Sequence-Specific DNA-binding Proteins: Protein Design, Structure Prediction, and Binding Prediction

Lilian McHugh

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2025

Reading Committee:

Frank DiMaio, Chair

David Baker

Barry Stoddard

Program Authorized to Offer Degree:

Biochemistry

University of Washington

Abstract

Sequence-Specific DNA-binding Proteins:
Protein Design, Structure Prediction, and Binding Prediction

Lilian McHugh

Chair of the Supervisory Committee:

Frank DiMaio
Department of Biochemistry

Sequence-specific DNA-binding proteins (DBPs) perform critical roles in biology and biotechnology, and have seen decades of effort to engineer, predict, and understand their functions. In this work, I present methods to design novel DBPs, predict the structures of protein-DNA complexes, and predict the binding specificities of structurally diverse DBPs. Made with custom computational methods, we screened over 100,000 designed DBPs and identified 44 that bound their intended targets with high affinity. Several of the designed DBPs are highly specific for their targets, as demonstrated by all-by-all cross-reactivity studies, mutation-scanning competition assays, and protein-binding microarrays. The designed DBPs bind consistently with their design models, as determined via interface ablation studies and crystallographic structure determination. For structure prediction, I developed and tested RosettaFold-NA (RFNA), the first end-to-end trained machine learning model that predicts the structures of any combination of protein and nucleic acids. RFNA accurately predicts about 30% of protein-nucleic acid complexes without sequence homology to the training set. For binding prediction, I curated a dataset of over 3000 DBPs with semi-manually-assigned DNA-binding domains and hundreds of thousands of corresponding experimentally-verified DNA target sequences. I fine-tuned RFNA and RoseTTAFold-allatom both on prediction of a binary binding / non-binding classification task and on prediction of distilled protein-DNA complex structures. In a retrospective analysis of design results, the resulting fine-tuned model is able to enrich for functional designed DBPs. Using a simulated annealing inference approach, the fine-tuned model can also predict DNA-binding profiles for validation set transcription factors with reasonable accuracy and efficiency.

For those who come after.

TABLE OF CONTENTS

Foreword

When we were wrapping up the original RFNA manuscript (chapter 3) and writing the first version of the DBP design manuscript (chapter 2), I was about half-way through my PhD. While that was a great deal of success for a graduate student to experience in just a couple years, I felt—and my advisors agreed—that I ought to have a project that was definitively mine. I was interested in a new design project with a focus on the flexibility of DNA, but my idea was too close to work being started by some undergraduates in the lab. Ultimately, I felt that working on another AI project to create a better DBP design filter would have the greatest impact both on the lab and the field.

I later became de-motivated by the gap between my original interest in protein design and the practical work of fine-tuning an AI structure prediction model for a non-structural task. Moreover, the intentional divide between my project and my colleague's work contributed to an extended period of self-isolation. In hindsight, I think I would have been both happier and more productive working on a project I was more passionate about and which was more collaborative with my peers. I often find myself comparing the work that I did in those first few years, in close collaboration with peers and mentors, with the work I did mostly alone in the later years. This comparison is obviously unfair, as are the feelings of inferiority and failure it inspires in me, but it comes up over and over again regardless.

Dear reader, do my choices to pursue impact over passion, and utility over creativity, and independence over collaboration, resonate with you? Have you felt the impostor syndrome, isolation, and sense of pointlessness that I have? If so, I encourage you to choose the path of kindness and togetherness.

The scarcity of funding and perceived precariousness of many scientific positions incentivize competition and independence, while the "novelty" standard incentivizes constantly doing something new instead of perfecting your existing work. However, the heart of science is and always will be community. No individual scientist, not even the most famous Nobel laureate, has an impact that can be measured without the context of their peers, mentors, and students. The strides we make toward learning, which is the core aim of all science, are always made by bringing together the work of many. We should not be ashamed of how our ideas and work depend on those of others—we should take pride in it.

Acknowledgements

First, I must of course acknowledge my advisors, mentors, colleagues, and collaborators, without whom this work would not exist. While this dissertation is published with a single author, in truth there are many, many people who contributed to the intellectual work presented here. Here is a non-exhaustive list, not in any particular order, of people who have contributed to or supported my scientific pursuits in graduate school: Frank DiMaio, David Baker, Barry Stoddard, Phil Bradley, Abhinav Nath, Luki Goldschmidt, Carson Adams, Gabi Reggiano, Marisa Brandys, Andrew Muenks, Joy Menten, Davi Nakajima An, Tabitha Tcheau, Guangfeng Zhou, Patrick Vecchiato, Tuscan Thompson, Jason Carmody, Adam Broerman, Andrew Favor, Andrew Kubaney, Avery Yang, Brian Coventry, Cameron Glasscock, Declan Evans, Liza Chernova, Enisha Sehgal, Han Raut Altae-Tran, Paul Kim, Paul Kwon, Riley Quijano, Thomas Schlichthaerle, Wei Chen, Yuliya Politanska, Beau Lonnquist, Emily Na, Rohith Krishna, Simon Mathis, Nate Corley, Minkyung Baek, Hanlun Jiang, Meghana Kshirsagar, Ivan Anischenko, Ian Humphreys, Pascal Sturmfels, Briar Huddy, Yang Hsia, Ryan Kibler, Florian Praetorius, Ajasja Ljubetic, Kelly Lee, Chip Asbury, Erin Kirschner, Sam Pellock, Kandise VanWormer, Ashley Vater, Kiera Sumida, Amir Motmaen, Cullen Demakis, Grace Hendricks, Jeremiah Sims, Marc Exposit, Naveen Jasti, Susana Vazquez Torres, Audrey O'Neill, Fernando Banales, Hugh Haddox, Christoffer Norn, Gyu Rie Lee, Lindsey Doyle, Zhe Li, Phil Leung, Justas Dauparas, Jue Wang, Preetham Venkatesh, Indrek Kalvet, and Xinting Li.

Of course, I also want to thank the people who got me this far. First, my parents, whose support for education throughout my life have helped shape who I am. Also, my brother Eric, for sharing with me many passions and acting as a sort of role model for me. Further, I want to thank my teammates and coaches from Science Olympiad, which I competed in from ages 12 to 18. From college, I want to thank Kate Creasey, who was the first scientist I really talked to about her research, and Jen Green, who pushed me to take scientific writing seriously. Many of these people were there to give me a push to excel when I might have otherwise made do with good enough.

Finally, I have to thank my partner Quinn and our two cats, Elias and Micah. Their love, affection, and support have made the past 4 years, in some ways, the best of my life.

Oh, and thank you, reader, for being interested in my work!

Chapter 1. Introduction

1.1 Biological significance of DNA-binding proteins

It is hard to overstate the importance of DNA-binding proteins (DBPs) in biology and biotechnology. The central dogma of molecular biology, the flow of information from DNA to protein, is centered around interactions between proteins and nucleic acids. DNA replication uses a variety of DBPs with a variety of functions. Essentially all of developmental biology, and most of cellular signaling, involve transcription factors, the poster children for sequence-specific DNA-binding. It would be both impossible and pointless for me to describe all of the types of DBPs and their importance here. Instead, I would direct you to any one of many reviews on the subject that have been written over the past 40+ years: [1–6]. For the rest of this introduction, I will cover the mechanics of DBPs that are important to understanding this dissertation.

1.2 Key principles of DNA-protein interactions

While there are many DBPs that bind to DNA with weak or no specificity and many more DNA-associated proteins that do not interact directly, this work focuses on direct interactions between sequence-specific DBPs and their cognate DNAs. A great deal of effort has been put into understanding how DBPs find and recognize their targets over the past decades. For more thorough coverage of these principles, I direct you to these excellent review articles on the subject: [7–9]

In many cases, our understanding of DNA recognition by DBPs derives from a combination of structure determination, mutation screening, and binding experiments. Many DBP families in nature recognize the same general consensus sequence across broad evolutionary time (with tailored individual preferences)[10], while others (notably C2H2 Zinc Fingers (ZFs) and TAL Effectors (TALEs)) are able to recognize a wide variety of sequences with only a few mutations.[11,12]

While the mechanisms encoding DNA recognition in DBPs are far from fully understood, there are a few core principles that we know of:

- “Direct readout”: interactions formed between the protein (usually sidechains) and the functional groups of the DNA in the major and minor grooves, which vary for each nucleotide. These are the most obvious and one of the most important sources of recognition, but they are not strictly required. Direct readout can be further broken down into types of interactions:
    - Sidechain-base hydrogen bonds, especially “bidentate” interactions between rigid groups with multiple hydrogen bond donors and/or acceptors. These are the easiest interactions to detect and engineer, but they require very precise placement of the sidechain in the interface.

    - Water-mediated hydrogen bonds, in which both the DNA and the protein form interactions with a stable or semi-stable water molecule in the interface. Each water-mediated hydrogen bond contributes somewhere between half and three-quarters as much to specificity as a direct hydrogen bond.[13]
    - Hydrophobic interactions, in which a non-polar sidechain or even just the carbon chain of a long polar sidechain interacts with a non-polar region of the DNA base, especially the methyl group of Thymine. This helps exclude water from regions of the interface where it would not form ideal hydrogen bonds, increasing the entropy of the system.

- "Indirect readout" or "shape readout" refers to any recognition of a DNA sequence that is dependent on the sequence-dependent conformational dynamics of DNA. We know that this is important for recognition because many DBPs are sequence specific but display few direct readout interactions in solved structures. Indirect readout can also allow DBPs to have sequence preferences extending outward from the physical interaction site. The exact mechanisms behind indirect readout are poorly understood, because modeling of DNA conformational dynamics is still extremely difficult despite many efforts to improve it.
    - For example, TATA-binding proteins, which recognize the AT-rich target site partially by binding in and widening the minor groove, prefer especially flexible DNA sequences.[14]
    - Another example involves an Arginine-rich tail, which binds more strongly to the narrower minor groove of certain DNA sequences.[15]
- Nonspecific interactions, which usually involve electrostatic interactions with the phosphate backbone as well as shape complementarity between the protein fold and the general shape of the DNA double helix.
    - Charge-charge interactions, where arginine and lysine sidechains of the protein are placed such that they can flexibly interact with the phosphate backbone.
    - Electrostatic complementarity, where the DBP fold has a feature that is complementary to the general shape and electrostatic surface of DNA, such as a positively-charged groove matching the curvature of the phosphate backbone
    - Helix dipole interactions, where the macro-dipole that forms in alpha helices due to the alignment of many hydrogen bonds is pointed toward the negatively-charged phosphate backbone. I understand these to do more to

guide the orientation of the protein relative to DNA than to provide affinity or specificity.[16]

- There are also several known types of recognition-driving interactions that do not fit neatly into any of these categories:
    - Shape-based group recognition, in which one or more protein sidechains pack tightly onto the shape of a particular DNA base. The best example of this is recognition of Adenine by an isoleucine sidechain in TALEs.[17]
    - Negative recognition, in which part of the protein packs against the DNA tightly in an orientation that would clash with a different nucleotide at the same position.[18]
    - Pi-pi stacking and cation-pi interactions, in which aromatic or positively-charged protein sidechains stack against the aromatic nucleobase with attractive quadrupole-quadrupole or quadrupole-charge interactions. Since any nucleotide could potentially form these interactions, specificity depends on both precise sidechain placement and the variable orientations preferred by different basepair combinations.[19]
    - Composite interactions where a single protein group can form multiple interactions with the DNA at once. For example, the CpG dinucleotide is commonly recognized by an Arginine sidechain which forms a bidentate hydrogen bond to the Guanine, a cation-pi interaction with the Cytosine ring, and hydrophobic interactions between its ethylene portion and the Thymine methyl group on the opposite strand.[20]

While I organized them into a few groups based on how specific they are for clarity, any of the above interactions could be formed more strongly with the recognized target, or equally for any DNA sequence. There is no set rule that can tell these cases apart, which is part of why we continue to develop new tools for modeling protein-DNA complexes.

To bind their preferred DNA target, DBPs need to exploit some combination of specific and non-specific interactions (which, as discussed above, cannot truly be separated from one another). To bind a target sequence with high affinity, a DBP necessarily must have some binding affinity for DNA in general, even if this comes just from a slight positive net charge. DBPs are also known to find their targets with a bind-and-search pattern, where the DBP first binds nonspecifically to DNA and performs a linear search along the DNA for the target sequence. During this process, the protein can repeatedly detach and reattach to the DNA. This process implies that there can be some affinity provided by

less-specific interactions with the DNA, which then help guide the formation of more-specific but harder-to-find interactions.[21]

Additional discussion of for which of these features are accessible with current methods and which we think to be most important is covered in chapter 2.

1.3 Definition and quantification of sequence-specificity

As this work is defined by its focus on sequence-specific DBPs, I find it necessary to clarify what exactly I mean when I say a DBP is "specific" to its target. In all cases, specificity of binding the target must be defined *relative* to binding something else; it cannot be quantified as an absolute (although I will still try).

A rigorous definition might therefore be "the ratio of the binding affinity to the target sequence compared to binding affinity to the off-target sequence." However, this definition is only rigorous if you are comparing to exactly one off-target sequence, which is rarely the case in any practical experiment. Moreover, it specifies binding affinity, which is laborious to measure experimentally and almost impossible to predict computationally. Ignoring both those issues, it also isn't clear whether the binding to the target and off-target are being measured independently or in competition with one another.

A more practical and flexible definition might be "the ratio of the binding signal observed for the target sequence compared to the binding signal observed for the off-target sequence." This definition, which I would call "cross-reactivity," is in fact the one used to visualize specificity in Figures 2.8 and 2.18.

A very similar definition, used for the experiment shown in Figure 2.9, is the "non-competition" definition: "the ratio of binding signal observed to the target while an off-target competitor sequence is present compared to the binding signal observed to the target while the same sequence is present as a competitor." This definition, as you can probably tell, is fairly specific to that particular assay. However, if you stretch it, you can extend it to create a convenient definition of absolute specificity: "the number of base pairs of the target at which any single mutation significantly reduces binding."

This "number of base-pairs" definition can be used to approximate how often you would expect the target sequence to appear by chance in a genome of a certain size. This definition is often haphazardly used to describe the specificity of Crispr-Cas systems because their guide RNAs match a fixed number of base-pairs of DNA.[22] However, it is not a true quantitative definition and it will not accurately predict off-target effects when the DBP is used in living cells. This is clearly shown by the off-target cleavage and editing effects that constantly plague applications of Crispr-Cas systems.[23]

Another commonly-used definition of specificity is the position probability matrix (PPM), which is convenient for visualizing and searching for likely binding sites. This is defined

as “the probability of finding each possible base at each position along a binding site, with the assumption that the effect of each position is independent.” This underlying assumption of the PPM is demonstrably incorrect, but it is a format that can be used for a variety of different experiments. For example, if you measure the binding affinity of a DBP to a series of different DNA sequences, you can use a Boltzmann distribution to compute the probability of each sequence, separate these by position, and construct a PPM. If you just have a large number of DNA sequences with enriched binding signal over a background of broad sequence space coverage, as is the case in several high-throughput experiments, you can align the hits and add up the counts of bases at each position to construct a position frequency matrix, then divide by the total count to get a PPM. More than anything else, the popularity of the PPM likely comes from the convenient visual it provides in the form of sequence logos, such as Figure 1.1.

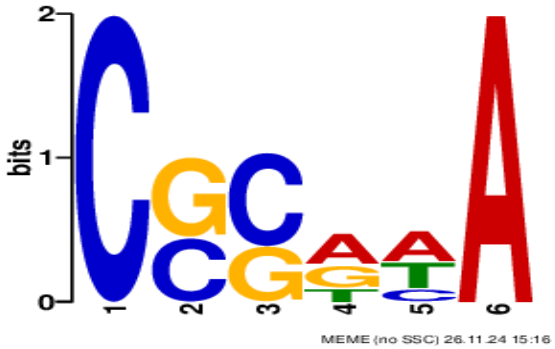


Figure 1.1 Example of a sequence logo for a PIM. The positions of the DNA motif are along the x-axis, and the y-axis is in bits of information, i.e. the log base 2 of the probability divided by the background probability

These logos, invented in 1990 to display consensus sequences[24], are usually made from position information matrices (PIMs). Briefly, the total information content of each site is computed based on the log-ratio of the probabilities versus a background distribution, then the information for each base is computed as the product of its probability and its site’s total information. The minimum value is 0, for sites where the distribution matches background, while the maximum value is usually 2 bits for positions with probability 1. In addition to being a pleasing visual, the PIM provides another opportunity for an absolute specificity definition: “the total information content of the PIM,” or roughly “the extent to which the binding motif constructed as a PPM for the protein differs from the expected random motif.” This measure will either be affected by the total number of positions considered in the constructed motif, or you can average it across the positions in the motif. Either way, it contradicts the core assumption needed for a PPM to make sense by trying to add together the information from multiple positions that must necessarily be independent. While the total information content, or the equivalent “cross-sum of per-position-fidelity and total number of positions with some fidelity” definition suggested by

Dr. Stoddard [informal communication], could theoretically predict how often a preferred site would appear in random DNA, I would avoid this definition. In many cases, you would be better off choosing some cutoff to consider the DBP “specific” at each position and using the “number of base-pairs” definition.

Another definition I use for DBP specificity is the “binary binding” definition. For this definition, we assume that a given DBP with either bind or not bind to any given DNA sequence. This definition is convenient in its simplicity – any of the above definitions can easily be simplified to convert to this one by choosing a cutoff for binding, and it does not require explicit comparison between multiple sequences. The definition is an obvious oversimplification of the reality of molecular recognition, but it can capture a good portion of the information—high-throughput experiments often identify the binding motif based on highly enriched outliers, which can be called the “binders”, while calling the background unenriched or random sequences “nonbinders.” This definition is used for much of my work in chapter 4.

A final definition I use for DBPs is the “specific or nonspecific” classification. While all DBPs will certainly have some degree of preference, we use this definition to identify and classify DBPs that have a strong preference for a clearly-defined motif or which have high scores in the “cross-reactivity” or “number of base pairs” quantifications. This definition is useful for evaluating the success of our work in DBP design, as this is the only definition that really defines what a “sequence-specific DNA-binding protein” *is* and what *is not*.

Ultimately, the definition of specificity you choose will be dependent on the particular experiment you are using to characterize a DBP’s specificity or on the computational method you are using to predict it. I use several of these definitions at various points throughout this work.

1.4 Problem definitions and specific aims

As stated in the title, the goal of this work is to develop methods to design sequence-specific DBPs, predict their structures, and predict their binding specificities (using the PPM or binary definitions). We set out with this goal because there is clearly both interest and difficulty in engineering DBPs as well as studying their structures and functions. These three goals are inter-connected: structure and function prediction have the potential to greatly accelerate and enhance design work, while the results of design work can help inform and evaluate efforts in structure and especially function prediction.

For clarity, I state here the specific aims of the three major sections of this work:

Aim 1: Develop a method to computationally design DBPs with novel structures, interfaces, and/or binding preferences, and which can be used to advance genetic engineering. My efforts towards Aim 1 are described in chapter 2.

Aim 2: Develop a method to computationally predict the structures of protein-nucleic acid complexes, especially for sequence-specific DBPs and including designed DBPs. My efforts towards Aim 2 are described in chapter 3.

Aim 3: Develop a method to computationally predict the binding specificities of DBPs, in a way that incorporates awareness of structural information and can transfer between protein families. My efforts towards Aim 3 are described in chapter 4.

Chapter 2. Design of sequence-specific DNA-binding proteins

Primary citation for this chapter:

Glasscock CJ*, Pecoraro R*, McHugh R*, et al. “Computational design of sequence-specific DNA-binding proteins.” *Nat. Struct. Mol. Bio.,* 2023.

Author’s note: this chapter describes a multi-year collaborative effort between me, Baker lab postdoc Cameron Glasscock, and fellow graduate student Robert Pecoraro, with help from many others. Our work was published as a pre-print on September 21, 2023 and has recently been accepted for publication in Nature Structural & Molecular Biology (in press). For clarity and convenience, I am reproducing here large sections of that work and providing some additional context where I feel it is helpful.

2.1 Introduction to chapter 2

2.1.1 Why computationally design DBPs?

Nature employs a wide diversity of DNA-binding protein (DBP) domains for targeting specific sequences[25], which are often structurally coupled to each other and to effector regions conferring enzymatic, binding, and regulatory functions.[26,27] Despite intensive study and substantial progress in the *in silic*o prediction of DNA binding specificities from complex structures[28], the DNA binding affinity and specificity of natural proteins remain difficult to predict[7], and the high free energetic cost of desolvating the highly polar DNA surface presents a challenge to the *de novo* design of DBPs. For these reasons, while computational design has displayed considerable recent success in generating binders to arbitrary protein structures[29], mostly at hydrophobic patches, computational approaches for DBP engineering have been limited to redesigning interfaces of existing native protein-DNA complex structures.[30–34] These efforts have been constrained by the rigid geometry of the starting scaffold shape and orientation relative to DNA[35], which restrict the possible target sequences that can be recognized.[36]

A more general solution to generating compact, customizable DBPs would enable modular, geometrically precise, and deliverable tools and be highly complementary to the state-of-the-art in gene regulation, gene editing, and nucleic acid diagnostics which primarily employ Cys_2His_2 zinc finger domains,[37] transcription activator-like effectors,[38] and CRISPR-Cas.[39] While these tools have proven powerful, each has limitations: ZFs can be laborious to engineer, and the size of TALE and CRISPR-Cas systems complicates their delivery in therapeutic applications; CRISPR-Cas systems also require an extra guide RNA component, and target sites are constrained by protospacer adjacent motif (PAM) requirements.[39] These systems will undoubtedly continue to be improved, but their constrained backbone topologies can limit precise control of interaction specificity and close integration with diverse effector domains.

Author's note: at the outset of this work, our personal understanding of some aspects of DBP function was lacking. For instance, we only really grasped the importance of rigid interface orientations and how they help confer binding specificity after *noticing differences between our own specific and non-specific designs. Indeed, we were largely motivated to work on this project not by the impact our designs would have on the DBP engineering field at large, but instead by what we could learn from the process of applying existing binder design tools to a new, more difficult class of target.*

2.1.2 Principles of DBP design

We reasoned that it would be possible to achieve DNA sequence recognition using small, compact proteins by sampling a wide variety of structures and binding modes to find those that are optimal for targeting specific sequences of interest. Our labs previously developed a general method for designing specific protein binders to arbitrary protein targets based on this concept[29], but sequence-specific DNA binding requires overcoming several additional challenges:

(1) Binding the DNA double helix, with major and minor grooves, requires sufficient shape complementarity with the DNA backbone to precisely position specific amino acid residues to interact with the DNA base edges.

(2) Recognition of DNA sequences requires distinguishing between the subtle differences between individual atom placements among the four bases[18,40], which alter the landscape of potential molecular contacts.

(3) In contrast to designed protein-protein contacts mostly mediated by orientation-agnostic hydrophobic patches[29], the majority of accessible DNA base atoms require hydrogen bond interactions with polar sidechains for specific recognition.[41] Not only are polar interactions harder to model accurately, both in terms of geometry and energy, but the longer polar sidechains have considerable conformational flexibility. Both of these factors make structure modeling more difficult and increasing opportunities for off-target base interactions through alternate sidechain rotamer conformations.

To address these challenges, we formulated a set of design principles (Fig. 2.1A) and sought to develop a design pipeline implementing them in the context of small helical DBP domains that target short DNA sequences (Fig. 2.1, B to G).

Author's note: For much of the time I worked on this project, our design philosophy relied heavily on the following process: 1) Stare at models of native DBPs in PyMol until you notice something that is different from our designs. 2) Find a way to quantify what you've observed, and check if the trend holds.

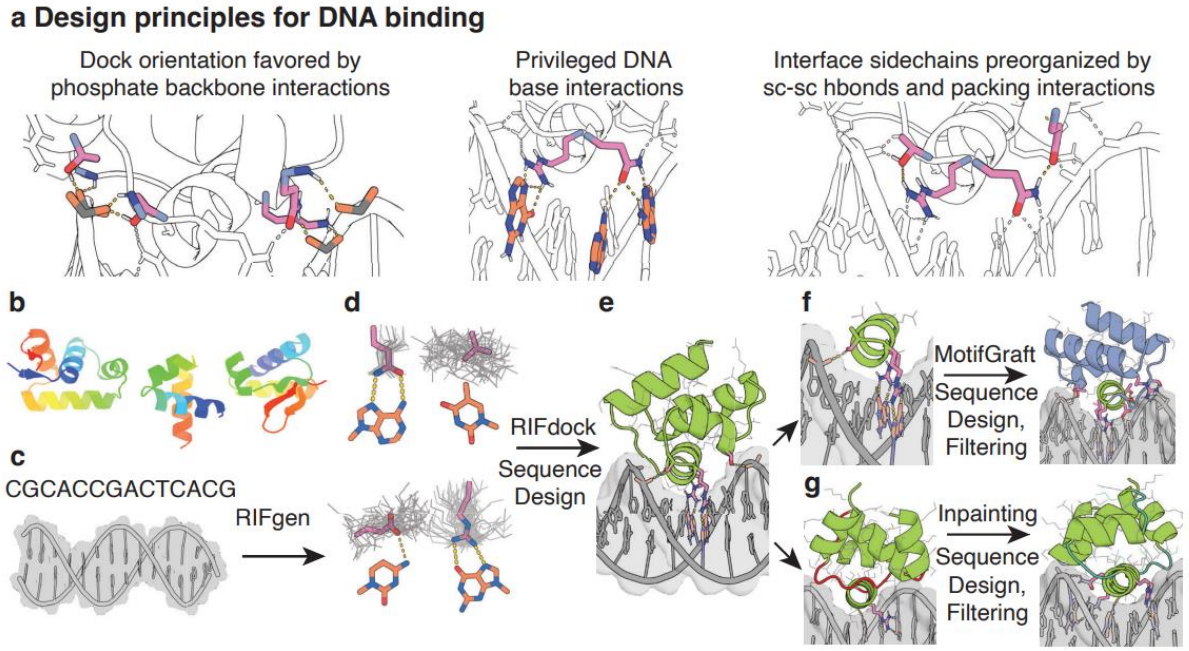


Figure 2.1 Overview of the DNA binder design pipeline (**A**) Design principles for design of sequence-specific DBPs. (**B**) HTH backbone scaffold library generated from metagenomic sequences. (**C**) DNA target, starting with either a specific nucleotide sequence modeled as B-DNA or a DNA crystal structure. (**D**) Generation of RIF (gray) to form base-specific hydrogen bonds and hydrophobic packing interactions. Example rotamers (pink) are generated for nucleotide bases (orange; clockwise from upper left: adenine, thymine, guanine, cytosine). (**E**) Docking of scaffolds onto the RIF to identify seed interactions and placements with base-specific contacts, followed by sequence optimization of the DNA-scaffold interactions using Rosetta or LigandMPNN-based sequence design and Rosetta modeling. (**F**) Recognition helices making multiple favorable interactions with the target are extracted from first round designs, and grafted onto the scaffold library, followed by further rounds of interface sequence design and filtering for favorable interactions. (**G**) Inpainting of the protein loops (red) results in new connecting loops (teal) between the helical portions of the design, followed by further rounds of interface sequence design and filtering.

2.2 Computational methods for DBP design

2.2.1 Protein scaffolds and DNA targets

For design challenge one, we hypothesized that interactions with the phosphate backbone, such as the backbone amide-mediated hydrogen bond interactions with DNA phosphate oxygens that are frequently observed in native DBP structures, could enable precise placement of designed scaffolds such that residues designed to make specific base contacts interact in the intended geometry. We reasoned that such satisfaction of the hydrogen bond requirements of the DNA backbone phosphates and the DNA bases would substantially constrain viable design geometries. We hypothesized that the helix-

turn-helix (HTH) DNA binding domain would be a good candidate for computational DNA binder design as it is relatively small, compact, and is capable of making direct contacts with DNA through a recognition helix within the DNA major groove.[42]

To generate a library of small (< 65 amino acid) and structurally diverse scaffolds, we took advantage of the vast amount of metagenome sequence data and the accuracy of deep learning-based protein structure prediction, shown in Figure 2.3.

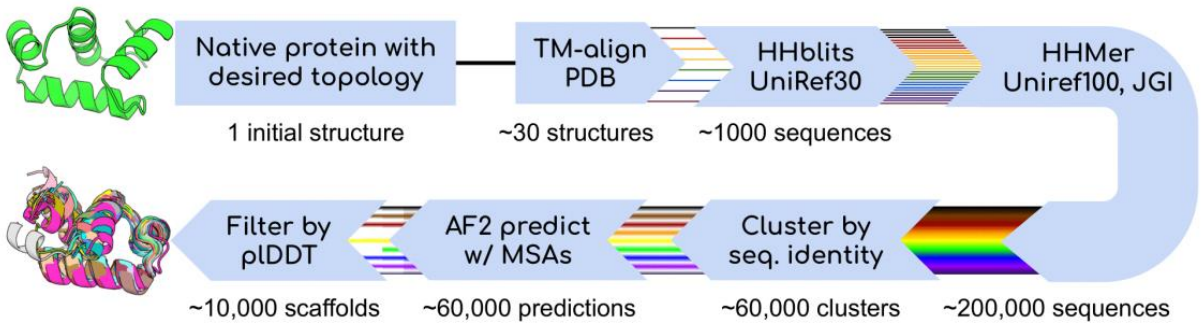


Figure 2.2 Visualization of scaffold library generation. Scaffolds deposited in the PDB with structural similarity to selected template backbones (PDB IDs: 1L3L[43] and 1PER[44]), were identified using TM-align.[45] Amino acid sequences of identified protein scaffolds were used as seeds to generate multiple sequence alignments (MSAs) using an HHBlits[46] search of the UniRef30 database.[47] Resulting MSAs were used for HMMer[48] searches of the JGI metagenome protein sequence databases[49] and the Uniref100 database.[47] HMMer search results were clustered to < 70% sequence identity using MMSeqs2[50] and MSAs were generated from each clustered sequence using HHBlits. AlphaFold2[51] was used to predict structures for each sequence using the generated MSAs. Resulting scaffolds were filtered for high confidence AlphaFold2 pLDDT scores, TMscore to the input backbone templates, and Rosetta score. Scaffolds of specific topologies were supplemented with additional AlphaFold2-predicted structures of transcription factor sequences identified from bacterial metagenomes using DeepTF.[52] PSSMs were generated for each scaffold using PSI-Blast[53] and custom code for use as constraints of Rosetta design.

Author's note: at one point, I was convinced that the reason HTH scaffolds would be better than other scaffolds had to do with the orientation of the electrostatic dipole that forms along the axis of the alpha helix. Figure 2.4 shows an example highlighting the difference.

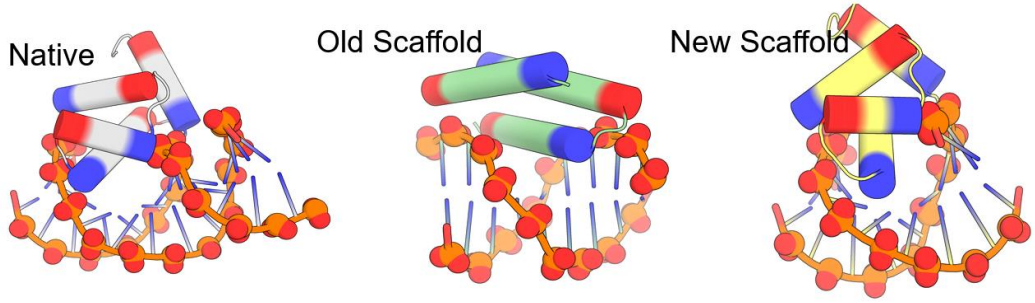


Figure 2.3 Comparison of Native HTH, non-HTH scaffold, and HTH scaffold with DNA. *Alpha helices are shown as cylinders with N-terminus in blue and C-terminus in red.*

We designed binders using a total of 15 DNA targets, listed in Table 2.1. The original targets (A, B, N, O) were selected because there were solved structures of native HTH-like DBPs bound to them. I generated the rest of the targets by hand, with the goal of covering a diverse range of sequence space, with some redundancy. Specifically, I chose twelve 10-basepair sequences such that each of the 64 possible trinucleotide sequences appeared in the set exactly three times. Then, a GC dinucleotide was appended to either end of each sequence, to improve duplex stability. I included redundancy in the hopes that cross-reactivity patterns could provide some insight into the particular binding motif for any successful designs. In practice, this redundancy was not helpful, although we did observe cross-reactivity for many designs.

Models of DNA duplexes for each target were generated by either (1) using the DNA portion of PDB structures 1BC8, 1YO5, 1L3L, 2O4A, or (2) using the software X3DNA[54] to generate ideal B-form DNA, followed by a constrained Rosetta relax of the DNA structure.

Table 2.1 Sequences of DBP design target DNAs

Name (PDB ID)	Sequence (forward / reverse strands)
A (1YO5)[55]	TAGCAGGATGTGT / ACACATCCTGCTA
B (1L3L)[43]	GCAGATCTGCACATC / GATGTGCAGATCTGC
C	CGACACCTGACGCG / CGCGTCAGGTGTCG
D	CGCTATCCAGAGCG / CGCTCTGGATAGCG
E	CGCGATGCTTCTCG / CGAGAAGCATCGCG
F	CGGCTGGATTACCG / CGGTAATCCAGCCG
G	CGAGAACATAGTCG / CGACTATGTTCTCG
H	CGGGGAAACGCCCG / CGGGCGTTTCCCCG
I	CGCCCAAAGCCGCG / CGCGGCTTTGGGCG
J	CGGAGGTAATGACG / CGTCATTACCTCCG
K	CGCACCGACTCACG / CGTGAGTCGGTGCG
L	CGGCCCTTTGCGCG / CGCGCAAAGGGCCG
M	CGCCGTTAGTGTCG / CGACACTAACGGCG
N (1BC8)[56]	TACCGGAAGTT / AACTTCCGGTA
O (2O4A)[57]	GCTAATATATGC / GCATATATTAGC

Author's note: for any future DBP design work, I would highly suggest picking target sequences that are either biologically relevant based on their locations in or around existing genes and promoters, or which are specifically rare in genomes of interest. For this work, arbitrary sequences were fine because our goal was to demonstrate a new method. However, the designs would be more practically useful if the targets were selected more carefully.

