UNIVERSITÄT BAYREUTH

Computational Design & Analysis of Specific Binding Pockets for Natural and Modified Amino Acids

DISSERTATION

zur Erlangung des akademischen Grades einer

Doktorin der Naturwissenschaften (Dr. rer. nat.)

an der Fakultät für Biologie, Chemie und Geowissenschaften

der Universität Bayreuth

vorgelegt von

Merve Ayyıldız

geboren in Diyarbakır, Türkei

Bayreuth, 2025

Die vorliegende Arbeit wurde in der Zeit von Oktober 2020 bis April 2025 in Bayreuth am Lehrstuhl für Biochemie unter Betreuung von Frau Prof. Dr. Birte Höcker angefertigt.

Vollständiger Abdruck der von der Fakultät für Biologie, Chemie und Geowissenschaften der Universität Bayreuth genehmigten Dissertation zur Erlangung des akademischen Grades einer Doktorin der Naturwissenschaften (Dr. rer. nat.).

Art der Dissertation: Monographie

Dissertation eingereicht am: 26.03.2025

Zulassung durch die Promotionskommission: 02.04.2025

Wissenschaftliches Kolloquium: 11.07.2025

Amtierender Dekan: Prof. Dr. Cyrus Samimi

Prüfungsausschuss:

Prof. Dr. Birte Höcker (Gutachterin)

Prof. Dr. Matthias Ullmann (Gutachter)

Prof. Dr. Carlo Unverzagt (Vorsitz)

Prof. Dr. Johannes Margraf

The presented dissertation is written as a monograph.

Parts of the work have already appeared in the following publications:

- Modular peptide binders–development of a predictive technology as alternative for reagent antibodies

  FJ Gisdon*, JP Kynast*, **M Ayyildiz***, AV Hine, A Plückthun, B Höcker, *Biological Chemistry*, 2022, 403 (5-6), 535-543 (* contributed equally to this work)

  https://doi.org/10.1039/D2RA00470D

- Complementary evaluation of computational methods for predicting single residue effects on peptide binding specificities

  M Ayyildiz*, J Noske*, FJ Gisdon*, JP Kynast*, B Hocker, *bioRxiv*, 2024. (* contributed equally to this work)

  https://doi.org/10.1101/2024.10.18.619108

Following articles are not part of this thesis:

- Identification of alternative allosteric sites in glycolytic enzymes for potential use as species-specific drug targets

  M Ayyildiz*, S Celiker*, F Ozhelvaci*, ED Akten, *Frontiers in Molecular Biosciences* 7, 88, 2020. (* contributed equally to this work)

  https://doi.org/10.3389/fmolb.2020.00088

- Effective drug design screening in bacterial glycolytic enzymes via targeting alternative allosteric sites

  I Turkmenoglu, G Kurtulus, C Sesal, O Kurkcuoglu, **M Ayyildiz**, S Celiker, F Ozhelvaci, X Du, G Liu, M Arditi, ED Akten, *Archives of Biochemistry and Biophysics*, 762, 110190, 2024.

  https://doi.org/10.1016/j.abb.2024.110190

Table of Contents

Summary

Reagent antibodies are widely used in research and diagnostics, yet more than half fail to recognize their targets or exhibit poor specificity, leading to unreliable results. Their production relies on costly and time-consuming immunization processes that are difficult to reproduce. These challenges highlight the urgent need for alternative, well-characterized binding systems that are both reliable and customizable. To address this, the multidisciplinary PRe-ART project aims to replace conventional reagent antibodies with engineered armadillo repeat proteins, which are designed to bind specific amino acid sequences through modular assembly.

This thesis contributes to this objective by focusing on the computational design and analysis of binding modules within the designed armadillo repeat protein scaffold, with a particular emphasis on phosphorylated amino acids. Given that phosphorylation of serine, threonine, and tyrosine residues is critical in signaling pathways and metabolism, the lack of high-performing monoclonal antibodies for their detection presents a major challenge. To overcome this, a computational pipeline was established to design binders for phosphotyrosine, phosphoserine, and phosphothreonine. The pipeline prioritized key interactions while minimizing unfavorable mutations, leading to focused binder library suggestions to generate rationally designed libraries. Although experimental validation is ongoing, preliminary results indicate promising binding results. To improve the accuracy of computational predictions, this thesis also evaluates different computational methods that are used for binder library designs, providing insights into their prediction accuracy upon point mutations. By evaluating strengths and weaknesses of these methods, systematic tendencies of them were discovered and a complementary approach that enhances binder design reliability is proposed. Finally, to efficiently validate promising computational predictions, binding affinity measurements for protein-peptide interactions using dArmRPs within our research group were established. Implementing this setup in our lab enabled faster and more practical testing of designed binders, providing a reliable framework for assessing the accuracy of computational predictions.

Altogether, this thesis advances the understanding of protein-peptide interactions and demonstrates how this knowledge can be leveraged for the rational design of protein-based binders. By integrating computational strategies with experimental validation, this work contributes to the development of reliable alternatives to traditional reagent antibodies.

Zusammenfassung

Reagenzien-Antikörper sind in der Forschung und Diagnostik weit verbreitet, doch mehr als die Hälfte erkennt ihre Ziele nicht oder weist eine geringe Spezifität auf, was zu unzuverlässigen Ergebnissen führt. Ihre Herstellung beruht auf kostspieligen und zeitaufwändigen Immunisierungsverfahren, die sich nur schwer reproduzieren lassen. Diese Herausforderungen unterstreichen den dringenden Bedarf an alternativen, gut charakterisierten Bindungssystemen, die sowohl zuverlässig als auch anpassbar sind. Das multidisziplinäre Projekt PRe-ART zielt darauf ab, herkömmliche Reagenzien-Antikörper durch konstruierte Armadillo-Repeat-Proteine zu ersetzen, die durch modulare Zusammenstellung spezifische Aminosäuresequenzen binden sollen.

Die vorliegende Arbeit trägt zu diesem Ziel bei, indem sie sich auf die rechnerische Entwicklung und Analyse von Bindungsmodulen innerhalb des entwickelten Armadillo-Repeat-Proteingerüsts konzentriert, wobei der Schwerpunkt auf phosphorylierten Aminosäuren liegt. Da die Phosphorylierung von Serin-, Threonin- und Tyrosinresten für Signalwege und den Stoffwechsel von entscheidender Bedeutung ist, stellt der Mangel an leistungsfähigen monoklonalen Antikörpern für deren Nachweis eine große Herausforderung dar. Um dieses Problem zu lösen, wurde eine computergestützte Pipeline zur Entwicklung von Bindereagentien für Phosphotyrosin, Phosphoserin und Phosphothreonin entwickelt. Die Pipeline priorisiert die wichtigsten Interaktionen und minimiert gleichzeitig ungünstige Mutationen, was zu gezielten Vorschlägen für Bindungsbibliotheken führt, um rational konzipierte Bibliotheken zu erstellen. Obwohl die experimentelle Validierung noch nicht abgeschlossen ist, deuten die vorläufigen Ergebnisse auf vielversprechende Bindungsergebnisse hin. Um die Genauigkeit der rechnerischen Vorhersagen zu verbessern, werden in dieser Arbeit auch verschiedene rechnerische Methoden bewertet, die für die Entwicklung von Bindungsbibliotheken verwendet werden, und Einblicke in ihre Vorhersagegenauigkeit bei Punktmutationen gegeben. Durch die Bewertung der Stärken und Schwächen dieser Methoden wurden systematische Tendenzen entdeckt, und es wurde ein komplementärer Ansatz vorgeschlagen, der die Zuverlässigkeit der Bindereagenziendesigns erhöht. Um vielversprechende rechnerische Vorhersagen effizient zu

validieren, wurden schließlich in unserer Forschungsgruppe Bindungsaffinitätsmessungen für Protein-Peptid-Wechselwirkungen mit dArmRPs durchgeführt. Die Implementierung dieses Aufbaus in unserem Labor ermöglichte eine schnellere und praktischere Prüfung der entworfenen Bindereagenzien und bot einen zuverlässigen Rahmen für die Bewertung der Genauigkeit der rechnerischen Vorhersagen.

Insgesamt trägt diese Arbeit zu einem besseren Verständnis von Protein-Peptid-Wechselwirkungen bei und zeigt, wie dieses Wissen für das rationale Design von proteinbasierten Bindereagenzien genutzt werden kann. Durch die Integration von Berechnungsstrategien mit experimenteller Validierung trägt diese Arbeit zur Entwicklung zuverlässiger Alternativen zu herkömmlichen Reagenzien-Antikörpern bei.

List of Figures

List of Tables

List of Abbreviations

ATLIGATOR	ATlas-based LIGAnd binding editor
BBK*	Branch and bound over K*
CD	Circular dichroism
dArmRP	Designed armadillo repeat protein
DARPin	Designed ankyrin repeat protein
DEE	Dead end elimination
E. coli	*Escherichia coli*
GMEC	Global minimum energy conformation
IMAC	Immobilized Metal Affinity Chromatography
IPTG	Isopropyl-β-D-thiogalactoside
LB	Lysogeny Broth
mAb	Monoclonal Antibody
MD	Molecular Dynamics
MW	Molecular Weight
nArmRP	Natural Armadillo Repeat Protein
PBS	Phosphate Buffered Saline
PCR	Polymerase Chain Reaction
PDB	Protein Data Bank
PRe-ART	Predictive reagent antibody replacement technology
PTMs	Post translational modifications
RMSD	Root mean square deviation
SCOP	Structural classification of proteins

sfGFP	Super Folder Green Fluorescent Protein
TB	Terrific Broth
VMD	Visual Molecular Dynamics

1. Introduction

1.1 Proteins

Proteins are fundamental macromolecules in all living organisms with diverse structures and functions. They are involved in nearly all biological processes, such as catalyzing biochemical reactions, regulating metabolism, facilitating protein synthesis, and modulating cellular signaling pathways (Pawson & Nash, 2003). These functions are primarily mediated through interactions with other molecules, such as small ligands, nucleic acids, and other proteins (Deribe, Pawson, & Dikic, 2010; Jones & Thornton, 1996; Marsh & Teichmann, 2015). Since proteins are involved in nearly every cellular process due to their interactions with other proteins, investigating proteins and their interaction partners is of great interest.

Proteins binding to other proteins occurs when they proteins form physical contacts that can be either reversible or irreversible, arising through specific binding events that contribute to cellular function (de Las Rivas & Fontanillo, 2010). These interactions can occur through structured (folded) regions of proteins or flexible, unstructured amino acid stretches known as peptides. Peptides, despite their simplicity as short chains of amino acids, mediate up to 40% of protein-protein interactions within cells, highlighting their biological importance (Diella et al., 2008; Petsalaki, Stark, García-Urdiales, & Russell, 2009). Understanding these interactions is crucial for advancing therapeutic development, synthetic biology, biotechnology, and fundamental biology (Fosgerau & Hoffmann, 2015). In this context, protein binders, which are protein-based affinity reagents that can selectively bind to target proteins, are becoming increasingly important tools for studying protein function in living cells and organisms (Harmansa & Affolter, 2018; Helma, Cardoso, Muyldermans, & Leonhardt, 2015). Protein binders are not only widely used as therapeutic agents for conditions such as cancer and autoimmune diseases (W. Chen, Ying, & Dimitrov, 2013; Weidle, Auer, Brinkmann, Georges, & Tiefenthaler, 2013), but are also

extensively employed as reagents in diagnostics, where they play a crucial role in detecting and quantifying specific proteins or biomarkers in various assays, including enzyme-linked immunosorbent assay (ELISA), western blots, affinity chromatography and immunohistochemistry (Alhajj, Zubair, & Farhana, 2025; Borrebaeck, 2000; Dimitrov, 2012).

1.2 Traditional Protein Binders (Antibody-Based Scaffolds)

Antibodies are the most widely used protein binding reagents both therapeutically and in diagnostic research. These are naturally occurring Y-shaped proteins produced by the immune system. They recognize and bind to epitopes on antigens, i.e., foreign molecules such as proteins that trigger an immune response, via highly variable regions known as complementarity-determining regions. (Schroeder & Cavacini, 2010). However, despite their versatility, antibodies also present several challenges, including their large size, complex production processes, and potential for immunogenicity. Antibodies that are used in therapeutic purposes are thoroughly tested, requiring approval at multiple stages before clinical use. However, this is often not the case for reagent antibodies (Bradbury & Plückthun, 2015).

Polyclonal antibodies, which are specific to multiple epitopes for instance, face significant challenges with batch-to-batch reproducibility (Bradbury & Plückthun, 2015). Each production batch may yield a different antibody mixture due to their reliance on the lifespan of the immunized animal. Once the animal is no longer available, reproducing an identical antibody mixture becomes nearly impossible, limiting their consistency over time. Monoclonal antibodies (mAbs) on the other side, have emerged as one of the most widely used and well-characterized protein binders that recognize specific proteins of interest since their first discovery in the 19th century (Lipman, Jackson, Trudel, & Weis-Garcia, 2005). These molecules can recognize and bind to specific target antigens with high affinity and specificity, while many commercial antibodies often lack these properties. Despite their advantages, mAbs present several limitations, including their complex production processes, potential for immunogenicity or even their intellectual

property protections (Reichen, Hansen, & Plückthun, 2014). The production of mAbs involves immunizing an animal, such as a mouse, with an epitope of a target antigen, stimulating the generation of B cells that produce antibodies specific to the antigen. These B cells are then fused with myeloma cells to generate hybridomas, which then secrete the desired monoclonal antibody (Köhler & Milstein, 1975).

The mAbs are usually selected for their specific target affinity by phage display (or other display techniques) where the mAb is expressed and presented on the phage (or cell) surface and the pools are subjected to several rounds of selection where the ones with best affinity are isolated, the DNA is extracted and further sequenced (Hanes & Plückthun, 1997; G. P. Smith, 1985). However, this technique relies on the immune system to recognize specific epitopes of antigens and, the immune response can lead to off-target binding, resulting in non-specific antibodies that may not bind as tightly or specifically as desired. In addition,the observation that batches of animal-derived antibodies widely vary has brought concern to the reproducibility of experiments (Baker, 2015). Furthermore, the production of these antibodies relies heavily on animal experiments, raising ethical questions, as it necessitates the use of live animals. This relience adds another layer of complexity to their widespread use. Production requires also significant amount of time and resources, since the process involves several steps of purification and characterization, selection has to be performed individually for each target and the selected binder is subsequently individually evaluated. Additionally, unlike well-characterized therapeutic antibodies, about half of the commercially available reagent antibodies have previously been shown to not function correctly either in terms of their specificity or in recognizing their target at all (Bradbury & Plückthun, 2015). Considering their shortcomings as being time- and resource-consuming in addition to having significant specificity issues, and considering their extensive use in research and diagnostics, the development of alternative affinity reagents became of great interest (Banta, Dooley, & Shur, 2013; Forrer, Stumpp, Binz, & Plückthun, 2003).

1.3 Modular Peptide Binders (Alternatives to Antibody-based Scaffolds)

Many protein scaffolds have been developed to bind epitopes, offering opportunities to design tailored applications (Skerra, 2007). These scaffolds include a variety of affinity reagents; however, some may have lower binding affinities than antibodies or may have limited target specificity (Luo, Liu, & Cheng, 2022). To address the limitations of antibodies, repeat proteins have emerged as promising scaffolds. Their modular, repeat-based structure allows for precise customization of binding properties, enabling high affinity and specificity. Unlike traditional antibodies, repeat proteins can be highly stable, can be produced in cost-effective bacterial systems, and exhibit low immunogenicity (Banta et al., 2013). Additionally, their ability to target both extracellular and intracellular proteins further increases their use in therapeutic, diagnostic context. A significant advantage of repeat proteins over conventional binders is their modular binding mechanism, where each repeat unit contributes to the interaction with the target in a predefined manner. This modularity facilitates engineering efforts to create binders tailored to specific sequences. One key area of interest for modular peptide binders is the detection of post-translational modifications, which often occur in disordered or unfolded regions of proteins for example, e.g. on western blots. This capability is crucial for biochemical assays that rely on precise recognition of such modifications, positioning modular peptide binders as valuable tools in experimental biology.

Among the repeat protein scaffolds, Designed Ankyrin Repeat Proteins (DARPins) stand out as a well-characterized and versatile example. DARPins offer customizable sizes and a concave binding site that is particularly favorable for engaging larger epitopes (Binz et al., 2004; Binz & Plückthun, 2005; Bradbury & Plückthun, 2015; Forrer et al., 2003). However, despite their advantages, DARPins must be developed anew for each target, requiring a time-intensive selection process using techniques such as ribosome display or phage display. Furthermore, their design is optimized for folded protein targets, making them less suitable for recognizing linear peptide sequences (Schilling et al., 2022). Nevertheless, DARPins are widely applied in fields

ranging from diagnostics to tumor targeting and even viral retargeting strategies (Münch et al., 2011).

In this context, armadillo repeat proteins (ArmRPs), which bind their peptides in an extended way, providing a pocket for each side chain, and giving access to a modular approach may provide an advantage. Their tandem repeat architecture enables the creation of continuous peptide-binding surfaces, making them particularly well-suited for recognizing linear peptide targets in extended conformations.

1.3.1 Naturally Occurring Armadillo Repeat Proteins

Natural Armadillo Repeat Proteins (nArmRPs) were first discovered in the early 1990s in the fruit fly *Drosophila*, and were named "armadillo" because of the segmented and armored appearance of embryo's lacking a functional version of the protein, reminiscent of the animal (Perrimon & Mahowald, 1987; Wieschaus & Riggleman, 1987). nArmRPs are a highly conserved protein family characterized by tandem repeat motifs, each typically consisting of around 42 amino acids that form three α-helices (H1, H2, H3), (Figure 1) and connecting loops. These helices fold into a triangular structural motif and stack in a repeating manner to create a superhelical structure (Peifer, Berg, & Reynolds, 1994; Reichen et al., 2014). While H3 contributes mainly to the interactions for target binding, H1 and H2 mainly make up the hydrophobic core (Groves & Barford, 1999). This unique conformation generates a groove or surface that facilitates interactions with other proteins (Coates, 2003; Kobe & Kajava, 2001). Short, stretched out peptides bound by ArmRPs are in a highly conserved conformation, and antiparallel orientation --the protein's N-terminus interacts with the peptide's C-terminus--. ArmRPs bind to extended peptides always in a modular way providing a pocket for each side chain.

The two well-characterized subfamilies of nArmRPs are importin-α and β-catenin, both of which are essential for key cellular processes such as cell adhesion and signaling pathways. They differ in net electrical charge of the target peptide and depending on the subfamily, bind to target

peptides containing either positively or negatively charged conserved residues. In the importin-α subfamily, a conserved Asn residue in H3 is present in nearly all repeats and forms hydrogen bonds that stabilize the bound peptide in an extended conformation. On the other hand, β-catenin contains substitutions such as histidine and glutamine in specific repeats, similar to Asn in importin-α subfamily for the peptide backbone stabilization. nArmRPs have characteristics to be used in the development of modular peptide binders, such as they bind extended peptides in a conserved, modular way and feature a repetitive, rigid structure ideal for engineering adaptable binding surfaces to match peptide lengths. However, they have limitations which are discussed below in more detail.

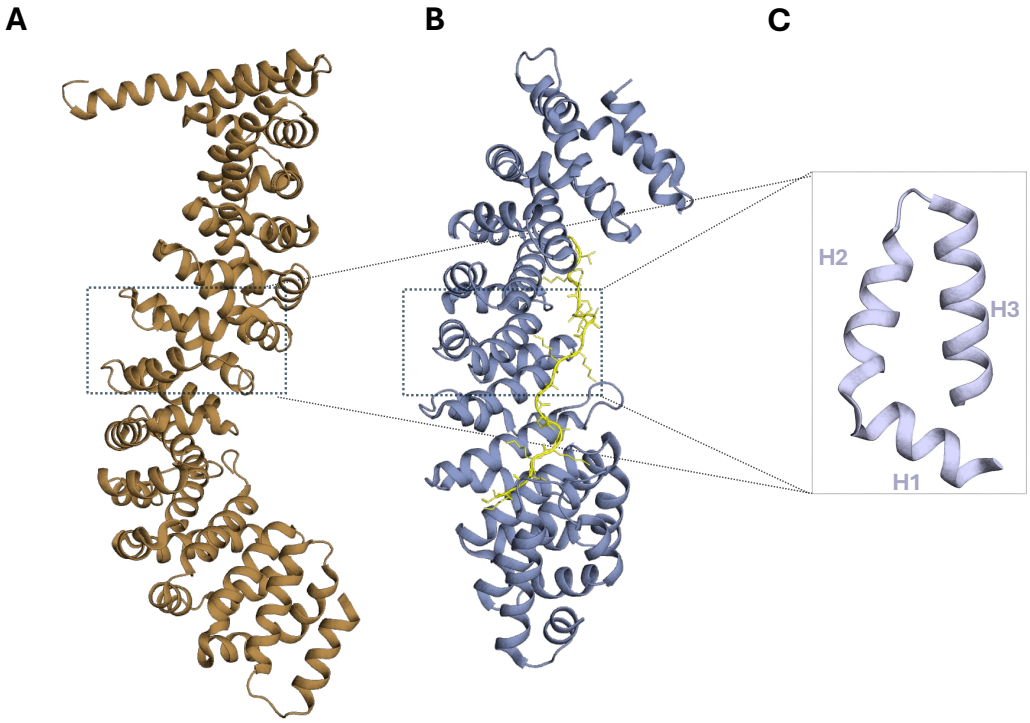

Figure 1: Structural representation of described nArmRPs. (A) Structure of S. cerevisiae importin-α (PDB-ID: 1EE5). (B) Crystal structure of Zebrafish β-Catenin (PDB-ID: 2Z6G). (C) One internal ArmRP repeat consisting of helices H1, H2 and H3.

1.3.2 Engineering Designed Armadillo Repeat Proteins

While natural nArmRPs show potential to be used as modular peptide binders with their binding modes, their variable curvature and low sequence identity among repeats restrict the amount of short peptides they can bind, particularly within a limited number of consecutive repeats. Additionally, the biophysical characteristics of these proteins are constrained by sequence diversity. Over more than ten years, the Plückthun Group successfully engineered a highly stable designed Armadillo Repeat Protein (dArmRP) by converting the irregular structure of natural ArmRPs into a protein with regular, stackable repeats, creating an efficient modular peptide binder (Gisdon et al., 2022).

The design process involved using a consensus approach, where sequences from the internal repeats of proteins like importin-α and β-catenin were aligned to identify conserved residues critical for stability and binding (Parmeggiani et al., 2008). These residue contacts were further optimized using molecular dynamics simulations. To protect the hydrophobic core, N- and C-terminal caps were created from the importin-α scaffold in *Saccharomyces cerevisiae*, with additional mutations suggested through further simulations (Alfarano et al., 2012; Madhurantakam, Varadamsetty, Grütter, Plückthun, & Mittl, 2012; Parmeggiani et al., 2008). The internal repeats, crucial for recognizing and binding target peptides, were optimized to improve both binding affinity and the biophysical properties of the protein. Reichen et al., 2016, identified a repeat pair from yeast importin-α with optimal curvature for binding extended peptides, which led to the design of dArmRP bound to a $(KR)_4$ peptide with picomolar affinity. Hansen et al., 2016 showed that increasing the number of internal repeats and peptide units improved binding affinity and contributes independently to the binding that confirms the modularity. Alanine scanning experiments confirmed that each binding pocket contributed regularly and consistently to the overall effect. Additionally, a crystal structure of the dArmRP with a bound $(KR)_5$ peptide (PDB-ID: 5AEI) was obtained, where the main interactions between the peptide and the dArmRP could be identified. The arginine in the peptide interacts with the Asn side chain of a dArmRP

repeat as in importin-α, which generates general, unspecific affinity and ensures regularity (Figure 2).

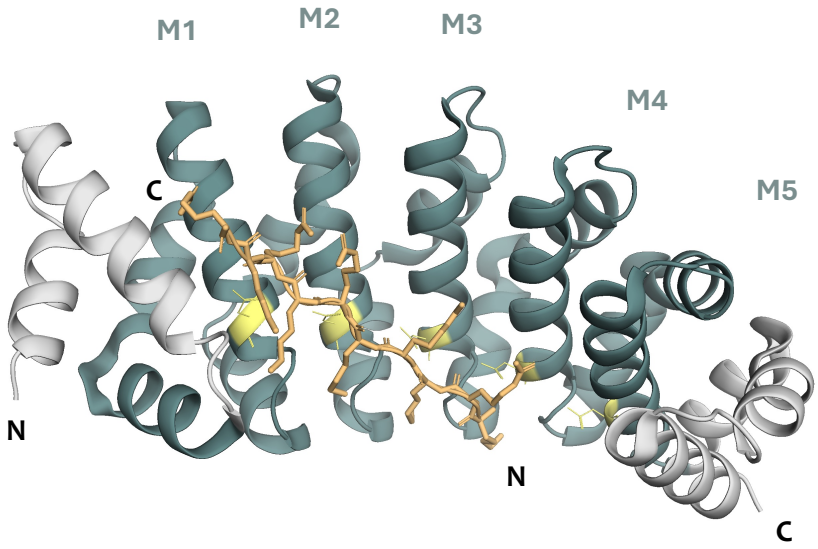

Figure 2: Structure of the dArmRP bound to peptide $(KR)_5$. The peptide, bound in an anti-parallel orientation is shown in orange sticks (PDB-ID: 5AEI). The protein is depicted as ribbons with the N-terminal and C-terminal cap colored in white. The five internal repeats, represented as M1-M5, each possessing three α-helices, are colored dark. Conserved Asn in each repeat is colored yellow (Adapted from Hansen et al., 2016).

An inherent problem with repeat proteins, particularly those like dArmRPs that bind repetitive sequences, is the potential for the target peptide to bind in multiple registers. The ways to overcome this issue, the peptide binding in different orientations, were examined by Ernst et al. 2020. To prevent the peptide from binding in undesired orientations, a lock was incorporated into the dArmRP by grafting a hydrophobic binding site observed in beta-catenin onto the dArmRP, thereby locking the peptide with the complementary sequence in place. This modification ensured that the peptide was locked in place with the complementary sequence aligned properly. The interaction of the lock was improved by mutual optimization of the pocket and the bound peptide, which were then confirmed by X-ray crystallography. Since dArmRPs are

designed as modular peptide binders, it is critical to carefully assess each module and the interactions between peptide side chains and the binding pockets. However, Hansen et al., 2016 pointed out that the curvature of the dArmRP scaffold and peptide binding were significantly influenced by crystal contacts. To prevent any unwanted effects from these crystal contacts and to shield the binding surface, Designed Ankyrin Repeat Proteins (DARPins) were fused to the dArmRPs. The resulting crystal structure of this ring-like construct demonstrated a fully shielded binding surface, ensuring consistent and regular peptide binding (Figure 3, PDB-ID:6SA8, and Ernst et al., 2019).

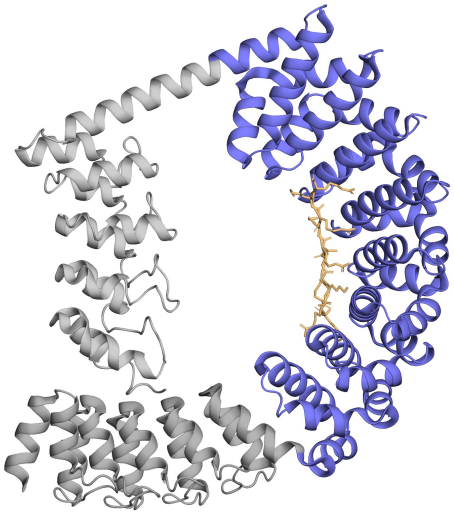

Figure 3: Structure of the dArmRP bound to the peptide $(KR)_5$ with fused DARPins. The peptide, bound in an anti-parallel orientation is depicted in orange (PDB-ID: 6SA8). The DARPin is colored in grey and the dArmRP is colored in purple (Figure is taken from Ayyildiz et al., 2024).

1.4 Predictive REagent Antibody Replacement Technology (PRe-ART)

The PRe-ART project aims to replace underperforming reagent antibodies with modular reagents based on a protein-peptide system. This modular reagents can be assembled to bind specifically and tightly to linear peptide targets for widespread use as protein-binding reagents. In doing so, it will remove the expense of a screening experiment for each new target by assembling prescreened modules to generate a binder specific to the full epitope (Gisdon et al., 2022). As a modular peptide binding scaffold, designed armadillo repeat proteins (dArmRPs) have been developed based on natural armadillo repeat proteins by the Plückthun group (see section 1.2.2). Importantly, the dArmRPs require a peptide target in an extended configuration, and the target protein must be in an unfolded state. Thus some of the potential applications of these binders are binding and recognition of unfolded stretches at the termini of proteins or in linker regions, denatured proteins in SDS-PAGE and western blots. Most importantly, many regions of particular interest, such as tails of receptors that are phosphorylated or the tails of histones that are methylated or acetylated (among others) are essentially unstructured, in addition to the intrinsically disordered proteins (IDPs). By generating individual armadillo repeat subunits that are dipeptide specific and sequence defined, peptide-specific, reproducible binder can be engineered by assembling the necessary modules (Figure 4). This makes them versatile and suitable for modifications in protein engineering approaches for further use in a wide variety of applications such as, with protein array systems for proteomics and clinical diagnostics based on this reagent platform, detection accessories, and protein purification kits.

