learning and molecular modeling to the study of biological macromolecules. These efforts have taken two distinct tracks, but are unified by a focus on studying proteins on a structural level.

Through the application of molecular dynamics and modeling, I have studied insulin from several angles, including the incorporation of non-canonical amino acids, and how these modifications might be responsible for the modification of critical properties such as hexamer dissociation and fibrillation formation. Additionally, I have probed how insulin behaves at the interface of water and silica, a property which is critical for the effective dissemination and administration of this therapeutic molecule. I have helped to develop a novel computationally guided workflow for integrating drug conjugates into antibody CDRs. This technique yields molecules which exhibit synergistic binding and an enhanced ability for selective binding.

The second major thrust of my research has focused on applying machine learning to protein engineering problems, particularly developing tools for working with structural data, and for making efficient re-use of data which has already been laboriously collected by other groups. The basic data parsing and processing tools which were created and refined over the course of my time at

Caltech has enabled many other projects, both of my own and of collaborators. Studies into the use of generative networks for protein-protein docking have been conducted which lend useful insights for network architecture, the inclusion of intermediate learning objectives, and overcoming sparsity. The technique introduced in our ICLR 2021 paper demonstrates a regularization method which enables data from past protein engineering campaigns to be leveraged to learn policies which optimally select molecules to synthesize in unrelated engineering efforts, to potentially save a significant amount of time and money for future projects.

Reference

Kendrew, J. C.; Bodo, G.; Dintzis, H. M.; Parrish, R. G.; Wyckoff, H.; Phillips, D. C. A “Three-Dimensional Model of the Myoglobin Molecule Obtained by X-Ray Analysis”. Nature 1958, *181* (4610), 662–666.

Published Content and Contributions

Ayya Alieva, Aiden Aceves, Jialin Song, Stephen Mayo, Yisong Yue, and Yuxin Chen2021. Learning to Make Decisions via Submodular Regularization. In International Conference on Learning Representations.

url: https://openreview.net/pdf?id=ac288vnG_7U

Aiden Aceves implemented all code used in the protein engineering portions of the paper, and planned and conducted experiments including data collection through analysis. Aiden Aceves additionally contributed to the writing of the paper, responding to reviewers, and preparing revisions. Aiden Aceves presented this work as a poster at ICLR 2021, and as an invited talk at AI LA's healthcare summit.

Jingzhou Wang, Aiden Aceves, and Stephen Mayo. In Preparation. CDRxAb: Antibody Small-Molecule Conjugates with Computationally Designed Target-Binding Synergy.

Aiden Aceves created the computational workflow for determining the optimal location for incorporating small-molecule conjugates and ran the workflow for the biotin proof-of-concept system published. Aiden Aceves parameterized and performed molecular dynamics simulations of the designed complex to help diagnose necessary stabilizing framework mutations. Aiden Aceves wrote the Methods section covering these aspects.

List of Tables and Figures

Table of Contents

Section 1: Machine Learning Projects

1.1 Creation and refinement of enabling tools: VoxLearn

When I began my PhD in 2017, nearly all machine learning in protein engineering had focused on relating amino-acid sequences directly to function. Such approaches had been favored out of simplicity, leaning heavily upon the axiom that all of the structural information about a protein may be gleaned from its sequence. Because protein function derives directly from structure, which in turn derives in part from sequence, models based only upon sequence data must also decode structure. Moreover, these approaches discredit the tremendous efforts by the community to solve, annotate, and collect protein structures in databases such as the RCSB (Berman et al., 2000). Citing a lack of well-developed code bases for doing machine learning on protein structure, or even on basic manipulations and preprocessing on 3D protein voxels, among my first projects at Caltech was to author tools to fill this gap. While such tools existed for 2D image processing, and are growing for other 3D applications, there were none in the protein design domain. The problem of how to effectively encode protein structures is itself an open question, with only three prior works (Lau et al., 2017, Torng et al., 2017, Wallach et al., 2015) having employed voxel grids to study proteins in any context. The tools I created, dubbed VoxLearn, would be used in a variety of projects at Caltech as well as at the companies we collaborated with.

VoxLearn was designed to enable the following workflow:

1. Generate data mapping protein structure to some label(s)
2. Transform that data into a voxelized format
3. Build a neural network to predict the label(s) from the voxelized data

VoxLearn includes utilities for taking a .pdb file and converting it into an atom dictionary. Users can augment this data with their own additional channels such as descriptors derived from molecular mechanics force fields. There are also a variety of preprocessing tools included in the package, including

- One hot encoding atom and amino acid feature vectors
- Rotating features to augment 3D protein data
- Cropping, blurring, and jittering data as inspired by image processing techniques

For machine learning, one needs to encode information in a format conducive to matrix operations. In order to capture 3D spatial information, a 4D tensor format was adopted, where the first three dimensions are the (x, y, z) coordinates of the protein, and the last dimension is the features associated with that voxel, such as the atomic identity or force field derived terms. Processed tensors may be combined as one very large file to be read into a machine learning framework, or loaded incrementally with a generator function. Both approaches are demonstrated in the code base. In most use cases, preparing

one large file results in substantially faster network training, but requires a tremendous amount of memory to be available.

Additional features have been added to this library over time, including the ability to parse and voxel encode small molecule file formats, updated generator functions to work with the latest version of Tensorflow, and a variety of neural network templates which have been adapted for protein engineering. I have additionally used this package to train three undergraduate students on the basics of neural networks and contributing to Git repositories. A separate version of the package has also been authored which is proprietary to Novartis, and is used in the modeling and cheminformatics groups.

1.2 Hello world: Experiments with a model system

Fluorescent proteins are an indispensable part of a molecular biologist's toolbox, and the Mayo lab has developed several of the most widely used examples, including mKelly1/2 (Wannier et al., 2018), mRouge, and mRojo (Chica et al., 2010). In this project, we utilized voxel-based representations of fluorescent protein structures to train deep learning models relating structure to emission spectra.

Starting from FPbase (Lambert, 2019), a dataset of fluorescent proteins with characterized emission spectra were cross-referenced with the RCSB to yield 186 structures. The data was curated to remove outliers (such as data collected at extreme pH's), voxelized, and used to train deep neural networks.

Each protein was centered around the fluorophores and atoms were encoded as one-hot vectors using VoxLearn. Using automated hyperparameter and architecture searching tools, we tested 3D networks inspired by Dual Path Networks (Chen et al., 2017), ResNet (He et al., 2015), and googLeNet (Szegedy et al., 2015), and after significant work to overcome bugs, succeeded in training a network producing a useful degree of accuracy.

This work was designed by me and used in mentoring Michelle Garcia, an undergraduate at Pomona College, although I ultimately learned a lot from her and the experience as well. Michelle began the work with minimal coding experience, but quickly became fluent in Python, and consumed machine learning research voraciously. She subsequently has presented this work at two conferences, and is currently doing an internship in machine learning before applying to grad school at Caltech in fall of 2021.

The following is a manuscript prepared by myself and Michelle Garcia. I designed the experiments, prepared data for analysis, provided feedback on analysis, and edited the manuscript. M.G. wrote the manuscript and implemented and tuned the networks presented therein. Note that figure numbers and citations are independent of the rest of the thesis.

A Three-Dimensional Convolutional Neural Network to Predict Fluorescent Protein Maximum Emission Peaks

Introduction

Presently, a wide range of fluorescent proteins are commercially available with maximum emission peaks ($\lambda_{emission}$) ranging from 420 nm, a deep violet, to 700 nm, a vibrant red.[1] Despite the variety, there are few fluorescent protein monomers with $\lambda_{emission}$ in the nearinfrared window of 650 nm to 700 nm–a variance likely due to the occupation of lowlying excited singlet and triplet states that may increase fluorophore reactivity.[2] The scarcity of far-red emitting fps is a major limitation for live-tissue imaging, where the nearinfrared window is favorable for light penetration and necessary for deep imaging.[3,4] Moreover, the comparably smaller energy difference between the transition of HOMO and LUMO states in far-red fp fluorophores ensures a reduction of autofluorescence, lightscattering, and phototoxicity.[2,5] Scherbo et al., 2007 reported the far-red fluorescent protein mKate, the current protein of choice for whole-body imaging, and their contributions increased deep imaging resolution with an invitation to close this ~50 nm gap to raise the sensitivity of whole-body imaging techniques.[3,4]

We endeavor to create far-red fluorescent protein molecular models for eventual synthesis in the laboratory by applying machine learning principles to guide our

efforts. Machine learning is the practice of using algorithms to learn patterns from raw data for representation in a model; the model can then infer patterns from newly generated data.[6] Neural networks, a type of ML model, are loosely related to the function of the human brain. The basic unit known as a node is the neuron, and the connections created between neurons happen, likewise, after a training period on raw data. In the past decade, neural network model performances have greatly improved due to the availability of larger datasets, integration of Graphical Processing Units (GPU), and creative model architectures. The remarkable network that pushed forward deep learning was the 2012 ImageNet LARGE Scale Visual Recognition Competition winner known as AlexNet.[7] The deep convolutional neural network (CNN) trained and evaluated with over 10 million labeled images corresponding to over 20,000 categories demonstrated that GPUs assist and greatly improve network learning rates, and CNNs can facilitate learning with more ease as there are sparser connections between nodes, thus less parameters to train. Various state-of-the-art networks since then have implemented unique architecture designs cross-disciplinarily with convolutional layers that perform a set of linear operations with an input, and pooling layers that create intermediate representations suitable for generalized learning. Thus, here, we utilize a three-dimensional convolutional neural network for fluorescent protein $\lambda_{\text{emission}}$ prediction given a molecular model.

The Dataset and Data Representation

The dataset consists of 130 fp subunits and monomer crystal structures obtained from FPbase, a fluorescent protein database moderated by the scientific community.[1] All crystal structures were captured in a pH range of 6-9. An in-house Python package named MoleculeCompletion was utilized to configure information in a protein data bank file to an accessible format for the neural network input layer. MoleculeCompletion encodes each atom in the monomer or subunit of an xmer on a 3D grid-like space into four-dimensional tensors, a process called voxelization.[8]

Two encoding methods were utilized. First, the protein atoms were one-hot encoded with each atom type (C, H, O, N, P, S, or other) corresponding to a batch, and the width, height, and features representative of the voxel grid (**Figure 1**). The second approach encoded the canonical amino acids and ascribed a random georgieV value, which would give a more descriptive numerical representation with the location of the amino acid, essentially it describes the physicochemical parameters that describe amino acid qualities such as hydrophobicity, volume, mutability, and more.[9] While the first approach most adequately captured the location of each atom in the protein, especially the geometry of the chromophore, the second approach created a more detailed matrix representation of the protein. The first provides more data points, but the latter possibly fewer more descriptive points. Consider though that the georgieV encoding of the protein could not adequately capture the chromophore as it had

an assigned value of zero, but rather it captured the amino acids around the chromophore.

In both approaches, we encoded a 32 x 32 x 32 voxel grid with 1 Å voxels centered around the chromophore. The complete structures of most protein subunits or monomers are not captured, yet the atoms around and including the chromophore are conserved. Given our limited data set, we created more structural representations of the same protein by rotating and translating the molecular model with quaternion rotation, a fundamental technique in 3D computer graphics where the sum of the coefficients of three variables representing vectors in space is one, and it is randomly off-centering the voxel grid 10% from the chromophore.[10]

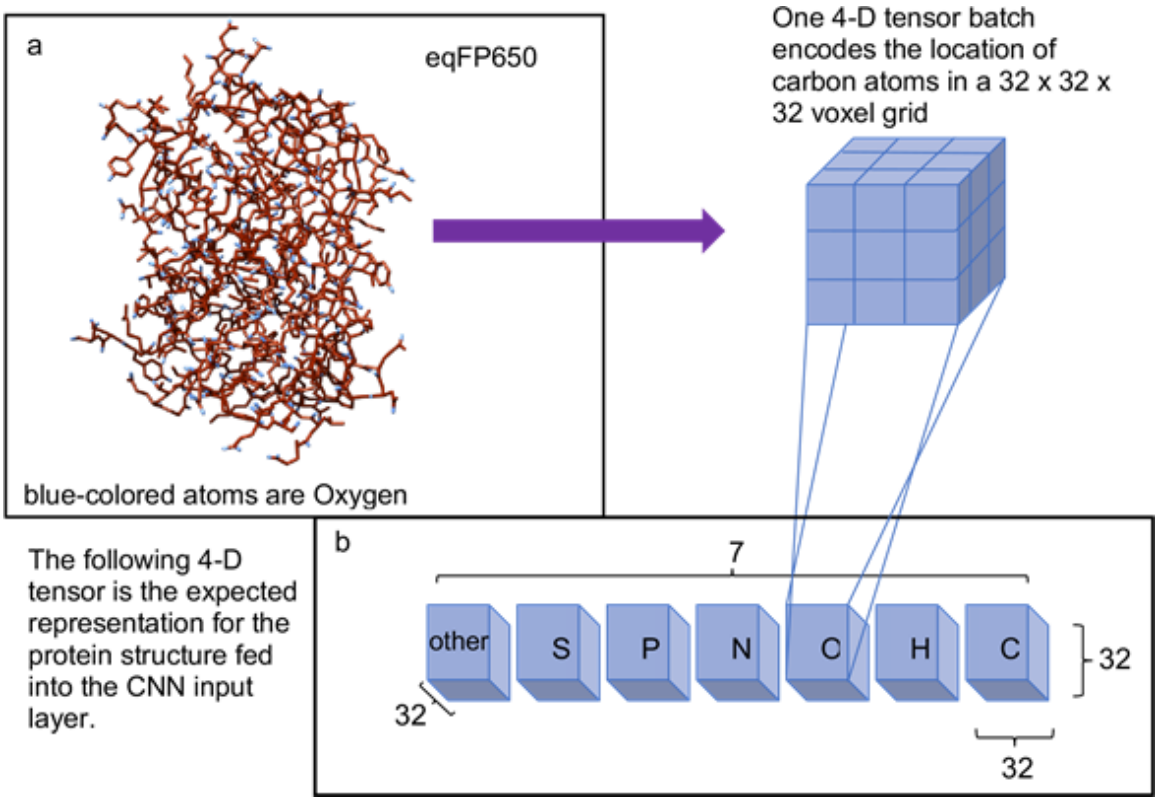


Figure 1. Crystal structure to 4D tensor data generation workflow (a) stick representation of eqFP650 (PDBID:4EDO, generated with chimera), one of the 130 proteins in our dataset. (b) 4D tensor with a batch size 7, width size 32, height size 32, feature size 32 for the one-hot encoding. The tensor batch size elongated to 21, when encoding with georgieV values.

Network Architecture

We drew inspiration from computer vision, specifically networks that have performed well in the IMAGENET Large Scale Visual Recognition Challenge. We adapted the two-dimensional network googLeNet[11] (ILSVRC 2014 winner) with a shelled out Alexnet to form our Network (**Table 1**).[7] We introduced the unique architectural motif implemented in GoogLeNet version 3 known as inception modules. These modules increase network depth through the organizational

method (**Figure 2**). Specifically, the 1 x 1 x 1 convolutions introduce sparsity into the network that addresses the overfitting of training data issue that most deep networks suffer.[12] The construction of a new network (FPCNN) with three-dimensional convolutions followed included the inception module adapted in version 3 of the GoogLeNet. FPCNN was hyperparameter tuned utilizing hyperband with a total of 480 different hyperparameters explored. The 14-layer deep model architecture that preformed best on our dataset has 3,261,785 total parameters for the one-hot encoding, and 3, 3,406,937 for the georgieV amino acid encoding (**Table 1**).

Table 1. FPCNN architecture for one-hot and georgieV encoding

Layer	output	Parameters
Conv3D	32 x 32 x 32 x 4	72,816
MaxPool3D	16 x 16 x 16 x 4	0
Inception 1	8 x 8 x 8 x 2	98,672
MaxPool3D	8 x 8 x 8 x 272	0
Inception 2	8 x 8 x 8 x 536	188,040
Inception 3	8 x 8 x 8 x 1072	673,520
Average Pooling 3D	2 x 2 x 2 x 1072	
Dense 1	256	2,195,712
Dense 2	128	32896
fully connected layer, with a linear activation, and one node		

Table 1. FPCNN architecture for one-hot and georgieV encoding

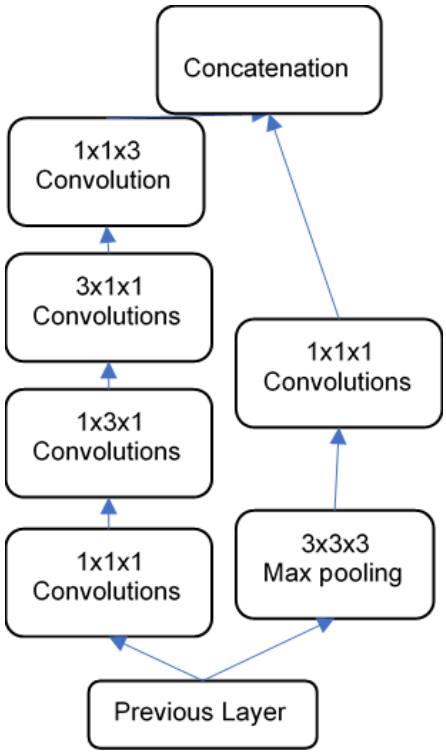


Figure 2. *FPNET Inception Module*. The 1 x 1 x 1 kernel size convolutions, followed by the 1x1x3, 1x3x1, 3x1x1, and 5 x 5 x 5 convolutions allow the network to explore more features, while reducing spatial dimension. This approach is adapted from the *ILSVRC* winner GoogLeNet.

Results and Discussion

The network was trained with the TensorFlow Keras API utilizing the Mean Squared Error loss function. The total set of 130 fp subunits and monomer crystal structures (all set) as well as a set of 35 fp's with $\lambda_{emission}$ greater than 600 nm (RFP only set) were utilized for 10-fold cross validation training—a common technique for small datasets. The model was trained by augmenting each model by a factor of four; at testing, each model was augmented 8

times, as described above in the data set and data representation section. The all set outputted 1,040 predictions, while the RFP only outputted 344 predictions. The Pearson ρ correlation was computed to measure the linear relationship between the predicted and actual scaled $\lambda_{emission}$ values (**Table 2**). In both data sets, one encoding outperformed the other. The georgieV encoding achieved a medium correlation value of 0.4977 when training and predicting with the RFP only data set, while the one-hot encoding achieved a low correlation of 0.3945 for all molecular models. The predictions of the best performing encoding type for each dataset linear regressions reveal a weak correlation indicated by a 0.2324 and 0.2478 R2.

	One-hot		georgieV	
	ρ	P-value	ρ	P-value
RFP	-0.1015	$4.714e^{-3}$	0.4977	$6.186e^{-23}$
All	0.3945	$4.564e^{-40}$	0.0761	$1.409e^{-2}$

Table 2. Pearson ρ Correlation and P-values regressions reveal a weak correlation demonstrating georgieV outperforms one-hot indicated by a 0.2324 and 0.2478 R^2 encoding with RFP only, while one-hot encoding (**Figure 3**). outperforms georgieV when trained with all.

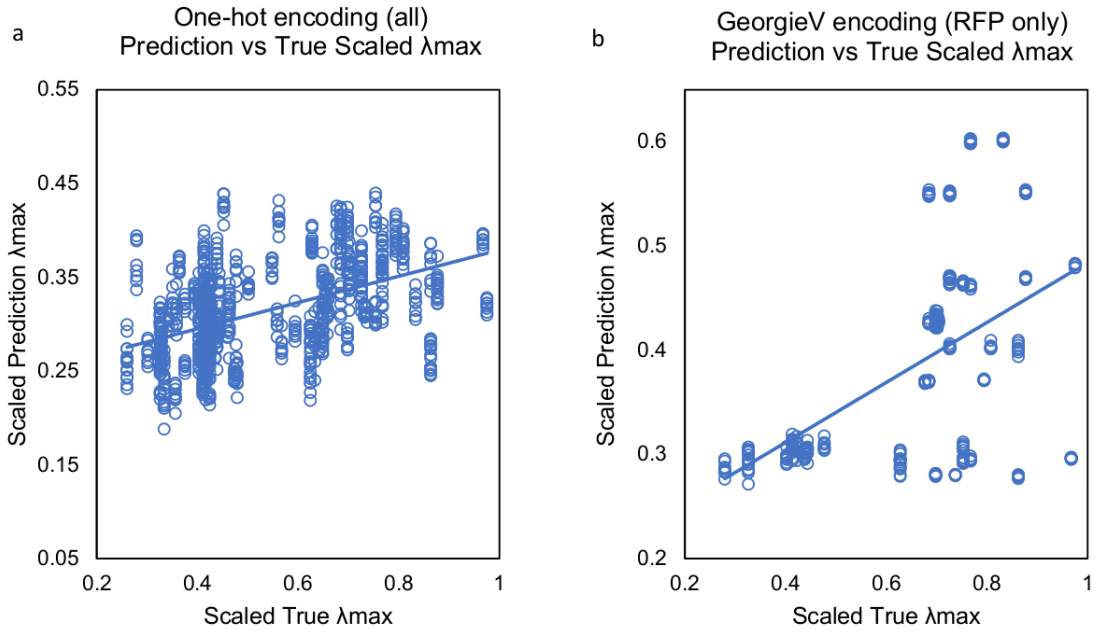


Figure 3. Linear regression plots for all FPCNN models that trained on the total fp data set and subset. Values are scaled according to the distribution of proteins. (a) the high-density region of data points noticeable at x = 0.4 corresponds to the disproportionate amount of green fluorescent proteins and derivatives trained. The R^2, 0.2324, displays a weak correlation between the Scaled True and Scaled Predicted values ($\lambda_{emission}$). y = 0.1401x + 0.2391 is the regression. (b) Qualitatively, a noticeable trend with respect to the regression line is the sparsity of the predictions towards the scaled value 1, which may represent worse predictions for non-monomeric far-red fps. The R^2, 0.2478, displays a weak correlation between the Scaled True and Scaled Predicted values ($\lambda_{emission}$). y = 0.288x + 0.1963 is the regression.

The unexpected correlation values suggest that there may not have been a clever diversification of the data. Of the 130 molecular models, only 36 were monomers, thus the encoded crystal structures may not be representative of key components that stabilize the chromophore for many proteins. Additionally, the georgieV encoding is unable to assign a descriptive value to any chromophore because it cannot be described within the 21 canonical amino acids. This

limitation may be indicated by virtually no correlation present with ρ of 0.0761 in the data set with 95 additional (not rfp) proteins. The correlation values may also have been impacted by the variation between every model trained in each fold. Each fold tested on a different subset of molecular models, which was chosen through a random seed, so controlling the ratio of monomers relative to the entire test set and differing $\lambda_{emission}$ was not achieved.

Conclusions

Although the machine learning model is not yet refined, the medium correlation provided by the georgieV encoded trained prediction suggests that three-dimensional convolutional neural networks can extract features relevant to the molecular model. Future efforts may include a binary categorization. Instead of predicting the $\lambda_{emission}$, the new CNN model could predict red fluorescence or not. Although the simplification of the problem could allow for greater accuracy, it would not be as robust to suggest a fully automated learning of biological features relevant to functional properties of biomolecules.

Notably, in both the one-hot and georgieV encoding, the chromophore and its function cannot be descriptively understood by FPCNN. Perhaps, a new approach that captures the subtlety in electronic states between red-shifted proteins could more precisely predict $\lambda_{emission}$. With further improvement, a refined FPCNN model could be used to predict the $\lambda_{emission}$ for computational molecular models not yet synthesized in the lab. In theory, this approach would

narrow the search and give some insight as to the necessary chromophore environment for far-red fluorescence.

References

(1) Lambert, T. J. FPbase: A Community-Editable Fluorescent Protein Database. *Nature Methods* **2019**, *16* (4), 277–278. https://doi.org/10.1038/s41592-019-0352-8.

(2) Basic Principles of Fluorescence Spectroscopy. In *Handbook of Fluorescence Spectroscopy and Imaging*; John Wiley & Sons, Ltd, 2011; pp 1–30. https://doi.org/10.1002/9783527633500.ch1.

(3) Shcherbo, D.; Merzlyak, E. M.; Chepurnykh, T. V.; Fradkov, A. F.; Ermakova, G. V.; Solovieva, E. A.; Lukyanov, K. A.; Bogdanova, E. A.; Zaraisky, A. G.; Lukyanov, S.; Chudakov, D. M. Bright Far-Red Fluorescent Protein for WholeBody Imaging. *Nature Methods* **2007**, *4* (9), 741–746. https://doi.org/10.1038/nmeth1083.

(4) Shcherbo, D.; Murphy, C. S.; Ermakova, G. V.; Solovieva, E. A.; Chepurnykh, T. V.; Shcheglov, A. S.; Verkhusha, V. V.; Pletnev, V. Z.; Hazelwood, K. L.; Roche, P. M.; Lukyanov, S.; Zaraisky, A. G.; Davidson, M. W.; Chudakov, D. M. Far-Red Fluorescent Tags for Protein Imaging in Living Tissues. *Biochemical Journal* **2009**, *418* (3), 567–574. https://doi.org/10.1042/BJ20081949.

(5) Shcherbakova, D. M.; Subach, O. M.; Verkhusha, V. V. Red Fluorescent Proteins: Advanced Imaging Applications and Future Design. *Angewandte Chemie International Edition* **2012**, *51* (43), 10724–10738. https://doi.org/10.1002/anie.201200408.

(6) Patterson, J.; Gibson, A. *Deep Learning: A Practitioner's Approach*; O'Reilly Media, Inc., 2017.

(7) Krizhevsky, A.; Sutskever, I.; Hinton, G. E. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems 25*; Pereira, F., Burges, C. J. C., Bottou, L., Weinberger, K. Q., Eds.; Curran Associates, Inc., 2012; pp 1097–1105.

(8) Cohen-Or, D.; Kaufman, A. Fundamentals of Surface Voxelization. *Graphical Models and Image Processing* **1995**, *57* (6), 453–461. https://doi.org/10.1006/gmip.1995.1039.

(9) Wittmann, B. J.; Yue, Y.; Arnold, F. H. Machine Learning-Assisted Directed Evolution Navigates a Combinatorial Epistatic Fitness Landscape with Minimal Screening Burden; preprint; *Bioinformatics*, 2020. https://doi.org/10.1101/2020.12.04.408955.

(10) Shoemake, K. Animating Rotation with Quaternion Curves. In *Proceedings of the 12th annual conference on Computer graphics*

and interactive techniques; SIGGRAPH '85; Association for Computing Machinery: New York, NY, USA, 1985; pp 245–254. https://doi.org/10.1145/325334.325242.

(11) Szegedy, C.; Wei Liu; Yangqing Jia; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going Deeper with Convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: Boston, MA, USA, 2015; pp 1–9. https://doi.org/10.1109/CVPR.2015.7298594.

(12) Arora, S.; Bhaskara, A.; Ge, R.; Ma, T. Provable Bounds for Learning Some Deep Representations. *Proceedings of the 31st International Conference on Machine Learning,* **2014***.*

(13) Fully Connected Layer: The brute force layer of a Machine Learning model https://iq.opengenus.org/fully-connected-layer/ (accessed Jul 7, 2020).

(14) Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Proceedings of the 31st International Conference on Machine Learning*, **2014**.

(15) Wannier, T. M.; Gillespie, S. K.; Hutchins, N.; McIsaac, R. S.; Wu, S.-Y.; Shen, Y.; Campbell, R. E.; Brown, K. S.; Mayo, S. L. Monomerization of Far-Red Fluorescent Proteins. *Proceedings of the National Academy of Science* **2018**, *115* (48), E11294–E11301. https://doi.org/10.1073/pnas.1807449115.

(16) Madeira, F.; Park, Y. mi; Lee, J.; Buso, N.; Gur, T.; Madhusoodanan, N.; Basutkar, P.; Tivey, A. R. N.; Potter, S. C.; Finn, R. D.; Lopez, R. The EMBL-EBI Search and Sequence Analysis Tools APIs in 2019. *Nucleic Acids Research* **2019**, *47* (W1), W636–W641. https://doi.org/10.1093/nar/gkz268.

(17) Wannier, T. M.; Moore, M. M.; Mou, Y.; Mayo, S. L. Computational Design of the β-Sheet Surface of a Red Fluorescent Protein Allows Control of Protein Oligomerization. *PLOS ONE* **2015**, *10* (6), e0130582. https://doi.org/10.1371/journal.pone.0130582.

This is the end of the manuscript.

1.3 Crystallography Package: Elucidating Density for Automatic Real Space Refinement

This project was initiated in September of 2019 as a collaboration between the Mayo Lab and the Genomics Institute of the Novartis Research Foundation, to create a machine learning tool to automate the reconstruction of Cartesian coordinates suitable for the PDB format from electron density maps.

The primary motivation for this project is the difficulty in refining large macromolecular structures when the initial model generated is outside of the radius of convergence. We expect that this would save users many hours of manually dragging residues in Coot, and be especially useful when the complete sequence of the protein is not known.

In the course of this work, we authored Python 3 wrappers for Phenix (Liebschner et al., 2019), the foremost electron density map processing tool. This included supporting tools to interoperate with the RCSB to fetch and prepare training data suitable for training models. These tools use a list of PDB codes, from which structure factor data and completed PDB files are retrieved. Structure factor data is processed with Phenix's Xtriage tool to generate MTZ reflection files, before processing with the maps tool and FFT tools to generate CCP4 electron density maps. These steps are completed in advance of any machine learning and are coded so as to allow for efficient partially parallel processing. In our tests, preparing 3300 maps takes approximately 36 hours on a 32 core workstation. The resulting ccp4 electron density maps can then be stored for further processing and featurization.

To enable simple processing with TensorFlow, a Keras Sequence object and TensorFlow data pipeline have been written which read in the ccp4 files, extract the unit cell information, and orthogonalize the ccp4 maps using the Gemmi library. These maps are then converted to dense tensors of floats which represent normalized electron density at each voxel in a 64x64x64 grid, where each voxel is a 1 Angstrom cube. The associated PDB maps are read in parallel to generate voxel labels which are one-hot encoded either as residue or atom types. These inputs and labels are generated together on the fly in batches which saves >99% of the storage space which would be required to precalculate and store them on disk. Because the convolutional network architecture employed by our models mandate that inputs and outputs maintain consistent sizes, we have enacted a routine to slice the electron density maps (inputs) and voxel maps of atomic positions (outputs) to fixed sizes, and conversely a routine for reconstructing full size atomic maps from a series of submaps.

The full code has been collected as a Github repository and will be released for others to adapt and build on. While we have implemented several network architectures in the repository, including SegNet (Badrinarayanan et al., 2016) and U-Nets (Ronneberger et al., 2015), we have not been able to achieve acceptable performance, and have had to spend significant time debugging various aspects of the data cleaning and streaming. Both of the main contributors to this project took on increased commitments to other projects (one a promotion to institute level leadership, and one a new sponsored research agreement), and accordingly this project will need to be carried on further by others using the code

we have released. The work we have done will simplify future efforts to continue to address this problem; using our repository, one can quickly narrow their focus to designing neural networks without concern for the complexities of Phenix, Gemmi, or the file formats.

1.4 Low data regimes - Transfer learning and Siamese networks: Protein solubility and serum albumin binding

This project originated in early 2019 in response to an RFP from Novo Nordisk for computational methods which could predict the plasma protein binding of small-molecule-peptide conjugates, such as the GLP-1 agonist Semaglutide. Serendipitously, I had recently been exploring data from eSOL (Niwa et al., 2009), a comprehensive characterization of the solubility of E. coli proteins. Building upon VoxLearn, I cross-referenced this data with structural data from RCSB and set out to train models which used protein structures to predict aqueous solubility. Initially this worked poorly, as we had a highly restricted dataset of only 130 proteins. Drawing from image processing literature, I decided to experiment with using a Siamese network architecture (Koch et al., 2015) as this setup has been known to perform well in low data settings. In the course of our work, it was discovered that networks trained in this way were capable of ranking proteins on the basis of solubility with a high degree of accuracy. Conjecturing that aqueous solubility is related to human serum albumin (HSA) binding, I next attempted to adapt the trained aqueous solubility model to predicting HSA binding. HSA is the most abundant protein in blood plasma

(Parviainen et al., 2011), and serves in a variety of roles, including as a carrier for aliphatic drug molecules which are otherwise poorly soluble. In this work, we examined the applicability of our trained solubility model to a variety of datasets, and found that it is capable of predicting HSA binding with a modest degree of accuracy, subject to several conditions. Notably, the method works better on smaller organic molecules rather than large macrocycles or polypeptides. We believe this may be related to the conformational complexity of these molecules, as unlike the protein dataset upon which we trained our original model, which had corresponding crystal structures, we must generate conformers for test data using non-exhaustive search algorithms.

In the summer of 2020, I additionally used this project to teach machine learning and cheminformatics basics to an undergraduate SURF student. Over the course of three months, this highly talented student recreated the work I had originally performed and continued by systematically exploring the effects of increasing the number of data augmentations and pairwise comparisons in the task of ranking.

The key difference between transfer learning and standard representation learning is that models are trained on a domain which has a significant amount of data available and subsequently applied to a different task, one where data is typically relatively more scarce (Ruder, 2017). The Siamese network architecture described above processes two input data in parallel, however the learned weights and biases from each input are tied until the last layers of the network, where the intermediate representations of the two data are compared to make a

conclusion about the relative label of the two data. In the case of the network we constructed, the output simply describes which of the two molecules input is more soluble (label 1) or less soluble (label 0).

As part of this project, we illuminated key features for the model to learn well (aside from the tuning of model hyperparameters), including the number of pairs, the number of augmentations of the data, and the method of generating conformers. To adapt a Siamese network to predicting properties of sets, we aggregate the pairwise comparisons to create a ranking problem, which performs multiple comparisons for each data in the input set before collecting the total number of “more soluble” (label 1) votes each protein received. Because the number of pairs of data points to compare rapidly expands as a function of the dataset size, as calculated by $C(n,r) = \frac{n!}{(r!(n-r)!)}$, we experimented with using less than exhaustive sets of random pairs. Additionally, we examined the effects of augmenting each input point by randomly rotating and transposing within the input voxel grid using VoxLearn. Finally, because the small molecule datasets we were working with do not have experimentally determined three-dimensional structures, we have compared six different methods for generating conformers. To determine the relative contributions of each of these factors, we have evaluated a pretrained model by prediction on a range of small molecule sets.

Results

To limit the search space required, we first evaluated a range of model architectures on a single dataset, eSol, before attempting to apply this model to additional datasets. Using automated hyperparameter tuning, we selected the model producing the lowest mean squared error. In general the best performing models had lower dropout rates and slower learning rates, and a higher number of pairs was better for the model, but with diminishing returns. In the training set, there are roughly 130 proteins. Using a number of pairs set to 40 provided the best result, (better than 1, 3, 5, 10, or 20 pairs). Setting the number to 80 pairs performed worse, and even worse (and slower) was setting the number of pairs to *n pick 2*.

Next we evaluated the effect of increasing the number of augmentations (rotations and translations) that were applied to the input data representations. By testing and training the model on more augmentations of the 3D molecules, the model was better able to predict the properties of the molecules, although as before, a point of diminishing returns was reached at 9 augmentations. Spearman rho was used to evaluate the fidelity of the predicted solubility, with the model producing rho of 0.62 (Figure 1).

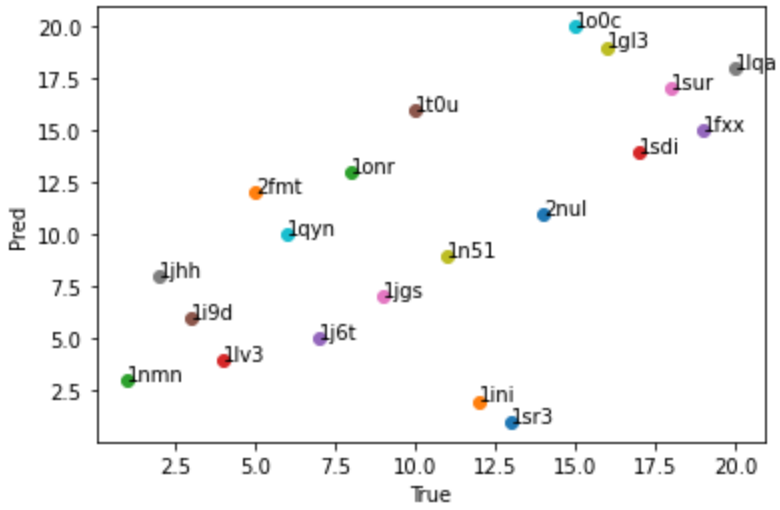


Figure 1: Predicted rank vs. True rank for proteins. In the graph, we tested our best, model and then calculated its estimate of the proteins' solubility based on its prediction. Taking that number, we ranked the proteins in order of decreasing solubility and compared it with the actual rank of the proteins' solubility. The spearman rho of this graph is 0.62. Presented results are ten fold cross-validated.

Following hyperparameter testing, we used the best model to make predictions on several data sets of small molecules from ChEMBL which describe binding affinity to serum albumin. No fine tuning of the models was performed on this new problem. In our initial efforts, we used Open Babel's gen3D operation (O'boyle et a., 2011) to generate conformers from the 1D SMILES format provided in ChEMBL, but recognizing that this is a key aspect of the workflow, we additionally embarked to evaluate the effects of varying the stringency of the conformer search, as well as including two commercial conformer generation packages, from Schrodinger (Watts et al., 2011), and Chemaxon (cxcalc).

Primary Results

Methods	CHEMBL3110981	CHEMBL1953042	CHEMBL633673	CHEMBL809966	CHEMBL888690	CHEMBL894775
schrodinger	ρ=0.713 p=0.0009	ρ=-0.112 p=0.5956	ρ=0.079 p=0.8287	ρ=0.723 p=0.0001	ρ=-0.1 p=0.798	ρ=0.444 p=0.0075
chemaxon	ρ=0.131 p=0.6042	ρ=0.06 p=0.7757	ρ=0.067 p=0.8548	ρ=0.624 p=0.0019	ρ=0.067 p=0.8647	ρ=0.327 p=0.0551
obabel_low	ρ=0.154 p=0.5424	ρ=-0.48 p=0.0177	ρ=0.309 p=0.3848	ρ=0.606 p=0.0028	ρ=0.717 p=0.0298	ρ=0.433 p=0.0094
obabel_med	ρ=0.393 p=0.1065	ρ=-0.457 p=0.0217	ρ=0.188 p=0.6032	ρ=0.703 p=0.0003	ρ=0.217 p=0.5755	ρ=0.083 p=0.6347
obabel_fast	ρ=0.245 p=0.328	ρ=-0.398 p=0.049	ρ=-0.248 p=0.4888	ρ=0.718 p=0.0002	ρ=0.483 p=0.1875	ρ=0.307 p=0.0726
obabel_best	ρ=0.193 p=0.4429	ρ=-0.2 p=0.3378	ρ=0.2 p=0.5796	ρ=0.726 p=0.0001	ρ=0.417 p=0.2646	ρ=-0.07 p=0.6894

Figure 2: Performance of transferred model with varying datasets and conformer generation routines. Methods correspond to software suites used to generate small molecule conformers. Obabel corresponds to OpenBabel using the gen3d command using low, med, fast, or best flags which effects a weighted rotor conformational search and conjugate gradient geometry optimization used in the process. Datasets are described in Table 1.

ChEMBL Dataset	Description of Assay	Description of Compounds Screened	Molecular Size / Description
CHEMBL3110981 (Han et al., 2013)	Binding affinity to human serum albumin at 100 ug/mL after 3 hrs relative to control	GLP-1 Conjugates of Dicoumarol	30-mer + linker + Dicoumarol
CHEMBL1953042 (Yang et al., 2012)	Binding affinity to human serum albumin by fluorescence quenching assay	Novel Gossypol derivatives	< 1000 MW small molecules
CHEMBL633673 (Knudsen et al., 2000)	Plasma half life determined in pigs	GLP-1 derivatives conjugated to fatty acids	30-mers + short chain fatty acids
CHEMBL809966 (Koehler et al., 2002)	In vitro binding affinity for rabbit serum albumin	Small organic molecules which are conjugated to the terminus of peptides	Mix of small linear 5-mers and large cyclic peptides
CHEMBL888690 (Svenson et al., 2007)	Binding affinity to BSA 1 by isothermal titration calorimetry	Short Cationic Antimicrobial Micropeptides	Chemically modified trimers
CHEMBL894775 (Šoškić et al., 2007)	Binding affinity to human serum albumin in vitro	Chemically modified indoles	< 280 MW

Table 1: Datasets evaluated for transfer learning. Six datasets characterizing the binding of small molecules and peptides were identified on ChEMBL and evaluated using the Siamese model trained on eSol.

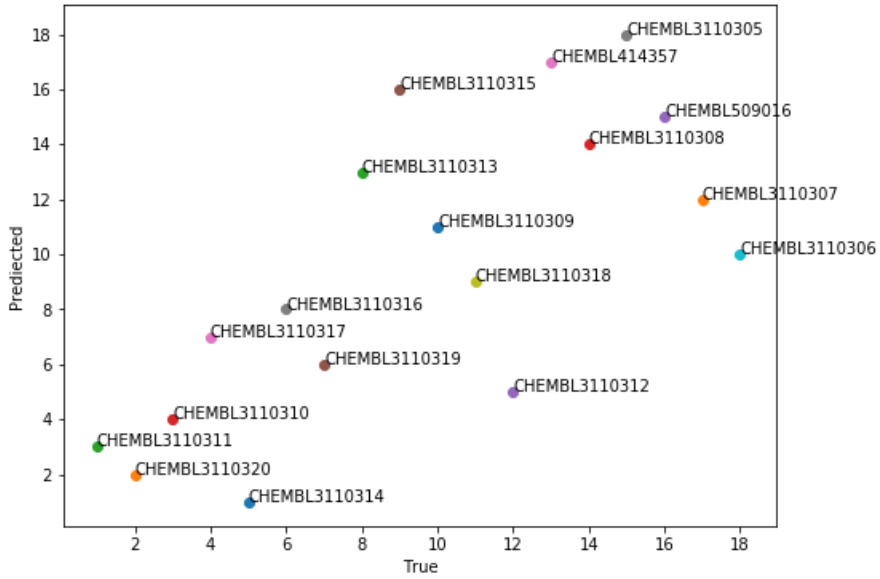


Figure 3: Predicted rank vs. True rank for small molecules. Here we tested our trained eSol model on CHEMBL3110981, a set of GLP-1 conjugates of Dicoumarol. This was the original dataset that inspired further exploration.

Discussion of Results

In examining these results, our initial observation is that the transferability of the model is highly dependent on the choice of conformer generation method. In general, the Schrodinger method, ConfGen, produced conformers that produced the strongest agreement with experimentally determined measures of serum albumin binding, in particular on assays 3110981 (Binding affinity to human serum albumin at 100 ug/mL after 3 hrs relative to control GLP-1 Conjugates of Dicoumarol, p=0.0009), 809966 (Binding affinity for rabbit serum albumin by mix of 5-mers and cyclic peptides, p=0.0001), and 894775 (binding of modified indoles to HSA p=0.0075). Notably OpenBabel outperformed the other paid competitor ChemAxon on two of the three datasets listed above.

Our second observation is that performance varied significantly between the six datasets. Of these, three datasets achieved p values which are statistically significant, as noted above. However, for the other three datasets, the p values achieved were quite poor (0.5956, 0.8287, 0.7980). In examining the experimental details of the six datasets, it is not immediately clear what may be influencing the performance of the transferred model. Of the three datasets which did not achieve statistically significant predictions with the transferred model using Schrodinger conformers, two were not assays of binding to human serum albumin, but rather bovine (in vitro) and porcine (in vivo). However, confounding the potential explanation of species differences is the fact that one dataset, CHEMBL3110981 performed poorly, despite using human serum albumin, and another, CHEMBL809966, performed very well, and used rabbit serum albumin. Variable size and complexity of the molecules in the different datasets is also not sufficient to explain the observed performance gap, as a mixture of datasets containing small molecules, peptides, and peptide-small molecule conjugates were evaluated, with no clear trend. To take the most conservative possible tack, we can state that with medium side peptides (larger than 5-mers) which were evaluated for binding against human serum albumin, the model performed well.

1.5 Leveraging existing datasets: Learning to active learn with submodular regularization

This project began when scrolling through a corpus of biochemical data representing a large investment of time and money and wondering how to make

use of it to aid in future, unrelated protein engineering projects. I worked on the project solo for approximately six months, before being invited by Dr. Yue to present it at his group's research meeting. At this meeting Dr. Yue, and his group members were interested in how I was adapting the algorithm from (Liu et al., 2018) to a batch setting, and a discussion was struck up with group members who were working on a method for data-driven normalization based on the submodular-norm loss. Ultimately, we merged our projects to form the paper we published at ICLR 2021. In writing this paper, my contributions were adapting the concept of submodularity to the active learning paradigm I had been previously working on, and conducting the published experiments on protein engineering. I contributed to writing all aspects of the paper except the set cover and Fashion MNIST experiments. I was also involved in the review and rebuttal process. I presented the paper as a poster alongside my co-author Ayya Alieva at ICLR, and also gave an invited talk based on the work at the AI LA Life Summit conference.

This work has also been made freely available as open source software on GitLab, and is also being employed by a graduate student in the Arnold lab for new domains of protein engineering.

LEARNING TO MAKE DECISIONS VIA SUBMODULAR REGULARIZATION

Ayya Alieva
Stanford University
ayya@stanford.edu

Aiden Aceves
Caltech
aaceves@caltech.edu

Jialin Song
Caltech
jssong@caltech.edu

Stephen Mayo
Caltech
steve@caltech.edu

Yisong Yue
Caltech
yyue@caltech.edu

Yuxin Chen
University of Chicago
chenyuxin@uchicago.edu

ABSTRACT

Many sequential decision making tasks can be viewed as combinatorial optimization problems over a large number of actions. When the cost of evaluating an action is high, even a greedy algorithm, which iteratively picks the best action given the history, is prohibitive to run. In this paper, we aim to learn a greedy heuristic for sequentially selecting actions as a surrogate for invoking the expensive oracle when evaluating an action. In particular, we focus on a class of combinatorial problems that can be solved via submodular maximization (either directly on the objective function or via submodular surrogates). We introduce a data-driven optimization framework based on the *submodular-norm* loss, a novel loss function that encourages the resulting objective to exhibit *diminishing returns*. Our framework outputs a surrogate objective that is efficient to train, approximately submodular, and can be made permutation-invariant. The latter two properties allow us to prove strong approximation guarantees for the learned greedy heuristic. Furthermore, our model is easily integrated with modern deep imitation learning pipelines for sequential prediction tasks. We demonstrate the performance of our algorithm on a variety of batched and sequential optimization tasks, including set cover, active learning, and data-driven protein engineering.

1 INTRODUCTION

In real-world automated decision making tasks we seek the optimal set of actions that jointly achieve the maximal utility. Many of such tasks — either deterministic/non-adaptive or stochastic/adaptive — can be viewed as combinatorial optimization problems over a large number of actions. As an example, consider the active learning problem where a learner seeks the maximally-informative set of training examples for learning a classifier. The utility of a training set could be measured by the mutual information (Lindley, 1956) between the training set and the remaining (unlabeled) data points, or by the expected reduction in generation error if the model is trained on the candidate training set. Similar problems arise in a number of other domains, such as experimental design (Chaloner and Verdinelli, 1995), document summarization (Lin and Bilmes, 2012), recommender system (Javdani et al., 2014), and policy making (Runge et al., 2011).

Identifying the optimal set of actions (e.g., optimal training sets, most informative experiments) amounts to evaluating the expected utility over a combinatorial number of candidate sets. When the underlying model class is complex and the evaluation of the utility function is expensive, these tasks are notoriously difficult to optimize (Krause and Guestrin, 2009). For a broad class of decision making problems whose optimization criterion is to maximize the decision-theoretic *value of information* (e.g., active learning and experimental design), it has been shown that it is possible to design surrogate objective functions that are (approximately) submodular while being aligned with the original objective at the optimal solutions (Javdani et al., 2014; Chen et al., 2015b; Choudhury et al., 2017). Here, the information gathering policies no longer aim to directly optimize the target objective value, but rather choose to follow a greedy trajectory governed by the surrogate function

that is much cheaper to evaluate. These insights have led to principled algorithms that enable significant gains in the efficiency of the decision making process, while enjoying strong performance guarantees that are competitive with the optimal policy.

Despite the promising performance, a caveat for these "submodular surrogate"-based approaches is that it is often challenging to engineer such a surrogate objective without an ad-hoc design and analysis that requires trial-and-error (Chen et al., 2015b; Satsangi et al., 2018). Furthermore, for certain classes of surrogate functions, it is NP-hard to compute/evaluate the function value (Javdani et al., 2014). In such cases, even a greedy policy, which iteratively picks the best action given the (observed) history, can be prohibitively costly to design or run. Addressing this limitation requires more automated or systematic ways of designing (efficient) surrogate objective functions for decision making.

Overview of main results. Inspired by contemporary work in data-driven decision making, we aim to learn a greedy heuristic for sequentially selecting actions. This heuristic acts as a surrogate for invoking the expensive oracle when evaluating an action. Our key insight is that many practical algorithms can be interpreted as greedy approaches that follow an (approximate) submodular surrogate objective. In particular, we focus on the class of combinatorial problems that can be solved via submodular maximization (either directly on the objective function or via a submodular surrogate). We highlight some of the key results below:

- Focusing on utility-based greedy policies, we introduce a data-driven optimization framework based on the "*submodular-norm*" loss, which is a novel loss function that encourages learning functions that exhibit "diminishing returns". Our framework, called LEASURE (Learning with Submodular Regularization), outputs a surrogate objective that is efficient to train, approximately submodular, and can be made permutation-invariant. The latter two properties allow us to prove approximation guarantees for the resulting greedy heuristic.
- We show that our approach can be easily integrated with modern imitation learning pipelines for sequential prediction tasks. We provide a rigorous analysis of the proposed algorithm and prove strong performance guarantees for the learned objective.
- We demonstrate the performance of our approach on a variety of decision making tasks, including set cover, active learning for classification, and data-driven protein design. Our results suggest that, compared to standard learning-based baselines: (a) at training time, LEASURE requires significantly fewer oracle calls to learn the target objective (i.e., to minimize the approximation error against the oracle objective); and (b) at test time, LEASURE achieves superior performance on the corresponding optimization task (i.e., to minimize the regret for the original combinatorial optimization task). In particular, LEASURE has shown promising performance in the protein design task and will be incorporated into a real-world protein design workflow.