2.2.2 Interface design

We docked the scaffolds against specific DNA target structures seeking to maximize the potential for specific sidechain-base interactions (Fig. 2.1, B to D). To do this, we extended the RIFdock approach[29] to protein-DNA interactions (details below), which finely samples many possible *de novo* docks for each scaffold. RIFdock begins by enumerating a large and comprehensive set of disembodied sidechain interactions, called a Rotamer Interaction Field (RIF), that make favorable interactions with the desired target. We focused RIF generation on polar and nonpolar interactions with nucleotide base atoms in the major groove of the DNA target, with an emphasis on protein sidechain-DNA base hydrogen bonding interactions that are statistically more probable in native protein-DNA complexes.[58] In RIFdock, we constrained the RIF DNA base-specific interactions to the HTH recognition helix to find placements with both mainchain-phosphate hydrogen bonds and base-contacting RIF sidechains, resolving design challenge 1.

RIFdock was allowed to target along the entire stretch of each target sequence. The RIF docking method performs a high-resolution search of continuous rigid-body docking space. RIF docking comprises two steps. In the first step, ensembles of interacting discrete sidechains (referred to as 'rotamers') tailored to the target are generated. Polar rotamers are placed on the basis of hydrogen-bond geometry whereas apolar rotamers are generated via a docking process and filtered by an energy threshold. Rotamers were only calculated for nucleotide base atoms in the major groove of the DNA target. All the RIF rotamers are stored in ~0.5 Å sparse binning of the six-dimensional rigid body space of their backbones, allowing extremely rapid lookup of rotamers that align with a given scaffold position.

To enrich for canonical protein-DNA hydrogen bond interactions, rotamers of ARG, GLN, and ASN forming bidentate hydrogen bonds with G and A bases were extracted from the PDB, clustered by RMSD, aligned to the DNA target at all G and A positions, and added to the RIF as hotspot residues.

To facilitate the next docking step, RIF rotamers are further binned at 1.0 Å, 2.0 Å, 4.0 Å, 8.0 Å and 16.0 Å resolution. In the second step, a set of scaffolds is docked into the produced rotamer ensembles, using a hierarchical branch-and-bound search strategy. Starting with the coarsest 16.0 Å resolution, an enumerative search of scaffold positions is performed: the designable scaffold backbone positions are checked against the RIF to determine whether rotamers can be placed with favorable interacting scores. All acceptable scaffold positions are ranked and promoted to the next search stage. Each promoted scaffold is split into 26 child positions in the six-dimensional rigid-body space, providing a finer sampling. The search is iterated at 8.0 Å, 4.0 Å, 2.0 Å, 1.0 Å and 0.5 Å resolutions. All RIF docks were required to utilize at least 1 hotspot residue to be saved as an output.

Author's note: We tested variations of RIFdock with several other additions, including control of helix-major-groove-depth and proximity of helix N-terminal caps to phosphate groups, but none of them showed improvement over the method described above.

2.2.3 Protein sequence design

To address design challenge 2 – recognizing specific DNA bases – we used either Rosetta-based sequence design or an extended version of the deep learning–based ProteinMPNN sequence design software[59] (Fig. 2.1E, method detailed below). The ProteinMPNN graphical model generates amino acid sequences purely based on protein backbone coordinates, and a recent extension to incorporate ligand and DNA atoms in the interaction graph, called LigandMPNN.[60] While the Rosetta-based sequence design protocol was constrained by a position-specific scoring matrix (PSSM) for each scaffold, LigandMPNN was purely based on the structure of the designed complex. To reduce the computational cost of full sequence design on the millions of generated scaffold docks for each target site, we first repacked only the RIF sidechain residues in the context of the target to remove potential clashes between designed sidechains. Docks for which good protein-DNA interactions could be achieved without sidechain clashes were then subjected to multiple iterations of full sequence design, alternating with Rosetta backbone relaxation to maximize complementarity to the target sequence. We generated 200,000–300,000 designed complexes per target.

Rosetta-based interface sequence design

A stripped-down version of the Rosetta score function was used to roughly design the interface of RIF dock outputs.[29] This step was primarily used to replace clashing residues before evaluating for design potential. Specifically, fa_elec, lk_ball [iso, bridge, bridge_unclp], and the _intra_ terms were disabled. All that remained were Lennard-Jones, implicit solvation and backbone-dependent one-body energies (fa_dun, p_aa_pp, rama_prepro). Additionally, flags were used to limit the number of rotamers built at each position.

After the rapid design step, the designs were minimized twice: once with a low-repulsive score function and again with a normal-repulsive score function. Rosetta ΔΔG and contact molecular surface were then calculated on the roughly designed interface. A maximum likelihood estimator was used to give each predicted design a likelihood that it should be selected to move forward. A subset of the docks to be evaluated were subjected to the full sequence design, and their final metric values calculated. With a goal threshold for each filter, each fully designed output can be marked as pass or fail for each metric independently. Then, by binning the fully designed outputs by their values from the rapid trajectory and plotting the fraction of designs that pass the goal threshold, the probability that each predicted design passes each filter can be calculated. From here, the probability of passing each filter may be multiplied together to arrive at the final probability of passing

all filters. This final probability can then be used to rank the designs and pick the best designs to move forward to full sequence optimization. Note that the rapid design protocol here is used merely to rank the designs, not to optimize them; the original docks are the structures carried forward.

These docked conformations passing the rapid design protocol were further optimized to generate shape- and chemically-complementary interfaces using a Rosetta FastDesign protocol, alternating between sidechain rotamer optimization and gradient descent–based energy minimization. Design was performed with a sequence profile constraint based on an MSA of the originating native scaffold sequence and cross-interface interactions upweighted to maximize contacts and shape complementarity. We did not allow Rosetta to repack or relax the DNA target during the design procedure. A python script was implemented to automatically carry out rapid design evaluation, pre-emption, and full sequence design. Computational metrics of the final design models were calculated using Rosetta, which includes ΔΔG, hydrogen bonds to base atoms, and contact molecular surface, among others, for design selection. ProteinMPNN was used to redesign non-interface residues in the final design step, before AF2 monomer validation.

LigandMPNN-based sequence design

LigandMPNN was used for sequence design in the context of DNA. The network was used to optimize the protein sequence for given protein-DNA complex structures during design, whereby amino acids were determined autoregressively by the identity and location of neighboring protein and DNA residues. When the full protein sequence was determined, it was threaded onto the input protein scaffold.

As in the above Rosetta-based interface sequence design protocol, the designs were minimized with a low-repulsive score function and again with a normal-repulsive score function, and Rosetta ΔΔG and contact molecular surface were calculated on the roughly designed interface. A maximum likelihood estimator was used to pre-empt design of poor docks as described in the above Rosetta-based sequence design protocol.

A python script was implemented to automatically carry out MPNN sequence design, rapid design evaluation, pre-emption, and Rosetta Relax. LigandMPNN temperatures of 0.2–0.3 were used earlier in the design process to increase the variability of amino acid sequences, while a temperature of 0.1 was used later to determine the more probable sequences. Key residues making base-specific hydrogen bonds with DNA atoms were fixed in later stages of the pipeline to encourage the design of supporting residues.

2.2.4 Design evaluation

From our large set of designs, we selected those with the most favorable free energy of binding (Rosetta ΔΔG), contact molecular surface area[29] and interface hydrogen bonds, the fewest interface buried unsatisfied hydrogen bond donors and acceptors, and with bidentate sidechain-base hydrogen bonding arrangements frequent in the Protein Data Bank (PDB) (more detail below).

AlphaFold2 (AF2) predictions

Following selection based on the above criteria, and clustering by sequence identity, the monomeric structures of the hundreds to thousands of remaining designs for each target were predicted based on their sequences using AF2, and designs that deviated from their original design models were discarded. The remaining predicted monomer structures were superimposed onto the design complex by alignment on the interface residues of the original design and relaxed with Rosetta in the context of the DNA.

AF2 structures were produced using the single sequence of each design. AF2 was run with model 1 and 12 recycles for each design. C-alpha RMSD of the AF2 structures to each respective design model were calculated. AF2 structures were superpositioned onto the DNA target using the backbone coordinates of interface residues within 8 Å of the DNA target. A fixed backbone Rosetta FastRelax was performed on each superpositioned complex and all relevant metrics were calculated on the final superpositioned design model.

Rosetta metrics details

Designs were filtered after each sequence design step and after superimposition of AlphaFold models for those with the most favorable free energy of binding (Rosetta ΔΔG), contact molecular surface area[29] and interface hydrogen bonds, the fewest interface buried unsatisfied hydrogen bond donors and acceptors, and those containing bidentate sidechain-base hydrogen bonding arrangements frequent in the PDB, including bidentate interactions of ARG-G, GLN-A, and ASN-A. Designs were additionally filtered for those with a high RotamerBoltzmann score (see below) among ARG, LYS, GLN, or ASN residues forming hydrogen bonds with bases (max rboltz RKQE) and those with a high median RotamerBoltzmann (median rboltz) score of all residues forming hydrogen bonds with bases.

Pre-organization

To address design challenge three – precise geometric sidechain placement – we hypothesized that specificity and affinity would be improved in designs with highly preorganized interface sidechains. We reasoned that preorganization would be especially important for long polar sidechains with many possible conformations. We achieved

preorganization through sidechain (sc)-sc hydrogen bonding and assessed it using the Rosetta RotamerBoltzmann calculation.[61] By selecting only designs with native-like preorganization of key contacts, we aimed to achieve the level of precision required for specific DNA binding.

The Boltzmann probability of finding a given rotamer in a specific state was evaluated using the RotamerBoltzmannWeight filter in Rosetta.[61] The RotamerBoltzmann score approximates preorganization of a given residue in the unbound state. All amino acid residues forming hydrogen bonds with DNA base or phosphate atoms were evaluated by this metric, which was calculated on the protein monomer in the unbound state. The metric was estimated by fixing neighboring sidechains and assessing the Boltzmann probability distribution on rotamers accessible by the sidechain of interest. In order to increase the likelihood of a given rotamer in the protein-DNA complex, designs with lower RotamerBoltzmann scores (a score of 0 implies the rotameric state is unpopulated and a score of 1 implies the state is the only populated state) were preferentially chosen, as known native protein-DNA crystal structures tend to contain preorganized amino acid residues, as shown in Figure 2.5.

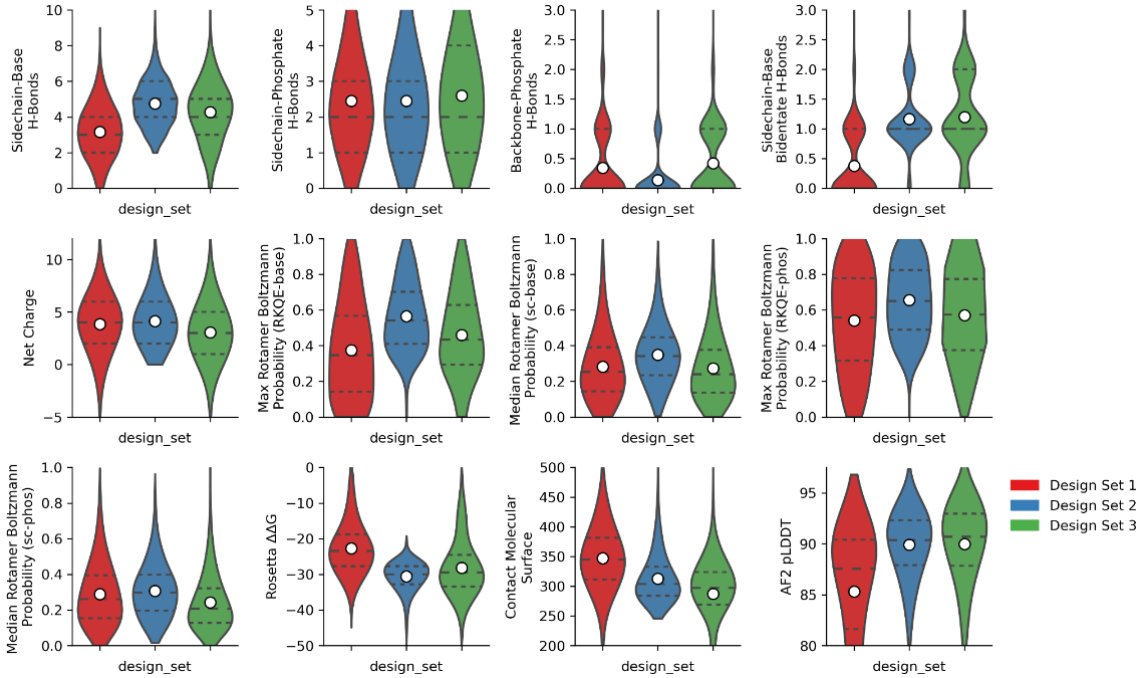


Figure 2.4 Distribution of metrics calculated on each ordered design set. Metrics were calculated on all designs after superposition of the AlphaFold2 predicted monomer onto the initial design complexes. Violin plots show lower quartile, median, and upper quartile (dashed lines). White circles represent the mean of each distribution. Design set 1 was produced using Rosetta sequence design and motif grafting, design set 2 was produced using LigandMPNN sequence design and motif grafting, and design set 3 was produced using LigandMPNN and Inpainting. White dots overlaid on violin plots represent the mean of each distribution while lower, middle, and upper lines represent first quartile, mean, and third quartile respectively.

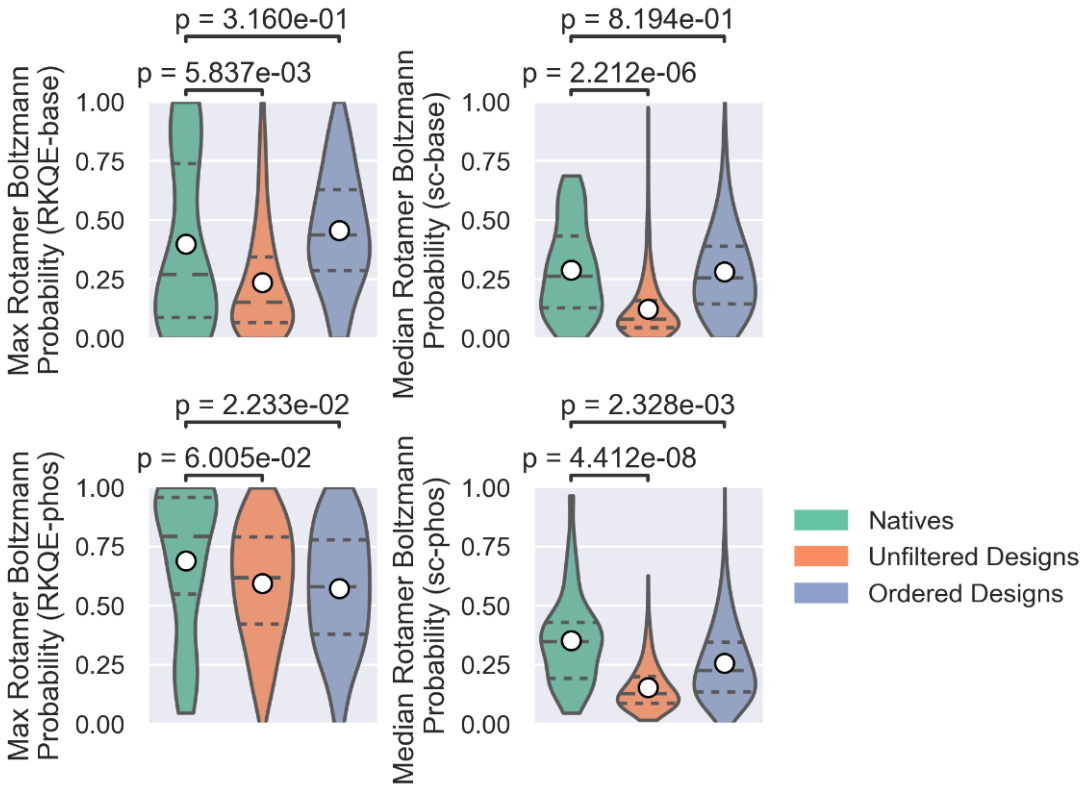


Figure 2.5 Native protein-DNA structures are enriched with highly preorganized residues forming hydrogen bonds with base atoms and the DNA phosphate backbone. The Rosetta RotamerBoltzmannWeight metric was calculated on natives (n=41 examples), unfiltered designs (n=2000 examples), and ordered designs (n=124,924 examples) as a proxy for sidechain preorganization. **Top: Left**, Maximum RotamerBoltzmann probability among longer RKQE sidechains. **Right**, Median RotamerBoltzmann probability for each design among all sidechains forming hydrogen bonds with bases. Among sidechains forming hydrogen bonds with base atoms, natives were significantly more preorganized than unfiltered designs. Ordered designs were filtered on the RotamerBoltzmann metric to achieve a distribution similar to natives. **Bottom: Left**, Maximum RotamerBoltzmann probability among RKQE sidechains. **Right**, Median RotamerBoltzmann probability among all sidechains forming hydrogen bonds with the phosphate backbone. Among sidechains forming hydrogen bonds with the phosphate backbone, both unfiltered and ordered designs were significantly less preorganized than natives. “Unfiltered Designs” were filtered by all other metrics except “Max RotamerBoltzmann Probability (RKQE-base)” and “Median RotamerBoltzmann Probability (sc-base)”. “Ordered Designs” were filtered on “Max RotamerBoltzmann Probability (RKQE-base)” or “Median RotamerBoltzmann Probability (sc-base)” and comprise all designs that were tested by yeast display in this study. Violin plots show lower quartile, median, and upper quartile (dashed lines). White circles represent the mean of each distribution.

2.2.5 Protein backbone resampling

Designs with the most favorable DNA binding interactions post-superimposition, as assessed with the above metrics, were selected for experimental characterization. To obtain additional high-quality designs, the DNA-interacting segments of the filtered designs were extracted, clustered, and grafted back into the original in silico scaffold library, followed by a second round of sequence design (Fig. 2.1F)[29]. We also diversified the best designs using RoseTTAFold Inpainting[62], focused on the resampling of scaffold loops, followed by sequence design (Fig. 2.1G). We generated at least 10,000 designs for each DNA target that passed all the structural and DNA interaction filters using a combination of these approaches.

Backbone resampling with motif grafting

The binding energy and interface metrics for all the continuous secondary structure motifs were calculated for the designs generated in the broad search stage.[29] The motifs with good interactions with the target were extracted and aligned using the target structure as the reference. All the motifs were then clustered based on an energy-based TM-align-like clustering algorithm without any further superimposition[45]. The best motif from each cluster was then selected based on the per-position weighted Rosetta binding energy, using the average energy across all the aligned motifs at each position as the weight. Around 500–2,000 best motifs were selected, and the scaffold library was superimposed onto these motifs using the MotifGraft mover[63]. Interface sequences were further optimized, and computational metrics were computed for the final optimized designs as described above.

Backbone remodeling with protein inpainting

Scaffold secondary structures were determined using DSSP[64]. ProteinInpainting contigs were generated for each design that mask scaffold loops longer than 4 residues and surrounding residues, while ensuring that all residues forming hydrogen bonds to the DNA backbone were conserved. 10–20 unique contigs were generated for each design and sequences were constrained to a maximum of 65 amino acids. ProteinInpainting outputs were aligned to the DNA target using fixed interface residues of the input structure. The aligned ProteinInpainting outputs were subject to several further LigandMPNN + FastRelax rounds before AF2 monomer prediction and superposition steps.

2.2.6 Energy function optimization

Author's note: When I started working on the DNA-binder project, the concept for my thesis was something like "a two-pronged approach to understanding DBPs by designing new ones de novo *and optimizing the Rosetta energy function for improved modeling." The approach I used was a derivative of the previously-published OptE method.[65] This optimization process is fundamentally similar to training a neural network – I developed a*

set of tests for DNA and protein-DNA modeling accuracy and minimized the total scores over the course of many iterations. As I see it now, the greatest weakness of my optimization work was the very small set of reference structures used. This limitation was necessary because of how long it takes Rosetta to properly relax a large number of structures and how many iterations are needed to optimize dozens of parameters with this approach. Regardless, the final energy function I produced continues to be used for Rosetta FastRelax and ΔΔG calculations in all of our DBP design campaigns. Until we have more efficient ways to perform said computations, it may be worth revisiting the energy function for further optimization.

Multiple steps of the DBP design pipeline involved sequence design and/or modeling protocols with Rosetta. To facilitate this, a new version of the Rosetta score function was trained to better evaluate the energy of protein-DNA interfaces. Additional flexibility of the DNA duplex was incorporated into Rosetta's rotamer optimization and gradient-based minimization modules using modifications of DNA dihedral angles.[66] Then, the score function was optimized using the same general method as previously published.[65]

The weights of individual terms in the score function were optimized to reproduce the geometries of DNA crystal structures. Specifically, the distributions of pairwise atomic distances, base-stacking and base-pairing geometries, and bond torsions were considered. Additional optimization was performed on tasks related to protein-DNA complex structures. These tasks included energy ranking of perturbed crystal structures, rotamer recovery in repacking crystal structures, and sequence recovery in redesigning the protein sequence of crystal structures. An additional weight was placed on the frequency of positively charged residues at interface positions, because previous score functions tended to overestimate the strength of solvent-exposed charged interactions. Similar geometric and design tasks were included for protein structures alone. Rosetta score weights optimized included partial atomic charges of protein and DNA, hydrogen bond strengths, and solvation energies. The resulting score function showed improvement across nearly all tasks, with the greatest improvements found in the protein-DNA energy ranking and sequence design. Figure 2.6 shows the optimization trajectory of the final energy function; Table 2.2 shows the change in scores on individual tests.

Author's note: Another large change we made to the energy function, but ultimately discarded, was a depth-dependent dielectric constant in the electrostatic potential. This term, implemented by Hugh Haddox, more accurately models the different electrostatic screening effects of solvent and protein interior. Rosetta's default distance-dependent dielectric constant dramatically over-weighs surface-exposed salt bridges. As a result, Rosetta FastDesign tends to place far more charged residues on the protein surface and the protein-DNA interface than would be seen in native proteins. My optimized energy function without the depth-dependent dielectric ultimately resolved this issue by changing

the solvation potential on charged residues, and the eventual switch to LigandMPNN made this point moot.

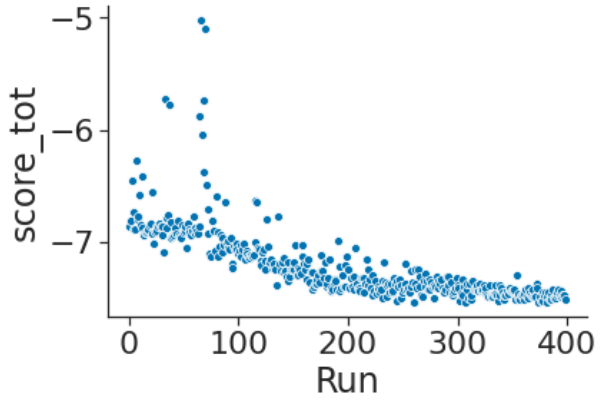


Figure 2.6 Example energy function optimization trajectory. X-axis shows number of optimization cycles, each of which consists of a single run of all the tests followed by an update to a single parameter. Y-axis shows the weighted sum of all tests' scores, which is the target of optimization.

Table 2.2 Breakdown of individual test scores in energy function optimization

Test	Initial score	Final score	Difference	% change
Decoy ranking	-0.662	-0.679	-0.017	2.6%
DNA sequence recovery (%)	-0.513	-0.507	+0.006	1.2%
DNA sequence recovery (KLD)	+0.104	+0.048	-0.056	53.8%
DNA distance distributions	+0.164	+0.157	-0.007	4.3%
DNA base-stacking distributions	+0.236	+0.224	-0.012	5.1%
DNA base-pair distributions	+0.304	+0.292	-0.012	3.9%
DNA torsion distributions	+0.049	+0.049	+/-0	0.0%
Protein distance distributions	+0.068	+0.053	-0.015	22.0%
Protein sequence recovery (%)	-0.343	-0.366	-0.023	6.7%
Protein sequence recovery (KLD)	+0.150	+0.064	-0.086	57.3%
Protein rotamer recovery	-0.673	-0.683	-0.010	1.5%
Native DBP redesign R/K frequency	+4.421	+1.632	-2.789	63.1%
Total (weighted sum)	-6.864	-7.517	-0.653	9.5%

2.3 Experiments and results

We created three sets of designs using variations of the overall design approach.

In the first set, we generated 21,488 designs using Rosetta-based sequence design, the motif grafting strategy, and our custom scaffold library of AF2-predicted native DNA-binding domains. In this set, the double-stranded DNA (dsDNA) targets were the DNA portions of co-crystal structures.

In the second design set, we generated 12,273 designs against the same DNA sequences, with the LigandMPNN sequence design strategy and the motif grafting approach for backbone resampling. In this case, rather than designing only against the dsDNA conformations found in each target's respective crystal structure, we also designed against straight B-DNA of the same sequences (6,608 designs B-form, 5,666 crystal-derived). The LigandMPNN approach was less effective at generating designs with high contact molecular surface, likely because of the ability of Rosetta to relax the protein backbone during sequence design but ultimately produced designs with more favorable free energy of binding (Rosetta ΔΔG) and an increased number of hydrogen bonds to bases (Figure 2.4 above).

Finally, in the third set we generated 100,000 designs using the LigandMPNN-based design pipeline and the inpainting-based backbone remodeling protocol against 11 unique B-DNA targets. To test if our method could generate binders to novel DNA sequences, design set 3 sequences were not derived from crystal structures and contain motifs not represented among DNA sequences bound by protein-DNA complexes in the PDB or in the JASPAR non-redundant transcription binding profile database.[67,68]

2.3.1 High-throughput screening

Separately for each set of designs, synthetic oligonucleotides (230 base pairs) encoding the 50–65-residue designed proteins were ordered in a single pool and cloned into a yeast surface-expression vector. Cells containing designs that bound each DNA target were enriched by several rounds of fluorescence-activated cell sorting (FACS) using fluorescently labeled target dsDNA oligos (details below). The naive and sorted populations for each DNA target were deep sequenced, and the frequency of each design in the starting population and after each sort was determined. From this analysis, we identified 97 designs that were substantially enriched (>100x) in DNA sequencing pools, compared to the unsorted library.

Author's note: These 97 designs were just those that were found enriched in the pool sorted against those designs' intended corresponding target; there may be other designs with specificity that were left out.

We tested these 97 designs as individual clones in a 96-well screening format and found detectable binding for 44 of them (Figure 2.7). The remainder may result from doublet transformants in the yeast pool or are very weak binders that were enriched under higher dsDNA oligo concentrations. Of the 44 successful designs, 30 were derived with targets modeled as ideal B-DNA, and 14 were derived with DNA crystal structure models as targets. For each of these designs, we knocked out the DNA binding interface by substituting the 2–3 residues making the most extensive interactions with the DNA bases. These knockout mutations completely or substantially disrupted binding for all designs

that had detectable binding on yeast (Figure 2.7), indicating that the functional designs are working as intended.

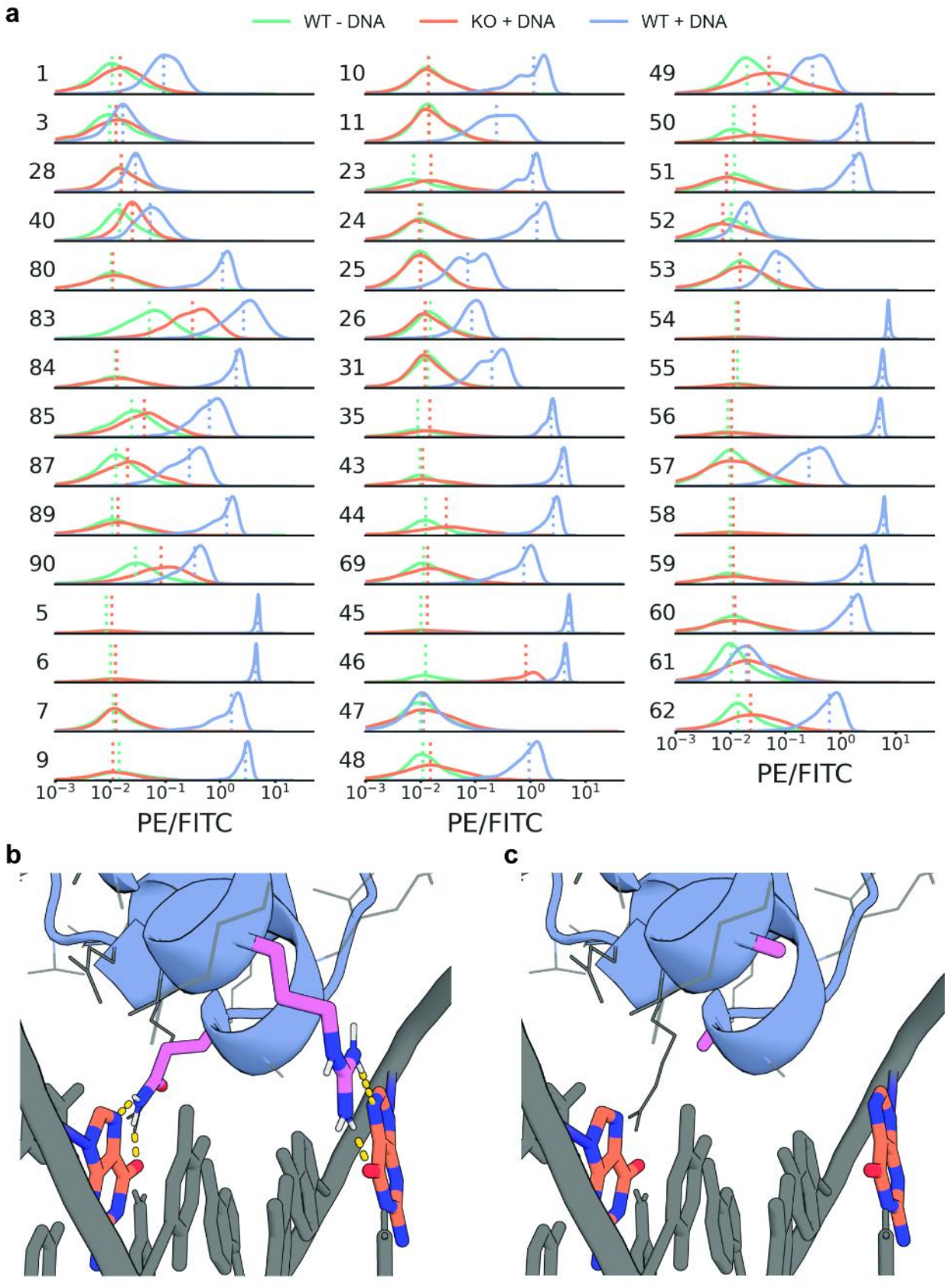
Yeast surface display

Saccharomyces cerevisiae EBY100 strain cultures were grown in C-Trp-Ura medium supplemented with 2% (w/v) glucose. For induction of expression, yeast cells were centrifuged at 6,000g for 1 min and resuspended in SGCAA medium supplemented with 0.2% (w/v) glucose at the cell density of 1 × 107 cells per ml and induced at 30 °C for 16–24 h. Cells were washed with PBS with 1% (w/v) BSA and labeled with biotinylated targets using two labeling methods: with-avidity and without-avidity labeling. For the with-avidity method, the cells were incubated with biotinylated target, together with anti-c-Myc fluorescein isothiocyanate (FITC) and streptavidin–phycoerythrin (SAPE). The concentration of SAPE in the with-avidity method was used at one-quarter of the concentration of the biotinylated targets. For the without-avidity method, the cells were first incubated with biotinylated targets, washed, and then labeled with SAPE and FITC.

Cell sorting of labeled yeast pools was performed using a Sony SH800S cell sorter. Libraries of designs were sorted using the with-avidity method for the first few rounds of screening to exclude weak binder candidates, followed by several without-avidity sorts with different concentrations of targets. For SSM libraries, two rounds of with-avidity sorts were applied and in the third round of screening the libraries were titrated with a series of decreasing concentrations of targets to enrich mutants with beneficial mutations.

For yeast display characterization of individual designs, including competition assays, DNA sequences encoding the proteins of interest were purchased as Integrated DNA Technologies (IDT) E-Blocks, transformed into yeast cells, and incubated in 96 well culture plates. Labeling with biotinylated dsDNA targets and SAPE/FITC was performed in a 96 well plate format. Of the 44 designs which were confirmed to bind their intended target in clonal yeast display experiments, (determined by knockout experiments, shown in Figure 2.8 below), we categorized the 14 with detectable binding to at most two of the thirteen tested DNA targets as specific binders and the remainder as nonspecific.

