UNIVERSITÄT
BAYREUTH

# Towards Modular Protein Detection
## Computational Methods to Support the Design of Peptide-Binding Pockets

**Dissertation**

Zur Erlangung des akademischen Grades eines
Doktors der Naturwissenschaften (Dr. rer. nat.)
an der Fakultät für Biologie, Chemie und Geowissenschaften
der Universität Bayreuth

Vorgelegt von

**Josef Paul Kynast**

aus Bayreuth

Bayreuth, 2023

Die vorliegende Arbeit wurde in der Zeit von Februar 2018 bis September 2023 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 eines Doktors der Naturwissenschaften (Dr. rer. nat.).

Art der Dissertation: Kumulative Dissertation

Dissertation eingereicht am: 19.09.2023

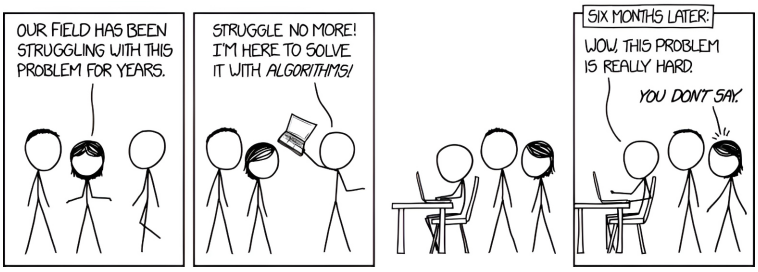
Zulassung durch die Promotionskommission: 27.09.2023

Wissenschaftliches Kolloquium: 15.01.2024

Amtierender Dekan: Prof. Dr. Cyrus Samimi

Prüfungsausschuss:

Prof. Dr. Birte Höcker (Gutachterin)
Prof. Dr. Matthias Ullmann (Gutachter)
Prof. Dr. Janosch Hennig (Vorsitz)
JProf. Dr. Meike Leiske




https://xkcd.com/1831/

# Contents

I. Nomenclature ..... V

II. Abstract ..... VII

III. Zusammenfassung ..... IX

IV. Introduction ..... 1

Prologue: How to Tame a Protein ..... 1

1. Proteins Are Controlled by Their Sequence ..... 1

From Sequence to Structure ..... 2

Conformational Complexity ..... 2

From Sequence to Function ..... 5

2. Protein Evolution as a Source of Inspiration ..... 6

Evolution of Functionality ..... 6

3. Protein Engineering and Design ..... 7

The Inverse Folding Problem ..... 7

*De Novo* Protein Design ..... 7

Protein Engineering ..... 8

Computational Guidance in Protein Design ..... 9

Rosetta ..... 9

OSPREY ..... 11

4. Engineering Function: Ligand Binding ..... 13

Design of a Modular Binding Reagent ..... 15

V. Aim ..... 18

VI. Synopsis ..... 19

Learning from Nature ..... 20

ATLIGATOR Web Server Enables Easy Usage ..... 22

Prediction of Binder Specificity ..... 24

Pipeline for Design of Binding Modules ..... 26

VII. Author Contributions ..... 29

VIII. Bibliography ..... 31

IX. Research Articles ..... 41

1. Modular Peptide Binders:
Development of a Predictive Technology as Alternative for Reagent Antibodies ..... 41

2. ATLIGATOR:
Editing Protein Interactions with an Atlas-Based Approach ..... 51

3. Atligator Web:
A Graphical User Interface for Analysis and Design of Protein–Peptide Interactions ..... 59

4. PocketOptimizer 2.0:
A modular framework for computer-aided ligand-binding design ..... 66

X. List of Publications ..... 79

Acknowledgements ..... 81

(Eidesstattliche) Versicherungen und Erklärungen ..... 83

# I. Nomenclature

| | |
|---|---|
| **ATLIGATOR** | **AT**las-based **LIGA**nd binding edi**TOR** |
| **BBK*** | **B**ranch and **b**ound over **K*** |
| **dArmRP** | **D**esigned **arm**adillo **r**epeat **p**rotein |
| **DEE** | **D**ead **e**nd **e**limination |
| **DNA** | **D**eoxyribo**n**ucleic **A**cid |
| **GMEC** | **G**lobal **m**inimum **e**nergy **c**onformation |
| **HTMD** | **H**igh-**t**hroughput **m**olecular **d**ynamics |
| **MARK*** | **M**inimization-**a**ware enumeration and **r**ecursive **K*** |
| **MD** | **M**olecular **d**ynamics |
| **NMR** | **N**uclear **m**agnetic **r**esonance |
| **PDB** | **P**rotein **D**ata**B**ank |
| **PRe-ART** | **P**redictive **re**agent **a**ntibody **r**eplacement **t**echnology |
| **RNA** | **R**ibo**n**ucleic **a**cid |

# II. Abstract

Our growing knowledge of the diverse ways in which proteins function has sparked the interest to reshape the protein world early on. The emergence of advanced computational and molecular biology techniques are catalysts for creative engineering of proteins - from building macromolecular structures from first principles to optimizing functional sites. Despite powerful methods to model and redesign proteins computationally, current methods struggle to reliably predict mutations to alter highly influential regions of protein structures like ligand binding sites. Nature has evolved robust binding sites on proteins in the course of millions of years of evolution. A growing number of structures is available in protein structure databases, which could be used to find, extract, and reuse highly evolved binding motifs in engineering applications.