2 RELATED WORK

Near-optimal decision making via submodular optimization. Submodularity is a property of a set function that has a strong relationship with diminishing returns, and the use of submodularity has wide applications from information gathering to document summarization (Leskovec et al., 2007; Krause et al., 2008; Lin and Bilmes, 2011; Krause and Golovin, 2014). The maximization of a submodular function has been an active area of study in various settings such as centralized (Nemhauser et al., 1978; Buchbinder et al., 2014; Mitrovic et al., 2017), streaming (Badanidiyuru et al., 2014; Kazemi et al., 2019; Feldman et al., 2020), continuous (Bian et al., 2017b; Bach, 2019) and approximate (Horel and Singer, 2016; Bian et al., 2017a). Variants of the greedy algorithm, which iteratively selects an element that maximizes the marginal gain, feature prominently in the algorithm design process. For example, in the case of maximizing a monotone submodular function subject to a cardinality constraint, it is shown that the greedy algorithm achieves an approximation ratio of $(1 - 1/e)$ of the optimal solution (Nemhauser et al., 1978).

In applications where we need to make a sequence of decisions, such as information gathering, we usually need to adapt our future decisions based on past outcomes. Adaptive submodularity is the corresponding property where an adaptive greedy algorithm enjoys a similar guarantee for maximizing an adaptive submodular function (Golovin and Krause, 2011). Recent works have explored optimizing the value of information (Chen et al., 2015b) and Bayesian active learning (Javdani et al., 2014; Chen et al., 2017a) with this property. Another line of related work is online setting (typically

bandits), which is grounded in minimizing cumulative regret (Radlinski et al., 2008; Streeter et al., 2009; Yue and Guestrin, 2011; Ross et al., 2013; Yu et al., 2016; Hiranandani et al., 2020).

Learning submodular functions. Early work focused on learning non-negative linear combinations of submodular basis functions (Yue and Joachims, 2008; El-Arini et al., 2009; Yue and Guestrin, 2011; Sipos et al., 2012), which was later generalized to mixtures of "submodular shells" (Lin and Bilmes, 2012). Deep submodular functions (Dolhansky and Bilmes, 2016) extend these ideas to more expressive compositional function classes by using sums of concave composed with modular functions. The theoretical question of the learnability of general submodular functions is analyzed in Balcan and Harvey (2018). Our goal is to encourage submodularity via regularization, rather than via hard constraints on the function class design.

Learning to optimize via imitation learning. Rather than first learning a submodular function and then optimizing it, one can instead learn to directly make decisions (e.g., imitate the oracle greedy algorithm). This area builds upon imitation learning, which learns a policy (i.e., a mapping from states to actions) *directly* from examples provided by an expert (e.g., an expensive computational oracle, or a human instructor) (Chernova and Thomaz, 2014). Classic work on imitation learning (e.g., the Dataset Aggregation (DAgger) algorithm (Ross et al., 2011)) reduce the policy learning problem to the supervised learning setting, which has been extended to submodular optimization by imitating the greedy oracle method (Ross et al., 2013). More generally, learning to optimize has been applied generically to improve combinatorial optimization solvers for focused distributions of optimization problems (He et al., 2014; Song et al., 2018; Khalil et al., 2016; Balunovic et al., 2018; Gasse et al., 2019; Song et al., 2020). Our approach bridges learning to optimize and learning submodular functions, with a focus on learning surrogate utilities using submodular regularization.

Learning active learning. Our approach is applicable to active learning, and so is related to work on learning active learning. The closest line of work learns a utility function as a surrogate for improvement in classifier accuracy (Konyushkova et al., 2017; Liu et al., 2018), which is then used as the decision criterion. However, prior work either used restricted function classes (Konyushkova et al., 2017), or very expressive function classes that can be hard to fit well (Liu et al., 2018). Our work can be viewed as a direct extension of this design philosophy, where we aim to reliably learn over expressive function classes using submodular regularization. Other related work do not directly learn an active learning criterion, instead encouraging sample diversity using submodularity (Wei et al., 2015) or the gradient signal from the classifier (Ash et al., 2020).

3 Background and Problem Statement

3.1 Decision Making via Submodular Surrogates

Given a ground set of items $\mathcal{V}$ to pick from, let $u : 2^{\mathcal{V}} \to \mathbb{R}$ be a set function that measures the *value* of any given subset[1] $A \subseteq \mathcal{V}$. For example, for experimental design, $u(A)$ captures the utility of the output of the best experiment; for active learning $u(A)$ captures the generalization error after training with set A. We denote a policy $\pi : 2^{\mathcal{V}} \to \mathcal{V}$ to be a partial mapping from the set/sequence of items already selected, to the next item to be picked. We use Π to denote our policy class. Each time a policy picks an item $e \in \mathcal{V}$, it incurs a unit cost. Given the ground set $\mathcal{V}$, the utility function u, and a budget k for selecting items, we seek the optimal policy π that achieves the maximal utility:

$$\pi^* \in \arg\max_{\pi \in \Pi} u(S_{\pi,k}). \tag{1}$$

$S_{\pi,k}$ is the sequence of items picked by π: $S_{\pi,i} = S_{\pi,i-1} \cup \{\pi(S_{\pi,i-1})\}$ for $i > 0$ and $S_{\pi,0} = \emptyset$.

As we have discussed in the previous sections, many sequential decision making problems can be characterized as constrained monotone submodular maximization problem. In those scenarios u is:

- **Monotone**: For any $A \subseteq \mathcal{V}$ and $e \in \mathcal{V} \setminus A$, $u(A) \leq u(A \cup \{e\})$.
- **Submodular**: For any $A \subseteq B \subseteq \mathcal{V}$ and $e \in \mathcal{V} \setminus B$, $u(A \cup \{e\}) - u(A) \geq u(B \cup \{e\}) - u(B)$.

[1]For simplicity, we focus on deterministic set functions in this section. Note that many of our results can easily extent to the stochastic, by leveraging the theory of adaptive submodularity (Golovin and Krause, 2011)

In such cases, a mypopic algorithm following the greedy trajectory of u admits a near-optimal policy. However, in many real-world applications, u is not monotone submodular. Then one strategy is to design a surrogate function $f : 2^{\mathcal{V}} \to \mathbb{R}$ which is:

- Globally aligning with u: For instance, f lies within a factor of u: $f(A) \in [c_1 \cdot u(A), c_2 \cdot u(A))]$ for some constants c_1, c_2 and any set $A \subseteq \mathcal{V}$; or within a small margin with u: $f(A) \in [u(A) - \epsilon, u(A) + \epsilon]$ for a fixed $\epsilon > 0$ and any set $A \subseteq \mathcal{V}$;
- Monotone submodular: Intuitively, a submodular surrogate function encourages selecting items that are beneficial in the long run, while ensuring that the decision maker does not miss out any actions that are "surprisingly good" by following a myopic policy (i.e., future gains for any item are diminishing). Examples that fall into this category include machine teaching (Singla et al., 2014), active learning (Chen et al., 2015a), etc.

We argue that in real-world decision making scenarios—as validated later in Section 6—the decision maker is following a surrogate objective that aligns with the above characterization. In the following context, we will assume that such surrogate function exists. Our goal is thus to learn from an *expert policy* that behaves greedily according to such surrogate functions.

3.2 Learning to Make Decisions

We focus on the regime where the expert policy is expensive to evaluate. Let $g : 2^{\mathcal{V}} \times \mathcal{V} \to \mathbb{R}$ be the score function that quantifies the benefit of adding a new item to an existing subset of $\mathcal{V}$. For the expert policy and submodular surrogate f discussed in Section 3.1, $\forall A \subseteq \mathcal{V}$ and $e \in \mathcal{V}$:

$$g^{\text{exp}}(A, e) = f(A \cup \{e\}) - f(A).$$

For example, in the active learning case, $g^{\text{exp}}(A, e)$ could be the expert acquisition function that ranks the importance of labelling each unlabelled point, given the currently labelled subset. In the set cover case, $g^{\text{exp}}(A, e)$ could be the function that gives the score to each vertex and determines the next best vertex to add to the cover set. Given a loss function ℓ, our goal is to learn a score function $\hat{g}$ that incurs the minimal expected loss when evaluated against the expert policy: $\hat{g} = \arg\min_g \mathbb{E}_{A,e}[\ell(g(A, e), g^{\text{exp}}(A, e))]$. Subsequently, the utility by the learned policy is $u(S_{\hat{\pi},k})$, where for any given history $A \subseteq \mathcal{V}$, $\hat{\pi}(A) \in \arg\max_{e \in \mathcal{V}} \hat{g}(A, e)$.

4 Learning with Submodular Regularization

To capture our intuition that a greedy expert policy tends to choose the most useful items, we introduce LeaSuRe, a novel regularizer that encourages the learned score function (and hence surrogate objective) to be submodular. We describe the algorithm below.

Given the groundset $\mathcal{V}$, let $f : 2^{\mathcal{V}} \to \mathbb{R}$ be any approximately submodular surrogate such that $f(A)$ captures the "usefulness" of the set A. The goal of a trained policy is to learn a score function $g : 2^{\mathcal{V}} \times \mathcal{V} \to \mathbb{R}$ that mimics $g^{\text{exp}}(A, x) = f(A \cup \{x\}) - f(A)$, which is often prohibitively expensive to evaluate exactly. Then, given any such g, we can define a greedy policy $\pi(A) = \text{argmax}_{x \in \mathcal{V}} g(A, x)$. With LeaSuRe, we aim to learn such function g that approximates g^{exp} well while being inexpensive to evaluate at test time. Let $D_{real} = \{(\langle A, x\rangle, y^{\text{exp}} = g^{\text{exp}}(A, x))\}_m$ be the gathered tuple of expert scores for each set-element pair. If the set $2^{\mathcal{V}} \times \mathcal{V}$ was not too large, the LeaSuRe could be trained on the randomly collected tuples D_{real}. However, $2^{\mathcal{V}}$ tends to be too large to explore, and generating ground truth labels could be very expensive. To leverage that, for a subset of set-element pairs in D_{real} we generate a set of random supersets to form an unsupervised synthetic dataset of tuples $D_{synth} = \{(\langle A, x\rangle, \langle A', x\rangle) | A \preceq A', \langle A, x\rangle \in D_{real}\}_n$ where A' denote a randomly selected superset of A. Define:

$$\text{Loss}(g, g^{\text{exp}}) = \sum_{\langle A,x\rangle, y^{\text{exp}} \in D_{real}} (y^{\text{exp}} - g(A, x))^2 + \lambda \sum_{(\langle A,x\rangle, \langle A',x\rangle) \in D_{synth}} \sigma([g(A', x) - g(A, x)]),$$

where $\lambda > 0$ is the regularization parameter and σ is the sigmoid function. Intuitively, such regularization term will force the learned function g to be close to submodular, as it will lead to larger losses every time $g(A', x) > g(A, x)$. If we expect f to be monotonic, we also introduce a second regularizer $\text{ReLu}(-g(A', x))$ which pushes the learned function to be positive. Combined, the loss

function becomes (used in Line 11 in Algorithm 1):

$$\text{Loss}(g, g^{\text{exp}}) = \sum_{\langle A,x\rangle, y^{\text{exp}} \in D_{real}} (y^{\text{exp}} - g(A,x))^2 + \lambda \sum_{(\langle A,x\rangle, \langle A',x\rangle) \in D_{synth}} \sigma([g(A',x) - g(A,x)]) + \gamma \sum_{\langle A',x\rangle \in D_{synth}} \text{ReLu}(-g(A',x)),$$

where γ is another regularization strength parameter. Such loss should push g to explore a set of approximately submodular, approximately monotonic functions. Thus, if f exhibits the submodular and monotonic behavior, g trained on this loss function should achieve a good local minima.

We next note that since $2^{\mathcal{V}}$ is too large to explore, instead of sampling random tuples for D_{real}, we use modified DAgger. Then g can learn not only from the expert selections of $\langle A, x\rangle$, but it can also see the labels of the tuples the expert would not have chosen.

Algorithm 1 Learning to make decisions via Submodular Regularization (LEASURE)

1: **Input**: Ground set $\mathcal{V}$, expert score function g^{exp},
2: regularization parameters λ, γ, DAgger constant β, the length of trajectories T.
3: initialize $D_{real} \leftarrow \emptyset$
4: initialize g to any function.
5: **for** $i = 1$ to N **do**
6: Let $g_i = g^{\text{exp}}$ with probability β.
7: Sample a batch of $T-$step trajectories using $\pi_i(A) = x_i = \text{argmax}_{x \in \mathcal{V}} g_i(A, x)$.
8: Get dataset $D_i = \{\langle A_i, x_i\rangle, g^{\text{exp}}(A_i, x_i)\}$ of labeled tuples on actions taken by π_i.
9: $D_{real} \leftarrow D_{real} \bigcup D_i$.
10: Generate synthetic dataset D_{synth} from D_{real}.
11: Train g_{i+1} on D_{real} and D_{synth} using the loss function above.
12: **Output**: g_{N+1}

Algorithm 1 above describes our approach. A trajectory in Line 7 is a sequence of iteratively chosen tuples, $(\langle \emptyset, x_1\rangle, \langle \{x_1\}, x_2\rangle, \langle \{x_1, x_2\}, x_3\rangle ..., \langle \{x_1, ..., x_{T-1}\}, x_T\rangle)$, collected using a mixed policy π_i. In Line 8, expert feedback of selected actions is collected to form D_i. Note that in some settings, even collecting exact expert labels g^{exp} at train time could be too expensive. In that case, g^{exp} can be replaced with a less expensive, noisy approximate expert $g_\epsilon^{\text{exp}} \approx g^{\text{exp}}$. In fact, all three of our experiments use noisy experts in one form or another.

5 ANALYSIS

Estimating the expert's policy. We first consider the bound on the loss of the learned policy measured against the expert's policy. Since LEASURE can be viewed as a specialization of DAGGER (Ross et al., 2011) for learning a submodular function, it naturally inherits the performance guarantees from DAGGER, which show that the learned policy efficiently converges to the expert's policy. Concretely, the following result, which is adapted from the original DAgger analysis, shows that the learned policy is consistent with the expert policy and thus is a *no-regret* algorithm:

Theorem 1 (Theorem 3.3, Ross et al. (2011))**.** *Denote the loss of $\hat{\pi}$ at history state H as $l(H, \hat{\pi}) := \ell(g(H, \hat{\pi}(H)), g^{exp}(H, \pi^{exp}(H)))$. Let $d_{\hat{\pi}}$ be the average distribution of states if we follow $\hat{\pi}$ for a finite number of steps. Furthermore, let D_i be a set of m random trajectories sampled with π_i at round $i \in \{1, \ldots, N\}$, and $\hat{\epsilon}_N = \min_\pi \frac{1}{N} \sum_{i=1}^{N} \mathbb{E}_{H_i \sim D_i}[l(H_i, \hat{\pi})]$ be the training loss of the best policy on the sampled trajectories. If N is $\mathcal{O}\left(T^2 \log(1/\delta)\right)$ and m is $\mathcal{O}(1)$ then with probability at least $1 - \delta$ there exists a $\hat{\pi}$ among the N policies, with $\mathbb{E}_{H \sim d_{\hat{\pi}}}[l(H, \hat{\pi})] \leq \hat{\epsilon}_N + \mathcal{O}\left(\frac{1}{T}\right)$.*

Approximating the optimal policy. Note that the previous notion of regret corresponds to the average difference in score function between the learned policy and the expert policy. While this result shows that LEASURE is consistent with the expert, it does not directly address how well the learned policy performs in terms of the gained utility. We then provide a bound on the expected value of the learned policy, measured against the value of the optimal policy. For specific decision making tasks where the oracle follows an approximately submodular objective, our next result, which is proved in the appendix, shows that the learned policy behaves near-optimally.

Theorem 2. *Assume that the utility function u is monotone submodular. Furthermore, assume the expert policy π^{exp} follows a surrogate objective f such that for all $A \subseteq \mathcal{V}$, $|f(A) - u(A)| < \epsilon_E$ where $\epsilon_E > 0$. Let $\hat{\epsilon}_N = \min_\pi \frac{1}{N} \sum_{i=1}^{N} l(H_i, \hat{\pi})$ be the training loss of the best policy on the sampled trajectories. If N is $\mathcal{O}\left(T^2 \log(1/\delta)\right)$ then with probability at least $1 - \delta$, the expected utility achieved by running $\hat{\pi}$ for k steps is*

$$\mathbb{E}[u(S_{\hat{\pi},k})] \geq (1 - 1/e)\mathbb{E}[u(S_{\pi^*,k})] - k(\epsilon_E + \Delta_{\max}\hat{\epsilon}_N) - O(1).$$

A closely related work in approximate policy learning is by Ross et al. (2013), which also builds upon DAGGER to tackle policy learning for submodular optimization, via directly imitating the greedy oracle decision rather than learning a surrogate utility. One key difference is that their approach can only yield guarantees against an artificial benchmark (a set or list of simpler policies that each independently selects an item to add to the action set), whereas our theoretical guarantees are with respect to the optimal policy in our class.

6 EXPERIMENTS

In this section, we demostrate the performance of LEASURE on three diverse sequential decision making tasks, namely set cover (SC), learning active learning (LAL) and protein engineering (PE).

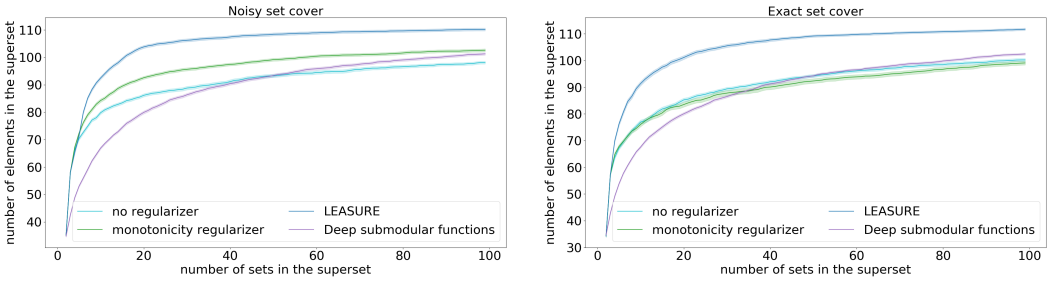
Baselines. We compare our approach to the Deep Submodular Function (DSF (Dolhansky and Bilmes, 2016)) and Deep Batch Active Learning by Diverse, Uncertain Gradient Lower Bounds (BADGE (Ash et al., 2020)). The DSF approach learns a submodular surrogate function $f : 2^{\mathcal{V}} \to \mathbb{R}$ that produces a score for each set $A \subset \mathcal{V}$. The architecture of the DSF forces the function f to be exactly submodular, as opposed to LEASURE, which is only encouraged to be submodular through a regularizer. However, the architecture and the training procedure of the DSF are quite restrictive, which does not allow the DSF to explore a large domain during training and restricts how expressive it can be compared to a standard neural network. Moreover, DSF are restricted to small $\mathcal{V}$, and the number of parameters increases with the cardinality of $\mathcal{V}$. That is not true for LEASURE, which number of parameters grows with the dimensionality of elements in $\mathcal{V}$. This makes DSF useful for small datasets, but makes it prohibitively expensive to use on larger problems. In fact, we could not compare LEASURE to DSF on LAL or PE tasks, as it was not feasible to train DSF on these sets. For LAL experiment, we also compare with a recent deep active learning approach (Ash et al., 2020). Finally, we want to add that LEASURE can be seamlessly integrated with any standard Machine Learning library, and since the architecture of the learned policy in LEASURE is not restrictive, any available optimization trick can be used to achieve better performance. In fact, existing 'imitation learning'-based approaches for LAL, such as Liu et al. (2018), can be viewed as special cases of LEASURE (i.e. without regularization). On the other hand, DSF cannot be as easily implemented, and the standard libraries are not optimized for the DSF architecture.

6.1 SET COVER

Before testing our approach on a real-world scenario, we showcase its performance on a simple submodular and monotonic maximization problem. Set cover is a classical example: given a set of elements $U = \{1, 2, ..., n\}$ (called the universe) and a collection of m sets $S = \{s_1, .., s_m\}$ whose union equals the universe, the set cover problem is to identify the smallest sub-collection of S whose union equals the universe. Formulated as a policy learning problem, the goal is to learn the score function $g : 2^S \times S \to \mathbb{R}$ such that for any $S_l \subset S, x \in S$,

$$g(S_l, x) \approx g^{\text{exp}}(S_l, x) = |\cup_{s \in S_l} s \cup x| - |\cup_{s \in S_l} s|.$$

Given g, we can then define a policy $\pi : 2^s \to S$ as $\pi(S_l) = \text{argmax}_{x \in S} g(S_l, x)$. During training, tuples $\{(S_l, x), g^{\text{exp}}\}$ are collected, and then g is trained on this set. We trained four different policies: a function g parametrized by a neural network with $MSE(g, g^{\text{exp}})$ as the loss, a function g with the same MSE loss and just a monotonicity regularizer, a function g trained using both monotonicity and submodular regularizers (LEASURE), as well as the Deep Submodular Function baseline (Dolhansky and Bilmes, 2016). We use a modified Deepset architecture (Zaheer et al., 2017) for modeling the permutation-invariant score networks g in both the SC and the LAL tasks, and provide the details in Appendix B. Our dataset is the subset of the Mushroom dataset (Lim, 2015), consisting of 1000 sets. Each set contains 23 mushroom species, and there are a total of 119



(a) (b)

Figure 1: Evaluating LEASURE against baselines on set cover instances

species. The goal is to train a policy to select the largest superset of these sets. We evaluate in two settings: Exact Set Cover, where we collect tuples $\{(S_l, x), g^{\text{exp}}\}$ for training, and Noisy Set Cover, where we have access only to $\{(S_l, x), g_\epsilon^{\text{exp}}\}$, where g_ϵ^{exp} is a noisy score. The networks are trained on rollouts of length 20 (i.e. on sets $\{S_l : |S_l| \leq 20\}$), and tested on rollout of length up to 100.

Figure 1 show the value of set cover as a function of the size of the superset. LEASURE significantly outperforms other learned policies, although Deep Submodular Function generalizes better to larger rollout lengths – LEASURE gets most of its set cover gains in the first 10-20 selected points, while Deep Submodular Function continues to noticeably improve past the training rollout length. Note that in Figures 1a & 1b, the competing baselines all exhibit a "diminishing returns" effect, resulting in a concave-shaped value function. With a submodular-norm regularizer, LEASURE quickly identified the sets with large marginal gains. This observation aligns with our analysis in Section 5.

6.2 LEARNING ACTIVE LEARNING ON FASHION MNIST

In this section we demonstrate the performance of LEASURE on a real-world task that is not submodular or monotonic, but usually exhibits submodular and monotonic behaviour.

In active learning, there is a partially labelled dataset $S = \{S_l, S_u\}$, where S_l is labelled and S_u is unlabelled, and a policy $\pi : 2^S \to S$. The labelled subset S_l can be used to infer from data (learn the image classifier, predict unlabelled protein fitness, etc). The goal of the policy is to select the smallest subset $S_\pi \subset S_u$ to label such that the accuracy of supervised learning from $S_\pi \cup S_l$ is maximized. Since selecting a subset is a prohibitively expensive combinatorial task, the policy is usually sequential. In particular, it selects points to add to S_π one by one (or in batches) using some score function $g(S_\pi \cup S_l, \cdot) : S_u \to \mathbb{R}$ to score each point $x \in S_u$ and then the policy labels the point with the largest score. If g were to be the first order difference of a submodular function f, i.e. $g(A, e) = f(A \cup \{e\}) - f(A)$, then the policy would be near-optimal. Moreover, as discussed above, intuitively we expect g to have this property in most cases, since adding an extra point to a larger set usually has less effect than adding the same point to a smaller subset of the set.

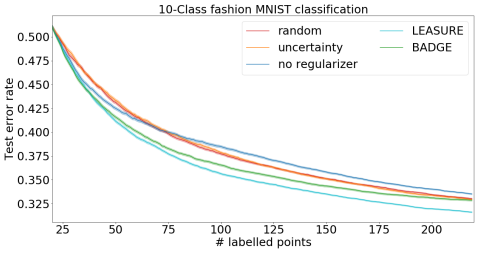
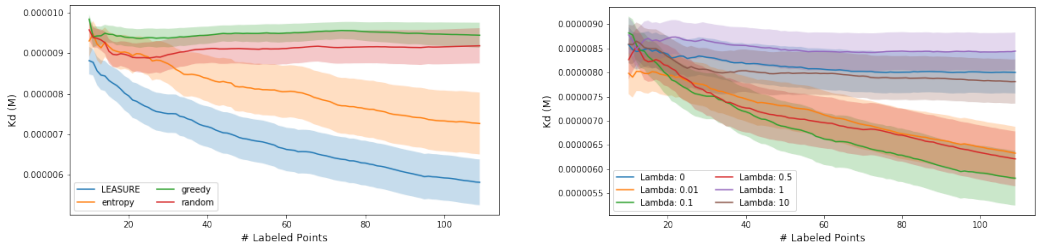


Figure 2: Combining submodular regularization with a learned active learning policy for 10-class Fashion-MNIST classification. The figure summarizes the classification error of a neural network trained on labelled images, as a function of the number of labelled images. Originally, random set of 20 images is selected, and then each policy greedily chooses the next image to label. The learned policies were trained on rollouts of length up to 30, and tested on rollouts of length 200. The "no regularizer" policy corresponds to Konyushkova et al. (2017), only in this case the features are parametrized by the neural network instead of being hand-engineered. "BADGE" corresponds to a sequential modification of (Ash et al., 2020). The results are averaged between 500 experiments, with standard error reported.

The above motivates the use of LEASURE in active learning (Figure 2). In this experiment, the set S is the Fashion-MNIST dataset consisting of greyscale images from one of 10 clothes classes (Xiao et al. (2017)). The goal was to learn a policy that greedily selects "the best" point $x^* \in S_u$ to label, such that a neural network classifier trained on the labelled set $S_l \cup \{x^*\}$ produces the most accurate classification of the unlabelled images. In particular, we trained the above function g to predict the



(a) Comparison to baseline methods

(b) Effect of scaling parameter lambda

Figure 3: Combining submodular regularization with a learned active learning policy for a protein engineering task. In (b), Lambda = 0 corresponds to the unregularized case. Error bars are plotted as standard error of the mean across 50 replicates.

accuracy gain g^{exp} from labelling a point. The accuracy gain g^{exp} was measured by training the neural network classifier on both S_l and $S_l \cup \{x\}$ and then recording the difference in validation set classification accuracy. Since obtaining exact g^{exp} for each datapoint is very expensive, we instead collected noisy labels $g_\epsilon^{\text{exp}} \approx g^{\text{exp}}$, obtained by training the classifier for only 10 epochs. The tuples $\{(S_l, x), g_\epsilon^{\text{exp}})\}$ were collected using DAgger with rollouts of length 30 (starting from a random batch of 20 images). For training, we used an initially unlabelled dataset with 60000 images, 2000 of which were set aside to use for evaluating validation accuracy. We trained two neural networks to approximate g - an unregularized one, and one with a monotonicity and a submodularity regularizer (i.e. LEASURE). See Appendix B for details on architecture and training procedure.

The trained policies were tested on a set of 8000 images, with additional 2000 set aside for validation. At test time, we again started with a random batch of size 20 and then used each policy to sequentially select additional 200 images to label (Figure 2). The recorded test error rate was collected using real g^{exp}, i.e. a classifier trained until training loss reaches a certain threshold. The experiment was benchmarked against the "random" policy that randomly picked the next point, the "uncertainty" policy that selected the next point by maximizing uncertainty, the "no regularizer" policy that used DAgger with MSE loss, and "BADGE" from Ash et al. (2020). See Appendix B for details. Even though LEASURE was trained on much shorter rollouts using very noisy labels, it still outperformed all other baselines. This confirms our intuition that the submodular regularizer allowed the learned score function g to find a local minima that generalizes well to out of sample.

6.3 PROTEIN ENGINEERING

By employing a large protein engineering database containing mutation-function data (Wang et al., 2019), we demonstrate that LEASURE enables the learning of an optimal policy for imitating expert design of protein sequences (see Appendix for detailed discussion of datasets). As in Liu et al. (2018) we construct a fully data-driven expert which evaluates via 1-step roll-out the effect of labeling each candidate data (in our case a protein mutant) with the objective of minimizing loss on a downstream regression task (predicting protein fitness).

When training the policy to emulate the algorithmic expert via imitation learning, we represent each state as two merged representations: (1) a fixed dimensional representation of the protein being considered (as the last dense layer of the network described in Appendix C), and (2) a similar fixed dimensional representation of the data already included in the training set (as a sum of their embeddings), including their average label value. At each step a random pool of data is drawn from the state space and the expert policy greedily selects a protein to label, which minimizes the expected regression loss on the downstream regression task (prediction of protein fitness). Once the complete pool of data has been evaluated, the states are stored along with their associated preference score, taken as their ability to reduce the loss in the 1-step roll out. Using these scores, the expert selects a protein sequence to add into the training set, and we retrain the model and use the updated model to predict a protein with the maximum fitness. This paired state action data is used to train the policy model at the end of each episode, as described in Liu et al. (2018). As we observe in Figure 3a, this method trains a policy which performs nearly identically to this 1-step oracle expert.

The use of submodular regularization enables the learning of a policy which generalizes to a fundamentally different protein engineering task. In our experiments, LEASURE is trained to emulate

a greedy oracle for maximizing the stability of protein G, a small bacterial protein used across a range of biotechnology applications (Sjbring et al., 1991). We evaluate our results by applying the trained policy to select data for the task of predicting antibody binding to a small molecule. As is the case with all protein fitness landscapes, the evaluation dataset is highly imbalanced, with the vast majority of mutants conferring no improvement at all. Because data is expensive to label in biological settings (proteins must be synthesized, purified and tested), we are often limited in how many labels can feasibly be generated, and the discriminative power among the best results is often more important than among the worst. To construct a metric with real-world applicability we assess each model by systemically examining the median Kd of the next ten data points selected at each budget, from 10 to 110 total labels. This method is utilized in recognisance of the extreme ruggedness of protein engineering landscapes, wherein the vast majority of labels are of null fitness, and the ability to select rare useful labels for the next experimental cycle is of key importance.

We observe that LEASURE outperforms all evaluated baselines, and that the inclusion of submodular optimization is mandatory to its success (Figure 3a). A greedy active learner which labels the antibody mutation with the best predicted Kd (the smallest) preforms approximately equivalently with selecting random labels. Use of dropout as an approximation of model uncertainty as in Gal and Ghahramani (2016) improves upon these baselines, although significant betterment is not achieved until approximately 35 labels are added. In comparison, the results from LEASURE diverge from all others nearly immediately, and the best model, which uses a lambda of 0.1, achieves a notable improvement in Kd, $5.81\,\mu$M, vs $7.27\,\mu$M achieved by entropy sampling. In support of methods success, we note that the learned policy preforms approximately as well as the greedy oracle which it emulates (Appendix Figure 7a). We observe that the results are robust within a range of possible lambda values (Figure Figure 3b and Appendix Figure 7b), and that without the use of submodular regularization the trained policy fails to learn a policy better than the selection of random labels. This is an important finding, as the method proposed by Liu et al. (2018) without LEASURE, has been shown to be a state-of-the-art method for imitation learning.

Based on these empirical results, LEASURE demonstrates significant potential as computational tool for *real-world automated experimental design tasks*: In particular, in the protein engineering task, LEASURE achieves the SOTA on the benchmark data-sets considered in this work. While LEASURE does involve repeated retraining of the protein engineering network, we observe that it returns strong results even with a single step of training. Additionally, the networks that are employed are very simple (Appendix C). This allows for reasonable training time (36 hours) and nearly instantaneous inference. Given the considerable time and cost of protein engineering, these computational budgets are quite modest. Protein engineering is a time consuming (months to years) and expensive undertaking (10's of thousands to millions of dollars). These projects usually strive to achieve the best possible results given a fixed budget. We have demonstrated in our work the ability deliver significant improvements in protein potency for the modest fixed budgets. Although the cost savings of engineering and testing an individual protein (or label) vary significantly based on the system, ranging tens to hundreds of dollars, we observe that to achieve a Kd of 8e-6 M LEASURE delivers an approximate cost savings of 65%, or 40 fewer labels than the next best method. The sequential synthesis and evaluation of each of these labels would likely span several months and additionally incur several thousands of dollars of materials costs.

7 CONCLUSION

In this paper, we introduce LEASURE, a data-driven decision making framework based on a novel submodular-regularized loss function. The algorithm was inspired by the recent developments of submodular-surrogate-based near-optimal algorithms for sequential decision making. We have demonstrated LEASURE on several diverse set of decision making tasks. Our results suggest that LEASURE can be easily integrated with modern deep imitation learning pipelines, and that it is efficient to run, while still reaching the best performance among the competing baselines. In addition to demonstrating the strong empirical performance on several use cases, we believe our work also provides useful insights in the design and analysis of novel information acquisition heuristics where traditional ad-hoc approaches are not feasible.

Acknowledgements. This research was supported in part by funding from NSF #1645832, NIH #T32GM112592, The Rosen Bioengineering Center, Raytheon, Beyond Limits, JPL, and UChicago CDAC via a JTFI AI + Science Grant. This work was additionally supported by NVIDIA corporation through the donation of the GPU hardware used in experiments.

REFERENCES

E. C. Alley, G. Khimulya, S. Biswas, M. AlQuraishi, and G. M. Church. 2019.

Jordan T. Ash, Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal. 2020. URL https://openreview.net/forum?id=ryghZJBKPS.

Francis Bach. Submodular functions: from discrete to continuous domains. *Mathematical Programming*, 175 (1-2):419–459, 2019.

Ashwinkumar Badanidiyuru, Baharan Mirzasoleiman, Amin Karbasi, and Andreas Krause. Streaming submodular maximization: Massive data summarization on the fly. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 671–680, 2014.

Maria-Florina Balcan and Nicholas JA Harvey. Submodular functions: Learnability, structure, and optimization. *SIAM Journal on Computing*, 47(3):703–754, 2018.

Mislav Balunovic, Pavol Bielik, and Martin T Vechev. Learning to solve smt formulas. In *NeurIPS*, pages 10338–10349, 2018.

Andrew An Bian, Joachim M Buhmann, Andreas Krause, and Sebastian Tschiatschek. Guarantees for greedy maximization of non-submodular functions with applications. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 498–507, 2017a.

Andrew An Bian, Baharan Mirzasoleiman, Joachim Buhmann, and Andreas Krause. Guaranteed non-convex optimization: Submodular maximization over continuous domains. In *Artificial Intelligence and Statistics*, pages 111–120, 2017b.

Niv Buchbinder, Moran Feldman, Joseph Naor, and Roy Schwartz. Submodular maximization with cardinality constraints. In *Proceedings of the twenty-fifth annual ACM-SIAM symposium on Discrete algorithms*, pages 1433–1452. SIAM, 2014.

K. Chaloner and I. Verdinelli. Bayesian experimental design: A review. *Statistical Science*, 10(3):273–304, 1995.

Yuxin Chen, S Hamed Hassani, Amin Karbasi, and Andreas Krause. Sequential information maximization: When is greedy near-optimal? In *Conference on Learning Theory*, pages 338–363, 2015a.

Yuxin Chen, Shervin Javdani, Amin Karbasi, James Andrew Bagnell, Siddhartha Srinivasa, and Andreas Krause. Submodular surrogates for value of information. In *Proc. Conference on Artificial Intelligence (AAAI)*, January 2015b.

Yuxin Chen, S. Hamed Hassani, and Andreas Krause. Near-optimal bayesian active learning with correlated and noisy tests. In *Proc. International Conference on Artificial Intelligence and Statistics (AISTATS)*, April 2017a.

Yuxin Chen, Jean-Michel Renders, Morteza Haghir Chehreghani, and Andreas Krause. Efficient online learning for optimizing value of information: Theory and application to interactive troubleshooting. In *Proceedings of the 33rd Conference on Uncertainty in Artificial Intelligence (UAI 2017)*, volume 2, pages 966–983. Curran Associates, Inc., 2017b.

Sonia Chernova and Andrea L Thomaz. Robot learning from human teachers. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 8(3):1–121, 2014.

Sanjiban Choudhury, Ashish Kapoor, Gireeja Ranade, and Debadeepta Dey. Learning to gather information via imitation. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 908–915. IEEE, 2017.

Brian W Dolhansky and Jeff A Bilmes. 2016.

Khalid El-Arini, Gaurav Veda, Dafna Shahaf, and Carlos Guestrin. Turning down the noise in the blogosphere. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 289–298, 2009.

Moran Feldman, Ashkan Norouzi-Fard, Ola Svensson, and Rico Zenklusen. The one-way communication complexity of submodular maximization with applications to streaming and robustness. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pages 1363–1374, 2020.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1050–1059, New York, New York, USA, 20–22 Jun 2016. PMLR. URL `http://proceedings.mlr.press/v48/gal16.html`.

Maxime Gasse, Didier Chételat, Nicola Ferroni, Laurent Charlin, and Andrea Lodi. Exact combinatorial optimization with graph convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 15580–15592, 2019.

Daniel Golovin and Andreas Krause. Adaptive submodularity: Theory and applications in active learning and stochastic optimization. *Journal of Artificial Intelligence Research*, 42:427–486, 2011.

He He, Hal Daume III, and Jason M Eisner. Learning to search in branch and bound algorithms. In *Advances in neural information processing systems*, pages 3293–3301, 2014.

Gaurush Hiranandani, Harvineet Singh, Prakhar Gupta, Iftikhar Ahamath Burhanuddin, Zheng Wen, and Branislav Kveton. Cascading linear submodular bandits: Accounting for position bias and diversity in online learning to rank. In *Uncertainty in Artificial Intelligence*, pages 722–732. PMLR, 2020.

Thibaut Horel and Yaron Singer. Maximization of approximately submodular functions. In *Advances in Neural Information Processing Systems*, pages 3045–3053, 2016.

Shervin Javdani, Yuxin Chen, Amin Karbasi, Andreas Krause, James Andrew Bagnell, and Siddhartha Srinivasa. Near-optimal bayesian active learning for decision making. In *In Proc. International Conference on Artificial Intelligence and Statistics (AISTATS)*, April 2014.

Ehsan Kazemi, Marko Mitrovic, Morteza Zadimoghaddam, Silvio Lattanzi, and Amin Karbasi. Submodular streaming in all its glory: Tight approximation, minimum memory and low adaptive complexity. In *International Conference on Machine Learning*, pages 3311–3320, 2019.

Elias Boutros Khalil, Pierre Le Bodic, Le Song, George Nemhauser, and Bistra Dilkina. Learning to branch in mixed integer programming. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

Ksenia Konyushkova, Raphael Sznitman, and Pascal Fua. Learning active learning from data. In *Advances in Neural Information Processing Systems*, pages 4225–4235, 2017.

Andreas Krause and Daniel Golovin. Submodular function maximization. *Tractability*, 3:71–104, 2014.

Andreas Krause and Carlos Guestrin. Optimal value of information in graphical models. *JAIR*, 35:557–591, 2009.

Andreas Krause, Ajit Singh, and Carlos Guestrin. Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies. *Journal of Machine Learning Research*, 9(2), 2008.

Jure Leskovec, Andreas Krause, Carlos Guestrin, Christos Faloutsos, Jeanne VanBriesen, and Natalie Glance. Cost-effective outbreak detection in networks. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 420–429, 2007.

Ching Lih Lim. A suite of greedy methods for set cover computation. 2015.

Hui Lin and Jeff Bilmes. A class of submodular functions for document summarization. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 510–520, 2011.

Hui Lin and Jeff Bilmes. Learning mixtures of submodular shells with application to document summarization. In *Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence*, UAI'12, page 479490, Arlington, Virginia, USA, 2012. AUAI Press. ISBN 9780974903989.

Dennis V Lindley. On a measure of the information provided by an experiment. *The Annals of Mathematical Statistics*, pages 986–1005, 1956.

Ming Liu, Wray Buntine, and Gholamreza Haffari. Learning how to actively learn: A deep imitation learning approach. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1874–1883, 2018.

Marko Mitrovic, Mark Bun, Andreas Krause, and Amin Karbasi. Differentially private submodular maximization: data summarization in disguise. In *International Conference on Machine Learning*, pages 2478–2487, 2017.

George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. An analysis of approximations for maximizing submodular set functionsi. *Mathematical programming*, 14(1):265–294, 1978.

Filip Radlinski, Robert Kleinberg, and Thorsten Joachims. Learning diverse rankings with multi-armed bandits. In *Proceedings of the 25th international conference on Machine learning*, pages 784–791, 2008.

Roshan Rao, Nicholas Bhattacharya, Neil Thomas, Yan Duan, Xi Chen, John Canny, Pieter Abbeel, and Yun S. Song. Evaluating protein transfer learning with tape. *Advances in neural information processing systems*, 32: 9689–9701, Dec 2019. ISSN 1049-5258. URL https://pubmed.ncbi.nlm.nih.gov/33390682. 33390682[pmid].

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. 2011.

Stephane Ross, Jiaji Zhou, Yisong Yue, Debadeepta Dey, and Drew Bagnell. Learning policies for contextual submodular prediction. In *International Conference on Machine Learning*, pages 1364–1372, 2013.

M. C. Runge, S. J. Converse, and J. E. Lyons. Which uncertainty? using expert elicitation and expected value of information to design an adaptive program. *Biological Conservation*, 2011.

Yash Satsangi, Shimon Whiteson, Frans A Oliehoek, and Matthijs TJ Spaan. Exploiting submodular value functions for scaling up active perception. *Autonomous Robots*, 42(2):209–233, 2018.

Adish Singla, Ilija Bogunovic, Gábor Bartók, Amin Karbasi, and Andreas Krause. Near-optimally teaching the crowd to classify. In *ICML*, volume 1, page 3, 2014.

Ruben Sipos, Pannaga Shivaswamy, and Thorsten Joachims. Large-margin learning of submodular summarization models. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 224–233, 2012.

U. Sjbring, L. Bjrck, and W. Kastern. Streptococcal protein G. Gene structure and protein binding properties. *J. Biol. Chem.*, 266(1):399–405, Jan 1991.

Jialin Song, Ravi Lanka, Albert Zhao, Aadyot Bhatnagar, Yisong Yue, and Masahiro Ono. Learning to search via retrospective imitation. *arXiv preprint arXiv:1804.00846*, 2018.

Jialin Song, Ravi Lanka, Yisong Yue, and Bistra Dilkina. A general large neighborhood search framework for solving integer linear programs. In *Advances in Neural Information Processing Systems*, 2020.

Matthew Streeter, Daniel Golovin, and Andreas Krause. Online learning of assignments. In Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems*, volume 22. Curran Associates, Inc., 2009. URL https://proceedings.neurips.cc/paper/2009/file/e0c641195b27425bb056ac56f8953d24-Paper.pdf.

C. Y. Wang, P. M. Chang, M. L. Ary, B. D. Allen, R. A. Chica, S. L. Mayo, and B. D. Olafson. ProtaBank: A repository for protein design and engineering data. *Protein Sci.*, 28(3):672, Mar 2019.

Kai Wei, Rishabh Iyer, and Jeff Bilmes. Submodularity in data subset selection and active learning. In *International Conference on Machine Learning*, pages 1954–1963, 2015.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017.

Baosheng Yu, Meng Fang, and Dacheng Tao. Linear submodular bandits with a knapsack constraint. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30, 2016.

Yisong Yue and Carlos Guestrin. Linear submodular bandits and their application to diversified retrieval. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc., 2011. URL https://proceedings.neurips.cc/paper/2011/file/33ebd5b07dc7e407752fe773eed20635-Paper.pdf.

Yisong Yue and Thorsten Joachims. Predicting diverse subsets using structural svms. In *Proceedings of the 25th international conference on Machine learning*, pages 1224–1231, 2008.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. Deep sets. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/f22e4747da1aa27e363d86d40ff442fe-Paper.pdf.

A PROOF FOR SECTION 5

A.1 PROOF OF THEOREM 2

Proof. The high-level idea is to first connect the total expected utility of the learned policy $\hat{\pi}$ with the expected utility of the expert policy $\pi^{\exp}$, following the analysis in DAgger (Ross et al., 2011). Then, we will use the fact that $\pi^{\exp}$ is greedy with respect to f, an approximation to the submodular utility function u, to bound the one step gain of the $\pi^{\exp}$ against the k step gain of running the optimal policy, and subsequently bound the total utility of the expert policy against the optimal policy. We would eventually obtain a similar result as Theorem 2, detailed as follows.

More concretely, following Theorem 3.4 in DAgger, we obtain that

$$\mathbb{E}[u(S_{\hat{\pi},k})] \geq \mathbb{E}[u(S_{\pi^{\exp},k})] - \Delta_{\max} k \hat{\epsilon}_N - O(1)$$

Here $\Delta_{\max}$ is the largest one-step deviation from $\pi^{\exp}$ that $\hat{\pi}$ can suffer. It is equivalent to the term u in the DAgger paper. Since f is ϵ-close to a monotone submodular function u, we know that $\Delta_{\max} \leq \max_{A \subset \mathcal{V}, |A|=k} f(A) \leq \max_{A \subset \mathcal{V}, |A|=k} u(A) + \epsilon_E$, which is a constant once u is given.

Next, since $\pi^{\exp}$ is greedily optimizing an ϵ_E-approximation to a monotone submodular function u, we know that

$$\mathbb{E}[u(S_{\pi^{\exp},k})] \geq (1 - 1/e)\mathbb{E}[u(S_{\pi^*,k})] - k\epsilon_E$$

following the proof from Theorem 5 in (Chen et al., 2017b).

Combining both steps, we have that

$$\mathbb{E}[u(S_{\hat{\pi},k})] \geq (1 - 1/e)\mathbb{E}[u(S_{\pi^*,k})] - k(\epsilon_E + \Delta_{\max}\hat{\epsilon}_N) - O(1)$$

which completes the proof.

□

B SUPPELEMENTAL DETAILS FOR THE SET COVER AND MNIST ACTIVE LEARNING EXPERIMENTS

We provide additional results for the set cover experiments, under the same experimental setup as Figure 1a and 1b. The subplots 4a and 4b show the mean square error of learned policy g as a function of the size of S_l. We provide a zoomed-in version of 4b in Figure 4c. In Figure 4c, we show it is clear that training the neural network on the monotonicity regularizer only does not help it learn out of sample - the error rapidly increases as soon as the test rollout length becomes larger than the training rollout length.

In Noisy Set Cover experiment (Figure 4a), each label of the element added to the superset was perturbed with $N(0, 1)$ noise. As a result, the variance of the total noise is linear in the number of sets. So, it is reasonable that the MSE error grows with number of sets - the policies cannot learn to predict random noise. While stochastic MSE of LEASURE and the no-regularizer policy are similar, LEASURE outperforms in the number of elements added, which is what matters in practice (Figure 1). These two figures confirm our intuition that when the problem is not exactly submodular, Leasure will still generalize better than no regularizer by learning to ignore small deviations from submodularity. Finally, it is also expected that DSF has a lower MSE than Leasure when the label noise is too large - Deep Submodular Functions are required to be submodular. When the stochasticity in the MSE becomes overwhelmingly large, that restrictive requirement becomes an advantage. However, when the MSE variance is not too large, the lack of expressiveness and the difficulty of optimization of DSF make it lose its advantage compared to Leasure.

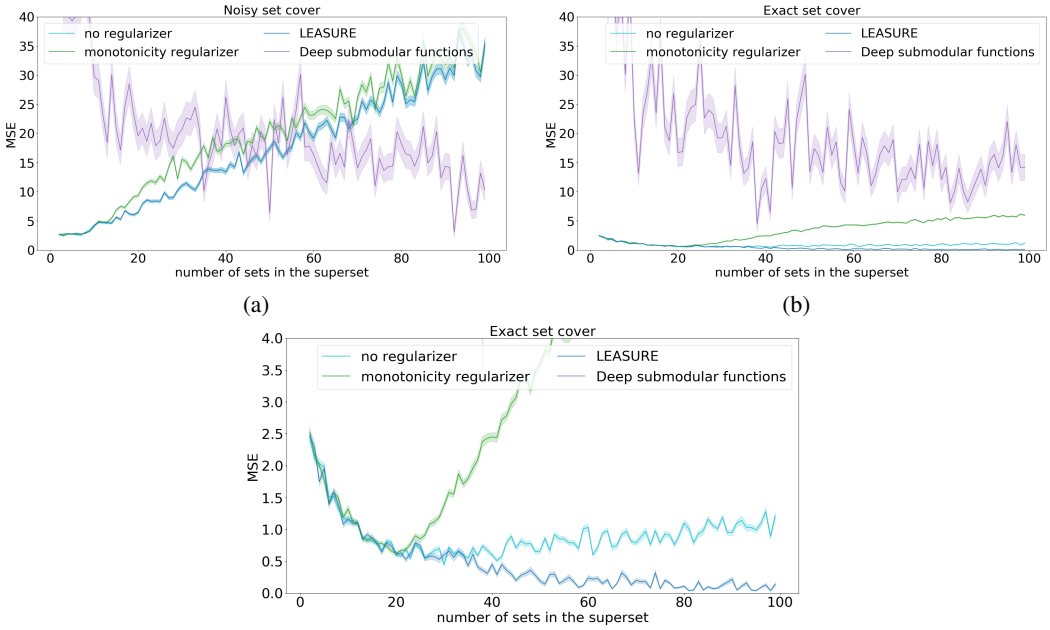


Figure 4: Supplemental results: Set cover

For completion, we also provide our architecture and parameter choices for both set cover and Learning Active Learning (LAL) on MNIST experiments. For set cover, the problem is too simple to require DAGGER (Ross et al., 2011). Instead, the tuples are generated randomly. For active learning on MNIST, the tuples are indeed generated using Algorithm 1. For MNIST, we first preprocessed our dataset with PCA, leaving the number of vectors necessary to achieve 80% covariance on the training set (24 vectors). That was necessary to allow the comparison with DSF. For set cover, each element was a set v containing 23 elements $v^1, v^2, .., v^{23}$, where v^i was an integer corresponding to the label of the species. As a neural network input, v was simply represented as a vector of $[v^1, ..., v^{23}]$.