(next page) **Figure 2.7 Clonal analysis of binder designs by yeast surface display confirms dsDNA-binding function. A**, Histograms of binding activity (PE/FITC) are shown for each design. Knockout sequences were created by mutating 1–3 key interface residues for base-specific contacts present in the wildtype (WT) design model. Samples of the WT design (WT+DNA, blue), and the knockout sequence (KO+DNA, orange) with target were analyzed after labeling with each respective dsDNA oligo at 1 μM with avidity (DBPs 7, 10, 11, 24, 25, 26, 28, 31, and 40 collected without avidity). The background signal of the wildtype design without dsDNA labeling (WT-DNA) is shown in green. Interface knockouts substantially disrupted dsDNA-binding in nearly all cases. **B**, Example (DBP43) of interface knockout of the original design model with base-specific hydrogen bonding ARG and GLN residues (pink). **C**, Model showing the two ALA substitutions (pink) of those residues.


a
WT - DNA
KO + DNA
WT + DNA
1
3
28
40
80
83
84
85
87
89
90
5
6
7
9
10
11
23
24
25
26
31
35
43
44
69
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
10^{-3} 10^{-2} 10^{-1} 10^{0} 10^{1}
PE/FITC
b
c

2.3.2 Specificity testing

We tested the specificity of our designs in three ways: cross-reactivity between DNA targets, competition strengths of mutated DNA targets, and full profile determination by protein-binding microarray (PBM).[69]

Author's note: I've listed these methods in order of increasing precision, which is accompanied by increasing effort and cost. Additionally, the PBM assay was done by our collaborator Olivier Boivin in Raluca Gordân's lab at Duke. As a result, we performed the cross-reactivity experiment for all 97 DBPs and 13 DNA targets, the competition assay for just 10 DBPs, and the PBM assay on just a few.

Cross-reactivity screening

We performed an all-by-all screen of DBP design hits to 13 unique dsDNA targets (Figure 2.8). Several designs exhibited a strong preference for only their designed target sequence (e.g. DBPs 6, 9, 62), others exhibited a strong preference for 2 or 3 of the sequence targets, and a few bound to most of the targets. To try to understand these observed binding preferences, each tested DNA sequence was threaded onto each design complex model at all possible base pair alignments, the alternative complex models were relaxed with Rosetta, and the model with the most favorable Rosetta ΔΔG was selected. We found a modest correlation between the predicted free energy of binding and the extent of off-target binding (Figure 2.8, yellow dots). For DBPs 44 and 89, Rosetta ΔΔGs comparable to the original targeted sequence were obtained for most of the off-target sites, consistent with the observed low specificity. Overall, we found that 14 designs bound with specificity closely consistent with the design models, including binders for 5 unique DNA sequences (Sequences A–E).

Specific binders were obtained for some sequence targets, such as Sequence D, at much higher rates than others, suggesting a preference for specific DNA motifs. Indeed, many of the Sequence D binder design models contain very similar interface hydrogen bond contacts. This may reflect a greater suitability of HTH scaffolds for some motifs over others, the specific DNA shape formed by the preferred target motifs, or an inherent preference of the LigandMPNN model.

Author's note: there are several designs with high apparent specificity in the all-by-all analysis, but which strongly prefer a target other than the one they were designed for. While these could be considered great successes of the screening approach – they are novel DBPs with high specificity! – we generally consider them failures of our design strategy. Several promising designs were also discarded due to purification difficulties.

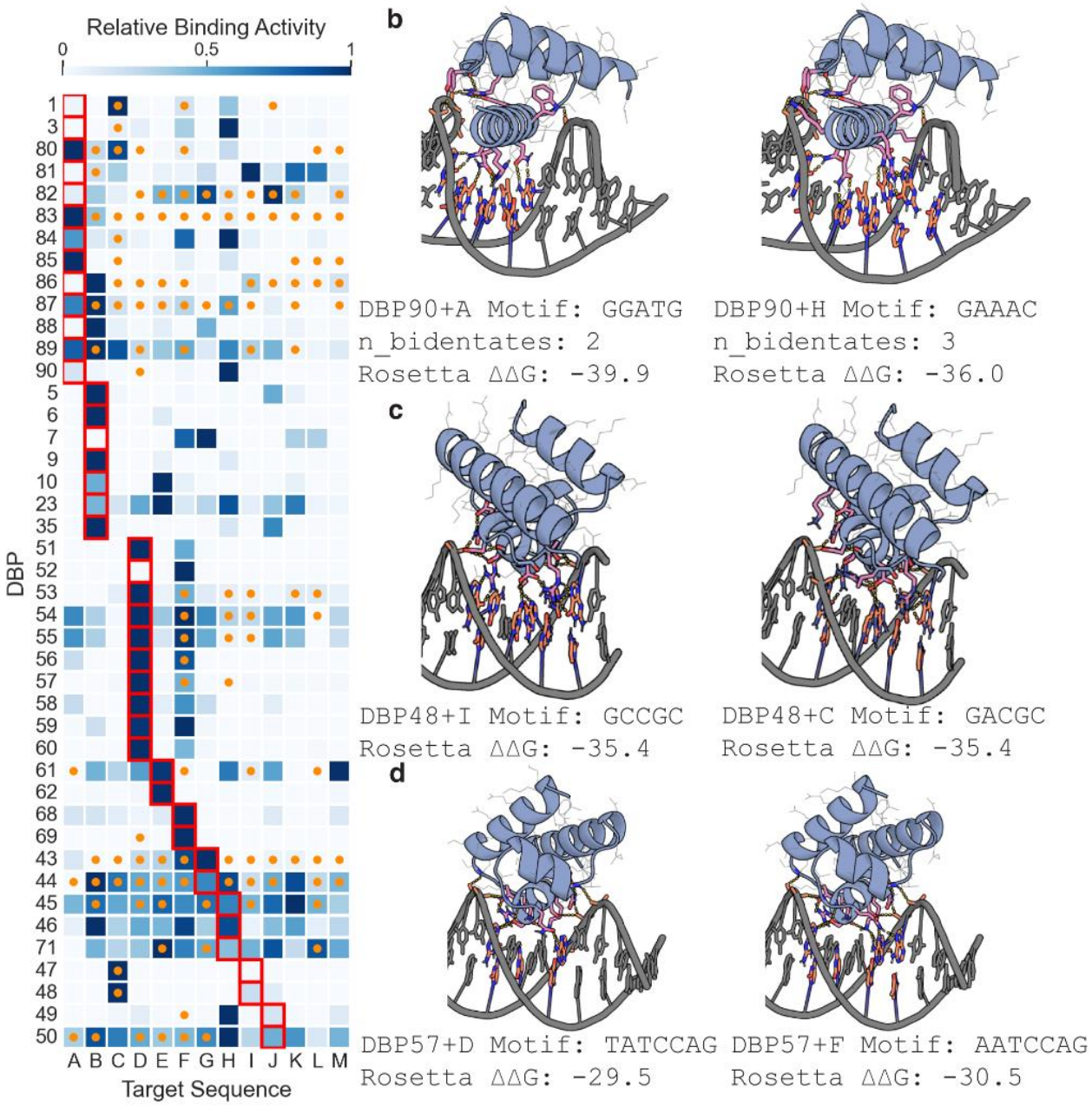


Figure 2.8 All-by-all analysis of selected designs by yeast surface display. A, Yeast surface display relative binding activity (Normalized PE/FITC) of each design labeled at 1μM dsDNA with avidity, normalized by design row. Red squares indicate the intended target sequence for each design. Orange dots indicate target sequences containing Rosetta-predicted binding motifs. Sequences were considered potential binding targets if they had Rosetta ΔΔG less than or equal to the designed complex. DBPs 83, 85, 65, 6, 9, 35, 69, 47, 48, 51, 56, 57, 60, and 62 were considered to preferentially bind less than 3 of the 13 tested DNA target sequences, including their designed target sequence. **B,** DBP90 bound weakly to its initial design target, but strongly to an alternate target sequence (H) with slightly higher Rosetta ΔΔG but also allowed for bidentate

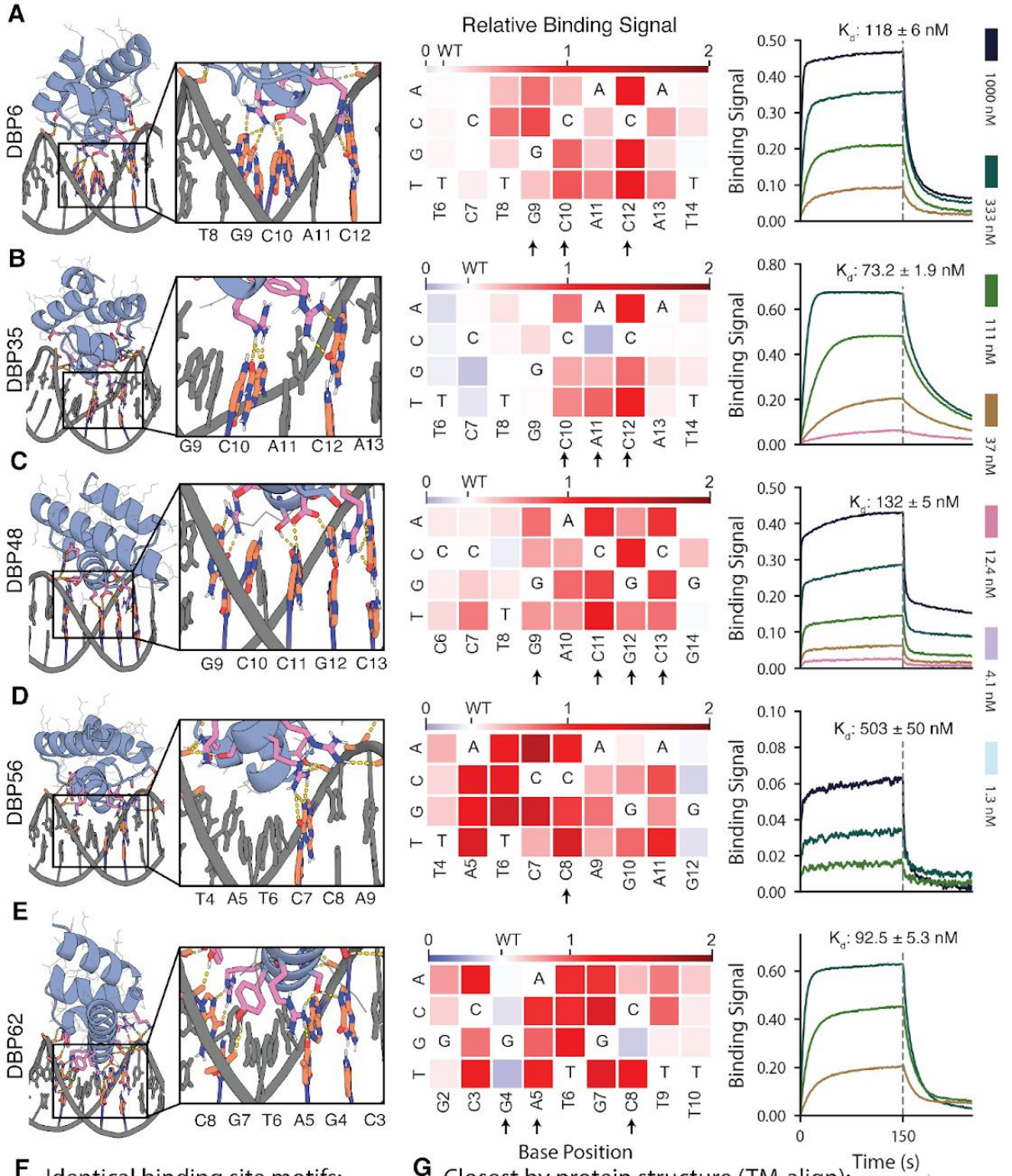
hydrogen bonds to 3 bases. Left: DBP90+A (on-target model), Right DBP90+H (alternate target model). **C,** DBP48 bound weakly to its initially designed target sequence, but strongly to Rosetta-predicted alternative target site (D) that differed by only 1 base pair across the interface and had equivalent Rosetta ΔΔG. Left: DBP48+I (on-target model), Right DBP48+D (alternate target model). **D**, DBP57 bound strongly to its initial design target as well as an alternate target that contained an identical 6 bp stretch (ATCCAG) at the binding interface. Left: DBP57+E (on-target model), Right: DBP57+C (alternate target model).

Competition assay

We used a yeast display competition assay to characterize the DNA binding site specificity of a subset of the designs (Figure 2.9, A to E, left). Addition of non-biotinylated competitor dsDNA to biotinylated target sequence reduced binding signal by flow cytometry, and scanning base substitutions through the competitor revealed positions important for binding (Figure 2.9, A to E, middle). DBPs 6, 35, 48, 56, and 62 exhibited specificities consistent with the designed sidechain-base interactions. For example, in DBP6, R31 and R36 in the design model form bidentate hydrogen bonds with the guanines of base pair positions G12 and C9, respectively, while T32 forms a hydrogen bond with C10. Substitution of the bases at positions 9, 10, and 12 eliminated competition, indicating specificity for the GCxG motif as expected (Figure 2.9 A). DBP62 exhibited specificity for its target site despite having relatively few base-specific hydrogen bonding interactions; specificity in this case may result from the very tightly packed interface (Figure 2.9 E).

Universal protein binding microarrays (uPBMs)

Universal PBM experiments were carried out following the standard PBM protocol.[69] Briefly, we first performed primer extension to obtain double-stranded DNA oligonucleotides on the microarray. Next, each microarray chamber was incubated with a 2% milk blocking solution for 1 h, followed by incubations with a PBS-based protein binding mixture for 1 h and with Alexa488-conjugated anti-His antibody (1:20 dilution, Qiagen 35310) for 1 h. The array was gently washed as previously described and then scanned using a GenePix 4400A scanner (Molecular Devices) at 5-μm resolution. Data were normalized and processed with standard analysis scripts. Results are shown in Figure 2.10.


A
DBP6
T8 G9 C10 A11 C12
Relative Binding Signal
0 WT 1 2
K$_d$: 118 ± 6 nM
Binding Signal
1000 nM
333 nM
B
DBP35
G9 C10 A11 C12 A13
K$_d$: 73.2 ± 1.9 nM
111 nM
37 nM
C
DBP48
G9 C10 C11 G12 C13
K$_d$: 132 ± 5 nM
12.4 nM
4.1 nM
D
DBP56
T4 A5 T6 C7 C8 A9
K$_d$: 503 ± 50 nM
1.3 nM
E
DBP62
C8 G7 T6 A5 G4 C3
K$_d$: 92.5 ± 5.3 nM
Base Position
Time (s)
F
Identical binding site motifs:
DBP6: TGCACA
1L3L: TGCACA
DBP48: GACGC
6EL8: GACGC
G
Closest by protein structure (TM-align):
DBP35: TGCACA
3JXC: TTAAAT
DBP56: ATCCAGA
4IWR: GTGACTT
DBP62: CGATGCT
2H27: GGAACTT

(previous page) **Figure 2.9 Designed DBPs bind with high affinity and specificity to the intended sites. (A to E)** Characterization of DBPs 6, 35, 48, 56, and 62, respectively. **Left**, Computational design models of characterized designs at the DNA-binding interface. DNA bases and protein residues involved in hydrogen bonding interactions are shown in orange and pink, respectively. Hydrogen bonds are highlighted with dashed yellow lines. **Middle**, Relative binding activity (PE/FITC normalized to the no-competitor condition) from flow cytometry analysis in yeast display competition assays with all possible DNA base mutations at each position of the competitor oligo. Blue indicates competitor mutations where competition was stronger than with the wild-type competitor, while red indicates competitor mutations where competition was weaker. Arrows indicate base pair positions contacted with hydrogen bonds or hydrophobic contacts to base atoms in the design model. **Right**, Binding of purified miniprotein designs to the DNA target with BLI. Each line represents biotinylated dsDNA target dilutions by ⅓. The highest DNA target concentration is indicated in each plot.

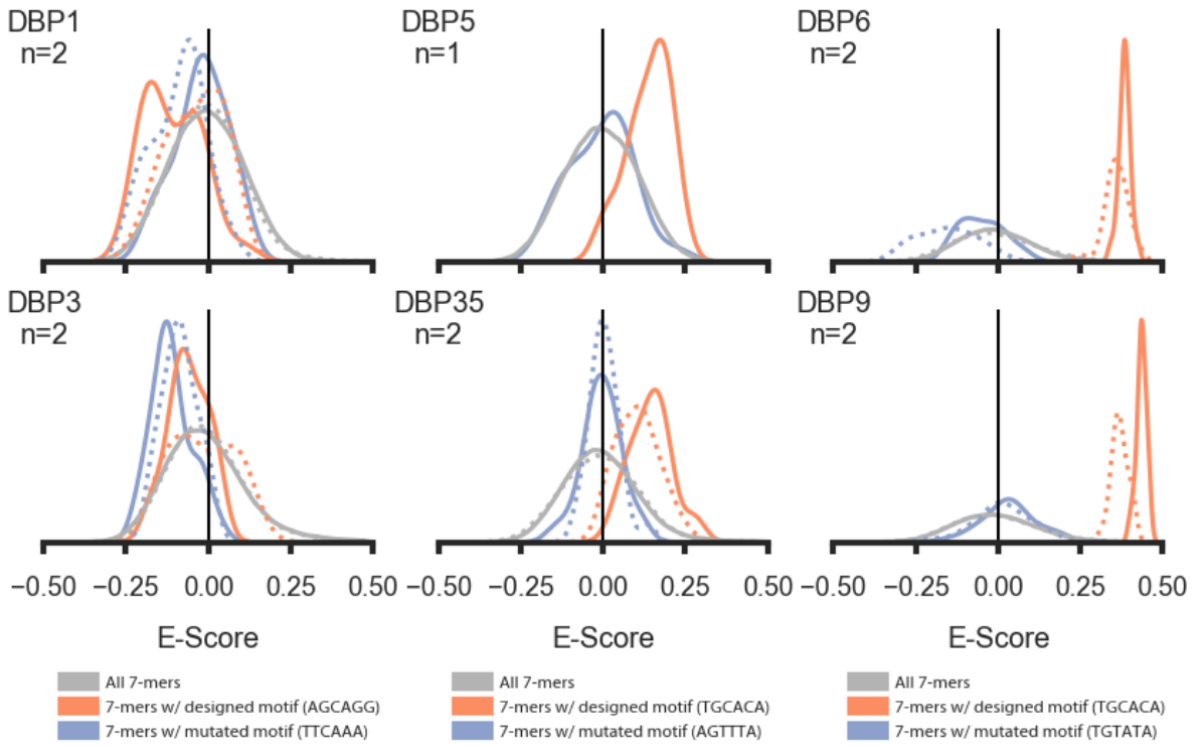


Figure 2.10 Analysis of DBPs with universal PBM experiments containing all 7-mers Solid lines represent replicate 1 while dashed lines represent replicate 2, where applicable. DBPs 6, 9, and 48 were highly specific to the intended target and the mean percentile rank of 7-mers containing the designed binding site 5-mer or 6-mer was 99.54%, 99.89%, and 97.59%, respectively. DBPs 5 and 35 were less specific to their target site but still preferred the target motif over sequences with a mutated binding motif (designed motif percentile 86.54% and 81.88%, respectively). DBPs 1 and 3 did not appear to have a preference to the designed target site (33.19% and 46.58%, respectively). *Author's note: the cross-reactivity analysis also showed that DBPs 1 and 3 preferred off-target sequences. We choose not to show binding as sequence logos, both because this view is more reflective of the relevant data and because the sequence logos do not appear to match our intended targets.*

2.3.3 Purification and crystallography

Author's note: While most of the lab work described above was done by my colleagues Cameron and Robert (see acknowledgements and introduction to chapter 2), I was solely responsible for the protein purification described here. Mostly, this was because I had prior experience with protein purification methods and optimizations. I had not, however, previously purified DNA-binding proteins. I learned the hard way how difficult it can be to separate E. coli DNA from positively-charged proteins, especially ones explicitly *designed to bind DNA with high affinity. Ultimately, the solution was to incubate with DNAse following the manufacturer's instructions, followed by running IMAC and SEC in very high salt buffers (1000-2000 mM). Fellow graduate students following my work have further optimized the protocol, but here is the one I used.*

DNA sequences encoding the proteins of interest were purchased as Integrated DNA Technologies (IDT) E-Blocks and incorporated into plasmids using Golden Gate assembly. The plasmids were then transformed into BL21(DE3) competent E. coli.

The transformation reactions were used to inoculate starter cultures in 5 mL or 25 mL of "Terrific Broth" (TB), supplemented with 1% (w/v) glucose and 50 mg/L kanamycin. After shaking overnight at 37°C, the starter cultures were diluted 50-fold into 50 mL or 500 mL of TB with kanamycin. These cultures were incubated at 37°C, shaking, until the optical density (OD) reached 0.6-0.8, at which point protein expression was induced by the addition of IPTG. The cultures were then further incubated overnight at 18°C.

Cells were harvested by centrifugation for 15 min at 3000g, pellets resuspended in lysis buffer (150 mM NaCl, 20 mM Tris-HCl, 0.5 mg/mL DNAse I, 1 mM PMSF, pH 8.0), the cells lysed by sonication, and the lysate clarified by further centrifugation for 30 min at 20,000g.

The supernatant was passed through Ni-NTA resin in a gravity column, and then the resin was washed with 20 column volumes of high-salt wash buffer (2 M NaCl, 20 mM Tris-HCl, 20 mM Imidazole, pH 8.0). Either (A) the His-tagged protein was eluted with 2 column volumes of elution buffer (1 M NaCl, 20 mM Tris, 250 mM Imidazole, pH 8.0), or (B) the resin was further washed with 5 column volumes of SNAC buffer (100 mM CHES, 100 mM Acetone oxime, 100 mM NaCl, 500 mM GnCl, pH 8.6), incubated in 5 column volumes of SNAC buffer + 0.2 mM NiCl2 on an orbital shaker at room temperature overnight, and collected as the column flow-through.

Author's note: a member of my committee suggests that you should use twin streptavidin tags rather than His tags for more specific and robust purifications, such as for crystallography.

Whether cleaved or not, the protein was concentrated to about 1 mL and loaded in 500 µL samples onto a Cytiva Superdex™ 75 Increase 10/300 GL gel filtration column

equilibrated in buffer (1 M NaCl, 20 mM Tris-HCl, pH 8.0). Fractions containing monomeric protein were pooled and concentrated to about 200 µL. Protein concentrations were estimated by absorbance at 280 nm. For proteins with no Trp, Tyr, or Cys residues, concentrations were approximated by Bradford reagent absorbance at 470 nm.

Most of the proteins expressed, were soluble, and appeared monodisperse on SEC, as shown in Figure 2.11. For proteins that expressed solubly, purification yields ranged from 0.1 – 1 mg of protein. Binding to the biotinylated dsDNA oligo was assessed 241 using biolayer interferometry, and all designs were found to bind with binding affinities ranging 242 from 30–500 nM (Figure 2.9, A to E, right).

Author's note: In Figure 2.11, I show specifically absorbance at 230 nm rather than the usual 280 nm , because some of these proteins lack aromatic residues and also because all of these protein samples were contaminated with detectable amounts of DNA (based on the 260:280 absorbance ratio). The y-axis is also normalized because these proteins have wildly different yields. Prior to normalization, the void peak (8 – 10 mL elution volume) is about the same height in all of these traces.

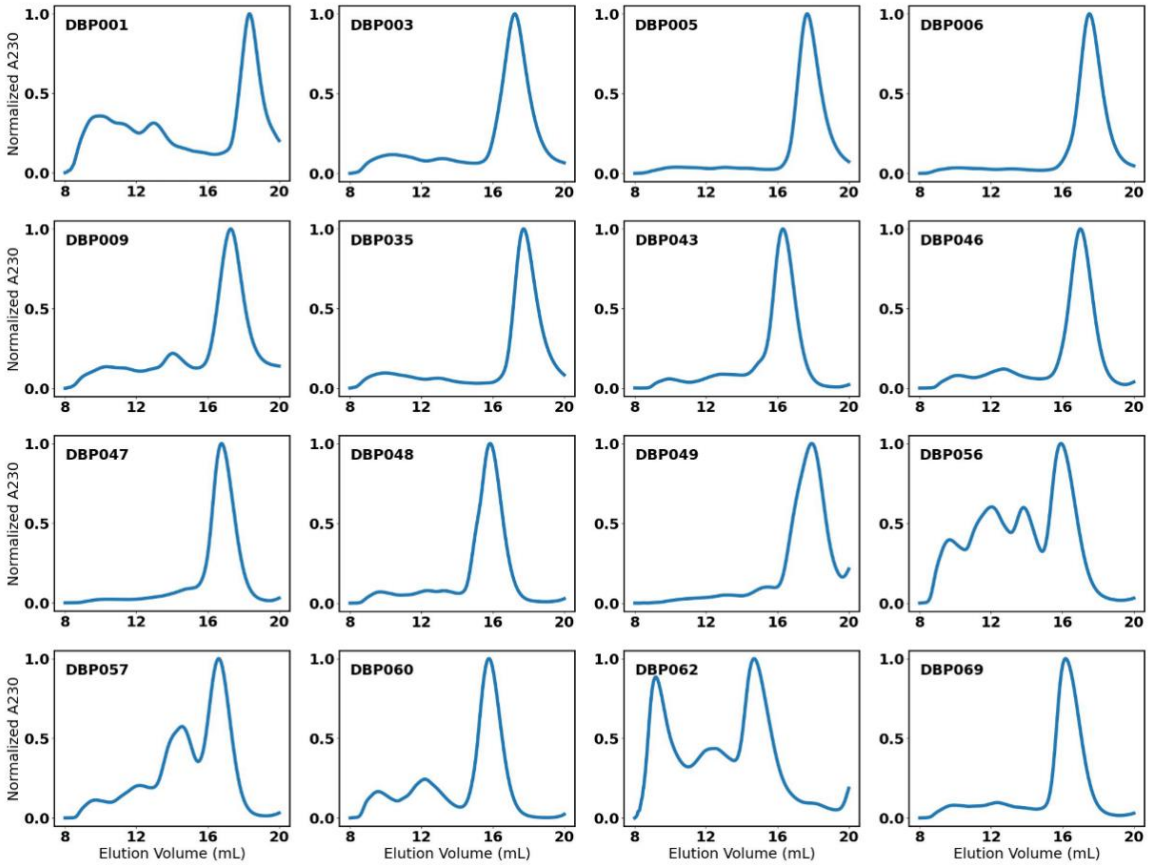


Figure 2.11 SEC traces of purified proteins. Normalized absorbance at 230 nm of elution over a Superdex™ 75 Increase 10/300 GL column. Each plot shows a separate protein sample, following IMAC purification, with the HIS-tag attached. In every case, the highest peak, corresponding to the protein of interest, was collected and used for *in vitro* experiments.

Crystallography

We solved the co-crystal structure of DBP48 in complex with its preferred target sequence and found very close agreement to the design model (Figure 2.12 A). Cα-RMSDs of the co-crystal structure to the design model were 0.64 Å for the binder alone, and 1.907 Å across all atoms of the protein-DNA complex.

Among residues forming key interactions with bases, R38 and S39 were in the closest agreement and formed the expected sidechain-base hydrogen bonds (Figure 2.12 B). D43 and R49 did not form the expected hydrogen bonds observed in the design model, likely due to slight differences in orientation of the binder to DNA and deviations from ideal B-DNA in the co-crystal structure. D43 was instead involved in a water-mediated hydrogen bond to C11 (Figure 2.12 C), and R49 was part of a hydrogen bond network involving the phosphate backbone. An additional water-mediated hydrogen bond was observed between S42 and A10. While water-mediated interactions are not considered by the Rosetta protocol used to build the side chains in the final design model, the LigandMPNN sequence design method may implicitly consider these: the PDB training set contains many examples of water-mediated hydrogen bonds, which are known to confer additional specificity in native DBPs.[13]

Extensive hydrogen bond networks were also observed with the DNA phosphate backbone, with most involved protein residues supported by sc-sc hydrogen bonds and packing interactions. These hydrogen bond networks with the phosphate backbone imply that much of the docking orientation is dominated by these interactions, suggesting that further enrichment for these features could improve design success rates.

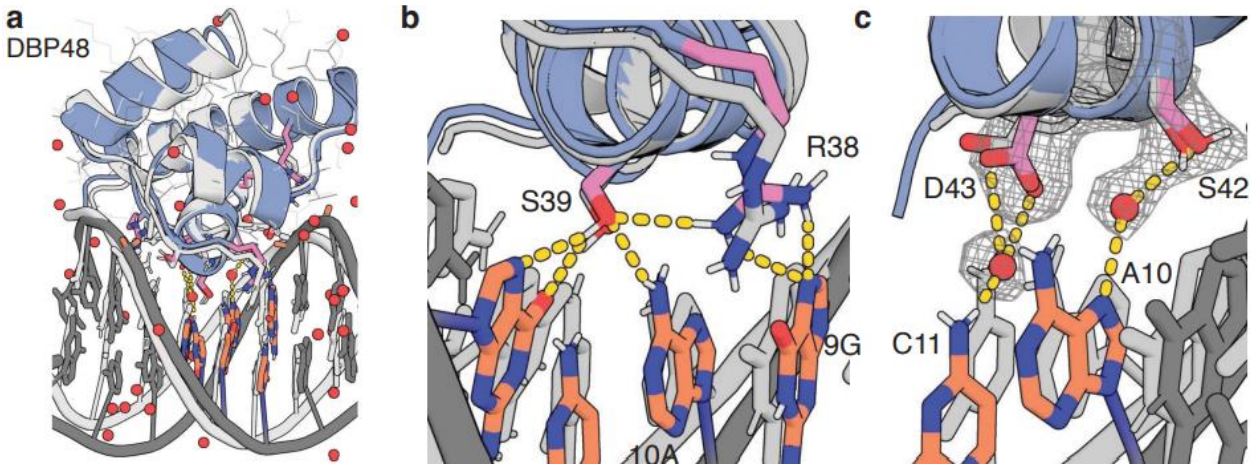


Figure 2.12 Structural validation of DNA binder designs. (A) Co-crystal structure of DBP48 (colored) and the design model (gray) are in close agreement. **(B)** Zoom544 in showing the close agreement of critical interface residues R38 and S39 between the crystal structure 545 and design model. **(C)** Close-up of water-mediated hydrogen bonds formed by S42 and D43.

Author's note: I would like to highlight Lindsey Doyle, a researcher in Barry Stoddard's lab, for her excellent work in crystallography. She was very gracious while I was struggling to optimize the purification protocol, and had to try many variations of DBPs and DNAs to get this structure. Below is her crystallization and structure determination protocol.

Purified DBP48 was complexed with duplex DNAs, of varying duplex length and a single 5'overhang base, to a final concentration of 176 μM DBP48 and 233 μM duplex DNA. Complexes were screened for crystals in several broad matrix screens using a mosquito robot (SPT LabTech), then possible hits were optimized in 24-well hanging drop trays with a 2 micro-liter drop containing a 1:1 ratio of complex to well solution and equilibrated over 1 mL of well solution.

A single diffraction quality crystal was obtained with duplex DNA of length 10 basepairs with a single base overhang at either end of the duplex (5'-ACCTGACGCGA-3', 3'-GGACTGCGCTT-5') and a well condition containing 200 mM ammonium acetate, 100 mM sodium acetate at pH 4.6, and 28% Polyethylene glycol 4000. The crystal was washed in well solution then flash cooled directly by plunging into liquid nitrogen.

Data were collected at the Advanced Light Source in Berkeley, CA on beam line 5.0.1 at a wavelength of 0.9762 Å and processed with DIALS.[70] Phases were determined via molecular replacement by searches with the original computational protein design and duplex DNA using Phaser[71] in the Phenix suite.[72] The top scoring molecular replacement solutions were run through a round of refinement with Phenix refine and further rounds of refinement with Phenix refine and rebuilding with Coot[73] were performed on the top scoring structure.

2.3.4 Transcriptional activation and repression

Author's note: I want to acknowledge my colleagues who conducted the activation and repression experiments, especially Baker lab undergraduate Beau Lonnquist and postdoctoral scholar Wei Chen. While I personally was not involved in these experiments, I'm including the results here as they are the most direct evidence that the DBPs I designed can be used practically in living cells.

Bacterial repression

We tested the ability of our designed DBPs to function in cells to regulate transcription. To assay transcriptional repression in *E. coli*, we constructed candidate NOT gates[74], where the input is a designed DBP under control of the IPTG-inducible PTac promoter and the output is yellow fluorescent protein (YFP) expression driven by a promoter incorporating the DBP DNA binding site.

Single DBP domains and two copies of the same DBPs tethered through a flexible linker failed to exhibit YFP repression upon IPTG induction, suggesting a need for higher affinity

binding, longer sequence recognition, and/or a bulkier binding protein for effective hindrance of transcription initiation by *E. coli* RNA polymerase.

To increase avidity and bulk, we positioned two copies of the same DBP (or one copy each of two different DBPs) on B-form DNA containing two palindromic copies of the target site (or the two different target sites), separated by different numbers of bases. We then used RF*diffusion*[75] to build out new protein backbone segments that either transition into the TetR homodimer[76] or interact directly in homo- or heterodimeric arrangements (Figure 2.13 A). Following sequence design with ProteinMPNN to favor folding and assembly of the extensions to the intended dimeric structure, we used AF2 (or ESMFold for TetR fusions) to predict the structures of the homo- and heterodimers and selected those that were close to the design models.

We experimentally characterized the ability of these designs to repress transcription from synthetic promoters incorporating two dimer binding sites (4 individual domain binding sites in total) flanking the -35 promoter region. Dose dependent repression (> 2 fold) was observed for 2 of 96 TetR-incorporating homodimeric designs and 18 of 192 entirely de novo homodimeric and heterodimeric repressors incorporating different designed DBPs (Figure 2.13 A).

All-by-all characterization of 6 selected designs and the corresponding cognate promoters showed considerable orthogonality (Figure 2.13, B and C), with up to 20-fold repression for cells with the cognate target. Notably, two *de novo* dimeric designs with DBP57 designed to bind palindromic arrangements of the cognate target site at different spacings and in different orientations were each specific for their intended target, indicating that a single domain can serve as the basis for creating an array of orthogonal repressors.