One such application is the establishment of a new peptide-binding reagent. Common protein detection relies mainly on antibodies which are derived from costly and ethically questionable immunization of mice. Moreover, it has been shown that commercially available reagent antibodies lack specificity and reproducibility (Bradbury & Plückthun, 2015; A. Gray et al., 2020). Thus, there is a need for alternative detection reagents. With the regularization of designed armadillo repeat proteins (dArmRP), a modular binding system was proposed to serve this purpose. These dArmRPs have been designed to regularly bind peptides in an extended fashion (Hansen et al., 2016). Each of the peptide side chains is detected by a specific binding pocket on the dArmRP. With the design of new binding pockets for all canonical or even post-translationally modified amino acids a pocket catalogue can be assembled. By recombining binding pocket modules for the targeted peptide residues, this system could deliver reliable and cheap alternative detection reagents (Gisdon et al., 2022).

This work introduces the software ATLIGATOR, ATLIGATOR web and PocketOptimizer 2.0, which all provide a significant support to the design of new binding modules for armadillo repeat proteins, or even other protein systems. ATLIGATOR extracts frequent interactions found in known protein structures which can be transferred to any protein scaffold. This transfer process could yield new or improved binder proteins. Moreover, an itemset mining algorithm detects frequent groups of interactions that can act as generalizable motifs. With a grafting functionality such motifs can be directly introduced in the corresponding ligand-binding sites. ATLIGATOR web extends this functionality with a user-friendly web interface to enhance the analysis and design process. An advanced design tool provides immediate visual feedback of the design process as well as features like manual mutations and Rosetta side chain repacking. Such

designs can be fed directly into protein redesign software for additional optimization of the binding capabilities. PocketOptimizer 2.0, as one example of such software, is the successor of PocketOptimizer that introduces beneficial mutations on small molecule-bindings sites. With this iteration, PocketOptimizer was modernized by removing deprecated dependencies and rewriting the code base in developer-friendly Python programming language. Version 2.0 also extends the functionality with a new user interface, more force fields and scoring functions as well as an advanced rotamer library.

This set of programs not only provides critical support to start the design of new binding pockets for the armadillo repeat system but is also applicable in other protein design approaches.

# III. Zusammenfassung

Unser wachsendes Wissen über die vielfältigen Funktionsweisen von Proteinen hat schon früh das Interesse geweckt, die Proteinwelt selbst zu gestalten. Das Aufkommen fortschrittlicher computergestützter und molekularbiologischer Techniken ist Antreiber für die kreative Arbeit an Proteinen - vom Aufbau makromolekularer Strukturen basierend auf grundlegenden Prinzipien bis hin zur Optimierung von funktionellen Regionen. Trotz leistungsfähiger Methoden zur computergestützten Modellierung und Neugestaltung von Proteinen ist es derzeit schwierig, Mutationen zuverlässig vorherzusagen, um einflussreiche Regionen von Proteinstrukturen wie Ligandenbindungsstellen zu verändern. Die Natur hat im Laufe von Millionen von Jahren Evolution robuste Bindungsstellen in Proteinen entwickelt. In Proteinstrukturdatenbanken ist eine wachsende Anzahl von Strukturen verfügbar, die verwendet werden können, um gut angepasste Bindungsmotive zu finden, zu extrahieren und in technischen Anwendungen wiederzuverwenden.

Eine solche Anwendung ist die Etablierung eines neuen Peptid-bindenden Reagenz. Der gängige Proteinnachweis beruht hauptsächlich auf Antikörpern, die aus einer kostspieligen und ethisch fragwürdigen Immunisierung von Mäusen stammen. Darüber hinaus konnte gezeigt werden, dass es kommerziell erhältlichen Reagenzien-Antikörpern an Spezifität und Reproduzierbarkeit mangelt, was zu problematischen experimentellen Ergebnissen führt (Bradbury & Plückthun, 2015; A. Gray et al., 2020). Daher besteht ein Bedarf an alternativen Nachweisreagenzien. Mit der Regularisierung von designten *Armadillo-Repeat*-Proteinen (dArmRP) wurde ein modulares Bindungssystem vorgeschlagen, um diesen Zweck zu erfüllen. Diese dArmRPs wurden entwickelt, um Peptide in einer gestreckten Form zu binden (Hansen et al., 2016). Jede der Peptidseitenketten wird durch eine spezifische Bindungstasche auf dem dArmRP detektiert. Mit dem Design neuer Bindungstaschen für alle kanonischen oder auch posttranslational modifizierten Aminosäuren kann ein Katalog von Bindungsmodulen zusammengestellt werden. Durch die Rekombination von Bindungstaschenmodulen für die Seitenketten der Zielsequenz könnte dieses System zuverlässige und kostengünstige alternative Nachweisreagenzien liefern (Gisdon et al., 2022).