Both set cover and MNIST used a modified Deepset architecture (Zaheer et al., 2017) for score networks as follows: Given a set $A = \{v_0, ..., v_k\} \subset V$ and a datapoint $v \in V$, the score network g first preprocesses all inputs $v_0, ..., v_k, v$ to obtain learned embeddings $\bar{v}_0, .., \bar{v}_k, \bar{v}$. (See Figure 5) Then, the elements in A are combined using Deepsets architecture to produce a learned set embedding $\bar{A}$. Finally, $\bar{A}$ and $\bar{v}$ are concatenated and then a learned linear layer and a Leaky ReLu nonlinearity are applied to produce $g(A, v)$. (See Figure 6). All dense layers have 64 neurons and a bias term. Using this Deepsets-like framework, we achieve permutation invariance of elements in set A while also keeping the network expressive enough to learn a wide range of functions.

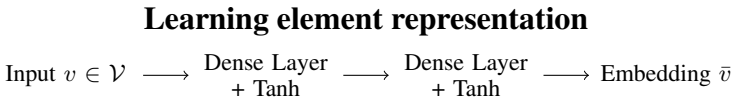


Figure 5: Score neural network architecture illustration

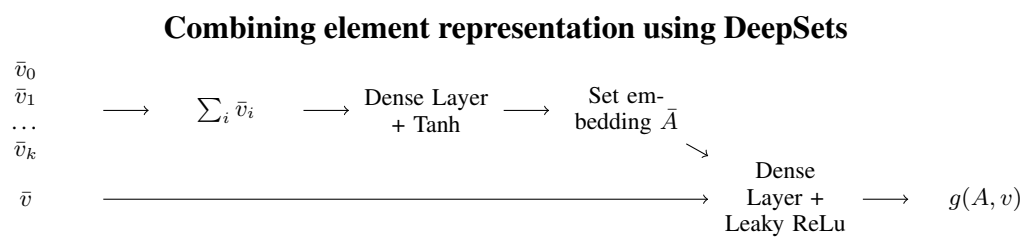


Figure 6: Score neural network architecture illustration.

For both tasks, the score networks are trained using ADAM with a learning rate of 1e-3. Beta parameter from Line 2 in LEASURE was picked randomly to be $\frac{4}{5}$. From experiments, the exact value of the parameter did not matter as long as it starts with at least $\frac{1}{2}$ and degrades towards almost 0 after N iterations. The λ and γ parameters were picked using a hyperparameter sweep in log space. As per our intuition, we have found that the strength of the parameters should reflect your certainty that the task is submodular and/or monotone. For set cover, $\lambda = 0.1, \gamma = 0.5$, while for active learning $\lambda = 0.001$ and $\gamma = 0.001$. Notice that the values are not comparable between different experiments: for MNIST Learning Active Learning (LAL), $g^{\text{exp}}(A, v) \in [0, 1)$ outputs the accuracy gain of adding v to A and training a supervised model on it; for set cover, $g^{\text{exp}}(A, v) \in \{0, 1\}$ outputs the number of new elements added to the set by adding x to A. For LAL, the values of g^{exp} are usually much smaller than 1, particularly for larger sets. Thus, the values for the two regularizers had to be smaller so that the model learns not just the regularizer.

Finally, we wanted to discuss our baselines in Fashion MNIST experiments. In Figure 2, we have four baselines: random, uncertainty, BADGE (Ash et al., 2020), and no regularizer. The no regularizer baseline was trained identically to LEASURE, except for the absense of submodularity and monotonicity regularizers. The no regularizer baseline performed well on the sets with up to 30 additional points - corresponding exactly to the length of the training rollouts. However, it failed to generalize. On the other hand, the submodular regularizer allowed the learned score function to find a local minima that generalized well to out of sample. Finally, BADGE did not seem to perform well when the number of datapoints in the set was large, likely because the gradient signal from adding any one additional datapoint was too weak and thus the selection of the next best datapoint was too noisy.

Some more details regarding BADGE (Ash et al., 2020). The authors do not learn a policy, instead, they use gradients of the classifier (gradient embedding) to select a useful, diverse batch. Although BADGE was originally made for a batch setting, the authors' main idea is still applicable to our case: they argued that the next datapoint(s) can be selected by looking at which fictitious labels would produce the largest gradients in the classifier network. Therefore, we replaced the kmeans++ algorithm the authors suggested with simply selecting the datapoint that corresponds to the largest gradient norm. This algorithm has an advantage that it does not require a trained policy network. However, it provides no guarantees about submodularity of the resulting policy, and, in our experiments, the performance degrades with the size of the set - likely because the gradient signal from

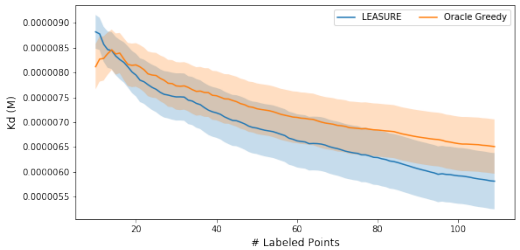
adding any one additional datapoint was too weak and thus the selection of the next best datapoint was too noisy. Since BADGE requires a neural network classifier/regressor, we could not use it as a baseline for Set Cover (Set Cover regression function is simply adding all elements in the superset).

The no-regularizer baseline is similar to that of Konyushkova et al. (2017). However, the problem considered in Konyushkova et al. (2017) is not compatible with most of the tasks we considered here (for MNIST, yes if we use random forest classifiers; but for others not). Furthermore, Konyushkova et al. (2017) treated the problem under a classical supervised learning setting this is often not desirable, given that we are learning a policy from non i.i.d. data samples.

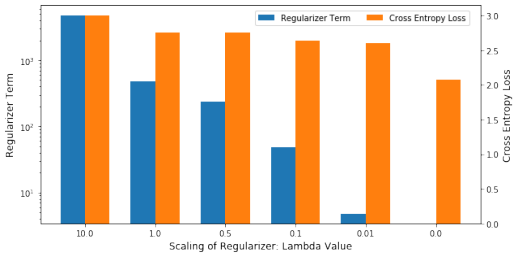
C SUPPLEMENTAL DETAILS FOR THE PROTEIN ENGINEERING EXPERIMENTS

Dataset Our datasets were identified in Protabank (Wang et al., 2019) for training of active learning policies and benchmarking of performance. In selecting datasets upon which to train our active learning models several factors were considered. As the state space of possible protein variants for typical engineering application is very large, size is our foremost criteria. Additionally it will be advantageous to use datasets which characterize mutations to all amino acids (as opposed to Alanine scans), and those which include epistatic interactions. We also desire to identify datasets which have a high quality, quantitative readout, such as calorimetry, fluorescence, or SPR data.

Protein Engineering Methods Embeddings of protein sequences were created using the TAPE repository (Rao et al., 2019) according to the UniRep system as first proposed in Alley et al. (2019). UniRep produces protein embeddings as a matrix of shape (length protein sequence, 1900), although we average together the embeddings only of positions being engineered to produce a consistent embedding of shape (1900,). We have implemented the active learning imitation learning algorithm proposed in Liu et al. (2018) to work with the protein embedding representations described above. Pseudocode for this method is presented in Algorithms 1 and 2 from the original work. As in Liu et al. (2018), our policy network consists of a single dense unit which acts sequentially on the pool of samples being considered to produce a preference score. Our downstream protein engineering network (which was used to compute the preference score of the expert policy) acts on the protein embeddings prepared using TAPE. The network consists of an attention layer, followed by a 1-dimensional convolution layer (128 filters, kernel size 3), before being flattened and applying two fully connected layers of 128 units each. When predicting protein fitness, dropout is applied with a probability of 0.5 and an additional dense layer is applied with one unit and linear activation. Both networks are trained using ADAM with a learning rate of 1e-3. The implementation of this part of the project is nearly identical to Liu et al. (2018), only changing the data representation, protein fitness network structure, and values of K (30), B (100) and T (20) as listed in the appendix of our work. Beta is fixed at 0.5, although the method was shown to be robust to a range of values. At training time, 100 labels are randomly selected for evaluating the effect of the greedy oracle, and 10 data are randomly selected to form the initial data set for learning. The superset is appended at each step of training the policy to maintain a size of 2x the labeled dataset. The training of a policy using these settings takes 36 hours on a modern multiprocessor computer equipped with an NVIDIA Titan V GPU.



(a) Comparison of policy to greedy oracle which it emulates



(b) Effect of scaling parameter lambda and empirical evidence for selecting its value

Figure 7: Supplemental results for the protein engineering experiments of Section 6.3: (a) We observe that the policy learned by LEASURE preforms approximately as well as the greedy oracle which it emulates. In this experiment the policy was derived from the training set, but the greedy oracle is operating on the test set. (b) Lambda linearly scales the value of the regularizer term. When lambda takes value 0.01, the magnitude of the (scaled) regularizer term (represented by the blue bar) aligns the best with the magnitude of the cross entropy loss (represented by the orange bar). This is consistent with what we observed in Figure 3b where $\lambda = 0.01$ leads to well-regularized model behavior.

1.6 Protein Docking with DNNs and CNN-F

Protein-protein interactions are the physical interaction of two or more proteins, often occurring in a cell. Such interactions are non-covalent, highly specific, and act to regulate or directly affect biochemical processes including metabolism, signaling, molecular chaperoning and translocation, and more. Protein-protein interactions are known to be involved in a range of diseases including cancer and immune disorders (Andreani et al., 2014). It is an established precept of molecular biology that structure dictates function, and accordingly the structures of protein-protein complexes are an important tool to understand and modulate the effect of these interactions. Because 3D structures of pairs of proteins are limited in the available literature and databases, simulating the structures of protein-protein complexes (known as protein-protein docking) has been a topic of academic interest since the 1970's (Wodak et al., 1978), and continues to be studied by researchers with a variety of backgrounds and interests. While significant progress has been made in the speed and accuracy of protein-protein docking, the combinatorial space of human proteins (the proteome) is vast–at least 30,000 proteins have been identified by researchers, with estimates of the total number of human proteins ranging from 30,000 to several billion (Smith et al., 2013). The use of deep generative models for protein-protein docking has the potential to further accelerate protein-protein docking, allowing for proteome-scale screening for new pairs of interactions, and the testing of newly discovered or designed molecules against the entirety of known protein structures. Interest in protein-protein docking has engaged

researchers from a variety of disciplines, with contests such as CAPRI (Lensink et al., 2016) serving as a unifying metric of accuracy and speed.

The dataset employed is a set of 700 antibody and protein antigen pairs, obtained from AbDb: The Antibody Structure Database (Ferdous et al., 2018). While the protein-protein interaction of antibody to antigen represents a special type of interaction, the geometric features and chemical aspects governing such interactions are the same as the broader set of protein-protein interactions. Using this set offers a number of ease-of-use advantages over picking structures from the Protein Data Bank, such as standardized residue numbering schemas and consistently formatted files. We have chosen to utilize a Siamese network architecture to first train models to predict which structures are capable of protein-protein interactions, without outputting a composite structure. Because the training set consists of structures of proteins in their interacting conformations, we utilize molecular dynamics to relax these proteins and achieve structures of the individual proteins which are closer to how they would exist unpaired. Using these relaxed protein forms, we will test various methods of accounting for conformational changes, as discussed in the previous section. This is not a direct replacement for protein-protein docking, but represents a critical and useful first step. Existing protein-protein docking software does not lend itself easily to proteome-level screening, as each pair of proteins to be screened for potential pairing requires 10+ minutes of CPU time (Huang et al. 2015). The proposed tool would run at GPU inference speeds, and alleviate the

need to screen unproductive pairs any further, whether that be by extant docking software or the following proposed tool.

To extend the discriminative tool described above, we further endeavor to train networks which take as inputs the coordinates of putative pairing partners, and directly output the coordinates of the resulting complex. We begin using an autoencoder-like network, and investigate the effects of a range of structured intermediate outputs. If necessary, additional training data can be cheaply obtained by sampling from the molecular dynamics trajectories obtained in the process of relaxing the training set, or by drawing from the PDB. Further elaboration on the subgoals of the project is provided in the following sections.

Addressing Issues of Output Sparsity

Mode collapse is a problem when applying generative networks to highly sparse voxel representations of protein structures. We have experimented with the use of blurring techniques which reduce sparsity (Kuzminykh et al., 2018), and loss functions such as Tversky loss (Salehi et al., 2017) which are suited to highly imbalanced labels.

The rendering of a composite structure with multiple modes, such as a protein-protein interaction depends upon information contained within both the individual components and their unified surface geometry and electrostatics. The generation of such complexes stands to benefit from expressively modeling dependencies, such as the angle of interaction, or the residues which contact in the assembled complex. By modeling such dependencies and including them

with the learned latent space of our models, we will be able to increase the expressive power of the generative network, while simultaneously decreasing the amount of information the latent space must encode. Rather than train a model to perform protein-protein docking end-to-end, it is possible that including structured intermediates and training a cascade of models will improve overall performance.

Molecular Dynamics to Account for Conformational Effects

Conformational changes upon binding of partner proteins are a leading cause of failures in rigid docking methods. Potential methods to account for such flexibility using probabilistic models might include naively training end-to-end models using the undocked conformations, applying rigid docking techniques to ensembles of conformers generated by molecular dynamics or rotamer sampling methods (Dunbrack et al., 2002). In order to allow such inquiry, we will perform a large scale molecular dynamics run on the entirety of our training corpus.

At its crux, protein-protein docking involves determining the appropriate rotation and translation of a pair or set of protein structures to determine a composite structure from structures of the individual components. Most existing protein-protein docking techniques accomplish this in two steps: 1) the proposal of potential protein-protein complex structures and 2) the scoring and optional refinement of the putative complexes generated in step 1. The proposal step of this process has been shown to be efficiently accomplished by fast Fourier transformation and geometric hashing techniques (Park et al., 2015). An implicit assumption of the aforementioned techniques is that the conformations of the

docked components diverge minimally from those of the individual components, as these methods treat the components as rigid. While this assumption is known to be false in many instances, rigid docking techniques have scored among the top results in recent CAPRI contests (Lensink et al., 2013). It is important to note that the results of the proposal step are not informative of whether the attempted docking is feasible in a biological system. During the scoring step, physics-based or empirically derived scoring functions are employed to measure sterics, electrostatics, hydrogen-bonding, and other relevant properties of the proposed complexes. This process ranks the proposed complexes from step 1, and provides metrics which a researcher may interpret to determine if the attempted interaction might truly occur in-vivo.

Despite the continued success of protein-protein docking techniques which treat the input structures as rigid, it has been noted that most failures to produce an accurate docking are due in part to protein conformational changes upon protein-protein interaction (Kaczor et al 2013)., Innovations for dealing with such flexibility can involve the application of rigid techniques to ensembles of conformations, or more explicitly modeling the flexibility of protein side-chains and backbones (Zacharias et al., 2010).

While generative models have been applied extensively to sequence-based biological representations (including the prediction of protein-protein interactions as in (Wang et al., 2018)), limited work has been undertaken with structural representations, particularly in the area of protein-protein docking. Previous docking-related works utilizing probabilistic models have focused on the docking

of small molecules to proteins. Most such examples apply representation learning to conformations resulting from an empirically-derived or physics-based docking program, adding a neural network to re-score the results (Pereira et al., 2016; Stepniewska-Dziubinska et al., 2018). Other works have sought to use the learned latent space of autoencoders to generate novel molecules (Gómez-Bombarelli et al., 2018), but cannot generate three-dimensional structures, and rely heavily on rule based refinement schemes. The most active area of structure-based machine learning has been concerned with the task of predicting protein structures from sequences. This is a problem of particular significance and difficulty, and deserves discussion.

State-of-the-art approaches to predicting protein structures from sequences most often focus on new innovations in generating residue-level contact-maps, such as the application of co-evolutionary data from large databases of sequences (most of which lack structures). Contact maps represent a protein as a square matrix, with each residue (amino-acid) represented along both the X and Y axes in the order of their sequence. This matrix is populated to indicate which residues are close to one-another in 3D space, either as a step function (e.g. the residues are < 8.0 angstroms apart), or as a continuous distance measurement. A key innovation of the recent AlphaFold technique introduced by DeepMind is the prediction of distributions over distances in a contact map. Once generated, these contact maps generally serve as restraints for off-the-shelf folding engines which formulate the problem as recovering the gram matrix of the coordinates, though some end-to-end coordinate-generating

pipelines have been published (Ingraham et al., 2019). Notably, in nearly all modern techniques the outputs must be of fixed-size, stemming from the use of convolutional layers to output the final matrix. This necessitates the cropping of proteins into subunits which are often not natural, and introduces an additional re-assembly step to the overall process.

Many prior works have demonstrated the ability of deep learning models to approximate complex many-to-one mappings. However the converse problem, learning to make diverse predictions from a simple but structured input, is less well understood.

This can be thought of as a one-to-many mapping which explicitly employs expressive dependencies. The use of Gaussian latent variables as explicit intermediates in conditional generative models has been shown to aid in inference tasks when the output space has multiple modes (Kihyuk et al., 2015), as is the case for predicting structures of mutable objects such as protein pairs. Sohn et al., 2015 demonstrate the use of a modified variational autoencoder structure, wherein the output is generated from the distribution of $p_\theta(y \mid x, z)$, where z is additional conditioning information injected at both the encoding and decoding phases of learning to increase the expressiveness of the latent space and the resulting output space. Many works have built upon this idea with varying forms of conditioning, such as Xu et al. (2018) who demonstrated the conditioning of a generative model upon a learned grammatical model to generate highly realistic and spatially correct images of objects, even

demonstrating the ability to apply transformations such as color swaps and object rearrangements.

In modeling physical objects with voxel based representations, binary voxel grids are commonly employed as indicating the occupancy of discretized positions. Sparsity has been a common problem for such representations, and a number of solutions have been put forth to address mode collapse resulting thusly, including re-scaling input values (Brock et al., 2016). Because atoms are highly localized and linked by bonds which are not easily encoded in voxel-space, we have observed in our previous work that extreme sparsity is common in such representations of biological structures, often with less than 0.1% of voxels populated. This has been approached by applying smoothing techniques such as Gaussian blurring or more complex wave transformations (Kuzminykh et al., 2018).

Experiments and Results

As a matter of practicality, we divide our docking pipeline into two phases: 1) identification of likely docking pairs, and 2) the generation of putative structural complexes. This division allows each network to encode information relevant to a particular task, and as a result of the complexity differences of the two models, will accelerate the overall application of the tool. Because we expect in most circumstances that a tested pair of structures will not interact (the tight control of protein-protein interactions is critical for their biological role as signaling molecules), running a generative 3D model for every pair would be a waste of

inference time. Every potential pair of molecules that we evaluate is initially evaluated by a binary classification network, which predicts whether or not the pair of structures is an interacting pair. We note that on average our generative structural models contain ~ 10x the parameters of the binary models we use to assess pairwise compatibility.

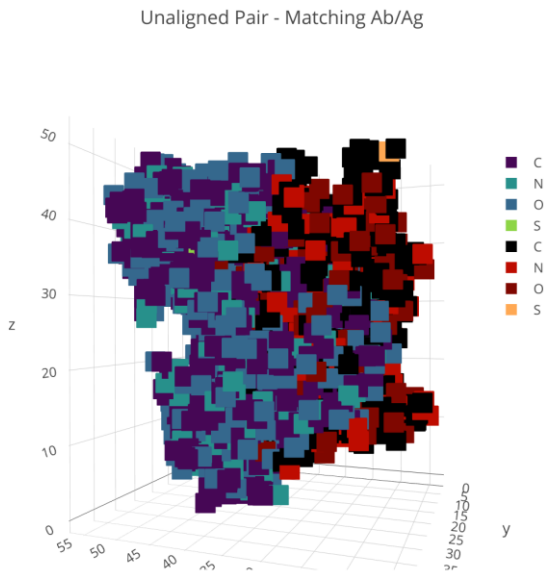


Figure 4: Sample input for phase 1 of docking pipeline. Output is binary decision of whether a pair of compounds is capable of forming productive pairing.

This initial binary classification network accepts pairs of structures as potential halves of a binding/interacting pair of proteins. So as to not bias the network, each structure is rotated randomly before being centered in the voxel grid, as illustrated in Figure 4. After extensive architecture searching, we settled upon a network structure inspired by Siamese networks (Koch et al., 2018). In

such networks, pairs of input structures are evaluated by a typical architecture of convolutional layers interspersed with max pooling layers, however two inputs are evaluated simultaneously and separately, only being combined at the end of the convolutional stack. Such an architecture is commonly depicted as having two parallel branches, with the weights forced to be identical between the two branches. A contrastive loss function was employed, which optimizes the Euclidean distance among points in the output space, specifically to minimize the distance between pairs of points with a positive label (subject to some predefined margin of sameness), while maximizing the distance between points bearing negative labels.

Upon hyperparameter tuning, this network was able to achieve an AUC-ROC of 0.95. We opted for this metric over accuracy, as we varied the ratio of positive pairs to negative pairs. Our network architecture comprises voxels grids of size 32x32x32 x 7 channels, where the seven channels correspond to the six most common atom types (C, H, O, N, P, S), with the seventh channel collectively denoting all other. Tools for parsing the most common molecule file types (PDB and SDF) were drawn from VoxLearn. Notable features of these parsers include the ability to alter the voxel sizes, number of voxels, apply data augmentation via rotations, and include Gaussian blurring or a null/empty channel (as will be discussed in the next section). After the input layer, the network consists of four 3D-convolutional layers, using a cubic kernel of size 4, with a rectified linear unit layer applied after each. Each of the first three convolutions is followed by a max pooling operation of size 2, and the final layer

is connected to a dense layer comprised of 4096 units with sigmoid activation. As is typical in Siamese networks, the Euclidean distance is calculated between the vectors resulting from two inputs, and the network is trained by applying the contrastive loss function. Using this schema, we achieve the reported AUC-ROC after 100 epochs of training, which takes approximately 48 hours on a single NVIDIA Tesla V100 GPU. We note that the AUC-ROC is reported as assessed on a withheld partition of 25% of the set, and has not plateaued after 48 hours, although training was terminated to conserve resources.

To accelerate learning by the network, a subsampling of ten examples of non-docking pairs were utilized for every positive docking pair-we observed similar AUC-ROCs over a range of ratios from 1:1 to 10:1. The final networks tested utilized voxels of size 4 angstroms, a substantial increase from the first networks we tested which utilized voxels of size 1 angstrom. While the choice of 1 angstrom was rooted in the length of an alkane bond, so as to disallow the colocalization of two atoms in the same voxel, increasing the size of voxels to 4 angstroms was observed to have negligible impact on AUC-ROC, but substantially decreased network sizes and training times.

After applying the binary classification network described above, one is left with a putative list of pairs of structures which productively interact to form bound complexes. To produce the desired output, a PDB file of the resulting complex, we introduce a second network which accepts as input the structures of each of the two components of the pair. A number of network architectures were tested, and it was quickly observed that each network would take multiple days to

evaluate, even with the use of techniques such as learning rate schedulers and early stopping. We note that this is in line with other works using voxelized representations of volumetric data (Wallach et al., 2015). To accelerate the process of architecture search, a toy problem was constructed to make use of much simpler data. Casting the problem to 2D, we considered the analogous problem of fitting together jigsaw puzzle pieces based only on the geometry of their edges. While previous works have described techniques for the arrangement of image fragments as uniform tiles (Noroozi et al., 2016), to the best of our knowledge, this was a novel problem. Specifically, we sought not only to align input shapes based on edge complementarity, but to also directly output an image of the aligned pair. The dataset for this problem was constructed using the python image library PIL. Specifically, irregular polygons were generated by sampling points on a circle around a center point, and adding random noise by varying the angular spacing and radial distance. Once a polygon was generated, a mask was applied to split the polygon roughly into two halves, with each half being saved as a separate 28x28 array of floats. Under such a scheme, we sought network architectures capable of predicting the rotation necessary to realign the two halves after a perturbation was applied. Data augmentation was performed in real time, with each half of the image being randomly rotated before learning and evaluation. This toy problem accelerated learning dramatically allowing us to now evaluate networks in minutes rather than days. After evaluating a number of potential architectures, we arrived at a network which first

concatenated the two input arrays along a new axis, and subjected the combined representation to a U-net like architecture (Ronneberger et al., 2015).

Specifically, the images were downsampled by blocks of stacked convolutions, interspersed with max pooling operations and batch normalization. After downsampling, the images were upsampled by applying upsampling operations, wherein the input tensors were scaled by a factor of two before applying further padded convolution operations, again using batch normalization. We observed that maintaining the minimum size of the representation to 7x7 before beginning upsampling improved the performance of this network, suggesting a lower bound to the necessary complexity of the latent space. Through the use of a sigmoid activation function, the network was able to output two channel images corresponding to the aligned input components, as shown in Figure 5. Upon observing some loss of fidelity at the edges of the reconstructed shapes, we decided to apply a GAN to further refine our output images. This proved to be a difficult task, requiring the careful tuning of the combined loss function from the GAN discriminator network and the RMSD loss assessed from reconstructing the images. Figure NNN demonstrates the final results from the best combination observed, wherein the GAN loss contributed 0.1% to the overall loss.

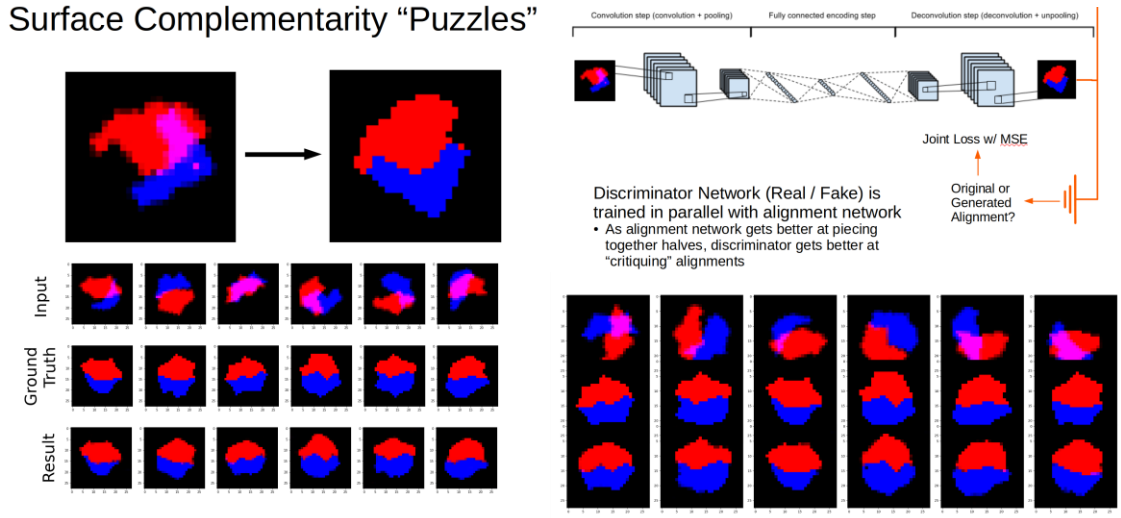


Figure 5: Creation of a toy problem to aid in tuning shape complementarity networks. (L) Casting the problem into a 2D toy problem. (R) Using a discriminator network and GAN training refines results.

Having found a suitable network architecture for learning the surface complementarity of 2D shapes, we adapted the best networks from our toy problem to work on voxelized representations of PDB files. After extensive testing the networks failed to routinely produce good results, often collapsing to an empty output. It was observed that the voxel representations of both the inputs and the ground truth results are ~99.9% empty. This is a consequence both of the one-hot encoding scheme used, and of the practice of only encoding atoms as points, rather than dispersed volumes. This issue was persistent across a wide range of tested conditions, including networks architectures, hyperparameters, and varying levels of data augmentation, and motivated us to undertake a treatment of the issue of sparsity in voxel representations.

Addressing Issues of Sparsity

To address sparsity-driven collapse in our generative models, we experimented with the use of loss functions which can be weighted with respect to precision and recall. As before, we constructed a simpler test system on which to develop and test our work. This system consisted of a variational autoencoder trained upon small molecule structures obtained from ChEMBL (Gaulton et al., 2017). Benefits of this system included smaller networks, and the availability of far more data upon which to train. Our best results were achieved with the Tversky loss function (Salehi et al., 2017), which is a weighted modification of the Dice similarity coefficient, typically written as depicted in Figure 6. Prior works have noted that adjusting the hyperparameters of this loss function can shift emphasis to rare labels, such as populated voxels in our networks. We further extended this loss function by including the capability for channel-wise weighting, first using both a simple frequency-based weighting and learned scalings thereof.

$$T(\alpha, \beta) = \frac{\sum_{i=1}^{N} p_{0i} g_{0i}}{\sum_{i=1}^{N} p_{0i} g_{0i} + \alpha \sum_{i=1}^{N} p_{0i} g_{1i} + \beta \sum_{i=1}^{N} p_{1i} g_{0i}}$$

Figure 6: Tversky Loss Function

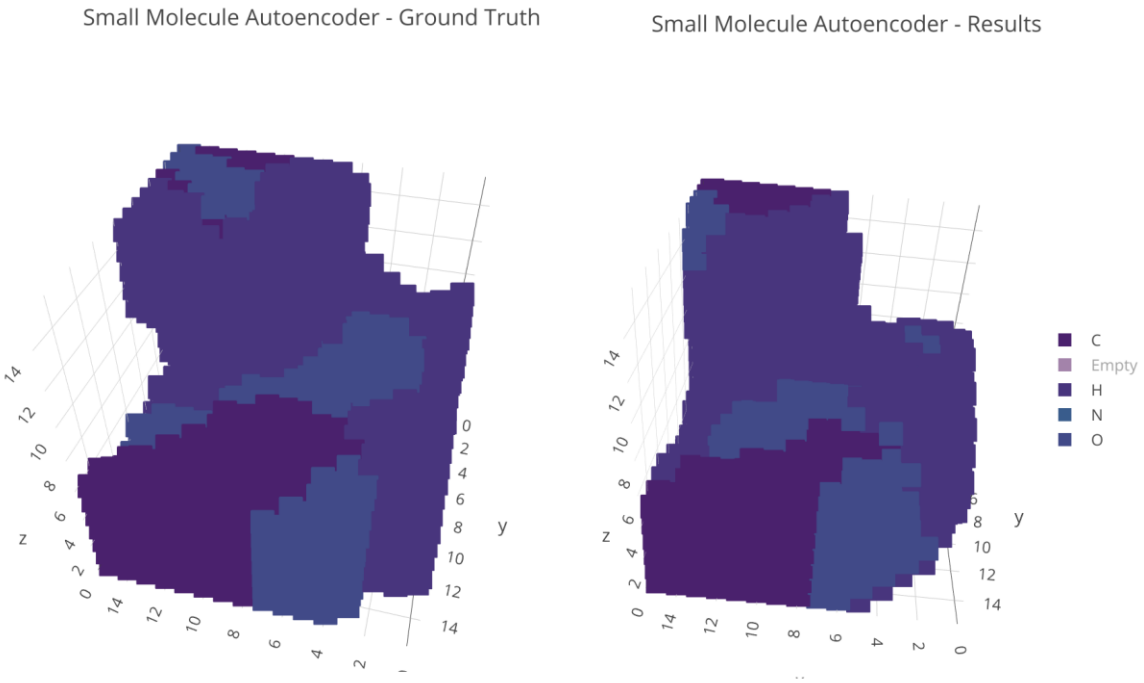


Figure 7: Autoencoder applied to small molecules showing Tversky loss and Gaussian blurring allows productive reproduction.

Additional performance benefits were obtained by including a channel explicitly demarcating empty voxels and using a softmax activation. This design was specifically introduced after noticing that our networks would commonly converge to outputting the per-channel (atom type) average uniformly across all voxels.

Another way to address sparsity is to directly reduce it by altering the design of the data representations. We achieved this several ways with varying levels of effect. Firstly, by increasing the size of the voxels, sparsity is reduced as each position is more likely to be occupied. This has the additional benefit of reducing the number of voxels needed to represent structures of a given size,

which decreases the size of our networks, but also reduces the fidelity of the representations by creating positions containing multiple atoms. Extending the concept of reducing sparsity by altering the data representations, we experimented with the effect of combining all channels as a sum (with and without clipping to 0/1). This effectively reduces the output spartity by a factor of 7, but similarly to increasing the voxel sizes, leads to a loss of fidelity in the reconstructed outputs.

The final technique we implemented to address sparsity in our voxelized representations was Gaussian blurring. Inspiration for this form of augmentation comes from the knowledge that atoms are not point masses, but rather clouds of electrons occupying a radius dictated by the atoms charge, hybridization, and chemical identity. Using this, we applied Gaussian blurring by setting sigma in accordance with the van der Waals radius of the atom type that each channel represented. This alteration reduced sparsity significantly, both in the input representations, and in the generated output voxels. Further experimentation revealed that applying a consistent sigma value for the Gaussian filter led to comparable benefit, despite lacking the neat chemical rationale of atomic radius-based scaling.

While each of the aforementioned techniques lead to qualitative increases in the stability of the networks we trained, collapse still occurred after a short period of training, typically 10 to 100 epochs, dependent on the learning rate. Such collapses occurred both with the use of GAN-based loss functions (wherein the generator loss was evaluated as the failure rate of a linked network

discriminating between its output and ground truth voxelized representations), Tversky reconstruction loss, and combinations of the two. Although all networks collapsed before producing high quality representations of the input molecules, two important observations were made which motivated our subsequent work. Firstly, it was observed that our networks could learn rotational equivalence, given sufficient data augmentation. This is an important result, as the definition of axes in a given molecule should not contribute to its identification or reconstruction. Secondly, we observed that our networks could construct reasonable representations of complexes of two proteins, given that the proteins were pre-oriented relative to each other in the inputs. This was initially received with disappointment, as orienting the two proteins relative to each other is a significant part of the problem we set out to solve. However it was soon realized that this observation was informative about what our networks were currently capable of encoding and expressing, molecular geometries relative to the input conformations. As the networks were not learning the necessary transformations to permute the inputs from random poses to the correct poses of the interacting complex, we sought next to augment our approach by learning a discrete intermediate which we could use to condition the generative network, namely the rotation and translation to apply to the randomized input poses.

Learning of Intermediate Representations

Because we observed that our generative networks were capable of reconstructing their inputs as a unified complex when the components were

supplied in the appropriate relative orientations, we believe that encoding this information is a bottleneck in the learning process, and an apt choice of an intermediate to learn. Our initial efforts towards this problem were concerned with the learning of a suitable translation and rotation to position one half of a protein-protein docking pair correctly relative to the other. Given the high level of redundancy in rotation matrices and the issue of gimbal lock in Euler angles, we initially sought to learn a quaternion representation of rotation in addition to the transformation in Cartesian space. Inspiration was drawn from the research area of camera relocalization, which seeks to determine the 6-degree-of-freedom camera position from either single or pairs of images. Generally these networks predict a position and camera angle relative to some recognized reference landmark, often using quaternions (Kendall et al., 2015, Kendall et al., 2017, Xiang et al., 2017). In these papers, it is observed that simultaneously learning both a translation (in the X/Y/Z planes) and a rotation to apply affords better results than applying either one alone, owing to the interdependence of the two factors. In an attempt to train networks capable of predicting the necessary translation and rotation to orient two proteins as they would appear in a protein-protein interaction, we began with creating utilities to randomly rotate and shift pairs of proteins away from their positions in known protein-protein interactions, calculating training labels on-the-fly relative to their starting positions. Despite experimenting with a variety of network structures, label encodings, and loss functions, we were unable to obtain network convergence.

To dissect the issues plaguing our learning, we sought to learn the translation and rotation components separately. With some effort, we succeeded in training networks that could predict the translational offset to position proteins relative to one another, but still could not learn to predict the necessary rotation to apply. Turning to the literature, we discovered that this is a relatively immature area of research (a good overview of the current state of the field is presented in Worrall et al., 2018). In an attempt to explore firsthand the process of learning to rotate voxel structures, we shifted our focus to training networks to align structures with themselves, given two copies of an identical structure, one of which is randomly perturbed. We approached this using a variety of techniques, including curriculum learning (increasing the complexity or degree of the perturbations as learning progressed) and varying representations of the applied rotations. Attempts were made to learn the rotation quaternion as a vector, as separate output tasks accomplished by discrete sub-networks, and as a combination of sign classification and absolute values as in Liao et al. (2019). Ultimately, we were able to succeed in learning self alignment as a vector of length three, representing Euler angles expressed as the cosine of the rotation in radians. The convergence of such networks were observed after employing curriculum training and several days of training, and ultimately were robust to evaluation with a holdout set.

Flexibility Dataset

Owing to their origin as x-ray crystal structures, the training data obtained from AbDb consists of paired structures of proteins in their interacting conformations. As we have discussed previously, evaluating models upon these data is a somewhat contrived exercise; in a real-world application, structures of putative binding pairs would not be in such pair-dependent, induced conformations. To relax these proteins and achieve structures of the individual proteins as they might exist unpaired, we sought to use molecular dynamics simulations. Despite decades of progress in the development of molecular mechanics force fields, libraries, and techniques, molecular dynamics calculations remain tricky to set up and run, requiring a cascade of preparation and post-processing steps for every molecule to be studied. Systems to be studied must be standardized to adhere to very strict formatting protocols before being patched to add any atoms which are missing (a common occurrence when solving electron density maps to determine X-ray crystal structures). After patching, special chemical moieties are added to cap the end of protein chains, and a periodic cell is constructed of the protein structure padded with a 12 angstrom margin of water molecules and pH balancing ions. Topology files describing the position of all atoms, bond lengths, dihedral angles, and a number of other constraints are prepared to correspond to a particular forcefield, in this case the Amber ff14SB force field (Maier et al., 2015). The system is then subject to an equilibration simulation to reduce the artifacts introduced in the process of crystallization and virtual system building. If this phase completes successfully, a

production run may then be undertaken to study the dynamics of the system over time. Because molecular dynamics is computationally very expensive, we opted to utilize the GPU accelerated library ACEMD (Harvey et al., 2009) and AWS. This required a non-trivial amount of engineering, as the academic version of ACEMD does not support running on machines with more than one GPU, and our initial approach of using an AWS Elastic Container Service Cluster did not support the most price efficient instance type (g2.2xlarge, per our internal benchmarking of ACEMD). To avert these limitations, we constructed a purpose-built Amazon Machine Image containing all 1687 systems to be simulated, as well as bash scripts which could be invoked using the launch time user data AWS CLI command. From a local machine, we coordinated the parsing out and monitoring of simulations: for each, a VM was spun up, equilibration and production simulations were run, and the resulting data was written to an S3 bucket. Each of the systems was simulated for 50 ns. In total 12,368 hours of GPU time were completed in two days, at an approximate cost of $2600. Substantial savings were brought about by using the optimal GPU instance type and preemptible spot instances. We calculated that it would have taken over 100 years to run these simulations on the Mayo lab’s CPU cluster (the GPU library is that good!)

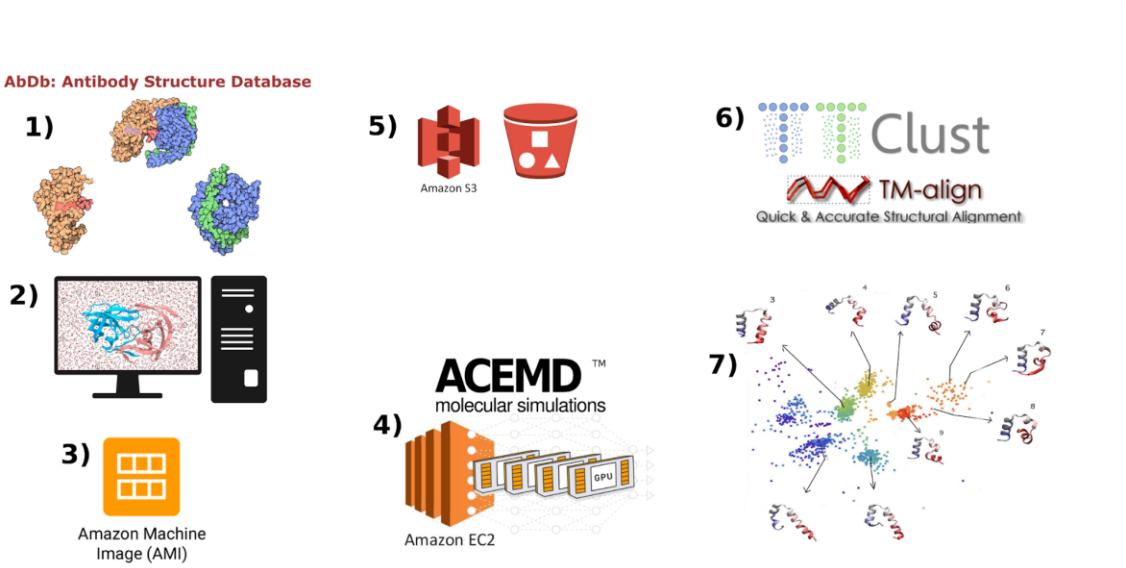


Figure 8 - Workflow for generation of flexibility dataset: 1) Structures are split into their component halves. 2) Structures are built on a local machine, including patching, standardization, addition of water molecules and ions. 3) Amazon Machine Image is created containing all prepared structures and scripts for equilibration and production MD runs. 4) Using launch time user data, systems are farmed out to cheapest possible AWS spot instances for simulation. 5) Results are written to S3 bucket as jobs complete. 6) Post processing is carried out locally: coordinates of each resulting trajectory are clustered and aligned to starting structures to determine divergence. 7) New PDB files are written for testing and training of docking networks.

For the purposes of this project, the desired output of the molecular dynamics simulations were the most likely structures of the unpaired proteins, and a set of alternative conformations for each of the systems (as a form of data augmentation). To achieve both of these outputs, the TTClust package was used to cluster the poses sampled across the dynamics trajectories on the basis of RMSD, with the optimal number of clusters being determined by the elbow

method of explained variance (Thorndike et al., 1953). The TMAlign algorithm (Zhang, 2005) was then used to align each cluster exemplar to the structure of the protein before MD, to determine how much each deviated from the structure of the protein as a half of the protein-protein docked pair. Using this information, along with the calculated occupancy of each cluster, we can infer which clusters are the most likely unbound configurations of the components of each pair, as well as guide the mining of more difficult tests for the previously described binary classification system.

Many setbacks were encountered in the process of carrying out the described work. Despite this, many lessons have been gained by recasting components of the problem as simpler toy problems, and by finding useful intermediate outputs that the models could solve. We believe that the highest yield problem yet to solve is addressing the sparsity of voxel grids, such as applying wave function inspired blurring, and learning the optimal rescaling of our currently binary occupancy labels.

References

Andreani, J. et al. “Evolution of protein interactions: From interactomes to interfaces”. Archives of Biochemistry and Biophysics. 554, 65–75 (2014).

Badrinarayanan, Vijay, et al. “SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation.” arXiv:1511.00561, (2016). arXiv.org, http://arxiv.org/abs/1511.00561.

Berman, H. M. “The Protein Data Bank”. Nucleic Acids Research. 28, 235–242 (2000).

Brock, A. et al., “Generative and Discriminative Voxel Modeling with Convolutional Neural Networks”, arXiv:1608.04236. (2016).

Chen, Yunpeng, et al. “Dual Path Networks.” arXiv:1707.01629 (2017). arXiv.org, http://arxiv.org/abs/1707.01629.

Chica, R. A., et al. “Generation of Longer Emission Wavelength Red Fluorescent Proteins Using Computationally Designed Libraries.” Proceedings of the National Academy of Sciences, 107(47), 20257–20262 (2010). doi:10.1073/pnas.1013910107.

CxCalc, MarvinBeans 18.29, ChemAxon https://www.chemaxon.com.

Dunbrack, R. L. Jr. “Rotamer Libraries in the 21st Century”. Current Opinion in Structural Biology. 12, 431–440 (2002).

Ferdous, S. et al. “AbDb: antibody structure database—a database of PDB-derived antibody structures.” Database. (2018), doi:10.1093/database/bay040.

Gaulton, A et al. “The ChEMBL database in 2007” Nucleic Acids Research. 45, D945–D954 (2016).

Gómez-Bombarelli, R. et al., “Automatic Chemical Design Using a Data-Driven Continuous Representation of Molecules”. ACS Central Science. 4, 268–276 (2018).

Han, Jing, et al. “Design, Synthesis, and Biological Activity of Novel Dicoumarol Glucagon-like Peptide 1 Conjugates.” Journal of Medicinal Chemistry, 56(24), 9955–9968 (2013)., doi:10.1021/jm4017448.

Harvey, M. J. et al. “ACEMD: Accelerating Biomolecular Dynamics in the Microsecond Time Scale”. Journal of Chemical Theory and Computation. 5, 1632–1639 (2009).

He, Kaiming, et al. “Deep Residual Learning for Image Recognition.” arXiv:1512.03385 (2015). http://arxiv.org/abs/1512.03385.

Huang, S.Y. “Exploring the potential of global protein–protein docking: an overview and critical assessment of current programs for automatic ab initio docking”. Drug Discovery

Today. 20, 969–977 (2015).

Ingraham, J. et al. “Learning Protein Structure with a Differentiable Simulator”. Openreview.net (2019), https://openreview.net/forum?id=Byg3y3C9Km.

Kendall, A., et al. “PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization” IEEE International Conference on Computer Vision (2015). http://dx.doi.org/10.1109/ICCV.2015.336.

Kendall, A. “Geometric Loss Functions for Camera Pose Regression with Deep Learning”, arXiv:1704.00390 (2017) https://arxiv.org/abs/1704.00390.

Kihyuk, S. et al. "Learning structured output representation using deep conditional generative models". Advances in neural information processing systems. (2015).

Knudsen, Lotte B. et al. “Potent Derivatives of Glucagon-like Peptide-1 with Pharmacokinetic Properties Suitable for Once Daily Administration.” Journal of Medicinal Chemistry, 43(9), 1664–1669 (2000). doi:10.1021/jm9909645.

Koch, Gregory et al. “Siamese Neural Networks for One-shot Image Recognition”. Proceedings of the 32nd International Conference on Machine Learning, Lille, France, 37 (2015). https://www.cs.cmu.edu/~rsalakhu/papers/oneshot1.pdf

Koehler, Michael F.t. et al. “Albumin Affinity Tags Increase Peptide Half-Life in Vivo.” Bioorganic & Medicinal Chemistry Letters, 12(20) 2883–2886 (2002). doi:10.1016/s0960-894x(02)00610-8.

Kuzminykh, D. et al. “3D Molecular Representations Based on the Wave Transform for Convolutional Neural Networks”. Molecular Pharmaceutics. 15, 4378–4385 (2018).

Lambert, TJ. “FPbase: a community-editable fluorescent protein database”. Nature Methods. 16, 277–278, (2019). doi: 10.1038/s41592-019-0352-8

Lau, Thomas et al. “Brendan - A Deep Convolutional Network for Representing Latent Features of Protein-Ligand Binding Poses”. Stanford CS231 Report (2017).

http://cs231n.stanford.edu/reports/2017/pdfs/531.pdf

Lensink, H.F. et al. “Docking, scoring, and affinity prediction in CAPRI”. Proteins: Structure, Function, and Bioinformatics. 81, 2082–2095 (2013).

Lensink, M.F. “Modeling protein-protein and protein-peptide complexes: CAPRI 6th edition”. Proteins: Structure, Function, and Bioinformatics. 85, 359–377 (2016).

Liao, S. et al. “Spherical Regression: Learning Viewpoints, Surface Normals and 3D Rotations on n-Spheres”, arXiv:1904.05404. (2019). https://arxiv.org/abs/1904.05404.

Liebschner, Dorothee, et al. “Macromolecular Structure Determination Using X-Rays, Neutrons and Electrons: Recent Developments in Phenix.” Acta Crystallographica Section D Structural Biology, 75(10),861–877 (2019). doi:10.1107/s2059798319011471.

Liu, Ming, et al. “Learning How to Actively Learn: A Deep Imitation Learning Approach.” Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2018, doi:10.18653/v1/p18-1174.

Maier, J. A. et al. “ff14SB: Improving the Accuracy of Protein Side Chain and Backbone Parameters from ff99SB”. Journal of Chemical Theory and Computation. 11, 3696–3713 (2015).

Niwa, T., et al. “Bimodal Protein Solubility Distribution Revealed by an Aggregation Analysis of the Entire Ensemble of Escherichia Coli Proteins.” Proceedings of the National Academy of Sciences, 106(11), 4201-4206 (2009). doi:10.1073/pnas.0811922106.

Noroozi, M. “Unsupervised Learning of Visual Representations by Solving Jigsaw Puzzles”, arXiv:1603.09246 (2016). https://arxiv.org/abs/1603.09246.

O'boyle, Noel M. et al. “Open Babel: An Open Chemical Toolbox.” Journal of Cheminformatics, 3(1), (2011). doi:10.1186/1758-2946-3-33.

Ronneberger, Olaf et al. “U-Net: Convolutional Networks for Biomedical Image

Segmentation.” arXiv:1505.04597 (2015). http://arxiv.org/abs/1505.04597.

Ruder, Sebastian. “Transfer Learning - Machine Learning’s Next Frontier.” Blog post. (2017). https://ruder.io/transfer-learning/

Park, H. et al. “High-resolution protein–protein docking by global optimization: recent advances and future challenges”. Current Opinion in Structural Biology. 35, 24–31 (2015).

Parviainen, Ville et al. “Relative Quantification of Several Plasma Proteins during Liver Transplantation Surgery.” Journal of Biomedicine and Biotechnology, 1–12, (2011). doi:10.1155/2011/248613.

Pereira, J.C. et al. “Boosting Docking-Based Virtual Screening with Deep Learning”. Journal of Chemical Information and Modeling. 56, 2495–2506 (2016).