Repression assay details

The pRF-TetR vector[74] was used for transcriptional repression assays in E. coli. A new version of this vector (pRF-BsmB1) was constructed by first removing the LuxR gene and then replacing the TetR gene, its terminator sequence, and regulated promoter with two BsmB1 cut sites such that new repressor variants and their associated promoters could be easily inserted via Golden Gate assembly.[77]

For DBPs tethered with a flexible linker, a flexible linker was used to connect the C- and N- termini of two copies of the DBP (linker1: KESGSVSSEQLAQFRSLD, linker2: EGKSSGSGSESKST, linker3: GGGGGGGG, linker4: GSGSGSGSGSGSGSGS).

Synthetic promoters were designed by inserting DNA binding sites around the consensus -10 and -35 elements of the E. coli RNAP promoter.

Genes encoding the single domain DBP, flexibly linked, TetR fusions, homodimers, and heterodimers were ordered as Twist synthetic gene fragments encoding the repressor

gene (using Twist codon optimization), a transcriptional terminator, and an associated synthetic promoter. Heterodimer constructs were encoded into bicistronic operons. Gene fragments were ordered containing BsmB1 cut sites on either end to allow for assembly into the modified pRF-BsmB1 vector.

Upon Golden Gate assembly with the BsmB1 Type II-S restriction enzyme, plasmids were transformed into NEB 5-alpha competent *E. coli* cells and streaked onto Luria-Burtani (LB) plates containing carbenicillin. All-by-all repressor constructs were cloned by digestion with BsiWI-HF (NEB) and BbsI (NEB), gel extraction of the backbone and promoter bands, followed by ligation with T4 DNA ligase and transformation into NEB 5-alpha competent *E. coli*.

Individual transformants were picked and verified via sanger sequencing. Sequence verified colonies were inoculated into 200 µL LB media containing carbenicillin for overnight growth in 96-well round bottom plates at 37℃ in a plate shaker. The following day, 2 µL of overnight cultures were transferred into a new plate containing 200 µL LB media containing carbenicillin and 1 mM IPTG and grown for ~18 hours in 96-well round bottom plates at 37℃.

Flow cytometry analysis of cultures was performed with an Attune NxT flow cytometer with autosampler. Flow cytometry data analysis was performed using custom python code and the CytoFlow python package. For each individual sample, gating was performed using the single component CytoFlow Gaussian Mixture Model and median BL1-A channel fluorescence was determined for all gated expression events of each sample. The median BL1-A channel fluorescence value of empty cells without a pRF vector was subtracted from the median BL1-A value of each sample. For each repressor variant, fold repression was calculated from at least 7 biological replicates as the ratio of median BL1-A channel fluorescence of the uninduced sample to the median BL1-A channel fluorescence of the induced sample.

Activation in mammalian cells

A set of synthetic transcription factors (synTFs) were created by fusing the GCN4 dimerization domain and the VP64 activation domain to the C-termini of DBPs 9, 35opt, 48, 57, and 60, which collectively recognize 3 unique motifs. The dimerization domain allows the DBPs to recognize a palindromic target sequence consisting of two binding motifs, increasing the binding affinity to the DNA sequence. We used the ENGRAM[78] recording technology to measure the activity of specific cis-regulatory elements (CREs) in HEK293 cells (Figure 2.13 D). In ENGRAM, each CRE drives expression of a uniquely barcoded pegRNA, which, upon expression, is recorded into the DNA TAPE at the HEK3 locus by prime-editor PEMax. After analyzing the barcode abundance for each individual CRE, we observed 3- to 5-fold activation for DBPs 9, 35opt, 57, and 60 (Figure 2.13 E).

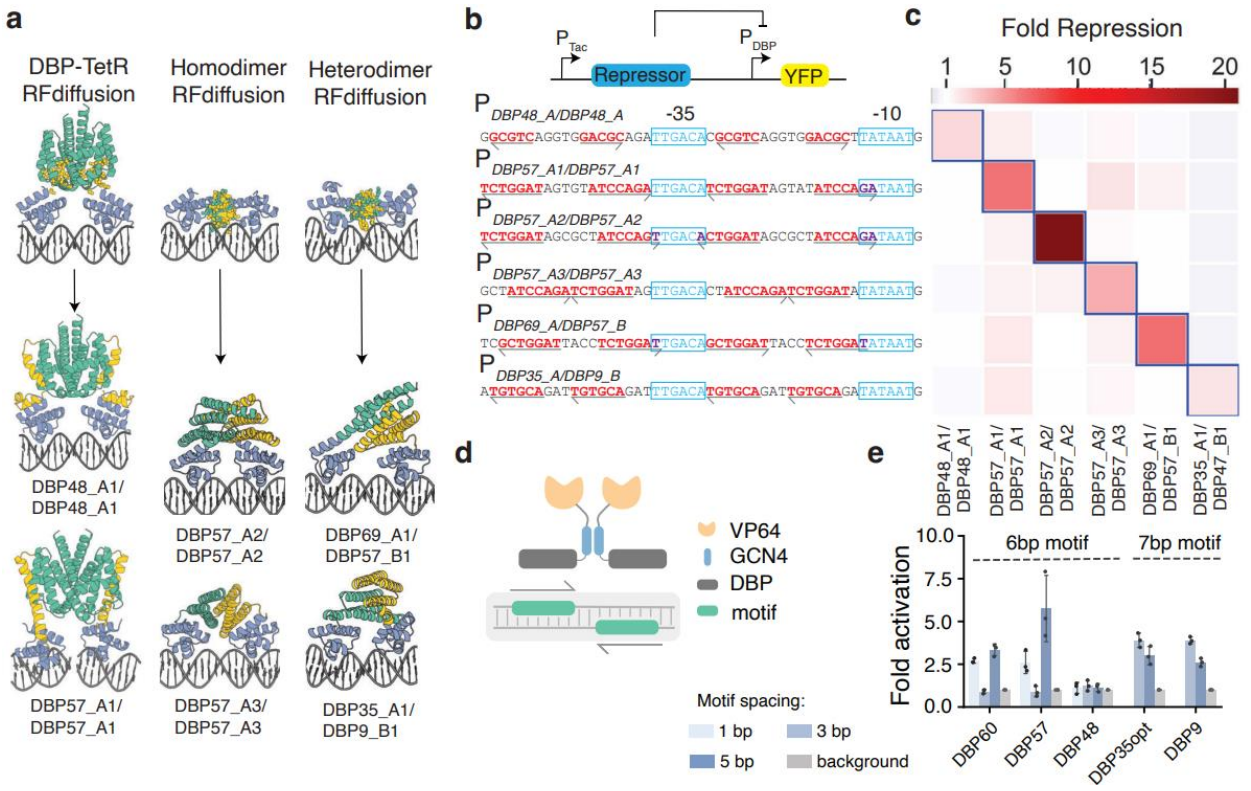


Figure 2.13 Designed DBPs function in living cells to direct transcriptional repression and activation. (A) Illustration of the RFdiffusion method for building out DBP domains into homo- or heterodimer arrangements, along with repressor designs selected for all-by-all repression assays. DBP48_A1/DBP48_A1 and DBP57_A1/DBP57_A1 are homodimer constructs transitioned into the TetR backbone; the remainder are de novo homo- or heterodimer constructs. **(B)** Transcriptional repression in *E. coli*. Functional IPTG-inducible repressor block transcription of YFP from a synthetic promoter containing the designed DBP binding sites (red text) around the -10 and -35 elements (blue text). Arrows indicate directionality of the binding site. **(C)** All-by-all orthogonality matrix showing fold repression of YFP Fluorescence from flow cytometry analysis of cells containing the successful NOT gate circuits. Blue outlines indicate on-target repressor-promoter pairs. **(D)** Transcriptional activation in HEK293T cells measured by ENGRAM. synTFs were created by fusing the GCN4 dimerization domain and the VP64 activation domain to the C-termini of the DBPs. The synTF-specific cis-regulatory elements (CRE) were created by evenly distributing palindromic binding motifs on a 130 bp transcriptionally inactive DNA sequence where each CRE drives a uniquely barcoded pegRNA for recording into DNA TAPE. **(E)** Fold activation of synTFs measured as normalized barcode abundance. Dots represent individual data points, bars represent mean fold activation, and error bars represent standard deviation of the mean relative barcode abundance (n=3 biological replicates).

2.4 Discussion of chapter 2

2.4.1 Novelty of designed DBPs

Comparison to native DBPs

Although some of the designs target DNA sequences found in crystal structures, the designed DBPs and their sequence preferences are novel.

We assessed this first by comparison of the binding site motifs to co-complex structures of native DBPs in the PDB containing a protein helix in contact with bases in the DNA major groove. We found that some designs (DBPs 6, 35, 48) preferred a similar motif as native DBP structures but had substantially unique interfaces and docking orientations, while other designs (DBPs 56, 62) bound novel sequences found neither in the PDB (Figure 2.9F above and Figure 2.14 A-C) nor in the JASPAR non-redundant transcription binding profile database (Figure 2.15).[67]

Our binder design method aims to effectively sample diverse scaffold-DNA docks to find solutions optimal for binding the target DNA sequence. The method could, in principle, recover solutions similar to known native DBP-DNA complexes. To investigate this, we compared the structures of our designed DBPs to native DBP domains in DNA co-crystal structures in the PDB by TM-align[45] (Figure 2.9G above and Figure 2.14 D-H). We found that the overall folds of the designed scaffolds had matches in the PDB, but the placement of the scaffold relative to the DNA generally differed, as expected given the *de novo* docking step in our approach. None of the closest matches by protein structure had more than 3 out of 7 common bases at the aligned DNA binding site positions, and the sidechain-base hydrogen bond networks differed substantially.

For all designs that bound their DNA targets, we also performed Blastp searches of the non-redundant protein sequences database[79] and found that most had sequence similarity to native metagenome protein sequences ranging from 40–60%.

Overall, these analyses suggest that our approach was able to utilize and expand upon the known native docking space, while exploring new sequence space, to identify effective DBP designs against the specified target sequences.

Author's note: as my committee member pointed out to me, we have not tested any negative control of comparable size to prove this statement true. For instance, designing 100,000+ protein sequences biased to form HTH folds with no specific target in mind, then screening those against the same DNA targets, could also potentially find novel sequence-specific DBPs. The next sections cover the work we did with less-costly negative controls, such as native redesign without the docking step, but we did not test those designs experimentally.

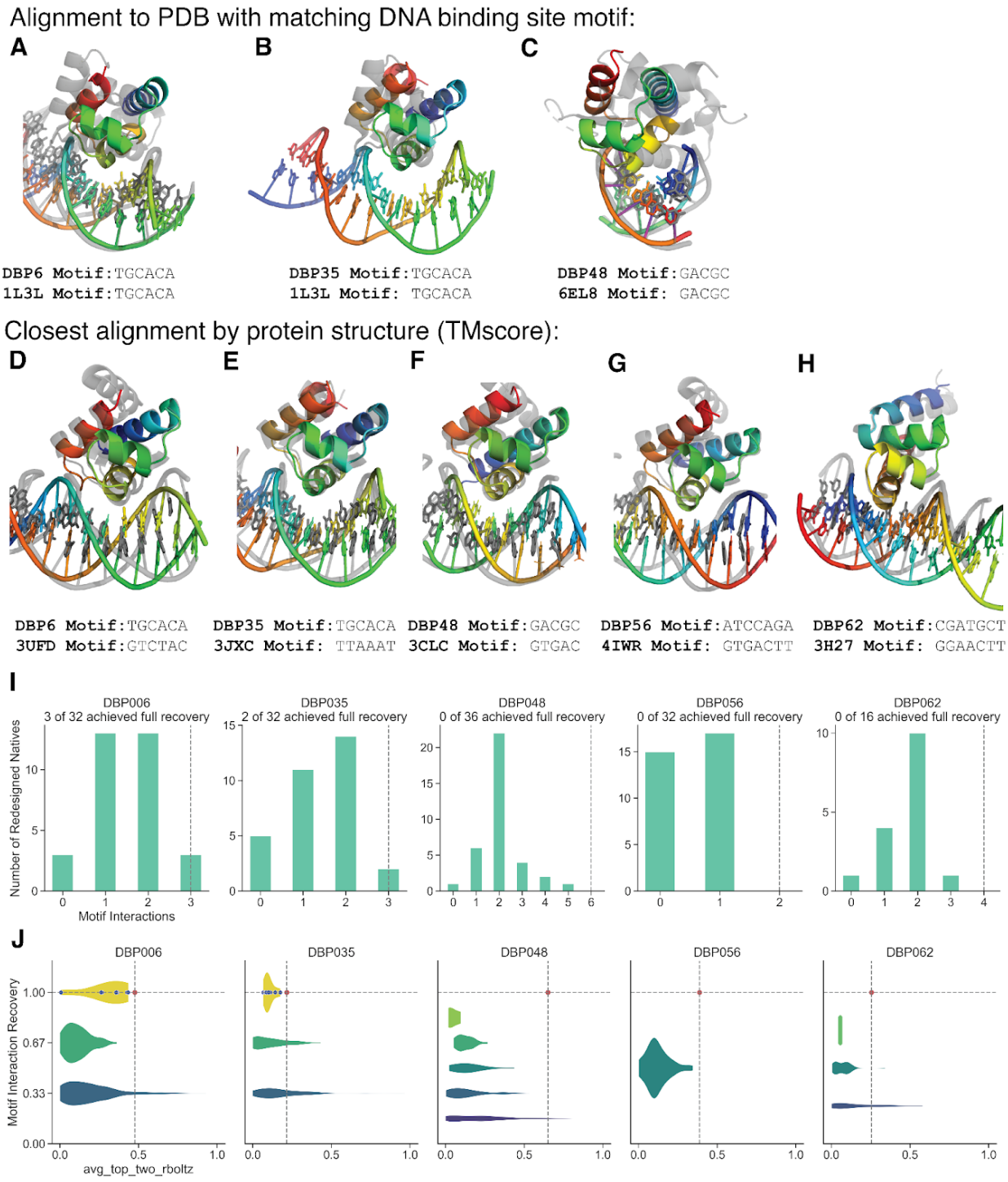


Figure 2.14 Comparison of designed DBPs with nearest native structure by target motif or protein structure. A–C, Alignment of DBP designs to PDB structures containing an identical DNA binding site motif. Native structures are shown in gray aligned to the DBP design (colored). DNA sequence matches were found by creating a set of all contiguous DNA binding site motifs in the PDB where any atom of a protein residue was within 5 Å of an atom in the contiguous DNA sequence motif. **D-H**, Structural alignment of DBP designs to nearest PDB structures by TM-align.

TM-align searches were performed on protein-DNA co-complex structures in the PDB to identify the nearest native protein scaffold. Nearest structures are shown in gray aligned to the DBP design (colored). **I**, Computed statistics on native DBPs in the PDB, redesigned in the presence of the designed DBP's DNA target motif. We examined whether the same amino acids formed hydrogen bonds with the same DNA base atoms. **J,** Analysis of sidechain preorganization for recovered motifs residues by average top two RotamerBoltzmann score. Violin plots show the distribution of *avg_top_two_rboltz* among recovered interacting residues for each design. Individual data points are shown for designs with full motif atom recovery (original design in red, best native redesigns in blue).

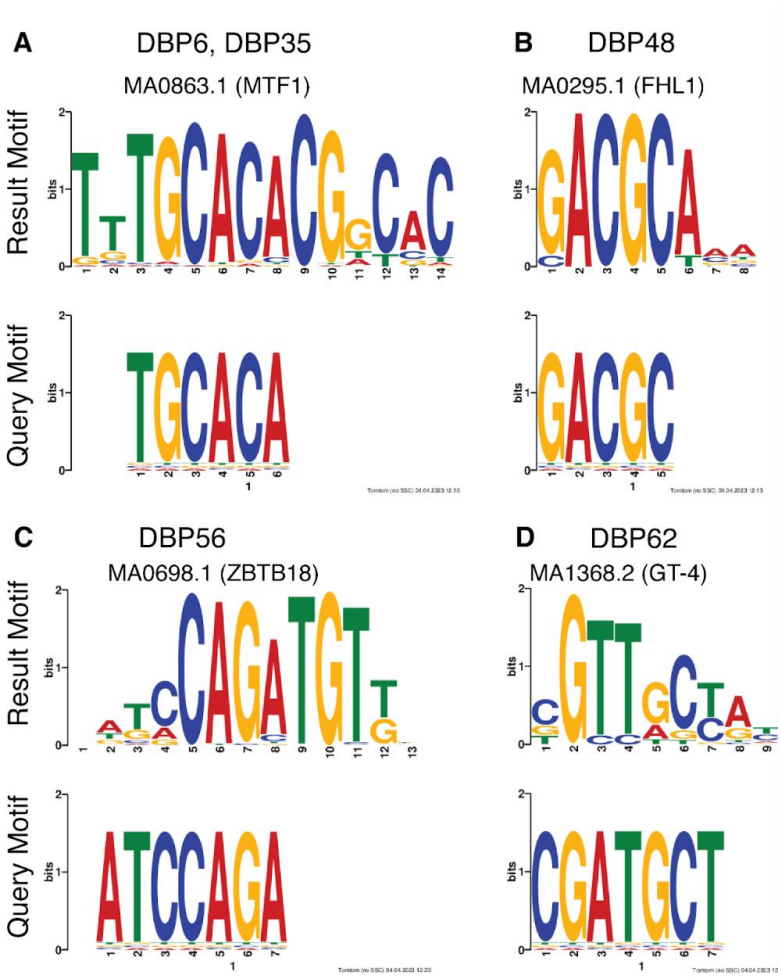


Figure 2.15 Comparison of designed DBP target motifs to JASPAR database. A-D, Comparison of the DNA binding site motifs found through motif searches of the JASPAR non-redundant transcription binding profile database. DBPs 6, 35, and 48 were found to bind highly similar sequences as the identified search hits in the JASPAR database; however, DBPs 56 and 62 were found to bind unique sequences compared to transcription factors with known specificity profiles. DBPs 6 and 35 were designed against the DNA sequence in the PDB structure 1L3L and thus were not expected to specifically bind a novel sequence. DBP48 was designed against a novel 14 bp sequence, but the experimentally verified 5 bp binding site motif was similar to the observed specificity of the FHL1 transcription factor.

Comparison to a native redesign approach

To evaluate the importance of backbone sampling through docking, we examined the ability of LigandMPNN-based sequence design to generate interfaces passing our *in silico* metrics when starting from crystal structures of native co-complexes rather than *de novo* docks. Starting from co-crystal structures with high TM-align scores to the designed DBPs, we mutated the DNA sequence *in silico* to the target sequence and redesigned the sequence using LigandMPNN. We found that designs based on fixed native backbones failed to recover most of the base-specific hydrogen bonds present in the designs produced by our docking pipeline (Figure 2.14 I).

In the few cases where native redesign did recover multiple base-specific hydrogen bonds, such as DBPs 6 and 35, the de novo-docked design models scored better on sidechain preorganization by the RotamerBoltzmann metric (Figure 2.14 J), suggesting non-hydrogen-bond features of the interface that may be critical for specific binding and require precise docking configurations.

Overall, our design method identifies designs that would not be identified through structure-based redesign and generates specific binders for unique DNA sequences that are not known to be recognized by native proteins.

2.4.2 Determinants of DBP design success

Across all targets, designs that bound specifically to their intended target (Figure 2.7 above) tended to have more sidechain- and mainchain-phosphate hydrogen bonds, lower Rosetta ΔΔG, and lower Cα RMSD of the AF2-predicted structure to the design model (Figure 2.16), while nonspecific binding was strongly correlated with a positive net charge. We did not observe enrichment of higher RotamerBoltzmann probabilities for base-hydrogen bonding sidechains, likely due to prior enrichment in the ordered design sets. However, we did observe enrichment of higher RotamerBoltzmann probabilities for phosphate hydrogen-bonding sidechains (Figure 2.16). Further enrichment for these metrics could increase design success rates.

A key feature of our design method is sampling from numerous diverse starting structures and docking positions to find complexes that can engage both the bases for sequence-specific recognition and the phosphate backbone to favor the designed binding mode. Like the most specific of our designs, native DNA binding proteins also have geometries enabling formation of mainchain-phosphate hydrogen bonds (Figure 2.17) and highly preorganized sidechain-phosphate hydrogen bonds (Figure 2.5 above). This is perhaps due to the inherent rigidity of these interactions which favor specific docks and restrict otherwise possible interactions of flexible sidechains with off-target DNA base atoms.

To explore the importance of phosphate contacts mediating specific docks for achieving specificity to a given target site, we performed LigandMPNN redesign of 14 hits from our

design campaigns against 100 randomly-generated target sequences. Upon Rosetta relaxation of the redesigned complexes in the presence of DNA, we observed that only 2 of the 100 sequences have as favorable Rosetta ΔΔGs and as many hydrogen bonds to bases, suggesting that the details of the scaffold backbone and dock make important indirect contributions to specificity by locking in the exact binding mode and narrowing the range of possible sidechain-base contacts. This makes it difficult to design DBPs to new DNA sequences through a native redesign approach and highlights the advantage of our computational sampling-based approach.

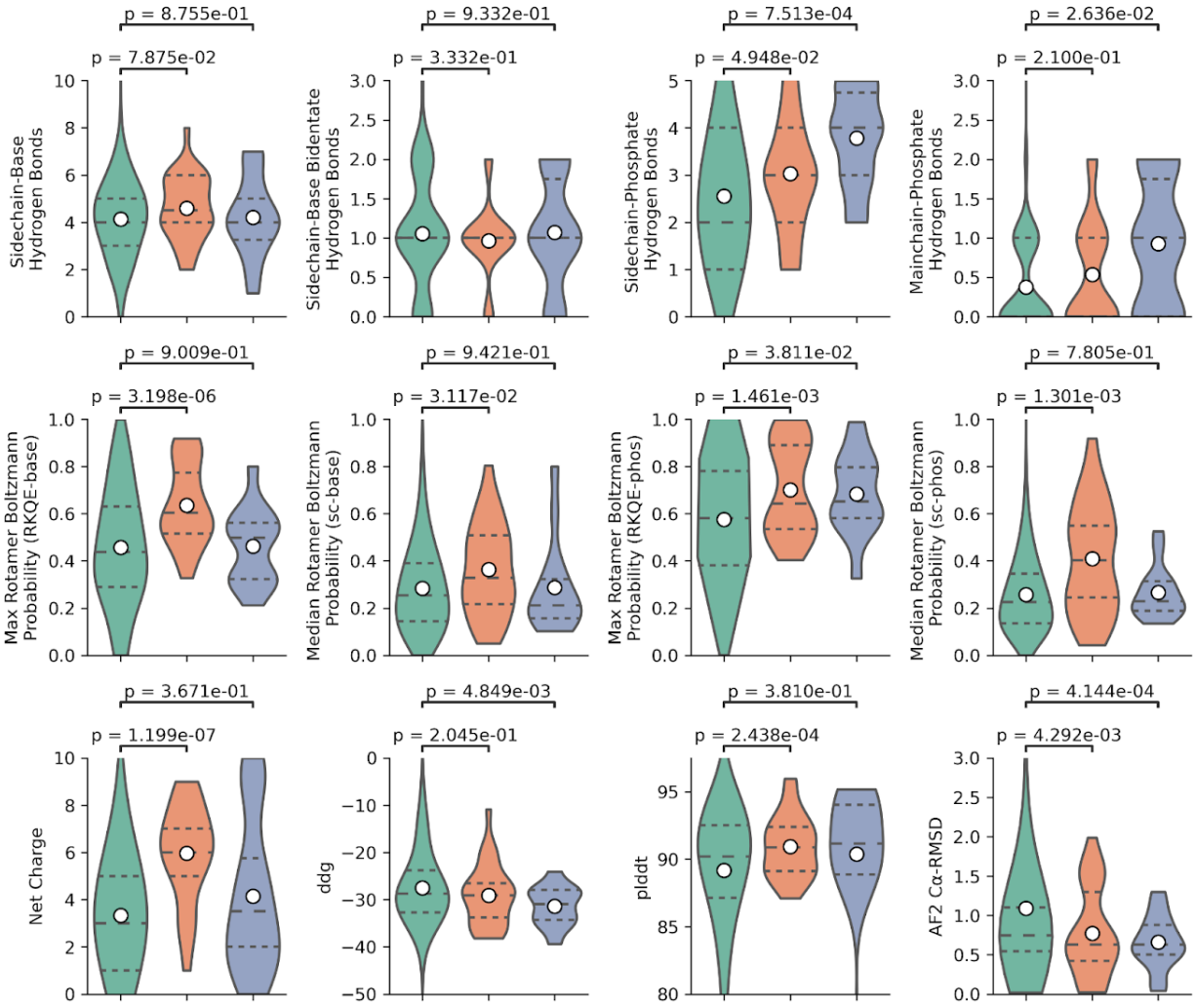


Figure 2.16 Power of computational metrics to predict binders. Metrics were calculated on all designs broken down into nonbinders (green, n=216), nonspecific binders (orange, n=30), and specific binders (blue, n=14). Hydrogen bonds and Net Charge are integer counts as determined by Rosetta. “ddg” is ΔΔG calculated by Rosetta. “plddt” is AF2 prediction pLDDT.

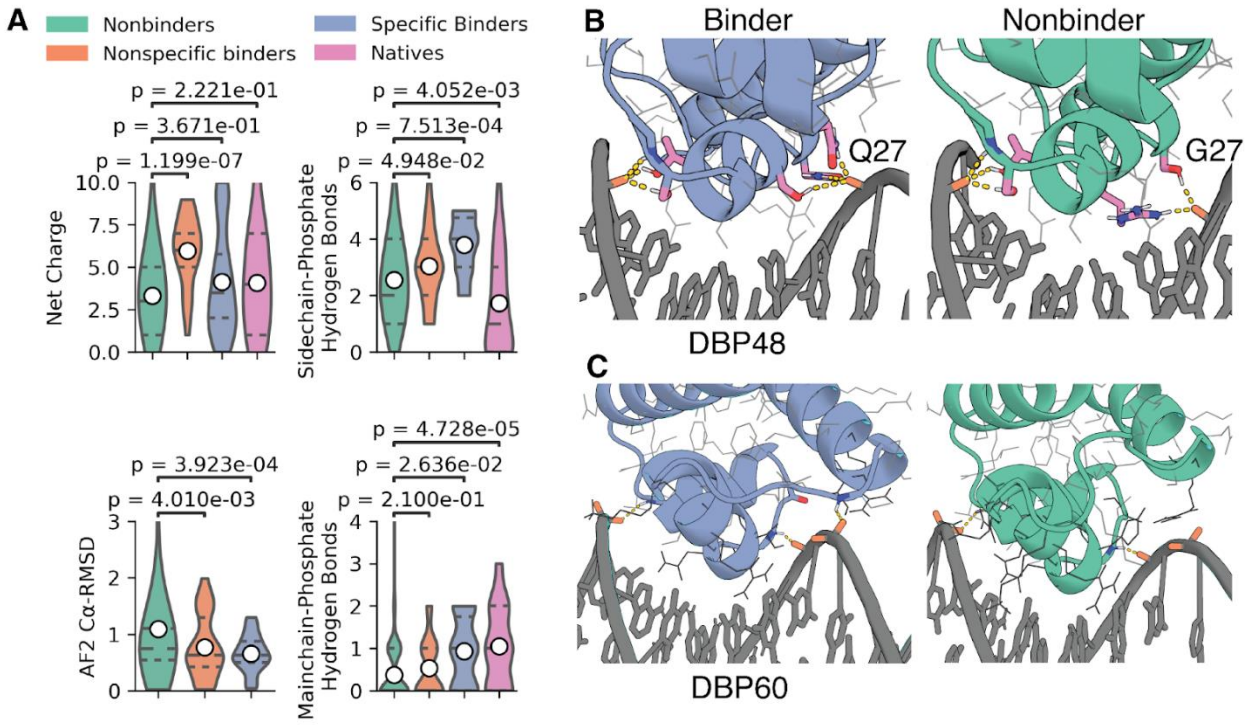


Figure 2.17 Mainchain-phosphate hydrogen bonds fix specificity and limit alternative target sites of DBP scaffolds. A, Comparison of the statistics calculated on nonbinders (n=216), nonspecific binders (n=30), specific binders (n=14), and native structures (n=41). High net charge was correlated with nonspecific binder designs, while sidechain-phosphate and mainchain-phosphate hydrogen bonds were more correlated with specific binder designs. In both cases, low Cα-RMSD of the initial design model to the AF2 model of the protein monomers was correlated with binding. Native structures have substantially more mainchain-phosphate hydrogen bonds than even the specific designs identified. **B,** Example of a key sidechain-phosphate hydrogen bond in the highly-specific DBP48 design and a nearly identical nonbinding design containing a Q27G that disrupted the interaction. **C,** Example of a mainchain-phosphate hydrogen bond in highly-specific DBP60 and a nearly identical nonbinding design with the terminal helix moved away from the phosphate backbone.

2.4.3 Applications and limitations

Our computational DNA binder design approach can generate DBPs that specifically bind arbitrary DNA sequences, including sequences that are not bound by known DBPs in the PDB or JASPAR databases. These designed DBPs function both *in vitro* and in living cells, as assessed through transcriptional repression and activation assays in both Escherichia coli and eukaryotic cells, respectively. The method samples structurally diverse HTH scaffolds to identify complexes that can facilitate specific contacts with the target DNA bases. The best designs were highly specific for their intended targets, and the crystal structure and specificity profiling assays strongly corroborate the computational design models. The modularity of the binding domains enables further

increases in specificity by rigidly positioning multiple modules along the DNA double helix using RF*diffusion.*

The design method presented in this chapter now enables the design of custom DNA-binding miniproteins to target specific DNA sequences for diverse applications in gene regulation and editing. Figure 2.18 highlights the orthogonality of five designed DBPs and their target sequences; this property is essential for creating precisely-controlled genetic circuits and for simultaneous editing of multiple DNA sites.

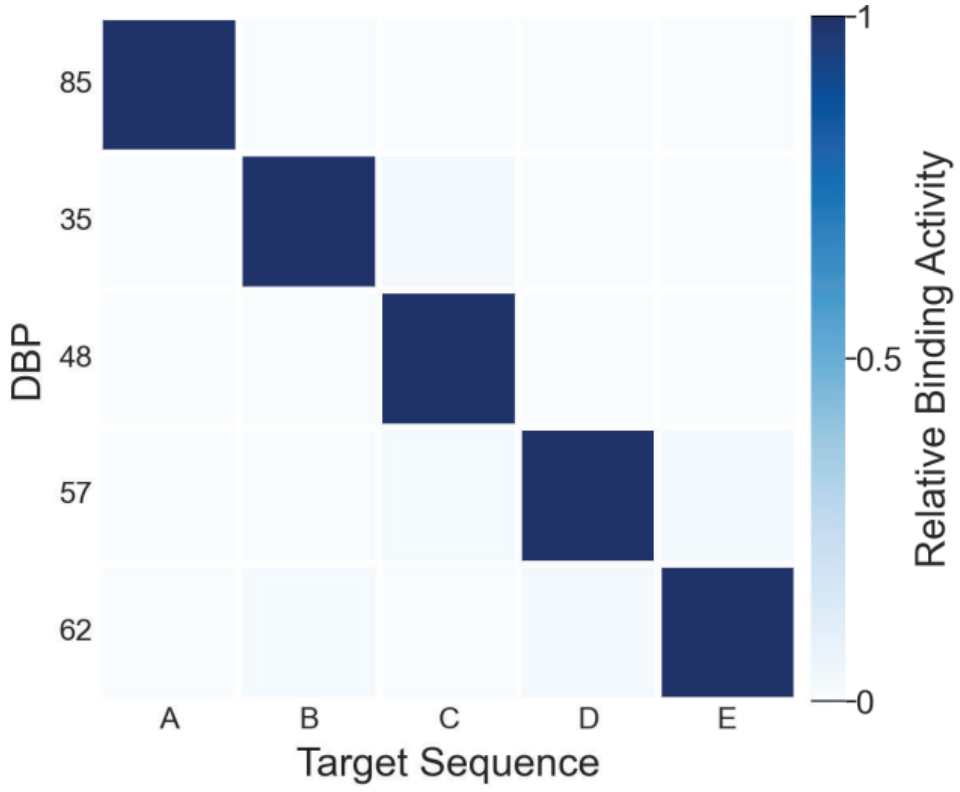


Figure 2.18 Designed DBPs achieve a high level of orthogonality. All-by-all orthogonality matrix for 5 designed DNA binders screened by yeast display, normalized by row, at a DNA concentration of 1 μM (with avidity). All off-target pairs bind with, at most, a 5-fold weaker binding signal than their on-target counterparts.

While we focused on design of HTH domains as a proof-of-principle, the method should be extensible to DBP families beyond the HTH domains used here, in particular those using a helix in the major groove for recognition such Cys2His2 ZF domains or homeodomains, by generating scaffolds models with structure prediction tools. Using scaffolds containing extensive β-sheets or loops, such as p53-like transcription factors[80] could be more difficult due to their increased flexibility, but should still be feasible.

Beyond extending to a broader range of scaffolds, future design improvements would include consideration of sequence-dependent DNA shape, induced fit, and the role of water-mediated hydrogen bonds, all of which are known to play important roles in sequence recognition.[13,81] We focused on design against a fixed target DNA sequence for simplicity and because of the accuracy and computational costs of existing tools to model DNA conformational flexibility and water-mediated interactions.

Future routes to incorporating these features in a computationally-efficient manner may include further use of ML models in the design process, such as existing or future variants of RoseTTAFold-NA[82] (see chapter 3) and AlphaFold3. [83] These models may be able to implicitly model water-mediated interactions and DNA shape.