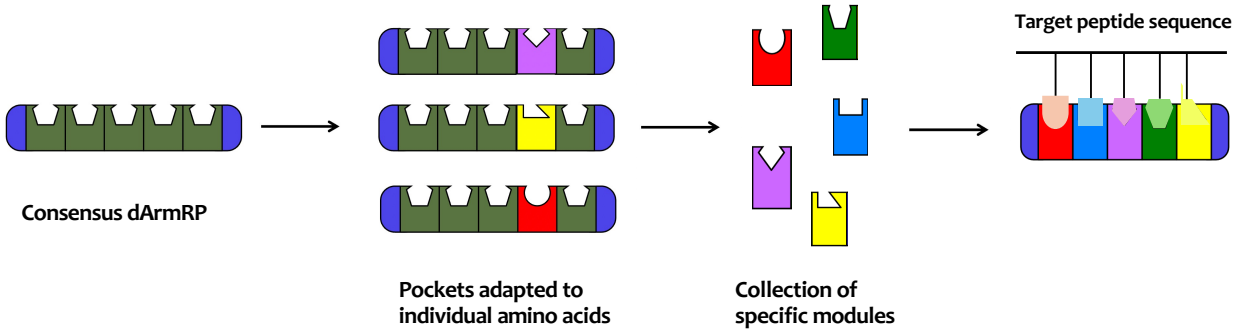

Figure 4: Generation of specific peptide binders with dArmRPs. Pre-selected specific modules are assembled to bind a longer peptide without the need to performing additional selections (Adapted from Henning, PhD Thesis, 2017).

One of the most important targets of the project is the development of binders for the phosphorylated amino acids in the context of detecting important phosphorylation sites. To exploit a catalog of pairs of binders for phosphorylated-unphosphorylated targets would enable visualizing the effect of candidate drugs on whole signaling pathways, and eventually to the entire phosphorylation based signaling of the cell (see section 1.5). Such an approach would accelerate mass spectrometry detection and thus permit to incorporate such a workflow in drug discovery programs. The ultimate aim is to have binding proteins for sites for which no traditional antibodies are available, or for which they are not specific enough. Initially this could be tested with those examples for which some traditional antibodies are available. Considering there are almost no antibodies available that exclusively recognize the non-phosphorylated form, this would fill a major gap in the field (for detailed information about PRe-ART, see Gisdon et al., 2022 and https://preart-2t.uni-bayreuth.de/).

The modules or repeat units of dArmRPs where each module binds two adjacent amino acids in alternating orientations, are derived from the importin-alpha framework. One binding pocket exhibits specificity for arginine, while the other is binding to lysine, with the lysine pocket being comparatively shallower (Figure 5) (Hansen et al., 2016). The interactions between dArmRPs and individual residues are highly specific, with each amino acid fitting into its respective pocket. In

PReART, individual modules are engineered to recognize specific amino acids. Figure 5 shows the binding pockets for arginine at position 6, and lysine binding pocket at position 5 of the peptide. Throughout this thesis, argining pocket was the primary focus, with efforts concentrated on modifying its specificity to accommodate other amino acids through targeted mutations of binding pocket residues.

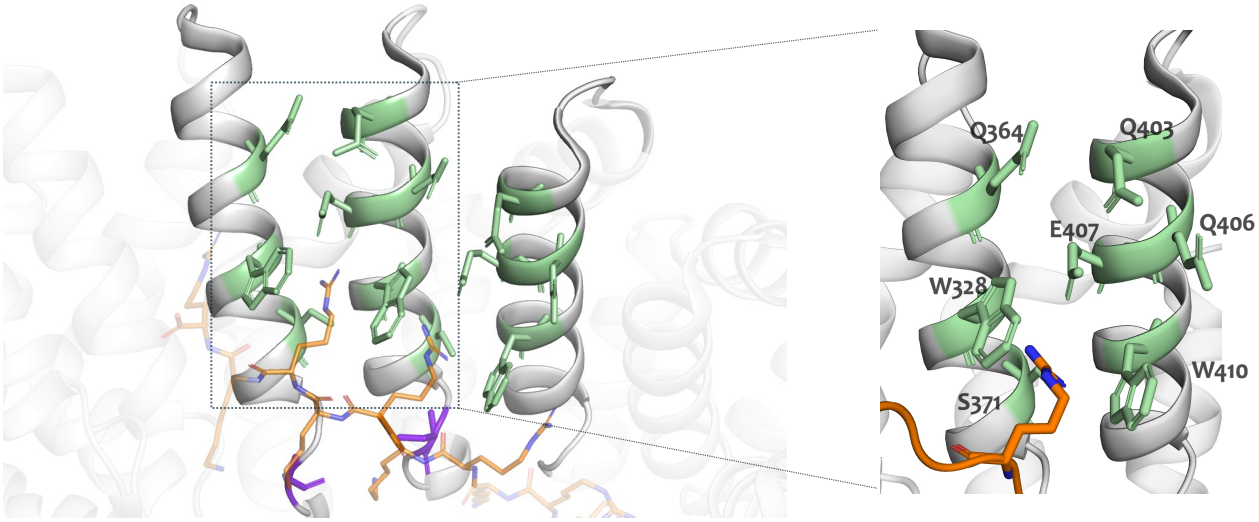

Figure 5: Visual representation of the mutatable positions in the standard arginine and lysine pockets. The residues of one argining pocket are highlighed with residue names and numbers (green, as sticks). In the lysine pocket, possible residues to be mutated are colored in purple sticks. The scaffold used for representation is PDB-ID: 6SA8.

Developing modular, sequence-specific binding strands for linear epitopes using dArmRPs requires an interdisciplinary effort. PReART project uses a feedback loop that combines computational modeling with experimental approaches to iteratively develop novel dArmRP (modules) (Figure 6). To design new modules, a key focus is on altering the pocket residues of dArmRPs to achieve high affinity binding for a particular amino acid, while achieving specificity over other amino acids. The research methodology is explained below in more detail below.

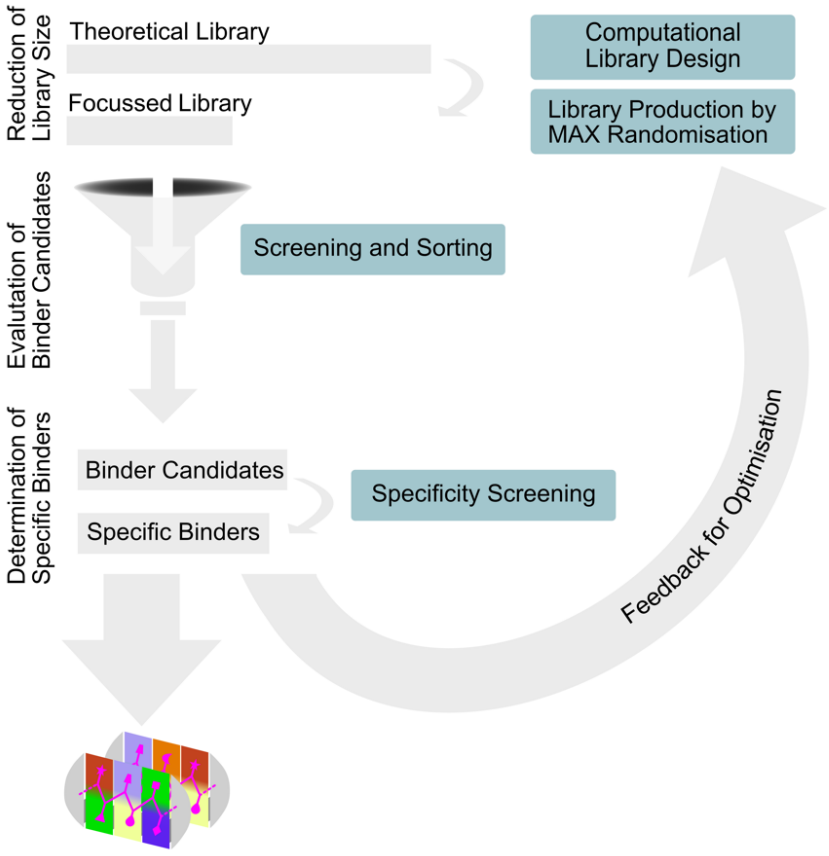

Figure 6: Workflow in the engineering of binding modules. Libraries are designed, synthesized, screened, and evaluated, providing feedback to the input techniques. The overall loop creates an ensemble of binding modules that can later be assembled to recognize predefined target peptides (Figure is taken from Gisdon et al., 2022).

1.5 Importance of Recognizing Phosphorylated Residues

Phosphorylation is a common reversible post-translational modification where a phosphate group (PO_4^{3-}) is added to a protein, typically by a protein kinase enzyme. This modification plays a crucial role in regulating the structure, function, localization, and activity of proteins within cells (Acconcia, Barnes, Singh, Talukder, & Kumar, 2007; Manning, Whyte, Martinez, Hunter, & Sudarsanam, 2002; Tarrant & Cole, 2009). Regarding cell signaling, phosphorylation is a critical mechanism in cellular signaling pathways; it transmits signals and can start a cascade of events. It functions as a molecular switch, directly modulating protein activity (Ardito, Giuliani, Perrone, Troiano, & Muzio, 2017; Harsha & Pandey, 2010). Although phosphorylation can occur on several amino acid residues in proteins such as histidine, aspartic acid, and glutamic acid, it is most commonly found on serine and threonine residues and comparably less abundant but as crucial on tyrosine residues (Ardito et al., 2017; Ubersax & Ferrell, 2007). Phosphorylation of tyrosine, serine, and threonine residues represents distinct types of post-translational modifications with unique roles in cell signaling and regulation. While phosphorylated tyrosine residues predominantly mediate signaling pathways initiated by receptor tyrosine kinases, phosphorylated serine, and threonine residues are involved in a wide range of cellular processes and signaling pathways, regulating protein function, interactions and cellular responses. However, they collectively regulate protein function and interactions, albeit through distinct mechanisms. Notably, post-translational modifications often occur in unstructured protein regions, such as receptor tails or signal transduction regions (Dyson & Wright, 2005; N. Liu, Guo, Ning, & Duan, 2020).

The focus of PRe-ART lies in harnessing the ability to bind linear target sequences in an unfolded state, thereby enabling specific targeting and investigation of post-translational modifications (Gisdon et al., 2022). Particularly intriguing is the prospect of developing binder pairs for phosphorylated and unphosphorylated targets, offering insights into the effects of candidate drugs on signaling pathways. This approach holds promise for integrating such workflows into

drug discovery endeavors. In addition, there is an opportunity to generate orthogonal pairs of binders that are identical, except having a pocket for Ser vs. pSer, Thr vs. pThr, and Tyr vs. pTyr and thereby, directly obtained from binding experiments of ratios of modified vs. unmodified amino acids on a particular receptor. Determination of modification content by binding proteins can be used with microscopy, ELISA, and miniaturized chips, and the current technology would expand this to less well-researched signaling proteins.

1.6 Research Methodology in PRe-ART

The design of specific protein-protein or protein-peptide interactions has seen rapid progress, driven by the use of both experimental screening and computational approaches (T. S. Chen & Keating, 2012). Experimental methods, such as molecular display technologies combined with fluorescence-activated cell sorting (FACS), enable the high-throughput screening of vast libraries of protein variants and facilitate the identification of those with desired binding properties. Computational modeling further enhances this process by identifying critical residues and predicting favorable mutations, thereby narrowing down the searchable sequence space and improving screening efficiency.

In PReART, experimental and computational approaches are applied to engineer binding pockets within dArmRPs, that are proteins designed to recognize specific amino acid side chains. Given the vast search space resulting from the randomization of multiple interacting positions, a hybrid approach combining experimental screening with computational design is crucial. ParaMAX randomization, based on MAX randomization invented by the Hine group (Ashraf et al., 2013; Chembath et al., 2022) that restricts randomization to specified residues, is used to create focused libraries for screening. Yeast surface display, combined with FACS, proves effective for screening these libraries, especially when targeting positively charged peptides. Unlike ribosome and phage display systems, which face challenges with negatively charged peptides, yeast display ensures more successful selections. Next-generation sequencing (NGS) is employed to assess the

diversity and quality of the library before screening, further enhancing the precision of the process (Gisdon et al., 2020).

1.7 Computational Strategies in the Design of Specific dArmRP Modules

Designing functional protein binding pockets is a significant challenge due to the vast combinatorial complexity of potential amino acid sequences and their corresponding structural conformations and interaction dynamics. Computational tools have become indispensable in addressing this complexity, enabling efficient pre-selection of potential binding modes for experimental validation. Since their emergence in the late 20th century, computational methods in protein engineering have evolved dramatically, from early protein folding predictions to state-of-the-art tools for rational design and de novo generation of proteins. One of the earliest steps towards rational protein design was taken in the 1980s with the development of site-directed mutagenesis by allowing scientists to introduce specific mutations into proteins and study their effects. In 1993, Frances Arnold used directed evolution, a technique mimicking natural selection in the laboratory, to evolve enzymes with improved properties (K. Chen & Arnold, 1993). Her work ultimately led to her receiving the Nobel Prize in Chemistry in 2018. In 2003, the Baker group developed Rosetta, a software suite that revolutionized computational protein design by enabling more accurate predictions of protein structures and the design of novel proteins (Rohl, Strauss, Misura, & Baker, 2004). Later in 2003, Top7 that has not been observed in nature was designed by the Baker group using the RosettaDesign software (Kuhlman et al., 2003). In 2024, David Baker was honored with the Nobel Prize in Chemistry for his contributions to computational protein design. More recently, in 2020, the DeepMind AI program AlphaFold achieved unprecedented accuracy in predicting protein structures, marking a significant leap forward for computational biology and opening new avenues for protein design and engineering, also received the 2024 Nobel Prize in Chemistry (Jumper et al., 2021). Today, machine learning-based applications are increasingly incorporated into protein design pipelines by many

researchers worldwide. However, these methods require careful consideration of high-quality data, relevant features, and appropriate models, as well as thorough validation through cross-validation and independent test sets. Especially for protein-peptide and protein-protein systems, machine learning or deep learning based methods are not as widely used for protein-small molecule systems (Ayyildiz, Noske, Gisdon, Kynast, & Höcker, 2024).

Recent advancements in computational protein design have significantly improved efficiency and accuracy through enhanced algorithms and computing technologies, such as GPUs and parallel processing (Lechner, Ferruz, & Höcker, 2018; H. Liu & Chen, 2023). These innovations enable flexible models, simultaneous sequence sampling, and tools like Molecular Dynamics (MD) simulations for analyzing protein stability and dynamics. Despite their potential, the high computational cost of MD simulations and the challenges of conformational space exploration still limits its usage (Lazim, Suh, & Choi, 2020). Tools such as Rosetta and FoldX are widely used to identify low-energy amino acid sequences that fold into target structures, addressing challenges in conformational space exploration and energy function accuracy. Within Rosetta, algorithms like FastDesign and CoupledMoves facilitate efficient exploration of sequence space. FastDesign optimizes sequences through iterative side chain repacking and energy minimization, while CoupledMoves simultaneously adjust backbone, sidechain conformations, and sequences to enhance sampling effectiveness. Engineered proteins would ideally have high specificities for their intended targets, but achieving interaction specificity by design can be challenging (T. S. Chen & Keating, 2012). To achieve the intricate level of specificity required for effective protein engineering, researchers often rely on a combination of computational tools that complement each other's strengths. The complexity of protein-protein interactions, which involves not only the recognition of binding partners but also the precise alignment and stabilization of these interactions, demands sophisticated modeling approaches. Methods, such as flex ddG in Rosetta and BBK* in OSPREY, target single residue mutations, with flex ddG incorporating backrub motion to estimate binding affinity changes and BBK* efficiently approximating binding affinities with continuous flexibility (Barlow et al., 2018; A. A. Ojewole, Jou, Fowler, & Donald, 2018).

For modeling and design of new pockets within PRe-ART project, the software suite ATLIGATOR was developed to identify promising mutations in binding pockets that may enable specific binding to desired peptides (Kynast, Schwägerl, & Höcker, 2022). Using a knowledge-based approach, ATLIGATOR extracts pairwise interactions from known structures to inform the design of new binding pockets, incorporating the detection of common interaction patterns for specific amino acid side chains. Binding pockets proposed by ATLIGATOR can be further evaluated using algorithms like flex ddG or BBK*. The calculation of binding energies has become a very valuable tool in specifying libraries for new binding pocket suggestions. Additional insights could be gained through Molecular Dynamics (MD) simulations. This combination of methods provides insights of potential binding pocket candidates.

2. Aim and Motivation of the Thesis

Reagent antibodies have long been fundamental in biomedical research and diagnostics, yet their widespread issues with specificity, poor characterization, and reliance on unstable cell lines undermine reproducibility. To overcome these limitations, the PRe-ART project aims to develop engineered binding proteins as a more reliable and customizable alternative. This objective is approached by combining protein engineering, library screening, and computational design.

The goal of this thesis is to support the design of modular binders capable of recognizing phosphorylated amino acids, specifically phosphotyrosine, phosphoserine, and phosphothreonine thereby contributing to the development of a designed armadillo repeat protein module repertoire. To achieve this, computational protein design methods were employed to generate binding modules within the designed armadillo repeat protein scaffold. The focus was on optimizing protein-peptide interactions and binding pocket specificity using the tools Rosetta and ATLIGATOR, while molecular dynamics simulations provided structural insights into further binding behavior.

To assess the computational methods that are employed during the binder development, a second objective was to systematically evaluate the programs predictive accuracy in assessing binding specificity upon mutations. The goal was to assess well-established algorithms and identify systematic trends and limitations in these approaches in order to help refine computational predictions and improve computational binder design strategies. Since the availability of benchmarks for these is limited, more variants need to be experimentally tested. Accordingly, binding affinity measurements had to be established to enable quick and systematic assessment of designed binders, a setup that facilitates testing and refining of computationally designed binders.

3. Computational Design of Specific Binding Pockets for Phosphorylated Amino Acids

3.1 Key Aspects in Protein Binding Pocket Design

A protein binding site, often referred to as the protein binding pocket or ligand binding site, is a specific region on the protein surface characterized by a cavity where molecules like small ligands or other proteins can attach. These binding pockets are typically defined by their three-dimensional shape, chemical composition, and electrostatic properties, enabling selective interactions with ligands through noncovalent interactions such as hydrogen bonds, hydrophobic forces, and electrostatic interactions (Bartlett, Porter, Borkakoti, & Thornton, 2002; Henrich et al., 2010). Recognizing these interaction principles is essential, especially when modifying residues to enhance binding or creating entirely new binding sites. In addition to small molecules, proteins often interact with other macromolecules, such as peptides or proteins, forming larger complexes that rely on specific binding pocket interfaces for binding affinity (Eaton, Gold, & Zichi, 1995). These interactions often require the optimization of small, highly specific binding sites, where minor adjustments in residue positioning can significantly affect both binding affinity and specificity. The complexity of designing such precise interfaces highlights the growing demand for sophisticated binding pocket design strategies that can accommodate both natural and synthetic peptides.

When compared to general protein and antibody design, pocket design brings unique challenges. One major difficulty is maintaining overall protein stability and proper folding while shaping the desired pocket. Additionally, side chain atom comformations and interactions play a critical role in pocket design, requiring careful modeling (Dou et al., 2017; Zhang, Shen, Liu, & Zitnik, 2024).

Several computational methods have been developed to address these challenges in (re)designing binding pockets in proteins (Tinberg et al., 2013). Traditional computational approaches provide insights into protein-ligand interactions and guide binding site optimization for specific properties. For instance, PocketOptimizer (Noske, Kynast, Lemm, Schmidt, & Höcker, 2023) uses scoring functions based on physical force fields to optimize binding pockets. Several Rosetta protocols, such as RosettaLigand, employ Rosetta's empirical-driven energy terms (Lemmon & Meiler, 2012). Docking tools such as AutoDock predict binding poses and estimate ligand binding affinity by calculating free energy changes through a semi-empirical scoring function, while HADDOCK is widely used for protein-protein and peptide docking, refining their structures through flexible docking protocols (Dominguez, Boelens, & Bonvin, 2003; Goodsell & Olson, 1990).

As explained in the section above (see section 1.4), PRe-ART project is based on designed ArmRPs, and these proteins bind linear epitopes in an almost fully extended way. Because of the alternating "up" and "down" orientation of the side chains of the bound peptide (Figure 2), each repeat unit carries two separate pockets and thus binds two amino acid side chains (Arg and Lys) adjacent in the sequence (Figure 5). This thesis focuses on the upper binding pocket, often referred to as the Arg binding pocket, since it accommodates arginine in the consensus design. Fully randomizing binding pocket positions is not applicable for both the generation and, afterward, the screening of the library. Hence, computational methods can help reduce the size of libraries by limiting the possibilities of amino acids for one or more positions. One of the aims of this doctoral study is to computationally design binding pockets and suggest a binder library for each phosphorylated amino acid: phosphoserine, phosphothreonine, and phosphotyrosine. These binding pockets are designed always in the context of a longer peptide, and the pSer, pThr, or pTyr pocket are positioned at the sixth position of the peptide.

In below, computational approach for designing pTyr, pSer, and pThr libraries were explained. Several methods were used to suggest focused binder libraries for pTyr, pSer, and pThr. First, the amino acids that are interacting with one of three phosphorylated amino acids found in the protein data bank (PDB) were investigated, and the most common interaction partners, as well as common groups of interactions, were identified using ATLIGATOR and ATLIGATOR-web (Kynast & Höcker, 2023; Kynast et al., 2022). Second; the CoupledMoves algorithm from Rosetta was used to optimize the protein-peptide interface (Ollikainen, de Jong, & Kortemme, 2015). Third, several promising sequence candidates were evaluated using the flex ddG algorithm for analtysing probable specificity of binding (Barlow et al., 2018), and lastly, MD simulations were conducted to capture the behavior of peptide-ligand interactions in the modeled pocket for the most promising pocket sequence (Figure 7).

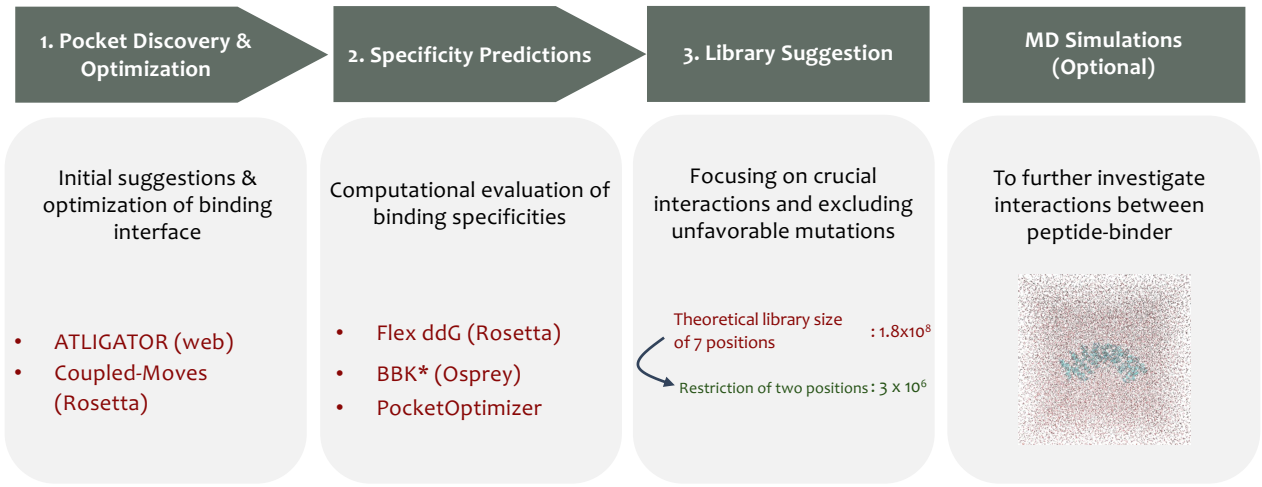

Figure 7: The general computational engineering approach that was established for desinging binder libraries. Workflow outlining the key steps: (1) Pocket Discovery and Optimization and (2) Specificity Predictions, each incorporating methods used in these steps, (3) Library suggestion by focusing on crucial interaction and (4) MD Simulations for detailed protein-peptide interaction analysis.

3.2 Computational Engineering Approach to Design Focused Libraries

3.2.1 ATLIGATOR & ATLIGATOR-web

Protein-protein or protein-peptide interactions are based on mutual interactions of amino acid residues, with certain residue-residue interactions being more crucial for overall interaction dynamics. Understanding how specific residue types interact is essential when creating newly designed proteins, specifically binding pockets. In this respect, ATLIGATOR (Atlas-based Ligand Binding Predictor) and ATLIGATOR-web toolchains, which support both manual and automated protein design backed by discovery algorithms, were used in this chapter. ATLIGATOR is a knowledge-based software tool written in Python, developed *in-house* to analyze protein-protein and protein-peptide interactions. The program relies on two important data sources; the *atlas*, which contains pairwise ligand-binder interaction information extracted from the PDB, and *pockets*, which describe frequent interaction patterns of a ligand and multiple binder residues. In addition to analyzing the *atlases* and their *pockets*, the ATLIGATOR can further be used to design binding pockets for ligand amino acids of interest by either allowing direct grafting (quick graft) of frequent interaction motifs onto a scaffold protein or by performing the design process manually by allowing selection of mutation independently (manual design). An overview of the ATLIGATOR toolchain is given in Figure 8. Each section of the ATLIGATOR toolchain is explained in more detail below with the explanation of ATLIGATOR-web (Kynast & Höcker, 2023; Kynast et al., 2022).

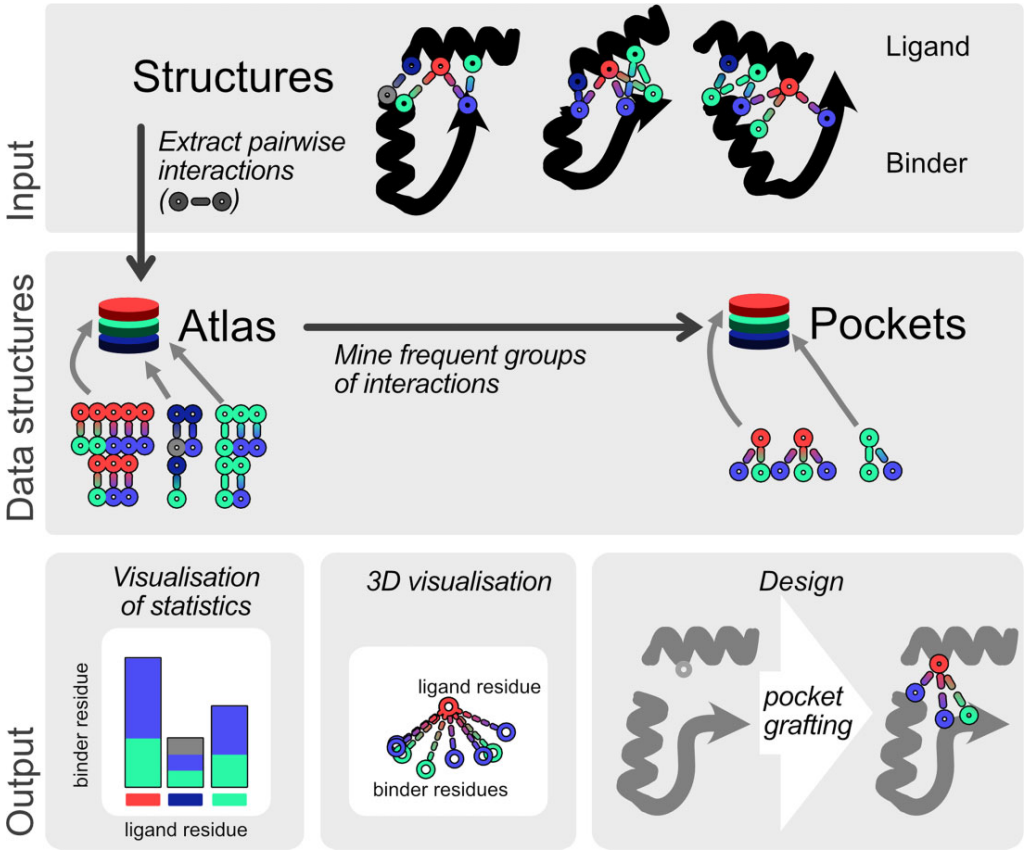

Figure 8: Overview of the ATLIGATOR tool chain. The ATLIGATOR based on two data structures *Atlas* and *Pockets*, capturing pairwise interactions from protein structures. ATLIGATOR supports both statistical and 3D visualizations. Additionally, *Pockets* can be grafting into provided protein structure (Figure is taken from Kynast J.P. et al, 2022).