Salehi, S.S.M. et al. “Tversky loss function for image segmentation using 3D fully convolutional deep networks”, arXiv:1706.05721 (2017). https://arxiv.org/abs/1706.05721.

Smith, L.M. “Proteoform: a single term describing protein complexity”. Nature Methods. 10, 186–187 (2013).

Šoškić, Milan, and Volker Magnus. “Binding of Ring-Substituted Indole-3-Acetic Acids to Human Serum Albumin.” Bioorganic & Medicinal Chemistry, 15(13), 4595-4600 (2007). doi:10.1016/j.bmc.2007.04.005.

Stepniewska-Dziubinska, M.M. et al. “Development and evaluation of a deep learning model for protein–ligand binding affinity prediction”. Bioinformatics. 34, 3666–3674 (2018).

Svenson, Johan, et al. “Albumin Binding of Short Cationic Antimicrobial Micropeptides and Its Influence on the in Vitro Bactericidal Effect.” Journal of Medicinal Chemistry, 50(14), 3334–3339, (2007). doi:10.1021/jm0703542.

Szegedy, Christian, et al. “Going Deeper with Convolutions.” 2015 IEEE Conference on Computer Vision and Pattern Recognition (2015). doi:10.1109/CVPR.2015.7298594.

Thorndike, R. L. “Who belongs in the family?” Psychometrika. 18, 267–276 (1953).

Torng, W.; Altman, R. B. “3D Deep Convolutional Neural Networks for Amino Acid Environment Similarity Analysis”. BMC Bioinformatics. 18(1), (2017).

Wallach, Izhar, et al. “AtomNet: A Deep Convolutional Neural Network for Bioactivity Prediction in Structure-Based Drug Discovery.” arXiv:1510.02855 (2015). http://arxiv.org/abs/1510.02855.

Wang, Y. et al. “Predicting Protein Interactions Using a Deep Learning Method-Stacked Sparse Autoencoder Combined with a Probabilistic Classification Vector Machine”. Complexity. (2018).

Wannier, Timothy M., et al. “Monomerization of Far-Red Fluorescent Proteins.” Proceedings of the National Academy of Sciences, 115(48), (2018). doi:10.1073/pnas.1807449115.

Watts, K. Shawn, et al. “ConfGen: A Conformational Search Method for Efficient Generation of Bioactive Conformers.” Journal of Chemical Information and Modeling. 50(4), 534-546, (2010). doi:10.1021/ci100015j.

Wodak, S.J. et al. “Computer analysis of protein-protein interaction”. Journal of Molecular Biology. 124, 323–342 (1978).

Worrall, D. “CubeNet: Equivariance to 3D Rotation and Translation”, arXiv:1804.04458 (2018). https://arxiv.org/abs/1804.04458.

Xiang, Y. et al. “PoseCNN: A Convolutional Neural Network for 6D Object Pose Estimation in Cluttered Scenes”, arXiv:1711.00199 (2017). https://arxiv.org/abs/1711.00199.

Xu, K. et al., “Deep Structured Generative Models”, arXiv:1807.03877 (2018).

https://arxiv.org/abs/1807.03877.

Yang, Jian, et al. “Synthesis and Antiviral Activities of Novel Gossypol Derivatives.” Bioorganic & Medicinal Chemistry Letters, 22(3), 1415-1420, (2012). doi:10.1016/j.bmcl.2011.12.076.

Zacharias, M. “Accounting for conformational changes during protein–protein docking”. Current Opinion in Structural Biology. 20, 180–186 (2010).

Zhang, Y “TM-align: a protein structure alignment algorithm based on the TM-score”. Nucleic Acids Research. 33, 2302–2309 (2005).

Chapter 2: Modeling and Molecular Dynamics

2.1 CDRexAb: Antibody Small-Molecule Conjugates with Computationally Designed Target-Binding Synergy

This work is the primary thesis project of Jingzhou Wang from the Mayo group. In contributing to this project, I created the computational method to virtually screen for the optimal linker length and attachment site to conjugate small molecules into antibody CDRs. This consisted of a series of programs which generated rotamer libraries of the grafted small molecule + linker on a Sun Grid Engine (Gentzsch, 2001) cluster and parallelized the task of grafting them into the desired contact site on the molecule to be bound by the CDRExAb. This created many thousands of putative complexes which were then scored by their ability to achieve a geometry suitable for incorporation to the CDR of the starting antibody, without clashes. Additionally, I parameterized the exogenous small molecule-linker constructs using Gaussian (Frisch et al., 2016) and Ambertools (Case et al., 2021), and performed molecular dynamics simulations which were used to identify the need for framework mutations.

Manuscript as Submitted- Note some supplementary notes have been removed for brevity. Citations are independent of rest of thesis.

Jingzhou Wang[a,b], Aiden J. Aceves[a], Stephen L. Mayo[a,b]

[a]Division of Biology and Biological Engineering, California Institute of Technology, Pasadena, CA 91125; and [b]Division of Chemistry and Chemical Engineering, California Institute of Technology, Pasadena, CA 91125

Abstract

Antibody-drug conjugates (ADCs), or chimeric modalities in general, combine the advantages and offset the flaws of their constituent parts to achieve a broader target space than traditional approaches of pharmaceutical development. We combine the concept of ADCs with the full atomic simulation capability of computational protein design to define a new class of molecular recognition agents: CDR-extended antibodies, abbreviated as CDRexAbs. A CDRexAb incorporates a small-molecule binding event into *de novo* designed antibody/target interactions, creating antibody small-molecule conjugates that bind tighter against the target of the small molecule than the small molecule itself. In a proof-of-concept study using monomeric streptavidin/biotin pairs at either a nanomolar or micromolar-level affinity, we efficiently designed nanobody-biotin conjugates that exhibited >20-fold affinity improvement against the protein targets, with step-wise optimization of binding kinetics and the overall stability. The workflow explored through this process could be potentially used as a new

way to optimize small-molecule based therapeutics, and explore new chemical and target space of molecular-recognition agents in general.

Significance

We defined a generalizable new method of optimizing molecular recognition reagents that involve synthetic small molecules, and demonstrated a successful application of this method on a model system. Instead of optimizing the functional groups of a small molecule by organic chemistry methods, we used computational protein design to build a conjugating antibody domain to create chimeric molecules with target-binding strength and other physicochemical properties efficiently tunable by the amino acid sequence of the antibody scaffold. This method expands the application scenarios of antibody-drug conjugates and brings in a previously-irrelevant chemical space into the optimization of small molecule binding events, potentially addressing some long-standing challenges of developing molecular binders.

Introduction

Most pharmaceutical mechanisms involve drug-target interactions that are mediated by synthetic small molecules or monoclonal antibodies—the two major drug modalities [1,2]. Despite impressive successes, many biological pathways are still difficult or even impossible to be pharmaceutically intervened, often because through existing approaches either the desired interactions are fundamentally difficult to be engineered, or the pharmacological trade-offs for

establishing the interactions outweigh potential benefits [3-6]. Therefore, new modalities that incorporate new chemistry and new biology are constantly created to realize a versatile toolkit that more easily tackles certain challenging targets, and also expands the targetable molecular space itself [7]. To create new modalities, one common way is combining existing modalities to consolidate individual advantages and offset individual flaws [7,8]. Antibody-drug conjugates (ADCs), for example, takes advantage of the excellent specificity and biological compatibility of monoclonal antibodies to improve therapeutic indices of existing small-molecule drugs [8,9]. Traditionally, the antibody and drug components of ADCs are separately developed and bind to different targets while in action [8,9]. Most current ADCs improve the specificity of conjugated drugs as they deliver the small molecules into cell targets through specific antibody-induced receptor endocytosis [9,10]. Some ADCs and peptide-drug conjugates were also reported to improve the metabolic stability, circulation half-life, and solubility of linked small molecules through antibody-associated pharmacokinetics, chemical environment around the conjugation sites, and linker design, indicating that protein conjugation could modulate a wide range of small-molecule properties [10-14]. Recently, Cheng *et al.* from Amgen developed ADCs whose antibody and drug components bind to an identical protein target to achieve synergistic binding/inhibition effects [15]. In their study, the co-crystal structure of a small molecule drug sitagliptin, a separately-developed antibody 11A19, and the protein target DPP-IV was solved first [16]. Based on the structure, optimal conjugation sites and linker sequences were then searched to create ADCs that

exhibited 13 to 32-fold IC50 improvement than sitagliptin alone against the target [15]. Cheng *et al.*'s work suggested that small molecule binding events could be directly optimized by conjugated antibodies, turning ADC technology into a potential tool to expand the chemical space and therefore target space of molecular recognition agents that involve synthetic small molecules.

Overall, the above discoveries demonstrated the potential for using rationally-designed antibody conjugation to optimize the mechanism of action, along with many other pharmacologically-relevant properties, of small-molecule based binders. However, to engineer the binding synergy required for this kind of applications, established methods that separately develop and characterize the antibody and small molecule components would be resource intensive, thus limiting the application scenarios. To realize the above-mentioned potential, a workflow that can rapidly determine a compatible antibody sequence and conjugation strategy for a to-be-improved small molecule binding event would be ideal. However, whether such workflow is technically achievable is still a question.

In this study, we explored the feasibility of computationally designing the antibody component of synergistically-binding ADCs. We introduced the concept of CDR-extended antibodies (CDRexAbs), which refer to computationally-designed antibodies whose complementary-determining regions (CDRs) contain a small molecule ligand that binds to a certain target, with surrounding CDR sequences tailored to strengthen the target-binding interactions (Figure 1A). At this initial stage, we focused our design on nanobodies, which are llama-derived

single-domain antibody fragments that can function by themselves, with attached Fc domains, or reformatted into IgGs [17-19]. Using a modified streptavidin-biotin interaction pair as model system, we demonstrated that with only the structural knowledge of small-molecule/target interactions, nanobody small-molecule conjugates can be computationally designed to bind tighter against the target than the small molecule itself. Through subsequent sequence design, the affinity, binding kinetics, and overall stability of the conjugates can be improved in a step-wise manner. ≥ 20-fold affinity improvements together with targeted kinetic-tuning can be achieved when the starting small-molecule/target affinity is as weak as 1 μM, or as strong as 7 nM. Exploration of various computational methods revealed key design principles, from which we proposed a general design strategy for this new potential modality.

Results and Discussion:

Computationally-designed nanobody small-molecule conjugation creates tighter binders against the small-molecule target protein

We first asked whether computationally-determined nanobody sequences and their designed conjugation to a small molecule can exhibit an enhanced binding affinity to the small-molecule target. Designing the antibody components of synergistically-binding ADCs will involve creating new antibody/target interface, which is challenging, largely because of the difficulty in predicting the global minimum conformation of antibody CDR loops against a targeted surface, while accurately modeling long structured loops remains a challenge in general

[20-23]. To restrict unpredicted CDR conformations that could lead to non-binding designs, we decided to adopt an approach similar to the anchored-design methods [24]. Anchored-design creates new protein-protein interfaces by first identifying hotspot residues that favorably interact with the target, then designing protein scaffolds to stabilize the anchoring hotspots [24-26]. For synergistically-binding ADCs, the conjugated small molecule can be viewed as a hotspot "residue" that interacts with the target protein. Therefore, to create co-targeting ADCs, the drug can be designed as an anchoring non-natural CDR residue that is strengthened by additional CDR-target interactions, integrating the drug-target interaction into the antibody-target binding event, and forcing the CDRs to more likely adopt the designed conformation.

We therefore finalized the design strategy into the following steps: predicting the optimal CDR binding poses against the target surface, and searching for the ideal conjugation strategy that accommodates both the optimized CDR pose and the target-small molecule interaction. For demonstration purpose, we chose monomeric streptavidin as our model target and biotin as our model small molecule. Streptavidin-biotin interactions have been extensively studied with high-resolution crystal structures available for reliable design. Tetrameric streptavidin binds to biotin with almost the highest-possible affinity, but multiple monomeric streptavidin constructs were reported with $>10^5$-fold reduced biotin-binding affinity [27,28]. So as a model system, monomeric streptavidin-biotin interaction pairs not only provide room for affinity

improvement, but also have a known affinity upper limit, thus ideal for method development.

To search the optimal CDR binding conformations, we first docked a starting nanobody scaffold onto a monomeric core-streptavidin structure with computationally-modeled side chain replacements S45A/T90A/D180A, which were reported to monomerize streptavidin and reduce the biotin-binding affinity to 1.7 μM [27], and then performed loop-modeling on docked poses to attempt optimizing CDR conformations against the target surface. Most of the top loop modeling solutions were not representative of naturally occurring interactions. To sample realistic CDR structures, we instead only searched around previously-observed nanobody CDR binding conformations [29]. We curated nanobody structures with diverse target-binding CDR conformations from PDB, and individually docked them onto the target surface (Fig. S1A). 2310 docked poses were generated and filtered to potentially identify most realizable binding conformations, returning 7 final binding poses (Fig. S1C). Optimal conjugation strategy was then searched on the finalized poses. We chose to conjugate biotin onto nanobody CDRs by the cysteine-maleimide chemistry, which is a commonly used conjugation method in ADCs (Fig. S2A) [30]. Biotin C2 maleimide was chosen to be the conjugation reagent. Optimal nanobody scaffolds and conjugation sites were determined by computationally screening a rotamer library of the cysteine-conjugated side chain on the finalized nanobody-streptavidin poses. The top-ranked conjugation plan was amino acid site 103 of the nanobody scaffold 4NBX.B (chain B of PDB structure 4NBX), which originally binds to a

target unrelated to any streptavidin construct [31]. From the relaxed structure of the conjugate named as “4NBX.B-biotin103” in complex with monomeric streptavidin, Y112 and R27 of 4NBX.B are predicted to form hydrogen bonds with the target surface, whereas in the original PDB structure, these two residues also participated in H-bond formation, indicating that the designed pose is closely related to the natural binding mode of 4NBX.B, and potentially stabilized by specific CDR-target interactions upon biotin anchoring (Fig. S2B-C).

We then synthesized 4NBX.B with site 103 mutated to cysteine, and performed conjugation with biotin C2 maleimide. We attempted to purify and refold the S45A/T90A/D128A mutant of core-streptavidin to perform binding measurement, but the resulted construct was unstable, as most proteins precipitated during refolding, and the refolded materials also quickly precipitated. Therefore, we aligned another previously-reported monomeric streptavidin construct, mSA, onto the triple-mutation streptavidin model that mSA is homologous to (sequence pairwise identity: 57%, structure RMSD: 0.5Å), and relaxed 4NBX.B-biotin103 against mSA (Fig. S2D) [28]. The 4NBX.B-biotin103/mSA model preserved the rotamer configuration of conjugated biotin against the triple-mutation streptavidin, and H bonds contributed by Y112 and R27 were also recapitulated, and potentially participated in a broader predicted H-bond network that incorporated biotin/mSA interactions, suggesting that 4NBX.B-C103biotin may bind to mSA with the designed beneficial synergy (Fig. S2D, 1B-C). Indeed, surface plasmon resonance (SPR) binding experiments

confirmed that under 25℃ 4NBX.B-biotin103 binds to immobilized mSA with a K_D of 1.8±0.1 nM, and mSA binds to immobilized biotin with a K_D of 7.0±0.1 nM, indicating a moderate 4-fold affinity improvement that is contributed by a higher k_{on} (Fig. 1D-E, 5A). Wildtype 4NBX.B did not show binding signal to mSA at concentrations up to 100 nM, indicating that the 4NBX.B-biotin103 conjugate binds to the targeted biotin binding pocket (Fig. 1E left panel). The SPR-measured biotin/mSA affinity is similar to previously-published fluorescence polarization spectroscopy data, which is 2.8±0.5 nM under 4℃ and 5.5±0.2 nM under 37℃ [28]. However, because the data fitting quality of the mSA/biotin binding curves is lower than the 4NBX.B-biotin103 binding curves, to confirm the estimated mSA/biotin affinity, we performed an alternative estimation by binding immobilized mSA to Smt3 SUMO protein that was biotinylated at the N terminus by biotin C2 maleimide. Smt3 SUMO protein has an unstructured N-terminus that we hypothesized would minimize the interaction between the protein components [32]. A similar K_D is estimated with high data-fitting quality, indicating that the measured biotin/mSA affinity is an accurate SPR estimation (Fig. 1D right panel).

To know whether computationally-designed nanobody conjugation shows improved affinity with weakly-binding small molecules, we created a single mutation S27A on mSA, whose counterpart S45A in wild type streptavidin reduces biotin-binding strength and was predicted by molecular dynamics (MD) simulation to minimally affect the overall structure [33]. On size-exclusion chromatography (SEC), mSA_{S27A} is eluted at the same time as mSA_{WT} (Fig.

S3A). SPR estimated that mSA_{S27A} binds to biotin with a K_D of 1.14±0.02 μM, while 4NBX.B-biotin103 binds to mSA_{S27A} with a K_D of 245±41 nM, indicating a similarly-moderate 5-fold improvement (Fig. 1F, 5B). Together, the above results showed that based on the sole structural information of a small molecule-target interaction, nanobody conjugation to the small molecule can be designed entirely by computational methods to exhibit an affinity-enhancing synergistic binding effect.

Sequence design further improves the binding affinity and kinetics for computationally designed conjugates

Next, we performed sequence design on the CDR loops of 4NBX.B-biotin103 to improve its binding affinity against mSA and further validate the accuracy of the modeled binding pose. We *in silico* analyzed each CDR amino acid site for its favorability of accepting mutations, and performed combinatorial designs on the mutable sites. Four combinations with different site-selection biases were tested in parallel, and the residue choices for each site were decided according to a published study on the sequence diversity of nanobody CDR loops [34]. Analysis of design outputs revealed that the design with sites 31, 32, 104, and 105 most frequently returned sequences that were likely to form additional H-bonds with mSA and were also energetically stable. The top-ranked variant by energy, v119 with CDR1 mutations M31H/D32A and CDR3 mutations N104S/W105H, was predicted to form new H-bonds with residue Q108 of mSA by H31 and with E105 of mSA by H105 (Table S1, Fig. 2A). The D32A mutation also eliminates a buried

and unpaired charged residue that does not participate in extensive H-bond network formation. SPR measured the K_D of 4NBX.B-biotin103 v119 against mSA_{WT} to be 0.9±0.2 nM, indicating a ~2-fold improvement from 4NBX.B-biotin103 WT (Fig. 2A, 5A). However, the K_D improvement was again mainly contributed by k_{on} increase, while the observed k_{off} values were only minimally different (Fig. 2A, 5A). To obtain a variant that would more significantly reduce the k_{off}, we picked variant v149 that has the highest number of predicted H-bond formation from the top 20 output sequences. 4NBX.B-biotin103 v149 has mutations M31R/D32S/N104A/W105R that were predicted to form more extensive H-bonds with Y96, E105, and Q108 of mSA, with a potential salt bridge between the nanobody R105 and mSA E105 (Fig. 2B). Interestingly, R-E interactions seemed to be frequently used by nanobodies, further validating this designed interaction [35]. Indeed, compared to v119, SPR measured a ~2-fold slower k_{off} and a ~4-fold faster k_{on} for v149, which together contribute to the K_D of 0.12±0.01 nM, indicating a >20-fold K_D improvement from biotin/mSA_{WT} affinity (Fig. 2B, 4A). However, according to SEC traces, v149 seemed to be very prone to aggregation, indicating protein instability (Fig. 3A).

Sequence design reduces aggregation while preserving the binding strength for the designed conjugates

One SEC, both 4NBX.B-biotin 103 WT and v119 showed single peaks eluted roughly at the same time as wild type 4NBX.B nanobody, indicating stabilized monomer foldedness (Fig. 3A). The reduced monomer stability of v149

agrees with its predicted lower energy score than v119 (Table S1). Since only four residues were designed, to improve the stability of v149, we hypothesized that further CDR designs would better accommodate the biotin103 side chain and the four H-bond contributing mutations, thus stabilizing the loop and overall structure. We therefore performed two additional rounds of CDR residue mutability analysis followed by in-parallel combinatorial designs on v149 until no further CDR mutations were predicted to be energetically favorable. Mutations accumulated in previous rounds of design were kept intact in subsequent rounds. In top 20 sequences ranked by energy score of both rounds of design, no additional H-bond was predicted to form with mSA, so the sequences with the best energy improvement were selected. The resulted variant, v186, has 6 additional CDR mutations Y101L/R107F /R56T/Y106K/D108A/Y110S on top of v149, and was predicted to preserve the H-bonds contributed by v149 mutations. Indeed, v186 seemed to bind to mSA_{WT} with very similar K_D as v149 (Fig. S5). However, SEC traces of v186 showed even worse aggregates formation than v149 (Fig. 3A).

MD simulations have been successfully applied to reveal the source of unexpected functional properties in designed proteins [36]. To understand the flaws of the structure and inform next design strategy, we perform MD simulation of 4NBX.B-biotin103 v186 in complex with mSA_{WT}. From the simulation, we noticed that the CDR3 loop that originally folded over the β-barrel framework region became gradually widened from the initial conformation, and eventually protruded away from the framework (supplementary movie). The apparently

destabilized loop-framework geometry suggests that the framework sequence is not fully compatible with the mutated CDR sequences, and needs to be optimized. We therefore performed framework sequence design on v186, and the top-ranked variant v186_Fr was predicted to form additional H-bonds with CDR3 residues through the F37Y mutation (Fig. 3C, Top). In addition, the A12V mutation also apparently increases the hydrophobic shielding of the β-barrel core (Fig. 3C, Bottom). Interestingly, when the same framework sequence design was performed on v149, different from the v186 design, the A12V/F37Y mutations were predicted to be less energetically favorable than the parent v149, suggesting that the v186 mutations were a prerequisite for the A12V/F37Y mutations to be beneficial (Table S1-2).

4NBX.B-biotin103 v186_Fr showed significantly reduced aggregation on SEC. Collected fractions excluding the aggregates peak did not re-aggregate once rerun on SEC (Fig. 3A, S3B). SPR measured the K_D of v186_Fr to be 0.20±0.03 nM, which preserved the >20-fold K_D improvement from biotin/mSA_{WT} (Fig. 3B top panel, 5A). The kinetics profile of v186_Fr against mSA_{WT} was also similar to v149 (Fig. 3B top panel, 5A). When binding to mSA_{S27A}, v186_Fr exhibited K_D to be 54±3 nM, indicating a ~20-fold K_D improvement contributed by both improved association rate and dissociate rate (Fig. 3B bottom panel, 5B).

To further investigate the functionally-relevant structural features of v186 and v186_Fr, we performed additional 100 ns MD simulations of v186 and v186_Fr against mSA_{WT} in triplicate. In general, during the simulations both the overall binding geometry of the conjugates and the conformation of the biotin103 side

chain remained constant with small structural RMSDs (Fig. S4, 4B first panel). The 4NBX.B nanobody scaffold has two solvent-inaccessible clusters of hydrophobic residues in the framework, one being the β-barrel core and another shielded by the CDR3 loop (Fig. 4A). Stable solvent inaccessibility and packing of hydrophobic patches is usually correlated with protein folding stability, which is in turn related to aggregation [9,37]. For the majority of time in the MD simulations, the solvent-accessible area for the two hydrophobic clusters of both v186 and v186_Fr was distributed around similarly-low values, indicating that both variants should be generally foldable (Fig. 4A). However, in contrast to v186_Fr, v186 displayed apparent sub-populations whose hydrophobic core and CDR3-shielded hydrophobic residues were significantly more solvent-accessible, indicating possible structural instability that agrees with the expected stabilization effects of F37Y and A12V in v186_Fr (Fig. 4A). Additional analysis of the v186_Fr/mSA$_{WT}$ interface from the simulations indicates high shape complementarity, large buried interface area, and close interface distance that remained generally constant along the timescale, in agreement with the measured sub-nanomolar affinity (Fig. 4B). Overall, the design calculation, experimental data, and MD simulations are well-correlated with each other.

Affinity and kinetics estimation of 4NBX.B-biotin103 WT, v119, v149, and v186_Fr were performed in biological triplicates. To make sure the prepared conjugates homogeneously harbor one biotin-maleimide “side chain” per nanobody molecule, we used intact-protein mass spectrometry (MS) to analyze

one of the SPR-measured triplicates for each of the above-mentioned nanobody-biotin variants, as we reason one replicate should be representative given the small batch-to-batch variations in measured affinities (Fig. 5A-B). Deconvolution of MS spectra only returned components with molecular weights (MWs) within 20Da from the expected values of mono-biotin conjugates, while each conjugated biotin-maleimide “side chain” would add an additional mass of 366Da, indicating that all tested materials were effectively mono-conjugated with biotin C2 maleimide (Fig. S6). Subpopulations with ~+/-17Da from the expected MWs were observed, and could be contributed by ring-open products of succinimides or ion adducts (Fig. S6).

Although the affinity and kinetics improvements are well correlated to the designed mutations, confirming whether the predicted interactions were accurately established require structural determination. Crystallization attempts using mSA_{WT} in complex with v186_Fr and v119 only produced crystalline that failed to increase in size. This observation and the fact that the affinities of the designed conjugates in this study are predominantly affected by the biotin-binding affinity potentially indicate that the protein-protein interactions are relatively flexible and dependent on the biotin anchoring, suggesting that further improvements over the protein-protein interface are possible. Testing whether *in vitro* evolution methods could further improve the affinity and kinetics of the designed conjugates against their target would be a necessary next step.

Summary and further testing of a computational workflow for creating synergistically-binding nanobody small-molecule conjugates

Based on the above design results, we summarized the design process into the following general workflow: docking a library of nanobody structures with diverse CDR sequences and conformations onto a desired target in complex with the to-be-conjugated small molecule, filtering binding poses to preserve ones that closely resemble the original binding mode of the original nanobody scaffold, screening the rotamer library of the conjugated small molecule onto the poses to identify most tolerable conjugation plan, and finally re-designing the sequences of both the nanobody CDR loops and framework to improve binding affinity, kinetics, and overall stability (Fig. 5C). Because 4NBX.B was not obtained by directly docking nanobody scaffolds against mSA, we re-performed the docking, filtering, and rotamer screening steps on mSA, and selected a different scaffold, 2X89.A, with biotin conjugated to site 57 (Fig. S7A, and B top panel). Similar to 4NBX.B-biotin103 v186_Fr, the selected pose of 2X89.A was predicted to interact with mSA_{WT} through a R-E interaction, together with other potential intermolecular H-bonds (Fig. S7B top panel). Since the original 2X89.A has an additional intra-CDR disulfide bond, to avoid over conjugation, the disulfide bond was replaced by two alanine residues. The resulted final conjugate, 2X89.A-CCAA-biotin57 binds to mSA_{WT} with a K_D of 0.8±0.2 nM, and remarkably, a k_{off} that is slightly better than our best designed 4NBX.B variant v186_Fr (Fig. 5A, S7B bottom panel). 2X89.A-CCAA-biotin57 aggregated obviously on SEC (Fig. S7C). To reduce aggregation, we constructed a rudimentary sequence design

pipeline that sways between CDR and framework design based on our previous experience on designing 4NBX.B conjugates, and applied the pipeline on 2X89.A (Fig. S7D). Top-ranked variants along the six rounds of CDR designs and one round of framework design showed first worsened then improved aggregation profile after 18 mutations were accumulated (Fig. S7E), similar to what we observed in the design process of 4NBX.B conjugates.

Conclusion:

Using mSA/biotin system, we demonstrated for the first time to our knowledge that with the sole structural information of a small molecule binding to its target, a complementary immunoglobulin domain conjugating to the small molecule can be designed entirely by computational methods to bind tighter against the target, further bridging the two worlds of small molecules and biologics. The binding interface for the designed conjugates comprise of both an ultra-deep pocket that is uncommon for antibodies, and broad contacting interface that is uncommon for small molecules [38,39]. Therefore, the chemical space and target space of traditional molecular recognition agents could be expanded in this manner, offering new potential solutions to a wide range of challenges, such as reutilizing failed small molecules or tackling undruggable targets in pharmaceutical development. Our results showed that the affinity, kinetics, and stability of the conjugates can be designed in a step-wise manner, indicating that the development process is highly tunable and multiple physicochemical properties can be simultaneously optimized.

Testing whether the design strategy introduced in this study works for therapeutically-relevant targets would be a crucial next step. It will also be beneficial to study whether the workflow works with virtually-docked small-molecule/target complexes. In addition, testing whether the workflow can engineer specificity in addition to affinity will be also highly desirable. There are many computational methods that could be used to improve the design strategy. Virtually recombining structural fragments was reported to help affinity maturation of computationally designed antibodies [26,40,41]. Specifically-tailored algorithms that put more bias in the formation of hydrogen-bonding networks were also proven to be useful to the affinity and specificity of designed protein/protein interfaces [42]. Advanced loop-modeling methods and ensemble design could also facilitate more accurate assessment of binding poses for the conjugates, and potentially engineer specificities [43,44].

Because the designed conjugates have the CDR loops chemically extended beyond the natural repertoire, we name the computationally-designed synergistically-binding antibody small-molecule conjugates to be CDR-extended antibody, abbreviated as CDRexAb.

Materials and Methods:

Computational design workflow for nanobody-biotin conjugates: The detailed description of the computational design workflows for the nanobody-biotin conjugates introduced in this study is included in the supplementary material.

Plasmids, expression cell lines, and cloning of protein variants: pRSET-mSA was a gift from Sheldon Park (Addgene plasmid # 39860) [28]. S27A mutation was created by site-directed mutagenesis using commercially-available kits (NEB). 4NBX.B_C103 and 2X89.A_CCAA_C57 sequences were directly ordered from IDT, and cloned into pHen6c vector by Gibson assembly using commercially-available reagents (NEB) [45]. The assembled pHen6c vectors harbor a PelB signal sequence before the N terminus of the nanobody sequence, allowing bacterial periplasmic expression [46]. Variants of 4NBX.B_C103 and 2X89.A_CCAA_C57 were created by mutagenic PCR and assembled into pHen6c vector by Gibson assembly using commercially-available reagents (NEB) [45]. 4NBX.B WT sequence with C103A mutation was created by site-directed mutagenesis using commercially available kits (NEB). Smt3 SUMO protein with an N-terminal cysteine was created from wild type Smt3 SUMO by mutagenic PCR, and subcloned into pY71A(lc) vector by Gibsom assembly using commercially-available reagents (NEB) [45].

Expression and purification of mSA streptavidin wild type and S27A variant: Expression, purification, and refolding of mSA variants followed published protocols with slight variations [28]. The expression plasmids were first transformed to E. Coli BL21-Gold (DE3) chemically competent cells (Agilent), which were then grown overnight in LB with 100 µg/mL of ampicillin (amp 100) at 37°C and 250 rpm. 1 mL of the overnight culture was used to inoculate 300 mL of TB medium (2.3 g KH_2PO_4, 16.4 g K_2HPO_4, 12 g tryptone, 24 g yeast extract, 4 mL glycerol, dissolved in water to 1 L volume) supplemented with 2 mM $MgCl_2$,

0.1% glucose, and amp 100. Inoculation was done at 37°C and 250 rpm until OD_{600} hit 1.5-2. Expression was induced by 1 mM IPTG at 28°C and 250 rpm for 18 hours. Cells were then centrifuged by 4500 g for 15 minutes at 4°C, and protein extraction was then performed using 50 mL of chemical lysis buffer. The buffer was composed of 1x PBS (137 mM NaCl, 2.7 mM KCl, 10 mM Na2HPO4, 1.8 mM KH2PO4, pH 7.4), 1x CelLytic B reagent (Sigma), 0.02 mg/mL DNase1, 0.2 mg/mL lysozyme, and 1 mM protease inhibitor AEBSF (Sigma). Pellets were resuspended in lysis buffer and nutated for 4 hours at room temperature. Cell lysate was then centrifuged at 15000 g for 30 minutes at 4°C, and the precipitates were used to refold and purify the protein.

Precipitates were first resuspended into 3 mL of 6 M guanidine hydrochloride in 1x TBS (50 mM Tris, 150 mM NaCl, pH 8.0) and incubated under 37°C for 30 minutes to solubilize the proteins. Un-dissolved materials were cleared by 15000 g centrifugation for 5 minutes at 4°C. Supernatants were then chilled on ice before added drop by drop into 40 mL of pre-chilled refolding buffer (50 mM Tris-HCl, 150 mM NaCl, 0.3 mg/mL D-biotin, 0.2 mg/mL oxidized glutathione, 1 mg/mL reduced glutathione, pH 8.0) while stirring. The refolding buffer with added mSA protein was then allowed to incubate on ice for another 2 hours with stirring before centrifuged by 15000 g for 30 minutes at 4°C to remove insoluble materials. The supernatants were then supplemented with 20 mM imidazole and then loaded onto 1 mL bed volume of Ni-NTA agarose beads (Qiagen) which were pre-washed with 5 column volumes of 1x PBS. Sample loading was performed by gravity flow. Column was then washed with 10 column

volumes of 1x PBS supplemented with 20 mM imidazole before 3 mL of 1x PBS supplemented with 500 mM of imidazole was used to elute the proteins. The entire Ni-NTA purification process was done at 4°C. The 3 mL of purified proteins were then dialyzed against fresh 1 L of 1x PBS (pH 8.0) under 4°C for 3 times using Slide-A-Lyzer 10 kDa molecular-weight cutoff dialysis cassettes (Thermo). Each dialysis step took longer than 4 hours. The dialyzed mSA products were further purified at 4°C by Superdex 75 10/300 GL SEC column (GE) using 1x PBS (pH 7.4) as running buffer, and the fractions corresponding to the monomeric peak were collected for subsequent experiments.

Expression and purification of Smt3 SUMO protein with N-terminal cysteine: The expression plasmids were first transformed into E. Coli BL21-Gold (DE3) chemically competent cells (Agilent), which were then grown overnight in LB with amp 100 and 1% glucose at 37°C and 250 rpm. 1 mL of the overnight culture was then used to inoculate 300 mL of TB medium with 2 mM $MgCl_2$, 0.1% glucose, and amp 100 at 37°C and 250 rpm until OD_{600} hit 1.5-2. Expression was then induced by 1 mM IPTG at 28°C and 250 rpm for 18 hours. Cells were then pelleted under 4500 g for 15 minutes at 4°C, resuspended in 50 mL of chemical lysis buffer supplemented with 5mM 2-mercaptoethanol (BME), and incubated for 4 hours at room temperature to extract the expressed proteins. Lysate supplemented with 20 mM imidazole was cleared by 15000 g centrifugation at 4°C for 30 minutes, and the supernatant was loaded onto 1 mL bed volume of Ni-NTA agarose beads (Qiagen) pre-washed with 5 column volumes of 1x TBS (pH 7.3). Sample loading was performed by gravity flow. The loaded column was

then washed with 5 column volumes of 1x TBS (pH 7.3) supplemented with 20 mM imidazole and 5 mM BME and another 5 column volumes of 1x TBS (pH 7.3) supplemented with 20 mM imidazole. 3 mL of 1x TBS (pH 7.3) supplemented with 500 mM of imidazole was used to elute the proteins. All Ni-NTA purification procedures were performed under 4°C. Purified proteins were then concentrated to ~0.5 mL using Amicon 10 kDa molecular-weight cutoff centrifuge filters (GE), and stored for subsequent experiments.

Expression and purification of nanobodies: The pHen6c expression plasmids were first transformed into E. Coli BL21-Gold (DE3) chemically competent cells (Agilent), which were then grown overnight in LB with amp 100 and 1% glucose at 37°C and 250 rpm. 1 mL of the overnight culture was then used to inoculate 300 mL of TB medium with 2 mM $MgCl_2$, 0.1% glucose, and amp 100 at 37^{o}C and 250 rpm until OD_{600} hit 1.5-2. Expression was then induced by 1 mM IPTG at 28°C and 250 rpm for 18 hours. Cells were then pelleted under 4500 g for 15 minutes at 4°C, and suspended in 12 mL of TES periplasmic extraction buffer (0.2 M Tris, 0.5 mM EDTA, 0.5 M sucrose, pH 8.0), supplemented with 5 mM BME if the nanobody had a cysteine handle for conjugation, before incubated on ice with shaking at 32 rpm for 1 hour [47,48]. 18 mL of 4x diluted TES buffer, supplemented with 5mM BME if the nanobody had a cysteine handle, was then added to the cells which were incubated on ice at 32 rpm for another hour [47,48]. After periplasmic extraction, the cells were pelleted by 15000 g at 4°C for 30 minutes, and the supernatants were supplemented with 20 mM of imidazole before loaded onto 1 mL bed volume of Ni-NTA agarose

beads (Qiagen) pre-washed with 5 column volumes of 1x TBS (pH 7.3). Sample loading was performed by gravity flow. The loaded column was then washed with 10 column volumes of 1x TBS (pH 7.3) supplemented with 20 mM of imidazole, before 3 mL of 1x TBS (pH 7.3) supplemented with 500 mM of imidazole was used to elute the proteins. For nanobodies with a cysteine handle, 5 mM BME was added to the first 5 column volumes of wash buffer, and the elution buffer was supplemented with 5 mM TCEP. The elution buffer was incubated with the beads for 30 minutes before eluting the proteins. The entire Ni-NTA purification process was done at 4°C. Purified nanobodies with a cysteine handle in 3 mL of the elution buffer were concentrated to ~0.5 mL using Amicon 10 kDa molecular-weight cutoff centrifuge filters (GE), and stored for subsequent experiments. 4NBX.B WT was instead further purified by Superdex 75 10/300 GL SEC column (GE) using 1x PBS (pH 7.4) as running buffer, and the fractions corresponding to the monomeric peak were collected for subsequent experiments.

Biotin C2 maleimide conjugation and purification of conjugates: Maleimide labeling on surface cysteines of nanobodies followed a published protocol with some modifications [49]. Purified nanobodies in storage were first incubated with another 5 mM TCEP supplement under 4°C for 2 hours, and then buffer exchanged to 1x TBS (pH 7.3) using HiTrap desalting columns (GE) under room temperature to remove TCEP. Thawed stock solutions (100 mM in DMSO) of biotin C2 maleimide (AnaSpec) were immediately added to the buffer-exchanged nanobodies to 1 mM final concentration before the reaction mixture was nutated under 4°C for 4 hours with tinfoil cover to avoid light contact. Excess maleimide

stock solutions were tossed away and not re-frozen for future experiments. Biotin C2 maleimide was in >20-fold molar excess over the nanobody in the reaction mixture. Finished reaction mixture was then filtered by 0.2 µm syringe filters (Thermo) to remove precipitated proteins, and then buffer exchanged to 1x TBS (pH 7.3) using PD-10 desalting columns (GE) under room temperature to remove excess maleimide reagents. The labeled nanobodies were further purified at 4°C by Superdex 75 10/300 GL SEC column (GE) using 1x TBS (pH 7.3) as running buffer, and the fractions corresponding to the monomeric peak were collected for subsequent experiments. Maleimide labeling of Smt3 SUMO protein with N-terminal cysteine followed the identical procedures as above.

Intact protein mass spectrometry (MS) workflow to analyze conjugation efficiency: HPLC-MSD (HP, Agilent) was used to assess the labeling efficiency of prepared nanobody-biotin conjugates. Conjugates were first dried out using a spin vacuum evaporator, and resuspended in 0.2% formic acid. A C3 HPLC column was used first to separate the protein sample before MS analysis. Before running samples, the column was first washed with isopropyl alcohol (IPA) to clean the column and also reveal background peaks irrelevant to our samples. Aanalysis of conjugation efficiency of 4NBX.B-based conjugates was performed by deconvoluting the eluted sample HPLC peak using the following parameters: positive adduct ion +H 1.0079 Da, negative adduct ion –H -1.0079 Da, molecular weight cutoff 5000-80000 Da, maximum charge 90, minimum peaks 5, ion PWHH 0.6 Da, molecular weight agreement 0.05%, noise cutoff 0, abundance cutoff 10%, molecular weight assignment cutoff 40%, and envelope cutoff 50%.

Deconvolution was performed in ChemStation (Agilent). For each sample of interest, about 0.1-1 µg of material was used for the above analysis.

Surface plasmon resonance (SPR) analysis of binding affinity and kinetics: A Biacore T200 instrument (GE) was used to perform SPR analysis. 4NBX.B WT, 4NBX.B-biotin103 conjugates, and 2X89.A-CCAA-biotin57 conjugates were first buffer-exchanged to HBS-EP+ buffer (Teknova) using Amicon 10 kDa molecular-weight cutoff centrifuge filters (GE). The concentrations of the conjugates were then determined by BCA assay using commercially-available kits (Thermo). The calibration curve for BCA assay was prepared using purified 4NBX.B WT, which was also buffer exchanged to HBS-EP+ but had concentrations determined by A_{280} readings using extinction coefficient 30035 $M^{-1}cm^{-1}$. For SPR analysis, biotin pentylamine (Thermo), mSA_{WT}, and mSA_{S27A} were respectively immobilized on CM5 censor chip (GE) by EDC/NHS amine coupling kit following standard protocol (GE). Binding kinetics were measured by single-cycle kinetics experiments. Biotin pentylamine was immobilized at 7.5 mM concentration to reach target surface density of ~200 resonance units (RUs) [50]. In order to compare how binding events changed in response to different surface densities, surfaces with three different densities of immobilized mSA_{WT} at ~1000 RU, ~2500 RU, and ~3000 RU were respectively prepared under immobilization concentrations 0.1 µM, 0.5 µM, and 1 µM. Immobilization of mSA_{S27A} was also performed at 0.01 µM and 0.05 µM concentrations with target surface density of ~200 and ~600 RU. The fitted affinities and kinetics of identical conjugates against the different densities were minimally different. Reference channels were

either treated with EDC/NHS using blank HBS-EP+ buffer, or 1 µM of 4NBX.B WT to assess if the conjugates would self-associate and roughly see if the conjugates un-specifically interacted with proteins not of interest. No visible signal differences against reference channels with or without immobilized 4NBX.B WT were observed for various tested conjugates. All immobilization samples were dissolved in acetate buffer (pH 4.5).

Binding experiments were performed under 25^{o}C. HBS-EP+ was used as running buffer. The flow channels were first incubated in the running buffer before analytes at 5 different concentrates were consecutively injected at 30 µL/min flow rate through both the reference channel and the sample channel with immobilized molecules of interest. After injections, the surface-bound analytes were allowed to dissociate for 10 minutes to generate dissociation curves. HBS-EP+ buffer then washed through both reference and sample channels continuously to allow the rest of the bound analytes to dissociate, in order to regenerate the surfaces for next binding experiments. Curve fitting of sensorgrams processed by subtracting the reference channel signal from the sample channel signal was performed in Biacore evaluation software using 1:1 kinetics model. No incompletely subtracted bulk contributions were observed in binding against immobilized biotin. For binding curves against immobilized mSA_{WT}, global fitting of bulk shifts was turned on as small bulk shift contributes before and after each injection event were distinctively observed. Bulk shift fitting was turned off in binding curves involving mSA_{S27A}, because potential bulk shift

signals would be obscured by the kinetics curves with fast dissociation rates and therefore not distinctively visible.

Molecular dynamics (MD) simulation protocols: Molecular Dynamics simulations were carried out using ACEMD (Acellera) [51]. Each system studied was placed in a box with dimensions selected to allow an excess length of 12 angstroms on each side. The system was solvated using the TIP3P water model [52], and ions were added to neutralize the overall charge. The built system was then minimized for 500 steps. Subsequently, a 5 ns equilibration was completed to allow the system to reach a stationary state, and a 100 ns production run was carried out at 300 degrees K. All experiments utilized the Amber ff14SB force field and a 4 femtosecond timestep [53]. Data from the equilibration run was not included in subsequent analysis, and where replicates were collected no part of the intermediate data was reused. Parameters for the biotin-CH2-CH2-succinimide-S-CH3 “side chain” were prepared using Antechamber and utilized RESP charges calculated with Gaussian 09 [54,55]. Calculation of solvent accessible surface area was performed using MDTraj, and hydrogen bonding was assessed using a tcl script written for VMD [56, 57].

Acknowledgement

We thank Monica C. Breckow for technical assistance. We thank Dr. Jost G. Vielmetter from Protein Expression Center at Caltech for consultation over Biacore experimental design and data interpretation. We thank Dr. Barry D.

Olafson and Paul Chang from Protabit LLC for technical consultation over Triad and Biograf implementation.

References:

1. D. C. Swinney, Biochemical mechanisms of drug action: what does it take for success? *Nature Reviews Drug Discovery* **3**, 801–808 (2004).

2. J. K. H. Liu, The history of monoclonal antibody development – Progress, remaining challenges and future innovations. *Annals of Medicine and Surgery* **3**, 113–116 (2014).

3. A. L. Hopkins, C. R. Groom, The druggable genome. *Nature Reviews Drug Discovery* **1**, 727–730 (2002).

4. G. L. Verdine, L. D. Walensky, The Challenge of Drugging Undruggable Targets in Cancer: Lessons Learned from Targeting BCL-2 Family Members. *Clin Cancer Res* **13**, 7264–7270 (2007).

5. M. A. Ayoub, *et al.*, Antibodies targeting G protein-coupled receptors: Recent advances and therapeutic challenges. *MAbs* **9**, 735–741 (2017).

6. R. B. Dodd, T. Wilkinson, D. J. Schofield, Therapeutic Monoclonal Antibodies to Complex Membrane Protein Targets: Antigen Generation and Antibody Discovery Strategies. *BioDrugs* **32**, 339–355 (2018).

7. E. Valeur, *et al.*, New Modalities for Challenging Targets in Drug Discovery. *Angewandte Chemie International Edition* **56**, 10294–10323 (2017).

8. G. D. L. Phillips, *et al.*, Targeting HER2-Positive Breast Cancer with Trastuzumab-DM1, an Antibody–Cytotoxic Drug Conjugate. *Cancer Res* **68**, 9280–9290 (2008).

9. Y. Feng, *et al.*, Conjugates of Small Molecule Drugs with Antibodies and Other Proteins. *Biomedicines* **2**, 1–13 (2014).

10. W. D. Hedrich, T. E. Fandy, H. M. Ashour, H. Wang, H. E. Hassan, Antibody–Drug Conjugates: Pharmacokinetic/Pharmacodynamic Modeling, Preclinical Characterization, Clinical Studies, and Lessons Learned. *Clinical Pharmacokinetics* **57**, 687–703 (2017).

11. D. Su, *et al.*, Modulating Antibody–Drug Conjugate Payload Metabolism by Conjugation Site and Linker Modification. *Bioconjugate Chemistry* **29**, 1155–1167 (2018).

12. B.-Q. Shen, *et al.*, Conjugation site modulates the in vivo stability and therapeutic activity of antibody-drug conjugates. *Nature Biotechnology* **30**, 184–189 (2012).

13. Y. Wang, *et al.*, Peptide–Drug Conjugates as Effective Prodrug Strategies for Targeted Delivery. *Adv Drug Deliv Rev* **110–111**, 112–126 (2017).

14. R. Y. Zhao, *et al.*, Synthesis and Evaluation of Hydrophilic Linkers for Antibody–Maytansinoid Conjugates. *J. Med. Chem.* **54**, 3606–3623 (2011).

15. A. C. Cheng, *et al.*, Structure-guided Discovery of Dual-recognition Chemibodies. *Scientific Reports* **8**, 7570 (2018).

16. J. Tang, *et al.*, An Inhibitory Antibody against Dipeptidyl Peptidase IV Improves Glucose Tolerance in Vivo. *J. Biol. Chem.* **288**, 1307–1316 (2013).

17. S. Muyldermans, Nanobodies: Natural Single-Domain Antibodies. *Annual Review of Biochemistry* **82**, 775–797 (2013).

18. I. Hmila, *et al.*, VHH, bivalent domains and chimeric Heavy chain-only antibodies with high neutralizing efficacy for scorpion toxin AahI′. *Molecular Immunology* **45**, 3847–3856 (2008).

19. C. I. Webster, *et al.*, Brain penetration, target engagement, and disposition of the blood–brain barrier-crossing bispecific antibody antagonist of metabotropic glutamate receptor type 1. *The FASEB Journal* **30**, 1927–1940 (2016).

20. S. Fischman, Y. Ofran, Computational design of antibodies. *Current Opinion in Structural Biology* **51**, 156–162 (2018).

21. J. C. Almagro, *et al.*, Second antibody modeling assessment (AMA-II). *Proteins: Structure, Function, and Bioinformatics* **82**, 1553–1562 (2014).

22. L. T. Dang, *et al.*, Receptor subtype discrimination using extensive shape complementary designed interfaces. *Nat Struct Mol Biol* **26**, 407–414 (2019).

23. K. Kundert, T. Kortemme, Computational design of structured loops for new protein functions. *Biological Chemistry* **400**, 275–288 (2019).

24. S. M. Lewis, B. A. Kuhlman, Anchored Design of Protein-Protein Interfaces. *PLoS ONE* **6**, e20872 (2011).

25. S. J. Fleishman, *et al.*, Computational Design of Proteins Targeting the Conserved Stem Region of Influenza Hemagglutinin. *Science* **332**, 816–821 (2011).

26. X. Liu, *et al.*, Computational design of an epitope-specific Keap1 binding antibody using hotspot residues grafting and CDR loop swapping. *Scientific Reports* **7**, 41306 (2017).

27. M. H. Qureshi, J. C. Yeung, S.-C. Wu, S.-L. Wong, Development and Characterization of a Series of Soluble Tetrameric and Monomeric Streptavidin Muteins with Differential Biotin Binding Affinities. *J. Biol. Chem.* **276**, 46422–46428 (2001).

28. K. H. Lim, H. Huang, A. Pralle, S. Park, Stable, high-affinity streptavidin monomer for protein labeling and monovalent biotin detection. *Biotechnology and Bioengineering* **110**, 57–67 (2013).

29. G. Nimrod, *et al.*, Computational Design of Epitope-Specific Functional Antibodies. *Cell Reports* **25**, 2121-2131.e5 (2018).

30. N. Jain, S. W. Smith, S. Ghone, B. Tomczuk, Current ADC Linker Chemistry. *Pharm Res* **32**, 3526–3540 (2015).

31. T. Murase, *et al.*, Structural basis for antibody recognition in the receptor-binding domains of toxins A and B from Clostridium difficile. *J. Biol. Chem.* **289**, 2331–2343 (2014).