In addition, use of generative AI approaches in protein design, such as RF*diffusion*-allatom[84], have become commonplace and future variants may become capable of joint backbone generation while accurately modeling DNA conformation *(Author's note: my colleague and friend Andrew Favor is currently preparing a manuscript on his method to co-generate protein with DNA and RNA, called RF-polydiff)*.

The ability to incorporate designed DBPs into transcriptional regulators through homo- and heterodimerization should allow expansion of orthogonal TF-operator pairs for more complex gene circuits.[85] As outlined in section 2.3.4, using RF*diffusion* it should be straightforward to fuse DNA-binding miniproteins together in a single chain in defined spatial orientations to allow specific targeting of longer target sites, or link DBPs with epigenetic modifiers or other effector recruiting domains to provide functionality beyond transcriptional activation and repression.

Computationally designed DBPs are also well-suited for the simultaneous recognition of both DNA sequence and shape, including non-B-form DNA structures that may occur in ~13% of the human genome.[86]

For any given backbone, there will likely be limitations in the ability to target certain DNA sequences, which likely explains the challenges in generating Cys2His2 zinc finger and other native scaffolds to bind to some target sites. Our approach provides a powerful new method for building DBPs with backbones tailored to specific sequences of interest, and we anticipate the method and the sequence specific designs should be widely useful in synthetic biology and other areas requiring sequence-specific DNA recognition.

2.4.4 Conclusion: have we achieved Aim 1?

The short answer is yes, technically. We have successfully created a method to computationally design DBPs with *de novo* docks, novel interfaces, and strong specificity for previously un-targeted DNA sequences.

The long answer is no. While we love to jokingly congratulate one another on having "solved DNA binder design," our method has substantial limitations that necessitate further development. There are currently at least 3 graduate students in the Baker lab, not including myself or Robert, whose entire theses are dedicated to "computational design of sequence-specific DBPs." The success rate of the method presented here is simply too low to be practical for continuing design work. To improve this, we continue to develop new tools for protein backbone generation, interface design, and especially design evaluation.

The even longer answer is partially. What started out as a few researchers applying methods designed for hydrophobic protein interfaces to create highly polar protein-DNA interactions has turned into one of the largest sub-groups of the Baker lab. There have been at least twenty individuals involved in either designing DBPs or using DBPs to design more advanced systems. These include DBPs fused to peptide-controlled hinge domains, resulting in novel transcription factors that can be induced by presence or absence of a separate helical peptide. Since we first designed sequence-specific DBPs successfully after years of trying, many people both inside and outside our institute have collaborated with us. I see this work as a crucial stepping-stone to eventually being able to truly design DBPs for *any* DNA sequence or structure.

Chapter 3. Structure prediction for protein-DNA complexes

3.1 Introduction to chapter 3

3.1.1 Structure prediction before and after AlphaFold

Attempts to predict the 3-dimensional structures of proteins based on their amino acid sequences date back to Anfinsen's discovery that such a thing ought to be possible—in the 1950s.[87] For 70 years, solving the "protein folding problem" was arguably *the* central focus of experimental, theoretical, and computational work in structural biology. In 2020, combining the results of all that work with further advances in machine learning and computing hardware, DeepMind created AlphaFold2 (AF2).[51] It cannot be overstated how much AlphaFold changed the field, and it released to the public about a month after I started studying DNA-binding proteins.

Unfortunately for the world and my work c. 2021, AF2 was limited to predicting the structures of proteins. That is, it had no concept of DNA. Indeed, the original AF2's restriction to monomeric, soluble proteins without ligands significantly limited its direct usefulness in many applications.[88] *Author's note: Many researchers around the world began studying what I call "AlphaFold-ology," i.e. trying to push the network beyond its original purpose using a variety of clever inference strategies.*

Around this time, my supervisors had been involved in creating AF2's first significant competition, called RoseTTAFold (RF).[89] While RF was directly inspired by AF2 and had noticeably worse performance, I had the advantage of working in the same lab that had created it. As an early-stage PhD student trying to design and model protein-DNA complexes, I had the opportunity to one-up AF2 with a fairly obvious extension from amino acids to nucleic acids.

3.1.2 Motivation in the context of design work

By January 2022, our efforts to computationally design sequence-specific DBPs had met with very limited success. That is to say, we had made one or two binders to one or two DNA targets, despite screening tens of thousands of designs (see Chapter 2). We were constantly looking for new ways to filter our designs to increase success rates. The most impactful filter, despite not modeling the DNA at all, was still AF2. We were hopeful that a structure prediction module capable of predicting DNA and protein could push us over the edge to truly "solve" DBP design.

Author's note: Around this time, I helped write an NSF grant for deep learning methods in studying biological macromolecules. We included descriptions of LigandMPNN, RosettaFoldNA, and a method called Generalized Atomic Accuracy Predictor (GAAP). I helped test this last method, which seemed especially suited to the task of filtering designed DBPs. However, its created graduated shortly after and I moved on to work on

structure prediction instead of following up on his work. For a description of GAAP and its performance on protein-DNA complexes, I direct you to Nao Hiranuma's dissertation, titled "Protein structure accuracy prediction with deep learning and its application to structure prediction and design." [90]

3.2 RosettaFoldNA

Primary reference for this section:

Baek M, McHugh R, et al. "Accurate prediction of protein-nucleic acid complexes using RoseTTAFoldNA." *Nature Methods*, 2023.

Author's Note: RosettaFoldNA was published in Nature Methods on November 23, 2023. For the convenience of the reader, I have reproduced the relevant portions of that work here.

3.2.1 Architecture

The architecture of RoseTTAFoldNA (RFNA) is illustrated in Figure 3.1. It is based on the three-track architecture of RoseTTAFold[89], which simultaneously refine three representations of a biomolecular system: sequence (1D), residue-pair distances (2D), and cartesian coordinates (3D). We extended all three tracks of the network to support nucleic acids in addition to proteins. The 1D track in RoseTTAFold has 22 tokens, corresponding to the 20 amino acids, a 21st "unknown" amino acid or gap token, and a 22nd mask token that enables protein design; to these, we added 10 additional tokens, corresponding to the 4 DNA nucleotides, the 4 RNA nucleotides, unknown DNA, and unknown RNA. The 2D track in RoseTTAFold builds up a representation of the interactions between all pairs of amino acids in a protein or protein assembly; we generalized the 2D track to model interactions between nucleic acid bases and between bases and amino acids. The 3D track in RoseTTAFold represents the position and orientation of each amino acid in a frame defined by three backbone atoms (N, CA, C), and up to four chi angles to build up the sidechain. For RoseTTAFoldNA, we extended this to include representations of each nucleotide using a coordinate frame describing the position and orientation of the phosphate group (P, OP1, OP2), and 10 torsion angles which enable building up of all the atoms in the nucleotide. RoseTTAFoldNA consists of 36 of these three-track layers, followed by four additional structure refinement layers, with a total of 67 million parameters.

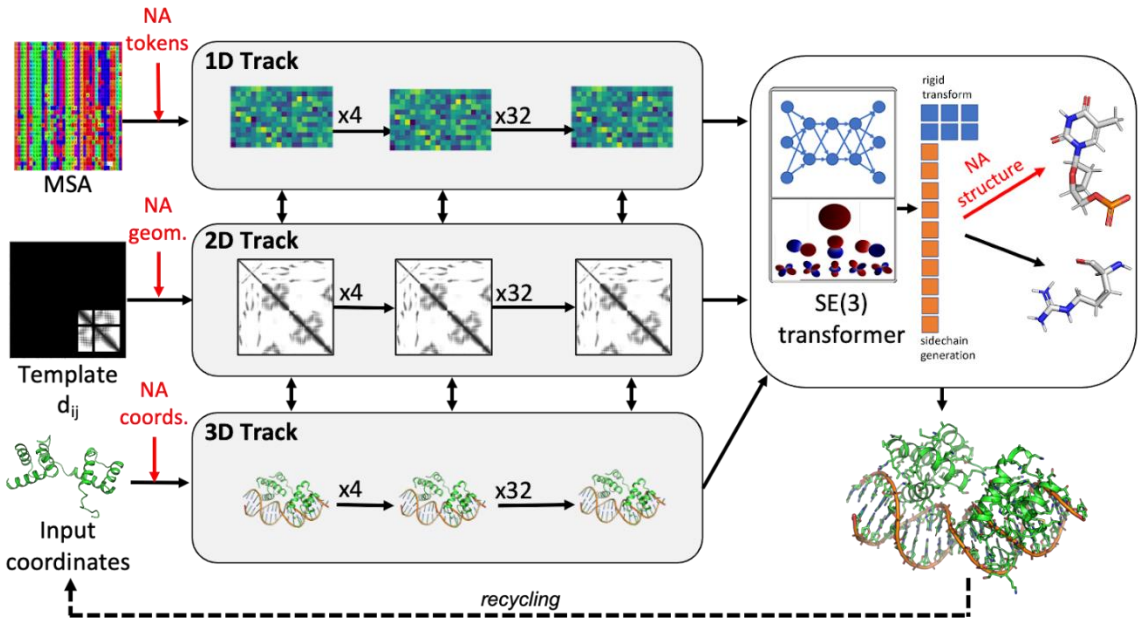


Figure 3.1 Overview of the architecture of RoseTTAFoldNA. The three-track architecture of RoseTTAFoldNA simultaneously updates sequence (1D), residue-pair (2D) and structural (3D) representations of protein/nucleic acid complexes. The areas in red highlight key changes necessary for the incorporation of nucleic acids: inputs to the 1D track include additional NA tokens, inputs to the 2D track represent template protein/NA and NA/NA distances (and orientations), and inputs to the 3D track represent template or recycled NA coordinates. Finally, the 3D track as well as the structure refinement module (upper right) can build all-atom nucleic acid models from a coordinate frame (representing the phosphate group) and a set of 10 torsion angles (6 backbone, 3 ribose ring, and 1 nucleoside).

3.2.2 Training and validation data

RosettaFoldNA was trained on a combination of protein monomers, protein complexes, RNA monomers, RNA dimers, protein-RNA complexes, and protein-DNA complexes, with a 60/40 ratio of protein-only and NA-containing structures. Assemblies with more than two chains (counting the DNA double helix as a single unit) were broken into pairs of interacting chains. For each input structure or complex, sequence similarity searches were used to generate multiple sequence alignments (MSAs) of related protein and RNA molecules. MSAs were not computed for DNA molecules. During training, 10% of the clusters were withheld for model validation.

The protein and protein complex data used in training was identical to that used in training RoseTTAFold2. Additional data from RNA and protein/nucleic acid complexes was added to this. To construct this dataset, all PDBs solved by NMR, crystallography, or cryoEM at better than 4.5Å resolution were collected. A dataset was constructed considering all PDB structures published at or before April 30, 2020, and collecting:

· All RNA single chains and all RNA duplexes. A duplex was defined by looking for pairs of RNA chains making at least 10 hydrogen bonds.

· All interacting protein/nucleic acid pairs. Interacting pairs were defined by counting the number of 7Å contacts between protein Cαs and any (non-hydrogen) nucleic acid atom; if there were more than 16 such contacts, the pair was considered interacting. Nucleic acid duplexes were included if the DNA or RNA chains made at least 10 hydrogen bonds.

For modeling, the full-length sequence was used. All nonstandard bases/amino acids were converted into a backbone-only "unknown" residue type. The dataset size was 7396 RNA chains and 23583 complexes. These were then clustered based on sequence similarity (hhblits[91] with a cutoff of E-value < 0.001 for proteins, or 80% sequence identity for RNA molecules), yielding 1632 nonredundant RNA clusters and 1556 nonredundant protein/NA clusters. These clusters were then split into training and validation sets, with clusters chosen for the training set; an example which contained any member (NA or protein) of a validation set cluster was assigned to the validation set. This led to 199 protein/NA clusters and 116 RNA clusters in the validation set.

Author's Note: For RNA, 80% sequence identity clustering is not necessarily effective at producing non-redundant clusters – a more refined strategy would cluster based on predicted or observed secondary structures.

Multiple sequence alignments (MSAs) were then created for all protein and RNA sequences in the training and validation set. Protein MSAs were generated in the same way as RoseTTAFold, using hhblits at successive E-value cutoffs (1e-30, 1e-10, 1e-6, 1e-3), stopping when the MSA contains more than 10000 unique sequences with >50% coverage. RNA MSAs were generated using a pared-down version of rMSA[92] that removes secondary structure predictions: sequences were searched using blastn[53] over 3 databases (RNAcentral[93], rfam[94], and nt[79]) to first identify hits, then using nhmmer to re-rank hits. We again use successive E-value cutoffs, stopping when the MSA contains more than 10000 unique sequences with >50% coverage.

Finally, to improve generalizability of protein/DNA interactions we added a few ways of randomly perturbing inputs during training. As many crystal structures of protein/DNA complexes involve short DNA chains with the binding motif in the middle, initial versions of the model had a strong preference to binding in the middle of any provided sequence. To deal with this, we added a random padding of 0-6 nucleotides to both ends of all native structures: a) containing double-stranded DNA, and b) making at least 3 base-specific contacts (using a cutoff distance of 3.4Å). This yielded 580 protein/DNA complexes. These added residues were not included in loss calculations but were

present in the predicted structures. Additionally, we also performed *negative training* for these same 580 complexes; all DNA bases forming base-specific contacts to the bound protein were randomly mutated (maintaining Watson-Crick base pairing), and the model was trained to move the protein and DNA far apart (by favoring the 6D "distogram" loss to place all its probability mass in the final bin).

For an independent test set, we took all structures published to the PDB between May 2020 and June 2022. Selection criteria and preprocessing were the same as for the training and validation data with two exceptions: a) only complexes fewer than 1000 residues plus nucleotides in length were considered; and b) for complexes containing more than one unique protein chains, paired MSAs were created by merging sequences from the same organism into a single combined sequence (following prior work by my friend Ian Humphreys).[95] This gave us 91 complexes with one protein molecule plus a single RNA chain or DNA duplex, 43 cases with a single RNA chain, and 106 cases with more than one protein chain or more than a single RNA chain or DNA duplex.

3.2.3 Loss Functions

The model was trained using a loss function similar to RoseTTAFold, where we take the weighted sum:

$$loss = w_{seq} \cdot seq + w_{6D} \cdot 6D + w_{str} \cdot str + w\_tors \cdot tors + w\_err \cdot err$$

Above, seq is the masked amino-acid recovery loss (no masking is applied to nucleotide sequences); 6D is the 6-dimensional "distogram" loss[96]; str is the structure loss, consisting of the average backbone FAPE loss[51] over all 40 structure layers of the network plus the allatom FAPE loss for the final model; tors is the torsion prediction loss averaged over the 40 structure layers; and err is the loss in pLDDT prediction.

FAPE loss is extended to nucleic acids in a straightforward manner from how it is implemented for amino acids. For backbone FAPE loss, the phosphate group in the nucleic acid backbone is treated as the nucleotides "frame," in the same way that N-Cα-C is used as an amino acid frame. For nucleic acid allatom FAPE loss, three-atom frames are constructed corresponding to each of the 10 "rotatable torsions" (see below for the definition), where the frame consists of the two bonded atoms defining the torsion plus an additional bonded atom, closer to the phosphate group in the bond graph. The cross product of these 10 frames with all atoms is used to calculate FAPE loss.

Following training with the above loss function, an additional "finetuning" phase is carried out, where additional energy terms are added to the loss function enforcing reasonable model geometry:

$$loss_{finetun} = loss + w\_LJ \cdot LJ + w\_hbond \cdot hbond + w\_geom \cdot geom + w\_pairerr \cdot pairerr$$

Above, LJ and hbond are the Lennard-Jones and hydrogen bond energies of the final structure (normalized by the number of atoms), using a reimplementation of the corresponding Rosetta energy terms[97]; geom is a term that enforces ideal bond lengths and bond angles around the peptide or phosphodiester bond connecting residues/nucleotides; and pairerr is a predicted residue-pair error.[51] The functional form of the geom term is identical to that of RoseTTAFold2, a linear penalty with a "flat bottom" plus or minus 3 degrees/0.02 Å from the ideal values.

3.2.4 Network training

The network was trained in two stages: an initial training period, and a fine-tuning period. In both, input structures were divided into 5 pools: a) protein structures, b) "distilled" protein structures (consisting of high-confidence AlphaFold2 predictions), c) protein complexes, d) protein/NA complexes, and e) RNA structures. Training sampled from each of these pools with equal probability (though later in training protein/NA frequency was increased to 25% and RNA frequency lowered to 15%). For both pools containing "complexes," an equal number of positive and negative examples were used in training. Negative examples consist of non-binding proteins or protein/NA pairs; the structure loss only penalizes each component individually, and the 6D loss favors placing negative binding examples far apart.

Examples larger than 256 residues/nucleotides in length were "cropped" to 256 residues in length. For protein-only data these crops were continuous sequences; for nucleic acids and nucleic-acid/protein complexes the cropping was a bit more complex. A graph was constructed where sequential residues/nucleotides had edges with weight 1, Watson/Crick base-paired nucleotides had weight 0, and protein/NA bases closer than 12Å (Cα to P) had a weight of 0. In negative cases, a single random protein/NA edge was given weight 0. Then minimum-weight graph traversal starting from a randomly chosen protein/NA edge was used to crop the model down to 256 residues/nucleotides. For RNA-only models the same strategy was used, though the starting point was a random nucleotide.

Training was carried out in parallel on 64 GPUs. A batch size of 64 was used throughout training with a learning rate of 0.001, decaying every 5000 steps. The following weights were used: w_{seq}=3.0, w_{6d}=1.0, w_{str}=10.0, w_{tors}=10.0, w_{err}=0.1. The Adam optimizer was used, with L2 regularization (coeff=0.01).

Following ~1e5 optimization steps, fine-tuning training was carried out. Here we increase the crop size to 384 and effective batch size to 128 and reduce the learning rate to 5e-

4. We used additional loss terms with weights w_{geom}=0.1, w_{LJ}=0.02, w_{hbond}=0.05, and $w_{pairerr}$=0.1, and optimized for an additional 30000 minimization steps. All told, training took approximately 4 weeks.

Author's Note: In reality, the training process was completed at least seven times over the course of about 8 months. In each iteration, we either made a substantial change to some handling of the data or fixed a significant bug. Example changes between versions include the introduction of paired MSAs for protein-RNA complexes and implementation of overlap prevention for homodimeric proteins. The core network architecture and loss function did not change between versions.

3.2.5 Evaluation methods and results

I evaluated the accuracy of predictions using several metrics. First and foremost is lDDT, which measures the accuracy of all *local* atom-pair distances, averaged across all atoms in the structure.[98] I also used interface_lDDT (i_lDDT), which is lDDT averaged over only the atoms found at the interface between chains in the correct structure. Furthermore, I used metrics from the CAPRI computational docking competition[99]: i_rms, the root-mean-square-deviation calculated on backbone atoms of interface residues after alignment on the same atoms; l_rms, the root-mean-square-deviation calculated on backbone atoms of the protein following alignment on all atoms of the DNA; and f_nat, the fraction of interface residue-pair interactions found in the native structure that are reconstructed in the prediction.

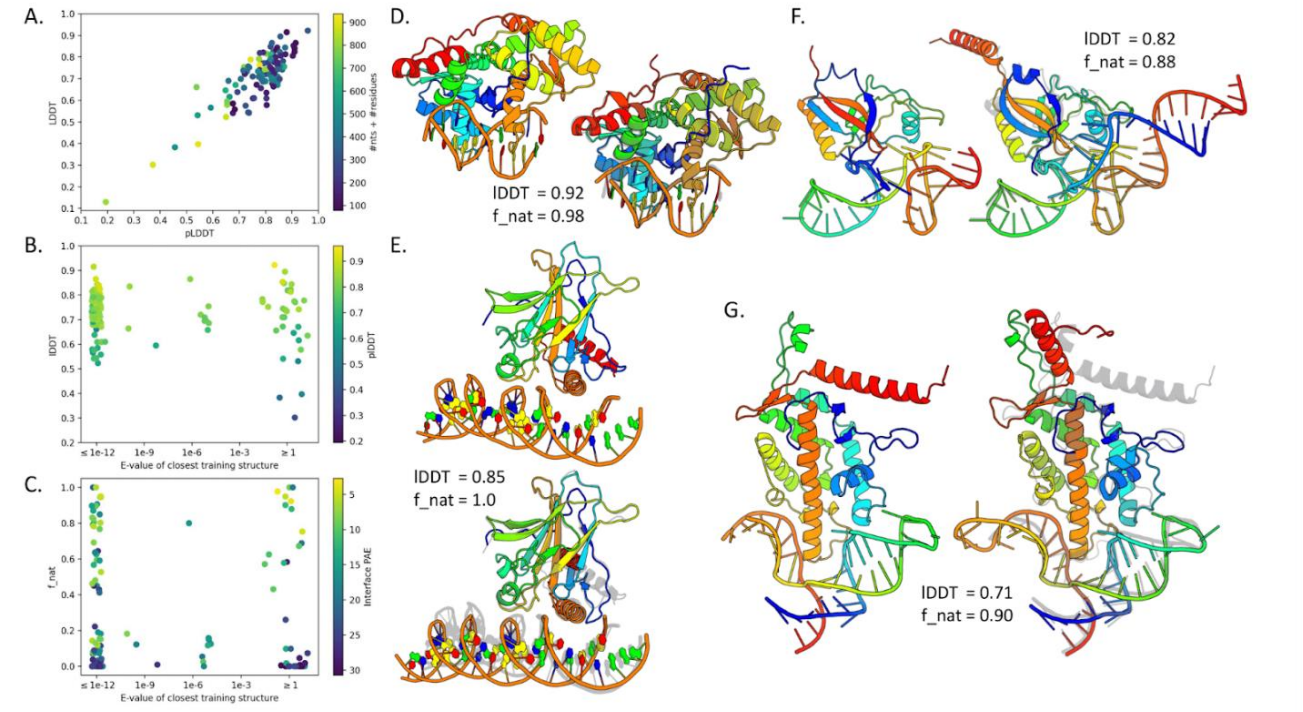


Figure 2. Protein - nucleic acid structure prediction (previous page). (A-C) Summary of results on 32 Protein/NA cluster representatives from the validation set and 84 Protein/NA structures released since May 2020. (A) Scatterplot of prediction accuracy (true lDDT to native structure) vs prediction confidence (lDDT predicted by the model) shows that the model correctly identifies inaccurate predictions. (B) The model seems to generalize well, with no clear performance difference between structures with and without sequence homologs in the Protein/NA training set. (C) Scatterplot of native interface contacts recapitulated in the prediction (f_nat) versus sequence similarity to training data. 35% of predictions are ranked “acceptable” or better by CAPRI metrics, and 78% of those with high confidence (mean interface PAE < 10). (D-G) Four examples of Protein/NA complexes without sequence homologs in the training set (PDB ids 5hlt, 3q05, 1p6v, and 4o26).[100–103]

I evaluated RFNA structure prediction performance on the combined validation and test sets (described in 3.2.2) of 224 monomeric protein/NA complexes. These inference predictions used 20 recycling steps and were otherwise identical to training steps. Results for this set are shown in Figure 3.2.

The predictions are reasonably accurate, with an average lDDT of 0.73 and 29% of models with lDDT>0.8 (19% of clusters, Figure 3.2A), and about 45% of models identify greater than half of the native contacts between protein and NA (f_nat > 0.5, 35% of clusters, Figure 3.2C). RoseTTAFoldNA, like RoseTTAFold and AlphaFold, outputs not only a predicted structure but also a predicted model confidence, and as expected the method correctly identifies which structure models are accurate. Although only 38% of the complexes (28% of clusters) are predicted with high confidence (mean interface PAE < 10), of those, 81% (78% of clusters) correctly model the protein/NA interface (“acceptable” or better by CAPRI metrics[99]). Over the 33 clusters with no detectable sequence similarity to training protein/NA structures, the accuracy is similar (average lDDT=0.68 with 24% of models >0.8 lDDT and 42% with f_nat > 0.5), and the model is still able to correctly identify accurate predictions—24% of predictions in this subset are predicted with high confidence, of which all 8 have acceptable interfaces according to CAPRI metrics. Four predictions of structures with no sequence homologs in the training set are shown in Figures 2D-G. These include the endonuclease BpuJ1, tumor antigen p53, SmpB bound to a tRNA-like RNA domain, and components of a telomerase reverse transcriptase. Inaccuracies in these predictions can be found in flexible terminal regions (Figures 3.2E, G), a slight tilt of the DNA double helix relative to the interface (Figure 3.2E), and slight deviations in RNA tertiary structure (Figures 3.2F, 3.2G), but the interfaces are clearly correct.

In cases where RoseTTAFoldNA fails to produce an accurate prediction, the most common cause is poor prediction of individual subunits, typically large multi-domain proteins, large RNAs (>100 nt), and small single-stranded nucleic acids. When the subunit predictions are accurate, the model in some cases identifies either the correct binding

orientation or the correct interface residues but not both. The remaining cases with completely incorrect interfaces often involve only glancing contacts or heavily distorted DNAs. It is possible that a different training schedule could reduce these errors, but it is likely due to limited training data in these regimes. Figure 3.3 illustrates some examples.

RoseTTAFoldNA prediction is not limited to complexes with only a single protein subunit. Figure 3.4 summarizes the performance of RoseTTAFoldNA on 161 multi-subunit protein/NA complexes, most of which are homodimeric proteins bound to nucleic acid duplexes. Performance is similar to that for monomeric protein/nucleic acid complexes, with an average lDDT=0.72 with 30% of cases >0.8 lDDT, and good agreement between confidence and accuracy (Figure 3A). Three examples are illustrated in Figure 3 (B-D), showing the ability of the model to predict complex structure as well as the “bending” of DNA induced by protein binding (Figure 3E). Figure 3 (F-G) shows another example where the relative positioning of protein domains is only made by co-predicting these complexes. Such effects would not be possible to predict by approaches that first generate models of the independent components and then rigidly dock them.

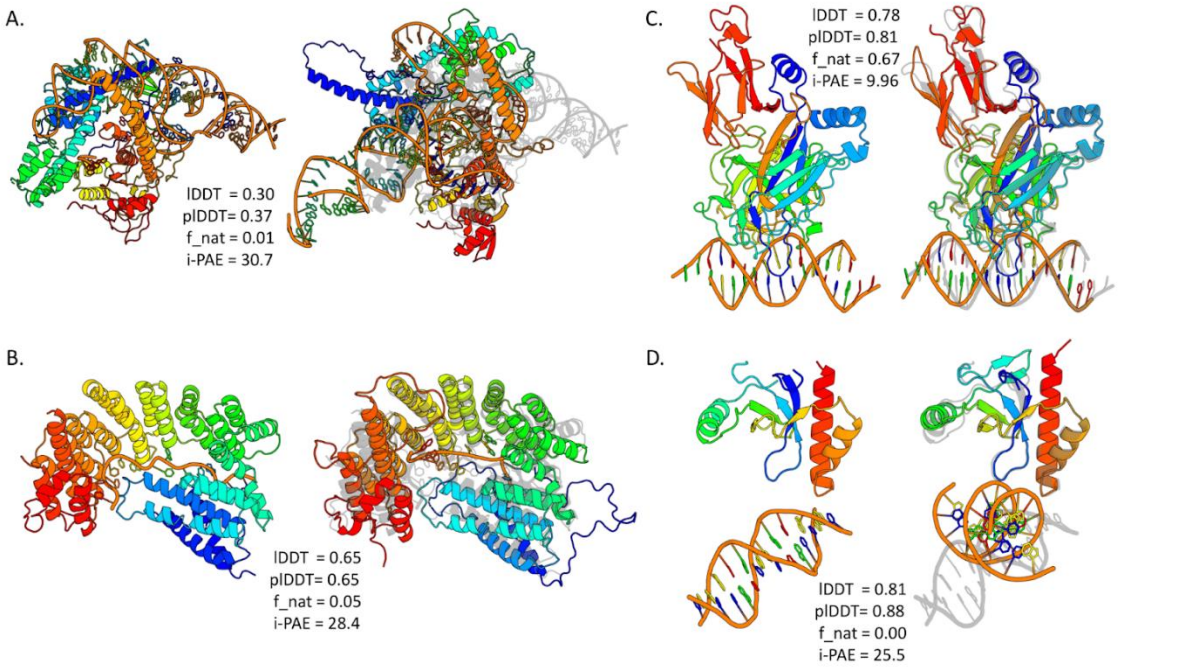


Figure 3.3 Failure modes of protein - nucleic acid structure prediction.
(A-D) Comparisons of representative predictions showing common failure modes of predictions in cases with no training-set homologs. Left is the deposited model, and right is the prediction. (A) Example where the individual subunits predict with poor accuracy, resulting in an incorrect overall complex (pdb ID: 6XMF[104]). Cases like this represent 50% of the examined failures and often result from very large or very small single-stranded nucleic acids (>100 or <20 nucleotides), large multi-domain proteins, or heavily distorted duplex DNAs. (B) Example where

the subunits predict with reasonable accuracy and the relative orientation is correct, but the details of the interface are wrong (pdb ID: 7A9X[105]). Cases like this represent 20% of the examined failures and can also result from small single-stranded nucleic acids or slight deviations in monomer structures. (C) Example where the subunits predict with high accuracy and the backbone-backbone binding mode is correct, but the interface is predicted at the wrong site on the DNA (pdb ID: 4J2X[106]). Cases like this represent 10% of the examined failures. (D) Example where both subunits predict correctly but the relative orientation and interface are incorrect (pdb ID: 7LH9[107]). Cases like this represent 20% of the examined failures and can result from distorted or non-duplex DNA structures or slight deviations in monomer structures.

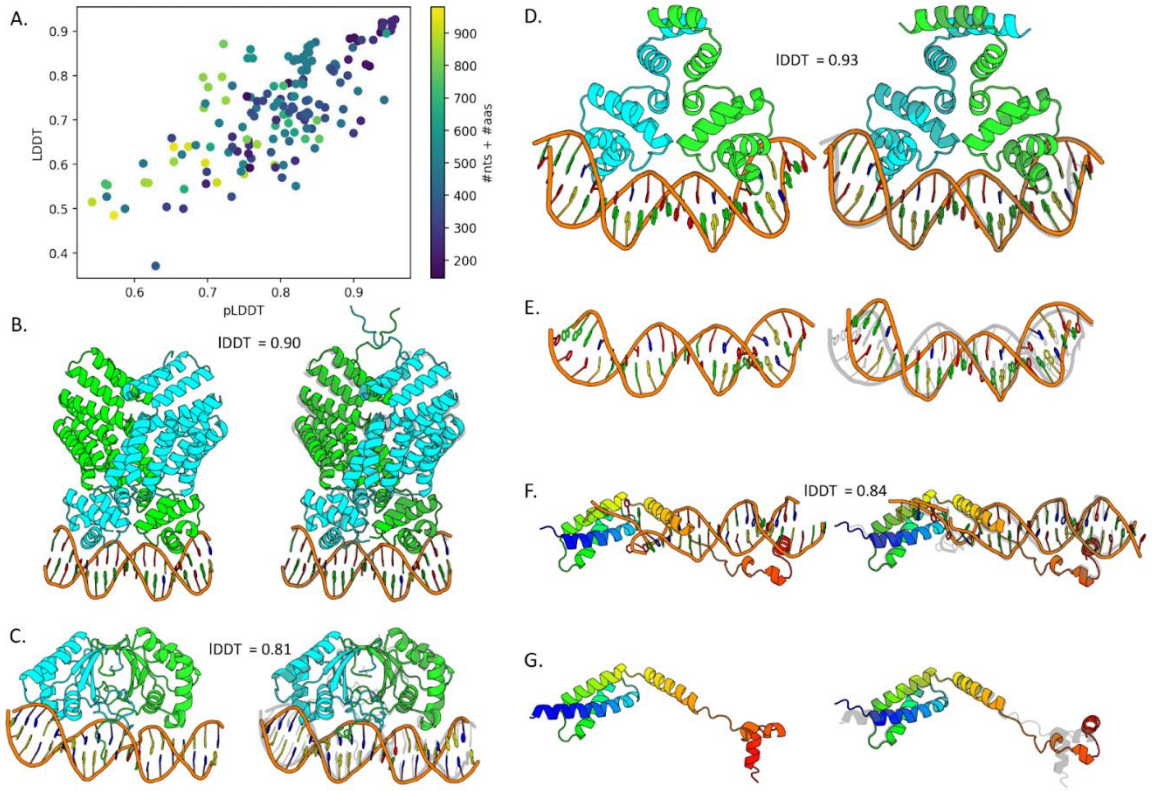


Figure 3.4 Modelling multichain protein-nucleic acid complexes. (A) Scatterplot of predicted model accuracy versus actual model accuracy for 161 protein/NA complexes with multiple protein chains or multiple nucleic acid chains/duplexes shows that the model accurately estimates error. (B-D, F) Examples of successful predictions without homologues in the training set, shown as the deposited model (left) and prediction (right).[108–111] (E, G) Example showing different predicted conformations of the same protein or DNA duplex alone (left) and with the other component (right), from the same complexes shown in D and F.

3.3 Discussion of chapter 3

3.3.1 Use cases and limitations of RFNA

While we were hopeful that RFNA could be used as a filter for designed DBPs, the model as published had little to no understanding of sequence specificity. Designed DBPs would sometimes dock with the correct orientation to the DNA, but there was no indication that RFNA could distinguish between specific and non-specific binders, or even non-binders.

Many researchers in biochemistry, molecular biology, genomics, and more were excited to try out RFNA when it released and in the years since. I have personally collaborated to use RFNA on a wide range of complexes, but none of those efforts have borne fruit. I believe this is largely due to the kind of complexes RFNA consistently fails on – large, multi-component complexes are simply outside of its capabilities.