ATLIGATOR-web offers a graphical user interface (GUI) to the ATLIGATOR tool, providing easy access and connecting its functions through several subsections for navigation. The first subsection, structures, allows users to create a structure collection by either uploading structures directly from their computer or specifying PDB codes or entries from the SCOPe database, which enables having structures in the structure collection with shared evolutionary backgrounds. Once the structures are uploaded, detailed information on individual interactions between residue pairs can be stored and visualized in a database called *atlases*. An *atlas* is a collection of data points that describe interactions between ligand residues and binder residues. Furthermore, the

tool facilitates the analysis of all pairwise interactions that a single residue forms with other residues in the selected structure collection. These pairwise interactions, stored in *atlases*, can then be exploited and grouped using algorithms within the next subsection, *pockets*. A *pocket* represents frequently occurring interaction patterns, where one residue's frequently interacting residues can be grouped and further analyzed. In the last subsection, *designs*, discovered pockets for the targeted amino acid can be transferred onto a scaffold of choice with a grafting algorithm of ATLIGATOR. This grafting involves applying a spatial similarity function to determine optimal positions for the residues in the selected pocket. Selected (pocket) positions can then be mutated accordingly, either after grafting or manually without grafting. Finally, in this section, ATLIGATOR-web has a function to relax this new scaffold, which uses the Rosetta fixbb side chain protocol to eliminate clashes and provide a more realistic representation. (Leaver-Fay et al., 2005).

In this section, the ATLIGATOR tool and ATLIGATOR-web were used to design pTyr binding pockets and suggest libraries for experimental evaluation. The data collected for pTyr was used also to inform the designs of pSer and pThr binding pockets. All crystal structures that include pTyr in the PDB database were searched via *in-house* Python script and structures were downloaded. These structures were added to a created structure collection in ATLIGATOR-web. Since there are fewer structures for phosphorylated amino acids in the PDB than standard amino acids, not only intermolecular interactions but intramolecular interactions were included in the search. An atlas was generated based on this structure collection and was used to find frequent interaction patterns stored in a pocket. In the last step, 6SA8 with a bound $(KR)_5$ peptide was uploaded as a scaffold, and the peptide position 6 was mutated to the targeted phosphorylated amino acid via the designs section in the ATLIGATOR-web. Both manual design where several promising amino acids were placed onto positions of inner shell residues and pocket grafting options were pockets that were suggested in previous steps placed onto a scaffold were used in order to determine the best possible combinations of mutations. Finally, the scaffold was minimized after every mutation. Here, primarily inner shell residues out of seven pocket residues that are position 3, 4, 6, and 7 were given more importance considering the distance between

ligand residue and pocket residue in terms of interactions (Figure 5). Structures were visualized via PyMOL to ensure the interaction between amino acids and the targeted residues. The current version of ATLIGATOR and ATLIGATOR-web do not support phosphorylated amino acids; therefore, the three most common phosphorylated amino acids, phosphotyrosine, phosphoserine, and phosphothreonine, were manually implemented in the local version in order to be able to mutate peptide position 6 to these amino acids via ATLIGATOR-web. All steps, except the pTyr search in the PDB database and implementation of phosphorylated amino acids, were carried out in the ATLIGATOR-web server.

3.2.2 CoupledMoves

Rosetta, developed through a collaborative community effort, offers a wide range of protocols built on physics- and knowledge-based potentials. One such protocol, CoupledMoves, is a Rosetta protocol for designing flexible backbone structures in small-molecule binding sites, protein-protein interfaces, and protein-peptide interactions (Ollikainen et al., 2015). It allows subtle adjustments to the protein backbone and side chain rotamers while maintaining energetically favorable interactions. Unlike traditional methods, CoupledMoves simultaneously moves the backbone and side chains during sampling rather than treating them separately. This coupled movement of the backbone torsion angles (phi and psi), side chain conformations, and ligand flexibility provides a more integrated and accurate approach to designing protein interactions. The protocol evaluates the energy of each conformation based on factors like van der Waals interactions, electrostatics, hydrogen bonding, and solvation effects. Conformations with lower or unchanged energy are accepted, while higher-energy states can be accepted probabilistically via the Metropolis criterion, allowing for occasional exploration of less favorable states (Metropolis, Rosenbluth, Rosenbluth, Teller, & Teller, 1953). This process is repeated through multiple iterations until the most energetically favorable structure is selected.

As a difference to non-coupled flexible backbone approaches from Rosetta, CoupledMoves, in the first two steps, moves the backbone and side chains and then follows the Monte-Carlo method to accept or reject the design instead of applying Monte-Carlo to accept or reject after each backbone and side chain moves (Rosenbluth & Rosenbluth, 1955). Unlike other flexible backbone design methods such as FastDesign or BackrubEnsemble (Loshbaugh & Kortemme, 2020; Maguire et al., 2021; C. A. Smith & Kortemme, 2008), where the backbone flexibility and sequence design are separated for acceptance during the sampling trajectory, CoupledMoves combines both movements of backbone and side chains in a single step where Monte-Carlo acceptance/rejection comes after. The backbone move can be applied with ShortBackrubMover, which is the default setting, and or with kinematic closure (KIC) with walking perturber or KIC with fragment perturber (Coutsias, Seok, Jacobson, & Dill, 2004). Nevertheless, CoupledMoves will not significantly alter the backbone. While CoupledMoves does not quantitatively rank binding affinities, it is a powerful tool for generating combinatorial libraries for screening, as it has been successfully employed to design a virtual library of mutants for engineering enzyme specificity (Ashworth et al., 2022).

In the thesis, CoupledMoves was used to explore and optimize the binder and the peptide interface interactions. CoupledMoves (version 57576) with KIC as backbone mover was used within Rosetta version 3.12. As an initial scaffold, binding pocket residues in the 6SA8 scaffold were mutated to Tyr-binder pocket residues based on the shared structure between pTyr and Tyr and similar interactions with similar residues could be a reasonable starting point. The peptide position 6 was mutated to pTyr via ATLIGATOR-web, and the peptide positions 4 and 8 were mutated to alanines via PyMOL. In the resfile, all seven positions in a binding pocket were allowed to be mutatable to any amino acid and positions 322, 326, 445, 449, and 452 were set to be repacked and minimized in order to provide flexibility to pocket positions. The other parameters were kept to default. 400 independent runs with 5000 moves were performed, and a total of 400 conformations of designed sequences were obtained for each design. To analyze the results, the *analyze_coupled_moves.py* script provided by the CoupledMoves GitHub tutorial

was run, which includes the sequence-logo Python module in order to extract the most frequently found mutations for the positions given. The script compares the distributions of output sequences, which are mutations enriched in the mutated structure over the wild-type crystal structure. Each logo consists of stacks of symbols, one stack for each position in the sequence. The overall height of the stack indicates the sequence conservation at that position. Additionally, the height of symbols within the stack indicates the relative frequency of each amino or nucleic acid at that position. The width of the stack is proportional to the fraction of valid symbols in that position. CoupledMoves was only used for pTyr binding pocket design.

3.2.3 Flex ddG

Flex ddG is a method developed within the Rosetta macromolecular modeling suite to estimate changes in binding free energy (ΔΔG) caused by point mutations at protein-protein or protein-peptide interfaces (Barlow et al., 2018). By incorporating backbone flexibility into the modeling process, flex ddG improves upon traditional rigid-body approaches, allowing for a more accurate representation of the conformational changes due to mutations. The method generates an ensemble of models by using the backrub protocol, which samples conformational changes around the specified mutation site, and calculates the average ΔΔG over this ensemble. The backrub protocol uses torsion angle minimization and side chain repacking, specifically targeting local backbone motions (Davis, Arendall, Richardson, & Richardson, 2006; Friedland, Linares, Smith, & Kortemme, 2008). This allows local conformational changes in protein structures, which has been shown to enhance the accuracy of stability predictions (Eccleston, Manko, Campino, Clark, & Furnham, 2023). It uses the Rosetta semi-empirical energy function and utilizes physical energies for the prediction with an optional generalized additive model (GAM) approach (Barlow et al., 2018). While it can be computationally intensive depending on the settings, flex ddG is particularly effective at capturing the nuanced effects of backbone and side chain adjustments, especially around the mutation site.

The flex ddG workflow begins with an initial structure undergoing global minimization and sampling of the backbone using the backrub approach before mutations are introduced. This is followed by packing of side chains on both wild-type and mutant models which both are minimized by Monte Carlo sampling of both backbone and side chain conformations. Then both models are scored via Rosetta's energy function, and the free energy difference (ΔΔG) between the wild-type and mutant protein is calculated (Equation 1) via:

$$\Delta\Delta G = \Delta G_{mut} - \Delta G_{WT} \qquad \text{Equation 1}$$

where ΔG_{WT} is the wild-type free energy and ΔG_{mut} is the mutant free energy; therefore negative values indicate better binding. The obtained flex ddG scores represent binding affinity changes in kcal·mol^{-1} relative to the respective alanine reference. Since relative ΔGs are calculated, it is not possible to draw a conclusion on the absolute binding affinity. But calculating the energy change allows to evaluate the stability of different amino acids binding to the respective binding pocket and thus allows to conclude about specificity.

In this chapter, flex ddG is used to evaluate designed ArmRPs in terms of specificity to different peptide variants. The 6SA8 structure with a bound $(KR)_5$ peptide was retrieved from the PDB database as a scaffold. Pocket residues were mutated to target specific amino acids based on previous studies. The peptide position 6 was first mutated from Arg to Ala to mitigate potential structural biases resulting from the initial positioning. Additionally, the flanking positions, peptide positions 4 and 8 were also mutated from Arg to Ala to prevent any interactions between these positions and the binding pocket of position 6. Structures were prepared in two ways; either all mutations including binding pocket and peptide mutations were applied using PyMOL's molecular sculpting function, which returns local atomic geometries, or designed structures were

downloaded from ATLIGATOR-web after binding pocket and peptide positions were mutated and relaxed.

The flex ddG protocol, as defined in Barlow et al., 2018, was modified for this study. The protein-peptide complex was placed in the *input* folder, and the chain IDs to be mutated were specified in the *chains_to_move.txt* file located within the same folder. The protocol *ddG-backrub.xml*, provided on the flex ddG GitHub page, was adapted to support the three modified amino acids. The XML file was written using RosettaScripts which is a scripting language interface for creating custom Rosetta protocols. The mutations were applied through the specified line in the modified XML file (see below), with the rest of the script remaining unchanged.

```
<MutateResidue name="mutate" target="6B" new_res="TYR:phosphorylated"/>
<MutateResidue name="mutate" target="6B" new_res="SER:phosphorylated"/>
<MutateResidue name="mutate" target="6B" new_res="THR:phosphorylated"/>
```

The parameters for the protocol were set to the recommended values. An ensemble of 250 output structures was generated, as this was decided as sufficient ensemble size to produce robust results for the system. For each structure, 35,000 Monte Carlo backrub steps were performed, with snapshots being saved every 7,000 steps, resulting in five final structures. The *analyze_flex_ddG.py* file, from the same GitHub page was used to analyze the results which returned the wild type and mutant interface dG. ddG score and the ddG score reweighted with the fitted GAM model were calculated and saved in a csv file. The ranking of the amino acids was calculated using the *calculate_ranking.py* script. The generated ensembles were extracted with *extract_structures.py* and visualized using PyMOL.

3.2.4 Molecular Dynamics (MD) Simulations

MD simulation is a computational method used to study the behavior of atoms and molecules at the molecular level over time. By capturing the motion of individual atoms and molecules within a system, MD simulations provide insights into the dynamic properties of complex biological structures such as proteins, enzymes, nucleic acids, and membranes (Zhang et al., 2009, Zhao and Caflisch 2015, Perez et al., 2016, Hollingsworth and Dror, 2018). They are particularly valuable in rational drug design to predict how drug molecules might bind to target proteins, evaluate drug stability, and assess drug-protein interactions. Additionally, MD simulations has played an important role in protein design, especially for protein-protein interfaces, such as in antibody-antigen complexes (Childers et al., 2017, Kralj et al., 2021).

MD simulations trace their origins to the 1950s, with inspiration by Monte Carlo simulations. The foundational principles were established by Alder and Wainwright in 1957, building on earlier work by Metropolis et al. 1953 and Rosenbluth and Rosenbluth 1955. Since the first application by Karplus and coworkers (McCammon, Gelin, and Karplus, 1977) by using empirical energy function, MD simulations have become a sophisticated and practical tool for studying dynamics and energetics of biomacromolecules, especially proteins (Dror, Dirks, Grossman, Xu, and Shaw, 2012; Karplus and McCammon, 2002). The primary objective is to gain a deeper understanding of molecular behavior by simulating and analyzing complex systems that would be difficult or impractical to study experimentally, thereby revealing key aspects of their dynamic properties and/or behavior. To simulate molecular systems, MD simulations calculate the potential energy of the system by integrating Newton's second law of motion. This process involves calculating the acceleration of each atom based on the force acting on it:

$$\mathrm{F} = \mathrm{ma} \qquad \text{Equation 2}$$

where *F* is the external force acting on the particle, *m* is the mass, and *a* is the acceleration. Using these forces, the acceleration of each atom in the system is calculated. By integrating the equations of motion, MD simulations generate a trajectory that describes the positions, velocities and accelerations of the particles over time. This trajectory allows for the calculation of average properties and the analysis of dynamic behaviors. Knowing the positions and velocities of atoms at any given moment enables the prediction of the system's state at any point in time. In MD simulations, the potential energy of a system at the atomistic level is calculated using a force field, which is a computational model that describes the forces between atoms or collections of atoms within molecules or between molecules. Force fields can be derived from experimental data, quantum mechanics calculations, or combination of both. While the concept is rooted in classical physics, force fields in chemistry are specifically parameterized to describe the energy landscape at the atomic scale.

Force fields include two classes of interactions: Bonded interactions within molecules and non-bonded interactions between molecules. Several popular force fields include, CHARMM (Chemistry at HARward Macromolecular Mechanics) (MacKerell et al., 1998), AMBER (Assisted Model Building with Energy Refinement) (Bayly et al., 1995), GROMOS (GROningen MOlecular Simulation) (Oostenbrink, Villa, Mark, and Van Gunsteren, 2004), and OPLS (Optimized Potentials for Liquid Simulations) (Jorgensen, Maxwell, and Tirado-Rives, 1996). These force fields differ in their energy functions and how they were parameterized, yet they all calculate the total energy of a system by summing bonded and non-bonded potential energies as functions of atomic coordinates. The accuracy of an MD simulation depends heavily on the choice of force field, as it determines how forces and energies are modeled, and a force field should accurately reproduce the behavior of the system being simulated. A typical MD simulation pipeline is shown in Figure 9.

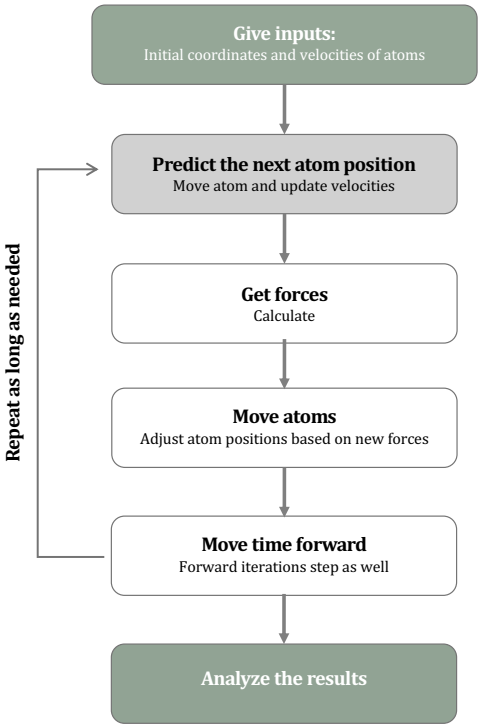

Figure 9: A common flowchart of MD simulation.

One of the drawbacks of MD simulations is that they are computationally demanding, requiring significant resources to capture the motion of large biological systems over biologically relevant time scales. Many biological processes, such as protein folding, ligand binding, or conformational changes, occur on milliseconds to seconds time scales making them difficult to simulate directly. Challenges also include modeling large systems, force field inaccuracies, particularly for complex cases such as phosphorylated amino acids. Poor sampling of rare events and long-range interactions can also lead to incomplete or biased results. However, advances in GPU acceleration and supercomputers that allow microsecond-scale MD simulations for relatively large protein systems can be performed in just a few days, a process that would have been unimaginable just a few years ago. Enhanced sampling techniques, and improved force fields, including polarizable models, are helping overcome these challenges, making MD simulations more accessible and accurate.

In this thesis, MD simulations were performed as an additional analysis method to observe interactions between a promising binder obtained from previous methods with peptides containing pSer and pTYR at the peptide position 6. For this purpose, nanoscale molecular dynamics (NAMD) software was used to run simulations together with analyzing methods such as root-mean-square deviation, root-mean-square fluctuation, and protein-ligand interaction analysis. The initial structure of the complex was obtained from the PDB database (PDB ID: 6SA8) and ATLIGATOR-web was used to mutate peptide position 6 from Arg to initially pSer and pTyr. PyMOL was used to mutate peptide position 6 from Arg to Trp. For the simulations, a structure with the bound peptide was used. Simulation systems were built with CHARMM-GUI and were solvated with a rectangular box of TIP3P water of 15.0 Å and neutralized with 0.15 millimoles per liter of sodium chloride and appropriate counter ions were added to neutralize the overall charge of the system. 0.15 M NaCl ions were added additionally. The CHARMM36 force field was employed to describe the interactions within the system. The system was minimized for 10000 steps with the conjugate gradient algorithm and equilibrated for 4 ns at a temperature of 310 Kelvin by using a 2fs integration time-step. Three independent 30 ns production runs were performed. Each simulation was run with periodic boundary conditions, with a temperature of 310 degrees Kelvin and a pressure of 1 atmosphere (NPT). Langevin dynamics was chosen as the control method to maintain a constant temperature (Chandrasekhar, 1943). The SHAKE algorithm was implemented to confine bonds involving hydrogen atoms in the confined state (Andersen, 1983). Descriptions of the simulations are given in Table 1.

Table 1: List of performed simulations with different systems. The binder sequence lists the residues in the side chain binding pocket 6 (see Figure 5).

Binder Sequence	Peptide Variants	Simulations
LKMKARQ , LKFKARQ	KRKRK(pTyr)KRKR	3x30 ns per system
LKMKARQ	KRKRK(pSer)KRKR	
	KRKRK(W)KRKR	
WT-binder	KRKRK(R)KRKR	

For analysis, first, the 30 ns long trajectories were first aligned to their initial frames and RMSDs (root mean square deviation) was then calculated to monitor structural stability. RMSD provides a numerical measure of the average distance between the corresponding atoms of two structures and it is widely used to assess the stability of simulated biomolecular systems. To visualize the simulations, Visual Molecular Dynamics (VMD) was employed. To evaluate the interactions between the binding pocket and peptide residues, ProLIF (Protein-Ligand Interaction Fingerprints) tool with a particular focus on analyzing contacts at position 6 was used. ProLIF is a Python based library that generates interaction fingerprints over the trajectory, which can then be analyzed to identify key residues and interactions. These interactions can be for example hydrogen bonds, hydrophobic contacts, or π-stacking.

3.3 Generated Libraries for Phosphorylated Amino Acids

3.3.1 pTyr Binding Pocket Suggestion

ATLIGATOR and ATLIGATOR-web

In order to investigate amino acids and amino acid groups in which any of the three phosphorylated amino acids; pTyr, pSer, and pThr, commonly interact in nature, ATLIGATOR and ATLIGATOR-web were used as the first step of the design pipeline (Figure 7). All crystal structures that include pTyr in their sequences, independent of them being protein or peptide, were searched in the PDB database, and a structure collection via the ATLIGATOR-web interface was created. Due to the limited number of structures for phosphorylated amino acids deposited in the PDB, not only the intermolecular interactions but also intramolecular interactions were included in the search for pTyr's interaction partners. In total, 697 structures that belong to 13 SCOPe families were found and included in the structure collection (Figure 10-A). Following the ATLIGATOR-web pipeline, pairwise interaction patterns between amino acid residues were computed and stored in a database on the web. For the 697 structures, pTyr is found to have a total of 8581 data points, which means 8581 interactions with other amino acids. The most frequently found amino acids among these interactions belong to Arg and Lys, with 1700 and 1133 data points, respectively. Following these two amino acids, Ser and His interact with pTyr with 925 and 666 data points, respectively (Figure 10-B). Thr and Ala were the following amino acids in the list with fewer interactions. Binding pockets based on these data points which were stored in the atlas, were created for pTyr. Then, with these data points, the most frequently found pockets were generated and are accessible in the *pockets* section. The most frequently found pocket for pTyr, was ''LKF'' with making up 13% of all pockets, followed by ''RRY'' and ''KKFS'' pockets with 11% and 4% respectively. These pockets do not necessary cover all the binding pocket residues (Figure 5, 7 binding pocket residues), but make up motifs that can be used in the design step. All these suggested pockets included Arg or Lys residues (Figure 10-C).

Finally, under the *design* section, several designs were tested based on interaction patterns with the 6SA8 scaffold. For each mutation or mutation combination, minimization was carried out using Rosetta minimization in the background of ATLIGATOR-web. During several design trials, the design options manual design and pocket grafting were used either separately or together. In any of the cases, pTyr was never positioned deeper between the helices 2 and 3, but it rather stayed on the surface as shown in Figure 10-D.

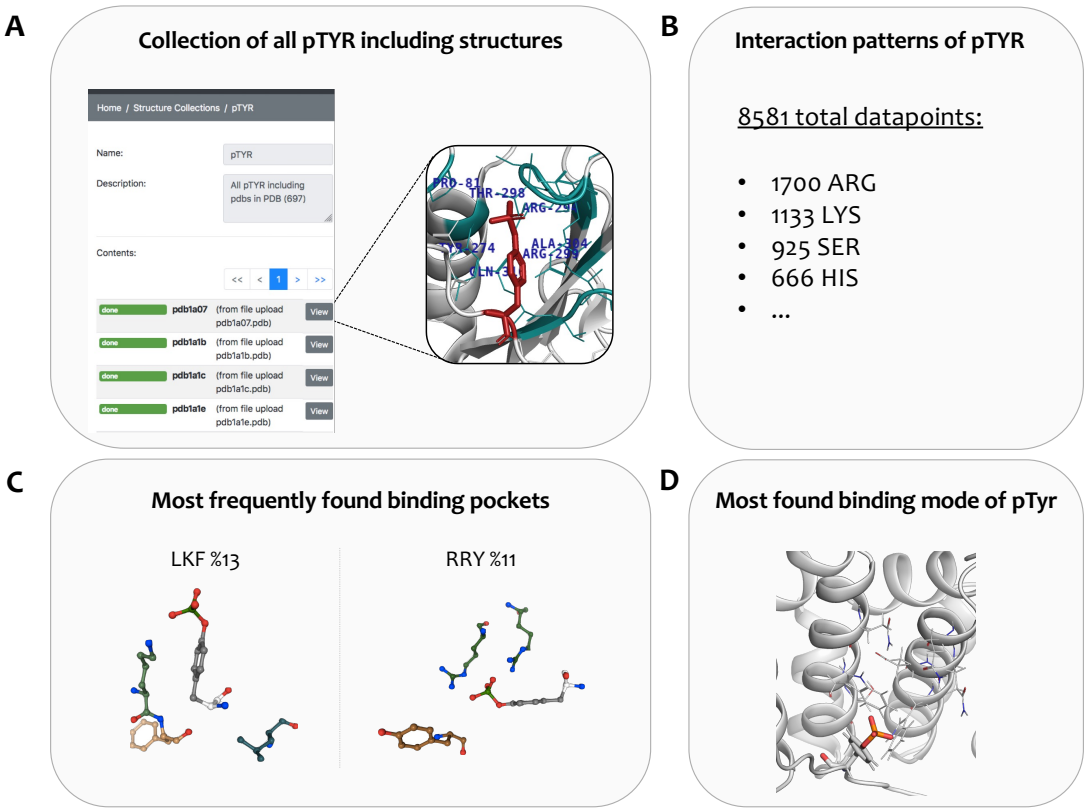

Figure 10: The results of the pipeline followed in ATLIGATOR-web. (A) Structure collection with all pTyr including structures were created with focus on one randomly selected structure. (B) Interaction patters of pTyr stored in atlas are listed. (C) Two of the most commonly found binding pockets for pTyr are given. (D) Binding modes of several designed binding pockets with pTyr in the peptide are shown.

Based on the interaction partners, discovered pockets and their placement on 6SA8, ATLIGATOR and ATLIGATOR-web analysis ended up with the following suggestions: For position 2 in the binding pocket, positively charged amino acids; Arg and Lys were found to be preferable particularly in the context of having positively amino acid in close proximity to pTyr in the peptide. In addition, during design trials using the pocket grafting option in ATLIGATOR-web, the discovered pockets were consistently grafted onto scaffold 6SA8 at pocket positions 2, 3 and 5. These positions are not only in closer proximity to pTyr but also highlight the potential importance of these sites. This observation was further explored in subsequent steps for designing a pTyr-binding pocket library.

CoupledMoves

Based on the results obtained from ATLIGATOR, the CoupledMoves algorithm was run to redesign the binding pocket of 6SA8 to accommodate a pTyr residue. Specifically, Arg at peptide position 6 was mutated to pTyr, binding pocket positions were mutated from wild type residues to Tyr-binder residues. Prior to running the algorithm, initial mutations were modeled in PyMOL. A total of 400 designed sequences were generated, from which a sequence logo was constructed to visualize the probability distribution of amino acids at each position within the binding pocket. Position 2 exhibited high conservation, predominantly favoring Lys, which appeared as the most energetically favorable amino acid, underscoring its critical role in binding specificity. Smaller amino acids such as Ala, followed by Thr, also showed moderate enrichment, potentially contributing additional space for accommodating pTyr at position 3. Position 7 showed a slight preference for Arg, consolidating the importance of a second positively charged residue in the binding pocket to complement Lys at position 2. Position 1 and position 4 displayed variability, likely due to their distance from the peptide and not having direct interactions with pTyr. This variability suggests that residues at these positions may have less impact on binding specificity and a meaningful conclusion for these positions is more challenging (Figure 11-A).

In addition to sequence logo analysis, generated structures were analyzed via PyMol and notable, no structures were generated, in which pTyr was located deeper between the helices 2 and 3. Instead, pTyr appeared to interact more on the surface, which is an agreement with the ATLIGATOR-web results (Figure 11-B).

Additionallly, flanking residues 4 and 8 were also mutated to valines intead of alanines, to evaluate their influence (Figure 11-C). However, no significant difference could have been observed in generated sequence logos or generated structures.

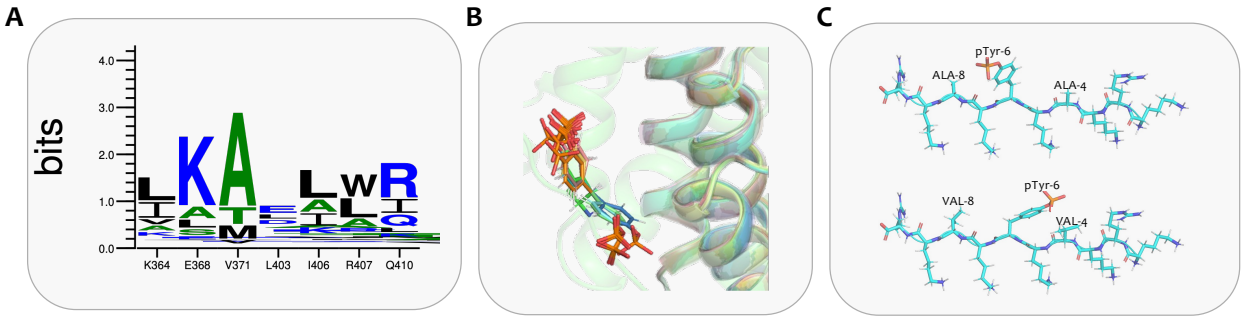

Figure 11: CoupledMoves analysis for pTyr binding pocket design. (A) A Sequence logo is created for analysis of results. X-axis shows the seven position in the binding pocket with initial amino acids. (B) Representative structures out of generated structures visualized in PyMOL. (C) Peptide sequences with different flanking residues are shown, where the upper one containes alanines and the lower one containes valines, depicted as sticks.