32. W. Sheng, X. Liao, Solution structure of a yeast ubiquitin-like protein Smt3: The role of structurally less defined sequences in protein–protein recognitions. *Protein Sci* **11**, 1482–1491 (2002).

33. F. Liu, J. Z. H. Zhang, Y. Mei, The origin of the cooperativity in the streptavidin-biotin system: A computational investigation through molecular dynamics simulations. *Sci Rep* **6** (2016).

34. C. McMahon, *et al.*, Yeast surface display platform for rapid discovery of conformationally selective nanobodies. *Nature Structural & Molecular Biology* **25**, 289–296 (2018).

35. L. S. Mitchell, L. J. Colwell, Analysis of nanobody paratopes reveals greater diversity than classical antibodies. *Protein Eng Des Sel* **31**, 267–275 (2018).

36. H. K. Privett, *et al.*, Iterative approach to computational enzyme design. *PNAS* **109**, 3790–3795 (2012).

37. A. Nisthal, C. Y. Wang, M. L. Ary, S. L. Mayo, Protein stability engineering insights revealed by domain-wide comprehensive mutagenesis. *PNAS* **116**, 16367–16377 (2019).

38. L. N. Makley, J. E. Gestwicki, Expanding the Number of 'Druggable' Targets: Non-Enzymes and Protein-Protein Interactions. *Chemical Biology & Drug Design* **81**, 22–32 (2013).

39. D. H. Nam, C. Rodriguez, A. G. Remacle, A. Y. Strongin, X. Ge, Active-site MMP-selective antibody inhibitors discovered from convex paratope synthetic libraries. *Proc. Natl. Acad. Sci. U.S.A.* **113**, 14970–14975 (2016).

40. D. Baran, *et al.*, Principles for computational design of binding antibodies. *Proceedings of the National Academy of Sciences*, 201707171 (2017).

41. C. A. Voigt, C. Martinez, Z.-G. Wang, S. L. Mayo, F. H. Arnold, Protein building blocks preserved by recombination. *Nature Structural Biology* **9**, 553–558 (2002).

42. S. E. Boyken, *et al.*, De novo design of protein homo-oligomers with modular hydrogen-bond network–mediated specificity. *Science* **352**, 680–687 (2016).

43. B. D. Allen, A. Nisthal, S. L. Mayo, Experimental library screening demonstrates the successful application of computational protein design to large structural ensembles. *PNAS* **107**, 19838–19843 (2010).

44. D. J. Mandell, E. A. Coutsias, T. Kortemme, Sub-angstrom accuracy in protein loop reconstruction by robotics-inspired conformational sampling. *Nature Methods* **6**, 551–552 (2009).

45. D. G. Gibson, *et al.*, Enzymatic assembly of DNA molecules up to several hundred kilobases. *Nat. Methods* **6**, 343–345 (2009).

46. D. Demeestere, *et al.*, Development and Validation of a Small Single-domain Antibody That Effectively Inhibits Matrix Metalloproteinase 8. *Molecular Therapy* **24**, 890–902 (2016).

47. K. Hand, M. C. Wilkinson, J. Madine, Isolation and purification of recombinant immunoglobulin light chain variable domains from the periplasmic space of Escherichia coli. *PLOS ONE* **13**, e0206167 (2018).

48. S. B. Hansen, N. S. Laursen, G. R. Andersen, K. R. Andersen, Introducing site-specific cysteines into nanobodies for mercury labelling allows de novo phasing of their crystal structures. *Acta Crystallographica Section D Structural Biology* **73**, 804–813 (2017).

49. T. Pleiner, *et al.*, Nanobodies: site-specific labeling for super-resolution imaging, rapid epitope-mapping and native protein complex isolation. *eLife* **4**, e11349 (2015)

50. M. L. B. Magalhães, *et al.*, Evolved streptavidin mutants reveal key role of loop residue in high-affinity binding. *Protein Sci* **20**, 1145–1154 (2011).

51. S. Doerr, M. J. Harvey, F. Noé, G. De Fabritiis, HTMD: High-Throughput Molecular Dynamics for Molecular Discovery. *J. Chem. Theory Comput.* **12**, 1845–1852 (2016).

52. W. L. Jorgensen, J. Chandrasekhar, J. D. Madura, R. W. Impey, M. L. Klein, Comparison of simple potential functions for simulating liquid water. *J. Chem. Phys.* **79**, 926–935 (1983).

53. J. A. Maier, *et al.*, ff14SB: Improving the Accuracy of Protein Side Chain and Backbone Parameters from ff99SB. *J. Chem. Theory Comput.* **11**, 3696–3713 (2015).

54. J. Wang, W. Wang, P. A. Kollman, D. A. Case, Automatic atom type and bond type perception in molecular mechanical calculations. *Journal of Molecular Graphics and Modelling* **25**, 247–260 (2006).

55. Gaussian 09, Revision A.02, M. J. Frisch, *et al.*, Gaussian, Inc., Wallingford CT, 2016.

56. R. T. McGibbon, *et al.*, MDTraj: A Modern Open Library for the Analysis of Molecular Dynamics Trajectories. *Biophysical Journal* **109**, 1528–1532 (2015).

57. W. Humphrey, A. Dalke, K. Schulten, VMD: Visual molecular dynamics. *Journal of Molecular Graphics* **14**, 33–38 (1996).

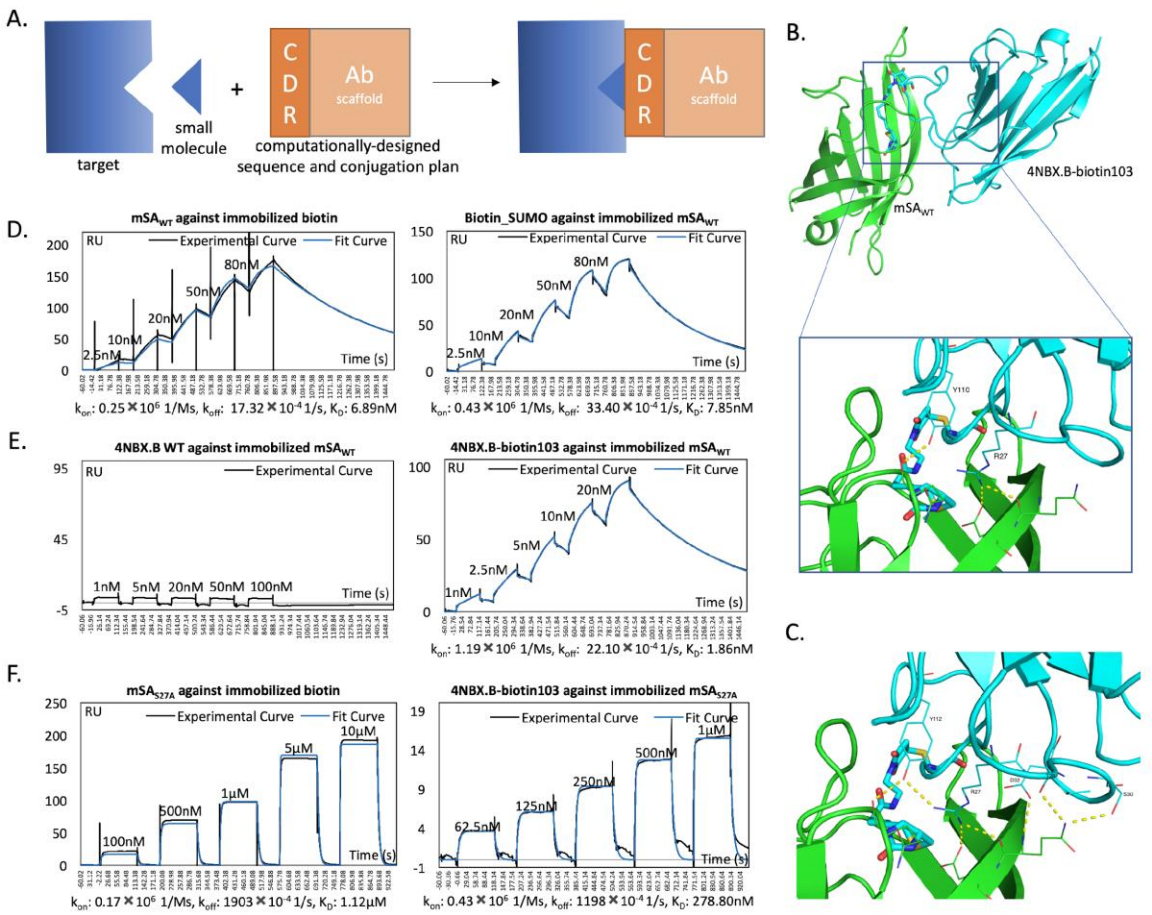


Figure 1: Computationally-designed nanobody-biotin conjugates bind stronger than biotin itself against mSA streptavidin. For SPR-measured K_D, k_{on}, and k_{off} results, data from one of the triplicates is shown here, and data from the other two replicates is in Fig. S5. (A). Schematic representation of the envisioned workflow. Given the availability of a small molecule and its target, the sequence of a complementary immunoglobulin domain and a conjugation plan with the small molecule are computationally determined to create conjugates that synergistically bind to the target. (B). Finalized model of 4NBX.B-biotin103 in complex with mSA streptavidin. The mSA is colored green, and nanobody scaffold is colored cyan. Biotin103 side chain is shown as stick, and the H-bond

forming potential of Y112 and R27 with mSA residues is also represented. (C). Y112 and R27 are predicted to participate in a broader potential H-bond network that involves biotin/mSA interactions. (D). SPR estimation of mSA/biotin binding parameters by two methods. (E). SPR measurements determined that 4NBX.B-biotin103 occupies the biotin-binding pocket of mSA with improved affinity and kinetics. (F). SPR measurements determined that 4NBX.B_biotin103 binds stronger towards a weaker biotin-binding mutant of mSA than biotin itself.

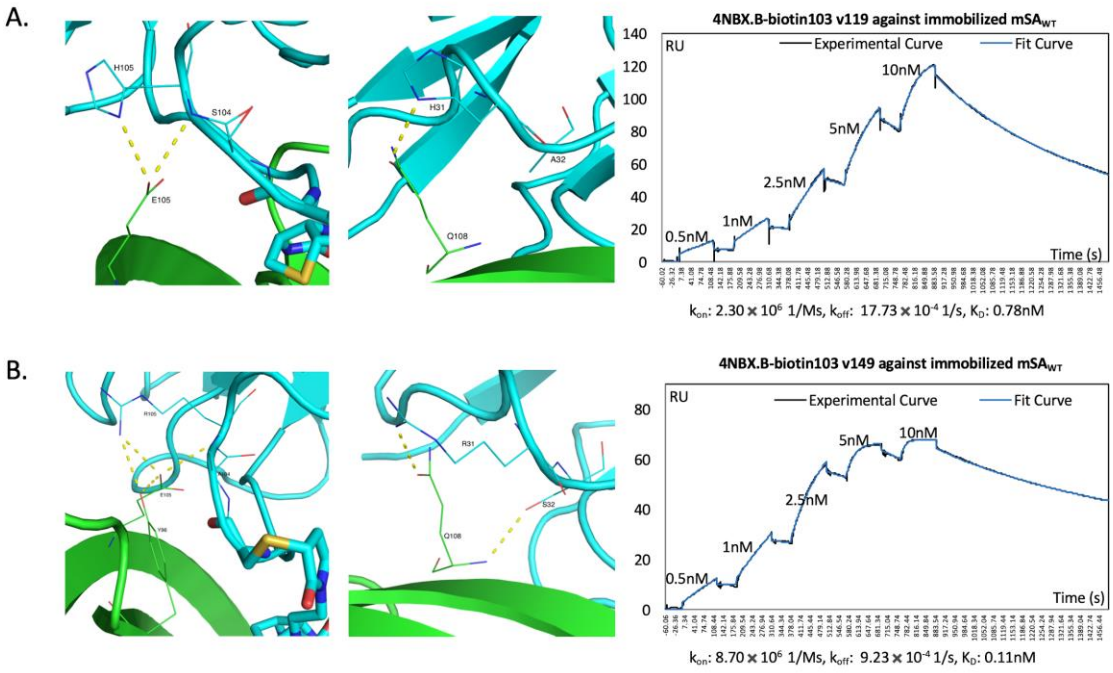


Figure 2: CDR sequence design enhanced the mSA-binding affinity and kinetics of 4NBX.B-biotin103. In the structural models, the mSA is colored green, and nanobody scaffold is colored cyan. Biotin103 side chain is shown as stick. For SPR-measured K_D, k_{on}, and k_{off} results, data from one of the triplicates is shown here, and data from the other two replicates is in Fig. S5. (A). Predicted

affinity-contributing mutations and SPR-measured binding profiles of 4NBX.B-biotin103 v119 against mSA_{WT}. (B). Predicted affinity-contributing mutations and SPR-measured binding profiles of 4NBX.B-biotin103 v149 against mSA_{WT}.

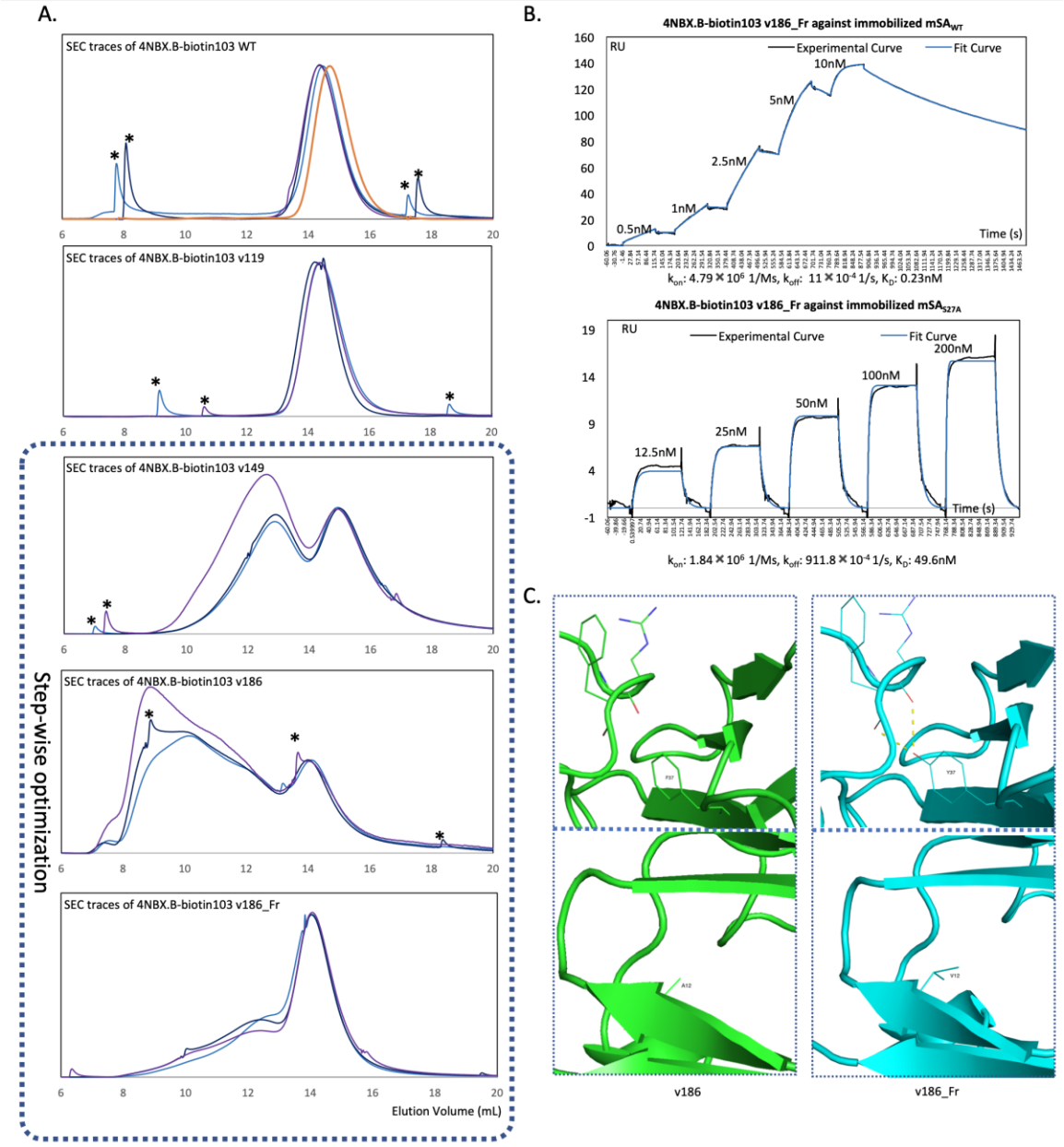


Figure 3. CDR sequence design followed by framework design monomerically stabilized the designed conjugates without imposing affinity penalty. (A). SEC traces of biological triplicates (colored by blue with different intensity) for designed 4NBX.B-biotin103 conjugates, normalized by monomer peak height for better comparison of aggregates formation. SEC trace

of 4NBX.B WT is overlaid with 4NBX.B-biotin103 WT traces and colored orange. * indicates peaks of sample-irrelevant instrument defect of the overall FPLC, please refer to Figure S3C for more details. SEC trace of 4NBX.B WT nanobody was overlaid with 4NBX.B-biotin103 WT traces as reference. (B). SPR-measured binding profile of v186_Fr against mSA_{WT} and mSA_{S27A} indicates that the improved binding affinity and kinetics in v149 are preserved. Data from one of the triplicates is shown here, and data from the other two replicates is in Fig. S5. (C). Structural representation of nanobody amino acid position 12 and 37 before and after framework redesign. v186 is colored as green, and v186_Fr is colored as cyan. Additional H-bonds introduced by F37Y with CDR3 residues are shown as dashes, while the relevant CDR3 residues are also shown in both v186 and v186_Fr models.

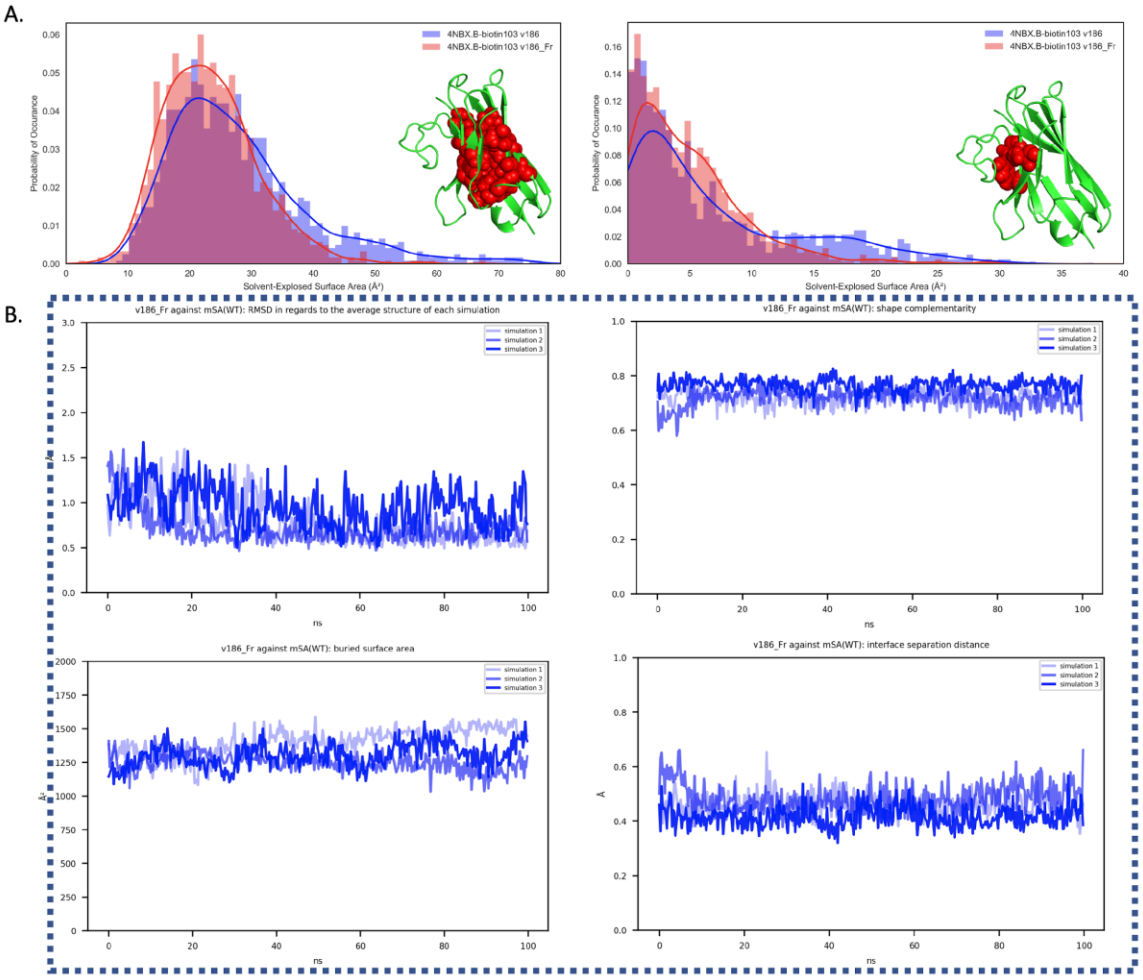


Figure 4. MD simulation reveals design flaws and validates design success. (A) MD simulation revealed possible origins for the improved monomeric stability of 4NBX.B-biotin103 v186_Fr compared to v186. Here shows the analysis of the solvent-accessible area for the selected hydrophobic residues of v186 and v186_Fr from 100ns MD simulations performed in triplicates. The selected residues are presented as spheres in the nanobody models shown on both panels. The observed distributions of the solvent accessible area for the selected residues from the 3X simulations of v186 and v186_fr are plotted into 80 bins along the x-axis (bars) with respective kernel density estimation (lines). Left

panel: analysis of the hydrophobic core residues. Right panel: analysis of the CDR3-shielded residues. (B). Analysis of the interaction interface in the triplicate MD simulations of 4NBX.B-biotin103 v186_Fr against mSA_{WT}. Traces from simulation replicates are plotted on top of each other along the 100ns timescales. Changes of whole-structure RMSD, interface shape complementarity, buried surface area, and interface separation distance along the time trajectories are plotted.

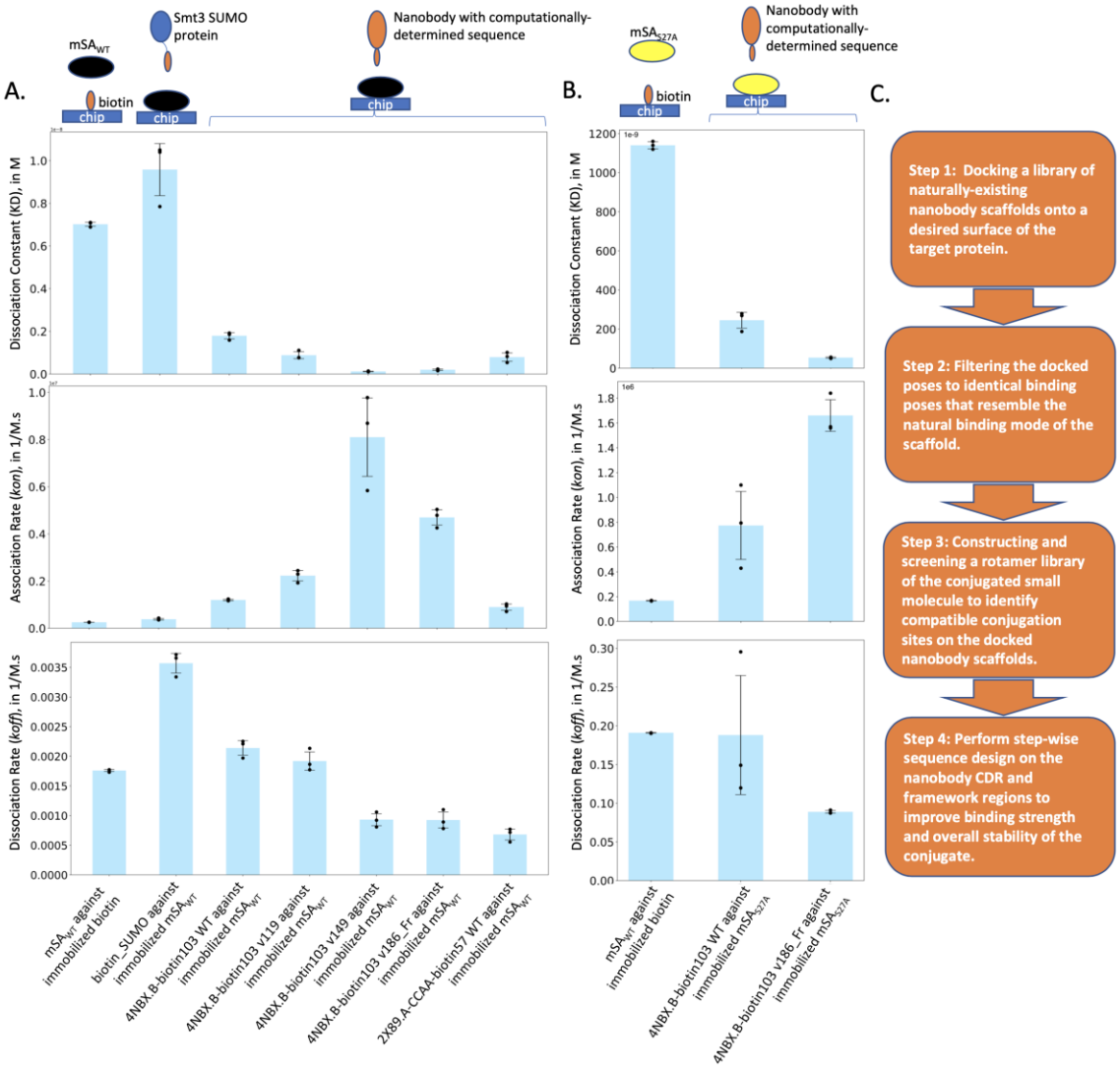


Figure 5: Summary of the design results for mSA-targeting CDRexAbs and the overall design workflow. (A-B). Summary of SPR-measured binding affinity and kinetics of nanobody-biotin conjugates and controls against mSA$_{WT}$ and mSA$_{S27A}$. Experimental designs for each SPR binding experiment are depicted by the cartoon above the lanes. Blue square: SPR cheap for immobilization. Spheres: molecules that are immobilized (attached to chip) or flew through (floating above the chip) during binding experiments. Different molecules are

represented by different colors. Individual data points represent measurements from biological triplicates. Error bars represent standard deviation. (C). Summary and proposal of a general design workflow for synergistically-binding nanobody-small molecule conjugates.

Supplementary Figures and Tables:

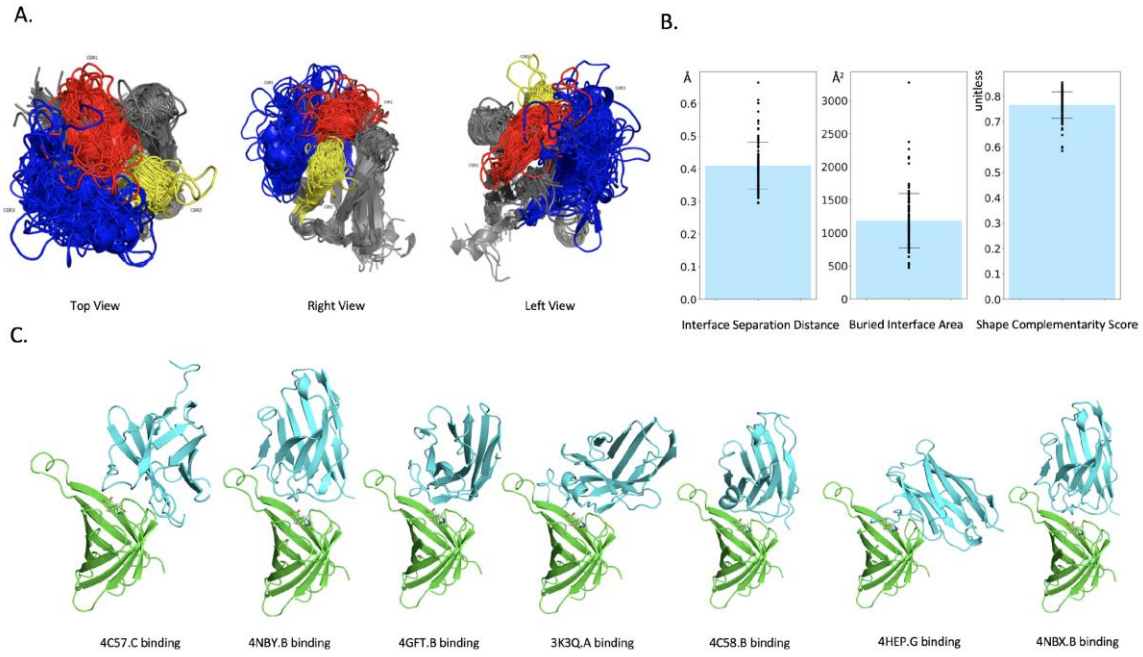


Figure S1: Searching for optimal CDR conformation against monomeric streptavidin model. (A). Whole structural-alignment of 154 curated PDB nanobody scaffolds with diverse CDR conformations and sequences. (B). Interface statistics of naturally occurring nanobody-target complexes. Error bars represent standard deviations. (C). The final 7 docked poses of nanobody scaffolds that passed the filters selecting poses that most likely recapitulate the

natural binding modes of the corresponding nanobody scaffolds. Streptavidin S45A/T90A/D180A is colored green, and nanobody scaffold is colored cyan.

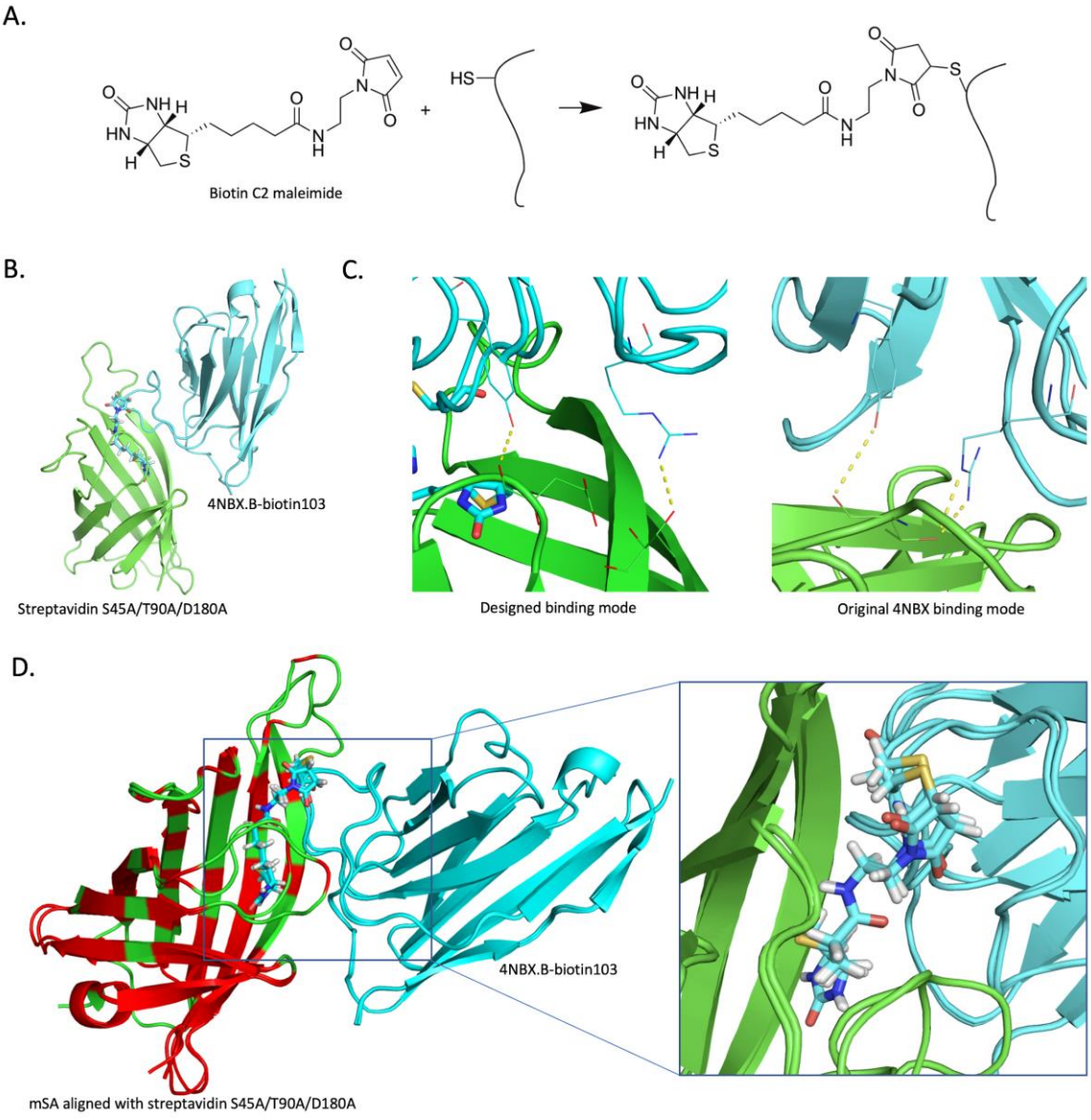


Figure S2: Identification of optimal conjugation strategy and finalized conjugate models. (A). Biotin conjugation was performed by biotin C2 maleimide with mutated cysteine residues. (B). Prepared structure of 4NBX.B-biotin103 in complex to streptavidin S45A/T90A/D180A. Streptavidin S45A/T90A/D180A is colored green, and nanobody scaffold is colored cyan. (C). The H-bond forming potential of Y112 and R27 in 4NBX.B nanobody was

predicted to be recapitulated in the designed binding pose with the streptavidin model. Streptavidin S45A/T90A/D180A is colored green, and nanobody scaffold is colored cyan. Biotin103 side chain is shown as stick. Y112 and R27 together with their predicted H-bond partners are shown as line. (D). Alignment results for prepared structures of 4NBX.B-biotin103 in complex with streptavidin S45A/T90A/D180A and mSA. Streptavidin models are colored green, and nanobody scaffolds are colored cyan. Residues that were identified by sequence alignment as pair-wise identical sequences are colored red. Biotin103 side chain from both models are shown as stick.

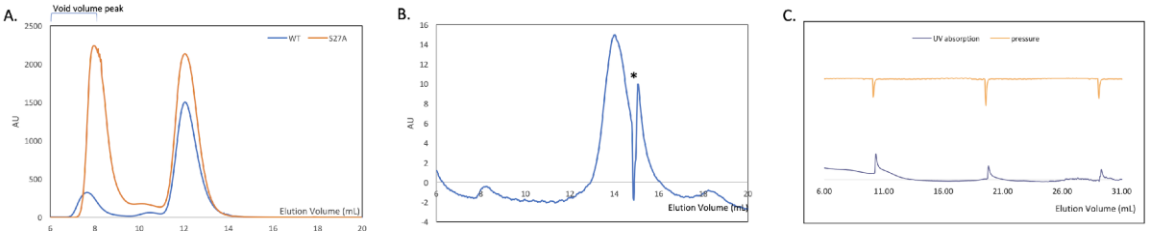


Figure S3: Additional Supporting SEC traces. A). SEC traces of mSA_{WT} and mSA_{S27A}. B). SEC rerun trace of collected monomeric fraction for 4NBX.B-biotin103 v186_Fr. * indicates peaks of sample-irrelevant instrument defect of the overall FPLC, please refer to section C for more details. C). Blank run of the FPLC to reveal the sample-irrelevant periodic peaks that were constantly observed in SEC data.

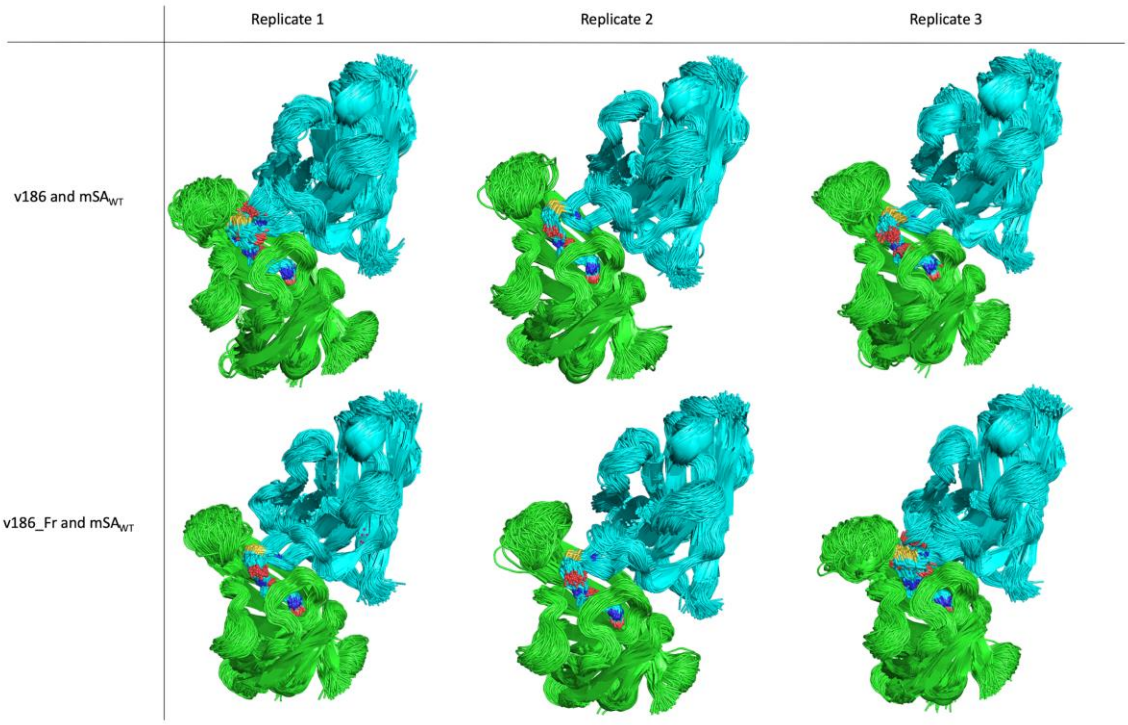


Figure S4. Summary of MD simulations performed for 4NBX.B-biotin103 v186 and v186_Fr against mSA$_{WT}$**.** For each simulation, 400 snapshots evenly spaced along the 100ns timescale are aligned together. green: mSA$_{WT}$. cyan: nanobody-biotin conjugates. The biotin103 “residue” is shown as stick.

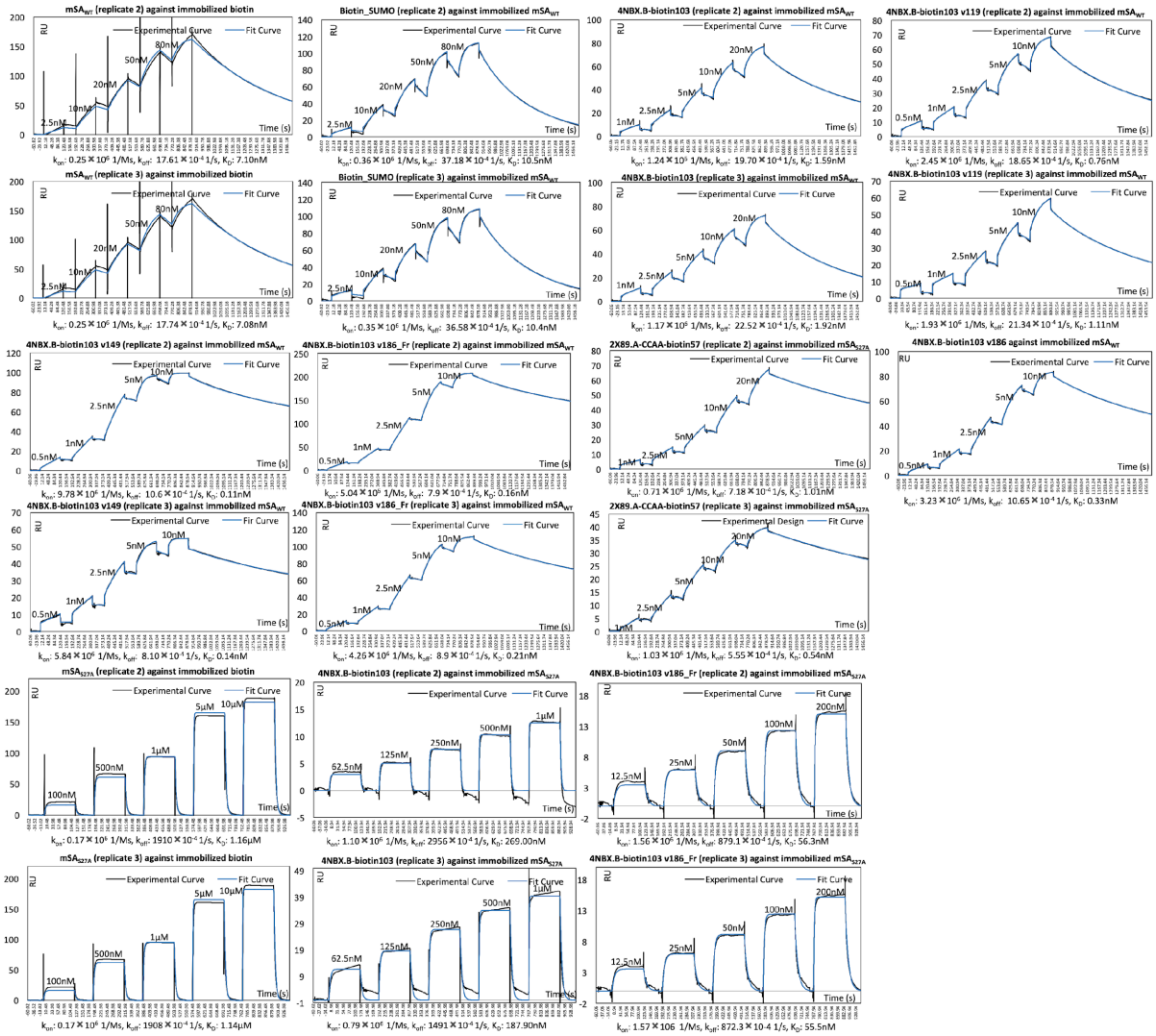


Figure S5: SPR measurements from the intermediate design variant 4NBX.B-biotin103 v186, and from additional biological replicates not shown in the main text but were included for affinity and kinetics estimation.

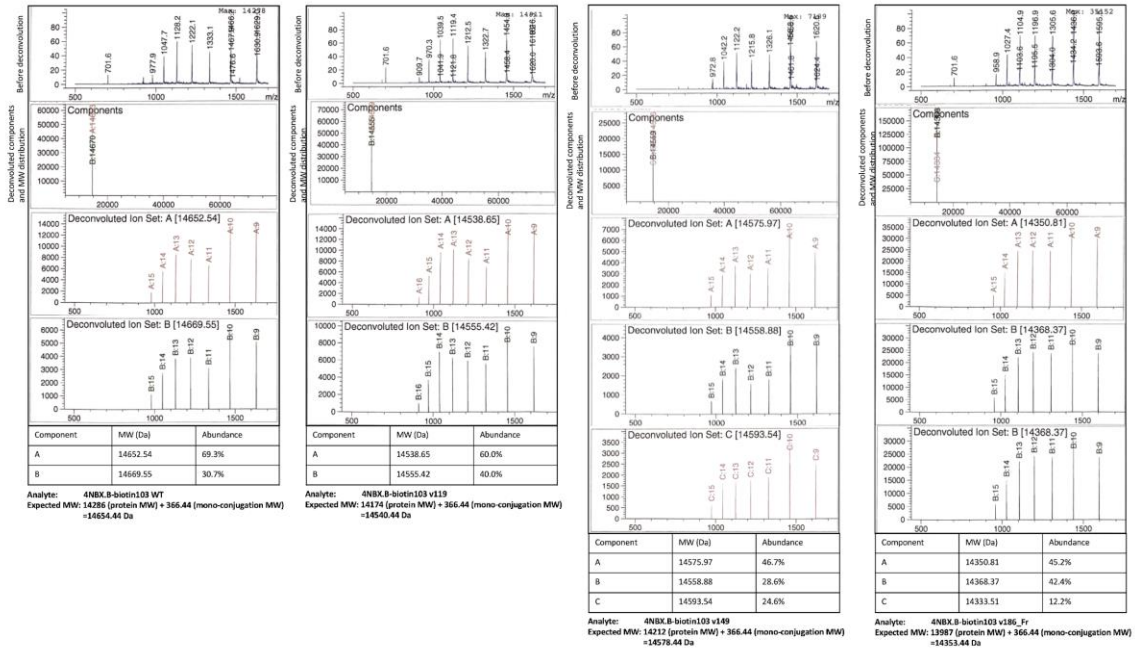


Figure S6: Intact-protein mass spectrometry (MS) confirmed mono-conjugated materials. MS deconvolution of nanobody-biotin conjugates only returned MWs within 20Da from expected MW of mono-conjugated materials.

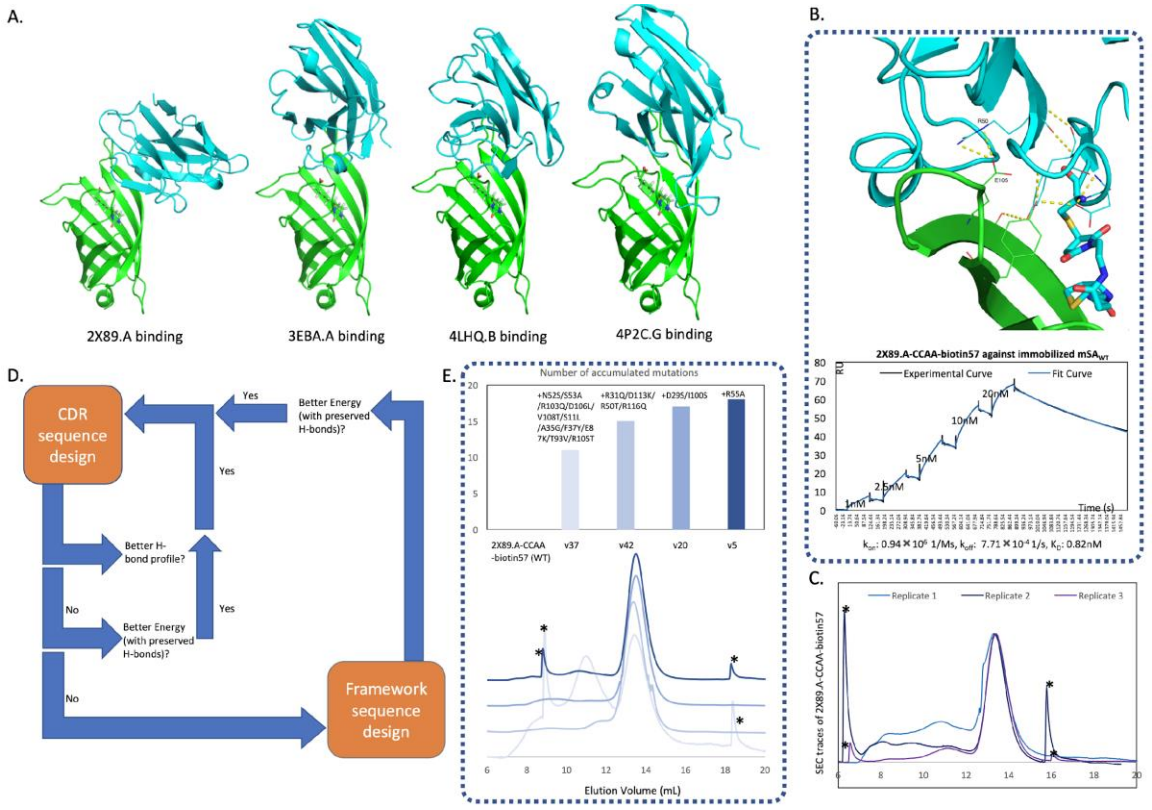


Figure S7: Designing and testing of a nanobody scaffold obtained by directly docking against mSA. (A). Finalized nanobody scaffolds and binding poses against mSA streptavidin. (B). Predicted H-bond formation profile and SPR binding curve for 2X89.A-CCAA-biotin57 WT against immobilized mSA$_{WT}$. Data from one of the triplicates is shown here, and data from the other two replicates is in Fig. S5. (C). Size-exclusion chromatography (SEC) traces of biological triplicates for 2X89.A-CCAA-biotin57, normalized by monomer peak height for better comparison of aggregates formation. * indicates peaks of sample-irrelevant instrument defect of the overall FPLC, please refer to figure S3C for more details. (D). A rudimentary sequence design pipeline that performs CDR and framework design in a step wise manner. (E). SEC traces and newly

accumulated mutations for sequence-designed variants of 2X89.A-CCAA-biotin57 conjugates. Peak of the monomeric fractions were normalized to identical heights. * indicates peaks of sample-irrelevant instrument defect of the overall FPLC, please refer to figure S5C for more details.

Supplementary Notes:

Computational design workflow for nanobody-biotin conjugates

Our in-house developed protein design suite TRIAD [58] together with third-party software PyMOL (Schrodinger) and OpenBabel [59] were used to perform protein computational design and analysis in this study. The detailed process and setup are described below.

A. Design process of 4NBX.B-derived nanobody-biotin conjugates

Searching optimal streptavidin-binding nanobody CDR conformations by docking and loop modeling

Monomeric streptavidin model S45A/T90A/D180A was prepared from crystal structure of wild type core tetrameric streptavidin (PDB ID: 1MK5). A single subunit was extracted and standardized by an in-house computational protein design suite TRIAD [58]. S45A/T90A/D128A substitution was then performed by the TRIAD sequence-design module. Initially, we attempted optimizing nanobody/streptavidin binding conformations by protein-protein docking followed by loop modeling of nanobody CDRs. We used a published nanobody structure (PDB ID: 5VNW, chain C) as the starting nanobody scaffold, with all CDR residues replaced to alanine by TRIAD sequence design, in a hope to avoid

sequence bias [60]. The nanobody scaffold was docked onto a set of manually selected surface residues surrounding the binding pocket of monomeric core streptavidin model S45A/T90A/D128A. Docking was performed by a previously-developed FFT-based docking program, and top 15 CDR binding poses were kept [61]. CDR loop modeling of each pose was then performed to attempt optimizing CDR conformation against the target by the TRIAD loop modeling module. Only the top one solution for each pose was kept, and no solution had reasonable CDR conformations beneficial for a good binding interface. The sequence of the core streptavidin with the triple-alanine mutations is attached below, and the residues selected as docking targets are highlighted in brackets:

EAGITGTWY(NQLGS)TFIVTAGADGALTGT(YEAAVGNAESRY)VLTGRY DSAPATDGSGTALGWTVA(WKNNYRNAHSA)ATWSGQYVGGAEARINTQ(WLL TSGTTEANAWKSTLVGHATFT)KVK.