RFNA's success rate of 30-40% on the kinds of complexes that had recently-solved structures unfortunately does not seem to transfer to the kinds of complexes that are difficult to solve by crystallography or cryoEM. Moreover, 30-40% success just feels unsatisfying to use for many people, although it was better than any other tool could do from sequence alone at the time.

3.3.2 Generative AI for protein-DNA structure prediction

Fortunately for all the scientists who found themselves disappointed by RFNA, DeepMind did not stop with AF2. About a year ago (i.e. in 2024), they released AlphaFold3 (AF3), which extends to all biological molecules with accuracy in line with what AF2 achieved for proteins.[83] The major advance is the use of generative AI – a diffusive model that operates on individual atom coordinates. Based on my own experience with RFNA and AF2, I think that much of the advantage of the generative approach comes from improvements in sampling. That is to say, the transformer-based structure prediction models already had exceptionally good confidence scoring – they would give the correct model a high score but would fail to find it on their own. AF3 and its derivatives in the field are better able to find the correct model because they sample at a wide range of noise levels.

I have contributed to development of our own in-house replication of AF3 (called RF-3; manuscript in preparation at the time of writing). Our model achieves equal or higher accuracy than AF3 on our validation dataset. The two major differences I added personally are (1) we have a much larger distillation set specifically for sequence-specific DBPs and (2) we use randomized padding of DNA sequences found in the PDB, not just distilled structures. While these changes are not expected to dramatically improve performance, I do expect our model to be somewhat better behaved when predicting long DNA chains.

Author's note: the randomized padding I describe relies on generation of uniform B-form DNA followed by alignment to the last non-overhanging base pair on either end. Both the uniformity and the imperfect alignment can introduce bias in the model's predictions, such as unrealistically straight chains and incorrect backbone geometry. While methods to predict DNA conformation exist for short chains such as transcription factor binding sites, I do not know of an accurate, generalized, and fast-running way to generate structures for long chains of DNA on the fly.

3.3.3 Conclusion: have we achieved Aim 2?

Computational structure prediction of protein-DNA complexes on the order of 100 – 1000 total amino acids + nucleotides is largely "solved" *for complexes that structurally resemble those found in the PDB.* Specifically, AF3 usually correctly predicts the structure of the DNA-binding protein domains and its correct orientation to DNA, but it still has several critical limitations, including:

1. Very large complexes, such as nucleosomes or viruses, will not fit in memory unless using a specialized setup (which almost all users do not have access to).
2. DNA is modeled as a single rigid conformation.
3. Water, counterions, and other solvents are not predicted (or at least, not well).
4. The model does not consistently distinguish between the cognate DNA and decoy DNA when predicting sequence-specific DBPs.

Problem 1 will eventually be solved by better hardware, even with the same software, but problems 2 and 3 are fundamental issues with the underlying assumptions that go into training a model like AF2 or AF3. We *have* to pretend that a rigid ground-state structure exists for our architecture and loss function to be logically sound. Furthermore, both 2 and 3 are problems not just of predicted structures, but of most experimentally-determined ones as well. Fully understanding the dynamics of and environmental effects on protein structure and complex formation are grand challenges in structural biology as a whole. If we intend to approach these challenges using AI models, we will need to continue collecting large, high-quality, publicly accessible datasets on which to train.

Problem 4 is the focus of the next entire chapter.

Chapter 4: Binding and specificity prediction for DBPs

4.1 Introduction to chapter 4

4.1.1 Motivation in the context of design and structure prediction

As I described in the discussions of chapters 2 and 3, our DBP design efforts were limited by a lack of protein-DNA structure prediction tools, while RFNA was unable to meaningfully predict sequence-specific DBPs. To combat both of these problems at once, I decided to develop a version of RFNA trained specifically for prediction of sequence-specific DBPs.

Author's note: I had a prior interest in studying how proteins recognize nucleic acids. I saw our design work at least partially as a way of learning more about molecular recognition. One of the first things I did after joining my lab was put together a dataset of proteins with both (a) DNA-bound structures in the PDB and (b) experimentally-determined binding profiles in the cis-BP database. I later used that same dataset to test my Rosetta implementation of structure-based specificity prediction.

4.1.2 Summary of alternative methods

Many attempts have been made to predict the binding specificity of transcription factors (TFs) and other DBPs. Indeed, my supervisor David Baker is an author on a paper titled "Protein-DNA binding specificity predictions with structural models" from 2005,[112] and my committee member Philip Bradley has one titled "Extensive protein and DNA backbone sampling improves structure-based specificity prediction for C2H2 zinc fingers" from 2011.[113] My rotation advisor Gabriel Varani also has one titled "An All-Atom, Distance-Dependent Scoring Function for the Prediction of Protein-DNA interactions from structure" from 2007,[114] which is conceptually similar to my own energy function optimization work described in section 2.2.6. The cis-BP database, which compiles experimentally-determined TF motifs, also contains millions of predicted TF motifs based on homology to TFs with known motifs.[115] AI-based methods to predict DBP binding profiles from an initial structure include DeepPBS from the Rohs lab[28] and NA-MPNN by Andrew Kubaney in the Baker lab (manuscript in prep at the time of writing). Both of these models can be used in conjunction with structure prediction by RFNA to generate motifs for DBPs without known binding modes, at least in theory. Methods to predict binding profiles using only unbound protein structure superimposed onto DNA date back to at least 2012.[116] There is also a long history of methods that predict DBP specificity based on sequence alone, usually within the scope of a single protein family. Tognon et al. benchmark several TF binding site prediction methods[117]; a review by Mou et al. covers recent advancements in deep learning methods for binding site predictions as of early 2025.[118]

4.2 Predicting specificity using mutation screening in Rosetta

4.2.1 Method and implementation

I unintentionally developed a method that is nearly identical to methods published 10 years prior, such as 3DTF[119] and PiDNA.[120] The concept is fairly simple: based on a co-crystal structure of a DBP and its cognate DNA from the PDB, evaluate the binding free energy (ΔΔG) for the original complex. Then, introduce all possible single mutations to the DNA without changing the backbone atoms (using X3DNA[54]) minimize the structures with Rosetta, and compute ΔΔG for each resulting complex. The ΔΔG scores are normalized by adding the minimum ΔΔG value to all of them and clamping the scores to range from 0 to 20 Rosetta energy units. Finally, the ΔΔG scores are converted to probabilities based on the Boltzmann distribution (note that the temperature factor kT is a free parameter which I optimized for my test set), and the resulting probabilities are taken for each mutation at each position to create a position probability matrix (PPM). The PFM was converted to a position information matrix (PIM) or position weight matrix (PWM) using tools in the logomaker python library[121]. Figure 4.1 shows a visualization of the protocol.

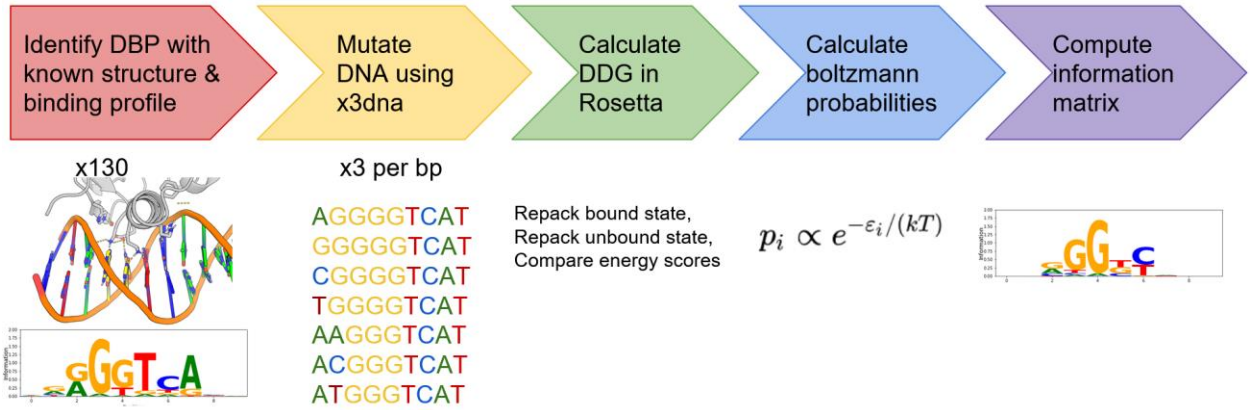


Figure 4.1 Visualization of Rosetta-based PPM prediction protocol.

4.2.2 Test dataset and results

I curated a set of DBPs for which both bound-state structures and specificity profiles (PWMs) were available in the PDB [[[]]] and cis-BP [[[]]], respectively. I manually aligned the DNA sequences in the PDB structures to the consensus sequences in the PWMs, discarding examples where the alignment was ambiguous or unclear. I found 130 protein-DNA complexes that fit these criteria, of which I was able to predict PPMs for 114.

I scored the PPMs by root-mean-square-deviation (RMSD) of all fields of the aligned PPMs, and compared them qualitatively by visualizing PIMs as sequence logos, also with logomaker.[121] I fit the free temperature parameter kT of the Boltzmann equation by scanning reasonable values and choosing the one which resulted in the minimum RMSD

across all 114 complexes, which turned out to be kT=3.0. I tested several variations of the protocol (e.g. repacking vs. relaxing vs. cartesian minimization of sidechains, flexible vs rigid DNA, coordinate constraints), and found that the best protocols were either A) repacking only the interface protein sidechains, followed by cartesian minimization of the full structure, both with all-atom coordinate constraints or B) take the ΔΔG of the complex as-is, with no repack or minimization. These conservative protocols performed well because the PPM calculations relied on small differences in ΔΔG, which could be caused by, for example, the formation of a single hydrogen bond.

Author's note: I used RMSD over the more common matrix-comparison measures KLD and cross-entropy, because both of those methods showed bias toward either very high or very low temperature parameters. At very high temperature, all profiles become uniform, while at very low temperature, all profiles become absolute with even the slightest difference in ΔΔG. Ultimately, I chose RMSD because it had the best correlation with PPMs that looked the most similar by eye.

Author's note: at one point, we tried using this protocol to filter designed DBPs, but it was simply too computationally intensive to run on hundreds of thousands of designs. Ultimately, this protocol was abandoned, though the dataset was later re-used.

The final protocol achieved a median RMSD of 0.23 across all 114 complexes, with 20 complexes having PPM-RMSDs below 0.15. Any RMSD below 0.2 can be considered a reasonable prediction, while RMSDs below 0.15 represent close matches to the true PPMs. Figure 4.2 shows examples of predicted and true PPMs with a range of RMSDs.

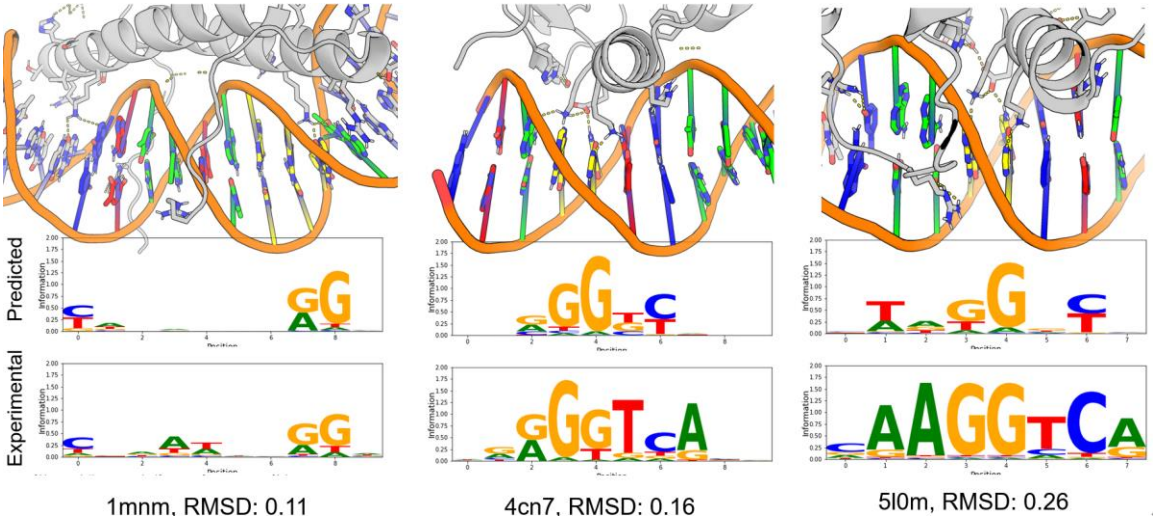


Figure 4.2 Examples of predicted and true PPMs of varying quality. Top: pymol visualizations of the deposited complex structures. Middle: predicted binding profile, visualized as PIMs. Bottom: true binding profile, also depicted as a PIM and aligned to the prediction. PDB IDs and profile RMSDs are listed below each example.

4.3 Fine-tuning RosettaFold to predict DBP binding / nonbinding

4.3.1 Concept and motivation

As described above, I saw a need for improved structure prediction tools for DBPs as filters for DBP design. Moreover, I found RFNA's lack of understanding of sequence-specificity to be a fundamental flaw in its performance. I saw fine-tuning the existing model for the task of predicting sequence-specificity as a clear road to improvement for RFNA itself and a promising path to improved DBP design success rates.

In comparison to alternate methods, we saw an opportunity to create a DBP specificity prediction tool that is structure-based but not dependent on an existing structure. Furthermore, we thought that our structure prediction model would be able to learn patterns in DBP specificity which are more general and transferable than what the sequence-only ML models could achieve.

Prior to this work, implementations of RosettaFold (including RFNA itself) already contained a concept of "negative" complexes—examples that contained two protein chains, or a protein and a nucleic acid, which did not actually form a complex. The original use of these negative examples was to simply drive them apart in the structures. That is, the loss applied to the distogram would use the maximum distance bin as the ground truth for all residue pairs across the negative interface.

For the original RFNA, the negatives originally consisted of randomly-selected proteins and randomly-selected DNA duplexes. We later expanded this definition of negatives to include artificially-generated (or "augmented") examples based on protein-DNA complexes in the PDB. To do this, we selected only complexes which had multiple sidechain-base hydrogen bonds at the protein-DNA interface, then randomly mutated the DNA sequence at those positions and considered those complexes as negatives. However, this implementation resulted in only a slight understanding of DBP specificity.

To further improve RFNA, I looked to the large databases of non-structural DBP specificity data, which I was already familiar with from my work on the Rosetta PPM predictor described above. We chose to incorporate this data in a way which required minimal changes to the architecture of RosettaFold: creating a dataset of known DBPs with DNA sequences known to be greatly preferred sequences (binding, positives) or known to not bind as strongly (non-binding, negatives).

However, we couldn't use these positives and negatives in quite the same way as we could the PDB-sourced and augmented protein-DNA complexes, as we did not have ground-truth structures. To circumvent this, we created a new auxiliary prediction head for the network, similar to the existing pLDDT and PAE confidence predictors, specifically to predict the "probability of binding," or p_bind.

4.3.2 Data sources and processing

Author's note: special thanks to Hanlun Jiang and Meghana Kshirsagar for their help in curating and processing the datasets described here. My training dataset is much larger and higher quality because of their contributions.

Broadly, we used multiple sources including the cis-BP[115], UniProbe[122], and TRANSFAC[123] databases as well as additional individual publications. In each case, we needed three types of data: protein sequences, binding DNA sequences, and non-binding DNA sequences. As the different sources used different formats and contained different types of experimental data, each required some extend of individualized processing. I will first describe how protein and DNA sequences were selected from each source, then describe further processing of proteins that was conducted for all sources.

Cis-BP

The original, and primary, data source for this project was cis-BP's set of direct motifs.[115] I used only those for which the raw data (as E_scores) was available. This limited this source to proteins studied by protein binding microarrays (PBMs); limiting cis-BP further to those examples with protein sequences available resulted in 2266 proteins.

The raw data from PBMs was available as enrichment scores (E_scores) ranging from -0.5 to +0.5 for all possible 8-basepair DNA sequences (8mers, 32768 unique scores). For each protein, the E_scores mostly follow a normal distribution centered around 0. I computed the mean and standard deviation of the distribution and took all 8mers with scores greater than 4 standard deviations above the mean as positives. For cases with fewer than 10 sequences passing that cutoff, I used 3 standard deviations instead. Cases which still had fewer than ten positive 8mers were discarded. As negatives, I took all 8mers with E_scores lower than the mean. This resulted in between 10 and 1000 positives and over 10,000 negatives, all exactly 8 basepairs in length, for each of 2067 proteins.

For use in creating distillation structures, I selected the top 3 most enriched 8mers for each protein.

UniProbe

The UniProbe database[122] compiles PBM experimental data and has significant overlap with the cis-BP direct motif data. As a result, this dataset was processed the same way as described for cis-BP, resulting in 488 sets of protein and DNA sequences.

SELEX and ChIP-seq data

Additional DBP specificity data were acquired from SELEX and ChIP-seq experiments (generously provided by Meghana Kshirsagar from her previous work[124,125] and other

sources[126,127]). Since these experiments result in detection of longer read sequences, we searched for regions of the read sequences that significantly matched the consensus motif from each dataset as constructed by MEME.[128] This resulted in at least 10 and at most 4,000 positive DNA sequences for each example, ranging in length from 14 to 40 basepairs. For distillation structures, we chose up to 10 top-ranked matches.

In cases where protein sequences were not provided in the source data, we matched the protein IDs to UniProtKB[129] and chose the most common isoform. To ensure that RosettaFold would be able to predict reasonable structures for these examples with longer DNA sequences, we predicted the complex structures with a version of RFNA fine-tuned on just the PBM data and discarded cases where the predicted interface was inconsistent with the computed binding motif.

As these data sources do not provide a clear way to identify non-binding DNAs, we generated random DNA sequences of the same length as the positives and for each protein selected 10,000 sequences which were incompatible with the computed binding motif.

Following these steps, the combined dataset up to this point included 2694 proteins with positive and negative DNA targets.

TRANSFAC

The TRANSFAC database[123] is by far the largest and most complete source for transcription factor binding data, but it is unfortunately proprietary and requires a paid license even for academic use. In addition to the high-throughput experiments used for the other sources, TRANSFAC compiles data from papers studying individual TFs and provides position frequency matrices (PFMs) for each TF as well as all observed bound sequences. In this case, we used the individual DNA sequences from the list as positives and generated random DNA sequences of the same length that did not match the PFM as negatives. For distillation, we took the 3 sequences from the list that best matched the overall PFM. The TRANSFAC database provides full-length protein sequences for all included TFs.

Combined with previous datasets, the TRANSFAC data increased the total number of unique DBPs with confirmed binding DNAs to 3399, in a dataset simply called “TF”.

Author’s note: the TRANSFAC dataset was incorporated into training later in the process; it was only used in the last training run. Note that TRANSFAC is also the only source we used that is not publicly available for free. While negotiating our paid license, we got explicit permission to use it in training an AI model and share the final model, as long as we did not share their dataset itself. Even so, this could create an issue for anyone trying to replicate our work.

Grouping by protein family

Most data sources provided annotations of information about the DBPs reported, including protein name, gene name, and protein family. However, the labeling and naming of protein families is highly inconsistent across sources. As such, I went through the lists of protein families provided and manually merged together different names for the same thing (such as “homeobox” and “homeodomain”, “C2H2_ZF” and “Zinc Finger C2H2”, etc.). To find less obvious groupings, I clustered all of the protein sequences using mmseqs2[50] and checked for cases where differently-labeled proteins clustered together. This latter approach also helped identify protein families for proteins without labels in the source data.

For the remaining cases that were unlabeled or labeled “unknown," I ran hhalign[46] against the identified DNA-binding domain sequences (see next section) and assigned each to the family of the best hit. Examples with no significant hits from hhalign were set aside as orphan TFs to be used in final testing of the method.

The data set was divided into training and validation groups by protein family. The validation families were chosen semi-randomly by sorting the families in order of number of examples and taking every seventh family, such that about 10% of total examples were in the validation set. I also made sure that at least one family with a large number of examples and low structural similarity to the large training-set families was included in validation. This family, which I often used as a litmus test for overall performance, was AP-2.

Identification of DNA-binding domains

Most data sources provided either unclear identification of the DNA-binding domains in the protein sequences, or none at all. However, we found that predictions of the full-length sequences were often poor quality due to large intrinsically-disordered regions and were not suitable for training.

After assigning the majority of proteins to families, I chose representatives for each of the ~80 distinct families in the dataset (generally, I just used the first example in each family alphabetically). For each representative, I identified the relevant DNA-binding domain using a combination of UniProt domain annotations[129], visual inspection of structures in the AlphaFold Protein Structure Database[130], and annotations in the source data. Then, I constructed Hidden-Markov sequence models (HMMs) for the representative domains using hhblits[91] (settings ‘-mact 0.35 -nodiff -n 4 -e 1e-3 -cov 75 -id 90’), and aligned each full-length protein sequence to its representative domain’s HMM with hhalign[46] (with setting ‘-id 100’,). The few cases which did not find hits in the hhalign search were discarded or resolved manually the same way as the representatives.

In the case of C2H2 zinc fingers, which can have a variable number of DNA-binding domains with high sequence similarity, the full-length protein sequences were used.

Protein MSA generation and clustering

Multiple sequence alignments (MSAs) were generated for the DNA-binding domains using the same script that was used for RoseTTAFold and RFNA (see section 3.2.2).

Domain sequences were clustered using mmseqs2[50] to a 1e-3 E-value.

Generation of distilled structures

To further leverage the body of non-structural work in studying DBPs, I generated a set of distilled protein-DNA complex structures, as was done previously by AlphaFold2 and RoseTTAFold for proteins. I used the same set of protein MSAs described above, and the top-ranked DNA sequences determined experimentally for each DNA-binding domain, also described above.

For each protein-DNA pair, the complex structure was predicted either by the original RFNA or by a previous version of the fine-tuned network. Predicted structures were filtered to those with high confidence metrics (mean pLDDT > 80 and mean interface PAE < 10). For examples with DNA sequences longer than 10 base pairs, we further filtered out predicted structures that did not form an interface at the correct site.

For all proteins, we also ran the same predictions with two copies of the protein present, along with DNA sequences padded out to at least 12 basepairs, to accommodate obligate homodimers. We used the dimer structures only when their predictions were more confident than their monomer equivalents. In total, there are about 27,000 predicted protein-DNA complex structures in this distillation set, referred to as “TF_distil”

Author’s Note: At one point, we tried additional filtering based on number of interface hydrogen bonds following a Rosetta constrained FastRelax, but we later discarded this idea. There is no universal number of hydrogen bonds that can be used for such a diverse group of DBPs, and we didn’t want to unnecessarily bias the network with structures refined by Rosetta.

4.3.3 Implementation

Architecture

As I briefly touched on in 4.3.1, the implementation revolves around an auxiliary prediction head that predicts the “probability of binding” (p_bind) for a protein and a DNA duplex, hereon referred to as the “binding head.” The output of the binding head is a single value ranging from 0 to 1.

The architecture is extremely simple, to the extent that I can reasonably provide here the python code for the binding head in its entirety (Box 4.1):

Box 4.1 Python code for the auxiliary binding head

```
class BinderNetwork(nn.Module):
    def __init__(self, d_pair=128, d_state=32, d_rbf=64, p_drop=0.15):
        super(BinderNetwork, self).__init__()

        self.rbf2attn = nn.Linear(d_rbf, 1)
        self.downsample = torch.nn.Linear(d_pair+2*d_state, 1)
        self.dropout = torch.nn.Dropout(p_drop)

        self.reset_parameter()

    def reset_parameter(self):
        nn.init.zeros_(self.downsample.weight)
        nn.init.zeros_(self.downsample.bias)
        nn.init.zeros_(self.rbf2attn.weight)
        nn.init.zeros_(self.rbf2attn.bias)

    def forward(self, pair, rbf_feat, state, seq, idx, bond_feats,
                    dist_matrix, same_chain):
        B, L = pair.shape[:2]

        # 1. get attention map
        # pair: (B, L, L, d_pair)
        attn = self.rbf2attn(rbf_feat)

        # 2. get logits
        left = state.unsqueeze(2).expand(-1,-1,L,-1)
        right = state.unsqueeze(1).expand(-1,L,-1,-1)
        logits = self.downsample(torch.cat((pair, left, right),
                                           dim=-1))

        # 3. dot product
        if (torch.sum(same_chain==0)==0):
            logits = logits.flatten()
            attn = attn.flatten()
        else:
            logits = logits[same_chain==0]
            attn = attn[same_chain==0]

        attn = F.softmax(attn, dim=-1)
        attn = self.dropout(attn)

        logits_inter = torch.mean( logits * attn, dim=0 ).nan_to_num()

        prob = torch.sigmoid( logits_inter )

        return prob
```

The input features to the binding head are “pair” and “state,” which are the output features of the 2D and 3D tracks of the RoseTTAFold architecture. These features are concatenated together, down-sampled by a linear layer with 288 parameters, and masked to only the cross-chain residue pairs. Then, the dot product is taken with a single attention layer with 64 parameters. Finally, we take the mean of the resulting logits and use a sigmoid function to convert to a probability.

The binding head itself has just these 352 parameters, but was trained while the full RosettaFold2 architecture was fine-tuned end-to-end.

Loss functions

As described in 4.3.2, the source protein-DNA complexes were divided into binding and nonbinding pairs, with many positive and negative DNA sequences for each protein. Positive protein-DNA complexes, both from the PDB and the TF dataset, were assigned a p_bind target score of 0.95, while negative pairs were assigned a p_bind target of 0.05. The predicted p_bind was evaluated by binary cross entropy (BCE) with the target p_bind.

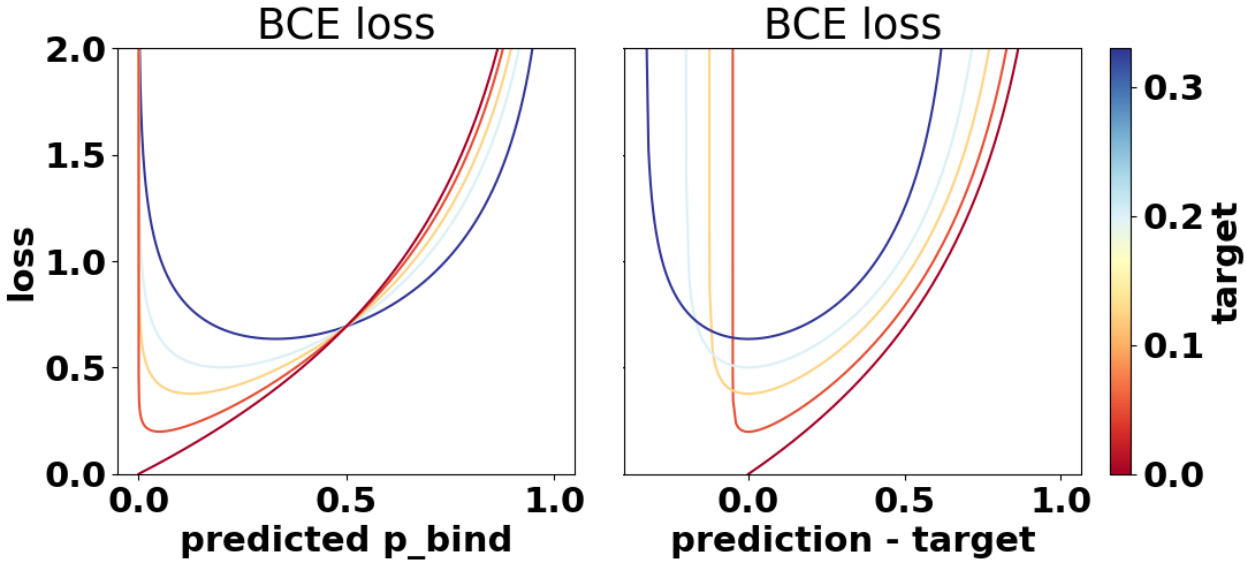


Figure 4.4 Plots of the binary cross entropy (BCE) loss function for a range of targets. Note how the loss curve is increasingly concave the farther the target gets from 0. While the minimum value of the loss changes at different offsets, it is always found where the prediction matches the target exactly. Larger target offsets from 0 also create more symmetrical loss function, which result in a smaller difference between the positive and negative examples' losses and therefore punish the network less for predictions closer to 0.5.

Author's note: We included the offset from 0 and 1 in the target values because the BCE function is approximately linear when approaching the target (e.g. BCE of 0.04 and 0 is 0.0408, BCE of 0.05 and 0 is 0.0513), and we didn't want to train a network that always predicts exactly 0 or 1. Figure 4.4 shows the effect of moving the target value on the BCE function. At one point, I thought the offset value, which causes the BCE to have a very strong gradient for over-confident predictions (e.g. BCE of 0.001 and 0.05 is 0.35 while BCE of 0.002 and 0.05 is 0.31 – a slope of 34!), was resulting in learning instability. After some testing with different offsets and even using a squared BCE function for increased concavity, I don't think it makes much difference at all.

Besides BCE, we also trained with all of the same loss functions used for RFNA, except that losses dependent on a ground truth structure were always 0 for examples in the TF set, as they could not be computed. Loss weights were largely unchanged from the fine-tuning stage of RFNA training, but in my final training run I increased the weight of the FAPE loss across interfaces and extended this parameter to include protein-DNA interfaces. Table 4.1 shows all of the loss weights used in my final training run.

Table 4.1 Loss weights for fine-tuning run TF9d

Loss description	*Loss name*	*Loss weight*
Bond lengths and angles in polymers	`bond`	`1.0`
Bond lengths and angles in small molecules	`atom_bond`	`1.0`
Torsion angles in polymers (?)	`skip_bond`	`1.0`
Distances in rigid groups	`rigid`	`1.0`
Clashing atoms	`clash`	`0.1`
Hydrogen bond geometries	`hb`	`0`
Lennard-jones potential	`lj_lin`	`0.75`
Distogram	`dist`	`1.0`
Masked token recovery	`str`	`10.`
All-atom lDDT	`lddt`	`0.1`
All-atom FAPE	`aa`	`3.0`
Non-protein FAPE	`lig_fape`	`1.0`
Protein / non-protein interface FAPE	`inter_fape`	`2.0`
Binding head BCE	`bind`	`0.1`

While not used as loss functions in the network, I tracked a few additional metrics to evaluate the performance of the model on the validation set during training, most notably precision and recall of TF examples.

Precision is defined as the proportion of all predicted positives that are also true positives, while recall is defined as the proportion of true positives that are also predicted positives. For both metrics, I considered examples where the network reported p_bind > 0.5 as predicted positives, while true positives were all examples using DNA sequences from the positive set. These metrics were tracked only for the non-structural TF dataset.

Training regimen

I trained the network using the same general protocol as was used in the fine-tuning stage of RFNA (see section 3.2.4). To this training protocol, we added the TF and TF_distil datasets, alongside the PDB monomer, distilled protein monomer, PDB protein complexes, PDB protein-NA complexes, and PDB RNA datasets. The sampling frequencies for these datasets used in the final training run are shown in Table 4.2. Note that positive and negative examples, including those of protein-multimer and protein-NA complexes in the PBD, are always sampled 1:1. This is to avoid introducing a general bias into the binding head, which was trained on all inputs that have multiple chains.

Within each dataset, protein or RNA clusters were sampled uniformly. Within each cluster, examples were sampled in proportion to the square root of their number of residues, to account for the effect cropping has on sampling of large structures. When sampling examples from TF datasets that were larger than the crop size of 384 tokens, we always included the full DNA duplex and a random contiguous section of the protein.

As with DNA sequences in RFNA training, we added random padding to DNA sequences in the TF and TF_distil datasets. For negatives, I ensured that the final DNA sequences after padding did not include any of possible positive sequences. Also for TF negatives, I ensured that each sampled sequence contained the same sub-sequence symmetry pattern (such as being its own reverse complement) as a randomly-selected positive sequence. For TF examples believed to be dimers, I padded the DNA sequences with additional unspecified DNA tokens out to a minimum length of 12 basepairs, as this was found to improve prediction quality.

The network was trained with 15% parameter dropout, 3 recycles per example, max crop size of 384, learning rate of 0.0005, and a loss accummulation batch size of 64.

Table 4.2 Dataset sampling frequencies for fine-tuning run TF9d

Dataset description	*Dataset name*	*Sampling frequency*
PDB protein monomer	pdb	0.10
Protein monomer distillation	fb	0.10
PDB protein complexes	compl	0.05
Generated negative protein complexes	neg_compl	0.05
PDB protein-NA complexes	na_compl	0.18
Generated negative protein-DNA complexes	neg_na_compl	0.18
PDB RNA-only	rna	0.02
PDB DNA-only	dna	0.02
TF positives	tf	0.075
TF negatives	neg_tf	0.075
TF distilled structures	tf_distil	0.15

Author's note: looking at these values now, I suspect that some of them are a bit unreasonable. For instance, 25% of examples being artificially-generated negative versions of protein-DNA complexes is almost certainly too high; neg_na_compl contains significantly fewer examples than na_compl, and the data points are of lower quality due to their augmented nature and the lack of cross-interface structural losses.We kept the tf and neg_tf datasets to fairly low rates to avoid destabilizing the network's structure prediction, but ultimately those are the only ones with new information over RFNA. I also think using the same binding head for both protein complexes and protein-DNA interfaces made learning unnecessarily difficult for the network (although it may also have helped prevent over-fitting to the relatively small TF dataset). While long enough training should theoretically resolve these issues, they highlight the inefficiency of our approach.

4.3.4 Iterative training and development process

Training run TF3

My first attempts at training RFNA for binding prediction started from the published RFNA checkpoint (named BFF22j). At this point, I used only the cis-BP data, did not have the TF_distil set, and did not have the manually-determined DNA-binding domains (I used a domain parsing script based on minimizing radius of gyration and maximizing core residues with Rosetta). I also did not have increased sampling frequency for protein-NA complexes, used a crop size of 256, did not have random padding applied to TF examples, and did not have increased weight on interface FAPE. The binding head also did not have the attention layer included and used just the linear down-sampling layer.