Flex ddG

Building upon the binding pocket residues identified by CoupledMoves, the LKFKARQ residues within the binding pocket of the 6SA8 structure was selected for further evaluation using the flex ddG protocol to assess its specificity for pTyr compared to other amino acids. To prepare the system for these calculations, the pocket residues were mutated to LKFKARQ, and peptide positions 4, 6, and 8 were mutated to alanine in PyMOL. The flex ddG protocol was then used to perform single point mutations at peptide position 6, internally mutating it to all 20 amino acids,

including pTyr from the initial residue of Ala residue. Reweighted ΔΔG scores, fitted to the Generalized Additive Model (GAM), were computed and ranked to evaluate the impact of each mutation on binding affinity within the protein-peptide complex. Mutants with ΔΔG scores lower than 0 kcal·mol^{-1} were considered potential candidates for improving binding affinity. However, pTyr was ranked among the least favorable target residues, with a positive ddG score, alongside other large amino acids such as Lys and Arg, for this particular binding pocket (Table 2). This suggests that, despite its intended specificity, the current binding pocket might be too compact to accommodate larger residues like pTyr, and more spacious binding pockets might be a more reasonable alternative. Despite the unfavorable ranking of the pTyr, other negatively charged amino acids, Glu and Asp, were ranked fifth and sixth, respectively. This may indicate that while the pocket may tolerate certain negatively charged residues, the specific structural or interaction properties of the target amino acid make it less favorable for binding. The high ranking of Trp, Phe and His could be explained by the potential for π-stacking interactions between the aromatic rings of these residues and the Phe at position 3 of the binding pocket, along with Rosetta's general preference for aromatic residues (Ayyildiz et al., 2024), which may contribute to their lower ΔΔG values.

Table 2: Calculated ddG values of peptide variants with the binder LKFKARQ is ranked the most favorable to the least mutation.

Residue Names	Flex ddG Scores (kcal/mol)
W	-2.02
F	-1.08
H	-0.95
L	-0.81
E	-0.78
D	-0.43
V	-0.13
Q	-0.09
N	0.01
Y	0.04
I	0.09
C	0.43
T	0.50
A	0.64
G	0.74
S	1.26
P	2.15
K	2.65
pTyr	2.72
R	3.24
M	4.93

For further evaluation, the top-ranking mutations (Trp, Phe,His) and the pTyr mutation were visualized in PyMOL. The aromatic residues indeed formed stabilizing π-stacking interactions with the binding pocket, supporting their favorable ranking. For pTyr, several structures were found between the helices 2 and 3 unlike generated structures via ATLIGATOR and CoupledMoves. However, these structures were very rare and the distance between these helices was observed to be relatively small for larger amino acids such as pTyr. Therefore, a distance between residues located in position 2 and 6 was measured in one generated model as of 4.8 Å. This proximity likely contributes to steric clashes between side chains, which could explain the unfavorable ΔΔG for pTyr (Figure 12-A). Additionally, multiple structures of pTyr were found positioned further away from the pocket, emphasizing the challenges of fitting this residue within the compact binding pocket (Figure 12-B).

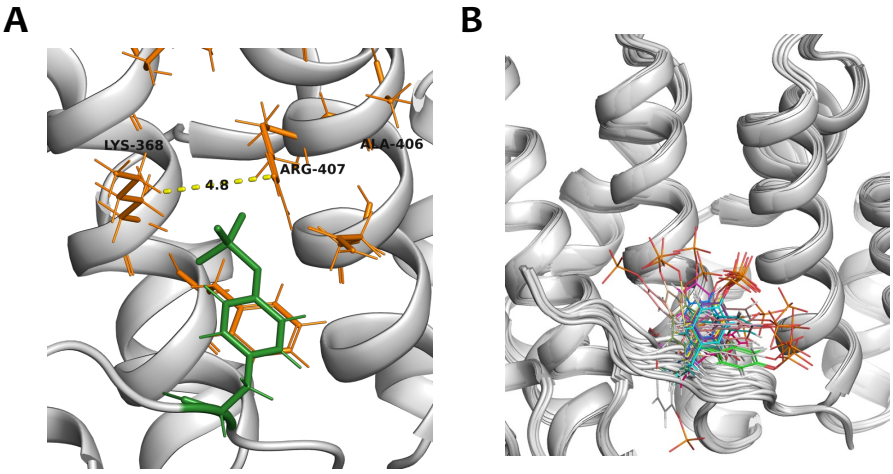

Figure 12: Structures of Flex ddG calculation of the binder LKFKARQ with pTyr in the peptide. (A) The binding pocket residues and pTyr are shown in orange and green sticks respectively. Distance between Pos2 (Lys-368) and Pos6 (Arg-407) is shown as Å in yellow dash. (B) Some of the pTyr ensembles created by flex ddG are shown in sticks.

Flex ddG was run on a total of 30 different binder-peptide complexes, each having a binding pocket with variations in one or two positions, such as LKFKRAQ (which gives more space by placing a small amino acid at position 6) or LKEKRAQ (for testing one negatively charged amino acid in the binding pocket). Although none of the calculated sequences resulted in a negative ddG score for pTyr, the analysis provided valuable insights into key positions in the binding pocket and the amino acids that influence ddG values the most. These findings highlight the importance of certain residues and their impact on the specificity of the complex, laying the groundwork for library suggestion.

Library Suggestion

To optimize pTyr binding, two distinct libraries, each with a size of 1.8×10^6 were suggested for further processing (Table 3). These libraries differ in the amino acid selection at the 5^{th} and 6^{th} position. Based on insights from ATLIGATOR, ATLIGATOR-web and CoupledMoves, position 2 was suggested to include a positively charged residue, Arg or Lys or partially positively charged residue such as His. Results from CoupledMoves were used as an input sequence for flex ddG calculations. Based on flex ddG predictions, positions 2, 3, 5 and 6 were identified as the key positions and having the main influence for the binding mode of pTyr. For the positions 5 and 6, to help avoiding an overrepresentation of positive charged residues for balanced electrostatic environment, only one of them were suggested to include positively charged residue.

For Library 1, position 5 was suggested to include Arg, Lys, and His. However, this selection may reduce the available space within the pocket, potentially making it too small for optimal pTyr binding. Therefore, this library was designed with the intention of positioning pTyr on the surface, as observed in the structures generated by ATLIGATOR-web and CoupledMoves. In contrast, Library 2 included Arg, Lys or His at position 6, allowing for a larger binding space for pTyr, therefore allowing pTyr possibly to be between the two helices. This design aimed to provide sufficient room for pTyr to fit inside the pocket. Besides these positions, position 3 was specifically avoided for having one of the large amino acids, considering once more the size of

pTyr. Additionally, one of the charged amino acids based on not having too many charged residues in the pocket was also avoided for this position. Across all methods, pTyr consistently did not fit well between helices 2 and 3, reinforcing the need for a surface-binding. As such, Library 1 was designed with the assumption that pTyr would adopt a standing-up conformation and interact with the surface of the pocket, while Library 2 was designed to allow pTyr to potentially bind deeper inside the pocket (Table 3).

Table 3: Suggested libraries for pTyr binder. Total number of amino acids for that position is written after amino acids, with "All" meaning 20 conanical amino acids.

Positions	**Library 1 (size: 1.8 x 10^6)**	**Library 2 (size: 1.8 x 10^6)**
1	All except CPG | 17	All except CPG | 17
2	KRH | 3	KRH | 3
3	FLIVASTMHN | 10	FLIVASTMHN | 10
4	All except CPG | 17	All except CPG | 17
5	KRH | 3	AVSTN | 5
6	AVSTN | 5	KRH | 3
7	all except CPGHKR | 14	all except CPGHKR | 14

3.3.2 pSer and pThr Binding Pockets Suggestions

Insights from pTyr binding pocket design

The promising binder sequence for pTyr, **LKFKARQ**, was selected to test also for pSer binding, considering their shared negatively charged phosphate group, which may facilitate interactions with similar amino acids. This binding pocket was chosen from the Library 1, where position 6 of the binding pocket was designed to favor large and positively charged amino acid. In this context, Arg was selected, given pSer's smaller size and its potential to fit between the helices 2 and 3, and forming favorable interactions with Arg in the binding pocket. Flex ddG calculations were performed on the complex, and the resulting ddG values are given in Table 2 and Table 4. The analysis ranked pSer as the second best choice for this pocket, following Trp in the peptide position at 6 (Table 4). The preference for Trp, despite its size, suggests that its side chain may form specific stabilizing interactions, such as π-stacking with nearby residues. To improve pSer binding over Trp, pocket position 3 was mutated from Phe to Met. This mutation worsened the binding for Trp while slightly improving it for pSer. Despite this change, Trp remained the top-ranked amino acid for this binder (Table 5). The results support the hypothesis that this pocket is promising for establishing binding specificity, particularly for small residues like pSer. The observed improvements in pSer ranking after the Phe-to-Met mutation further suggests that optimizing pocket composition can selectively enhance affinity for desired residues.

Table 4: Calculated ddG values of peptide variants with the binder LKFKARQ. pSer in the peptide position 6 was included in the calculation in addition to previous calculated amino acids.

Residue Names	Flex ddG Scores (kcal/mol)
W	-2.02
pSer	-1.18
F	-1.08
H	-0.95
L	-0.81
...	-0.78
pTyr	2.72
R	3.24
M	4.93

Table 5:Calculated ddG values for some peptide variants with the binder LKMKARQ including pSer.

Residue Names	Flex ddG Scores (kcal/mol)
W	-1.6
pSer	-1.3
F	-0.1
H	-0.1
pTyr	1.47

Molecular Dynamics Simulations

Since the pocket with *LKMKARQ* showed promising results for pSer based on flex ddG predictions, short MD simulations were performed out to evaluate the interaction patterns throughout the simulation. Three independent 30 ns simulations were carried out for both peptides containing eitherpSer or Trp in the position 6 with the pocket *LKMKARQ***.** Simulations were conducted always as protein-peptide complexes. Interaction analyses were conducted using ProLIF (called as fingerprint analysis) as an average of 3 independent simulations to quantify and compare the interactions between the binding pocket and the mutated peptide residues. In Table 2.3, the ligand column indicates the peptide sequence where position 6 was mutated to either pSer or Trp (pSer6 and Trp6). The interaction column lists the residues in the pocket that interacted with the ligand for at least 0.05% of the simulation time. The right column provides the frequency of each interaction as a percentage of all simulation frames.

As expected, Asn372, which plays a critical role in the fixation of the backbone by interacting with the peptide (see section 1.2), showed interactions with the peptide in over 99% of the simulation frames, regardless of the mutation at position 2. In addition to Asn372, Arg407 emerged as a significant interacting residue in both cases. However, it interacted with pSer in 94% of the simulation frames, compared to 65% for Trp. This difference in interaction frequency highlights the potential for enhanced specificity toward pSer. In addition to Arg407, interactions with Lys2 were observed in both Ser and Trp simulations, with Ser interacting with Lys2 in 75% of the simulation frames, compared to 87% for Trp, which is favored for Trp binding (Table 6, Table 7). Overall, the MD simulations confirm the flex ddG results, emphasizing the strong interaction between pSer and Arg407 as a key determinant of binding specificity.

Table 6: Interaction analysis of binder LKMKARQ with pSer mutation in the peptide.

Ligand	Interacting Residue	Frequency (%)
pSer6	ASN372	0.99
	ARG407	0.94
	MET371	0.90
	LYS368	0.75
	GLN410	0.05

Table 7: Interaction analysis of binder LKMKARQ with Trp mutation in the peptide.

Ligand	Interacting Residue	Frequency (%)
pTrp6	ASN372	0.99
	ARG407	0.66
	MET371	0.90
	LYS368	0.87
	GLN410	---

To provide a baseline for comparison, extended MD simulations of 200 ns were performed on the wild-type protein-peptide complex. These simulations were repeated three times, yielding consistent and robust results. The WT complex demonstrated remarkable stability throughout the simulations, with arginines and lysines maintaining their positions within the binding pocket from the initial frame onward. RMSD plots for these simulations were highly stable, indicating no significant structural deviations or residue dissociations. For pTyr binding, several binding pockets

including different mutations and with the bound peptide including pTyr in peptide position 6 were also simulated. Simulations for each protein-peptide complexes were performed once for 30 ns. Despite the selection of initial structure with pTyr inside the helices 2 and 3, the simulations showed that pTyr consistently dissociated from the binding pocket within the first nanoseconds. These results are consistent with the previous methods suggesting that pTyr is binding in this conformation. However, no additional simulations were performed for pTyr.

Library Suggestion

Based on the results from the previous analysis explained above, a single library with the size of 4.5×10^6 was suggested for further evaluation (Table 8). While this library shares similarities with those previously suggested for pTyr except the positions 2 and 5. pSer and pThr are similar in size and structure; both are uncharged polar amino acids with a hydroxyl (-OH) group, with pThr having an additional methyl group. Considering this similarity, the same library should be tested for pSer and pThr. This increases the chances to find binders with a high affinity to either pSer or pThr, and specificity between these two amino acids could be investigated in later stages.

Table 8: Suggested library for pSer binder. Total number of amino acids for that position is written after amino acids.

Positions	**Library (size: 4.5×10^6)**
1	All except CPG | 17
2	KRHTSHAQND | 10
3	MLINQD | 6
4	All except CPG | 17
5	AVSTNDIL | 8

6	KRH | 3
7	all except CPGHKRFYW | 11

4. Computational Evaluation of Peptide Binding Specificities

4.1 The Challenge of Computing Single Residue Effects in Protein-Peptide Interfaces

Proteins are essential molecular machines that perform diverse cellular functions, often through specific interactions with other molecules (see section 1.1 and section 3). The individual interactions between amino acids in binding interfaces of protein and binding partners mostly determine the strength of the binding and its specificity towards its target/each other. Mutations in these residues can significantly alter binding affinity, affecting the protein's activity, stability and function by inducing changes in the protein's conformation. Even a single amino acid change can alter the binding affinity, either weakening it for one target or enhancing it for another, thus altering specificity.

Understanding and modifying protein–target interactions is the key to design of binding proteins to target other proteins or peptides. To assess how mutations affect binding affinity in protein-protein complexes, reliable 3D structures are often needed, and experimental methods like Surface Plasmon Resonance (SPR) and Isothermal Titration Calorimetry (ITC) provide valuable insights, although they can be costly and resource-intense. Computational pre-evaluations can help narrow down candidates for experimental testing, using methods like binding free energy calculations to predict mutation impacts. Binding free energy (ΔG) measures the stability and strength of a protein complex, with lower (more negative) values indicating stronger interactions. Accurate modeling of bound and unbound states is essential, and methods like thermodynamic integration, molecular dynamics simulations, or quantum mechanics approaches often correlate well with experimental results, despite being time-intensive. Recent advances in machine learning and deep learning offer promising enhancements in both speed and accuracy for binding free energy predictions. For example, neural networks are being used to estimate binding affinity

from molecular structures, though challenges remain in seamlessly integrating these models into workflows (Bogdanova & Novoseletsky, 2024; Guo & Yamaguchi, 2022). ML methods can require large datasets and may be sensitive to specific features or parameters, limiting their generalizability across systems. Additionally, while most approaches predict binding versus non-binding outcomes, estimating the effects of single mutations remains particularly challenging.

In this section, conceptually different computational methods to address the challenge of accurate prediction of single residue effects in protein-protein interfaces were evaluated. Here, three established, physics-based approaches, namely flex ddG from Rosetta Design Suite, BBK* from OSPREY and PocketOptimizer developed in our group, were compared to highlight their strengts and weaknesses via using a novel dataset of single point mutations. On the one hand, flex ddG calculates the change in binding affinity (ddG) upon mutation, and creates diverse ensembles with the backrub approach (see section 3.2). It offers accurate and reliable predictions by allowing for local backbone flexibility and utilizing a sophisticated energy function. On the other hand, OSPREY employs deterministic algorithms to guarantee finding the global minimum energy conformation in a discrete conformational space. The K* algorithm within OSPREY optimizes protein sequence and structure simultaneously, evaluating both the bound and unbound states of a protein-ligand complex and approximating the partition function for direct calculation of binding affinities. In addition to these methods, PocketOptimizer, an *in-house* tool is included. PocketOptimizer generates an ensemble of the remodeled bound state and determines the energetically best combination of sidechain rotamers and the ligand conformation and position in the binding pocket.

4.2 Available Computational Design Tools

4.2.1 OSPREY

The OSPREY (Open-Source Protein REdesign for You) suite is a comprehensive computational platform for protein engineering and redesign (Hallen et al., 2018; A. Ojewole et al., 2017). Its primary aim is to identify protein mutants with desired target properties, such as improved stability or altered binding affinity. It is one of the most commonly used protein design software, where application fields ranging from drug discovery and antibody design (Rudicell et al., 2014; Surpeta, Sequeiros-Borja, & Brezovsky, 2020).

OSPREY has been successfully applied to optimize protein small molecule interactions (Guerin, Kaserer, & Donald, 2022; Kaserer & Blagg, 2018) as well as to design peptide inhibitors of protein–protein interactions (Roberts, Cushing, Boisguerin, Madden, & Donald, 2012). Central to OSPREY's design capabilities is its ability to model protein flexibility and explore large conformational spaces, which is critical for predicting realistic structures as proteins naturally undergo conformational changes. This flexibility is represented through rotamers, which are discrete conformations of amino acid side chains. To overcome the limitations of traditional discrete rotamer modeling, OSPREY incorporates continuous rotamers, enabling a more accurate representation of side chain flexibility (Gainza, Roberts, & Donald, 2012; Georgiev, Lilien, & Donald, 2008). By evaluating the energetic favorability of these rotamers in different contexts, OSPREY constructs conformational ensembles, which represent multiple low-energy states instead of relying on a single global conformation. To optimize the search process, OSPREY initially narrows the search space by applying a range of algorithms derived from extensions of the dead-end elimination (DEE) technique. DEE systematically exludes rotamers that cannot contribute to the global minimum energy conformation (GMEC) or any low-energy state, even when accounting for backbone and side chain flexibility. Once the search space is narrowed, OSPREY applies a branch-and-bound algorithm, which efficiently explores the remaining

conformations to identify the GMEC, if desired, it provides a list of low energy structures. This algorithm, inspired by the A* search technique, is particularly effective in balancing computational cost with thorough exploration.

To further enhance its capabilities, OSPREY integrates the K* algorithm, which approximates protein–ligand binding constants based on an ensemble-based approach. Unlike methods that rely solely on the GMEC, K* evaluates multiple low-energy conformations, increasing the likelihood of identifying biologically relevant binding modes. The binding constant (K_a) derived from K* is represented as the ratio of partition functions for the bound and unbound states. This provides a robust metric for binding affinity, with the Log_{10} K* score serving as a predictive measure with higher values indicate stronger binding affinities, while lower values suggest weaker interactions. By comparing Log_{10} K* scores for the wild-type and mutated structures, OSPREY can assess the impact of specific mutations on binding affinity and stability.Building on K*, the BBK* (Branch-and-Bound K*) algorithm further optimizes the search for protein-ligand solutions by incorporating a branch-and-bound strategy that systematically explores conformational spaces. The BBK* algorithm is based on the approximation of the partition functions for the bound (protein-ligand complex) and unbound (free protein and ligand) states of a system (Table 9). The calculated K* scores for a protein-ligand complex are defined as the quotient of the bound and unbound partition function and were proven to exactly approach the binding affinity constant K_a under accurate conditions (Krismer et al., 2024; Lilien, Stevens, Anderson, & Donald, 2005). BBK* uses shortcuts to efficiently explore conformational space, skipping configurations that cannot contribute meaningfully to the solution.

Beyond its algorithm, OSPREY employs a comprehensive scoring function to evaluate the energetic impact of protein-ligand interactions. This scoring function considers van der Waals interactions, electrostatics, solvation effects, and hydrogen bonding, providing a detailed picture of the factors driving binding affinity.

In this study, the BBK* algorithm implemented in OSPREY3.2.304 (Hallen et al., 2018) was used for prediction calculations. After structure preparation (for details see end of the section 4.2), side chains of binding pocket residues were selected to be flexible with continuous flexibility given to inner shell residues (Figure 5, position 3, 4, 6, and 7). For models based on the crystal structure 6SA8, peptide position 6 was also given as continuous flexiblity position and it was mutated to alanines together with peptide position 4 and 8 as the initial amino acids and set to be mutated to other amino acids of interest. For models based on the crystal structure 5AEI structure, continuous flexibilty was applied to respective positions. For the data processing, the comparison of the predicted scores were restricted to the available experimental data. The obtained K* scores from BBK* were converted into approximate pK_D values. As an uncertainty range, the obtained upper and lower bounds of the K* score were taken (Ayyildiz et al., 2024).

4.2.2 Rosetta

As discussed in sections 3.3.2 and 3.3.3, the Rosetta Design Suite offers a comprehensive framework for computational protein design studies. Specifically, the flex ddG protocol within Rosetta calculates changes in binding affinity upon mutation via incorporating backrub motion and generating a diverse ensemble. For its energy function it uses a physics-based force field that includes also empirical terms (Table 9).

In this work, the flex ddG algorithm implemented in Rosetta 3.12 was used for prediction calculations. After structure preparation for each binder (for details see end of the section 4.2), a Python script was modified for calculations. For each model, 250 output structures were generated allowing backrub for 35000 steps. For models based on 6SA8, peptide position 6 was mutated to alanines together with peptide position 4 and 8 as an initial amino acids and set to be mutated to all other interested amino acids. Models based on the 5AEI structure, peptide position 4 was mutated to alanine together with peptide position 2 and 6 and position 4 was set be mutated to all other amino acids of interest. For modeling, the Talaris all-atom energy function

was used and the score analysis was performed as described using the corresponding reweighting scheme based on a generalized additive model. For data processing, the comparison of the predicted scores were restricted to the available experimental data. The obtained K* scores from BBK* were converted into approximate pK_D values.

4.2.3 PocketOptimizer

PocketOptimizer is an *in-house* computational tool designed for the targeted optimization of binding pockets to improve ligand binding affinity, specificity, and overall stability, making it particularly useful for protein engineering and drug discovery (Noske et al., 2023).

The software provides a modular framework, allowing users to combine different modules and guiding them step-by-step through the design process. Within this flexible setup, users can experiment with a variety of force fields, sampling procedures, and scoring functions to identify the most effective binding-pocket mutations. PocketOptimizer supports both the CHARMM36 and AMBER ff14SB force fields, enabling physics-based modeling of molecular interactions. To predict mutations that enhance binding affinity, PocketOptimizer generates an ensemble of the bound state and determines the optimal combination of side chain rotamers and ligand positions within the binding pocket. Using the Dunbrack rotamer library which is a statistical collection of preferred side chain conformations based on high-resolution protein structures, the software explores realistic configurations for each mutation. In addition, C.M. Lib backbone-independent rotamer library is also provided as an option. The scoring function then calculates binding energies in $kcal \cdot mol^{-1}$, with the energy function assessing mutations by determining their energetic impact on binding affinity. Through iterative calculations and energy assessments, PocketOptimizer selects mutations that yield the most favorable binding energy and it plays an essential role in the rational design of proteins for targeted applications.

In this section, PocketOptimizer 2.0 was used for prediction calculations with the Amber ff14SB force field (Noske et al., 2023). After structure preparation for each binder (for details see end of the section 4.2), all pocket positions were mutated to the respective pocket residues before the energy calculations in both crystal structure complexes (PDB-IDs: 6SA8 and 5AEI). Rotamers from the Dunbrack backbone-dependent rotamer library were selected for rotamer sampling for all pocket positions, and for the mutatable ligand position the C.M. Lib backbone-independent rotamer library was used. One combination of pocket rotamers and a single ligand pose was selected by PocketOptimizer that represent the GMEC (Table 9). No error is computed since there is only one complex structure.

Table 9: Summary of used algorithms. Run times are estimated times for dARmP protein-peptide complex.

Method	**Summary**	**Run time (for dArmRP system)**
Flex ddG (Rosetta)	Calculates the binding affinity change of the complex. It is backrub-based approach to generate diverse ensembles.	2-5 days
BBK* (OSPREY)	Approximation of the partition function for binding affinity estimation. It provides continuous flexibilty.	days to weeks
PocketOptimizer (*in-house*)	Identificatiın of GMEC for remodeled bound state. It has static rotamers.	hours

The methods described above were applied to a dArmRP system, as described in Ayyildiz et al., 2024. Two high-quality crystal structures of these proteins were used to compare computational predictions with experimental binding affinities. This approach aimed to assess the strengths and limitations of the methods, thereby not only evaluating the influence of structural differences on prediction but also providing critical insights to refine design strategies with greater precision. Based on the predictions of above mentioned methods, the benefit of a complementing combination of results from various computational sources will also be shortly discussed.

The evaluation included five experimentally validated designed binders, each including a distinct binding pocket; Arg-binder (WT-binder or Arg binding pocket) (Figure 5), Tyr-binder, Trp-binder, His-binder and Ile-binder (Stark et al., 2024). As most interactions at the protein-peptide interface remained constant, the focus was placed on a small variable interaction region. To generate peptide variants for analysis via the different methods, first peptide position 4, 6, 8 of the 6SA8 scaffold, or at the peptide position 2, 4, 6 of the 5AEI scaffold, were mutated from arginines to alanines.Before performing the calculations, structure preparation was conducted using two structures, PDB IDs 6SA8 and 5AEI, both of which were solved in complex with the $(KR)_5$ ligand and obtained from the Protein Data Bank. Both of these two scaffolds were used for calculations in order to observe the effect of structural differences on the results. The 6SA8 scaffold includes a dArmRP-fusion with a designed ankyrin repeat protein (DARPin), which was specifically engineered to protect the peptide-binding interface from crystal contacts and thus altered the binding complex geometry (see section 1.2). To reduce computational cost, the DARPin component was removed, along with all ions and water molecules, using MolekuleKit. The necessary mutations for the binders were introduced using PyMOL version 2.3, while protonation of the models was carried out using MoleculeKit, which uses PropKa 3.2.

4.3 Prediction Performance Tested on a dArmRP Benchmark

To evaluate the accuracy of the computational predictions, three key analyses were performed. First, the predicted binding specificities were compared with experimentally determined values using Pearson correlation, a statistical measure that quantifies the linear relationship between two variables, commonly used to assess predictive accuracy in computational modeling. Second, to examine method-specific biases toward different amino acid types, a relative bias was calculated for each target peptide based on the corresponding amino acid with the available data set. Third, computational methods were systematically compared to identify their complementary strengths and limitations, providing insights into their relative performance.

4.3.1 Evaluation of Predictive Methods for Pocket Specificity Analysis

In Figure 13, correlation plots of BBK*, flex ddG and PocketOptimizer for the Arg-binder using 6SA8 as a scaffold are given. In the experiments the Arg-binder demonstrates high specificity, with low nanomolar K_D values for the positively charged amino acids Arg and Lys, and micromolar K_D values for the negatively charged amino acids Asp and Glu. This highlights a clear distinction for Arg, except in comparison with Lys. However, K_D values for other amino acids cluster near the center, with partially overlapping error margins, which makes them hard to distinguish. The specificity prediction of BBK* shows the strongest overall agreement with the experimental data, achieving a Pearson's R correlation of 0.861. It accurately captures the distinction between positively charged residues and negatively charged residues, with a slight overprediction for His. All other amino acids cluster in the middle which reflects their close experimental values (Figure 13-A). In contrast, flex ddG shows the lowest correlation among the three methods with 0.317 as Pearson's R, although it captures the overall trend by accurately identifying between positively charged residues and negatively charged residues. Aromatic residues such as Trp and Phe, which were predicted to be bound more tightly than Arg, could be one reason for this lower correlation (Figure 13-B). Meanwhile, PocketOptimizer achieved a moderate correlation with 0.562

Pearson's R, and demonstrated a similar ability to distinguish positively charged amino acids from negatively charged ones, but with less accuracy than BBK* (Figure 13-C).

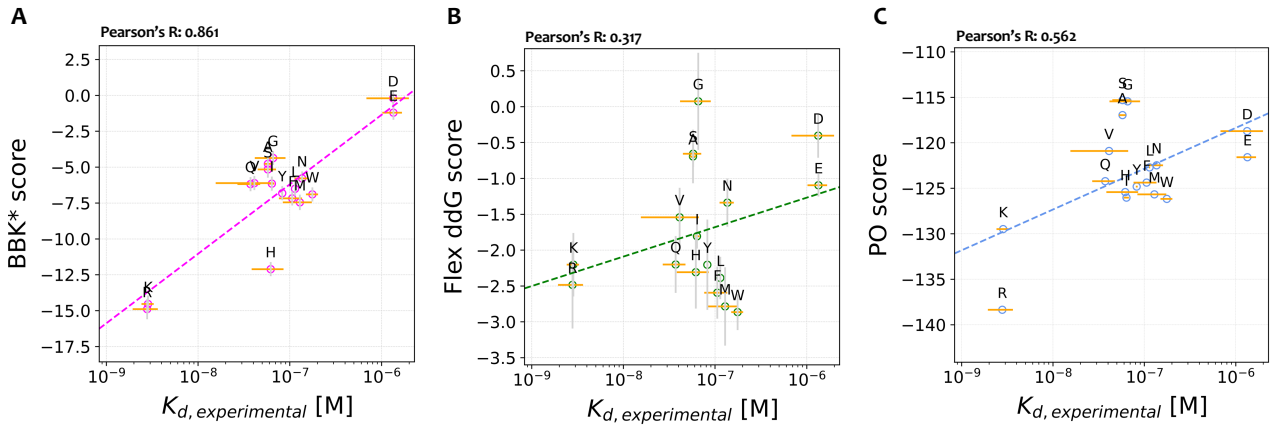

Figure 13: Correlation of calculated and experimentally determined binding specificities using crystal structure 6SA8. Specificity predictions compared to experimental data for the Arg-binder pocket (Linear fits shown in dashed lines). Binding specificity predictions from (A) BBK*, (B) flex ddG and (C) PocketOptimizer were correlated with experimentally determined binding specificities. Pearson correlations are given inside the corresponding plots (Adapted from Ayyildiz et al., 2024).