<u>Searching optimal streptavidin-binding nanobody CDR conformations by docking native CDR sequences and conformations</u>

154 nanobodies that have higher-than-3Å resolution and continuous electron density with diverse target-binding CDR conformations were fetched from published nanobody/target complexes from PDB (please refer to section C below), and CDRs for each nanobody were subsequently annotated, following the nanobody CDR-mapping criteria in a previously-published study [34] but with softer edge cutoffs so that the sampled geometries of nanobody approaching are not too stringent. Each nanobody was docked against the previously-selected

binding pocket surface of the triple-mutation monomeric streptavidin model by the annotated CDRs, and top 15 poses for each docking trial were kept. Initially, we still performed alanine-replacement on all CDR residues before docking. However, several CDR sequence design trials on a docked structure output mostly small amino acids, indicating that the designability might be restricted in alanine-replaced docking complexes. Therefore, we decided to dock the 154 nanobody structures with native CDR sequences. The TRIAD surface-complementarity module was used to perform interface analysis of naturally-occurring nanobody-target complex structures to return respective statistics of interface separation distance, shape complementary, and buried interface area for future design reference (Fig. S1B) [62]. 2310 docked poses were generated, and then filtered by three selection steps to identify most realizable binding poses. Step one selected poses with interface separation distance, shape complementary score, and buried interface area within 1 standard deviation of naturally-occurring nanobody-target structures (Fig. S1B), returning 231 poses. Step two then selected nanobody poses that use >80% of the residues that participate in the original target-binding interface in the docked interface, and returned 31 poses. Identification of interface residues in both original PDB structures and docked structures was performed by a publically-available PyMOL script that selects interface residues by changes of solvent accessibility [63]. The last step selected poses that directly blocked the biotin binding pocket, and returned the final 7 poses (Fig. S1C). The degree of binding pocket blockage was reflected by using PyMOL to calculate the change of solvent accessible area of

the following selected residues that are a subset of the docking target residues and closely surround the biotin molecule (highlighted in brackets below with the rest of the streptavidin sequence shown as reference):

EAGITGTWYN(QLGS)TFIVTAGADGALTGTYEAA(VGNA)ESR(Y)VLTGRYDSA PATDGSGTALGWTVA(W)KNNYRN(AHS)AATWSGQYVGGAEARINTQ(W)L(L) T(S)GTTEANAWKSTLV(G)H(A)T(FT)KVK.

Rotamer library generation of biotin C2 maleimide side chain and conjugation plan determination

A rotamer library of the biotin-CH2-CH2-succinimide-S-CH3 “side chain” was constructed by OpenBabel [59], and virtually screened against CDR amino acid locations of the 7 poses to find the optimal conjugation sites. In the rotamer library, the biotin portion remains intact, while the CH2-CH2-succinimide-S-CH3 portion is diverse in torsion angles. Screening was done by measuring the distance between the terminal carbon of the rotamer and the Cβ of respective CDR residues, steric clash between the rotamer and the streptavidin, steric clash between the rotamer and the nanobody, and the angle of the rotamer terminal carbon approaching the respective attachment spot. Conjugation geometries that clashed with streptavidin by <1 unit, clashed with the nanobody by <15 units, approached the attachment spot by 100-120 degree, and were <1 Å away from the Cβ of screened conjugation sites were kept. Measurements were performed only against CDR residues that are originally alanine, which is similar to cysteine in size and usually does not perform important structural role, as we

hypothesized that making an alanine to cysteine mutation would less likely cause serious structural consequences to the nanobody. Alanine 103 on 4NBX.B was the only conjugation site that passed the above filters, and the rotamer that clashed least with both the streptavidin and nanobody was selected for further processing. To prepare the final conjugation structure, excess atoms were deleted and a bond was made between the Cβ of the biotin-CH2-CH2-succinimide-S-CH3 “side chain” to the Cα of 4NBX.B site 103. The conjugated structure of 4NBX.B-biotin against the S45A/T90A/D128A streptavidin was then relaxed by Biograf [67], with force restraints placed to maintain the biotin-streptavidin hydrogen bonds and torsion angles of the aliphatic arm portion of biotin. To prepare 4NBX.B-biotin103/mSA structure, the crystal structure of mSA (PDB ID: 4JNJ) was aligned to the modeled triple-mutation streptavidin structure, and 4NBX.B-biotin103 structure was relaxed by Biograf with the same force restraints [67].

Subsequent sequence design of 4NBX.B-biotin103 conjugates

CDR sequence design was first performed on the prepared 4NBX.B-biotin103/mSA$_{WT}$ model. CDR residues were determined following the CDR-mapping criteria in a published study [34], and specifically by first aligning the 4NBX.B model with an example nanobody structure in agreement with that criteria (PDB ID: 5VNW, chain C), and then selecting the corresponding CDR residues on 4NBX.B. Residues 27-34 were selected as CDR1, 47-60 were selected as CDR2, and 98-111 were selected as CDR3. Single-point mutation

scan was performed on each CDR location with reduced sets of amino acids that were reported to be frequently used in each corresponding CDR position [34]. The amino acid sets for each position were as follows: 27-WT/N/S/T/Y, 28/51-WT/I, 29-WT/S/F, 34-WT/M, 47-WT/F/L, 48-WT/V, 49/60/98-WT/A, 50/55-WT/A/G/S/T, 52-WT/A/D/G/Q/S/T, 54-WT/G, 56-WT/I/N/S/T, 57/58-WT/T, 59-WT/N/Y, 60/111-WT/Y, 109-WT/F/H/L/Y, and 30/31/32/33/53/99/100/101/102/104/105/106/107/108/110-WT/A/R/N/D/Q/E/G/H/I/L/K/F/P/S/T/W/Y/V. Position 103 with the attached biotin "side chain" was left un-designed and the coordinates for all atoms were left unchanged. Rosetta force field with covalent terms was used during the calculations [64]. Biograf-relaxed 4NBX.B-biotin103/mSA$_{WT}$ structure was used as structural input. During the design calculations, residues that were within 10 Å from the site under design calculation were allowed to repack. Each rotamer optimization for the site under design calculation was initiated by random rotamer configurations, and then repacked while the Cα backbone was allowed to relax through Cartesian minimization to optimize the structures with different sequence choices, which were then ranked given the energy scores of the corresponding modeled structures after iterative rotamer repacking and backbone relaxation. The chemical attributes of the biotin103 "side chain" were generated by TRIAD and then used during the design calculations of the energy scores. 10 runs with different random seeds were performed for each design calculation, and averaged to reflect the final amino acids preference for each site that underwent single-site mutation calculations. Mutation choices with lower Rosetta energy unit

than the WT amino acids were kept as designable mutations. Designability of each site was reflected by the sum of Rosetta energy unit differences of designable mutations from the corresponding WT amino acid choice. As the result, the output designable sites ranked by designability were as follows: 105, 109, 107, 104, 106, 32, 108, 31, and 56. Those sites were also separately grouped into two bins. Bin 1 contains sites that interact with the original target of 4NBX.B: sites 105, 104, 32, and 31, of which the order was ranked by designability. Bin 2 contains sites that do not interact with the original target of 4NBX.B: sites 109, 107, 106, 108, and 56, of which the order was ranked by designability.

Combinatorial designs with different choices of designable sites were performed in parallel. Combinatorial design 1 was performed on all the 9 designable sites. Combinatorial design 2 was performed on the designable sites in bin 1 only. Combinatorial design 3 was performed on the designable sites in bin 1 and bin 2, with an exception that for bin 2 sites, only mutation choices that are different from WT amino acid with >1 Rosetta energy units were used. Combinatorial design 4 was performed on the designable sites in bin 1 and bin 2, with an exception that for bin 2 sites, only the top-ranked mutation choice by energy score was used.

Combinatorial designs were performed with the same configurations as single-site mutation designs that were described before, with one difference: the output sequences from the 10 parallel design runs were re-ranked by threading the sampled sequences in each run individually onto the backbone of the input structure, followed by rotamer repacking and backbone Cartesian minimization.

The TRIAD-modeled structures and Rosetta energy scores of the top 20 sequences of the re-ranked sequences were used to evaluate design results.

Structures of the top 20 sequences for the 4 combinatorial designs were analyzed by PyMOL to identify intermolecular H-bonds between mSA_{WT} and 4NBX.B-biotin103 variants, and intramolecular H-bonds within 4NBX.B-biotin103 variants, using a publically-available PyMOL script that relies on the “find_pairs” command module of PyMOL [65]. The goal was to find sequences with improved overall energy score, new intermolecular H-bonds with mSA_{WT}, and intramolecular H-bond profile comparable to 4NBX.B-biotin103 WT, as the imbalance of forming new interactions with targets and keeping the structural integrity was a common reason behind the failure of designing protein-protein interactions [66]. All combinatorial designs output sequences with improved energy scores, but only combinatorial design 2 output the top 20 sequences with an average number of intermolecular H-bond higher than that of the 4NBX.B-biotin103 WT against mSA_{WT}. The top 20 sequences in combinatorial design 2 also had the highest average number of intramolecular H-bonds in the nanobodies among the 4 designs. Variant v119 was the top-ranked sequence in combinatorial design 2, and variant v149 had the highest number of predicted intermolecular H-bonds among the top 20 sequences (Table S1).

To improve the stability of v149, we hypothesized that a suitable method would optimize the protein structure while keeping the designed interactions contributed by R31/S32/A104/R105, the new anchoring spots, unchanged, without altering the target backbone structure too much. Therefore, we devised a sequential

design workflow that creates stepwise local structural optimizations that compensate for the mutations built-up in previous steps. As the result, subsequent rounds of CDR sequence designs were performed on v149. For each round, single-point mutation scan was first performed on the CDR residues using the identical setup as the first round of design. The calculation results were processed in the same way as the first round of design, with an exception that only mutation choices that are different from WT amino acid with >1 Rosetta energy units were kept for all further combinatorial design calculations. Next, skipping the sites that were mutated in previous rounds of design, four combinatorial designs were performed on the designable sites reported by the single-point mutation scan calculation. Combinatorial design 1 was performed on all designable sites with the reported designable mutation choices. Combinatorial design 2 was performed only the designable sites in bin 1 with the reported designable mutation choices. Combinatorial design 3 was performed on top 5 designable sites with the reported designable mutation choices. Combinatorial design 4 was performed on 5 designable sites ranked by designability, but with a bias on sites in bin 1. In other words, sites in bin 2 were not used unless the number of sites in bin 1 was smaller than 5. All combinatorial design calculations were performed and processed under the same setup as the first round of design.

The second round of design was performed using the output structure of v149 from the first round of design as input. No improvement in the number of intermolecular H-bond formation was observed for the outputs of all the 4

combinatorial designs, while the numbers of intramolecular H-bond were minimally different among the designs. Therefore, the design result with the biggest overall difference in energy score among the top 20 sequences against v149, combinatorial design 4, was chosen, and of which the sequence with the best energy score, v149 plus Y101L/R107F, was selected as the input structure for the third round of design. Combinatorial design 4 was performed with sites 27/59/101/107/110. In the third round, again no improvements in the number of intermolecular and intramolecular H-bonds were observed among the 4 combinatorial designs. So, the sequence with the best energy score, v186 (v149 plus Y101L/R107F/R56T/Y106K/D108A/Y110S), of combinatorial design 1 whose top 20 sequences showed biggest overall improvement in energy scores against the input sequence was selected. Combinatorial design 1 was performed with sites 27/29/56/106/108/110. A further round of CDR design was performed on v186 and all combinatorial design results returned sequences with worse energy score than v186.

Because v186 turned out to be even more prone to aggregation than v149, and based on MD simulation results of v149 against mSA_{WT}, we hypothesized that only mutating CDRs was not sufficient. Therefore, we proceeded to design the framework regions of v149. Because the frameworks of nanobodies are highly conserved [34], a suitable sets of amino acid choices and locations would be crucial for the design calculation. Because the previous CDR designs were based on a published summary of nanobody CDR sequence diversity, we referred to the framework sequence used in that study for

framework sequence design [34]. We aligned 4NBX.B-biotin103 v186 with chain C of 5VNW, identified framework sites where the two nanobodies differ, and performed a combinatorial design with the selected sites being one or the other amino acid choice. Site positions and sequence choices were as follows: 5-V/G, 12-A/V, 35-A/G, 37-F/Y, and 40-P/A. The v186 structure output by the third round of design was used as input, and the configurations and processing of design calculation were the same as the combinatorial CDR sequence designs described previously. The top-ranked sequence by energy score was v186_Fr (v186 plus A12V/F37Y) (Table S2). Framework design with identical amino acid sites, sequence choices, and calculation configurations was performed using v149 as input, and the design results were also reported in this study for comparison (Table S3).

As a comparison, we performed two rounds of CDR sequence design using the above-described configurations on v119. Combinatorial designs in both rounds failed to produce variants with new inter-molecular H-bond formation in the top 20 variants, so the combinatorial designs with best overall energy improvement than the input sequence were chosen, and the top-ranked sequences by energy score in those designs were used as input for further rounds of design and experimental testing. No improvements in kinetics and affinity were observed in these selected sequences (data not shown), in agreement with the unchanged inter-molecular H-bond profiles and with what we observed for v186 versus v149.

B. Design process of 2X89.A-derived nanobody-biotin conjugates

Docking, pose filtering, rotamer screening, and binding pose generation for 2X89.A-CCAA-biotin57 WT against mSA_{WT}

The 154 nanobody structures from PDB with native CDR sequences were docked against a manually-selected set of surface residues around the biotin-binding pocket of mSA, in the same way as described in part A. The amino acids being docked against are highlighted below by brackets with the rest of mSA sequence shown for reference:

GAEAGITGTWYN(QSG)STFTVTAGADGNLTGQY(ENRAQGTG)C(QNSP)YTLTGRYNGTKLEWRVEWN(NSTENCH)SRTEWRGQYQGGAEARINTQWNLT(YEGGSGPATEQGQDT)FTKVK.

Filtering procedures of the docked poses were also identical to those introduced in part A, with an exception that the following residues (highlighted by brackets) were selected as the target for binding pocket blockage filter:

GAEAGITGTWYN(QS)GSTFTVTAGADGNLTGQYENRAQGTGCQNSPYTLTGRYNGTKLEWRVEWNNSTENCHSRTEWRGQYQGGAEARINTQWNLT(YEGGSGPATEQGQDT)FTKVK.

6 poses that respectively comprised nanobodies 2X89.A, 3EBA.A, 4LHQ.B, 4OCL.C, 3V0A.C, and 4P2C.G passed the series of filters. Visual inspection of the poses revealed that the 4OCL.C and 3V0A.C binding poses showed significant contacts that are mediated by nanobody residues outside of the CDRs, so the corresponding two poses were discarded since these binding

modes were potentially unrealistic. As the result, 4 final binding poses were kept for the evaluation of optimal conjugation sites (Fig. S7A).

The biotin-CH2-CH2-succinimide-S-CH3 rotamer library built in part A was used again for rotamer screening. Because the design process of the 4NBX.B-derived CDRexAbs demonstrated that our design capability allowed structural stability of the conjugates to be designed after binding synergy was designed, we did not put much emphasis on preserving structural integrity in the early stage of the design process of 2X89.A-derived CDRexAbs. Therefore, instead of screening only against alanine CDR residues, all CDR residues were screened by measuring the distance between the terminal carbon of the rotamer and the Cβ of respective CDR residues, steric clash between the rotamer and the streptavidin, steric clash between the rotamer and both proteins, and the angle of the rotamer terminal carbon approaching the respective attachment spot. Conjugation geometries that clashed with streptavidin by <0.5 unit, clashed with both proteins by <10 units, approached the attachment spot by 100-120 degree, and were <2 Å away from the Cβ of screened conjugation sites were kept. Only one rotamer that was screened against I57 of 2X89.A passed the filter. The final conjugation structure was prepared by Biograf, under the same parameters as described in part A [67].

To remove the intra-CDR disulfide bond in the Biograf-relaxed structure, C33A/C104A mutations were introduced by TRIAD sequence design module to create the finalized model of 2X89.A-CCAA-biotin57/mSA for further sequence design optimization.

Subsequent sequence design of 2X89.A-CCAA-biotin57 conjugates

Summarizing the experience from the design process of the 4NBX.B-biotin103 conjugates, we gained the following insights into the sequence design principles of CDRexAbs:

1. Performing sequential rounds of design on limited sets of amino acid sites and choices that are recommended by iterative energetic and structural analysis allows functionally-improved CDRexAb mutants to be discovered without experimentally screening a large set of sequences.

2. New intermolecular interactions between the nanobody scaffold and the target can be engineered first before further mutations are introduced to optimize the structural integrity of the conjugates.

3. Simply mutating CDR residues is not sufficient for structural optimization of the conjugates, and mutated CDR residues likely need accommodation by introducing mutations in the β-barrel framework region.

Based on the above principles, we established a rudimentary sequence design pipeline, and tested it on 2X89.A-CCAA-biotin57 to create mutants that are less prone to aggregation. The pipeline is detailed below (Fig. S7D):

1. The pipeline performs sequential rounds of sequence design that is either restricted on CDR residues or framework residues. The first round of design is performed on CDR residues. Residues that are mutated in previous rounds are kept from mutation in further rounds.

2. H-bonds are still the only intermolecular interactions that are explicitly evaluated after design calculations and biased towards for sequence selection,

but other types of interactions can be evaluated if they are deemed to be crucial for specific scenarios.

3. CDR sequence design follows the procedures of designing the CDR loops of v149, as described in part A, with one exception: besides the four combinatorial designs, it is optional that two additional combinatorial designs can be performed in parallel, respectively on all identified sites and amino acid choices in bin 2, and on the top 5 designable sites with a bias on sites in bin 2. All combinatorial designs are evaluated together as described in part A.

4. Framework design follows the procedures of designing the framework of v186, as described in part A.

5. To evaluate the design results of CDR design, the following steps are used:

a). Sequences that showed worse energy score than the immediate parent sequence are discarded.

b). Sequences with the number of intermolecular H-bonds lower than the immediate parent sequence are discarded.

c). If no sequences survived filters a and b, perform framework design on the immediate parent sequence.

d). For sequences pass the filters, the sequence that has the highest number of intermolecular H-bonds and the best energy score among sequences that share the same number of intermolecular H-bonds is kept as input for the next round of CDR design.

6. To evaluate the design results of framework design, the following steps are used:

a). Sequences that showed worse energy score than the immediate parent sequence are discarded.

b). Sequences with the number of intermolecular H-bonds lower than the immediate parent sequence are discarded.

c). Sequences with the number of nanobody intramolecular H-bonds lower than the immediate parent by >1 are discarded.

d). If no sequences survived filters a-c, design fails.

e). For sequences pass the filters, the sequence that has the highest energy score is kept as input for the next round of CDR design.

The outputs after rounds 3, 5, 6, and 7, which are variants v37, v42, v20, and v5 were selected for experimental testing.

References:

1. D. C. Swinney, Biochemical mechanisms of drug action: what does it take for success? *Nature Reviews Drug Discovery* **3**, 801–808 (2004).

2. J. K. H. Liu, The history of monoclonal antibody development – Progress, remaining challenges and future innovations. *Annals of Medicine and Surgery* **3**, 113–116 (2014).

3. A. L. Hopkins, C. R. Groom, The druggable genome. *Nature Reviews Drug Discovery* **1**, 727–730 (2002).

4. G. L. Verdine, L. D. Walensky, The Challenge of Drugging Undruggable Targets in Cancer: Lessons Learned from Targeting BCL-2 Family Members. *Clin Cancer Res* **13**, 7264–7270 (2007).

5. M. A. Ayoub, *et al.*, Antibodies targeting G protein-coupled receptors: Recent advances and therapeutic challenges. *MAbs* **9**, 735–741 (2017).

6. R. B. Dodd, T. Wilkinson, D. J. Schofield, Therapeutic Monoclonal Antibodies to Complex Membrane Protein Targets: Antigen Generation and Antibody Discovery Strategies. *BioDrugs* **32**, 339–355 (2018).

7. E. Valeur, *et al.*, New Modalities for Challenging Targets in Drug Discovery. *Angewandte Chemie International Edition* **56**, 10294–10323 (2017).

8. G. D. L. Phillips, *et al.*, Targeting HER2-Positive Breast Cancer with Trastuzumab-DM1, an Antibody–Cytotoxic Drug Conjugate. *Cancer Res* **68**, 9280–9290 (2008).

9. Y. Feng, *et al.*, Conjugates of Small Molecule Drugs with Antibodies and Other Proteins. *Biomedicines* **2**, 1–13 (2014).

10. W. D. Hedrich, T. E. Fandy, H. M. Ashour, H. Wang, H. E. Hassan, Antibody–Drug Conjugates: Pharmacokinetic/Pharmacodynamic Modeling, Preclinical Characterization, Clinical Studies, and Lessons Learned. *Clinical Pharmacokinetics* **57**, 687–703 (2017).

11. D. Su, *et al.*, Modulating Antibody–Drug Conjugate Payload Metabolism by Conjugation Site and Linker Modification. *Bioconjugate Chemistry* **29**, 1155–1167 (2018).

12. B.-Q. Shen, *et al.*, Conjugation site modulates the in vivo stability and therapeutic activity of antibody-drug conjugates. *Nature Biotechnology* **30**, 184–189 (2012).

13. Y. Wang, *et al.*, Peptide–Drug Conjugates as Effective Prodrug Strategies for Targeted Delivery. *Adv Drug Deliv Rev* **110–111**, 112–126 (2017).

14. R. Y. Zhao, *et al.*, Synthesis and Evaluation of Hydrophilic Linkers for Antibody–Maytansinoid Conjugates. *J. Med. Chem.* **54**, 3606–3623 (2011).

15. A. C. Cheng, *et al.*, Structure-guided Discovery of Dual-recognition Chemibodies. *Scientific Reports* **8**, 7570 (2018).

16. J. Tang, *et al.*, An Inhibitory Antibody against Dipeptidyl Peptidase IV Improves Glucose Tolerance in Vivo. *J. Biol. Chem.* **288**, 1307–1316 (2013).

17. S. Muyldermans, Nanobodies: Natural Single-Domain Antibodies. *Annual Review of Biochemistry* **82**, 775–797 (2013).

18. I. Hmila, *et al.*, VHH, bivalent domains and chimeric Heavy chain-only antibodies with high neutralizing efficacy for scorpion toxin AahI′. *Molecular Immunology* **45**, 3847–3856 (2008).

19. C. I. Webster, *et al.*, Brain penetration, target engagement, and disposition of the blood–brain barrier-crossing bispecific antibody antagonist of metabotropic glutamate receptor type 1. *The FASEB Journal* **30**, 1927–1940 (2016).

20. S. Fischman, Y. Ofran, Computational design of antibodies. *Current Opinion in Structural Biology* **51**, 156–162 (2018).

21. J. C. Almagro, *et al.*, Second antibody modeling assessment (AMA-II). *Proteins: Structure, Function, and Bioinformatics* **82**, 1553–1562 (2014).

22. L. T. Dang, *et al.*, Receptor subtype discrimination using extensive shape complementary designed interfaces. *Nat Struct Mol Biol* **26**, 407–414 (2019).

23. K. Kundert, T. Kortemme, Computational design of structured loops for new protein functions. *Biological Chemistry* **400**, 275–288 (2019).

24. S. M. Lewis, B. A. Kuhlman, Anchored Design of Protein-Protein Interfaces. *PLoS ONE* **6**, e20872 (2011).

25. S. J. Fleishman, *et al.*, Computational Design of Proteins Targeting the Conserved Stem Region of Influenza Hemagglutinin. *Science* **332**, 816–821 (2011).

26. X. Liu, *et al.*, Computational design of an epitope-specific Keap1 binding antibody using hotspot residues grafting and CDR loop swapping. *Scientific Reports* **7**, 41306 (2017).

27. M. H. Qureshi, J. C. Yeung, S.-C. Wu, S.-L. Wong, Development and Characterization of a Series of Soluble Tetrameric and Monomeric Streptavidin Muteins with Differential Biotin Binding Affinities. *J. Biol. Chem.* **276**, 46422–46428 (2001).

28. K. H. Lim, H. Huang, A. Pralle, S. Park, Stable, high-affinity streptavidin monomer for protein labeling and monovalent biotin detection. *Biotechnology and Bioengineering* **110**, 57–67 (2013).

29. G. Nimrod, *et al.*, Computational Design of Epitope-Specific Functional Antibodies. *Cell Reports* **25**, 2121-2131.e5 (2018).

30. N. Jain, S. W. Smith, S. Ghone, B. Tomczuk, Current ADC Linker Chemistry. *Pharm Res* **32**, 3526–3540 (2015).

31. T. Murase, *et al.*, Structural basis for antibody recognition in the receptor-binding domains of toxins A and B from Clostridium difficile. *J. Biol. Chem.* **289**, 2331–2343 (2014).

32. W. Sheng, X. Liao, Solution structure of a yeast ubiquitin-like protein Smt3: The role of structurally less defined sequences in protein–protein recognitions. *Protein Sci* **11**, 1482–1491 (2002).

33. F. Liu, J. Z. H. Zhang, Y. Mei, The origin of the cooperativity in the streptavidin-biotin system: A computational investigation through molecular dynamics simulations. *Sci Rep* **6** (2016).

34. C. McMahon, *et al.*, Yeast surface display platform for rapid discovery of conformationally selective nanobodies. *Nature Structural & Molecular Biology* **25**, 289–296 (2018).

35. L. S. Mitchell, L. J. Colwell, Analysis of nanobody paratopes reveals greater diversity than classical antibodies. *Protein Eng Des Sel* **31**, 267–275 (2018).

36. H. K. Privett, *et al.*, Iterative approach to computational enzyme design. *PNAS* **109**, 3790–3795 (2012).

37. A. Nisthal, C. Y. Wang, M. L. Ary, S. L. Mayo, Protein stability engineering insights revealed by domain-wide comprehensive mutagenesis. *PNAS* **116**, 16367–16377 (2019).

38. L. N. Makley, J. E. Gestwicki, Expanding the Number of 'Druggable' Targets: Non-Enzymes and Protein-Protein Interactions. *Chemical Biology & Drug Design* **81**, 22–32 (2013).

39. D. H. Nam, C. Rodriguez, A. G. Remacle, A. Y. Strongin, X. Ge, Active-site MMP-selective antibody inhibitors discovered from convex paratope synthetic libraries. *Proc. Natl. Acad. Sci. U.S.A.* **113**, 14970–14975 (2016).

40. D. Baran, *et al.*, Principles for computational design of binding antibodies. *Proceedings of the National Academy of Sciences*, 201707171 (2017).

41. C. A. Voigt, C. Martinez, Z.-G. Wang, S. L. Mayo, F. H. Arnold, Protein building blocks preserved by recombination. *Nature Structural Biology* **9**, 553–558 (2002).

42. S. E. Boyken, *et al.*, De novo design of protein homo-oligomers with modular hydrogen-bond network–mediated specificity. *Science* **352**, 680–687 (2016).

43. B. D. Allen, A. Nisthal, S. L. Mayo, Experimental library screening demonstrates the successful application of computational protein design to large structural ensembles. *PNAS* **107**, 19838–19843 (2010).

44. D. J. Mandell, E. A. Coutsias, T. Kortemme, Sub-angstrom accuracy in protein loop reconstruction by robotics-inspired conformational sampling. *Nature Methods* **6**, 551–552 (2009).

45. D. G. Gibson, *et al.*, Enzymatic assembly of DNA molecules up to several hundred kilobases. *Nat. Methods* **6**, 343–345 (2009).

46. D. Demeestere, *et al.*, Development and Validation of a Small Single-domain Antibody That Effectively Inhibits Matrix Metalloproteinase 8. *Molecular Therapy* **24**, 890–902 (2016).

47. K. Hand, M. C. Wilkinson, J. Madine, Isolation and purification of recombinant immunoglobulin light chain variable domains from the periplasmic space of Escherichia coli. *PLOS ONE* **13**, e0206167 (2018).

48. S. B. Hansen, N. S. Laursen, G. R. Andersen, K. R. Andersen, Introducing site-specific cysteines into nanobodies for mercury labelling allows de novo phasing of their crystal structures. *Acta Crystallographica Section D Structural Biology* **73**, 804–813 (2017).

49. T. Pleiner, *et al.*, Nanobodies: site-specific labeling for super-resolution imaging, rapid epitope-mapping and native protein complex isolation. *eLife* **4**, e11349 (2015)

50. M. L. B. Magalhães, *et al.*, Evolved streptavidin mutants reveal key role of loop residue in high-affinity binding. *Protein Sci* **20**, 1145–1154 (2011).

51. S. Doerr, M. J. Harvey, F. Noé, G. De Fabritiis, HTMD: High-Throughput Molecular Dynamics for Molecular Discovery. *J. Chem. Theory Comput.* **12**, 1845–1852 (2016).

52. W. L. Jorgensen, J. Chandrasekhar, J. D. Madura, R. W. Impey, M. L. Klein, Comparison of simple potential functions for simulating liquid water. *J. Chem. Phys.* **79**, 926–935 (1983).

53. J. A. Maier, *et al.*, ff14SB: Improving the Accuracy of Protein Side Chain and Backbone Parameters from ff99SB. *J. Chem. Theory Comput.* **11**, 3696–3713 (2015).

54. J. Wang, W. Wang, P. A. Kollman, D. A. Case, Automatic atom type and bond type perception in molecular mechanical calculations. *Journal of Molecular Graphics and Modelling* **25**, 247–260 (2006).

55. Gaussian 09, Revision A.02, M. J. Frisch, *et al.*, Gaussian, Inc., Wallingford CT, 2016.

56. R. T. McGibbon, *et al.*, MDTraj: A Modern Open Library for the Analysis of Molecular Dynamics Trajectories. *Biophysical Journal* **109**, 1528–1532 (2015).

57. W. Humphrey, A. Dalke, K. Schulten, VMD: Visual molecular dynamics. *Journal of Molecular Graphics* **14**, 33–38 (1996).

58. Protabit LLC - Triad. Available at: https://triad.protabit.com

59. N. M. O'Boyle, *et al.*, Open Babel: An open chemical toolbox. *Journal of Cheminformatics* **3**, 33 (2011).

60. Y. Mou, P.-S. Huang, F.-C. Hsu, S.-J. Huang, S. L. Mayo, Computational design and experimental verification of a symmetric protein homodimer. *Proc. Natl. Acad. Sci. U.S.A.* **112**, 10714–10719 (2015).

61. P.-S. Huang, J. J. Love, S. L. Mayo, Adaptation of a fast Fourier transform-based docking algorithm for protein design. *J Comput Chem* **26**, 1222–1232 (2005).

62. M. C. Lawrence, P. M. Colman, Shape complementarity at protein/protein interfaces. *J. Mol. Biol.* **234**, 946–950 (1993).

63. PyMOL Wiki – InterfaceResidues. Available at: https://pymolwiki.org/index.php/InterfaceResidues. [Accessed May 24, 2018]

64. R. F. Alford, *et al.*, The Rosetta All-Atom Energy Function for Macromolecular Modeling and Design. *J. Chem. Theory Comput.* **13**, 3031–3048 (2017).

65. Queen's University Protein Function Discovery and Department of Biomedical and Molecular Sciences Molecular Modeling and Crystallographic Computing Facility - PyMOL Script repository. Available at: http://pldserver1.biochem.queensu.ca/~rlc/work/pymol/. [Accessed November 1, 2018]

66. P.-S. Huang, J. J. Love, S. L. Mayo, A de novo designed protein–protein interface. *Protein Science* **16**, 2770–2774 (2007).

67. S. L. Mayo, B. D. Olafson, W. A. Goddard, DREIDING: a generic force field for molecular simulations. *J. Phys. Chem.* **94**, 8897–8909 (1990).

This is the end of the CDRxAb Manuscript

2.2 ncAA containing insulins: Design insights via molecular dynamics

Recombinant insulin is a crucial therapeutic protein which is administered to millions of patients affected by diabetes mellitus (WHO, 2016). In normal function, insulin is produced and secreted from beta cells of the pancreas, and binds to the insulin receptor to induce the cellular uptake of blood glucose (Haeusler et al., 2018). Patients with diabetes have decreased pancreatic output of insulin, which must be supplemented by therapies to enhance insulin signaling, or in some cases turn to insulin replacement therapy via regular self-administered subcutaneous injections.

Over the past fifty years of recombinant insulin use, significant effort has gone into creating stable formulations, as well as variations which have modified pharmacokinetics, particularly fast-acting insulins. Insulin exists as a hexamer bound to zinc and phenolic ligands, (Derewenda et al., 1989) and upon injection into subcutaneous tissue, dissociates from a hexamer to a dimer, before finally crossing the capillary membrane as a monomer. The rate limiting step between the injection of inactive hexamer and insulin monomers reaching their intended receptors is the dissociation of hexamer to monomer. A typical paradigm for managing diabetes is reactive management which involves patients measuring their blood glucose after meals and administering a calculated dose of insulin. In this case, it is desirable that insulin enacts its biological effects as rapidly as possible. Because of this, significant engineering work has been done to create insulin variants which are destabilized in the hexameric form through mutagenesis.

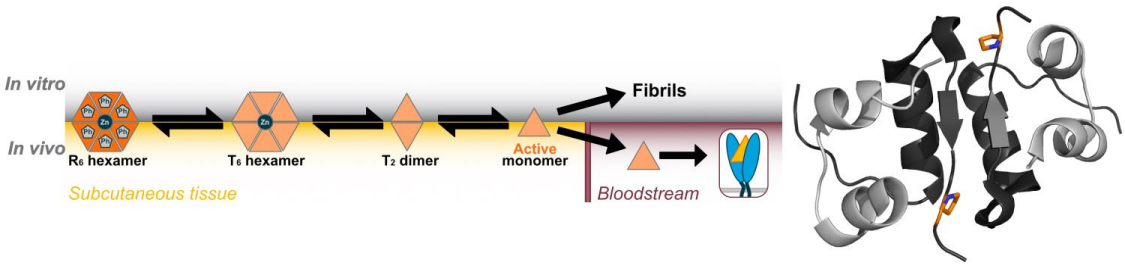


Figure 9: Insulin oligomeric states. Pharmaceutical preparations of insulin exist as a hexamer bound to zinc and phenolic ligands. Upon injection into subcutaneous tissue, hexamers dissociate as to dimers, before finally crossing the capillary membrane as a monomer. Image used with permission from Stephanie L. Breunig (SLB).

Insulin lispro, the first marketed fast acting insulin (Holleman et al., 1997), disfavors the association of subunits into higher order oligomers by switching Proline B28 and Lysine B29 near the C-terminus of the B-chain (Bakaysa et al., 1996). By moving lysine further from the terminus of the chain, additional backbone flexibility is imparted compared to the constrained phi and psi angles of proline. This is believed to remove hydrophobic packing interactions at the dimer interface (Brems et al., 1992). As proline is unique in its restricted conformational range, an intriguing approach to further modify insulin is through the introduction of non-canonical proline variants.

The Tirrell lab has done pioneering work for incorporating novel non-canonical prolines and other amino acids (Lieblich et al., 2018, Fang et al., 2018, 2019), including incorporating nine variants into both insulin and insulin lispro, and for each variant, characterizing dissociation rates and fibrillation (Figure 10).

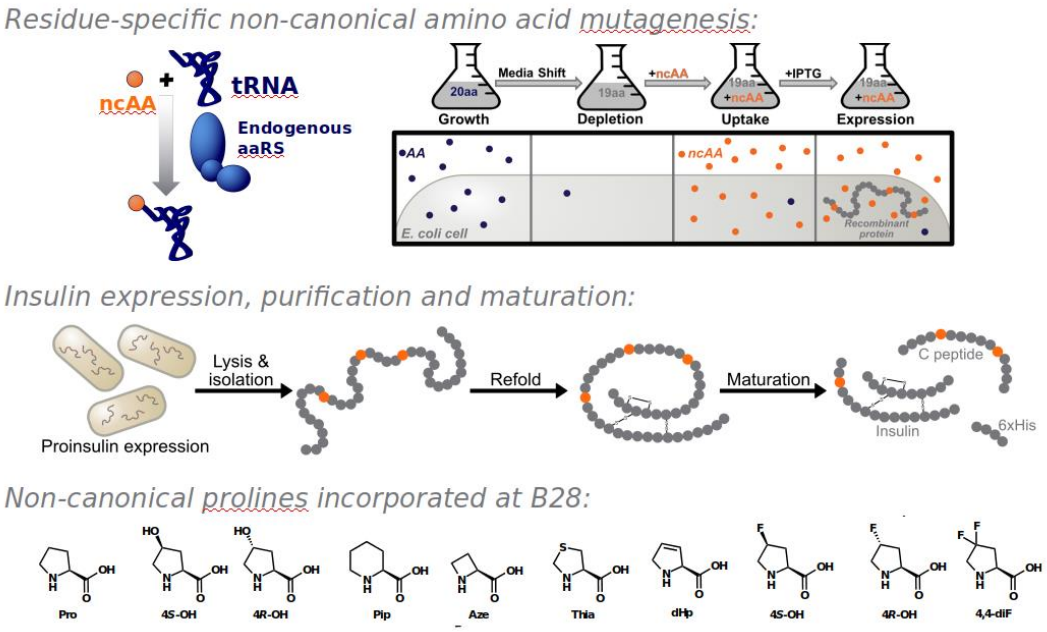
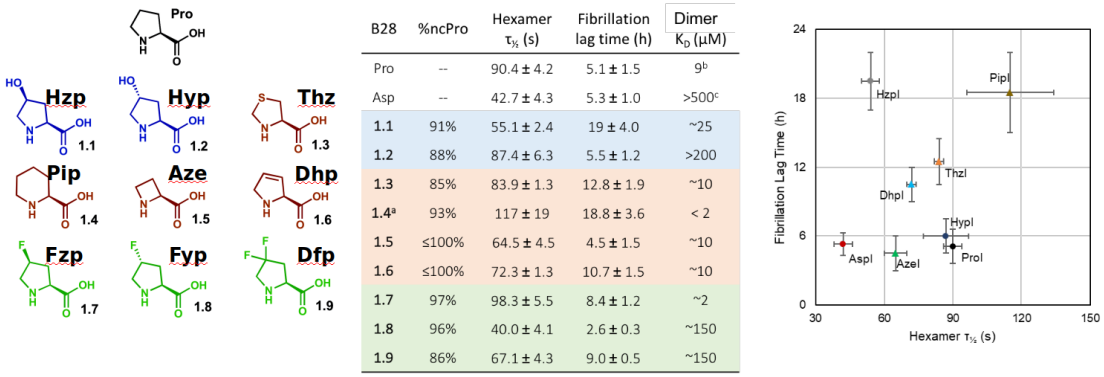


Figure 10: Incorporation of non-canonical prolines into insulin and insulin lispro. Courtesy of SLB, used with permission.

Containing a single, key proline residue, insulin is an ideal system to study the biophysical effects of non-canonical proline variants. We have built structures of all oligomeric (R6, T6, T2, and monomer) forms incorporating each non-canonical proline, including wild-type, insulin, and lispro variants. Recognizing the need to develop rapid acting insulins which are simultaneously resistant to fibrillation, we use all-atom molecular dynamics to understand the variabile half-lifes and fibrillation of insulin analogs containing these ten non-canonical prolines. Through the simulation and analysis of more than 50 distinct systems, we have arrived at the hypothesis that ring puckering contributes to both the fibrillation of insulin and the proposed additional novel non-canonical prolines to synthesize and incorporate.



B28	%ncPro	Hexamer $\tau_{½}$ (s)	Fibrillation lag time (h)	Dimer K_D (μM)
Pro	--	90.4 ± 4.2	5.1 ± 1.5	9[b]
Asp	--	42.7 ± 4.3	5.3 ± 1.0	>500[c]
1.1	91%	55.1 ± 2.4	19 ± 4.0	~25
1.2	88%	87.4 ± 6.3	5.5 ± 1.2	>200
1.3	85%	83.9 ± 1.3	12.8 ± 1.9	~10
1.4[a]	93%	117 ± 19	18.8 ± 3.6	< 2
1.5	≤100%	64.5 ± 4.5	4.5 ± 1.5	~10
1.6	≤100%	72.3 ± 1.3	10.7 ± 1.5	~10
1.7	97%	98.3 ± 5.5	8.4 ± 1.2	~2
1.8	96%	40.0 ± 4.1	2.6 ± 0.3	~150
1.9	86%	67.1 ± 4.3	9.0 ± 0.5	~150

Figure 11: Non-canonical prolines integrated into insulin and characterized From Fang et al. (2018) and Lieblich et al. (2018).

In order to conduct molecular dynamics studies of the prolines containing the non-canonical amino acids engineered by the Tirrell lab, we first set out to derive force field parameters for each proline. For each variant, MarvinSketch was used to generate an initial conformer, with three-dimensional coordinates refined in accordance with the Dreiding force field (Mayo et al., 1990). These conformers included N-methyl and acetyl caps. Gaussian 09 (Frisch et al., 2016) was used to optimize geometry and fit RESP charges[1], and Antechamber (Wong et al., 2006) was used to complete the remaining fields of Amber ff14SB force field (Maier et al., 2015) library files.

Starting with a crystal structure of hexameric insulin determined by the Tirrell lab, we performed in silico mutagenesis using xleap (Ambertools) and our previously generated library files to generate hexamers of insulin containing each non-canonical proline. These hexamers were subsequently split into dimers and

[1] Note to anyone following this path: there is a silent bug in G09 Rev B that complicates this step. Please see http://www.ub.edu/cbdd/?q=content/gaussian09-bug-fix for a patch.

monomers in silico to further allow for study of the lower order oligomers. Although this may not be as ideal as starting from crystal structures of an insulin dimer or monomer, crystal structures of such oligomeric forms are not available. This process was also recreated using an insulin lispro crystal structure as a starting point to generate variants containing each non-canonical proline.

All systems except monomers were simulated with three independent replicates, with six replicates performed for monomers. Each simulation includes independent minimization, equilibration, and 100 ns production run using the AceMD engine (Harvey et al., 2009).

	T2		R6	
	Glu 21 Sidechain	Glu 21 Backbone	Glu 21 Sidechain	Glu 21 Backbone
HYP	11.22%	**0.46%**	0.50%	**0.40%**
HZP	10.36%	**17.62%**	18.58%	**17.28%**

Table 3: Occupancy of hydrogen bonds as assessed by molecular dynamics. We observe significantly enhanced hydrogen bonding in insulin containing HZP vs. insulin containing HYP, particularly in the R6 oligomeric form. This is in good agreement with what is observed in crystal structures collected by the Tirrell lab.

In prior work, the Tirrell lab had identified a unique hydrogen bond formed by insulin containing Hzp, which formed across the intersubunit interface,

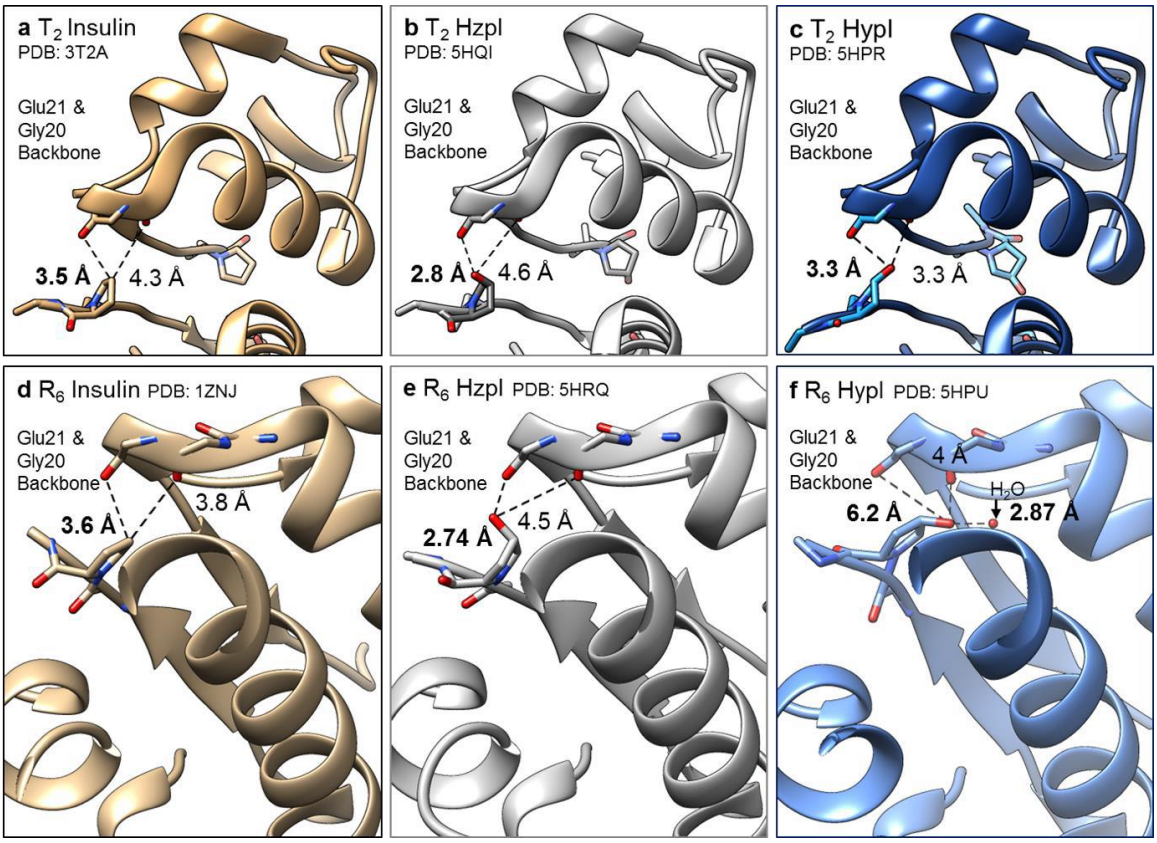


Figure 12: Crystal structures of HZP and HYP. Adapted from Lieblich et al. (2018).

involving Glu 21, it is hypothesized that this bond helps to stabilize dimers, and could be part of the reason that Hzp-containing proline is resistant to fibrillation, which begins from the monomeric form (Ahmad et al., 2005). This hydrogen bond was similarly observed in our molecular dynamics studies, where we compared the hydrogen bonding of Hzp and diastereomer Hyp. Notably, in the T2 (dimeric) form, Hzp forms hydrogen bonds across the intersubunit interface in 27.98% of simulation frames, versus 11.68% of the time for Hyp. This difference is even more pronounced in the R6 (hexameric) form, where Hzp forms an intersubunit

hydrogen bond involving Glu 21 in 35.84% of frames, as opposed to in only 0.90% of frames in insulin containing Hyp.

To further extend the examination of roles of hydrogen bonds in these variants, we continued to perform a detailed examination of hydrogen bonding in monomer, dimer, and hexameric form of each insulin, both in wild-type insulin and lispro variants. While some interesting observations were made, ultimately we tested 100's of putative hydrogen bonding pairs in each variant, and it became difficult to ascertain which correlations were meaningful. Upon applying multiple hypothesis corrections, it became clear that these correlations were spurious. This was an important moment for the project, as it underlined how rich the dataset we had was, and that it would be crucial not to brute search for statistically significant correlations. From this point forward, we decided that all analyses needed to be directed by specific hypotheses.

As the non-canonical prolines being integrated are all at the C-Terminus of the B-chain of insulin, we focused several of our analyses of the systems on this region. Using the monomeric simulations previously described, we measured the solvent accessible surface area of the last four residues of the B-chain, frame by frame using VMD (Humphrey et al., 1996). It was observed that the mean surface area of each variant correlated positively with increasing fibrillation lag time as shown in Figure 13. We additionally calculated the backbone RMSD of the B-chain C-terminus and observed a similar trend. Together these data suggest that insulin variants possessing large solvent-exposed and mobile B-chain C-termini are more resistant to fibrillation. This combination of properties

may reduce the ability of insulin to form polymers by allowing monomers to adopt a wider range of conformations, including those which are divergent from and incompatible with those of nascent fibrils.

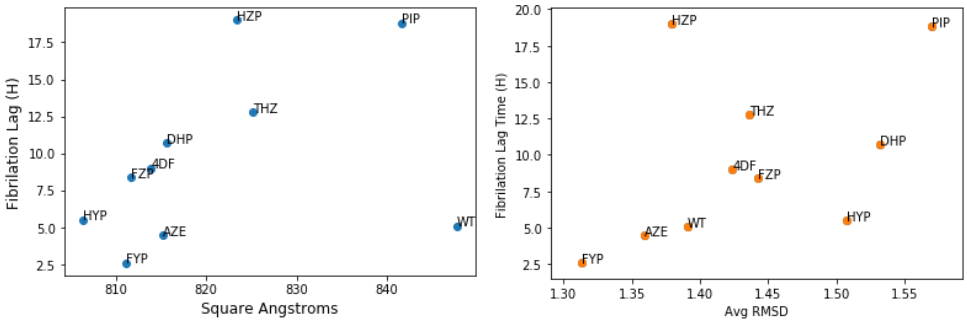


Figure 13: Fibrillation lag time versus (Left) Solvent accessible surface area of the B-chain C-terminus and (Right) the RMSD of the B-chain C-terminus.

Seeking to understand this observation with higher granularity, we examined the torsional angles of the C-terminus of the B-chain and tested several theories about hydrogen bonding (both inconclusive) before looking into the puckering of the proline variants in the molecular dynamics simulations. Proline is unique among amino acids as its side chain composes a heterocycle which significantly influences its backbone conformation. Proline is known to adopt a puckered state which relieves steric strain, and to exist in both endo and exo conformations in crystal structures (Wu, 2013). As the pucking preference of proline controls the position of beta, gamma, and delta hydrogens, and in the case of non-canonical prolines, other substituents, we hypothesized that puckering preferences may impact the properties of proline in multiple ways.