Aside from these differences, the early training runs were mostly the same as later ones. My first two attempts were unstable or overfit due to too-small batch size and/or the use of full-length protein sequences, but my third run, aptly named TF3, had some promising results.

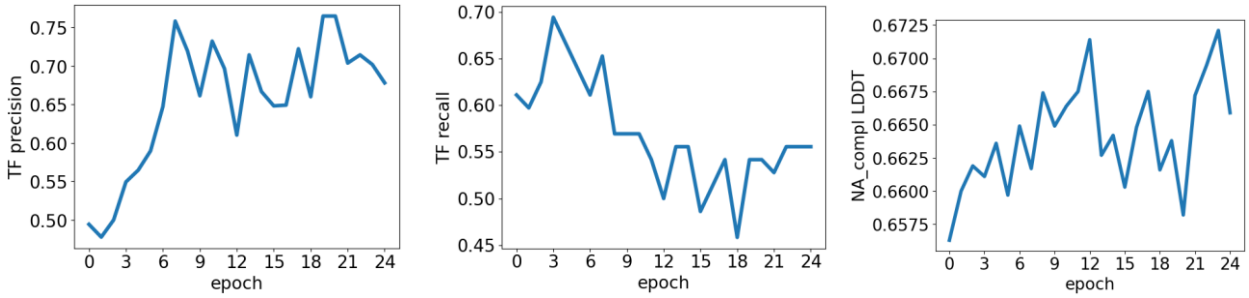


Figure 4.5 Training curves for run TF3. (left) Precision for binding vs non-binding DNAs in the TF validation set. **(center)** Recall for the TF validation set. **(right)** Structure prediction accuracy as lDDT for the PDB protein-NA complex validation set. All of these metrics are better with larger values. The x-axis shows training time in epochs, where each epoch is 75 learning steps and 4800 total examples. Note that some instability could be due to picking a different subset of positives and negatives in each run through the TF validation set.

As shown in Figure 4.5 above, the network was able to learn to distinguish the TF validation set with about 75% precision and about 55% recall, while also increasing mean lDDT on the PDB protein-NA validation set from 0.65 to 0.67. While a seemingly small difference, this is a pretty clear demonstration that using non-structural data can improve the performance of a structure prediction network. This result is impressive especially considering that it was trained on a fairly small dataset processed imperfectly for just over 20 epochs (~100,000 examples, compared to the ~2.5 million examples used for RFNA).

Training run TF4

Following the promising results of run TF3, I decided to port the code for training with the TF dataset to the then-new RF2-allatom (RF2aa) repository[84]. During this time, we also created an initial version of the TF_distil dataset and included it in training. Compared to RFNA, RF2aa has slightly worse general performance in predicting protein-DNA complexes; however, this did not seem to affect its ability to learn the TF binding task.

The main difference observed in TF4, compared to TF3, is a substantial increase in validation set precision alongside generally increased confidence. That is to say, TF4 tends to predict p_bind scores much closer to 0 and 1 while TF3 tends to predict a broader range of scores. Figure 4.6 shows training curves for TF4 as well as comparison to TF3 on a test set of the top-ranked DNAs for the TF set proteins.

Author's note: For run TF4, I did not separate the TF_distil set into training and validation the same way I did for the TF binding set. As such, many of the same proteins present in the TF binding validation set were also present in the TF_distil training set. While the distillation set uses a much smaller set of DNA sequences than the binding task, memorization of the distillation examples allowed this version of the network to perform much better than it should have on the validation set. This discrepancy is especially notable for the test set, which uses the same DNA sequences as the distillation set. This mistake was corrected for later training runs.

Training run TF7

Author's note: Runs TF5 and TF6 were both short-lived runs with minor differences from previous versions.TF6 in particular showed serious problems with over-fitting to the TF data while reducing accuracy on all structure prediction tasks, likely due to a too-high weight on the binder loss, over-sampling the TF dataset, or .

There are two main differences between TF4 and TF7: (1) I fixed the data leak between the TF_distil training set and the TF validation set, creating a new TF_distil validation set with the same split as TF; (2) I added about 500 additional proteins with positive and negative DNA sequences from SELEX and ChIP-Seq experiments. TF7 was one of my longest runs, running for about 100 epochs / 480,000 examples. Figure 4.7 shows training curves for run TF7.

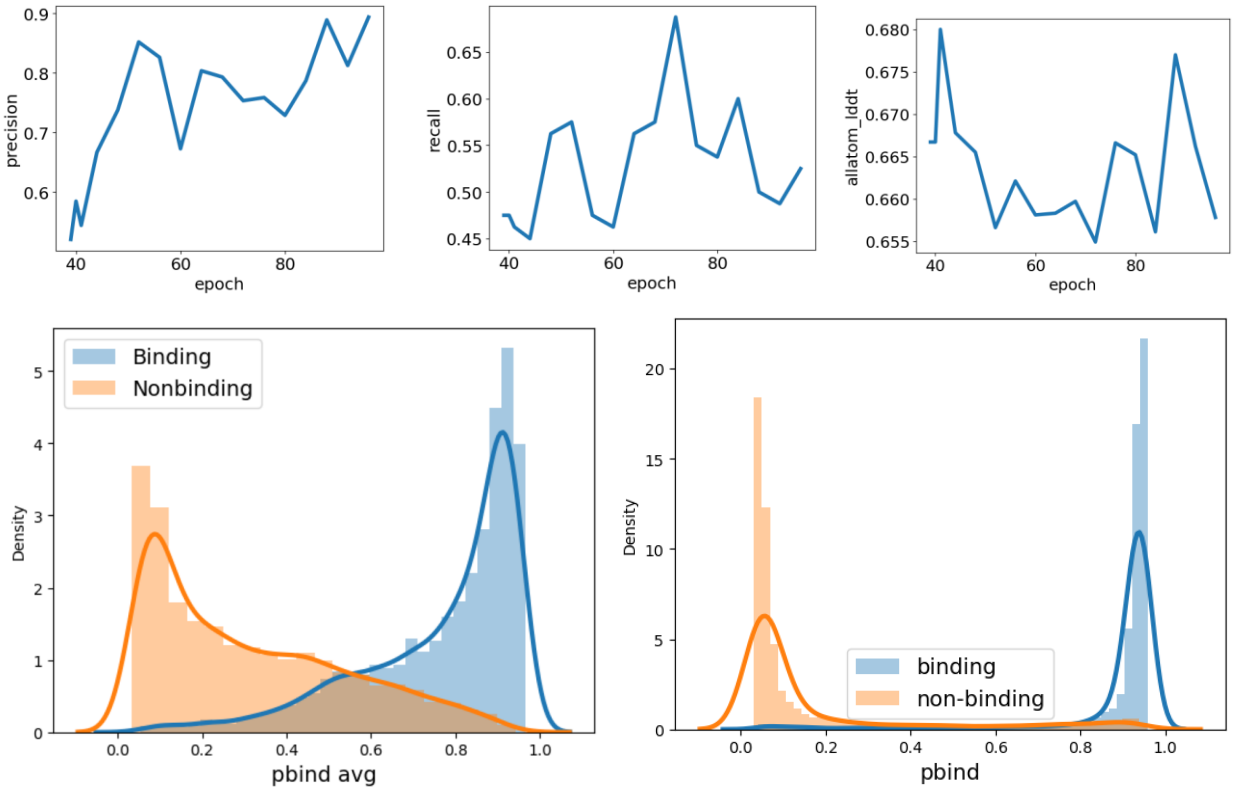


Figure 4.6 Training curves for run TF4 (top) and comparison to TF3 on a test set (bottom). (Top left) Model TF4 achieved precision as high as 90% on the TF validation set. **(Top center)** TF4 maintained recall around 50-55%, in line with TF3's performance. **(Top right)** TF4 did not show improvement in accuracy of predicting protein-NA complexes from the PDB, but showed significant variability due to low sample number. **(Bottom left)** TF3 was able to distinguish binding and non-binding sequences for proteins in the TF set (both training and validation). **(Bottom right)** On the same set of proteins and DNA sequences, TF4 could distinguish the binding and non-binding sequences with near-perfect accuracy; if plotted as an ROC curve, TF4 has an AUC of 0.99 on this task. This astonishingly good performance should be attributed to the fact that the binding complexes in the test were almost all provided as structures in the TF_distil training set for this run.

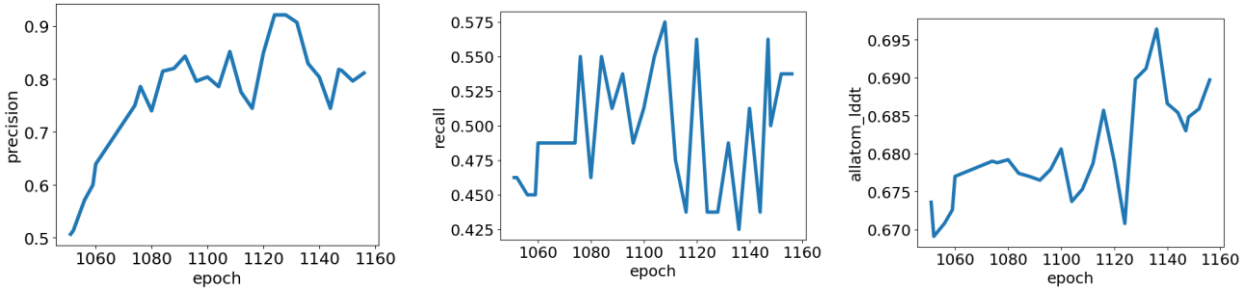


Figure 4.7 Training curves for run TF7. (Left) Run TF7 achieved precision in line with TF4 (peaking above 90%), without the data leak issue from TF_distil. **(Middle)** TF7 maintained recall in line with previous models, at about 55%. **(Right)** TF7 showed significant improvement in protein-NA structure prediction accuracy, peaking very close to 0.7 (compared to 0.65 for RFNA).

TF7 not only learned the binding task quite well, with ROC-AUC of 0.9 or higher on training families and 0.7 or higher on validation families, but also had the best accuracy at protein-DNA interfaces of any model to date. Figure 4.8 shows the difference in performance between RF2aa and its fine-tuned TF7 variant on just protein-DNA interfaces from the PDB validation set. The success rate of near-correct structures (with interface lDDT greater than 0.3) jumped from about 40% to about 60% (note that the distribution is bimodal, so comparing measures of center like median and mean is less meaningful).

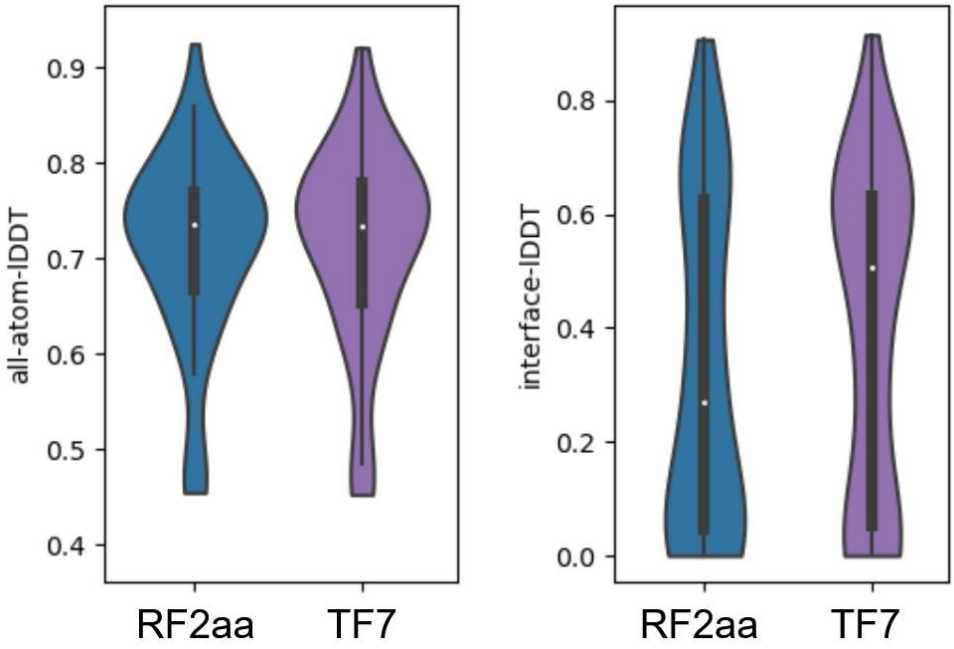


Figure 4.8 Comparison of RF2aa prior to fine-tuning to TF7. Left shows the distribution of mean all-atom-lDDT scores for the structures predicted; right shows the same averaged over only interface residues. The inner black plot for each is a box-and-whisker plot showing min, Q1, Q3, and max values, with median marked by a white dot. Note that interface-lDDT is a volatile metric; predictions which are clearly incorrect will generally have a score of 0, while any score above 0.3 will be generally a correct structure and the spread above that mostly reflects the accuracy of sidechain placements.

Author's note: TF7 was the first (and unfortunately last) time this project felt like a remarkable success. I presented my results as a talk at RosettaCon 2023 with a fairly enthusiastic response from interested parties. To my knowledge, this was the first time it was shown that the accuracy of a structure prediction network on a structure prediction task could be significantly improved by incorporating non-structural data. (Although, in retrospect it seems likely that the largest improvement came from structure distillation, which is a common technique in the field).

Training run TF8

Following the observation that TF7 could be used to filter designed DBPs, but only if DNA target sequences were trimmed to about 8 basepairs (see section 4.3.5), it was apparent that the model still had significant limitations. It seemed clear that this particular limitation arose from the abundance of specifically 8mer DNA sequences, derived from PBM data, in both the TF and TF_distil datasets.

In an attempt to counteract this issue, I decided to introduce significant amounts of sequence padding to the TF training data. Each time a positive or negative sequence was loaded in training, I padded it with random additional sequences from the negative list out to a maximum total length of 40 basepairs.

Author's note: Because everyone always loves to bring up this possibility, I included a protocol to ensure that additional positive binding sites would not be introduced by mistake. My initial version of it, which compared lists of lists, slowed down run-time dramatically, requiring as many as 13 CPU threads loading data to keep up with a single GPU running the model. I implemented a clever, much faster version which converted the DNA sequences to numbers (0 for A, 1 for C, 2 for T, 3 for G) and identified sequence matches by just subtracting whole matrices at once. I am especially proud of the trick to find many DNA sequences' complements at once by adding 2 and taking mod-4 of the result. My CPU can compute this for a thousand 100-bp sequences in 1 millisecond.

Contrary to my expectations, the addition of significant padding did not, in fact, help the network learn to properly identify binding sites within longer DNA sequences. Instead, the network, unable to predict the structures of the new TF examples it was seeing, learned to memorize just the sequences of the positive DNAs. We can tell this because the network's recall of positive sequences in the validation set drops rapidly during training, as shown in Figure 4.9. A checkpoint of TF8 mid-way through training (around epoch 1100) is not terrible, but is still notably worse than TF7 on all three metrics.

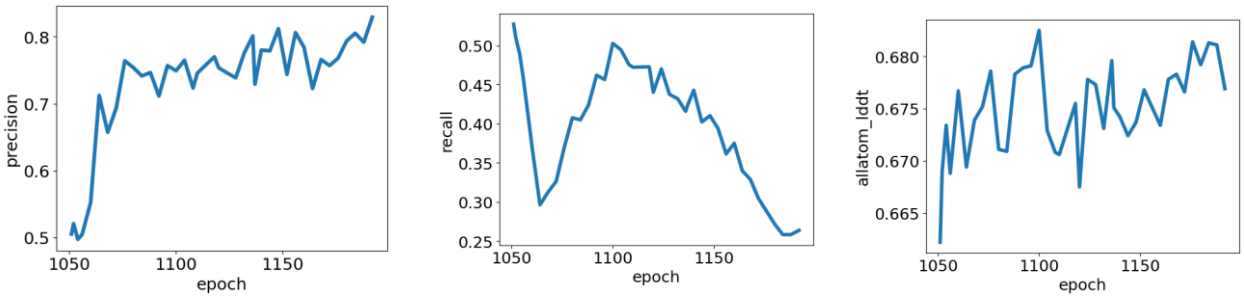


Figure 4.9 Training curves for run TF8. (Left) Run TF8 rapidly achieved precision close to 75%, followed by a very slow increase to about 80%. **(Middle)** TF8 rapidly loses recall, dropping down to 30%. It briefly recovers mid-way through training, before dropping again. **(Right)** TF8 shows little change in structure prediction accuracy, with perhaps a slight upward trend.

Training run TF9

Following the failure of TF8, we decided we needed to take a step back and make larger improvements to the network to do better than TF7.

First, we tried introducing an attention layer to the binding head, hoping that it would help the network learn to focus on only the residues that are actually important for binding, such as the ones at the interface in the predicted structure.

Next, we overhauled our processing of protein sequences from all sources. Prior to this, we had been using protein sequences parsed into domains by RF2 prediction followed by a geometric parsing script. I noticed that many of these parsed domains were of very poor quality, often including only a single helix or strand from a much larger domain. It seemed obvious that the network could not learn structural determinants of DNA-binding specificity with such poor protein sequences, so I undertook the effort of manually parsing representatives into the correct DNA binding domains (as described above). Along the way, we also added even more data from additional sources.

Trying to prevent overfitting, I added parameter dropout to the binding prediction head, reduced the amount of padding used for TF examples, lowered the weight of the binder loss, reduced the sampling frequency of all TF data, and tried using a modified form of the BCE loss that was more concave.

The training run incorporating these changes, called TF9, showed no significant difference from TF8, as shown in Figure 4.10.

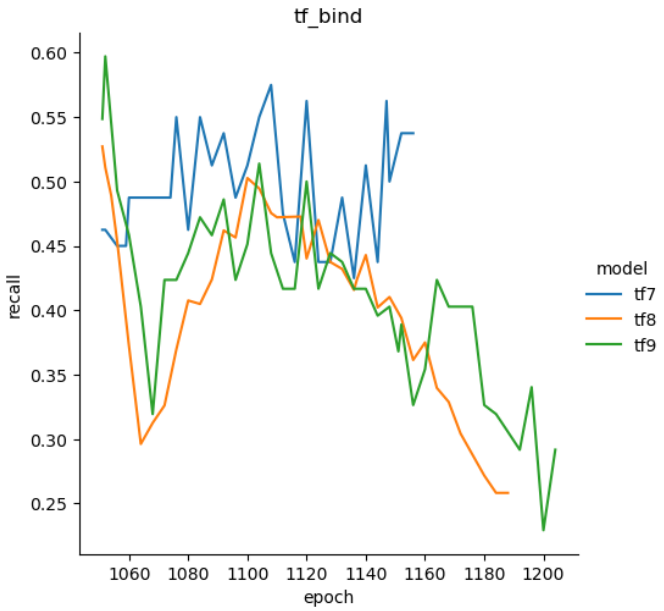


Figure 4.10 Comparison of training runs TF7, TF8, and TF9. These three runs all started from the same checkpoint, but had significant differences as described above. However, TF8 and TF9 followed nearly identical training trajectories, with the same drop in recall.

Training run TF9-C

At this point, I had determined that the only way to overcome the limitations of TF7 would be to either (A) significantly improve the underlying structure prediction model or (B) drastically change our approach and reprocess the entire dataset. Since our main challenge was prediction of the correct binding site within longer DNA chains, I felt that approach A was more likely to make a difference.

One idea we'd had for a long time to improve RFNA and now RF2aa was to change the placement of the DNA frames. For both proteins and nucleic acids, RFNA models the full 3D structure by predicting the 6 translational and rotational degrees of freedom of a backbone frame, plus additional degrees of freedom for sidechain torsions. Originally, RFNA used a frame for DNA (and RNA) that was centered at the phosphate group on the DNA backbone, because this is the most analogous to the backbone frame used for proteins. However, nucleic acids have far more torsional degrees of freedom in its backbone than amino acids do, and the placement of the nucleobase is both the most important and generally the best-resolved part of the DNA structure. Thus, we decided to move the frame from the phosphate group to the C1' atom on the sugar ring, as shown in Figure 4.11. Now, rather than needing to accurately model as many as 10 consecutive torsion angles to place the sidechain, the model could place the entire nucleobase ring based on a single torsion from the predicted frame.

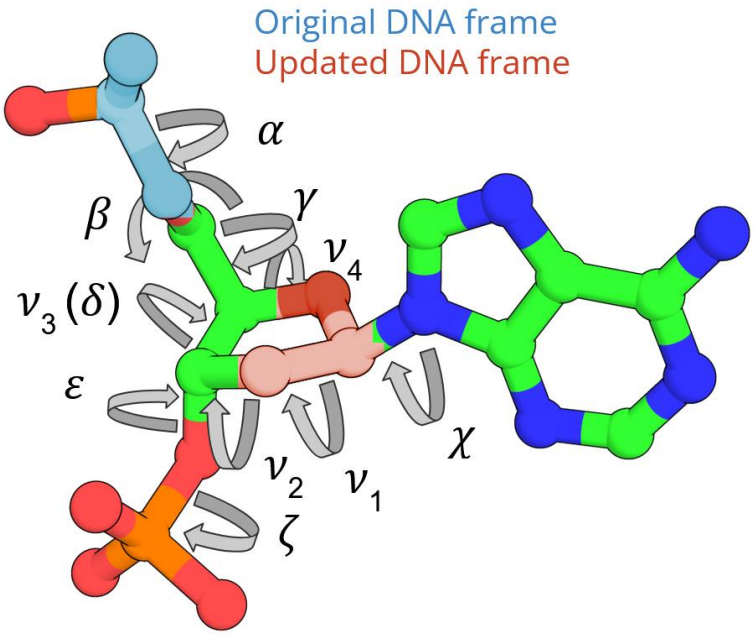


Figure 4.11 Diagram of the torsion angles of a nucleotide. The original and updated DNA frames are highlighted in blue and red, respectively. The original frame is OP1-P-O5', while the updated frame is C2'-C1'-O4'.

Author's note: The list of bond lengths, angles, and transformations used to actually generate a 3D structure from a set of translations, rotations, and torsions is hard-coded into RosettaFold individually for each token. Making this work for nucleic acids was one of the hardest tasks of the original RFNA project, alongside creating the training dataset. As such, changing the frames was a highly laborious and largely manual task undertaken not by me but by my advisor Frank DiMaio, and I thank him for his effort in this.

For the first time since the official RFNA release, we found a way to improve our structure prediction for nucleic acids themselves! Run TF9-C showed improvements across the board in modeling DNA (all-atom lDDT from 0.78 to 0.81), RNA (0.70 to 0.72) and protein-NA complexes (0.68 to 0.70).

Midway through training TF9-C, I saw that these improvements were limited only to increasing the intra-chain accuracies of DNA and RNA, while protein-DNA interfaces were not improving. I decided to implement an extra loss term, the FAPE loss scored only for residues forming interfaces, and apply a high weight to it (5-times higher than general FAPE). This resulted in a significant increase in interface accuracy, from mean all-atom lDDT of 0.20 to 0.24, but also dramatic decreases in accuracy of all individual chains.

As a compromised model, I rolled back training to before this last change and changed the interface FAPE to be only 2-times upweighted. The resulting model, called TF9-D, was able to achieve the same improvement in interface accuracy while maintaining overall accuracy in line with previous models. The comparison between TF9-C and TF9-D is shown in Figure 4.12.

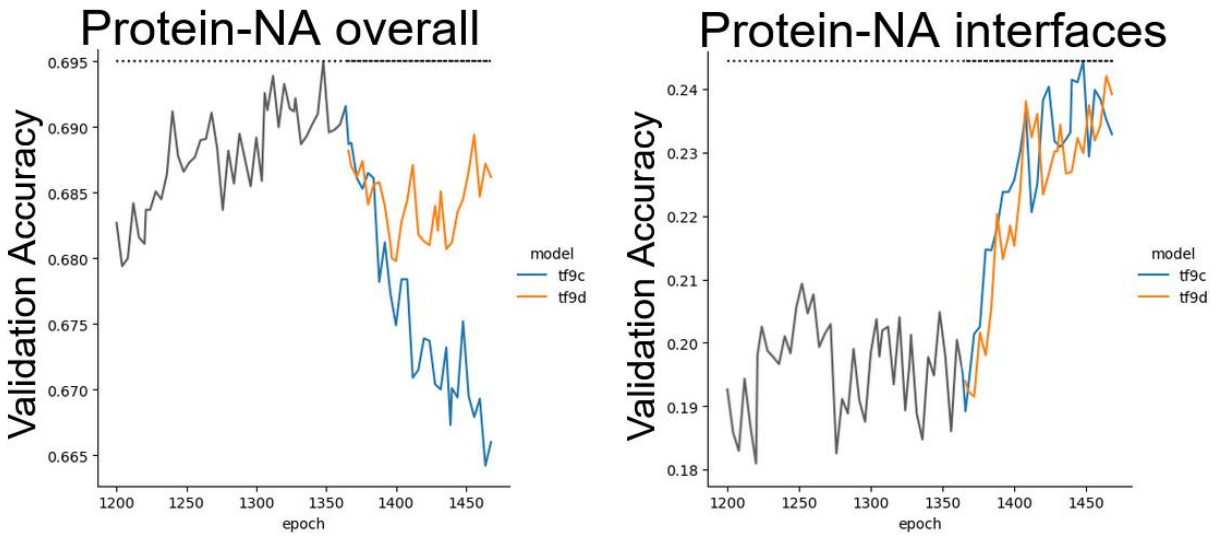


Figure 4.12 Comparison of models TF9-C and TF9-D on the protein-NA validation set. TF9-C had significantly reduced overall accuracy but increased interface accuracy (blue lines). Meanwhile, TF9-D achieved the same improvement in interface accuracy without any decrease in overall accuracy compared to the point where they branched (orange lines). The gray line shows the trajectory of training from which both 9-C and 9-D diverged.

4.3.5 Application: DBP design selection

As was one of its original purposes, I and others have used the fine-tuned version RFNA as a filter for designed DBPs.

I first tried this on a retrospective set of around 500 designed DBPs, all of which passed the filters described in section 2.2.4, and all of which were tested experimentally. Of these, 11 were found classified as specific binders and 26 as nonspecific binders. Note that this dataset requires distinguishing binding for a large number of proteins with a smaller set of DNA sequences, unlike the training and test tasks which involve the reverse problem.

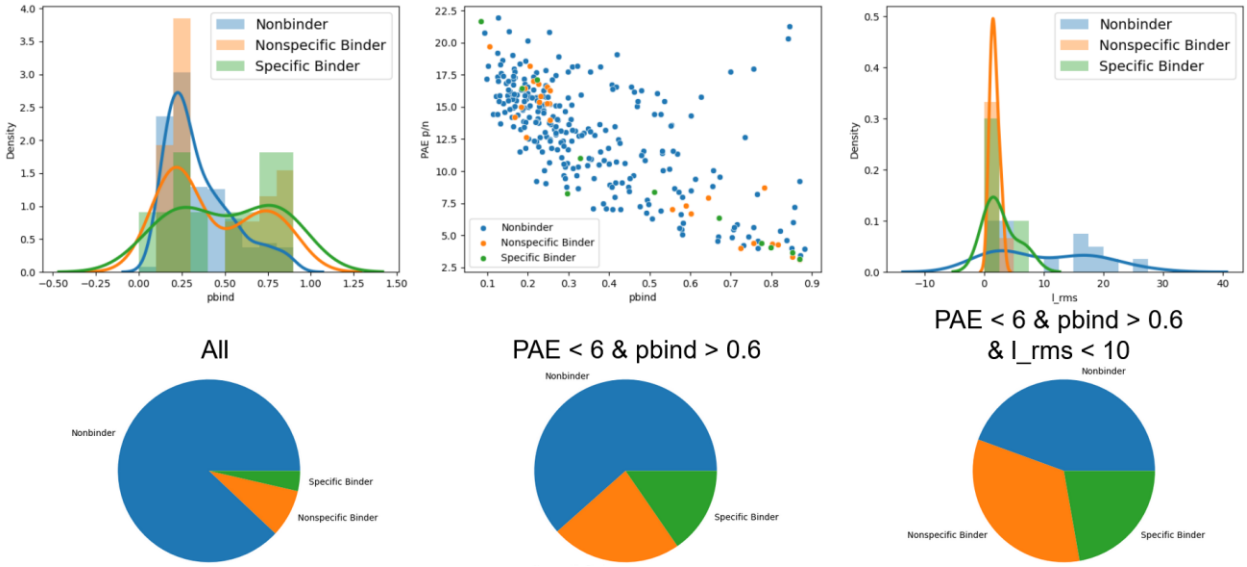


Figure 4.13 Retrospective analysis of designed DBPs using TF7. All plots use blue color for nonbinders, orange for nonspecific binders, and green for specific binders. **(upper left)** Distributions of DBP predictions' p_bind scores, overlayed for the three groups. The nonbinder group is mostly in the p_bind < 0.5 population, while both binder groups are evenly split above and below 0.5. **(upper center)** Scatterplot of predicted designs with interface PAE on the y-axis and p_bind on the x-axis. The cluster in the lower right of the plot, which would pass filters on both metrics, is enriched for binders over nonbinders. **(upper right)** Distributions of DBP predictions' I_rms to corresponding design models. The nonbinder population is spread across a wide range of I_rms values, while all binders have I_rms < 10. **(lower left)** Pie chart of the initial design set, which has already passed all prior energetic, geometric, and AF2 filters. **(lower center)** Pie chart of the design set after filtering on confidence metrics PAE and p_bind. The binders make up a much larger portion of the population. **(lower right)** Pie chart of the design set after filtering further on agreement between prediction and design models. The binders now make up the majority of the population, despite having started as a small minority.

I trimmed the DNA sequence of each design to the 8 contiguous basepairs closest to the protein, then predicted each complex using the single protein sequence plus the 8-basepair DNA sequence. I evaluated the performance of the model's confidence scores (pLDDT, interface PAE, and p_bind) as well as agreement with the design model (I_rms:

RMSD of the protein after alignment of the DNA and i_rms: RMSD of residues forming the interface in the design model after alignment on the same set).

Based on my analysis, I found that p_bind, interface PAE, and l_rms had similar overall performance as classifiers for the set (ROC-AUC of ~0.6), but l_rms had the highest early enrichment for binders over non-binders. This is reasonable, as l_rms most directly considers whether the prediction agrees with the design model. However, using l_rms limits this approach to use cases where a design model exists, which would not be the case for finding binding sequences for native DBPs.

Using a combination of filters (interface PAE < 6, p_bind > 0.6, l_rms < 10), I was able to enrich the binding population from about 7% to about 60%. Figure 4.13 shows the results of this analysis. Assuming the designs sampled here retrospectively would be representative of future designs, this result implied that filtering with TF7 could increase future design campaign success rates by 5- to 10-fold.

However, the later design campaigns by my peers in the Baker lab, which *have* used either TF7 or TF9-D for a filtering step, have not seen the promised increase in success rate. We believe the discrepancy is due to an inherent bias in the fine-tuned RFNA models for particular folds and docking orientations of DBPs which are present in the training set; designs similar to these happened to be present in the small set of successful designs.

4.3.6 Application: predicting DBP specificity profiles

Author's note: Along with the DBP design filter application, I see this as a necessary task for a general DBP specificity predictor to be able to perform. I did not develop an inference protocol to test whether my models could actually do this during my training and development process, because I was focused only on optimizing the training performance itself. In hindsight, creating a test for this essential task that could be run on each iteration of the model could have helped guide development and identify whether a model was good enough to ship. When I finally realized this, I worked with a rotation student, Tabitha Tcheau, who implemented and tested the protocol I proposed.

In theory, any model which can quantitatively rank the possible DNA sequences (or give them binary classifications) could be extended to construct full specificity profiles in the form of position weight matrices (PWMs). The naïve approach to do this with my fine-tuned RosettaFold model would be to predict a DBP of interest together with each possible DNA sequence of the desired length. However, this would be computationally intractable for studying any significant number of DBPs – you would need over 32,000 separate structure prediction runs for a single protein limited to binding 8 basepairs of DNA.

To improve the efficiency, I proposed two possible ways to extract similar information from the model with few prediction runs: (1) develop a protocol using Monte Carlo simulated

annealing to identify and rank the sequences most preferred by the model, then use those to construct a profile; or (2) run structure prediction of the protein alongside a smaller set of longer DNA sequences which contain all possible short motifs, such as a set of DeBruijn sequences[131], then use the sequences bound in any high-confidence predictions to construct a profile.

As expected based on prior results, method 2 proved intractable due to my model's general difficulties with predicting proteins bound to long DNAs. Additionally, we found that the model had a bias to place the protein closer to the center of the DNA (almost always with 20 basepairs of the center point), despite our efforts to reduce this effect with padding.