The same set of calculations was performed using the crystal structure 5AEI (Figure 14). The predictions obtained with BBK* and PocketOptimizer closely resembled those based on the scaffold 6SA8. While BBK* predictions remained consistent across these two scaffolds, the PocketOptimizer predictions exhibited a stronger emphasis on Phe and His when using the 5AEI structure. In the case of flex ddG, Pearson's R correlation was lower when using the 5AEI structure, likely due to the peptide variants containing positively and negatively charged amino acids were being effectively distinguished only when the 6SA8 structure was used (Figure 13). Additionally, small amino acids such as Gly, Ala, and Ser were generally predicted to have lower binding affinities compared to most other amino acids.

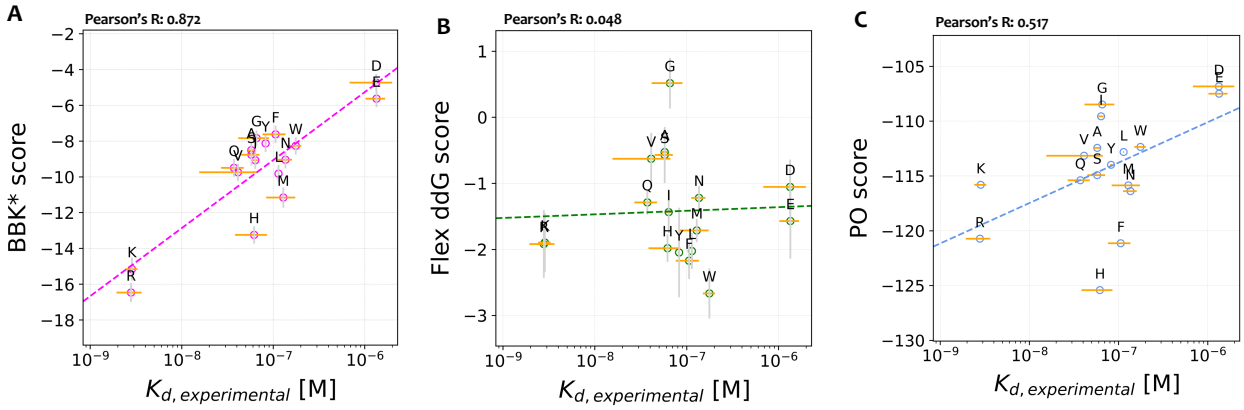

Figure 14: Correlation of calculated and experimentally determined binding specificities using crystal structure 5AEI. Specificity predictions compared to experimental data for the Arg-binder pocket (Linear fits shown in dashed lines). Binding specificity predictions from (A) BBK*, (B) flex ddG and (C) PocketOptimizer were correlated with experimentally determined binding specificities. Pearson correlations are given inside the corresponding plots (Adapted from Ayyildiz et al. 2024).

Despite variability in correlation values, all three methods showed some ability to reproduce the experimental trends for specific amino acids. BBK* and PocketOptimizer results closely aligned with the experimental data, while flex ddG demonstrates potential despite its lower overall correlation for the Arg-binder in both scaffolds.

Following the evaluation of the Arg-binder, the four other binders, Tyr-, Trp-, His-, and Ile-binder, whose sequences are known, were modeled using PyMOL and the methods were assesed the same way as the Arg-binder. BBK* predictions calculated using the structure 6SA8, show a range of correlations across the four pockets with Pearson's R values of 0.259, 0.355, 0.813 and 0.166, for Tyr, Trp, His and Ile binding pockets, respectively (Figure 15). While Pearson's R values for Tyr and Trp pockets are not as high as for the Arg pocket, Tyr is predicted as the 4th best ligand for its pocket, and Trp is predicted as the 3rd best ligand, indicating that the method captures general trends. However, a noticeable overprediction emerges in these pockets where His and Arg are predicted to be better than they really are in Tyr and Trp pockets. For the His binding pocket BBK* achieves a very good prediction, where His is accurately ranked as one of the top ligands. However, an overprediction can be also seen in this pocket where Arg is predicted to be better

than His for the His pocket. This suggests a potential tendency for BBK* to overrepresent the positively charged amino acids. The value for His in the His binding pocket might even be benefiting from this overprediction. For the Ile pocket, BBK* predictions exhibit poor correlation and overprediction of His and Arg, which contributes to the overall poor agreement (Figure 15-A).

On the other hand, flex ddG achieves moderate to strong correlations for Tyr and Trp pockets with 0.656 and 0.774 Pearson's R compared to BBK*. For the Tyr-binder, Tyr is ranked as one of the best ligands together with other aromatic residues and for the Trp binder, Trp is accurately ranked as the best ligand, aligning well with the experimental data. However, a positive tendency for aromatic residues could be observed which also shows in the His and Ile-binder. In the His-binder, His is predicted to be the 3rd best ligand for its pocket, however, the over prediction of Tyr and Trp plays a role in the lower R value of 0.311. The Ile-binder shows the lowest correlation among all binders and all methods, with a near-zero correlation (R: 0.03) and a tendency for Tyr, Trp, His and Phe residues can be also observed for this binder's prediction (Figure 15-B).

PocketOptimizer on the other hand shows a more uniform, though moderate performance across all evaluated pockets, with R values of 0.569, 0.566, 0.624 and 0.194 for Tyr, Trp, His and Ile binders, respectively. However, in the Tyr, Trp and His binder plots, it is possible to observe some tendencies for certain amino acids, such as His, Arg and Trp. This may indicate that although PocketOptimizer carries some similarities to both BBK* and flex ddG, it does not demonstrate this as much as the other methods do, which may explain the more uniform and moderate performance of it. However, the most challenging case of the Ile binder shows also only a weak correlation of 0.166 R (Figure 15-C). In summary, the Ile binder correlations are consistently poor across all methods (Ayyildiz et al., 2024).

The predictions based on the scaffold 5AEI, show only slight differences among binders and methods (see Appendix for the correlation plots calculated using 5AEI scaffold).

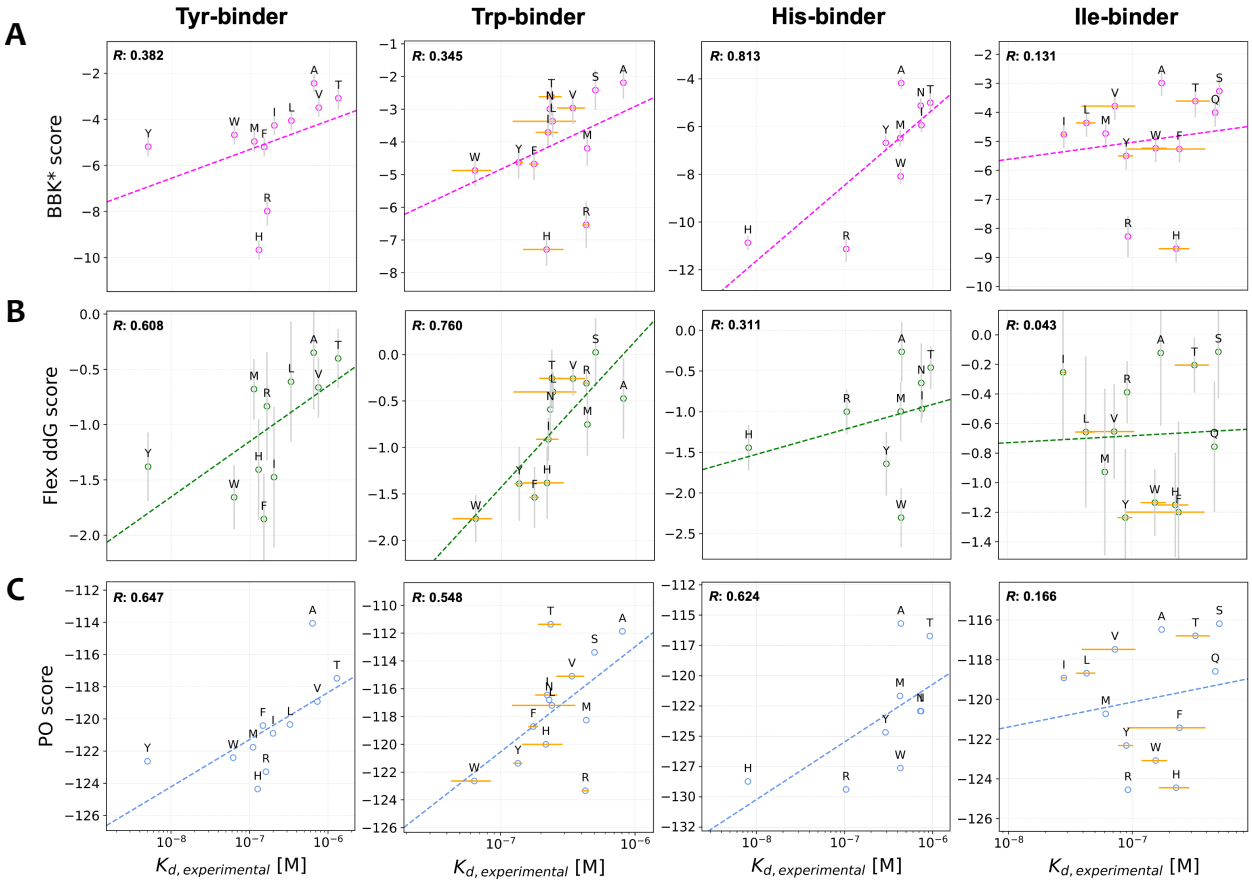

Figure 15: Correlation between calculated binding specificity predictions and experimental binding specificities for the Tyr, Trp, His, and Ile binding pockets using 6SA8 as scaffold. Correlation between experimental measurements for each binder with the calculations from BBK* (A), from flex ddG (B) and from PocketOptimizer (C) are given with their corresponding Pearson correlations (Taken from Ayyildiz et al., 2024).

Overall, the results highlight the strengths and limitations of each method. BBK* shows strong alignment with the experimental data for the His binder, but it also shows tendencies that affect its performance in other pockets. Flex ddG demonstrates the best overall performance for the Tyr and Trp-binder, however, as with BBK*, its tendencies for these amino acids affect its performance for other binders. PocketOptimizer provides more balanced predictions but lacks the precision of the other methods in specific cases. These observations emphasize the importance of combining these methods to improve the overall prediction accuracy.

4.3.2 Tendencies of Predictive Methods

Computational methods can overestimate the energetic contributions of certain amino acids, leading to systematic biases in binding predictions. Therefore, in this study further analysis to quantify these effects has been conducted. To this end, a relative bias was calculated for each target peptide in relation to its corresponding amino acid (Figure 16). The results indicate a consistent tendency to overemphasize larger amino acids while underestimating smaller ones. This is evident in the distribution of relative offsets, where predictions for small amino acids such as Gly, Ala, and Ser are shifted towards the left, whereas larger amino acids appear shifted twoards the right. Such biases may stem from the nature of rotamer sampling, which can lead to excessive packing within the binding pocket. Additionally, limitations in the scoring functions may contribute to this effect by failing to fully account for steric constraints or the reduced interaction surface of smaller residues. It is important to note that Lys, Glu, and Asp are represented by fewer data points, suggesting pointing a lower availability of reliable predictions for these amino acids. In contrast, Leu and Ile appear more evenly distributed, indicating a more balanced modeling of their energetic contributions across different binding pockets. Interpreting these biases is further complicated by the small dataset, as well as the lack of experimental validation for several binding pockets, which limits the robustness of the analysis. Moreover, while relative offsets were applied to correct for these biases, their normalization introduces variability in absolute error, leading to potential inconsistencies across different pockets. Despite these challenges, addressing such biases is essential for improving computational design strategies.

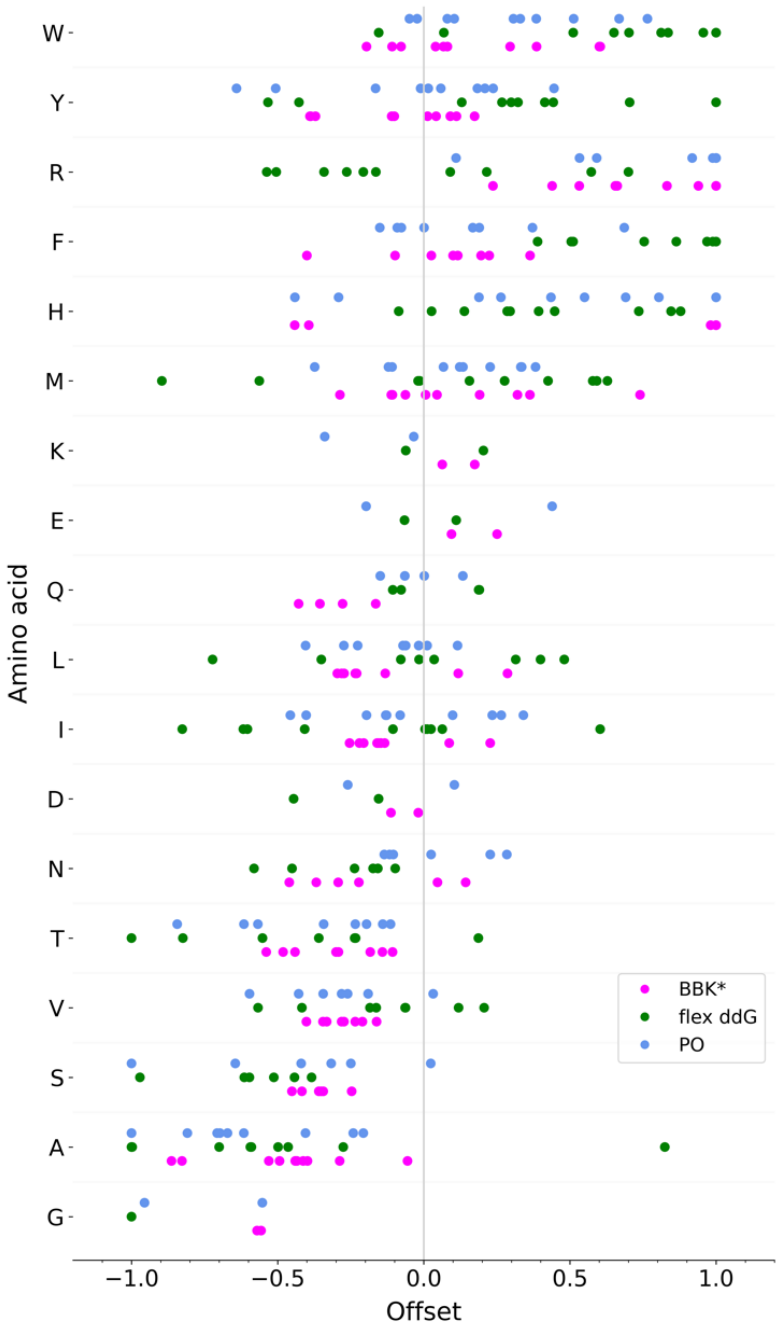

Figure 16: Individual relative offsets from optimal fit for individual amino acid targets. Amino acids are listed at the y-axis according to their relative mass (Figure is taken from Ayyildiz et al., 2024).

4.3.3 Correlation of Computational Predictions

The variability in predictive performance across five binding pockets for individual computational methods highlights the limitations of relying on a single approach. To address this, predictions from the three methods were correlated, with the expectation that their combined strengths could offer deeper insights for designing and evaluating new binding pockets.

In Figure 17, the correlation between the three methods based on their predictions from two of the five binders are given. In the Arg binding pocket, flex ddG showed limitations in accurately predicting Arg. However, the other two methods, outperformed flex ddG, providing a more reliable assessment (Figure 17-A). On the other hand, in the Tyr binding pocket, flex ddG demonstrated better accuracy, and corrected the overestimation tendencies observed for His and Arg amino acids in BBK* and PocketOptimizer (Figure 17-B). The correlations for the remaining three binders show slight variations, which can be found in Appendix.

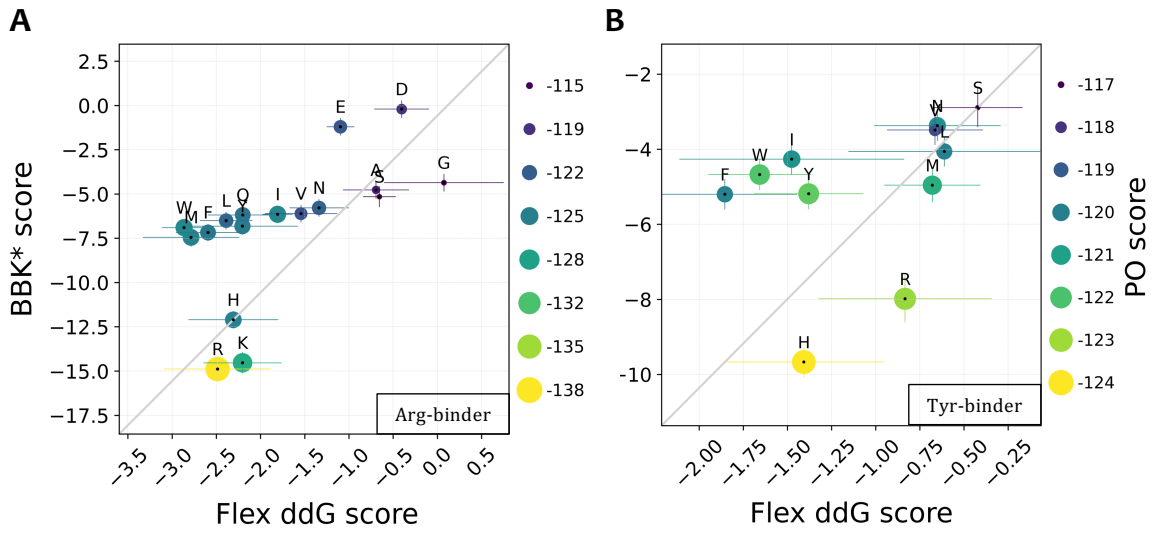

Figure 17: Correlation of specificity predictions from all three methods. BBK*, flex ddG, and PocketOptimizer predictions for (A) Arg and (B) Tyr binders were obtained using the crystal structure 6SA8 as the scaffold (Figure is adapted from Ayyildiz et al., 2024).

The underlying hypothesis was that correlating these methods would help alleviate individual biases by allowing opposing tendencies to cancel each other out. Besides the results presented in Figure 17, the biases largely remained, likely due to the similar tendencies shared across methods. This finding underscores an important consideration in computational binding pocket design; while method combination remains a promising strategy for reducing systematic errors, its effectiveness depends on the diversity of the underlying algorithms. These results highlight the distinct strengths and weaknesses of the three methods, emphasizing their complementary nature and their relative contributions to capturing binding specificity. More broadly, they underscore the value of integrating multiple computational approaches to enhance predictive accuracy and address the challenges posed by diverse binding pockets. With access to more high-quality experimental data and further refinement of predictive models, it may become possible to better characterize these interactions and improve computational design strategies. Additionally, incorporating methods with fundamentally different scoring principles or including more comprehensive predictive calculations could further improve bias correction and enhance the robustness of computational predictions.

5. Experimental Evaluation of Peptide Binding Specificities

5.1 The Experimental Set-up

Within PReART, the aim is to create specific binding pockets for each amino acid and facilitate the combination of these designed binding modules in a required way without any exhausive computational or experimental work. As explained in previous chapters, the computational pipeline for specificity prediction of promising sequences from designed pocket libraries, was established in Bayreuth. Based on computational suggestions, libraries are then produced produced by our collaboration partners at Aston University (UK) and tested at Universität Zürich (CH). However, this workflow can be inconveniently time-consuming and quick testing of some mutations for any interesting binding pocket or pocket combinations based on computational work would be useful. Therefore, I established the experimental workflow for affinity measurements of binder-peptides in our lab. In this thesis, “binder” refers to any designed armadillo scaffold without its peptide bound, “binding pocket” (Figure 5) refers to one module of the binder that consists of seven amino acids that can be mutated in order to alter binding specificity and/or binding affinity towards amino acids in the peptide position six and “peptide” refers to an extended peptide that is bound to a binder.

In this section, the established protocol will be explained in detail. The initial plasmids for the expression of the WT-binder and WT-peptide were received from the Plückthun Lab. The WT-binder is a designed armadillo repeat protein that consists of 5 internal modules and includes a lock for Q and A amino acids of the bound peptide, which prevents peptide sliding. The WT-peptide construct includes superfolderGFP that is needed for the fluorescence binding assays. The arginine residue at position 6 of the WT-peptide was mutated to different amino acids via site directed mutagenesis. All proteins were expressed in *E.coli*, purified and affinity constants (K_D) were measured and compared to measurements from the Plückthun Lab. After the

successful establishment of the protocol, three promising GLN-binders – referring to three different binding pockets - that were designed for a glutamine residue at position 6 in the peptide to have a high affinity and a specifity (Bachelor Thesis, Freund 2021), were tested experimentally.

5.2 Experimental Methods

Cloning of peptide variants via Site Directed Mutagenesis

To ensure that the protocol was correctly established, the binding constants of the WT-binder to different peptide variants was measured. To generate the different peptide variants, the WT-peptide plasmid, KRKRKRQRAR-sfGFP, was mutated at position 6. Forward and reverse oligonucleotides for the mutation of the codons for Arg to the ones for Gln, Glu, Asn, Tyr, Leu, and Ser were designed using the Agilent Genomics QuickChange® PrimerDesign webtool, and the genes were ordered from Eurofins. PCR mixture with 10 μM of each respective forward and reverse strands of oligonucleotides were annealed in a total volume of 50 μL with ddH2O (see Table 5.1, 5.2 PCR mixture and PCR profile).

Table 10: Composition of the PCR reaction. Reaction was followed with to amplify double stranded DNA with specific primers.

Compound	Final Concentration	Volume
Primer forward (fw)	10 μM	2.5 μL
Primer reverse (rw)	10 μM	2.5 μL
deoxyribose nucleoside triphosphate (dNTPs)	0.3 μM	1.5 μL
DNA template	50-100 ng	1 μL
KAPA HiFI High Fidelity Buffer 5x	-	10 μL
KAPA Polymerase	2u/μl	0.5 μL

ddH_2O	-	ad 50 µL

Table 11: Site-directed mutagenesis PCR temperature profile.

Steps	Temperature (°C)	Duration (seconds)	Cycles
Initial Denaturation	96	30	1
Denaturation	96	10	28
Annealing	53	10	
Elongation	72	240	
Final elongation	72	420	1
Hold	4	∞	-

1 µL DpnI (20000 units/ml) were added to the PCR products for the digestion of the original plasmid and incubated at 37 °C for about 1 hour. Then, it was used to transform TOP10 cells as described below (see Protein Transformation, Expression and Purification). 2-3 colonies were picked from each plate and resuspended in 8mL LB media with 8 µL AMP (from 100 µg/mL). These pre-cultures were put into a 37 °C shaker overnight, then DNA was prepared using a kit (Table 4.7), and a sample was prepared for sequence verification by Eurofins.

Golden Gate cloning

Gene fragments for 3 GLN-binders were ordered from Twist, as these binders required several mutations in the binding pocket of the WT-binder. These gene fragments were cloned into pEM3BTC vector (Michel, Plückthun, & Zerbe, 2018) by Golden-Gate cloning (Table 5.3). The reaction was set-up in a microcentrifuge tube on ice. DNA fragments were mixed with nuclease-free water. NEBridge Ligase Master Mix was added and mixed by pipetting 3 times, as a last

component Type IIS restriction enzyme was added and mixed by pipetting 5 times. The mixture was incubated at 37 °C for 15 min and deactivated by heating at 60 °C for 5 min. 10 µl of the product was transformed into chemically competent cells by following the steps described below (see Protein Transformation, Expression and Purification) and they were verified by sequencing.

Table 12: Golden-Gate protocol. (Taken from NEB website)

Components	Amount
NEBridge Ligase Master Mix	5 µl
DNA Fragments*	0.05 pmol each
Type IIS Restriction Enzyme**	x µl
Nuclease-free Water	Y µl
Total Reaction Volume	15 µl

*DNA Fragments = Vactor and Insert, 0.05 pmol in a 1:1 ratio (ratio could be optimized but wasn't necessary here)

**Type IIS Restriction Enzyme = BsaI-HFv2 (NEB #R3733) 1 µl (20 U)

Protein Transformation, Protein Expression and Purification

All provided DNAs which include genes encoding WT-protein, WT-peptide, in addition to sequence verified genes DNAs of three GLN-binders and peptide variants were used for transforming cells in the following way; 50 µL of chemically competent *E. coli* cells were thawed on ice for 5 minutes, 1-2 µL (≈ 150 ng) of plasmid DNA was added and the cells were incubated for 10 minutes on ice followed by a heat shock at 42 °C for 1 minute. Cells were incubated on ice for 2 minutes, then 900 µL LB medium were added and the cells were incubated for 45 minutes at 37 °C in 800 rpm for regeneration. Afterwards, 200 µL of the transformation reaction cells were

plated on LB-agar plates supplemented with 100 μg/ml ampicillin (AMP). Plates were incubated at 37 °C overnight. When transforming cells with plasmid DNAs encoding HRV-3C protease, which is necessary to separate the $(His)_6$-tagged GB1 domain, instead of AMP, kanamycin was used.

For plasmid amplification 50 ng of the DNA were used to transform into *E. coli* TOP10 or DH5α competent cells the plates were incubated at 37 °C overnight. The next day, 5 mL E.coli TOP10 or DH5α cultures were grown in LB-medium overnight at 37 °C. The plasmids were purified by using NucleoSpin® Plasmid kit (Machery & Nagel) according to the manufacturer's instructions. The DNA concentration was determined by measuring the absorbance at 260 nm with an UV/Vis spectrometer (Equation 3). Samples were applied on a μ-cuvette (Eppendorf). A260/A280 ratios were used to assess purity of the DNA samples. The DNA samples whose sequence were known were stored at -20.

$$C = A_{260} \text{x } 50 \ \frac{\mu g}{ml} \qquad \text{Equation 3}$$

c: DNA concentration μg/ml

A260: Absorbance at 260 nm

For the expression of proteins, first a pre-culture was prepared by inoculating with a single colony that was picked from plates after the transformation. Each colony was mixed with 15 ml LB and 100 μg/ml AMP, and then pre-culture was incubated overnight or at least 6 hours at 37 °C at 180 rpm. 10 ml pre-culture were transferred into autoclaved 1 L TB media, and expression was induced at OD_{600} of 0.6-0.9 with 1 mM IPTG. The expression culture was incubated for 16 hours at 30 °C and 240 rpm.

Cells were harvested by centrifugation at 4 °C, with 4000 rpm for 15 minutes. Supernatant was discarded and the cell pellets were resuspended in 30 ml Buffer A (see Table 4.6). If not used

immediately, the resuspended cell pellets were once more centrifuged at 15000 rpm on the bench centrifuge for 10 minutes and cell pellets were stored at -20 °C (JLA-8.1000 Beckmann Coulter rotor). Resuspended cells were lysed with a Branson Ultrasonics 250 Sonifier by sonification of cells 3x3 minutes with one minute break in between on ice, using a duty cycle of 40% and an output power of 4 for releasing the intracellular components including proteins. After sonification, the lysate was centrifuged for 1 hour at 18000 rpm and 4 °C to separate the soluble protein from cellular debris.

The pellet was discarded, and the supernatant was collected and loaded onto a 5 ml HisTrap HP column to purify by Immobilized Metal Affinity Chromatography (IMAC). The HisTrap column was previously equilibrated with 10 column volumes of buffer A (see Table 5.6). Samples were washed with 15 column volumes of buffer A, and eluted with 100 ml gradient buffer A to buffer B. The eluted 3C-protease was dialyzed overnight in 2 liter dialysis buffer in a 12-14 kDa MWCO dialysis membrane at 4 °C. The previously purified 3C-proteases (about 2 mg) was added to eluted proteins for the cleavage of $(His)_6$-GB1 fusion and they were also dialyzed overnight in 2 liter dialysis buffer in a 12-14 kDa MWCO dialysis membrane at 4 °C. After separation of the target protein from $(His)_6$-tagged species, the sample was reapplied on a 5 mL HisTrap HP column, and purified protein was collected. These samples were dialyzed twice overnight, once in 2 liter analysis buffer, once for at least 8 hours and again for at least 3 hours. All the samples that were not used immediately were flash-frozen in liquid nitrogen and stored at -80 °C. Proteins intended for affinity measurements were also dialyzed in PBS buffer. However, no differences could be measured in the measurements between the samples dialyzed in analysis or PBS buffer.