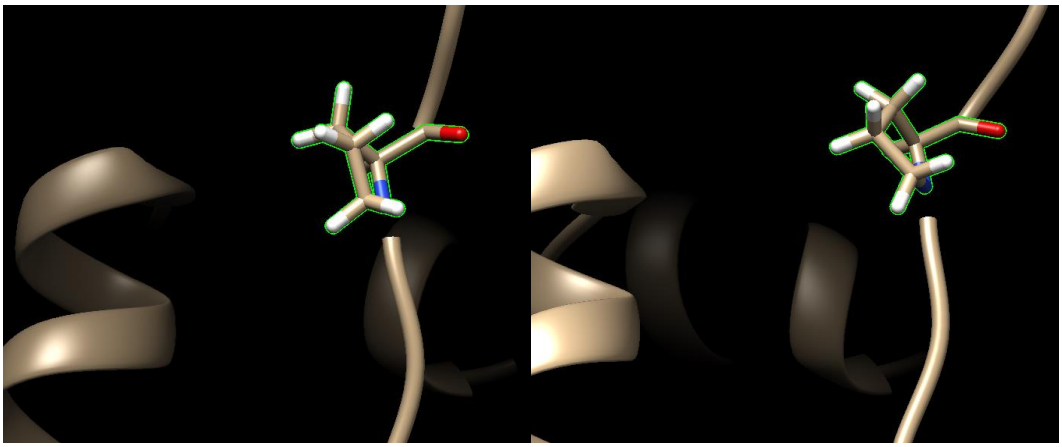


Figure 14: Exo (L) and Endo (R) puckering of proline as seen in an insulin monomer

Using cpptraj (Roe et al., 2013), we calculated the pseudorotation angle and amplitude of pucker of all simulations, and using this information we classified each frame as being either endo or exo. This data is presented in Figures 15 and 16. Across the 7 proline variants which contain five-membered rings, it was observed that fibrillation lag time is positively correlated with preference for endo pucking (Figure 14). We additionally observe that the slower fibrillating variants of insulin exhibit a longer dwell in the endo pucker, that is once they adopt an endo pucker, they are more likely to stay in it for subsequent frames of the simulation. Interestingly, this trend is insignificant for lispro variants containing the non-canonical amino acids, potentially hinting that the effect of being in the endo puckering state is additionally dependent on the proline's interaction with residues in proximity to position B29. To explore this idea, we proceeded to investigate hydrogen bonding involving the engineered prolines, however no significant correlations were observed between bonding patterns and puckering states.

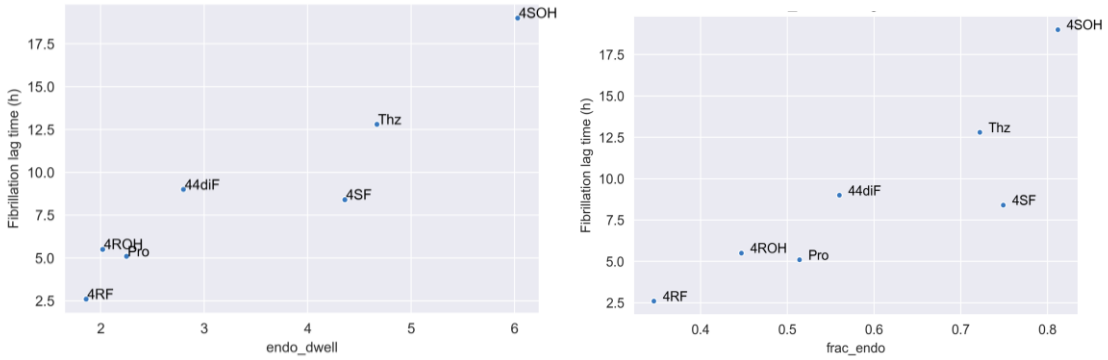


Figure 15: Fibrillation lag time of insulin monomers versus (L) propensity for staying in an endo pucker and (R) the overall amount of time spent in an endo pucker.

To further explore the function of introducing non-canonical prolines into insulin, our collaborators in the Tirrell lab next will synthesize an additional set of proline variants and integrate them into insulin for characterization. This set of thirteen proline variations is shown in Figure 18. Based on our previous observation of pseudorotation as a key correlate of monomer fibrillation, we have prioritized variants AB and CB for immediate characterization. Each of these variants has considerable bias towards endo pucker, even more so than the slow fibrillating, highly endo biased 4SOH variant.

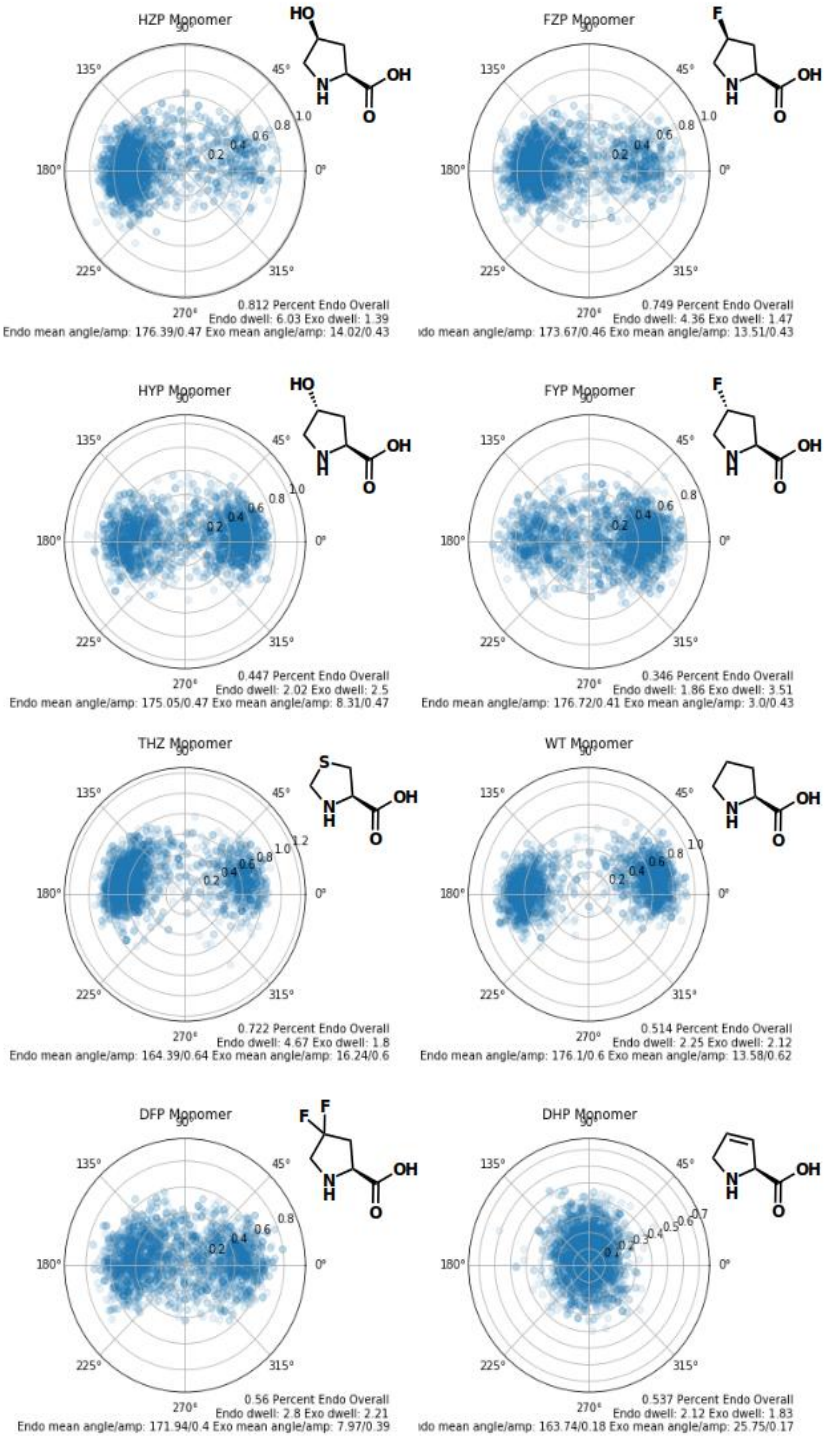


Figure 16: Puckering preferences of characterized non-canonical amino acids integrated at B28 (wild-type insulin).

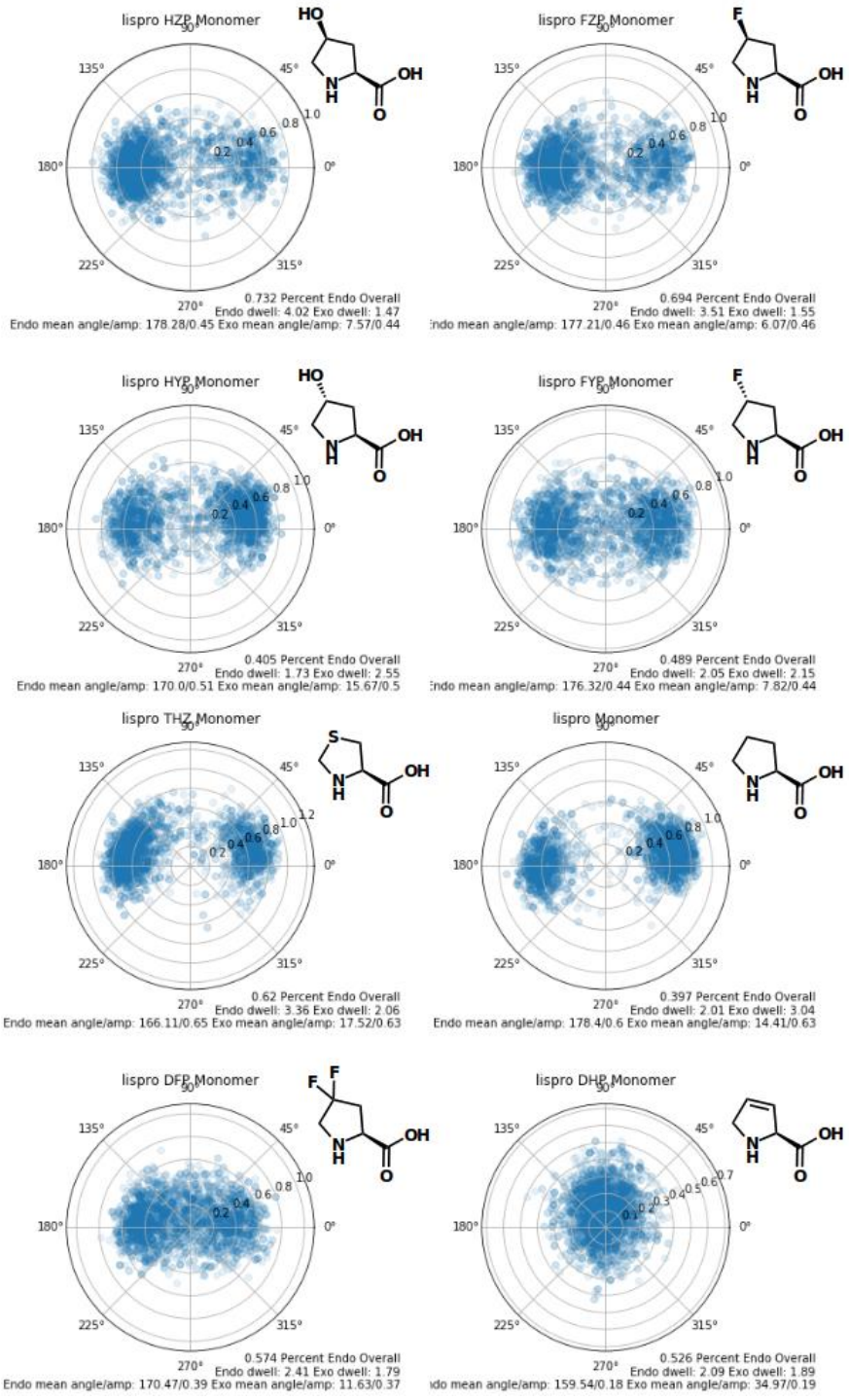


Figure 17: Puckering preferences of characterized non-canonical amino acids integrated at B29 (insulin lispro).

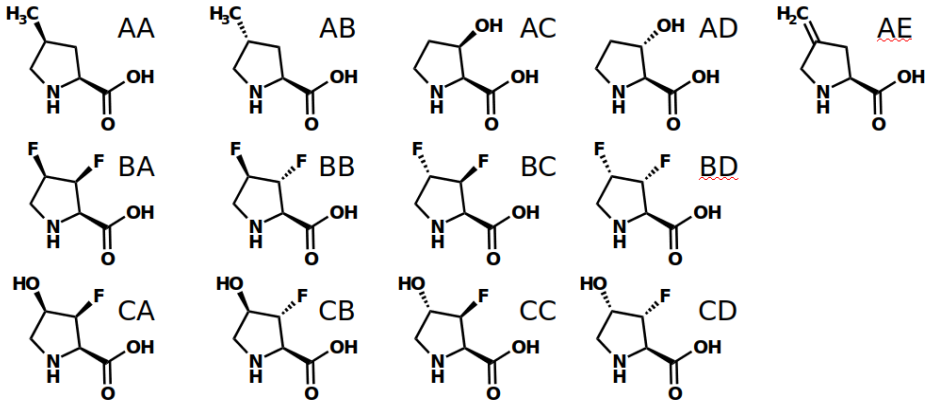


Figure 18: Non-canonical prolines which can be integrated by the Tirrell lab. We have modeled each of these variants and apply our hypothesis about endo puckering to prioritize future work.

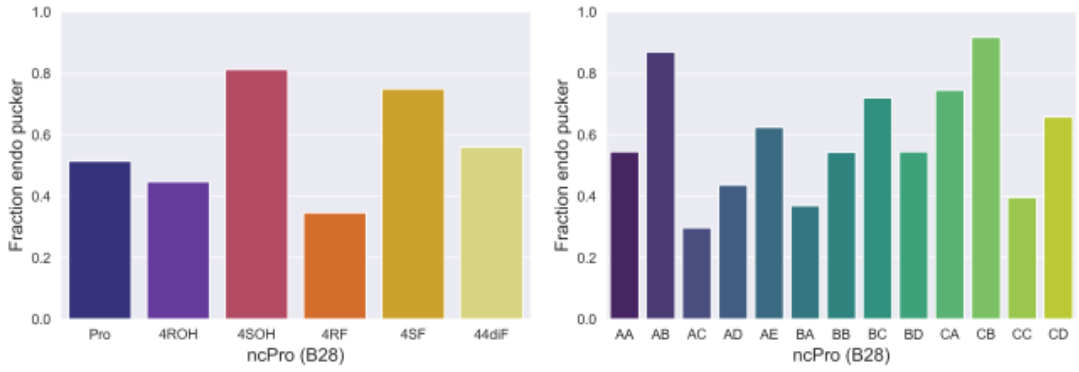


Figure 19: Endo puckering preference (L) existing, characterized non-canonical amino acids and (R) proposed variants to be synthesized and integrated.

Together, these results suggest that the puckering of proline is potentially a contributing factor to the rate at which insulin fibrillation. Based on this hypothesis, our collaborator in the Tirrell lab was produced insulin containing the “AB” proline mutant which was subsequently characterized and found to have a fibrillation half-life of 16.6 +/- 2 hours. This is the second longest fibrillation lag time of any non-canonical variant studied, and in close agreement with what would be expected based on the endo preference of the molecule, as shown in

Figure 20. Luckily, old data was additionally uncovered concerning insulin containing proline AD, which similarly showed good agreement with our model. Further testing of this theory by integrating the proposed insulin variant CB and others will help us to gain confidence and motivate further investigation of the mechanism of this effect.

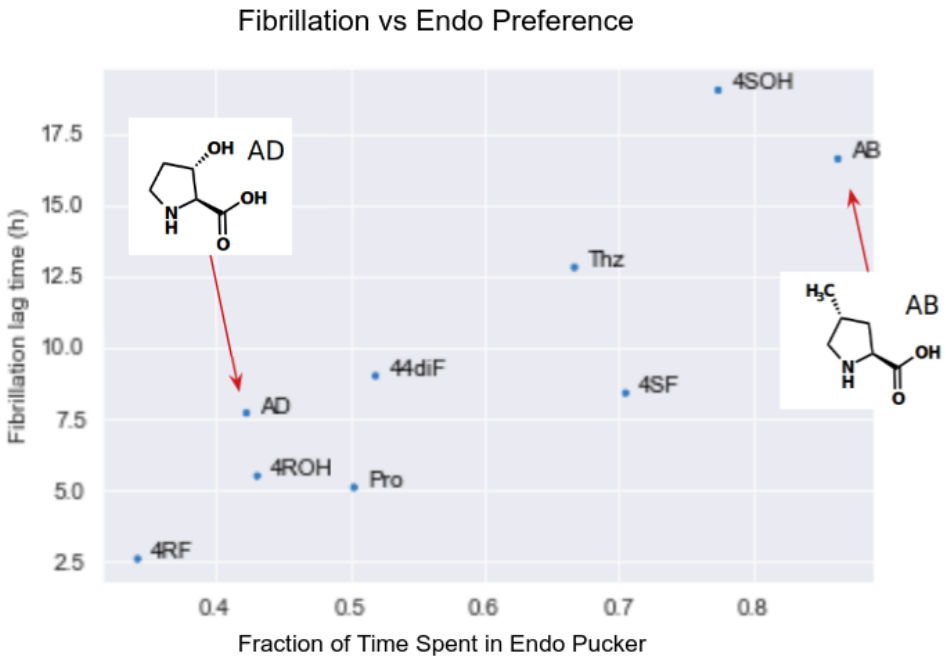


Figure 20: Data collected based on predictions from puckering. Prolines AB and AD were characterized after we proposed the link between fibrillation and endo puckering bias.

2.3 Simulation of insulin in solution and on surfaces: Becton Dickinson project

In the Fall of 2020, we began a project with Becton Dickinson, a leading global producer of diabetes injection devices with approximately 8 billion injection devices produced annually and generating over one billion dollars in revenues. When insulin and other proteins are stored in glass vials and syringes, they are subject to undergoing unfolding and aggregation on the surfaces that they contact. Studying the mechanisms by which this unfolding occurs is difficult, slow, and requires specialized techniques (Angelo et al., 2013, Weltz et al., 2015). Building upon our previous work in modeling insulin, we have endeavored to develop molecular dynamics models which can be used to understand interfacial unfolding events.

In the first phase of this project, we constructed hexameric and monomeric models of insulin, insulin lispro (Humalog), and insulin aspart (Novolog), and conducted simulations using NAMD (Phillips et al., 2020). Although we had already done similar simulations in our non-canonical proline project, we now switched from using AceMD to NAMD, as the latter offers a wider range of force fields and more granular control over simulations. To begin our analysis, we simulated each of the insulin varieties using standard minimization, equilibration, and production protocols. Each system was simulated for three independent replicates of 100 nanoseconds. Upon visual review of each system, we observed that the termini of the B-chain in each protein were highly mobile, while the rest of the proteins were relatively stationary. To investigate in a quantitative manner,

we next identified two areas of interest, the C and N termini of the B-chain, with the C terminus being identified as the final 10 residues of the chain (ERGFFYTPKT) and the N terminus being six residues (FVNQHL). These selections were made upon viewing the molecular dynamics trajectories of the protein monomers, and observing that these segments bookend the relatively stationary helix between them in the B chain. To facilitate analysis, an exemplar structure was selected for each trajectory by clustering the simulation frames according to the method described by (Kelly et al., 1996), and selecting the frame with the closest RMSD to the most populous cluster. Following this, for each simulation, each frame was aligned to its respective exemplar structure, excluding the N and C termini of the B-chain from the calculation. The RMSD of each terminus was subsequently calculated using the position of the backbone atoms compared to the reference structure. We observe statistically significant differences between the RMSD of aspart and lispro for both termini, although significant variability is present in between the replicates of each system, and no clear trend is observable between fibrillation and either RMSD. We observe from plotting the frame-by-frame RMSDs that while the mean RMSDs of each replicate/system are generally similar, the systems sample rare but highly divergent conformations (Figure 21).

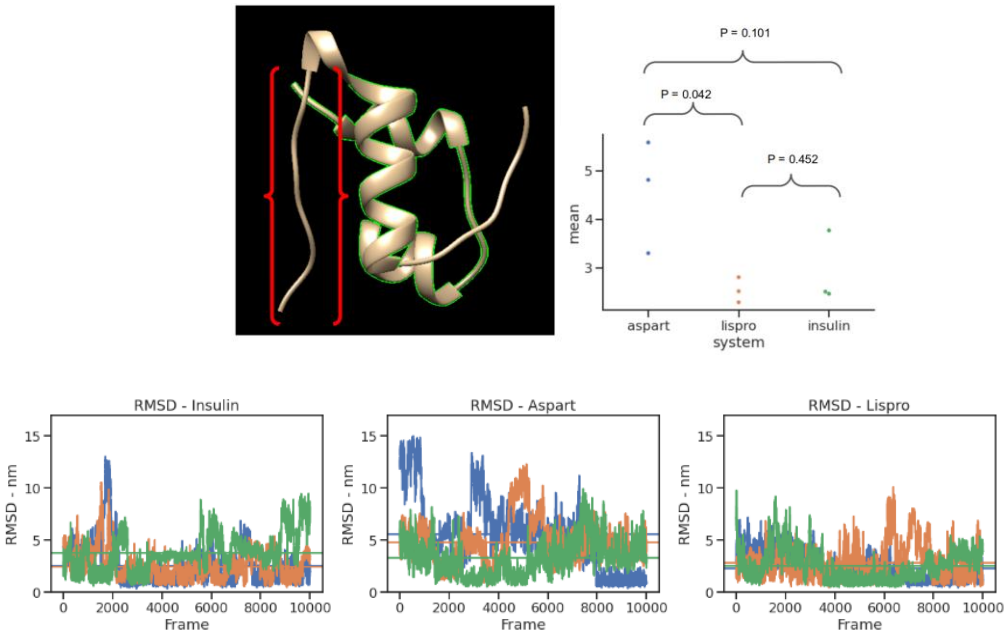


Figure 21: Examination of the RMSD of B-chain C-terminus (final 10 residues). Clockwise from top left: image of insulin monomer; RMSD calculations are done by aligning on the segments highlighted in green and calculating the backbone RMSD of the bracketed red section; the mean RMSD of each simulation is plotted; frame-by-frame plots of RMSD over the course of the simulations.

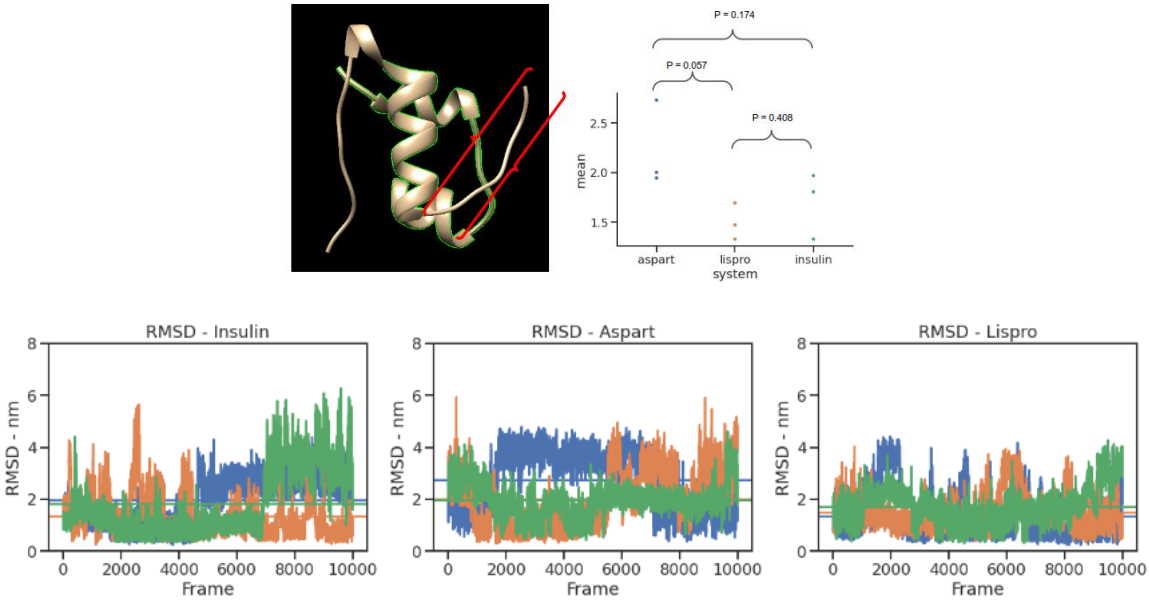


Figure 22: Examination of the RMSD of B-chain N-terminus (final 6 residues). Clockwise from top left: image of insulin monomer; RMSD calculations are done by aligning on the segments highlighted in green and calculating the backbone RMSD of the bracketed red section; the mean RMSD of each simulation is plotted; frame-by-frame plots of RMSD over the course of the simulations.

In addition to the RMSD of each terminus, we additionally examined the solvent accessible surface area and subset of that surface which is hydrophobic across each simulation. While considering the mean values of these properties produced indistinguishable results, the correlation matrices of the solvent accessible surface area for each residue across the simulations revealed insights about the dynamics of the proteins. In particular, we observe the last seven residues of insulin's B-chain C-terminus to behave in a concerted manner, with

particularly strong correlations between this section and both termini of the A-chain (Figure 23).

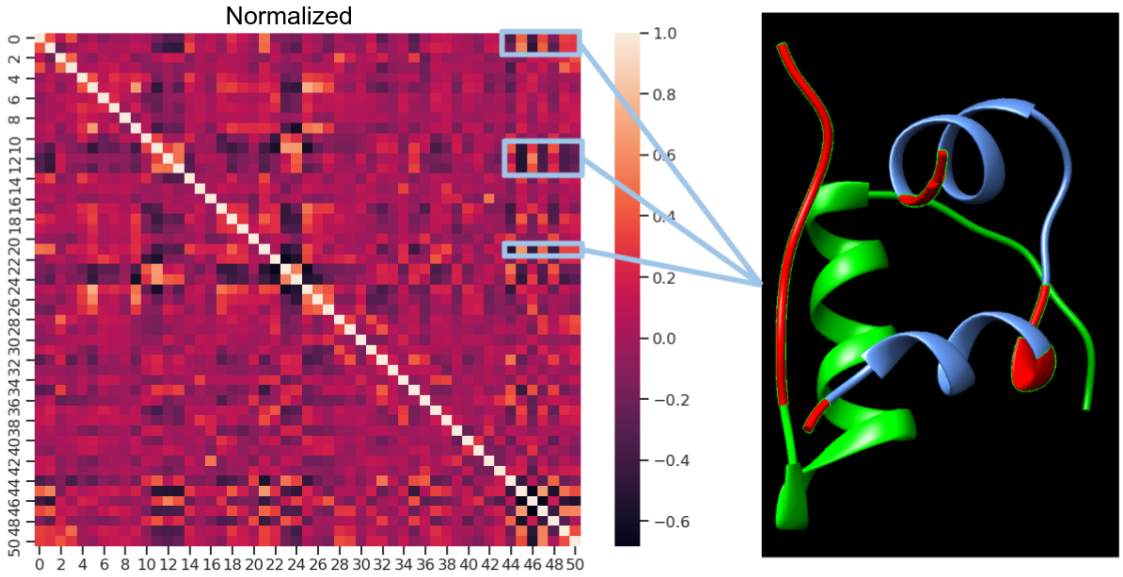


Figure 23: Correlation matrix of solvent accessible surface area between residues of insulin: highlighted on the right are residues which are observed to interact in a concerned manner with the B-chain C-terminus.

However, this observation does not hold true for insulin aspart, where the solvent accessible surface area of Leucine A12 is instead seen to correlate negatively with that of the adjacent residues both preceding and following it (Figure 24).

Unlike either wild-type or aspart insulin, insulin lispro does not exhibit any discernible correlation among the solvent accessible surface area of its residues.

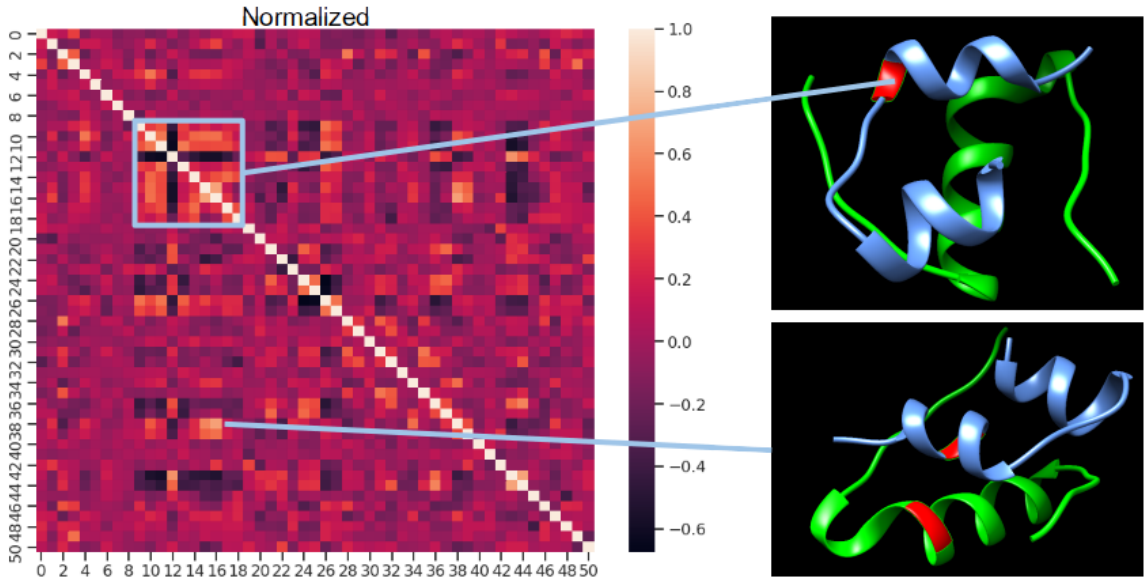


Figure 24: Correlation matrix of solvent accessible surface area between residues of insulin aspart: highlighted on the right are residues which are observed to interact at A12.

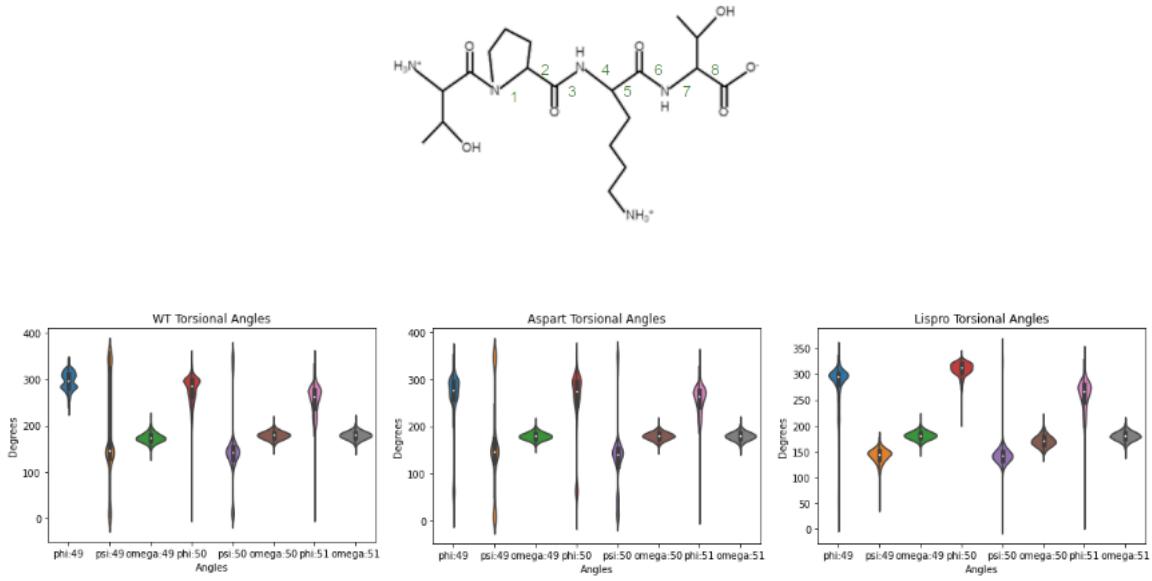


Figure 25: Torsional angles of B-Chain C-terminus: (Top) schematic of wild-type insulin, with bonds labeled 1-8 corresponding with violin plots left to right; (Bottom) Violin plots of torsional angles adopted by each system.

To understand the variability of backbone torsional angles in the three variants, we further examined the range of phi, psi, and omega angles exhibited in the backbones of each of the three variants (Figure 25). The violin plots generated show the expected shift of phi angle flexibility when proline is either removed (insulin aspart) or shifted one position closer to the C-terminus (insulin lispro).

Having demonstrated that we could simulate the relevant systems in solution, we next moved to simulating them in the presence of glass surfaces. The simulation of silica surfaces represents a much more challenging application than the simulation of proteins in solution. Although efforts have been made for many years to create tools for such simulations, a poor understanding of the surface properties of silica limited the implementation of atomistic forcefields. Progress with magic angle spinning, atomic force microscopy, and tunneling microscopy have recently allowed groups to characterize silica containing materials, and subsequently refine computational forcefields to useful standards of accuracy (Emami et al., 2014). While previous studies have identified surface polarity as a potentially driving factor of insulin adsorption onto silica surfaces (Nejad et al., 2017, 2018), we believe our work is the first study of insulin interacting with pH appropriate siloxide deprotonation.

Silicon dioxide surfaces have a highly variable range of surface chemistries depending on the conditions of manufacture and ionic strength of exposed solution. In this project we have simulated quartz, a geometrically consistent form

of silica characterized by two silanol groups per superficial silicon atom in a geminol configuration.

Methods

We construct sheets of quartz which are 75x75x15 Angstroms and ionize the surface in a manner consistent with a pH of 7.4. These sheets are placed in fully periodic cells which are 75x75x90 Angstroms in diameter, with the quartz sheet placed normal to the long axis of the box, and precisely in the middle of the system (Figure 26). TIP3 water molecules (MacKerell et al., 1998) are placed in the box for an average of 13,100 waters per system. In simulations containing protein, the completed system is generated by adding insulin or a derivative at the center of the system, before moving 30 angstroms along the Z-axis to start approximately 10 angstroms from the surface of the quartz sheet. The protein is then rotated about its geometric center by applying a randomly generated rotational matrix and waters clashing with the protein are removed to yield the completed system.

In order to prevent the quartz slab from drifting, we apply positional restraints to the internal SiO subunits which are > 5 angstroms from the water interface. The system is minimized before a standard equilibration protocol and a 100 nano second production run using the CHARMM-IFF forcefield (Emami et al., 2014).

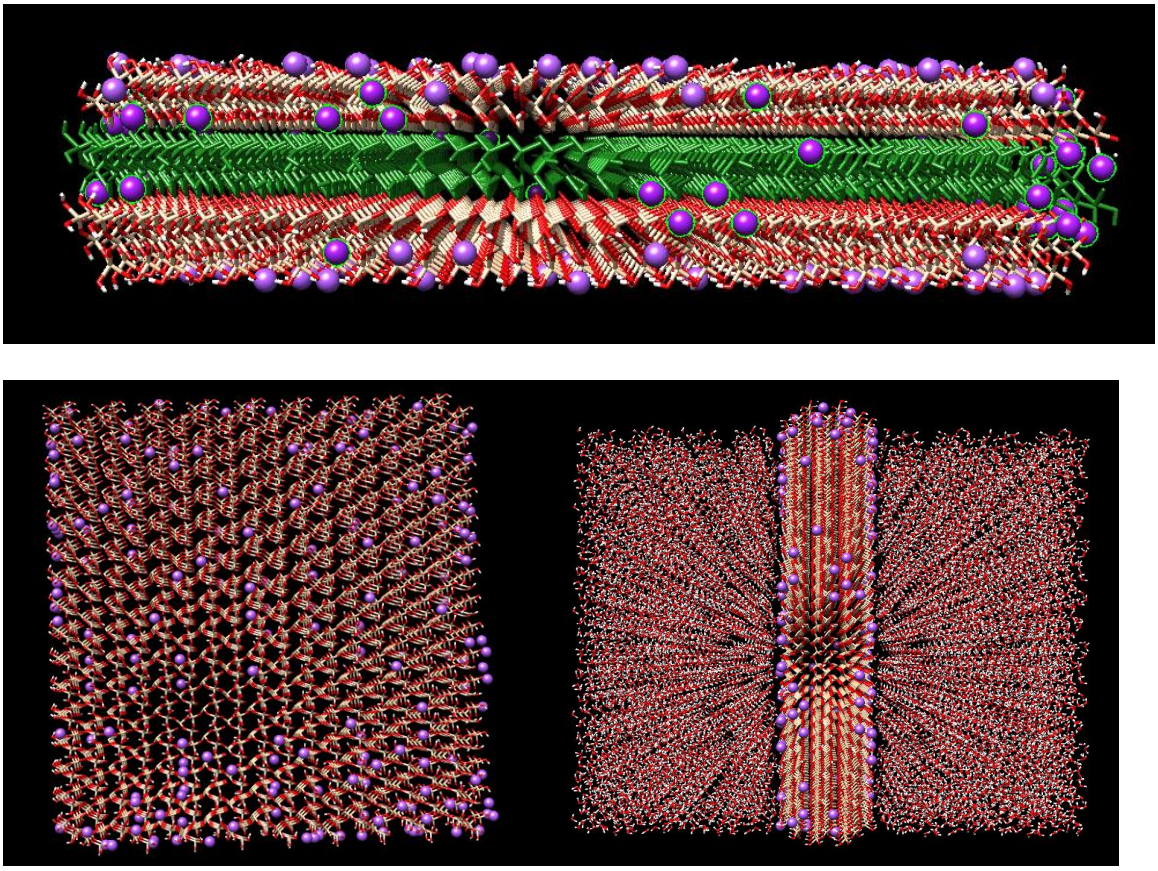


Figure 26: Quartz slabs as simulated: (Top) Side view of slab. Residues colored in green are restrained to prevent slab from drifting in the periodic box. (Left) Top view demonstrating randomly deprotonated silanols. (Right) Side view with 90x75x75 Angstrom water box with ~13,100 TIP3 waters.

Results

We first validate the size of our simulation system by assessing the formation of distinct water layers on the surface of the quartz sheet. By calculating the radial distribution function describing the distance between water molecules and surface silanols, we are able to see the arisal of three distinct layers of water solvating the surface of quartz (Figure 27). The first layer spans from the surface of quartz to approximately 1.8 angstroms, the second from 1.8

to 3.25 angstroms, with the remainder of waters being the bulk disordered phase. Using a similar protocol, we additionally observe an approximately three angstrom layer of ordered water around insulin which is restrained to negate translational motion. Together these results support that the 9 angstrom distance separating insulin (and derivatives) from quartz in our systems is sufficient to not introduce bias to the behavior of the combined system.

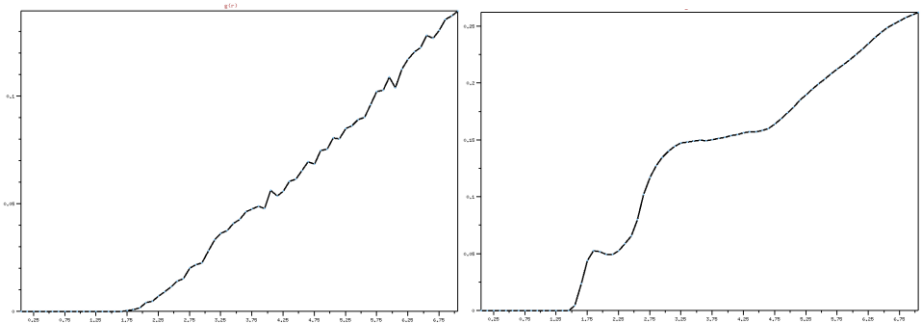


Figure 27: Radial distribution function: (L) Before simulation, water molecules are uniformly distributed and exhibit a monotonically increasing cumulative probability density. (R) After simulation, three distinct layers are observed.

We performed three replicates of each insulin, insulin aspart, and insulin lispro in the system described, employing distinct, randomly generated surfaces (differing in sites of silanol deprotonation), and randomized starting orientations of proteins. Somewhat surprisingly, we observed that close contacts between proteins and quartz occurred in each simulation. In simulations of wild-type insulin, A8-10 (TSI), B1-2 (FV), and B29 (K) interact with the surface in each respective replicate (Figure 28). In all three replicates of insulin lispro, B1-2 (FV) interacts with the surface with B28,30 (K,T) participating transiently in one replicate. In two of three replicates of insulin aspart, B1-2 (FV) interacts with the

quartz surface, and B22 (R) interacts with the quartz in the third.

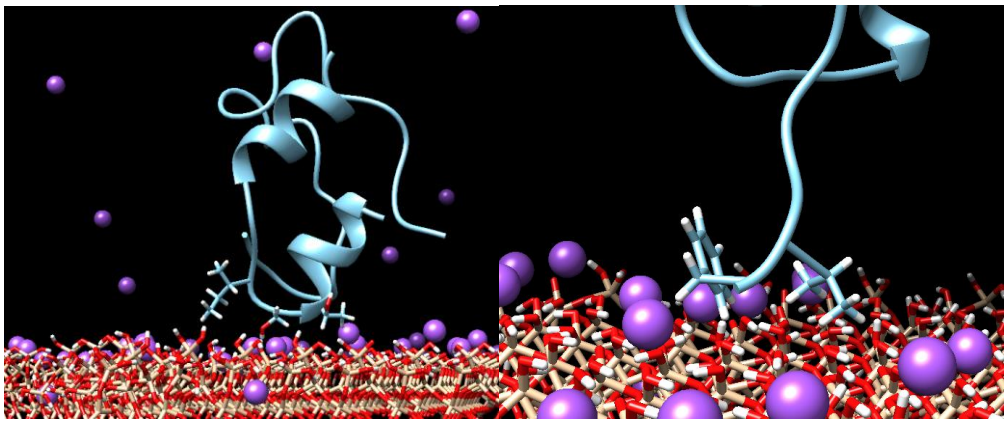


Figure 28: Wild-type insulin monomers interacting with quartz surface. (L) WT + Surface 1: A8-10 (TSI) binds to surface (R) WT + Surface 2: B1-2 (FV) interacts with surface transiently.

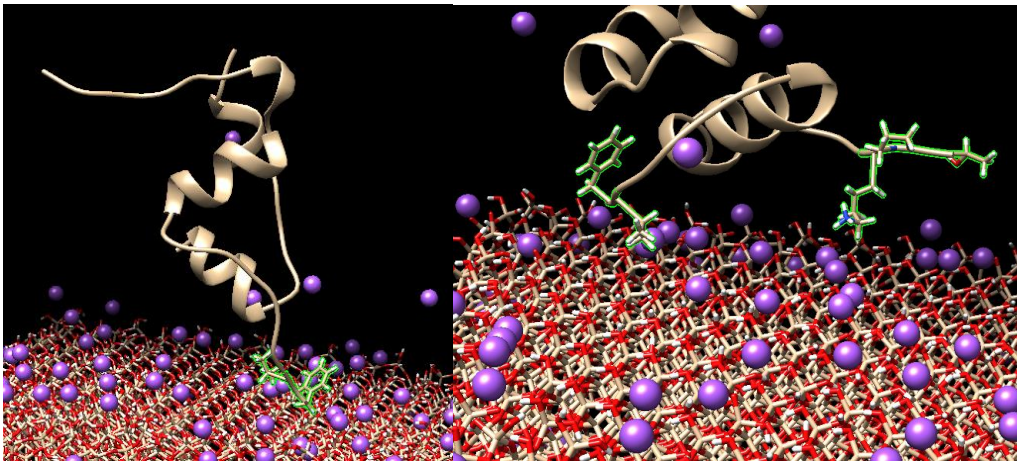


Figure 29: Insulin lispro interacts with quartz surface: (L) Lispro + Surface 3: B1-2 (FV) interacts with surface (R) Lispro + Surface 1: B1-2 (FV) interacts with surface. B28,30 (K,T) participate transiently.

The observation that insulin and its derivatives contacted the quartz surface in every simulation was unexpected, so to further validate that we were not starting the proteins too close to the surface, and thus introducing an artificial

effect, we next expanded the size of the systems. Larger periodic cells of size 200Å x 75Å x 75Å were constructed to allow the protein to start an additional 55 angstroms away from the surface, and the vertical motion of the protein was analyzed. To characterize the position of the protein relative to the quartz surface, the Z-coordinate of the geometric center of all protein atoms was calculated, and plotted for the duration of the simulations (Figures 30, 31).

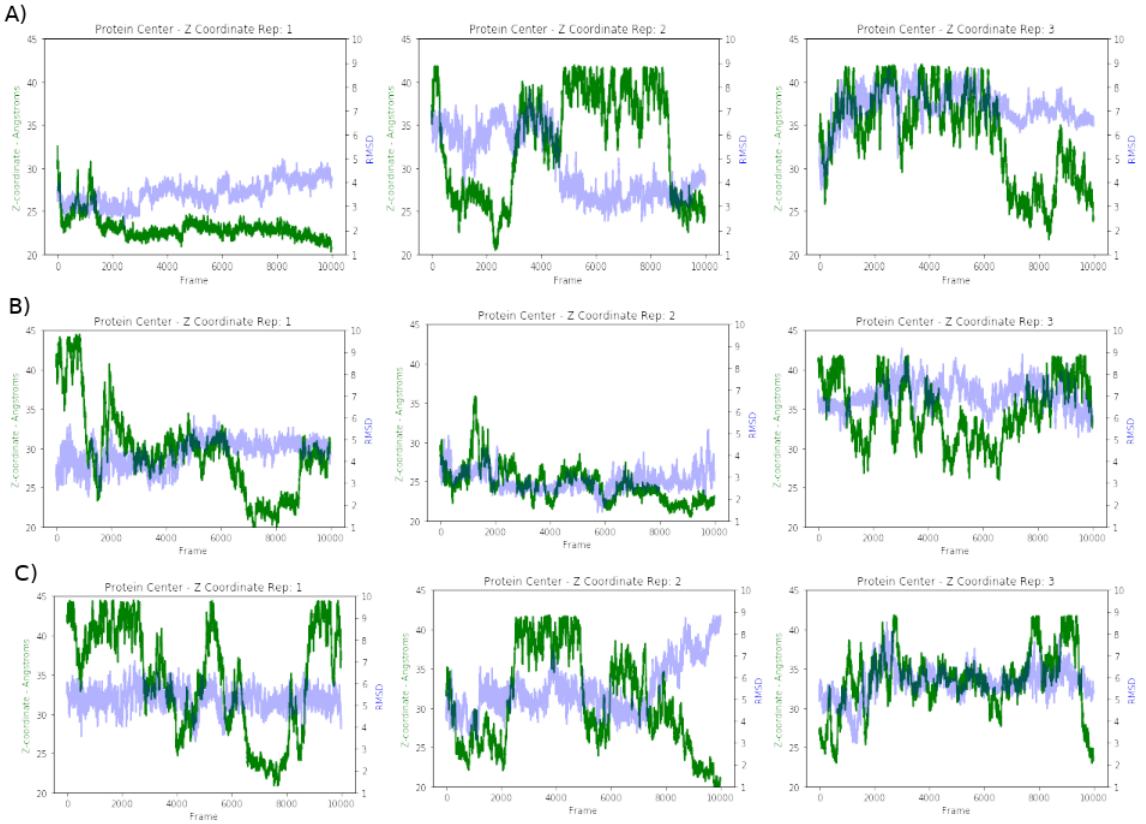


Figure 30: Z-Coordinate of protein center over simulated trajectory. Green line depicts the absolute value of the insulin center of mass. Blue line depicts RMSD of molecule against reference structure (crystal structure). (Top) Wild-type insulin; (middle) insulin aspart; (bottom) insulin lispro.

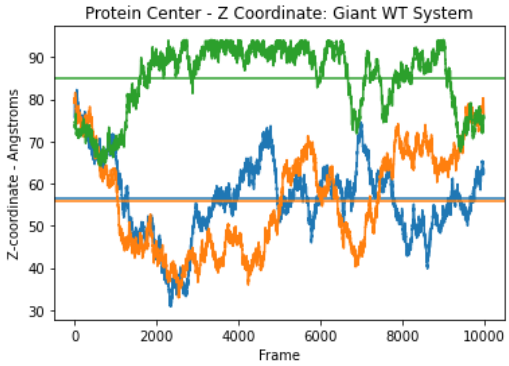


Figure 31: Z-Coordinate of enlarged system

In the simulation of the larger system, in which insulin starts 85 angstroms away from the quartz sheet, we observe that two of three replicates drift 45 angstroms closer to the quartz over the course of 100 ns, and the third system moves roughly 15 angstroms closer. In none of these simulations does insulin contact the surface of the quartz despite an average maximum displacement in this system of 35 angstroms. Given that the original systems started with a gap of 10 angstroms between protein and quartz surface, it stands to reason that the contact could have thus been caused by random walk. Although we could further extend the size of our systems, or run longer simulations to ensure that even in this setting protein-surface contact was reliably achieved, taken together the data suggests that our initial 90x75x75 angstrom systems are of sufficient size to view the contact as random and not influenced by the starting state of the systems.

To appropriately represent the protonation state of the surface at a pH of 7.4, we have randomly deprotonated 51 silanols and added sodium counter ions to the quartz surface. While these sodium ions are free to diffuse about the system and sometimes

do, they often are retained in close proximity to the quartz surface. In viewing the trajectories of the systems, we observed transient interactions between aromatic amino acids and ions on the surface of quartz that resemble cation-pi interactions. As an initial test of this, we next prepared fully protonated, ion-free quartz surfaces, and replicated our simulations with wild-type insulin. These experiments disproved our hypothesis that cation-pi interactions are the primary driver of surface interaction, as all three replicates quickly contacted the quartz surface, and were retained for an extended amount of time despite a complete lack of ions (Figures 32, 33).