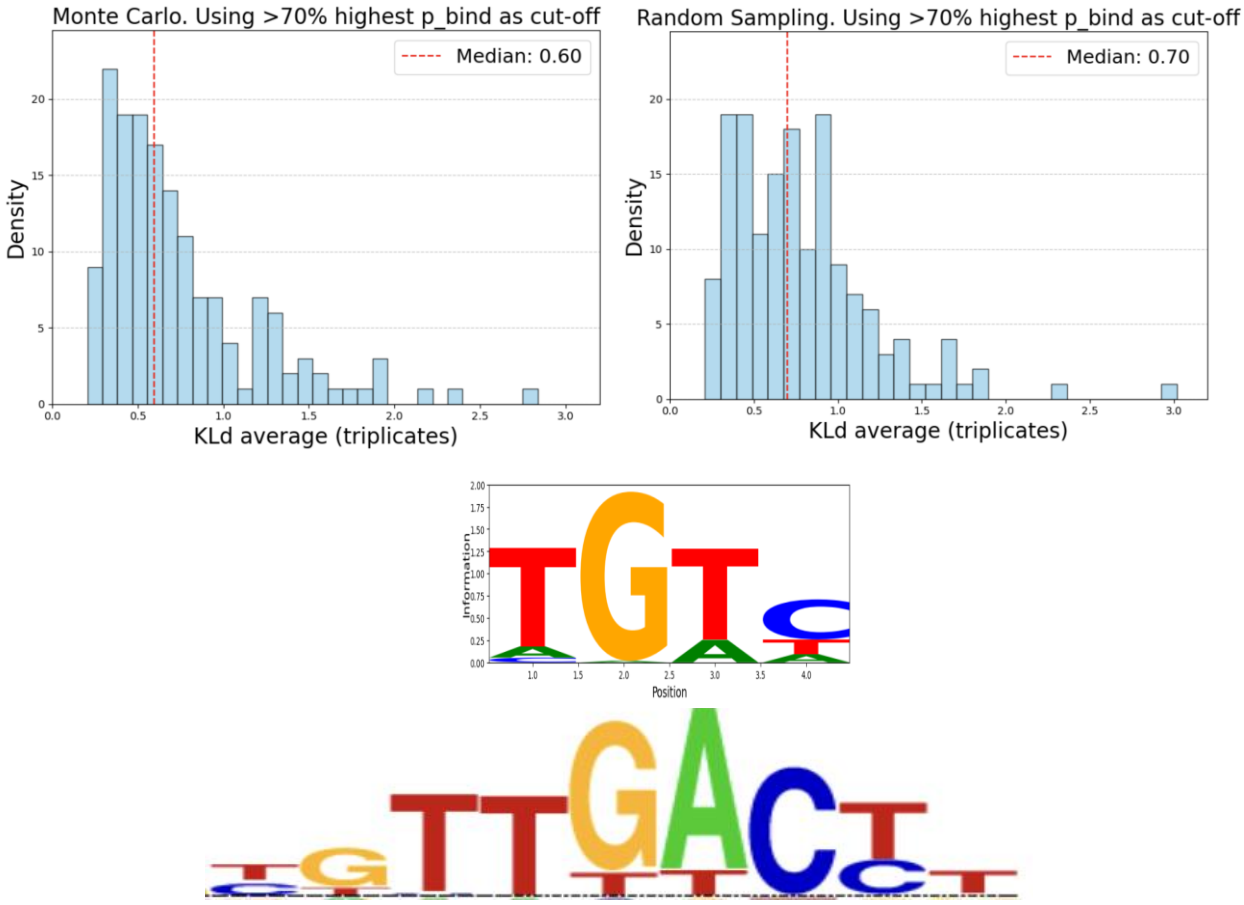


Figure 4.14 Results of the MCMC simulated annealing protocol for DBP profile prediction. **(Top)** Histogram of KLD scores for the predicted profiles compared to the aligned portions of the experimentally-determined motifs, using the MCMC protocol (left) or an equal number of samples chosen randomly (right). The MCMC protocol significantly outperforms random sampling, as evaluated by the median KLDs and the shapes of the distributions. **(Bottom)** Example of a KLD of 0.6, which is the median result from the MCMC protocol. In this example, which is the protein WRKY43, the predicted motif is both visually and quantitatively similar to the aligned portion of the observed motif, although its coverage is far from complete.

To implement method 1, which is equivalent to hallucinating a DNA sequence in the presence of a protein using the structure prediction output as a guide, we wrote an inference script that runs the model iteratively on a fixed protein sequence and a variable DNA sequence. We mutated the DNA sequence following a simple metropolis criterion Monte Carlo (MCMC) algorithm. In each step, we chose a random mutation from the previous sequence, predicted the complex, and used -1 * p_bind as a makeshift energy term for accepting or rejecting mutations. The temperature factor in the metropolis criterion starts very high to allow a broad search of sequence space, then is reduced to slowly approach a local minimum (i.e. simulated annealing; the specific protocol used an initial temperature value of 1 and a linear decay rate of 0.004 per step). We selected all DNA sequences sampled with p_bind scores greater than 80% of the maximum p_bind score observed and created a position probability matrix (PPM) from these sequences using MEME[128]. The predicted PPMs were compared to the experimentally-determined PPMs in cis-BP based on Kullback-Leibler divergence (KLD).

The parameters of this protocol were optimized for a single protein in the validation set, then the protocol was tested on all proteins from the cis-BP-derived TF validation set. Figure 4.14 summarizes the results of the final version of the MCMC protocol and shows an example median-scoring PPM compared to the experimental motif. We found that annealing trajectories with 500 steps could find global minima and generate profiles nearly as well as brute-force sampling of all sequences. The 500-step search is more than 100 times more efficient than exhaustive sampling and provides better performance than a random sample of 500 sequences. This indicates that the MCMC protocol is sufficient for sampling from sequence space to find the sequences most preferred by the model. The final motifs are reasonably accurate, but probably not better than could be achieved with other methods; this protocol is ultimately limited by the quantitative accuracy of the model's p_bind prediction.

4.4 Discussion of chapter 4

4.4.1 Impact and limitations of the fine-tuned RosettaFold

Author's note: shortly after I finished training TF9-D, DeepMind's shared results of their AF3 model, which outperformed RFNA as published and my fine-tuned models. At that point, we largely abandoned fine-tuning work in favor of trying to fundamentally change RosettaFold to be in line with AF3's performance. Ultimately, our best performance came in the form of a full replication of AF3's architecture, with slightly different training data. Some of the biggest differences are in the protein-DNA complex portion of the data, thanks to my work in curating a large, high-quality distillation set. Regardless, I was quite discouraged and went a long time without any new results. Finally, within a few weeks of writing this, three different methods that depend on my work in fine-tuning RFNA have been written as manuscripts and soon preprints: RF-polydiff, which Andrew Favor trained

starting from model TF-9D itself to generate DNA and RNA structures; RF-3, which incorporates the TF_distil dataset in its training as well as a DNA padding module I developed; and NA-MPNN, a DNA and RNA sequence design tool which was partially trained and evaluated on the TF and TF_distil datasets I curated. Moreover, my colleagues recently obtained a crystal structure for a protein fusion of two of our designed DBPs, created with RF-polydiff and selected by prediction with my fine-tuned RFNA. My model's predicted structure closely matches the crystal structure, while both AF3 and its competitor Boltz fail to predict the correct complex.

In short, fine-tuning a protein structure prediction model for predicting details of protein function will always be limited by three things: the performance of the underlying prediction model, the quality of the protein function data, and the strength of the connection between these two components.

In this work, I started with a state-of-the-art protein-DNA complex structure prediction model, but the state of the art advanced rapidly during the time I spent optimizing.

I believe that I have curated the largest and highest-quality combination of DNA-binding domain sequences and corresponding DNA-binding activity data, but ultimately the dataset consists of just over 3000 unique proteins total. This represents a vast amount of experimental work done to characterize thousands of proteins and their binding to millions of total DNA sequences, but it is nowhere near the scale we would need to truly create a general DBP function predictor.

The connection between the data and the model is perhaps the weakest part of this work. Simply put, treating DNA-binding specificity as a binary problem is over-simplified. I relied on this assumption because there is no other interpretation that can clearly be applied to data sources with very different formats and underlying experiment. Regardless, the assumption itself is flimsy at best and the connection to structure prediction using a BCE loss is simply not strong enough to drive learning of difficult tasks like finding a binding site.

Despite these limitations, my fine-tuned RosettaFold model and the dataset it was trained on have already made significant impacts on projects involving nucleic acids in my lab. My distillation set has been used in at least four other methods besides the one described here, all of which will eventually be published (and one, RF-polydiff, was trained directly from one of my models; its publication will include the weights and code for using TF9-D).

I demonstrated that functional data could help guide structure prediction models and vice versa, which has inspired a number of other projects fine-tuning RosettaFold to predict various protein functions.

My fine-tuned models themselves have continue to be used in our lab's DBP design efforts, alongside further improvements to backbone generation and sequence design.

It has not been done yet, but I believe my model or AF3-like models trained similarly could be used to generate a distillation set of protein-DNA complexes orders of magnitude larger than the one presented here. Assuming that a protocol like the MCMC I described is able to find plausible DNA sequences for a given DBP, and that the underlying structure prediction model's confidence terms can identify when a high-quality structure is produced, then all we would need as an input is the sequence of a DNA-binding domain. A quick search of UniProt [[[[]]]] finds almost 2 million protein sequences with annotations for DNA-binding domains. If we can predict cognate DNAs and complex structures for even a tenth of these, it would expand my distillation set by 100-fold.

4.4.2 Conclusion: have we achieved Aim 3?

Based on the results I have, I find it hard to conclusively answer this question one way or the other. Yes, I trained a model that can predict both structure and DNA-binding specificity of DBPs. My best models can distinguish between binding and non-binding DNAs for proteins in families dissimilar to those in the training set, with at least moderate accuracy. However, I have not yet demonstrated their ability to simultaneously predict both the structure and DNA-binding profile of orphan or unstudied DBPs.

The models I trained are clearly and demonstrably useful in some cases, but they have never felt quite *useful enough*. Indeed, even a version of RosettaFold with perfect binding/non-binding discrimination for DBPs would still be a terribly inefficient tool for studying DBPs found in nature. And then still, binding discrimination is not the same thing as true function prediction – to truly say we understand DBPs fully, I think we ought to be able to predict absolute binding affinities, kinetics, and conformational dynamics for any protein-DNA pair. Further beyond that would be to predict the effects of solvent, genomic context, and other proteins in the environment on said properties.

In short, it will be a long time before we can truly achieve Aim 3 in its entirety, but this may be a step in the right direction.

Author's note: Truly, this project continues to be a work in progress. For the last 18 months, training progress on the DNA binding task stalled first due to a focus on improving structure prediction, then while I waited for our team to replicate AF3's performance, then due to my impending need to graduate and find a job. The last of these was made more difficult and more urgent when a new federal government decided to slash funding for science and education. For a long time, this project has felt nearly-complete. In many ways, model TF7 was as successful as it needed to be. We probably should have done a bit more inference and evaluation work with that model and published it as it was. Now, the total re-implementation of this approach to an AF3-like model is mostly complete, but not to the point of having new results. My advisor is making plans to follow up on this work with one of our collaborators and hopefully publish it soon.

References

1. Takeda, Y., Ohlendorf, D. H., Anderson, W. F. & Matthews, B. W. DNA-binding proteins. *Science* **221**, 1020–1026 (1983).

2. Lemon, B. & Tjian, R. Orchestrated response: a symphony of transcription factors for gene control. Genes & development 14, 2551–2569 (2000).

3. Badis, G. *et al.* Diversity and complexity in DNA recognition by transcription factors. *Science* **324**, 1720–1723 (2009).

4. Zaret, K. S. & Carroll, J. S. Pioneer transcription factors: establishing competence for gene expression. *Genes & development* **25**, 2227–2241 (2011).

5. Lambert, S. A. *et al.* The human transcription factors. *Cell* **172**, 650–665 (2018).

6. Strader, L., Weijers, D. & Wagner, D. Plant transcription factors—being in the right place with the right company. *Current opinion in plant biology* **65**, 102136 (2022).

7. Slattery, M. *et al.* Absence of a simple code: how transcription factors read the genome. *Trends in biochemical sciences* **39**, 381–399 (2014).

8. Siggers, T. & Gordân, R. Protein–DNA binding: complexities and multi-protein codes. *Nucleic acids research* **42**, 2099–2111 (2014).

9. Garvie, C. W. & Wolberger, C. Recognition of specific DNA sequences. *Molecular cell* **8**, 937–946 (2001).

10. Todeschini, A.-L., Georges, A. & Veitia, R. A. Transcription factors: specific DNA binding and specific gene regulation. *Trends in genetics* **30**, 211–219 (2014).

11. Sera, T. & Uranga, C. Rational design of artificial zinc-finger proteins using a nondegenerate recognition code table. *Biochemistry* **41**, 7074–7081 (2002).

12. Richter, A., Streubel, J. & Boch, J. TAL effector DNA-binding principles and specificity. in *TALENs: Methods and Protocols* 9–25 (Springer, 2016).

13. Jayaram, B. & Jain, T. The role of water in protein-DNA recognition. *Annu. Rev. Biophys. Biomol. Struct.* **33**, 343–361 (2004).

14. Starr, B. D., Hoopes, B. C. & Hawley, D. K. DNA bending is an important component of site-specific recognition by the TATA binding protein. *Journal of molecular biology* **250**, 434–446 (1995).

15. Wang, D., Ulyanov, N. B. & Zhurkin, V. B. Sequence-dependent Kink-and-Slide deformations of nucleosomal DNA facilitated by histone arginines bound in the

minor groove. *Journal of Biomolecular Structure and Dynamics* **27**, 843–859 (2010).

16. Murphy, F. v & Churchill, M. E. A. Nonsequence-specific DNA recognition: a structural perspective. *Structure* **8**, R83–R89 (2000).

17. Boch, J. *et al.* Breaking the code of DNA binding specificity of TAL-type III effectors. *Science* **326**, 1509–1512 (2009).

18. Rohs, R. *et al.* Origins of specificity in protein-DNA recognition. *Annual review of biochemistry* **79**, 233–269 (2010).

19. Wilson, K. A. *et al.* Landscape of π–π and sugar–π contacts in DNA–protein interactions. *Journal of Biomolecular Structure and Dynamics* **34**, 184–200 (2016).

20. Wilson, K. A. *et al.* Landscape of π–π and sugar–π contacts in DNA–protein interactions. *Journal of Biomolecular Structure and Dynamics* **34**, 184–200 (2016).

21. Halford, S. E. An end to 40 years of mistakes in DNA–protein association kinetics? *Biochemical Society Transactions* **37**, 343–348 (2009).

22. Hu, L., Li, Y., Wang, J., Zhao, Y. & Wang, Y. Controlling CRISPR-Cas9 by guide RNA engineering. *Wiley Interdisciplinary Reviews: RNA* **14**, e1731 (2023).

23. Zhang, X.-H., Tee, L. Y., Wang, X.-G., Huang, Q.-S. & Yang, S.-H. Off-target effects in CRISPR/Cas9-mediated genome engineering. *Molecular therapy Nucleic acids* **4**, (2015).

24. Schneider, T. D. & Stephens, R. M. Sequence logos: a new way to display consensus sequences. *Nucleic acids research* **18**, 6097–6100 (1990).

25. Luscombe, N. M., Austin, S. E., Berman, H. M. & Thornton, J. M. An overview of the structures of protein-DNA complexes. *Genome biology* **1**, reviews001-1 (2000).

26. Villegas Kcam, M. C., Tsong, A. J. & Chappell, J. Rational engineering of a modular bacterial CRISPR–Cas activation platform with expanded target range. *Nucleic Acids Research* **49**, 4793–4802 (2021).

27. Wilken, M. S. *et al.* Quantitative dialing of gene expression via precision targeting of KRAB repressor. (2020).

28. Mitra, R. *et al.* Geometric deep learning of protein–DNA binding specificity. *Nature Methods* **21**, 1674–1683 (2024).

29. Cao, L. *et al.* Design of protein-binding proteins from the target structure alone. *Nature* **605**, 551–560 (2022).

30. Ashworth, J. *et al.* Computational redesign of endonuclease DNA binding and cleavage specificity. *Nature* **441**, 656–659 (2006).

31. Ashworth, J. *et al.* Computational reprogramming of homing endonuclease specificity at multiple adjacent base pairs. *Nucleic acids research* **38**, 5601–5608 (2010).

32. Thyme, S. B. *et al.* Exploitation of binding energy for catalysis and design. *Nature* **461**, 1300–1304 (2009).

33. Ulge, U. Y., Baker, D. A. & Monnat Jr, R. J. Comprehensive computational design of mCreI homing endonuclease cleavage specificity for genome engineering. *Nucleic acids research* **39**, 4330–4339 (2011).

34. Liu, X., Meger, A. T., Gillis, T. & Raman, S. Computation-guided redesign of promoter specificity of a bacterial RNA polymerase. *Biorxiv* 2011–2022 (2022).

35. Milk, L., Daber, R. & Lewis, M. Functional rules for lac repressor–operator associations and implications for protein–DNA interactions. *Protein Science* **19**, 1162–1172 (2010).

36. Huang, P.-S., Boyken, S. E. & Baker, D. The coming of age of de novo protein design. *Nature* **537**, 320–327 (2016).

37. Klug, A. The discovery of zinc fingers and their applications in gene regulation and genome manipulation. *Annual review of biochemistry* **79**, 213–231 (2010).

38. Joung, J. K. & Sander, J. D. TALENs: a widely applicable technology for targeted genome editing. *Nature reviews Molecular cell biology* **14**, 49–55 (2013).

39. Wang, J. Y. & Doudna, J. A. CRISPR technology: A decade of genome editing is only the beginning. *Science* **379**, eadd8643 (2023).

40. Seeman, N. C., Rosenberg, J. M. & Rich, A. Sequence-specific recognition of double helical nucleic acids by proteins. *Proceedings of the National Academy of Sciences* **73**, 804–808 (1976).

41. Coulocheri, S. A., Pigis, D. G., Papavassiliou, K. A. & Papavassiliou, A. G. Hydrogen bonds in protein–DNA complexes: Where geometry meets plasticity. *Biochimie* **89**, 1291–1303 (2007).

42. Harrison, S. C. & Aggarwal, A. K. DNA recognition by proteins with the helix-turn-helix motif. *Annual review of biochemistry* **59**, 933–969 (1990).

43. Zhang, R. *et al.* Structure of a bacterial quorum-sensing transcription factor complexed with pheromone and DNA. *Nature* **417**, 971–974 (2002).

44. Rodgers, D. W. & Harrison, S. C. The complex between phage 434 repressor DNA-binding domain and operator site OR3: structural differences between consensus and non-consensus half-sites. *Structure* **1**, 227–240 (1993).

45. Zhang, Y. & Skolnick, J. TM-align: a protein structure alignment algorithm based on the TM-score. *Nucleic acids research* **33**, 2302–2309 (2005).

46. Steinegger, M. *et al.* HH-suite3 for fast remote homology detection and deep protein annotation. *BMC bioinformatics* **20**, 473 (2019).

47. Mirdita, M. *et al.* Uniclust databases of clustered and deeply annotated protein sequences and alignments. *Nucleic acids research* **45**, D170–D176 (2017).

48. Eddy, S. R. A new generation of homology search tools based on probabilistic inference. in *Genome Informatics 2009: Genome Informatics Series Vol. 23* 205–211 (World Scientific, 2009).

49. Chen, I.-M. A. *et al.* The IMG/M data management and analysis system v. 7: content updates and new features. *Nucleic acids research* **51**, D723–D732 (2023).

50. Steinegger, M. & Söding, J. MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nature biotechnology* **35**, 1026–1028 (2017).

51. Jumper, J. *et al.* Highly accurate protein structure prediction with AlphaFold. *nature* **596**, 583–589 (2021).

52. Kim, G. B., Gao, Y., Palsson, B. O. & Lee, S. Y. DeepTFactor: A deep learning-based tool for the prediction of transcription factors. *Proceedings of the National Academy of Sciences* **118**, e2021171118 (2021).

53. Altschul, S. F. *et al.* Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic acids research* **25**, 3389–3402 (1997).

54. Wang, Y. *et al.* Analysis of the 2.0 Å crystal structure of the protein− DNA complex of the human PDEF Ets domain bound to the prostate specific antigen regulatory site. *Biochemistry* **44**, 7095–7106 (2005).

55. Mo, Y., Vaessen, B., Johnston, K. & Marmorstein, R. Structures of SAP-1 bound to DNA targets from the E74 and c-fos promoters: insights into DNA sequence discrimination by Ets proteins. *Molecular cell* **2**, 201–212 (1998).

56. Yamasaki, K., Akiba, T., Yamasaki, T. & Harata, K. Structural basis for recognition of the matrix attachment region of DNA by transcription factor SATB1. *Nucleic acids research* **35**, 5073–5084 (2007).

57. Lu, X. & Olson, W. K. 3DNA: a software package for the analysis, rebuilding and visualization of three-dimensional nucleic acid structures. *Nucleic acids research* **31**, 5108–5121 (2003).

58. Luscombe, N. M., Laskowski, R. A. & Thornton, J. M. Amino acid–base interactions: a three-dimensional analysis of protein–DNA interactions at an atomic level. *Nucleic acids research* **29**, 2860–2874 (2001).

59. Dauparas, J. *et al.* Robust deep learning–based protein sequence design using ProteinMPNN. *Science* **378**, 49–56 (2022).

60. Dauparas, J. *et al.* Atomic context-conditioned protein sequence design using LigandMPNN. *Nature Methods* 1–7 (2025).

61. Fleishman, S. J., Khare, S. D., Koga, N. & Baker, D. Restricted sidechain plasticity in the structures of native proteins and complexes. *Protein Science* **20**, 753–757 (2011).

62. Wang, J. *et al.* Scaffolding protein functional sites using deep learning. *Science* **377**, 387–394 (2022).

63. Leaver-Fay, A. *et al.* ROSETTA3: an object-oriented software suite for the simulation and design of macromolecules. in *Methods in enzymology* vol. 487 545–574 (Elsevier, 2011).

64. Kabsch, W. & Sander, C. Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers: Original Research on Biomolecules* **22**, 2577–2637 (1983).

65. Park, H. *et al.* Simultaneous optimization of biomolecular energy functions on features from small molecules and macromolecules. *Journal of chemical theory and computation* **12**, 6201–6212 (2016).

66. Yanover, C. & Bradley, P. Extensive protein and DNA backbone sampling improves structure-based specificity prediction for C2H2 zinc fingers. *Nucleic acids research* **39**, 4564–4576 (2011).

67. Castro-Mondragon, J. A. *et al.* JASPAR 2022: the 9th release of the open-access database of transcription factor binding profiles. *Nucleic acids research* **50**, D165–D173 (2022).

68. Gupta, S., Stamatoyannopoulos, J. A., Bailey, T. L. & Noble, W. S. Quantifying similarity between motifs. *Genome biology* **8**, R24 (2007).

69. Berger, M. F. & Bulyk, M. L. Universal protein-binding microarrays for the comprehensive characterization of the DNA-binding specificities of transcription factors. *Nature protocols* **4**, 393–411 (2009).

70. Winter, G. *et al.* DIALS: implementation and evaluation of a new integration package. *Biological Crystallography* **74**, 85–97 (2018).

71. McCoy, A. J. *et al.* Phaser crystallographic software. *Applied Crystallography* **40**, 658–674 (2007).

72. Liebschner, D. *et al.* Macromolecular structure determination using X-rays, neutrons and electrons: recent developments in Phenix. *Biological Crystallography* **75**, 861–877 (2019).

73. Emsley, P., Lohkamp, B., Scott, W. G. & Cowtan, K. Features and development of Coot. *Biological crystallography* **66**, 486–501 (2010).

74. Stanton, B. C. *et al.* Genomic mining of prokaryotic repressors for orthogonal logic gates. *Nature chemical biology* **10**, 99–105 (2014).

75. Watson, J. L. *et al.* De novo design of protein structure and function with RFdiffusion. *Nature* **620**, 1089–1100 (2023).

76. Lutz, R. & Bujard, H. Independent and tight regulation of transcriptional units in Escherichia coli via the LacR/O, the TetR/O and AraC/I1-I2 regulatory elements. *Nucleic acids research* **25**, 1203–1210 (1997).

77. Engler, C., Kandzia, R. & Marillonnet, S. A one pot, one step, precision cloning method with high throughput capability. *PloS one* **3**, e3647 (2008).

78. Chen, W. *et al.* Multiplex genomic recording of enhancer and signal transduction activity in mammalian cells. *Biorxiv* 2011–2021 (2021).

79. Sayers, E. W. *et al.* Database resources of the national center for biotechnology information. *Nucleic acids research* **52**, D33–D43 (2024).

80. Stroud, J. C., Lopez-Rodriguez, C., Rao, A. & Chen, L. Structure of a TonEBP–DNA complex reveals DNA encircled by a transcription factor. *Nature structural biology* **9**, 90–94 (2002).

81. Mitra, R., Cohen, A. S., Sagendorf, J. M., Berman, H. M. & Rohs, R. DNAproDB: an updated database for the automated and interactive analysis of protein–DNA complexes. *Nucleic acids research* **53**, D396–D402 (2025).

82. Baek, M. *et al.* Accurate prediction of protein–nucleic acid complexes using RoseTTAFoldNA. *Nature methods* **21**, 117–121 (2024).

83. Abramson, J. *et al.* Accurate structure prediction of biomolecular interactions with AlphaFold 3. *Nature* **630**, 493–500 (2024).

84. Krishna, R. *et al.* Generalized biomolecular modeling and design with RoseTTAFold All-Atom. *Science* **384**, eadl2528 (2024).

85. Jones, T. S., Oliveira, S. M. D., Myers, C. J., Voigt, C. A. & Densmore, D. Genetic circuit design automation with Cello 2.0. *Nature protocols* **17**, 1097–1113 (2022).

86. Guiblet, W. M. *et al.* Long-read sequencing technology indicates genome-wide effects of non-B DNA on polymerization speed and error rate. *Genome research* **28**, 1767–1778 (2018).

87. Anfinsen, C. B. Principles that govern the folding of protein chains. *Science* **181**, 223–230 (1973).

88. Agarwal, V. & McShan, A. C. The power and pitfalls of AlphaFold2 for structure prediction beyond rigid globular proteins. *Nature chemical biology* **20**, 950–959 (2024).

89. Baek, M. *et al.* Accurate prediction of protein structures and interactions using a three-track neural network. *Science* **373**, 871–876 (2021).

90. Hiranuma, N. *Protein Structure Accuracy Prediction with Deep Learning and its Application to Structure Prediction and Design*. (University of Washington, 2022).

91. Remmert, M., Biegert, A., Hauser, A. & Söding, J. HHblits: lightning-fast iterative protein sequence searching by HMM-HMM alignment. *Nature methods* **9**, 173–175 (2012).

92. Zhang, C., Zhang, Y. & Pyle, A. M. rMSA: a sequence search and alignment algorithm to improve RNA structure modeling. *Journal of Molecular Biology* **435**, 167904 (2023).

93. RNAcentral 2021: secondary structure integration, improved sequence search and new member databases. *Nucleic acids research* **49**, D212–D220 (2021). 1.

94. Kalvari, I. *et al.* Rfam 14: expanded coverage of metagenomic, viral and microRNA families. *Nucleic acids research* **49**, D192–D200 (2021).

95. Humphreys, I. R. *et al.* Computed structures of core eukaryotic protein complexes. *Science* **374**, eabm4805 (2021).

96. Yang, J. *et al.* Improved protein structure prediction using predicted interresidue orientations. *Proceedings of the National Academy of Sciences* **117**, 1496–1503 (2020).

97. Alford, R. F. *et al.* The Rosetta all-atom energy function for macromolecular modeling and design. *Journal of chemical theory and computation* **13**, 3031–3048 (2017).

98. Mariani, V., Biasini, M., Barbato, A. & Schwede, T. lDDT: a local superposition-free score for comparing protein structures and models using distance difference tests. *Bioinformatics* **29**, 2722–2728 (2013).

99. Lensink, M. F. & Wodak, S. J. Docking, scoring, and affinity prediction in CAPRI. *Proteins: Structure, Function, and Bioinformatics* **81**, 2082–2095 (2013).

100. Probst, M. *et al.* Structural insight into DNA-assembled oligochromophores: crystallographic analysis of pyrene-and phenanthrene-modified DNA in complex with BpuJI endonuclease. *Nucleic acids research* **44**, 7079–7089 (2016).

101. Petty, T. J. *et al.* An induced fit mechanism regulates p53 DNA binding kinetics to confer sequence specificity. *The EMBO journal* **30**, 2167–2176 (2011).

102. Gutmann, S. *et al.* Crystal structure of the transfer-RNA domain of transfer-messenger RNA in complex with SmpB. *Nature* **424**, 699–703 (2003).

103. Huang, J. *et al.* Structural basis for protein-RNA recognition in telomerase. *Nature structural & molecular biology* **21**, 507–512 (2014).

104. Li, Z., Zhang, H., Xiao, R., Han, R. & Chang, L. Cryo-EM structure of the RNA-guided ribonuclease Cas12g. *Nature chemical biology* **17**, 387–393 (2021).

105. Hillen, H. S. *et al.* The pentatricopeptide repeat protein Rmd9 recognizes the dodecameric element in the 3′-UTRs of yeast mitochondrial mRNAs. *Proceedings of the National Academy of Sciences* **118**, e2009329118 (2021).

106. Collins, K. J., Yuan, Z. & Kovall, R. A. Structure and function of the CSL-KyoT2 corepressor complex: a negative regulator of Notch signaling. *Structure* **22**, 70–81 (2014).

107. Pourfarjam, Y., Ma, Z., Kurinov, I., Moss, J. & Kim, I.-K. Structural and biochemical analysis of human ADP-ribosyl-acceptor hydrolase 3 reveals the basis of metal selectivity and different roles for the two magnesium ions. *Journal of Biological Chemistry* **296**, (2021).

108. Hellert, J. *et al.* The 3D structure of Kaposi sarcoma herpesvirus LANA C-terminal domain bound to DNA. *Proceedings of the National Academy of Sciences* **112**, 6694–6699 (2015).

109. Grenha, R. *et al.* Structural basis for the activation mechanism of the PlcR virulence regulator by the quorum-sensing signal peptide PapR. *Proceedings of the National Academy of Sciences* **110**, 1047–1052 (2013).

110. Shevtsov, M. B. *et al.* Structural analysis of DNA binding by C. Csp231I, a member of a novel class of RM controller proteins regulating gene expression. *Biological Crystallography* **71**, 398–407 (2015).

111. Šoltysová, M. *et al.* Structural insight into DNA recognition by bacterial transcriptional regulators of the SorC/DeoR family. *Biological Crystallography* **77**, 1411–1424 (2021).

112. Morozov, A. v, Havranek, J. J., Baker, D. & Siggia, E. D. Protein–DNA binding specificity predictions with structural models. *Nucleic acids research* **33**, 5781–5798 (2005).

113. Yanover, C. & Bradley, P. Extensive protein and DNA backbone sampling improves structure-based specificity prediction for C2H2 zinc fingers. *Nucleic acids research* **39**, 4564–4576 (2011).

114. Robertson, T. A. & Varani, G. An all-atom, distance-dependent scoring function for the prediction of protein–DNA interactions from structure. *PROTEINS: Structure, Function, and Bioinformatics* **66**, 359–374 (2007).

115. Lambert, S. A. *et al.* Similarity regression predicts evolution of transcription factor sequence specificity. *Nature genetics* **51**, 981–989 (2019).

116. Chen, C.-Y. *et al.* Predicting target DNA sequences of DNA-binding proteins based on unbound structures. *PloS one* **7**, e30446 (2012).

117. Tognon, M., Kumbara, A., Betti, A., Ruggeri, L. & Giugno, R. Benchmarking transcription factor binding site prediction models: a comparative analysis on synthetic and biological data. *Briefings in Bioinformatics* **26**, bbaf363 (2025).

118. Mou, M., Zhang, Z., Pan, Z. & Zhu, F. Deep Learning for Predicting Biomolecular Binding Sites of Proteins. *Research* **8**, 0615 (2025).

119. Gabdoulline, R., Eckweiler, D., Kel, A. & Stegmaier, P. 3DTF: a web server for predicting transcription factor PWMs using 3D structure-based energy calculations. *Nucleic Acids Research* **40**, W180–W185 (2012).

120. Lin, C.-K. & Chen, C.-Y. PiDNA: predicting protein–DNA interactions with structural models. *Nucleic acids research* **41**, W523–W530 (2013).

121. Tareen, A. & Kinney, J. B. Logomaker: beautiful sequence logos in Python. *Bioinformatics* **36**, 2272–2274 (2020).

122. Hume, M. A., Barrera, L. A., Gisselbrecht, S. S. & Bulyk, M. L. UniPROBE, update 2015: new tools and content for the online database of protein-binding

microarray data on protein–DNA interactions. *Nucleic acids research* **43**, D117–D122 (2015).
123. Matys, V. *et al.* TRANSFAC® and its module TRANSCompel®: transcriptional gene regulation in eukaryotes. *Nucleic acids research* **34**, D108–D110 (2006).
124. Yuan, H., Kshirsagar, M., Zamparo, L., Lu, Y. & Leslie, C. S. BindSpace decodes transcription factor binding signals by large-scale sequence embedding. *Nature methods* **16**, 858–861 (2019).
125. Kshirsagar, M., Yuan, H., Ferres, J. L. & Leslie, C. BindVAE: Dirichlet variational autoencoders for de novo motif discovery from accessible chromatin. *Genome biology* **23**, 174 (2022).
126. Jolma, A. *et al.* DNA-binding specificities of human transcription factors. *Cell* **152**, 327–339 (2013).
127. Nitta, K. R. *et al.* Conservation of transcription factor binding specificities across 600 million years of bilateria evolution. *elife* **4**, e04837 (2015).
128. Bailey, T. L. & Elkan, C. Fitting a mixture model by expectation maximization to discover motifs in bipolymers. (1994).
129. UniProt: the universal protein knowledgebase in 2023. *Nucleic acids research* **51**, D523–D531 (2023).
130. Varadi, M. *et al.* AlphaFold Protein Structure Database: massively expanding the structural coverage of protein-sequence space with high-accuracy models. *Nucleic acids research* **50**, D439–D444 (2022).
131. Orenstein, Y. & Shamir, R. Design of shortest double-stranded DNA sequences covering all k-mers with applications to protein-binding microarrays and synthetic enhancers. *Bioinformatics* **29**, i71–i79 (2013).