Protein concentrations were determined using an UV/Vis spectrometer (BioPhotometer) to measure the absorbance at 280 nm and the concentration was determined using the Beer-Lambert Law in which the concentration of the absorbing species can be obtained with the measured absorbance and the corresponding molar extinction coefficient of the target protein (Equation 4), obtained from ExPASy ProtParam Tool.

$$A_{280} = \frac{c \text{ x } d \text{ x } \varepsilon_{280} \cdot}{MW}$$

$$c = \frac{A_{280} \text{ x } MW \cdot}{d \text{ x } \varepsilon_{280}} \qquad \text{Equation 4}$$

c : Protein concentration [g/l]

A_{280}: Absorbance at 280 nm

MW: Molecular weight [g/mol]

d: Path length [cm]

ε_{280}: Molar extinction coefficient at 280 nm [1/ m x cm]

Before the binding affinity assay, the purity of all protein variants was analyzed with non-reducing sodium dodecyl sulfate polyacrylamide gel electrophoresis (SDS-PAGE). Samples were prepared by adding 20 µl of protein samples mixed with 20 µl dye buffer (Lämmli) containing DTT and heated at 99 °C for 5 minutes before the application on the gel. PierceTM unstained protein MW marker (ThermoFisher) was used as a standard, and staining was performed with InstantBlue and de-stained with ddH_2O. Gels were imaged using the E-Box VX2 20 M. The composition of SDS-PAGE and SDS-sample buffer is given in Table 4.6. Proteins (when needed) were concentrated to up to 500 µM for the affinity measurements using Amicon® Ultra-15 ml tubes with 10K cutoff, by following the guidelines of the manufacturer. The proteins were stored at room temperature, at 4°C and at -80°C, for stability observations.

Circular dichroism spectroscopy

All circular dichroism (CD) measurements were performed on a JASCO-710 spectropolarimeter with a peltier element to analyze secondary structure of the protein and its thermal stability. The measurement was conducted in a 1 mm quartz cuvette at 20°C (0.1 nm data pitch, response 2 s, 100 nm min-1 scanning speed, and 1.0 nm bandwidth). Far-UV spectra were recorded from 195 to 260 nm with a pitch of 0.1 nm at 20°C. Each protein was recorded five times and results were averaged. The CD signal was corrected by buffer subtraction and converted to mean residue ellipticity (MRE). Folding-unfolding-refolding mechanism was also monitored by following the change in the molar ellipticity at 222 nm from 20°C to 90°C with a temperature increase of 1 °C/min. After buffer subtraction CD data was converted to the mean residue ellipticity using Equation 5.

$$[\theta]_{MRE} = \frac{\theta \cdot MW}{10 \cdot c \cdot d \cdot (n-1)} \qquad \text{Equation 5}$$

$[\theta]_{MRE}$: Mean Residue Ellipticity $[\frac{deg \cdot cm2}{dmol}]$

MW: Molecular Weight $[\frac{g}{mol}]$

c: Concentration $[\frac{mg}{ml}]$

n: Number of amino acids

d: Path length [cm]

Dissociation Constant (K_D) Determination by Fluorescence Anisotropy

All affinity measurements were conducted by fluorescence anisotropy (Cheow et al., 2014; Rossi & Taylor, 2011). Fluorescence anisotropy is a technique that measures the degree of polarization of fluorescent light emitted by a molecule when it is excited with polarized light. This technique provides information about the rotational motion and interactions of molecules in their environment. In this thesis, it is used to study protein-ligand interactions by measuring the binding of a fluorescent ligand (peptide-GFP) to a larger protein (binders). The background of the technique can be summarized as following:

The fluorophore is excited with polarized light. The emitted light is then measured in two orthogonal polarization directions: parallel (I_{vv}) and perpendicular (I_{vH}) to the excitation light (Equation 6). Fluorescence anisotropy (A) is calculated using the formula with the correction factor (G):

$$A = \frac{IVV - GIVH}{IVV + 2IVH} \qquad \text{Equation 6}$$

A higher anisotropy value indicates less rotational motion, suggesting larger or more rigid molecules. Lower anisotropy indicates more rapid rotational motion, suggesting smaller or more flexible molecules. Anisotropy can be used to study the binding interactions between a protein and a ligand. As the ligand binds to the protein, its rotational freedom decreases, resulting in increased anisotropy. By measuring anisotropy at various ligand concentrations, the binding constant (K_D) can be determined.

In this thesis, all measurements were conducted on the TECAN Spark® II microplate reader using flat-black 96-well plates (ThermoFisher). The concentration of the WT-peptide containing sfGFP was kept constant (100 µl peptide/well) at a concentration of 10 nM whereas the binders were

diluted (dilution factor 0.6) over the plate with a 10 µM starting concentration (Figure 18). For each plate quadruplicates were pipetted whereby one dataset consists out of 24 measured points. 100 µl PBS buffer (pH 7.4, 137 mM NaCl, 3 mM KCl, 8 mM NaPi, 45 mM KPi) supplemented with Tween 20 (0.05%) was added to each well. The polarization data was averaged and fitted in the Fit-o-Mat (Möglich, 2018) with the Hill-Langmuir equation.

Normalized data was generated by normalizing each separate dataset of one plate, followed by averaging all normalized quadruplicate values.

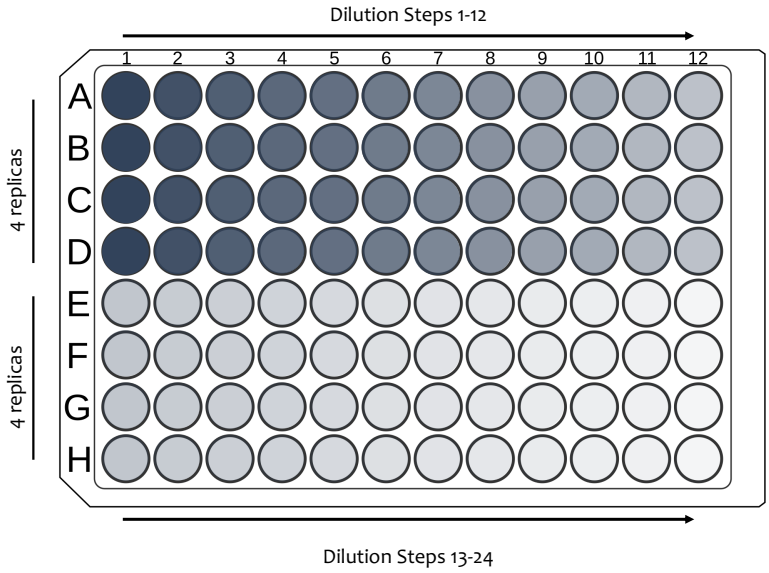

Figure 18: Pipetting scheme for the fluorescence anisotropy measurements. Wells A to D are the starting wells with the highest binder concentrations and wells E to H are the 13 to 24th wells used after A-D 12.

5.3 Materials Used in Experimental Set-up

Table 13: List of used bacterial strains.

Bacteria (*E.coli*) strain	
BL21 (DE3)	*E. coli* B F– dcm ompT hsdS($r_B^-m_B^-$) gal λ(DE3)
TOP10	F^- mcrA Δ(*mrr-hsd*RMS-*mcr*BC) Φ80*Lac*ZΔM15 Δ*Lac*X74 *rec*A1 *ara*D139 Δ(*araleu*) 7697 *gal*U *gal*K *rps*L (StrR) *end* A1 *nup*G
DH5α	F^- endA1 glnV44 thi-1 recA1 relA1 gyrA96 deoR nupG purB20 ϕ80dlacZΔM15 Δ(lacZYA-argF)U169, hsdR17(rK–mK+), λ–

Table 14: List of used media and antibiotics.

Growth media and antibiotics	**Content**	**Manufacturer**
Lysogeny broth (LB)	Tryptone (10 g/l), yeast extract (5 g/l), NaCl (10 g/l), pH 7.0 ± 0.2	Carl Roth
Terrific broth (TB)	Caseine 12 g/l, Yeast extract 24 g/l, K_2HPO_4 9.4 g/l, KH_2PO_4 2.2 g/l, pH 7.2 ± 0.2	
Ampicillin (Amp)		
Kanamycin (Kan)		

Table 15: List and composition of used buffers.

Purification Buffers	Content	pH
Buffer A	50 mM $Na_2HPO_4xH_2O$, 20 mM imidazole,500 mM NaCl, (NaN_3 – 30 μM - optional)	7.7
Buffer B	50 mM $Na_2HPO_4xH_2O$, 500 mM imidazole,500 mM NaCl, (NaN_3 – 30 μM - optional)	7.7
Dialysis Buffer	50 mM $Na_2HPO_4xH_2O$,100 mM NaCl, (NaN_3 – 30 μM - optional)	7.7
Analysis Buffer	20 mM $Na_2HPO_4xH_2O$,150 mM NaCl, (NaN_3 – 30 μM - optional)	7.0
PBS	137 mM NaCl, 3 mM KCl, 8 mM $Na_2HPO_4xH_2O$, 1.5 mM KH_2PO_4	7.0
SDS-PAGE		
Stacking gel	45 mL Separating Gel Mix, 10% Aps 0.5 ml, TEMED 0.05 ml	-
Separating gel	24 mL Separating Gel Mix, 10% Aps 0.25 ml, TEMED 0.025 ml	-
SDS-Loading Buffer	200 mM DTT, 100 mM Tris pH 6.8 20% (w/v) glycerol,4% SDS bromphenol blue	-
SDS Staining Solution	Coomassie Blue G-250 (110 g), phosphoric acid (80 g), ethanol, (50 g), water (850 mL)	-
Running Buffer (10x)	-	-

Analytics		
CD	10 nM $NaH_2PO_4xH_20$	7.5

Table 16: List of purification kits.

Name of Kit	**Supplier**
NucleoSpin® Plasmid	Macherey-Nagel

Table 17: List of used enzymes and respective buffers.

Enzyme & Respective Buffers	**Supplier**
3C Protease	*In-house* preparation (Plückthun Lab)
BamHI	New England Biolabs
DpnI	
Phusion® HF DNA polymerase	
Phusion HF buffer	
T4 DNA Ligase	
T4 DNA Ligase Buffer	
SDS loading Buffer	
SDS staining solution	

Running Buffer (10x)

Table 18: List of chemicals.

Chemicals	Manufacturer
Agarose	Carl Roth
Coomassie Brilliant Blue G-250	SERVA
DTT	Carl Roth
Ethanol	Carl Roth
Glycerol	VWR Chemicals
HCl	Fisher Chemicals
Imidazole	Sigma-Aldrich
Isopropyl-β-D-thiogalactopyranoside (IPTG)	VWR Chemicals
NaCl	Fisher Chemicals
Na_2HPO_4	Grüssing GmbH
NaH_2PO_4	Grüssing GmbH
NaOH	AppliChem
Sodium Azide ($NaN_{3)}$	VWR Chemicals
Tween 20	

Table 19:List of used equipments.

Name	Manufacturer
Äkta™ Pure	GE Healthcare
BD 53 Heating Cabinet	Binder
BioPhotometer®	Eppendorf AG
C24 Incubator Shaker	New Brunswick Scientific GmbH
Centrifuge 5424	Eppendorf AG
Centrifuge 5810 R	Eppendorf AG
CD spectrophotometer-J710	JASCO Corporation, Tokyo (JPN)
Electrophoresis Power Supply-EPS 301	GE Healthcare
HisTrap™ HP column 5mL	
Incubator	Binder
Incubator Shaker Series Innova® 4	New Brunswick Scientific GmbH
JA 25.50 Rotor	Beckmann Coulter
JLA 8.1000 Rotor	
Nun 96-MicroWell flat black	Thermo Scientific
pH 211 Microprocessor pH Meter	Hanna instruments
Research® plus pipettes: 0.1-2.5 µL, 0.5-10 µL, 10-100 µL,100-1000 µL,	Eppendorf AG

8-channel 0.5-10 µL, 8-channel 30-300 µL	
Spark® multimode microplate reader	TECAN
Sonifier 250 (CE)	Branson Ultrasonics
Thermoblock	Dixell
Vortex Mixer	VWR

Table 20: List of consumables.

Name	Company
Amicon® Ultra-15 centrifugal filter 10 kDa MWCO	Merck
Milli-Q water system	
MWCO Millipore	

5.4 Establishing the Purification Protocol with WT Proteins

After transformation in *E.coli* BL21 cells with the received plasmids, WT-binder and WT-peptide were expressed and purified from the soluble fraction of the cells using the IMAC protocol described above (see section 5.2). Both could be purified with high yields (WT-binder: 1.73 mg/ml ~35 ml and WT-peptide: 9.12 mg/ml 10 ~ml total). An SDS-PAGE was used to check the purity of the proteins obtained (Figure 19).

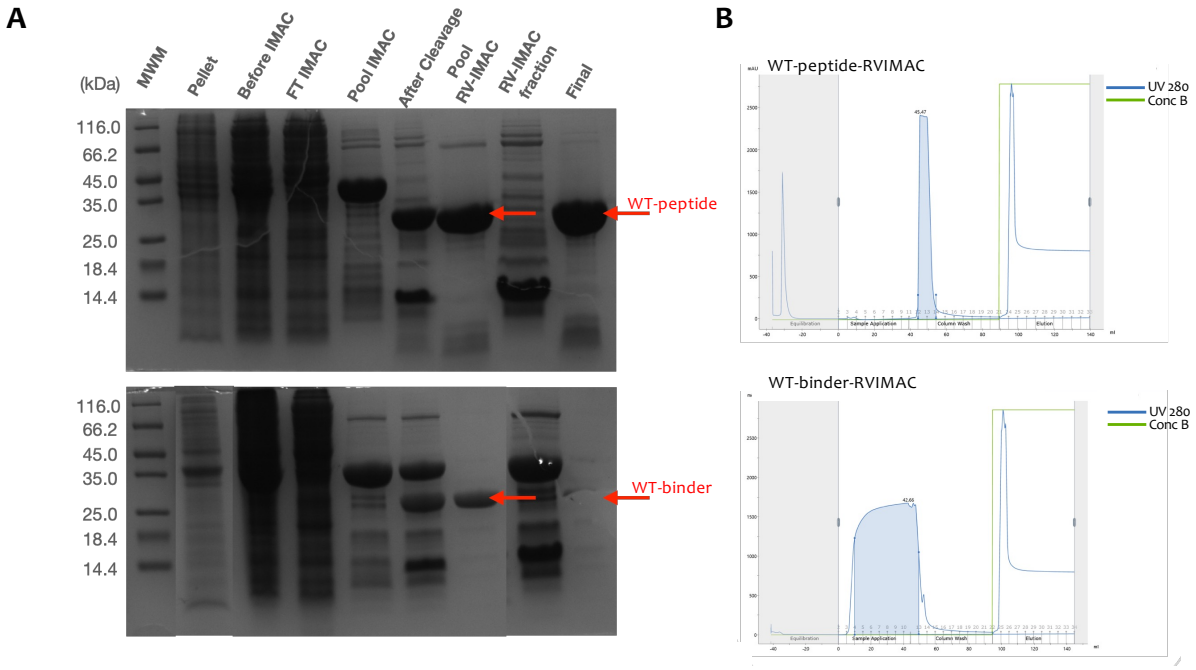

Figure 19: Purification of WT-binder and WT-peptide. (A) Purification followed on SDS-PAGE. The red arrows mark the proteins of interest after reverse IMAC (RV-IMAC, see section 5.2) and final corresponds to proteins collected after second analysis ON; upper SDS-PAGE shows WT-peptide, bottom one shows WT-binder. (B) Elution profiles of proteins after second IMAC. Absorption [mAU] is represented in blue. For the RV-IMAC chromatogram the concentration of Buffer B is represented in green.

After successful purification of the proteins, dissociation constants (K_D) were determined by fluorescence anisotropy according to a protocol where sfGFP-labeled peptides were used at a constant concentration while binders pipetted in decreasing concentration (5.2.1 Exprimental Methods).

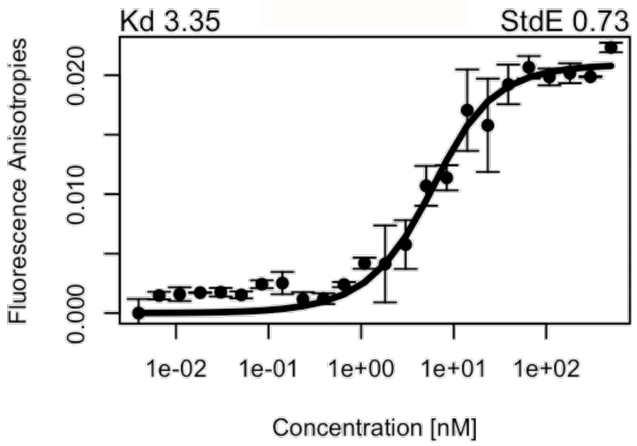

Figure 20: Determining the dissociation constant of WT-binder and WT-peptide using fluorescence anisotropy. Measurement was carried out at a constant peptide concentration of 5 nM and a starting concentration of 5 μM protein. Baseline is normalized (see section 5.2). The obtained K_D is 3.35 nM which is fits with the K_D measured in Plückthun Lab (3.3 nM).

After successfully establishing the K_D measurements protocol, the residue at the peptide position 6, KRKRK**R**QRAR, was mutated from Arg to Gln, Glu, Asn, Tyr, Leu, and Ser by site-directed mutagenesis. The sequences for peptide-Q (KRKRK**Q**QRAR), peptide-E, peptide-N, peptide-M, all with sfGFP were confirmed by Sanger sequencing (Eurofins Genomics). Each mutant peptide was expressed and purified with the same protocol, and resulted in high yields as comparable to WT-peptide. However, with Tyr, Ser and Leu, no colony with the desired sequences were found and remained to be tested. It should be mentioned that with these constructs double mutations have been observed in the peptide, such as; KRKRK**Y**K**Y**QRAR, where two positions of the peptide contained Tyr, instead of only at position 6. The peptide also consisted of 12 residues instead of 10. After sequencing, peptide variants with the correct sequences were used to transform BL21

cells. They were expressed and purified and dissociation constants for WT-binder with different peptide variants were measured as described in 5.2.1 Experimental Methods(Table 5.12). In general, the determined binding constants are lower than the ones our colleagues in Zürich reported. In particular measurements with the E-peptide deviated by tenfold. However, the overall trend stayed the same.

Table 21:The K_D values of the WT-binder with peptide variants. Values are compared to values measured in Plückthun Lab. K_D are given as nM.

Measurements	WT-peptide	Peptide-Q	Peptide-N	Peptide-E
K_Ds in Bayreuth	3.35 ± 0.7	22.9 ± 3.6	80.0 + 17.2	117.0 + 13.6
K_Ds in Zurich	3.3	37.2 + 10.4	136.1 + 23.9	1343.0 + 323

5.5 Testing of Gln-Binder Designs

After successful establishment of the binding assay, three GLN-binders designs were tested. Binders are named as Binder-1,-2, and -3. Binder-1 (pocket sequence: qEsqqeR) is selected to be tested first as it was predicted to be the best binder for the GLN-peptide (KRKRK**Q**QRAR), and Binder-2 (pocket sequence is: qEsqqQK) was predicted to be the second best binder for GLN-peptide according to the computational analysis (Freund E., Bachelor Thesis, 2021). The small letters in the pocket sequence refer to no mutations at this position, while capital letters refers to mutation at this position to the written amino acid. Both sequences were included in the GLN binder library suggested by Freund, for further evaluation by the Hine and Plückthun groups. The Binder-3 (pocket sequence: qDRqRQE) was optimized with PocketOptimizer (Noske et al., 2023) which is based on Binder-1, and whose sequence was not included in the above-mentioned library (Freund, Bachelor Thesis, 2021).

All binders were initially tested with peptide-Q to measure binding affitinity. Given the expectation of high affinity, the first assay was set up with a starting concentration of 5 µM of Binder-1 and 5 nM of peptide-Q. However, no binding was observed. Further raising of the starting concentration of Binder-1 up to 167 µM, binding to peptide-Q could still not be detected. This was unexpected, as the peptide residues, except for the mutation of position 6 from Arg to Gln, remained the same. While Binder-1 was hypothesized to have higher affinity to peptide-Q over WT-peptide, the observed trend of binding curve would likely fall within the micromolar range, which is higher than expected or beneficial within the frame of the PRe-ART project.

To ensure that Binder-1, which contains two mutations in the binding pocket, still folds correctly, circular dichroism (CD) spectroscopy was performed. The resulting spectra showed a predominantly alpha-helical structure, consistent with the profile of WT-binder. Binding affinities for Binder-2 and Binder-3 were measured in the same way as Binder-1, and none of the proteins showed a promising binding curve. Similar to Binder-1, their CD spectra closely resembled those of the WT-binder.

One possible explanation for the significant decrease in binding affinity, despite the proteins being correctly folded, could be issues related to the pocket formation itself. Specifically, highly charged residues such as R and E in the binding pocket, may block the binding site or alter the overall structure. Additionally, oligomerization could prevent the binding as well, further blocking interactions with the peptide.

Despite the challenges in identifying a high-affinity and specific binder among millions of sequences, this study provides insights into binder selection and highlights critical considerations for improving future design strategies.

Here, the successful development and validation of a binding assay for protein-peptide interactions, based on designed armadillo repeat protein variants, has been presented. This assay enables rapid and direct evaluation of potential binding pockets predicted by computational methods, thereby contributing to a deeper understanding of these computational approaches and facilitating improvements in future predictive accuracy. The assay demonstrates reliability, with all components, including proteins and peptides, being purifiable in a time-efficient manner and yielding high concentrations. Furthermore, the system supports versatility, as different binder sequences can be readily generated through site-directed mutagenesis. Once all peptide-sfGFP constructs are purified, assay preparation and execution can be achieved with efficienctly.

6. Conclusion and Outlook

The overall goal of this thesis is to contribute to the computational design of protein binding modules capable of specifically recognizing amino acids, using a regularized armadillo repeat protein scaffold as part of the PRe-ART project. By integrating multiple computational strategies and validating key aspects, this work provides a systematic approach to developing and assessing binder libraries, with a particular focus on phosphorylated amino acids.

A computational workflow was established to design targeted binder libraries for pTyr, pSer and pThr by identifying critical interactions and evaluating pocket specificity. Through this approach, certain residues at key positions such as Arg and Lys were identified as crucial for binding phosphorylated amino acids. The findings provide insights into how pTyr specifically interacts within natural as well as potential new binding pockets. Subsequently, two targeted libraries for pTyr and one targeted binder library for pSer and pThr were suggested. The designed libraries were followed up with in-vitro studies by our collaboration partners. The libraries were generated and preliminary in-vitro results confirm the ability of selected variants to bind pTYR with high affinity and specificity. In addition, results also hint at the ability of several sequences to successfully distinguishing pTyr from its unmodified counterpart, Tyr. This first validation underscores the potential of computational protein design. The followed workflow is general and can also be applied for designing specific binders for other amino acids.

To further refine computational binder design, this work also assessed the predictive accuracy of three well-established computational protein design methods for evaluating single mutations in a protein-peptide system of dArmRP. Since these methods are essential not only for designing phosphorylated binders but also for developing binders for other amino acids in the future, their accuracy and reliability were systematically analyzed. A standardized framework was developed to ensure a consistent comparison of their predictions, revealing distinct tendencies across the three approaches. While BBK* and flex ddG demonstrated high accuracy in certain cases, their

predictive power was influenced by systematic biases. PocketOptimizer, on the other hand, provided more balanced predictions but lacked the precision required in specific contexts. These findings underscore the challenges of single-residue predictions and emphasize the importance of understanding method-specific tendencies. Based on these results, a complementary analysis integrating predictions from all three methods was suggested, which could enhance the accuracy of computational binder design. Moreover this work established a test set that will serve as benchmark for future method development.

Beyond computational predictions, during this work, I established an experimental workflow for testing binding modules in the designed armadillo repeat protein system in our lab. Point mutations introduced through site-directed mutagenesis enabled the generation of peptide variants, which were expressed and purified with high yields for binding experiments. Three binders with distinct binding pockets were also successfully expressed and purified, providing a solid foundation for future experimental validation. Binding affinity experiments for the wild-type binder-peptide system yielded results that were consistent with both computational predictions and prior studies from our collaboration partners. The system established in this work enables the rapid and efficient evaluation of designed binding pockets, making it a valuable tool for future protein engineering efforts.

Overall, this thesis presents a comprehensive and integrative approach to design and evaluate protein binding pockets, bridging computational predictions with experimental validation. The computational workflow developed here is generalizable and can be applied to other amino acids, expanding its potential use in rational protein design. Moreover, by critically assessing computational methods and establishing experimental validation strategies, this work lays the foundation for future advancements in designing highly specific protein binders. As more high-quality experimental data become available and computational tools continue to evolve, further improvements in predictive accuracy and design strategies can be achieved, ultimately broadening the applicability of protein engineering approaches.

Building on these findings, future work could focus on expanding the binder library design strategy to a broader range of amino acids and post-translational modifications, particularly in cases where structural data is limited. Another important direction would be refining the computational methods used for single-residue predictions by incorporating hybrid approaches that combine physics-based modeling with machine learning techniques. Finally, applying these strategies to biologically or therapeutically relevant targets could help bridge the gap between computational protein design and real-world applications.

Bibliography

Acconcia, F., Barnes, C. J., Singh, R. R., Talukder, A. H., & Kumar, R. (2007). Phosphorylation-dependent regulation of nuclear localization and functions of integrin-linked kinase. *Proceedings of the National Academy of Sciences of the United States of America*, *104*(16), 6782–6787. https://doi.org/10.1073/pnas.0701999104

Alfarano, P., Varadamsetty, G., Ewald, C., Parmeggiani, F., Pellarin, R., Zerbe, O., … Caflisch, A. (2012). Optimization of designed armadillo repeat proteins by molecular dynamics simulations and NMR spectroscopy. *Protein Science*, *21*(9), 1298–1314. https://doi.org/10.1002/pro.2117

Alhajj, M., Zubair, M., & Farhana, A. (2025). Enzyme Linked Immunosorbent Assay. Treasure Island (FL).