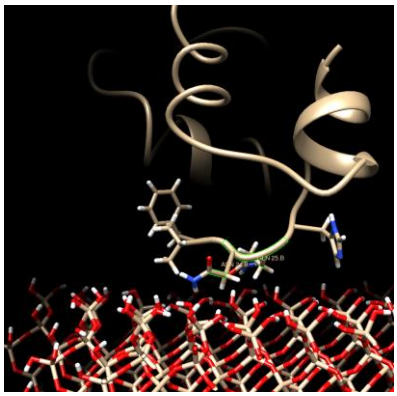


Figure 32: Wild-type insulin interacting with fully protonated quartz surface.

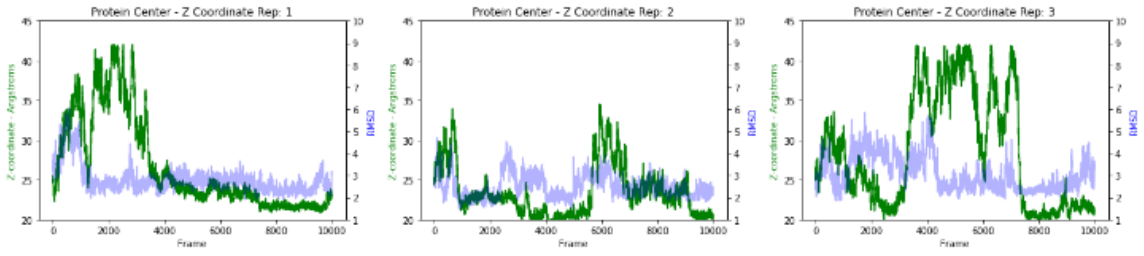


Figure 33: Z-coordinate of insulin interacting with quartz surface.

To cross-examine this hypothesis, we next evaluated the angles between each aromatic side chain and the quartz surface. If cation-pi interactions were

occurring, we would expect to see a bias for the aromatic rings of side chains to adopt conformations perpendicular to the quartz sheet (Figure 34).

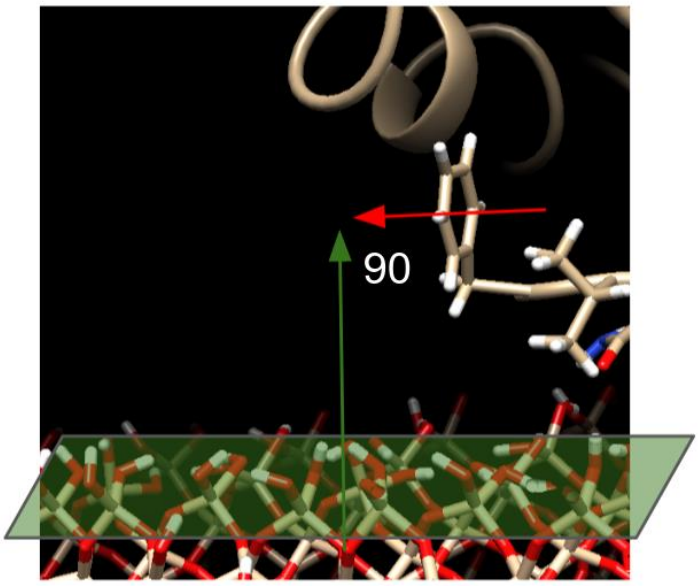


Figure 34: Representation of angle between quartz surface and aromatic rings. Note that values range between 0 and 90 degrees.

To test this, we first calculated normal vectors to the surface and each aromatic ring at every step in the trajectory. The angle between these vectors can subsequently be calculated as the arc cosine of the normalized dot product of the vectors. Despite our initial visual observation of transient cation-pi interactions, they were generally not observed in this numerical analysis.

However, A14 (Y) was observed to be biased toward adopting a perpendicular angle to the surface in a distance dependent manner (Figure 35).

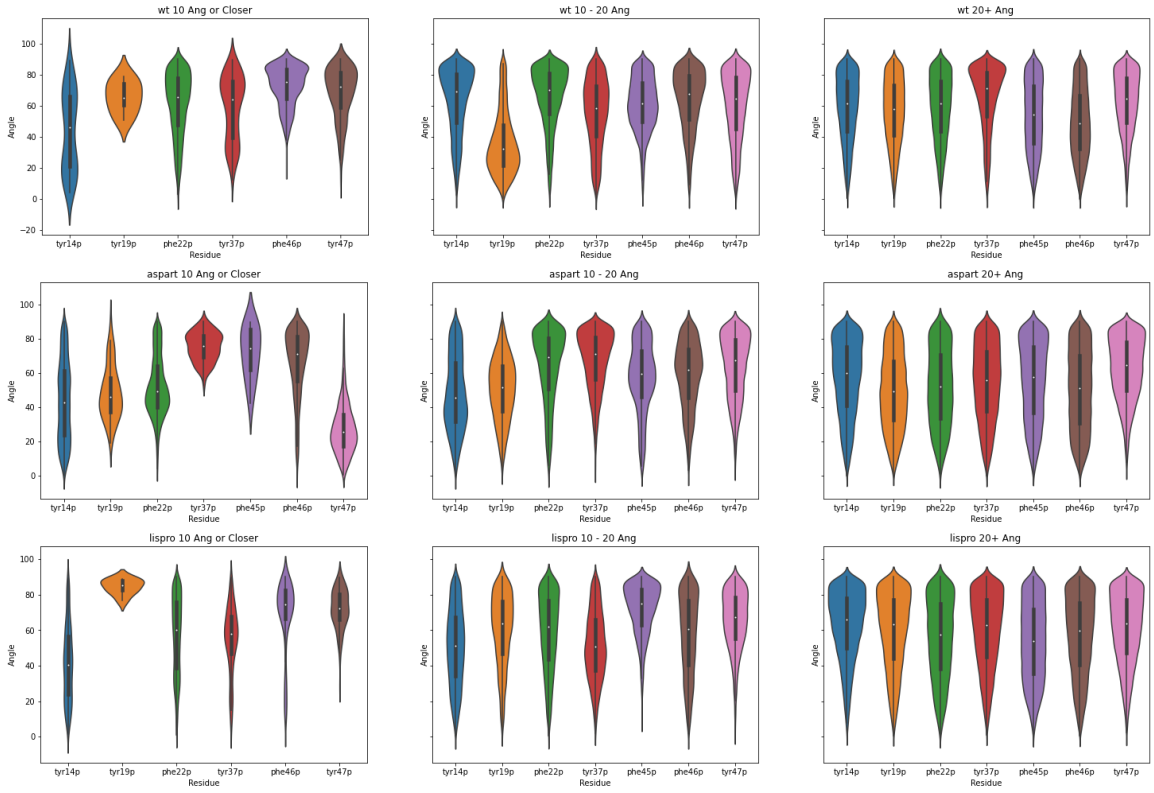


Figure 35: Distribution of angles between aromatic side chains and quartz surface, stratified by distance from surface: (Top) Wild-type insulin; (Middle) insulin aspart; (Bottom) insulin lispro.

To closer examine this, we constructed a more granular analysis by identifying the closest sodium ion to each aromatic ring at every step of the simulations and plotting these distances as a function of time. These plots are shown in Figures 36-38, and allow for a relatively easy view of when aromatic rings are in proximity to cations on the surface of the quartz sheet. Filtering for when these rings are within 6 angstroms of the surface to identify when a cation pi

interaction can occur (Gallivan et al., 1999), we observe that A14 (Y), B1 (F), and B25(F) fall within range for 1% or more of frames for each system. Notably, in one simulation of insulin aspart, B25 is observed to be in range to form a productive cation-pi interaction for 51% of frames, or more than 200x more frequently than A19 (Y). In another replicate of insulin aspart, B1 is in range for a cation-pi interaction for 23% of frames. Similarly for a replicate of insulin lispro, B25 is in range for a cation-pi interaction for 11% of frames. Taken together, these data support that cation-pi interactions contribute to the association of insulin derivatives and quartz surfaces, but are likely not the sole driver. We note that in many of the frames, the aromatic ring of B1 is at the wrong angle to form cation-pi interactions and instead appears to be perpendicular to the face of the quartz sheet.

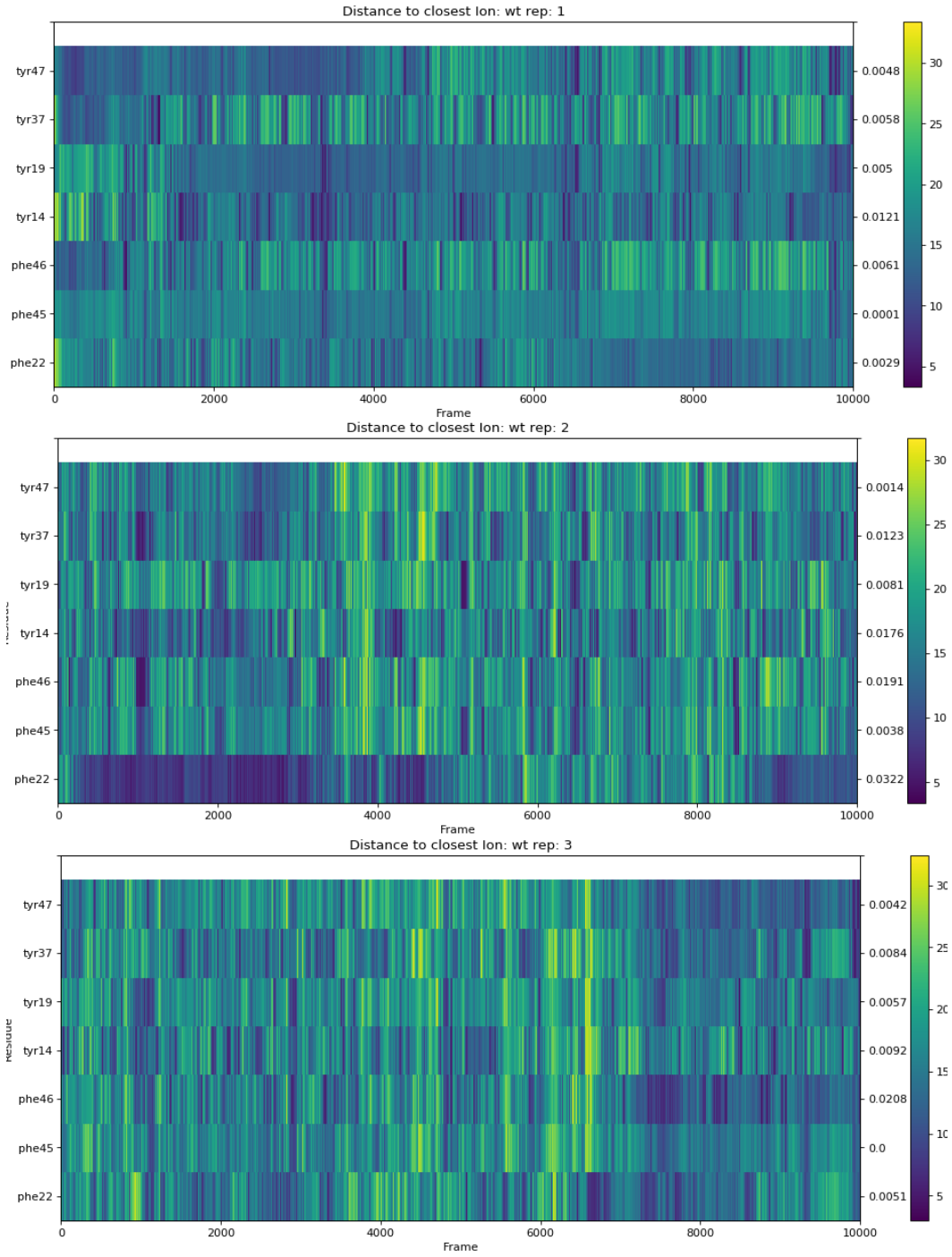


Figure 36: Distance from each aromatic ring in protein side chain to the closest ion. Wild-type insulin.

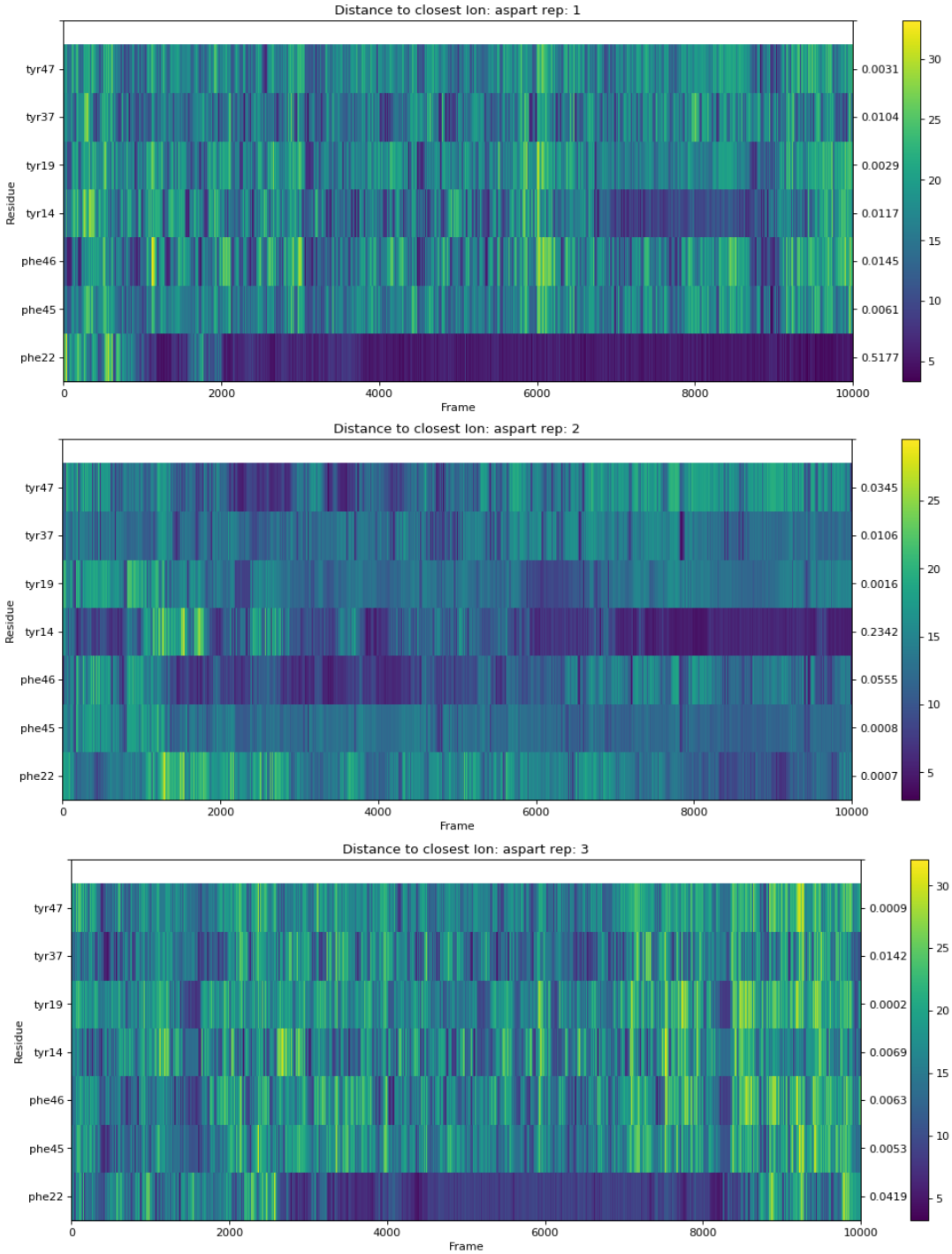


Figure 37: Distance from each aromatic ring in protein side chain to the closest ion. Insulin aspart.

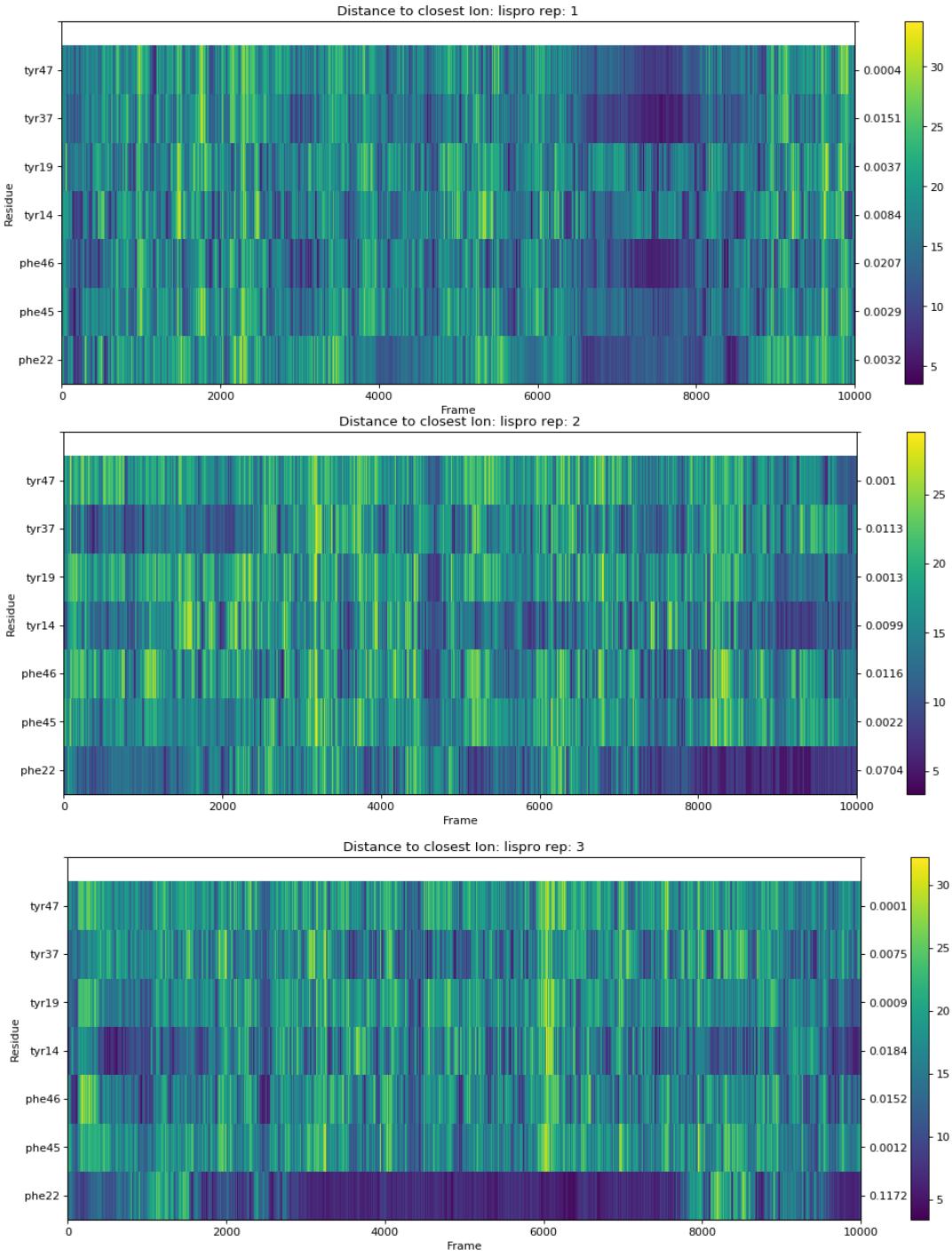


Figure 38: Distance from each aromatic ring in protein side chain to the closest ion. Insulin lispro.

To further assess key aspects of the interactions between insulin and quartz, we conducted a similar analysis to assess the distance from all side chain residues to the quartz surface, regardless of amino acid identity (Figures 39-41). This revealed that the termini of the proteins were by far the most likely to be in close proximity to the quartz surface, and most so the N-terminus of the B chain. In all three lispro replicates, and one replicate each of aspart and wild-type insulin, this region spends 10%+ of the simulation in close proximity to the quartz surface. To gain insight to the factors governing hydrogen bonding to the simulated quartz surfaces, we next examined the hydrogen bonding patterns of the three studied insulin variants. In three of the nine simulations, we note that B1 (F) is the most common hydrogen bond, including being present for nearly 70 of 100 nanoseconds in one replicate of insulin aspart. This observation aligns well with our previous observations that the B-chain N-terminus is a key contributor to association with quartz surfaces. Interestingly, des-B1 insulin is known to exhibit similar insulin receptor binding and glucose reduction efficacy in human subjects, suggesting further investigation of this variant in applications sensitive to fibrillation in the presence of glass surfaces.

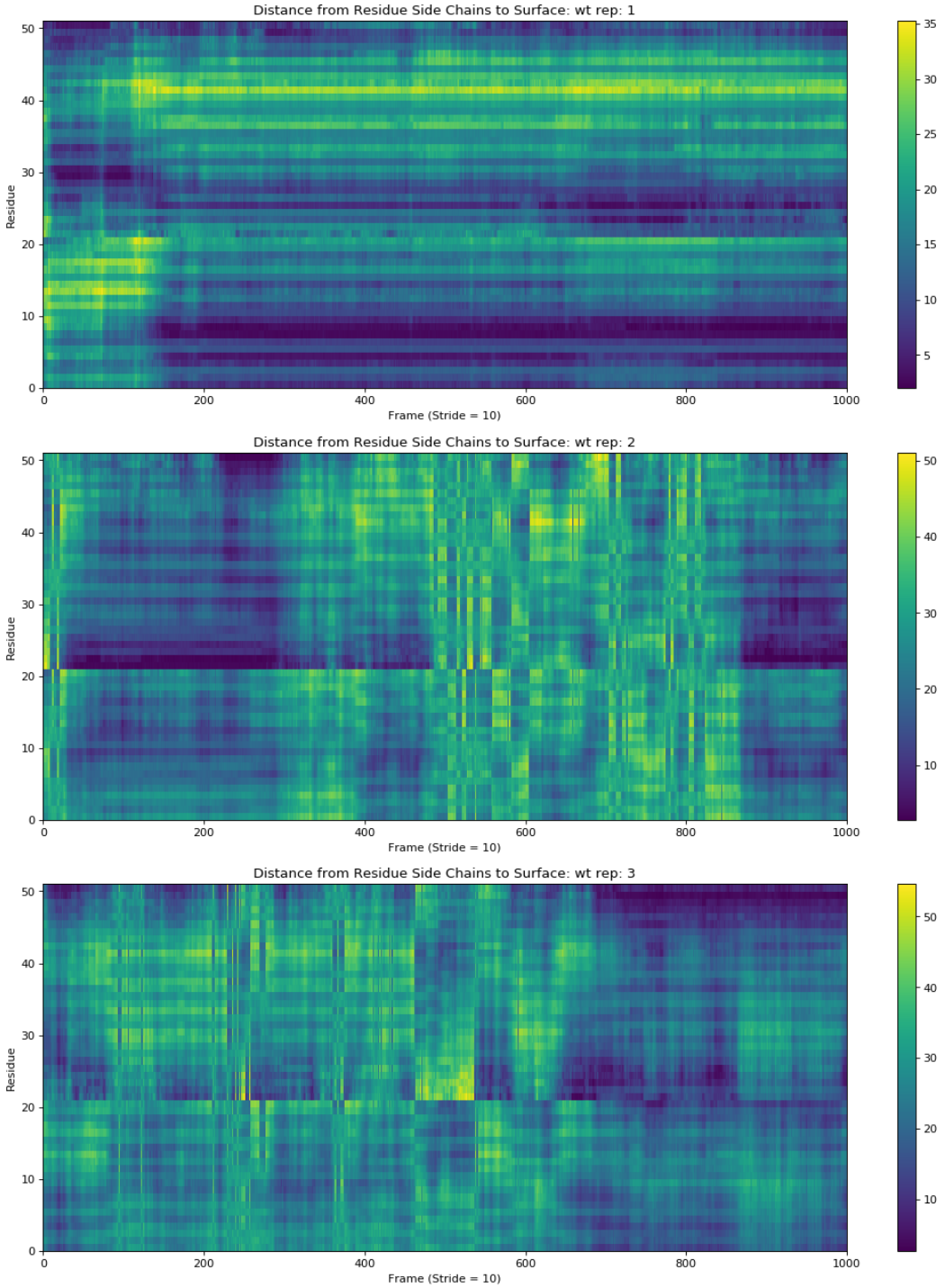


Figure 39: Distance from side chain geometric centers to quartz surface. Wild type insulin.

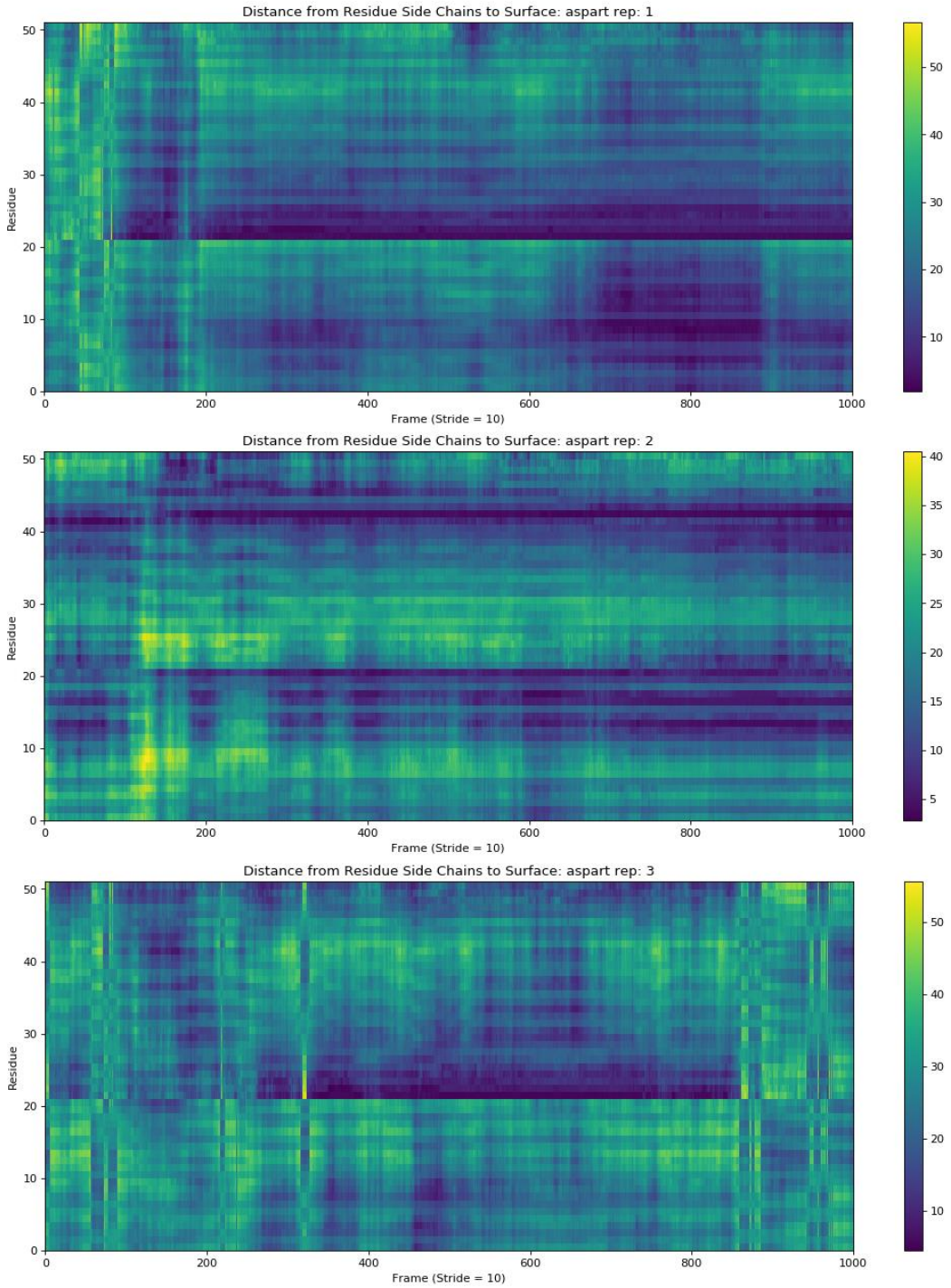


Figure 40: Distance from side chain geometric centers to quartz surface. Insulin aspart.

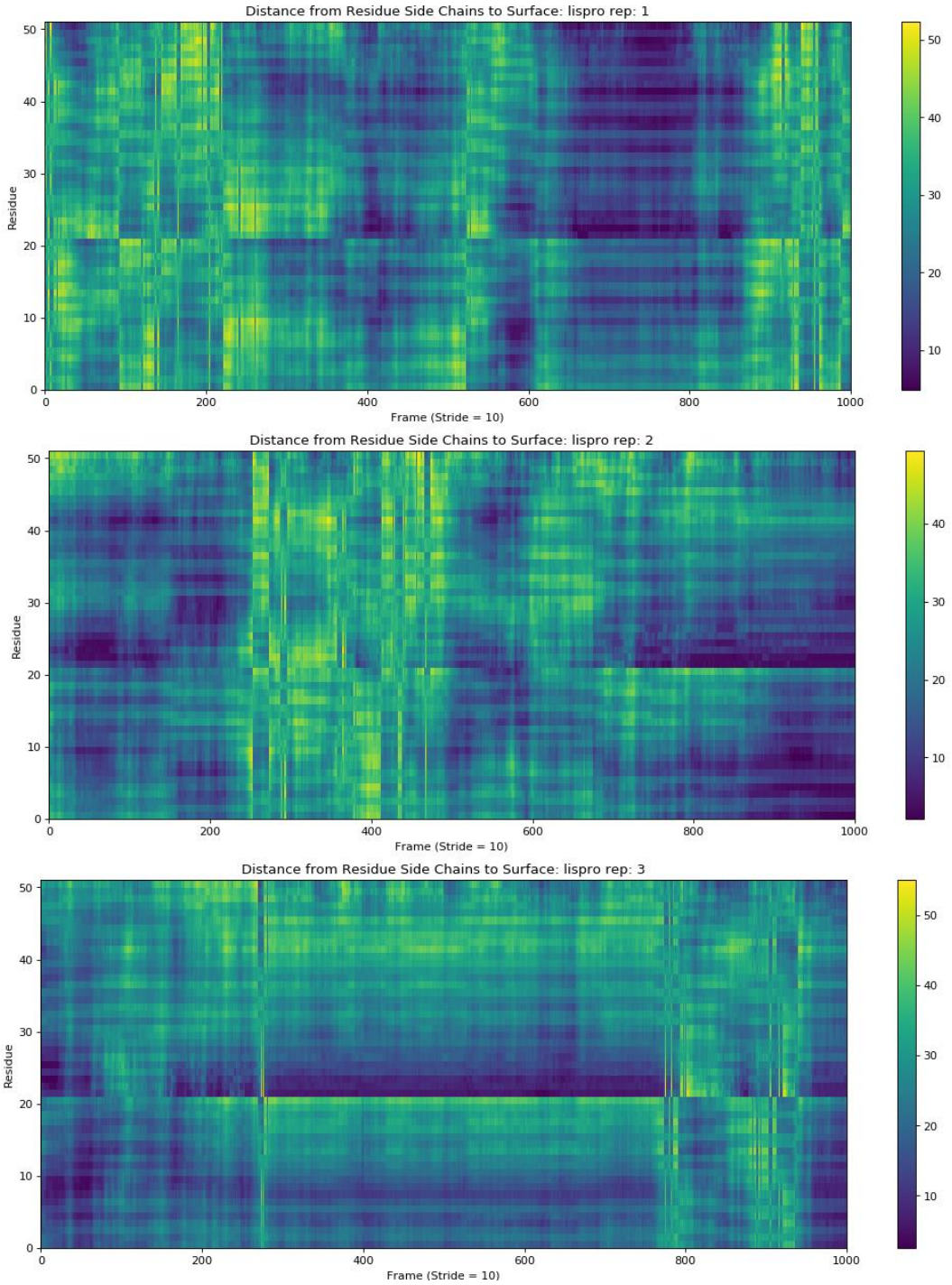


Figure 41: Distance from side chain geometric centers to quartz surface. Insulin lispro.

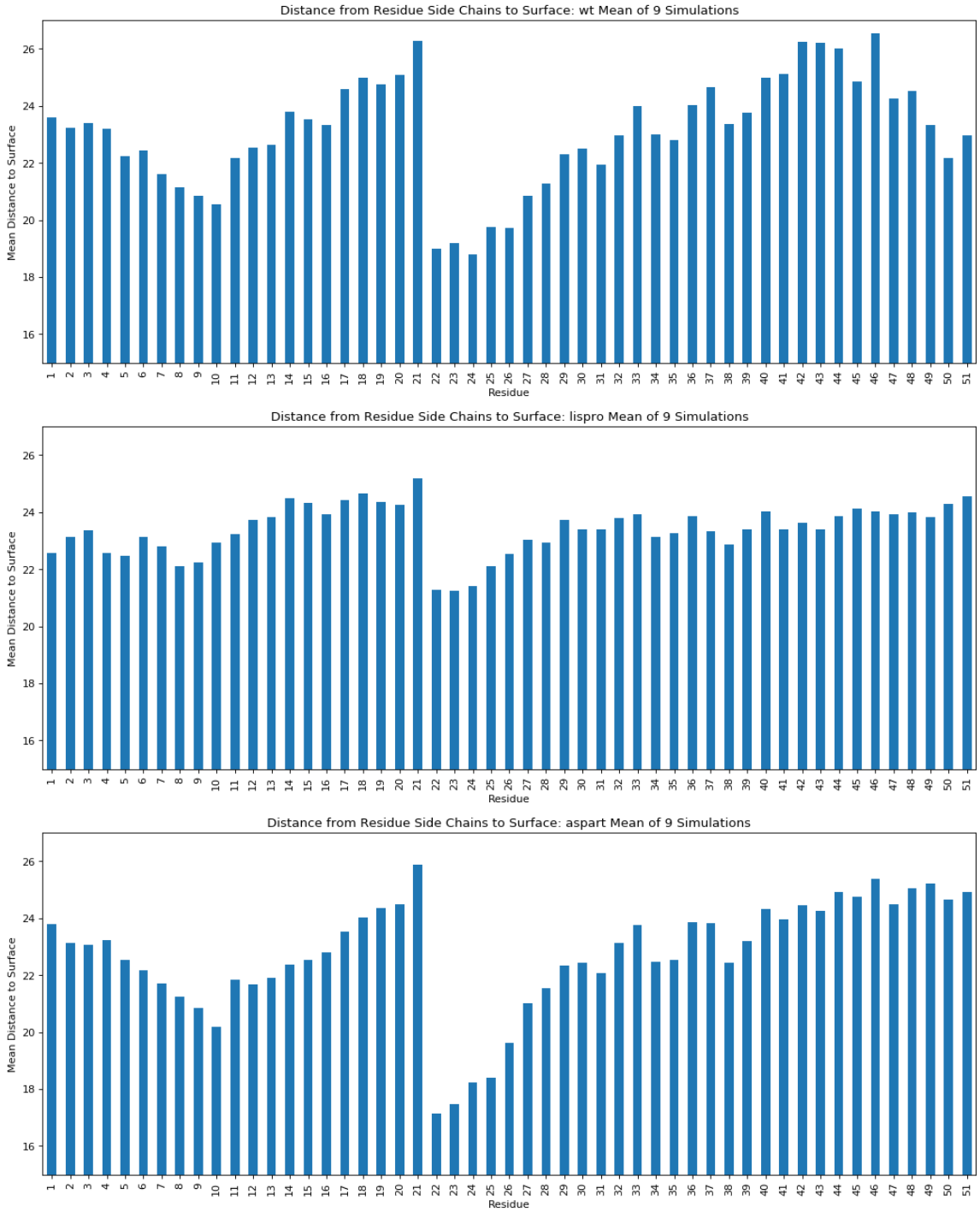


Figure 42: Average distance between insulin side chains and quartz surface. (Top) Wild-type insulin; (Middle) Insulin lispro; (Bottom) Insulin aspart.

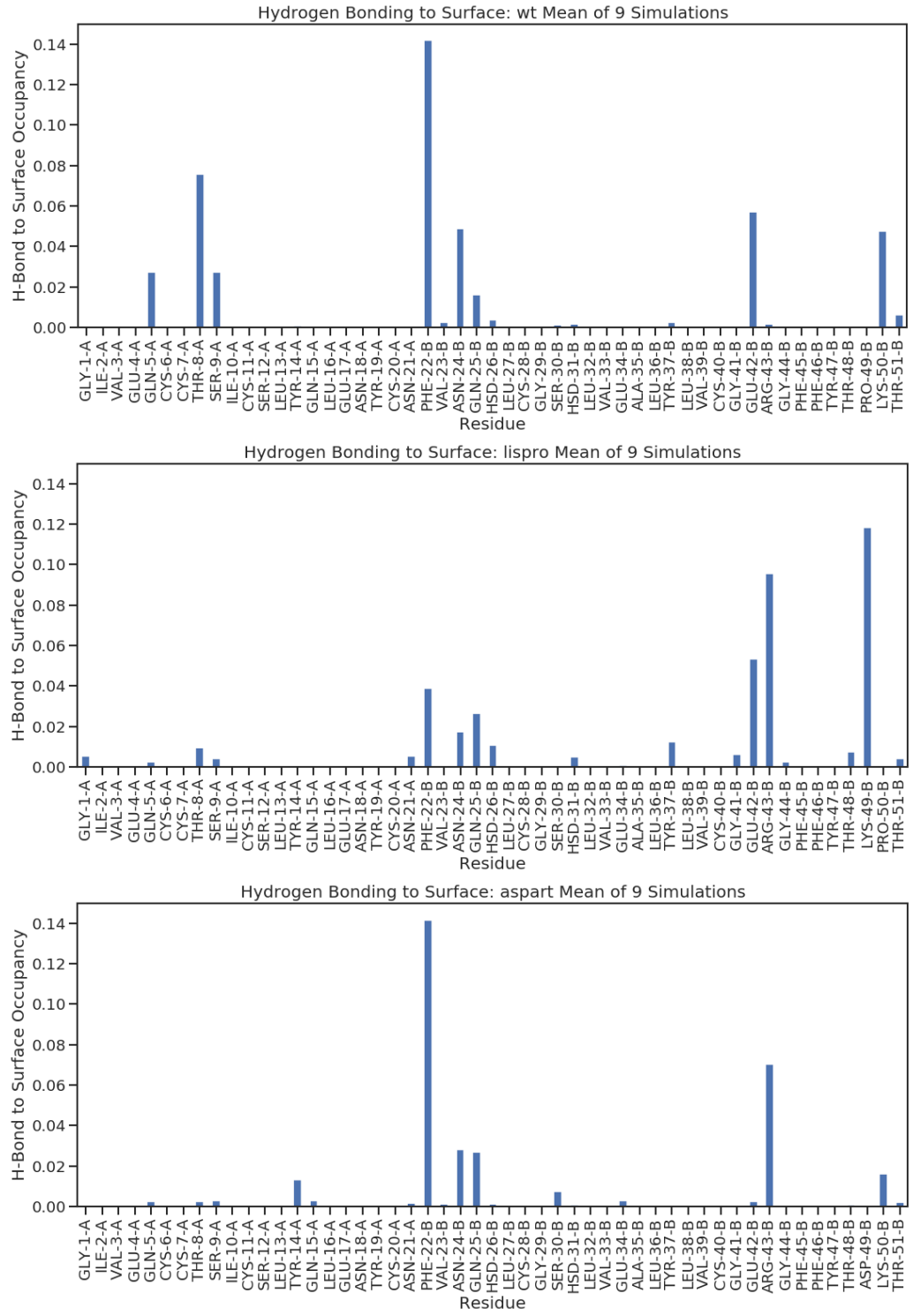


Figure 43: Hydrogen bonding occupancy (Top) Wild-type insulin; (Middle) Insulin lispro; (Bottom) Insulin aspart.

References:

Ahmad, Atta, et al. “Early Events in the Fibrillation of Monomeric Insulin”. Journal of Biological Chemistry. 280(52), 42669–42675 (2005). doi:10.1074/jbc.m504298200.

Angelo, J.M., et al. “Characterization of Cross-Linked Cellulosic Ion-Exchange Adsorbents: 1. Structural Properties.” Journal of Chromatography A. 1319, 46–56 (2013). doi:10.1016/j.chroma.2013.10.003.

Bakaysa, D. L. et al., “Physiochemical Basis for the Rapid Time-Action of LysB28ProB29-Insulin: Dissociation of a Protein-Ligand Complex”. Protein Sciences. 5(12), 2521–2531 (1996).

Brems, D. N. et al., Altering the Association Properties of Insulin by Amino Acid Replacement. Protein Engineering. 5(6) 527–533, (1992).

Case, D.A. et al., Amber 2021, University of California, San Francisco.

Derewenda, U. et al., “Phenol Stabilizes More Helix in a New Symmetrical Zinc Insulin Hexamer”. Nature. 338, 594–596 (1989).

Emami, F.S., et al. “Force Field and a Surface Model Database for Silica to Simulate Interfacial Properties in Atomic Resolution.” Chemistry of Materials. 26(8), 2647–2658 (2014). doi:10.1021/cm500365c.

Fang, K. Y. et al., “Incorporation of Non-Canonical Amino Acids into Proteins by Global Reassignment of Sense Codons”. Methods Molecular Biology. (2018).

Fang, K. Y. et al., “Replacement of ProB28 by Pipecolic Acid Protects Insulin against Fibrillation and Slows Hexamer Dissociation”. Journal of Polymer Science Part A Polymer Chemistry. 57(3), 264–267 (2019).

Frisch, M.J. et al., Gaussian 09, Revision B.01. Gaussian, Inc., Wallingford CT, (2016).

Gallivan, JP et al., “Cation-Pi Interactions in Structural Biology.” Proceedings of the

National Academy of Sciences. 96(17), 9459–9464 (1999). doi:10.1073/pnas.96.17.9459.

Gentzsch, W. “Sun Grid Engine: towards Creating a Compute Power Grid.” Proceedings First IEEE/ACM International Symposium on Cluster Computing and the Grid. (2001). doi:10.1109/ccgrid.2001.923173.

“Global Report on Diabetes”. World Health Organization. (2016).

Haeusler, R. A.; et al., “Biochemical and Cellular Properties of Insulin Receptor Signalling”. Nature Reviews Molecular Cell Biology. 19, 31–44 (2008).

Harvey, M. J., et al., “ACEMD: Accelerating Biomolecular Dynamics in the Microsecond Time Scale”. Journal of Chemical Theory and Computation. 5(6), 1632–1639 (2009). doi:10.1021/ct9000685.

Holleman, F. et al., Insulin Lispro. New England Journal of Medicine. 337(3), 176–183 (1997).

Humphrey, W. et al., "VMD - Visual Molecular Dynamics", Journal of Molecular Graphics. 14, 33-38 (1996).

Kelley, L.A., et al., “An Automated Approach for Clustering an Ensemble of NMR-Derived Protein Structures into Conformationally Related Subfamilies”. Protein Engineering, Design and Selection, 9(11), 1063–1065, (1996). doi:10.1093/protein/9.11.1063.

Lieblich, S. A., et al., D. A. “4S-Hydroxylation of Insulin at ProB28 Accelerates Hexamer Dissociation and Delays Fibrillation”. Journal of the American Chemical Society. 139, 8384–8387 (2017).

Mackerell, A. D., et al., “All-Atom Empirical Potential for Molecular Modeling and Dynamics Studies of Proteins.” The Journal of Physical Chemistry B. 102(18), 3586–3616 (1998). doi:10.1021/jp973084f.

Maier, J. A. et al., “ff14SB: Improving the Accuracy of Protein Side Chain and Backbone Parameters from ff99SB”. Journal of Chemical Theory and Computation. 11, 3696–3713 (2015).

MarvinSketch, MarvinBeans 18.29, ChemAxon https://www.chemaxon.com

Mayo, Stephen L., et al., “DREIDING: a Generic Force Field for Molecular Simulations”. The Journal of Physical Chemistry, 94(26), 8897–8909 (1996). doi:10.1021/j100389a010.

Nejad, M.A. et al., “Insulin Adsorption on Crystalline SiO2: Comparison between Polar and Nonpolar Surfaces Using Accelerated Molecular-Dynamics Simulations”. Chemical Physics Letters. 670, 77-83 (2017). doi:10.1016/j.cplett.2017.01.002.

Nejad, M.A. et al., “Insulin Adsorption on Functionalized Silica Surfaces: an Accelerated Molecular Dynamics Study”. Journal of Molecular Modeling. 24(4) (2018). doi:10.1007/s00894-018-3610-2.

Phillips, JC et al., “Scalable molecular dynamics on CPU and GPU architectures with NAMD”. Journal of Chemical Physics. (2020). doi:10.1063/5.0014475

Roe, D.R. et al., “PTRAJ and CPPTRAJ: Software for Processing and Analysis of Molecular Dynamics Trajectory Data”. Journal of Chemical Theory and Computation. 9(17), 3084–3095 (2013). doi:10.1021/ct400341p.

Wang, J. et al., "Automatic atom type and bond type perception in molecular mechanical calculations". Journal of Molecular Graphics and Modelling. 25 (2006).

Weltz, JS., et al. “Surface-Mediated Protein Unfolding as a Search Process for Denaturing Sites.” ACS Nano. 10(1), 730-738 (2015). doi:10.1021/acsnano.5b05787.

Wu, Di., “The Puckering Free-Energy Surface of Proline.” AIP Advances. 3(3) (2013). doi:10.1063/1.4799082.

Chapter 3: Mentoring and Other Experiences

While it may not be following canon to include activities outside of the (computer) lab in one's thesis, I have spent a significant amount of time mentoring, volunteering, and working as an intern, and these experiences have shaped my life and impacted others, so a depiction of my time at Caltech would be incomplete without them.

Mentoring

I have been extremely privileged to have had my mother as a mentor and model of educational and professional success, and I recognize that this privilege comes with the responsibility to help others as a mentor whenever possible. While an undergraduate, I spent a significant amount of time tutoring and mentoring high school students at an economically disadvantaged high school, so when the opportunity was presented for me to mentor college undergraduates while at Caltech, I eagerly accepted it. This began in the summer of 2020, when I was tasked with helping three bright young women to learn the basics of machine learning, and how it can be applied to protein engineering.

For each student I mentored, I designed a curriculum and adapted past work to create a project suitable for full-time study. This involved weekly scheduled meetings, intervening calls, and many daily exchanges to help maintain a smooth progression.

Two of the three students came not knowing how to program whatsoever. Indeed, when asked to log into the campus HPC cluster, one remarked that she could not find the login button on the website. This was an important reminder for me! After three years in a small bubble of Caltech graduate students, I was forced to stop assuming things about what people know and retool my communications. This took some work, especially since I was working with students a decade younger than me.

Over the course of the summer and fall working with these students, they each became proficient, then masters at using tensorflow, compute clusters, and cheminformatics / protein libraries. Working with three students in parallel was an interesting lesson as well in the varied styles by which different people learn.

In addition to helping these students learn about how computation can be applied to protein engineering, we also spent a significant amount of time discussing possible career paths and graduate school. One student I worked with presented our work at multiple conferences over the summer, and at one site, she was offered a spot in a competitive internship program at a leading pharmaceutical company. I am tremendously proud to have been in my own small way involved in helping this student to find a career path that excited her, doubly so because she was the first person in her family to graduate from college. I remain in touch with all three students and have continued to work with them after their fellowships formally ended, including helping to review and edit graduate school applications.

In addition to working with these undergraduate students, I have also assumed a primary role in training three new graduate students (all in the 2020-2021 school year) to use molecular dynamics, high performance computing resources, and to perform data analysis on the resulting experiments.

Leadership and Work Experiences

Internships

In early 2018, Jay Bradner visited Caltech and gave a great talk about his lab's work developing PROTACs. After the talk, I approached him to chat about how cheminformatics and machine learning might be applied to the space. After a few minutes discussing potential approaches, Jay offered to put me in touch with the head of bioinformatics at the Novartis Institutes for BioMedical Research (NIBR). A few months later, I started an internship at NIBR San Diego working with the head of structural biology to build tools for studying RNA structure. This work drew closely from the work I had done in the previous terms on VoxLearn, and would ultimately lead to the structural biology project I worked on in subsequent years, and a lasting friendship with the NIBR head of biologics. Another key part of this internship was the experience of working in a large company. Although NIBR San Diego operates semi-independently from NIBR and Novartis, with 600 employees on site, this was by far the biggest company I had worked for. There were many nice aspects of this environment (such as being on the beach every day by 4:00 PM), but ultimately I realized that this was not a match for my personality. This internship taught me that I need to work

somewhere where I can be involved in multiple aspects of any given project, and assume greater responsibility for the overall success of the work.

In the summer of 2020, I began my second internship, working for Vida Ventures. In the course of my time here, I helped the team to source, evaluate, and present several dozen deals, including working on two that the group funded. Working as part of a team of five investment professionals, this was a great opportunity to own all aspects of the workflow. Additionally, with the very high rate of work, this position brought me into contact with a tremendous number of entrepreneurs and technologies. This aspect was doubly valuable to me, as it allowed me to realize, one, that I deeply enjoy the social aspects of the work, building a network and rapport with fellow scientists and business people, and two, that working in VC would be a great finishing school before I returned to my long term goal of being an entrepreneur.

Leadership Positions

On my third day at Caltech, I attended an orientation event and met the then current head of the Caltech biotech club, who was soon to depart for a postdoc position. Seizing upon the opportunity, I quickly agreed with another new student seated with us that we would restart the club and start planning events for that fall. Over the past four years of leading and co-leading this group, I have planned and run events which brought executives, alumni, entrepreneurs, and financiers to campus. These events have scaled from large lecture halls to

intimate dinners, and provided a glimpse into a wide range of career paths outside of academia. I am proud of the time I committed to this group, as it helped many of my peers to form connections, get internships, and will hopefully soon lead to careers. This leadership position was a good learning experience for me as well, as it required me to practice delegating to others, as well as to learn how to motivate. This experience has also helped forge ties to the Catech community which will endure long beyond the time of my attendance.