Andersen, H. C. (1983). Rattle: A "velocity" version of the shake algorithm for molecular dynamics calculations. *Journal of Computational Physics*, *52*(1), 24–34. https://doi.org/10.1016/0021-9991(83)90014-1

Ardito, F., Giuliani, M., Perrone, D., Troiano, G., & Muzio, L. Lo. (2017). The crucial role of protein phosphorylation in cell signalingand its use as targeted therapy (Review). *International Journal of Molecular Medicine*, *40*(2), 271–280. https://doi.org/10.3892/ijmm.2017.3036

Ashraf, M., Frigotto, L., Smith, M. E., Patel, S., Hughes, M. D., Poole, A. J., … Hine, A. V. (2013). ProxiMAX randomization: a new technology for non-degenerate saturation mutagenesis of contiguous codons. *Biochemical Society Transactions*, *41*(5), 1189–1194. https://doi.org/10.1042/BST20130123

Ashworth, M. A., Bombino, E., De Jong, R. M., Wijma, H. J., Janssen, D. B., McLean, K. J., & Munro, A. W. (2022). Computation-Aided Engineering of Cytochrome P450 for the Production of Pravastatin. *ACS Catalysis*, *12*(24), 15028–15044. https://doi.org/10.1021/acscatal.2c03974

Ayyildiz, M., Noske, J., Gisdon, F. J., Kynast, J. P., & Höcker, B. (2024). Complementary evaluation of computational methods for predicting single residue effects on peptide binding specificities. *BioRxiv*, 2024.10.18.619108. https://doi.org/10.1101/2024.10.18.619108

Baker, M. (2015). Reproducibility crisis: Blame it on the antibodies. *Nature*, *521*(7552), 274–276. https://doi.org/10.1038/521274a

Banta, S., Dooley, K., & Shur, O. (2013). Replacing antibodies: Engineering new binding proteins. *Annual Review of Biomedical Engineering*, *15*, 93–113. https://doi.org/10.1146/annurev-bioeng-071812-152412

Barlow, K. A., Ó Conchúir, S., Thompson, S., Suresh, P., Lucas, J. E., Heinonen, M., & Kortemme, T. (2018). Flex ddG: Rosetta Ensemble-Based Estimation of Changes in Protein-Protein Binding Affinity upon Mutation. *Journal of Physical Chemistry B*, *122*(21), 5389–5399. https://doi.org/10.1021/acs.jpcb.7b11367

Bartlett, G. J., Porter, C. T., Borkakoti, N., & Thornton, J. M. (2002). Analysis of catalytic residues in enzyme active sites. *Journal of Molecular Biology*, *324*(1), 105–121. https://doi.org/10.1016/S0022-2836(02)01036-7

Binz, H. K., Amstutz, P., Kohl, A., Stumpp, M. T., Briand, C., Forrer, P., … Plückthun, A. (2004). High-affinity binders selected from designed ankyrin repeat protein libraries. *Nature Biotechnology*, *22*(5), 575–582. https://doi.org/10.1038/nbt962

Binz, H. K., & Plückthun, A. (2005). Engineered proteins as specific binding reagents. *Current Opinion in Biotechnology*, *16*(4), 459–469. https://doi.org/10.1016/j.copbio.2005.06.005

Bogdanova, E. A., & Novoseletsky, V. N. (2024). ProBAN: Neural network algorithm for predicting binding affinity in protein-protein complexes. *Proteins*, *92*(9), 1127–1136. https://doi.org/10.1002/prot.26700

Borrebaeck, C. A. K. (2000). Antibodies in diagnostics - From immunoassays to protein chips. *Immunology Today*, *21*(8), 379–382. https://doi.org/10.1016/S0167-5699(00)01683-2

Bradbury, A., & Plückthun, A. (2015). Reproducibility: Standardize antibodies used in research. *Nature*, *518*(7537), 27–29. https://doi.org/10.1038/518027a

Chandrasekhar, S. (1943). Stochastic Problems in Physics and Astronomy. *Rev. Mod. Phys.*, *15*(1), 1–89. https://doi.org/10.1103/RevModPhys.15.1

Chembath, A., Wagstaffe, B. P. G., Ashraf, M., Amaral, M. M. F., Frigotto, L., & Hine, A. V. (2022). Nondegenerate Saturation Mutagenesis: Library Construction and Analysis via MAX and ProxiMAX Randomization. *Methods in Molecular Biology (Clifton, N.J.)*, *2461*, 19–41. https://doi.org/10.1007/978-1-0716-2152-3_3

Chen, K., & Arnold, F. H. (1993). Tuning the activity of an enzyme for unusual environments: sequential random mutagenesis of subtilisin E for catalysis in dimethylformamide. *Proceedings of the National Academy of Sciences*, *90*(12), 5618–5622. https://doi.org/10.1073/pnas.90.12.5618

Chen, T. S., & Keating, A. E. (2012). Designing specific protein-protein interactions using computation, experimental library screening, or integrated methods. *Protein Science*, *21*(7), 949–963. https://doi.org/10.1002/pro.2096

Chen, W., Ying, T., & Dimitrov, D. S. (2013). Antibody-based candidate therapeutics against HIV-1: implications for virus eradication and vaccine design. *Expert Opinion on Biological Therapy*, *13*(5), 657–671. https://doi.org/10.1517/14712598.2013.761969

Cheow, L. F., Viswanathan, R., Chin, C.-S., Jennifer, N., Jones, R. C., Guccione, E., … Burkholder, W. F. (2014). Multiplexed Analysis of Protein–Ligand Interactions by Fluorescence Anisotropy in a Microfluidic Platform. *Analytical Chemistry*, *86*(19), 9901–9908. https://doi.org/10.1021/ac502605f

Coates, J. C. (2003). Armadillo repeat proteins: beyond the animal kingdom. *Trends in Cell Biology*, *13*(9), 463–471. https://doi.org/10.1016/S0962-8924(03)00167-3

Coutsias, E. A., Seok, C., Jacobson, M. P., & Dill, K. A. (2004). A Kinematic View of Loop Closure. *Journal of Computational Chemistry*, *25*(4), 510–528. https://doi.org/10.1002/jcc.10416

Davis, I. W., Arendall, W. B., Richardson, D. C., & Richardson, J. S. (2006). The backrub motion: How protein backbone shrugs when a sidechain dances. *Structure*, *14*(2), 265–274. https://doi.org/10.1016/j.str.2005.10.007

de Las Rivas, J., & Fontanillo, C. (2010). Protein-protein interactions essentials: Key concepts to building and analyzing interactome networks. *PLoS Computational Biology*, *6*(6), 1–8. https://doi.org/10.1371/journal.pcbi.1000807

Deribe, Y. L., Pawson, T., & Dikic, I. (2010). Post-translational modifications in signal integration. *Nature Structural & Molecular Biology*, *17*(6), 666–672. https://doi.org/10.1038/nsmb.1842

Diella, F., Haslam, N., Chica, C., Budd, A., Michael, S., Brown, N. P., … Gibson, T. J. (2008). Understanding eukaryotic linear motifs and their role in cell signaling and regulation. *Frontiers in Bioscience : A Journal and Virtual Library*, *13*, 6580–6603. https://doi.org/10.2741/3175

Dimitrov, D. S. (2012). Therapeutic proteins. *Methods in Molecular Biology (Clifton, N.J.)*, *899*, 1–26. https://doi.org/10.1007/978-1-61779-921-1_1

Dominguez, C., Boelens, R., & Bonvin, A. M. J. J. (2003). HADDOCK: A Protein−Protein Docking Approach Based on Biochemical or Biophysical Information. *Journal of the American Chemical Society*, *125*(7), 1731–1737. https://doi.org/10.1021/ja026939x

Dou, J., Doyle, L., Greisen, P., Schena, A., Park, H., Johnsson, K., … Baker, D. (2017). Sampling and energy evaluation challenges in ligand binding protein design. *Protein Science*, *26*(12), 2426–2437. https://doi.org/10.1002/pro.3317

Dyson, H. J., & Wright, P. E. (2005). Intrinsically unstructured proteins and their functions. *Nature Reviews Molecular Cell Biology*, *6*(3), 197–208. https://doi.org/10.1038/nrm1589

Eaton, B. E., Gold, L., & Zichi, D. A. (1995). Let's get specific: the relationship between specificity and affinity. *Chemistry and Biology*, *2*(10), 633–638. https://doi.org/10.1016/1074-5521(95)90023-3

Eccleston, R. C., Manko, E., Campino, S., Clark, T. G., & Furnham, N. (2023). A computational method for predicting the most likely evolutionary trajectories in the stepwise accumulation of resistance mutations. *ELife*, *12*, 1–36. https://doi.org/10.7554/eLife.84756

Forrer, P., Stumpp, M. T., Binz, H. K., & Plückthun, A. (2003). A novel strategy to design binding molecules harnessing the modular nature of repeat proteins. *FEBS Letters*, *539*(1–3), 2–6. https://doi.org/10.1016/S0014-5793(03)00177-7

Fosgerau, K., & Hoffmann, T. (2015). Peptide therapeutics: current status and future directions. *Drug Discovery Today*, *20*(1), 122–128. https://doi.org/https://doi.org/10.1016/j.drudis.2014.10.003

Friedland, G. D., Linares, A. J., Smith, C. A., & Kortemme, T. (2008). A simple model of backbone flexibility improves modeling of side-chain conformational variability. *Journal of Molecular Biology*, *380*(4), 757–774. https://doi.org/10.1016/j.jmb.2008.05.006

Gainza, P., Roberts, K. E., & Donald, B. R. (2012). Protein Design Using Continuous Rotamers. *PLOS Computational Biology*, *8*(1), 1–15. https://doi.org/10.1371/journal.pcbi.1002335

Georgiev, I., Lilien, R. H., & Donald, B. R. (2008). The minimized dead-end elimination criterion and its application to protein redesign in a hybrid scoring and search algorithm for computing partition functions over molecular ensembles. *Journal of Computational Chemistry*, *29*(10), 1527–1542. https://doi.org/https://doi.org/10.1002/jcc.20909

Gisdon, F. J., Kynast, J. P., Ayyildiz, M., Hine, A. V., Plückthun, A., & Höcker, B. (2022). Modular peptide binders-development of a predictive technology as alternative for reagent antibodies. *Biological Chemistry*, *403*(5–6), 535–543. https://doi.org/10.1515/hsz-2021-0384

Goodsell, D. S., & Olson, A. J. (1990). Automated docking of substrates to proteins by simulated annealing. *Proteins: Structure, Function, and Bioinformatics*, *8*(3), 195–202. https://doi.org/https://doi.org/10.1002/prot.340080302

Groves, M. R., & Barford, D. (1999). Topological characteristics of helical repeat proteins. *Current Opinion in Structural Biology*, *9*(3), 383–389. https://doi.org/10.1016/s0959-440x(99)80052-9

Guerin, N., Kaserer, T., & Donald, B. R. (2022). RESISTOR: A New OSPREY Module to Predict Resistance Mutations. *Journal of Computational Biology*, *29*(12), 1346–1352. https://doi.org/10.1089/cmb.2022.0254

Guo, Z., & Yamaguchi, R. (2022). Machine learning methods for protein-protein binding affinity prediction in protein design. *Frontiers in Bioinformatics*, *2*(December), 1–11. https://doi.org/10.3389/fbinf.2022.1065703

Hallen, M. A., Martin, J. W., Ojewole, A., Jou, J. D., Lowegard, A. U., Frenkel, M. S., … Donald, B. R. (2018). OSPREY 3.0: Open-source protein redesign for you, with powerful new features. *Journal of Computational Chemistry*, *39*(30), 2494–2507. https://doi.org/10.1002/jcc.25522

Hanes, J., & Plückthun, A. (1997). In vitro selection and evolution of functional proteins by using ribosome display. *Proceedings of the National Academy of Sciences of the United States of America*, *94*(10), 4937–4942. https://doi.org/10.1073/pnas.94.10.4937

Hansen, S., Tremmel, D., Madhurantakam, C., Reichen, C., Mittl, P. R. E., & Plückthun, A. (2016). Structure and Energetic Contributions of a Designed Modular Peptide-Binding Protein with Picomolar Affinity. *Journal of the American Chemical Society*, *138*(10), 3526–3532. https://doi.org/10.1021/jacs.6b00099

Harmansa, S., & Affolter, M. (2018). Protein binders and their applications in developmental biology. *Development (Cambridge)*, *145*(2). https://doi.org/10.1242/dev.148874

Harsha, H. C., & Pandey, A. (2010). Phosphoproteomics in cancer. *Molecular Oncology*, *4*(6), 482–495. https://doi.org/10.1016/j.molonc.2010.09.004

Helma, J., Cardoso, M. C., Muyldermans, S., & Leonhardt, H. (2015). Nanobodies and recombinant binders in cell biology. *Journal of Cell Biology*, *209*(5), 633–644. https://doi.org/10.1083/jcb.201409074

Henrich, S., Salo-Ahen, O. M. H., Huang, B., Rippmann, F., Cruciani, G., & Wade, R. C. (2010). Computational approaches to identifying and characterizing protein binding sites for ligand design. *Journal of Molecular Recognition*, *23*(2), 209–219. https://doi.org/10.1002/jmr.984

Jones, S., & Thornton, J. M. (1996). Principles of protein-protein interactions. *Proceedings of the National Academy of Sciences of the United States of America*, *93*(1), 13–20. https://doi.org/10.1073/pnas.93.1.13

Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., … Hassabis, D. (2021). Highly accurate protein structure prediction with AlphaFold. *Nature*, *596*(7873), 583–589. https://doi.org/10.1038/s41586-021-03819-2

Kaserer, T., & Blagg, J. (2018). Combining Mutational Signatures, Clonal Fitness, and Drug Affinity to Define Drug-Specific Resistance Mutations in Cancer. *Cell Chemical Biology*, *25*(11), 1359-1371.e2. https://doi.org/https://doi.org/10.1016/j.chembiol.2018.07.013

Kobe, B., & Kajava, A. V. (2001). The leucine-rich repeat as a protein recognition motif. *Current Opinion in Structural Biology*, *11*(6), 725–732. https://doi.org/https://doi.org/10.1016/S0959-440X(01)00266-4

Köhler, G., & Milstein, C. (1975). Continuous cultures of fused cells secreting antibody of predefined specificity. *Nature*, *256*(5517), 495–497. https://doi.org/10.1038/256495a0

Krismer, L., Schöppe, H., Rauch, S., Bante, D., Sprenger, B., Naschberger, A., … Heilmann, E. (2024). Study of key residues in MERS-CoV and SARS-CoV-2 main proteases for resistance against clinically applied inhibitors nirmatrelvir and ensitrelvir. *Npj Viruses*, *2*(1), 23. https://doi.org/10.1038/s44298-024-00028-2

Kuhlman, B., Dantas, G., Ireton, G. C., Varani, G., Stoddard, B. L., & Baker, D. (2003). Design of a Novel Globular Protein Fold with Atomic-Level Accuracy. *Science*, *302*(5649), 1364–1368. https://doi.org/10.1126/science.1089427

Kynast, J. P., & Höcker, B. (2023). Atligator Web: A Graphical User Interface for Analysis and Design of Protein–Peptide Interactions. *BioDesign Research*, *5*, 11. https://doi.org/10.34133/bdr.0011

Kynast, J. P., Schwägerl, F., & Höcker, B. (2022). ATLIGATOR: editing protein interactions with an atlas-based approach. *Bioinformatics*, *38*(23), 5199–5205. https://doi.org/10.1093/bioinformatics/btac685

Lazim, R., Suh, D., & Choi, S. (2020). Advances in molecular dynamics simulations and enhanced sampling methods for the study of protein systems. *International Journal of Molecular Sciences*, *21*(17), 1–20. https://doi.org/10.3390/ijms21176339

Lechner, H., Ferruz, N., & Höcker, B. (2018). Strategies for designing non-natural enzymes and binders. *Current Opinion in Chemical Biology*, *47*, 67–76. https://doi.org/https://doi.org/10.1016/j.cbpa.2018.07.022

Lemmon, G., & Meiler, J. (2012). Rosetta ligand docking with flexible XML protocols. *Methods in Molecular Biology*, *819*, 143–155. https://doi.org/10.1007/978-1-61779-465-0_10

Lilien, R. H., Stevens, B. W., Anderson, A. C., & Donald, B. R. (2005). Algorithm for Protein Redesign and Its Application Synthetase A Phenylalanine Adenylation Enzyme. *Journal of Computational Biology*, *12*(6), 740–761.

Lipman, N. S., Jackson, L. R., Trudel, L. J., & Weis-Garcia, F. (2005). Monoclonal versus polyclonal antibodies: Distinguishing characteristics, applications, and information resources. *ILAR Journal*, *46*(3), 258–267. https://doi.org/10.1093/ilar.46.3.258

Liu, H., & Chen, Q. (2023). Computational protein design with data-driven approaches: Recent developments and perspectives. *WIREs Computational Molecular Science*, *13*(3), e1646. https://doi.org/https://doi.org/10.1002/wcms.1646

Liu, N., Guo, Y., Ning, S., & Duan, M. (2020). Phosphorylation regulates the binding of intrinsically disordered proteins via a flexible conformation selection mechanism. *Communications Chemistry*, *3*(1), 1–9. https://doi.org/10.1038/s42004-020-00370-5

Loshbaugh, A. L., & Kortemme, T. (2020). Comparison of Rosetta flexible-backbone computational protein design methods on binding interactions. *Proteins: Structure, Function, and Bioinformatics*, *88*(1), 206–226. https://doi.org/https://doi.org/10.1002/prot.25790

Luo, R., Liu, H., & Cheng, Z. (2022). Protein scaffolds: antibody alternatives for cancer diagnosis and therapy. *RSC Chemical Biology*, *3*(7), 830–847. https://doi.org/10.1039/d2cb00094f

Madhurantakam, C., Varadamsetty, G., Grütter, M. G., Plückthun, A., & Mittl, P. R. E. (2012). Structure-based optimization of designed Armadillo-repeat proteins. *Protein Science*, *21*(7), 1015–1028. https://doi.org/10.1002/pro.2085

Maguire, J. B., Haddox, H. K., Strickland, D., Halabiya, S. F., Coventry, B., Griffin, J. R., … Kuhlman, B. (2021). Perturbing the energy landscape for improved packing during computational protein design. *Proteins: Structure, Function and Bioinformatics*, *89*(4), 436–449. https://doi.org/10.1002/prot.26030

Manning, G., Whyte, D. B., Martinez, R., Hunter, T., & Sudarsanam, S. (2002). The Protein Kinase Complement of the Human Genome. *Science*, *298*(5600), 1912–1934. https://doi.org/10.1126/science.1075762

Marsh, J. A., & Teichmann, S. A. (2015). Structure, dynamics, assembly, and evolution of protein complexes. *Annual Review of Biochemistry*, *84*, 551–575. https://doi.org/10.1146/annurev-biochem-060614-034142

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., & Teller, E. (1953). Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, *21*(6), 1087–1092. https://doi.org/10.1063/1.1699114

Michel, E., Plückthun, A., & Zerbe, O. (2018). Peptide-Guided Assembly of Repeat Protein Fragments. *Angewandte Chemie International Edition*, *57*(17), 4576–4579. https://doi.org/https://doi.org/10.1002/anie.201713377

Möglich, A. (2018). An Open-Source, Cross-Platform Resource for Nonlinear Least-Squares Curve Fitting. *Journal of Chemical Education*, *95*(12), 2273–2278. https://doi.org/10.1021/acs.jchemed.8b00649

Münch, R. C., Mühlebach, M. D., Schaser, T., Kneissl, S., Jost, C., Plückthun, A., … Buchholz, C. J. (2011). DARPins: an efficient targeting domain for lentiviral vectors. *Molecular Therapy : The Journal of the American Society of Gene Therapy*, *19*(4), 686–693. https://doi.org/10.1038/mt.2010.298

Noske, J., Kynast, J. P., Lemm, D., Schmidt, S., & Höcker, B. (2023). PocketOptimizer 2.0: A modular framework for computer-aided ligand-binding design. *Protein Science*, *32*(1), 1–8. https://doi.org/10.1002/pro.4516

Ojewole, A. A., Jou, J. D., Fowler, V. G., & Donald, B. R. (2018). BBK∗ (Branch and Bound over K∗): A Provable and Efficient Ensemble-Based Protein Design Algorithm to Optimize Stability and Binding Affinity over Large Sequence Spaces. *Journal of Computational Biology*, *25*(7), 726–739. https://doi.org/10.1089/cmb.2017.0267

Ojewole, A., Lowegard, A., Gainza, P., Reeve, S. M., Georgiev, I., Anderson, A. C., & Donald, B. R. (2017). OSPREY Predicts Resistance Mutations Using Positive and Negative Computational Protein Design. In I. Samish (Ed.), *Computational Protein Design* (pp. 291–306). New York, NY: Springer New York. https://doi.org/10.1007/978-1-4939-6637-0_15

Ollikainen, N., de Jong, R. M., & Kortemme, T. (2015). Coupling Protein Side-Chain and Backbone Flexibility Improves the Re-design of Protein-Ligand Specificity. *PLoS Computational Biology*, *11*(9), 1–22. https://doi.org/10.1371/journal.pcbi.1004335

Parmeggiani, F., Pellarin, R., Larsen, A. P., Varadamsetty, G., Stumpp, M. T., Zerbe, O., … Plückthun, A. (2008). Designed Armadillo Repeat Proteins as General Peptide-Binding Scaffolds: Consensus Design and Computational Optimization of the Hydrophobic Core. *Journal of Molecular Biology*, *376*(5), 1282–1304. https://doi.org/https://doi.org/10.1016/j.jmb.2007.12.014

Pawson, T., & Nash, P. (2003). Assembly of cell regulatory systems through protein interaction domains. *Science*, *300*(5618), 445–452. https://doi.org/10.1126/science.1083653

Peifer, M., Berg, S., & Reynolds, A. B. (1994). A repeating amino acid motif shared by proteins with diverse cellular roles. *Cell*, *76*(5), 789–791. https://doi.org/10.1016/0092-8674(94)90353-0

Perrimon, N., & Mahowald, A. P. (1987). Multiple functions of segment polarity genes in Drosophila. *Developmental Biology*, *119*(2), 587–600. https://doi.org/https://doi.org/10.1016/0012-1606(87)90061-3

Petsalaki, E., Stark, A., García-Urdiales, E., & Russell, R. B. (2009). Accurate prediction of peptide binding sites on protein surfaces. *PLoS Computational Biology*, *5*(3). https://doi.org/10.1371/journal.pcbi.1000335

Reichen, C., Hansen, S., & Plückthun, A. (2014). Modular peptide binding: From a comparison of natural binders to designed armadillo repeat proteins. *Journal of Structural Biology*, *185*(2), 147–162. https://doi.org/https://doi.org/10.1016/j.jsb.2013.07.012

Roberts, K. E., Cushing, P. R., Boisguerin, P., Madden, D. R., & Donald, B. R. (2012). Computational Design of a PDZ Domain Peptide Inhibitor that Rescues CFTR Activity. *PLOS Computational Biology*, *8*(4), 1–12. https://doi.org/10.1371/journal.pcbi.1002477

Rohl, C. A., Strauss, C. E. M., Misura, K. M. S., & Baker, D. (2004). Protein Structure Prediction Using Rosetta. In *Numerical Computer Methods, Part D* (Vol. 383, pp. 66–93). Academic Press. https://doi.org/https://doi.org/10.1016/S0076-6879(04)83004-0

Rosenbluth, M. N., & Rosenbluth, A. W. (1955). Monte carlo calculation of the average extension of molecular chains. *The Journal of Chemical Physics*, *23*(2), 356–359. https://doi.org/10.1063/1.1741967

Rossi, A. M., & Taylor, C. W. (2011). Analysis of protein-ligand interactions by fluorescence polarization. *Nature Protocols*, *6*(3), 365–387. https://doi.org/10.1038/nprot.2011.305

Rudicell, R. S., Kwon, Y. Do, Ko, S.-Y., Pegu, A., Louder, M. K., Georgiev, I. S., … Nabel, G. J. (2014). Enhanced potency of a broadly neutralizing HIV-1 antibody in vitro improves protection against lentiviral infection in vivo. *Journal of Virology*, *88*(21), 12669–12682. https://doi.org/10.1128/JVI.02213-14

Schilling, J., Jost, C., Ilie, I. M., Schnabl, J., Buechi, O., Eapen, R. S., … Forrer, P. (2022). Thermostable designed ankyrin repeat proteins (DARPins) as building blocks for innovative drugs. *The Journal of Biological Chemistry*, *298*(1), 101403. https://doi.org/10.1016/j.jbc.2021.101403

Schroeder, H. W. J., & Cavacini, L. (2010). Structure and function of immunoglobulins. *The Journal of Allergy and Clinical Immunology*, *125*(2 Suppl 2), S41-52. https://doi.org/10.1016/j.jaci.2009.09.046

Skerra, A. (2007). Alternative non-antibody scaffolds for molecular recognition. *Current Opinion in Biotechnology*, *18*(4), 295–304. https://doi.org/10.1016/j.copbio.2007.04.010

Smith, C. A., & Kortemme, T. (2008). Backrub-like backbone simulation recapitulates natural protein conformational variability and improves mutant side-chain prediction. *Journal of Molecular Biology*, *380*(4), 742–756. https://doi.org/10.1016/j.jmb.2008.05.023

Smith, G. P. (1985). Filamentous fusion phage: novel expression vectors that display cloned antigens on the virion surface. *Science (New York, N.Y.)*, *228*(4705), 1315–1317. https://doi.org/10.1126/science.4001944

Stark, Y., Menard, F., Jeliazkov, J. R., Ernst, P., Chembath, A., Ashraf, M., … Plückthun, A. (2024). Modular binder technology by NGS-aided, high-resolution selection in yeast of designed armadillo modules. *Proceedings of the National Academy of Sciences*, *121*(27), e2318198121. https://doi.org/10.1073/pnas.2318198121

Surpeta, B., Sequeiros-Borja, C. E., & Brezovsky, J. (2020). Dynamics, a powerful component of current and future in silico approaches for protein design and engineering. *International Journal of Molecular Sciences*, *21*(8). https://doi.org/10.3390/ijms21082713

Tarrant, M. K., & Cole, P. A. (2009). The chemical biology of protein phosphorylation. *Annual Review of Biochemistry*, *78*, 797–825. https://doi.org/10.1146/annurev.biochem.78.070907.103047

Tinberg, C. E., Khare, S. D., Dou, J., Doyle, L., Nelson, J. W., Schena, A., … Baker, D. (2013). Computational design of ligand-binding proteins with high affinity and selectivity. *Nature*, *501*(7466), 212–216. https://doi.org/10.1038/nature12443

Ubersax, J. A., & Ferrell, J. E. (2007). Mechanisms of specificity in protein phosphorylation. *Nature Reviews Molecular Cell Biology*, *8*(7), 530–541. https://doi.org/10.1038/nrm2203

Weidle, U. H., Auer, J., Brinkmann, U., Georges, G., & Tiefenthaler, G. (2013). The emerging role of new protein scaffold-based agents for treatment of cancer. *Cancer Genomics and Proteomics*, *10*(4), 155–168.

Wieschaus, E., & Riggleman, R. (1987). Autonomous requirements for the segment polarity gene armadillo during Drosophila embryogenesis. *Cell*, *49*(2), 177–184. https://doi.org/10.1016/0092-8674(87)90558-7

Zhang, Z., Shen, W. X., Liu, Q., & Zitnik, M. (2024). Efficient generation of protein pockets with PocketGen. *Nature Machine Intelligence*. https://doi.org/10.1038/s42256-024-00920-9

Appendix

A1 CoupledMoves Resfile.

NATRO
start
322 A NATAA
326 A NATAA
361 - 374 A NATAA
401 - 416 A NATAA
445 A NATAA
449 A NATAA
452 A NATAA
1 - 10 B NATAA
364 A ALLAA
368 A ALLAA
371 A ALLAA
403 A ALLAA
406 A ALLAA
407 A ALLAA
410 A ALLAA

A2 flex ddG XML file for pTyr mutation.

```
<ROSETTASCRIPTS>
 <SCOREFXNS>
  <ScoreFunction name="fa_talaris2014" weights="talaris2014"/>
  <ScoreFunction name="fa_talaris2014_cst" weights="talaris2014">
   <Reweight scoretype="atom_pair_constraint" weight="1.0"/>
   <Set fa_max_dis="9.0"/>
  </ScoreFunction>
 </SCOREFXNS>

 <!-- ### Only required input file (other than PDB) - mutation resfile ### --
>


 <!-- #### All residues must be set to be NATAA packable at top of resfile ###
-->
 <TASKOPERATIONS>
  <ReadResfile name="res_mutate" filename="%%mutate_resfile_relpath%%"/>
 </TASKOPERATIONS>
 <RESIDUE_SELECTORS>
  <Task name="resselector" fixed="0" packable="0" designable="1"
task_operations="res_mutate"/>
  <Neighborhood name="bubble" selector="resselector" distance="8.0"/>
  <PrimarySequenceNeighborhood name="bubble_adjacent" selector="bubble"
lower="1" upper="1"/>
  <StoredResidueSubset name="restore_neighbor_shell"
subset_name="neighbor_shell"/>
  <Not name="everythingelse" selector="restore_neighbor_shell"/>
 </RESIDUE_SELECTORS>
 <TASKOPERATIONS>
  <OperateOnResidueSubset name="repackonly"
selector="restore_neighbor_shell">
   <RestrictToRepackingRLT/>
  </OperateOnResidueSubset>
  <OperateOnResidueSubset name="norepack" selector="everythingelse">
   <PreventRepackingRLT/>
  </OperateOnResidueSubset>
```

```
  <UseMultiCoolAnnealer name="multicool" states="6"/>
   <ExtraChiCutoff name="extrachizero" extrachi_cutoff="0"/>
   <InitializeFromCommandline name="commandline_init"/>
   <RestrictToRepacking name="restrict_to_repacking"/>
 </TASKOPERATIONS>
 <FILTERS>
      Calculates the side-chain RMSD before and after simulation
      <SidechainRmsd name="rmsd" threshold="10" include_backbone="1"
res1_pdb_num="6" res2_pdb_num="6"/>
 </FILTERS>
 <MOVERS>
   <StoreResidueSubset name="neighbor_shell_storer"
subset_name="neighbor_shell" residue_selector="bubble_adjacent" />

 <AddConstraintsToCurrentConformationMover name="addcst"
use_distance_cst="1" coord_dev="0.5" min_seq_sep="0" max_distance="9"
CA_only="1" bound_width="0.0" cst_weight="0.0"/>
   <ClearConstraintsMover name="clearcst"/>
   <MinMover name="minimize" scorefxn="fa_talaris2014_cst" chi="1" bb="1"
type="lbfgs_armijo_nonmonotone" tolerance="0.000001"
max_iter="%%max_minimization_iter%%"
abs_score_convergence_threshold="%%abs_score_convergence_thresh%%"/>
   <PackRotamersMover name="repack" scorefxn="fa_talaris2014"
task_operations="commandline_init,repackonly,norepack,multicool"/>
   <MutateResidue name="mutate" target="6B" new_res="TYR:phosphorylated"/>
   <ReportToDB name="dbreport" batch_description="interface_ddG"
database_name="ddG.db3">
     <ScoreTypeFeatures/>
     <ScoreFunctionFeatures scorefxn="fa_talaris2014"/>
     <StructureScoresFeatures scorefxn="fa_talaris2014"/>
   </ReportToDB>
   <ReportToDB name="structreport" batch_description="interface_ddG_struct"
database_name="struct.db3">
     <PoseConformationFeatures/>
     <PdbDataFeatures/>
     <JobDataFeatures/>
```

```
<ResidueFeatures/>
     <PoseCommentsFeatures/>
     <ProteinResidueConformationFeatures/>
     <ResidueConformationFeatures/>
   </ReportToDB>
   <SavePoseMover name="save_wt_bound_pose" restore_pose="0"
reference_name="wt_bound_pose"/>
  <SavePoseMover name="save_backrub_pose" restore_pose="0"
reference_name="backrubpdb"/>
   <SavePoseMover name="restore_backrub_pose" restore_pose="1"
reference_name="backrubpdb"/>
   <InterfaceDdGMover name="int_ddG_mover"
wt_ref_savepose_mover="save_wt_bound_pose" chain_name="%%chainstomove%%"
db_reporter="dbreport" scorefxn="fa_talaris2014"/>

 <ScoreMover name="apply_score" scorefxn="fa_talaris2014_cst" verbose="0"/>
<!-- This ParsedProtocol allows the ddG calculation to take place multiple
times along the backrub trajectory, if desired -->
   <ParsedProtocol name="finish_ddg_post_backrub">
     <Add mover_name="save_backrub_pose"/>
     <Add mover_name="structreport"/>
     <Add mover_name="repack"/>
     <Add mover_name="addcst"/>
     <Add mover_name="minimize"/>
     <Add mover_name="clearcst"/>
     <Add mover_name="save_wt_bound_pose"/>
     <Add mover_name="structreport"/>
     <Add mover_name="restore_backrub_pose"/>
     <Add mover_name="mutate"/>
     <Add mover_name="addcst"/>
     <Add mover_name="minimize"/>
     <Add mover_name="clearcst"/>
     <Add mover_name="structreport"/>
     <Add mover_name="int_ddG_mover"/>
   </ParsedProtocol>
       Set side-chain moves to only include residues within 6 angstrom shell
```

```
<Sidechain name="sidechain"
task_operations="restrict_to_repacking,commandline_init,extrachizero"/>
   <BackrubProtocol name="backrub" mc_kt="1.2"
ntrials="%%number_backrub_trials%%"
pivot_residue_selector="restore_neighbor_shell"
task_operations="restrict_to_repacking,commandline_init,extrachizero"
recover_low="0" trajectory_stride="%%backrub_trajectory_stride%%"
trajectory_apply_mover="finish_ddg_post_backrub"/>
        During Monte Carlo, alternate between backrub moves (75%) and side-chain
moves (25%)
        <ParsedProtocol name="backrub_protocol" mode="single_random">
           <Add mover_name="backrub" apply_probability="0.75"/>
           <Add mover_name="sidechain" apply_probability="0.25"/>
        </ParsedProtocol>
   Set up Monte Carlo simulation with 10,000 steps and kT=0.6
      <GenericMonteCarlo name="backrub_mc" mover_name="backrub_protocol"
scorefxn_name="fa_talaris2014"/> backrubtrials="10000" temperature="0.6"
preapply="0"/>
 </MOVERS>
 <APPLY_TO_POSE>
 </APPLY_TO_POSE>
 <PROTOCOLS>
   <Add mover_name="addcst"/>
   <Add mover_name="apply_score"/> <!-- Necessary to initialize neighbor graph
-->
   <Add mover_name="neighbor_shell_storer"/>
   <Add mover_name="minimize"/>
   <Add mover_name="clearcst"/>
   Calculate RMSD before simulation
        <Add filter_name="rmsd"/>
        Run backrub simulation
        <Add mover_name="backrub_mc"/>
   Calculate RMSD after simulation
        <Add filter_name="rmsd"/>
 </PROTOCOLS>
 <OUTPUT />
</ROSETTASCRIPTS>
```

A3 OSPREY BBK* running file.

```
###
OSPREY 3.2
###
import osprey
from osprey import jvm
def show_energy_breakdowns(confAnalysis):
   t = jvm.getInnerClass(osprey.c.energy.ResidueForcefieldBreakdown, 'Type')
   print("Forcefield breakdown:
\n{}\n".format(confAnalysis.breakdownEnergyByPosition(t.All)))
   print("Electrostatic breakdown:
\n{}\n".format(confAnalysis.breakdownEnergyByPosition(t.Electrostatics)))
   print("van der Waals breakdown:
\n{}\n".format(confAnalysis.breakdownEnergyByPosition(t.VanDerWaals)))
   print("Solvation breakdown:
\n{}\n".format(confAnalysis.breakdownEnergyByPosition(t.Solvation)))
   print("Offsets breakdown:
\n{}\n".format(confAnalysis.breakdownEnergyByPosition(t.Offsets)))
osprey.start(heapSizeMiB=30000,
          enableAssertions=False,
          stackSizeMiB=None,
          garbageSizeMiB=10000,
          allowRemoteManagement=False,
   attachJvmDebugger=False
          )
# read a PDB file for molecular info
mol = osprey.readPdb('./osprey_Arg-binder.pdb')
# choose a forcefield
ffparams = osprey.ForcefieldParams()
#ffparams = osprey.ForcefieldParams(osprey.Forcefield.AMBER)

#ffparams.solvationForcefield = osprey.SolvationForcefield.PoissonBoltzmann
templateLib = osprey.TemplateLibrary(

    ffparams.forcefld,
  )
```

```
# define the protein strand
#protein = osprey.Strand(mol, templateLib=templateLib, residues=[‘178’, ‘514’])
#protein.flexibility[‘371’].setLibraryRotamers(osprey.WILD_TYPE).addWildTypeR
otamers()#.setContinuous()
#protein.flexibility['407'].setLibraryRotamers(osprey.WILD_TYPE).addWildTypeR
otamers()
protein = osprey.Strand(mol, templateLib=templateLib, residues=['179', '514'])
protein.flexibility['364'].setLibraryRotamers(osprey.WILD_TYPE).addWildTypeRo
tamers()#.setContinuous()
protein.flexibility['368'].setLibraryRotamers(osprey.WILD_TYPE).addWildTypeRo
tamers().setContinuous()
protein.flexibility['371'].setLibraryRotamers(osprey.WILD_TYPE).addWildTypeRo
tamers().setContinuous()
protein.flexibility['403'].setLibraryRotamers(osprey.WILD_TYPE).addWildTypeRo
tamers()
protein.flexibility['406'].setLibraryRotamers(osprey.WILD_TYPE).addWildTypeRo
tamers()#.setContinuous()
protein.flexibility['407'].setLibraryRotamers(osprey.WILD_TYPE).addWildTypeRo
tamers().setContinuous()
protein.flexibility['410'].setLibraryRotamers(osprey.WILD_TYPE).addWildTypeRo
tamers().setContinuous()
#protein.flexibility['414'].setLibraryRotamers(osprey.WILD_TYPE).addWildTypeR
otamers()#.setContinuous()
# define the ligand strand
ligand = osprey.Strand(mol, templateLib=templateLib, residues=['1', '10'])
#ligand.flexibility['6'].setLibraryRotamers(osprey.WILD_TYPE).addWildTypeRota
mers().setContinuous()
ligand.flexibility['6'].setLibraryRotamers(osprey.WILD_TYPE, 'ARG', 'SER',
'GLY', 'VAL', 'LEU', 'ILE', 'MET', 'PHE', 'TYR', 'TRP', 'ASN', 'HIS', 'ALA',
'ASP', 'GLU', 'GLN', 'LYS').addWildTypeRotamers().setContinuous()
# make the conf space for the protein
#bbflex1protein = osprey.c.confspace.CATSStrandFlex(protein,'367','372')
#bbflex2protein = osprey.c.confspace.CATSStrandFlex(protein,'406','414')

#proteinStrandAndFlex = [protein, bbflex2protein, bbflex1protein]

proteinStrandAndFlex = [protein]
```

```
proteinConfSpace = osprey.ConfSpace([proteinStrandAndFlex])
# make the conf space for the ligand
bbflex = osprey.c.confspace.CATSStrandFlex(ligand,'2','8')
ligandStrandAndFlex = [ligand, bbflex]
#ligandStrandAndFlex = [ligand]
ligandConfSpace = osprey.ConfSpace([ligandStrandAndFlex])
# make the conf space for the protein+ligand complex
complexConfSpace = osprey.ConfSpace([proteinStrandAndFlex,
ligandStrandAndFlex])
parallelism = osprey.Parallelism(cpuCores=16)
gpuParallelism = osprey.Parallelism(cpuCores=4, gpus=1, streamsPerGpu=1)
minimizingEcalc = osprey.EnergyCalculator(complexConfSpace, ffparams,
parallelism=parallelism, isMinimizing=True)
# BBK* needs a rigid energy calculator too, for multi-sequence bounds on K*
rigidEcalc = osprey.SharedEnergyCalculator(minimizingEcalc,
isMinimizing=False)
## define energies of conformations?
#def confEcalcFactory(confSpace, ecalc):
#       eref = osprey.ReferenceEnergies(confSpace, ecalc)
#       return osprey.ConfEnergyCalculator(confSpace, ecalc,
# configure BBK*
bbkstar = osprey.BBKStar(
        proteinConfSpace,
        ligandConfSpace,
        complexConfSpace,
        numBestSequences=18,
        epsilon=0.68, # you probably want something precise
        writeSequencesToConsole=True,
        writeSequencesToFile='bbkstar.results.tsv'
)
# configure BBK* inputs for each conf space
for info in bbkstar.confSpaceInfos():
 print ("### Started: {}".format(info))
     # how should we define energies of conformations?
     eref = osprey.ReferenceEnergies(info.confSpace, minimizingEcalc)
```

```
 # reference energy with residue entropy
      #eref = osprey.ReferenceEnergies(info.confSpace, minimizingEcalc,
addResEntropy=True)
      info.confEcalcMinimized = osprey.ConfEnergyCalculator(info.confSpace,
minimizingEcalc, referenceEnergies=eref)
      # with residue entropy
      #info.confEcalcMinimized = osprey.ConfEnergyCalculator(info.confSpace,
minimizingEcalc, referenceEnergies=eref, addResEntropy=True)
      # compute the energy matrix
      ematMinimized = osprey.EnergyMatrix(info.confEcalcMinimized,
cacheFile='emat.{}.dat'.format(info.id))
      def makeAStarMinimized(rcs, emat=ematMinimized):
         return osprey.AStarTraditional(emat, rcs, showProgress=False)
info.confSearchFactoryMinimized =
osprey.BBKStar.ConfSearchFactory(makeAStarMinimized)
     # BBK* needs rigid energies too
      confEcalcRigid =
osprey.ConfEnergyCalculatorCopy(info.confEcalcMinimized, rigidEcalc)
      ematRigid = osprey.EnergyMatrix(confEcalcRigid,
cacheFile='emat.{}.rigid.dat'.format(info.id))
      def makeAStarRigid(rcs, emat=ematRigid):
         return osprey.AStarTraditional(emat, rcs, showProgress=False)
      info.confSearchFactoryRigid =
osprey.BBKStar.ConfSearchFactory(makeAStarRigid)
      # how should we score each sequence?
      # (since we're in a loop, need capture variables above by using
defaulted arguments)
      def makePfunc(rcs, confEcalc=info.confEcalcMinimized,
emat=ematMinimized):
            return osprey.PartitionFunction(
                  confEcalc,
                  osprey.AStarTraditional(emat, rcs,
showProgress=False),
                  osprey.AStarTraditional(emat, rcs,
showProgress=False),
                  rcs
            )
```

```
      info.pfuncFactory = osprey.KStar.PfuncFactory(makePfunc)
import jpype

import jpype.imports
from jpype.types import *
import java.lang
# run BBK*
try:
   scoredSequences = bbkstar.run(minimizingEcalc.tasks)
except java.lang.Exception as ex:
   print(ex)
  ex.printStackTrace()
   exit(1)
# make a sequence analyzer from the configured KStar instance
# (you could also give it a configured BBKStar instance if you have that instead)
analyzer = osprey.SequenceAnalyzer(bbkstar)
# use results
numConfs = 500 # Number of conformations written in PDB output
for scoredSequence in scoredSequences:
        print("result:")
        print("\tsequence: {}".format(scoredSequence.sequence))
        print("\tK* score: {}".format(scoredSequence.score))
        # write the sequence ensemble, with up to numConfs of the lowest-energy
conformations
        analysis = analyzer.analyze(scoredSequence.sequence, numConfs)
        print(analysis)
        counter = 0
        for confAnalysis in analysis.ensemble.analyses:
                print ("####Note-counter: Sequence {}, Conformation
{}".format(scoredSequence.sequence, counter))
                show_energy_breakdowns(confAnalysis)
                counter+=1
        analysis.writePdb(
                'seq.{}.pdb'.format(scoredSequence.sequence),
                'Top {} conformations for sequence {}'.format(numConfs,
scoredSequence.sequence)
        )
```

A4. flex ddG running files for specificity evaluations.

```
Import socket
import sys
import os
import subprocess
use_multiprocessing = True
if use_multiprocessing:
   import multiprocessing
   max_cpus = 64
rosetta_scripts_path =
os.path.expanduser("/rosetta/3.12/main/source/bin/rosetta_scripts.static.linuxgccrelease")
nstruct = 250
max_minimization_iter = 5000
abs_score_convergence_thresh = 1.0
number_backrub_trials = 35000
backrub_trajectory_stride = 5000
path_to_script = 'ddG-backrub.xml'
residue_to_mutate = ('B', 6, '') # Residue position to perfrom saturation mutatagenesis. Format:
(Chain, PDB
residue number, insertion code).
def run_flex_ddg_saturation( name, input_path, input_pdb_path, chains_to_move, mut_aa,
nstruct_i
):
   output_directory = os.path.join( './output_saturation')
   if not os.path.isdir(output_directory):
      os.makedirs(output_directory)
   mutation_chain, mutation_resi, mutation_icode = residue_to_mutate
   resfile_path = os.path.join( output_directory, 'mutate_%s%d%s_to_%s.resfile' %
(mutation_chain,
mutation_resi, mutation_icode, mut_aa) )
   with open( resfile_path, 'w') as f:
      f.write( 'NATRO\nstart\n%d%s %s PIKAA %s\n' % (mutation_resi, mutation_icode,
mutation_chain,
mut_aa) )
   flex_ddg_args = [
      os.path.abspath(rosetta_scripts_path),
      "-s %s" % os.path.abspath(input_pdb_path),
      '-parser:protocol', os.path.abspath(path_to_script),
      '-parser:script_vars',
```

```
      'chainstomove=' + chains_to_move,

       'mutate_resfile_relpath=' + os.path.abspath( resfile_path ),
 'number_backrub_trials=%d' % number_backrub_trials,
       'max_minimization_iter=%d' % max_minimization_iter,
       'abs_score_convergence_thresh=%.1f' % abs_score_convergence_thresh,
       'backrub_trajectory_stride=%d' % backrub_trajectory_stride ,
       '-restore_talaris_behavior',
       '-in:file:fullatom',
       '-ignore_unrecognized_res',
       '-ignore_zero_occupancy false',
       '-ex1',
       '-ex2',
    ]
    log_path = os.path.join(output_directory, 'rosetta.out')
    print( 'Running Rosetta with args:' )
    print( ' '.join(flex_ddg_args) )
    print( 'Output logged to:', os.path.abspath(log_path) )
    print()
    outfile = open(log_path, 'w')
    process = subprocess.Popen(flex_ddg_args, stdout=outfile, stderr=subprocess.STDOUT,
close_fds
= True,
cwd = output_directory)
    returncode = process.wait()
  outfile.close()
if __name__ == '__main__':
    mutation_chain, mutation_resi, mutation_icode = residue_to_mutate
    cases = []
    for nstruct_i in range(1, nstruct + 1 ):
       for case_name in os.listdir('inputs'):
          case_path = os.path.join( 'inputs', case_name )
          for f in os.listdir(case_path):
             if f.endswith('.pdb'):
                input_pdb_path = os.path.join( case_path, f )
                break
          with open( os.path.join( case_path, 'chains_to_move.txt' ), 'r' ) as f:
             chains_to_move = f.readlines()[0].strip()
```

```
          for mut_aa in 'ACDEFGHIKLMNPQRSTVWY':
             cases.append( ('%s_%s%d%s' % (case_name, mutation_chain, mutation_resi,
mutation_icode),

case_path,
input_pdb_path, chains_to_move, mut_aa, nstruct_i) )
   if use_multiprocessing:
  pool = multiprocessing.Pool( processes = min(max_cpus, multiprocessing.cpu_count()) )
   for args in cases:
      if use_multiprocessing:
         pool.apply_async( run_flex_ddg_saturation, args = args )
      else:
         run_flex_ddg_saturation( *args )
   if use_multiprocessing:
      pool.close()
      pool.join()
```

A5. Experimental binding affinity data for binders.

Table 22: Experimental binding affinity data for Arg-binder used for comparison with calculated scores.

Arg-binder (QWSQQEW)		
Peptide	K_D [nM]	Error [nM]
R	2.787	0.85
K	2.85	0.44
Q	37.245	37.245
V	41.055	25.52
A	57.57	4.79
S	57.64	13.0
H	61.8	23.48
I	63.63	4.14
G	65.665	24.19
Y	82.52	1.94
F	105.92	29.81
L	113.95	1.63
M	128.64	44.92
N	136.1	23.90
W	176.3	26.45
D	1334	651.95
E	1343.5	323.15

Table 23: Experimental binding affinity data for Tyr-binder used for comparison with calculated scores.

Tyr-binder (KEVLIRQ)		
Peptide	K_D [nM]	Error [nM]
Y	5	-
W	62	-
M	111	-
H	127	-
F	148	-
R	162	-
I	199	-
L	327	-
A	636	-
V	731	-
T	1296	-

Table 24: Experimental binding affinity data for Trp-binder used for comparison with calculated scores.

Trp-binder (TATAWRT)		
Peptide	K_D [nM]	Error [nM]
W	64	21
Y	135	11
F	176	15
H	218	73
I	223	42
N	230	-
T	236	46
L	241	119
V	340	80
R	428	28
M	435	-
S	500	-
A	806	-

Table 25: Experimental binding affinity data for Ile-binder used for comparison with calculated scores.

Ile-binder (FALYDRV)		
Peptide	K_D [nM]	Error [nM]
I	28	1
L	43	8
M	61	-
V	73	34
R	93	-
Y	90	12
W	155	36
A	173	-
H	227	63
F	241	150
T	325	100
Q	470	-
S	507	-

Table 26: Experimental binding affinity data for His-binder used for comparison with calculated scores.

His-binder (DYTDWQA)		
Peptide	K_D [nM]	Error [nM]
H	8	-
R	104	-
Y	292	-
M	425	-
W	430	-
A	436	-
N	727	-
I	740	-
T	930	-

A6. Correlation between predicted and experimentally determined binding specificities.

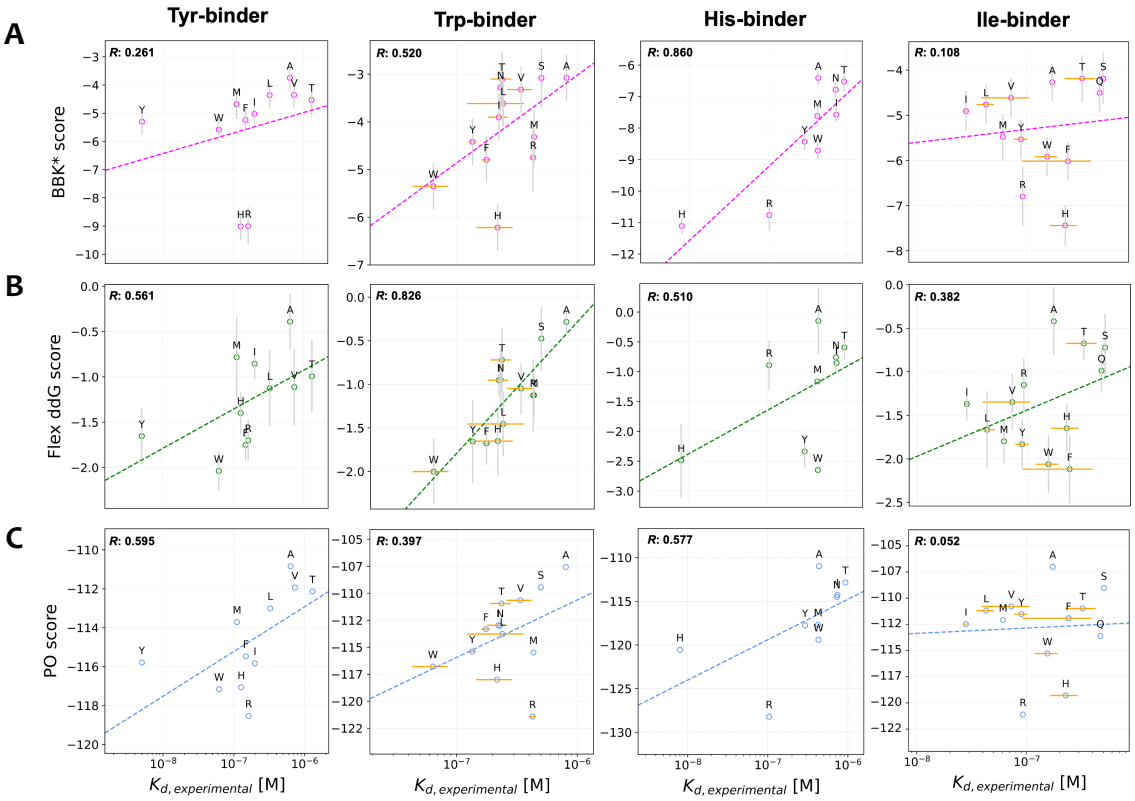

Figure 21: Correlation between calculated binding specificity predictions and experimental binding specificities for the Tyr, Trp, His, and Ile binding pockets using 5AEI as scaffold. Correlation between experimental measurements for each binder with the calculations from BBK* (A), from flex ddG (B) and from PocketOptimizer (C) are given with their corresponding Pearson correlations (Taken from Ayyildiz et al., 2024).

A7. Correlation between predicted and experimentally determined binding specificities.

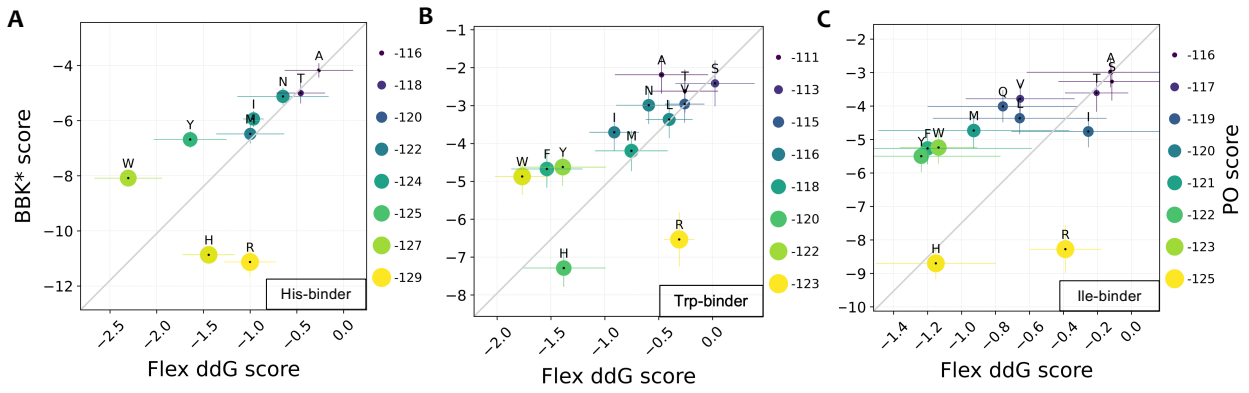

Figure 22: Correlation of specificity predictions from all three methods. BBK*, flex ddG, and PocketOptimizer predictions for (A) His and (B) Trp and (C) Ile binders were obtained using the crystal structure 6SA8 as the scaffold.

A8 Primers.

Table 27: Used primers for peptide variants.

Primer Name	Primer Sequence (5'-3')
QA6Q-sfGFP-forward	CCAAACGCAAGCGTAAGCAGGCACGTCAGCGCGGCG
QA6Q-sfGFP-reverse	CGCCGCGCTGACGTGCCTGCTTACGCTTGCGTTTGG
QA6E-sfGFP-forward	CCAAACGCAAGCGTAAGGAAGCACGTCAGCGCGGCG
QA6E-sfGFP-reverse	CGCCGCGCTGACGTGCTTCCTTACGCTTGCGTTTGG
QA6Y-sfGFP-forward	CCAAACGCAAGCGTAAGTATGCACGTCAGCGCGGCG
QA6Y-sfGFP-reverse	CGCCGCGCTGACGTGCATACTTACGCTTGCGTTTGG
QA6L-sfGFP-forward	CCAAACGCAAGCGTAAGCTGGCACGTCAGCGCGGCG
QA6L-sfGFP-reverse	CGCCGCGCTGACGTGCCAGCTTACGCTTGCGTTTGG
QA6N-sfGFP-forward	CCAAACGCAAGCGTAAGAACGCACGTCAGCGCGGCG
QA6N-sfGFP-reverse	CGCCGCGCTGACGTGCCAGCTTACGCTTGCGTTTGG
QA6S-sfGFP-forward	CCAAACGCAAGCGTAAGAGCGCACGTCAGCGCGGCG
QA6S-sfGFP-reverse	CGCCGCGCTGACGTGCGCTCTTACGCTTGCGTTTGG
QA6M-sfGFP-forward	CCAAACGCAAGCGTAAGATGGCACGTCAGCGCGGCG
QA6M-sfGFP-reverse	CGCCGCGCTGACGTGCCATCTTACGCTTGCGTTTGG

Acknowledgements

The biggest thanks goes to my supervisor, Prof. Dr. Birte Höcker, who gave me the opportunity to gain so many insights into scientific work, and the opportunity to grow independently. I truly appreciate her support and the time she invested in me.

I would like to also thank the committee members for their time and consideration in evaluating my dissertation.

I want to thank my colleagues for a great time and many interesting discussions. Special thanks to Noelia for the warm welcome, Josef and Florian for their insightful discussions and motivation. I am particularly grateful to Jakob, not only for his contributions to my work but also for his invaluable support in editing my thesis and for his friendship, which made this journey all the more enjoyable. I am also grateful to Steffen for his technical expertise and willingness to help at every stage. I feel fortunate to have worked with such great people and to be part of a supportive research environment. Many thanks to both current and former lab members, in no particular order, Sabrina, Onur, Sergio, Sina, Sooruban, Pascal, Florian M., Emily, Andi, Lukas, Julian, Guto, Surbhi, Abhishek, Basti, Vanessa, Mark, Marc, Olivier, Boje, Anna, Katha, Johanna, Gabi, Anke, Ina.

I extend my gratitude to our collaboration partners, particularly Prof. Andreas Plückthun, for welcoming me into his group in Zürich. Special thanks to Erich, Bastian, Songyuan, Peter L., Yvonne, Gwen and Jonas for their warm welcome and valuable discussions. Additionally, I appreciate Prof. Anna Hine, as well as Anu and Mo, for their discussions and kindness throughout this journey.

A very special thanks goes to Elena Köstner and Silvia whose support was essential in completing this thesis. I truly appreciate their help and encouragement throughout this journey.

Many thanks to all my friends, special thanks to Serkan, Melis, Sevilay, Burak, Fatih and Christian for their unwavering support, one could not ask for better friends.

Last but not least, my deepest gratitude goes to my parents and my sister, Deniz, for their unconditional love and support.

This thesis is the result of not just my efforts but the collective support and guidance of many incredible individuals. I am truly thankful to each and every one of them.

(Eidesstattliche) Versicherungen und Erklärungen

Hiermit versichere ich eidesstattlich, dass ich die Arbeit selbstständig verfasst und keine anderen als die von mir angegebenen Quellen und Hilfsmittel benutzt habe (vgl. Art. 97 Abs. 1 Satz 8 BayHIG) (§ 8 Satz 2 Nr. 3).

Hiermit erkläre ich, dass ich die Dissertation nicht bereits zur Erlangung eines akademischen Grades eingereicht habe und dass ich nicht bereits diese oder eine gleichartige Doktorprüfung endgültig nicht bestanden habe (§ 8 Satz 2 Nr. 3).

Hiermit erkläre, dass ich Hilfe von gewerblichen Promotionsberatern bzw. –vermittlern oder ähnlichen Dienstleistern weder bisher in Anspruch genommen wurde noch künftig in Anspruch genommen wird (§ 8 Satz 2 Nr. 4).

Hiermit erkläre ich mein Einverständnis, dass die elektronische Fassung der Dissertation unter Wahrung meiner Urheberrechte und des Datenschutzes einer gesonderten Überprüfung unterzogen werden kann (§ 8 Satz 2 Nr. 7).

Hiermit erkläre ich mein Einverständnis, dass bei Verdacht wissenschaftlichen Fehlverhaltens Ermittlungen durch universitätsinterne Organe der wissenschaftlichen Selbstkontrolle stattfinden können (§ 8 Satz 2 Nr. 8).

_____

Ort, Datum, Unterschrift