In der vorliegenden Arbeit werden die Software ATLIGATOR, ATLIGATOR web und PocketOptimizer 2.0 vorgestellt, die alle eine wesentliche Unterstützung für das Design neuer Bindungsmodule für *Armadillo-Repeat*-Proteine – oder sogar andere Proteinsysteme - bieten. ATLIGATOR extrahiert häufige Wechselwirkungen, die in bekannten Proteinstrukturen zu finden sind und auf Proteingerüste übertragen werden können. Dieser Transferprozess könnte

zu neuen oder verbesserten Bindeproteinen führen. Darüber hinaus erkennt ein *Itemset-Mining*-Algorithmus häufige Gruppen von Interaktionen, die verallgemeinerbare Motive darstellen können. Mit einer Transferfunktion können solche Motive direkt in die entsprechenden Ligandenbindungsstellen eingebracht werden. ATLIGATOR web erweitert diese Funktionalität um eine benutzerfreundliche Weboberfläche, um den Analyse- und Designprozess zu verbessern. Ein fortschrittliches Design-Werkzeug bietet sofortiges visuelles Feedback zum Design-Prozess sowie Funktionen wie manuelle Mutationen und das Umpacken von Rosetta-Seitenketten. Solche Designs könnten direkt in Protein-Anpassungs-Software eingespeist werden, um die Bindungsfähigkeiten weiter zu optimieren. PocketOptimizer 2.0, als ein Beispiel für eine solche Software, ist der Nachfolger von PocketOptimizer, der in der Lage ist, vorteilhafte Mutationen an Bindungsstellen für kleine Moleküle zu identifizieren. Mit dieser Iteration wurde PocketOptimizer modernisiert, indem veraltete Abhängigkeiten entfernt und die Codebasis in der entwicklerfreundlichen Programmiersprache Python neu geschrieben wurde. Die Version 2.0 erweitert den Funktionsumfang um eine neue Benutzeroberfläche, mehr Kraftfelder und *Scoring*-Funktionen sowie eine erweiterte Rotamer-Bibliothek.

Diese Softwareanwendungen bieten nicht nur eine wichtige Unterstützung bei der Entwicklung neuer Bindungstaschen für das *Armadillo-Repeat*-system, sondern sind auch in anderen Proteindesignansätzen anwendbar.

# IV. Introduction

## Prologue: How to Tame a Protein

In times of global challenges, scientific progress can be a driver for innovative approaches. Proteins have been proven to be versatile molecular machines that can even be repurposed to tackle some of these challenges in medicine and biotechnology. To embrace this approach of engineering proteins to serve our needs, we need to obtain a deep understanding of what influences a protein to behave as we desire. In the following chapters, I will navigate through what we know and do not know about proteins and how this knowledge can be utilized to engineer a dedicated function into a protein.

## 1. Proteins Are Controlled by Their Sequence

Proteins are pivotal factors in life as they are employed in a myriad of cellular processes. They can act as transporters and anchors, reporters and detectors, barriers and enclosures and even molecular factories. This variety is defined by their modularity which is based on 20 canonical amino acids that share a primary amine and a carbon acid group. By forming amide bonds – also referred to as peptide bonds in this context – they construct a linear polypeptide chain. While the backbone of this chain is almost independent of the amino acid combination, each amino acid harbours a different residual group called the side chain. The combination of amino acids - and thus side chains - in the polypeptide chain defines the sequence of a protein. Consequently, the immense variety of proteinogenic features is encoded in the protein sequence (as depicted in Figure 1).

## From Sequence to Structure

Today, we can see proteins as three-dimensional arrangements of the linear amino acid sequence they are composed of. With techniques like x-ray crystallography, nuclear magnetic resonance (NMR) spectroscopy or cryo-electron microscopy, researchers have solved the structures of hundreds of thousands of proteins or protein-ligand complexes (Berman et al., 2000). Besides that, recent advances in computational methodologies for structure prediction yielded hundreds of millions of protein structure models that add up to existing experimental results (David et al., 2022; Lin et al., 2023; Varadi et al., 2022). What can be observed in the corresponding protein structure databases is a gigantic variety of three-dimensionally folded structures. Assuming the observed conformation is the native one, each of those structures is a result of the protein sequence, as proposed by Anfinsen:

> *"the three-dimensional structure of a native protein in its normal physiological milieu [...] is the one in which the Gibbs free energy of the whole system is the lowest; that is, that the native conformation is determined by the totality of the inter-atomic interactions and hence by the amino acid sequence"* (Anfinsen, 1973)

Today, this native conformation could be described as a native ensemble of conformations since proteins that are able to adopt multiple conformations have been described (Cordes et al., 2000; Meier et al., 2007; Nowak et al., 2014; Van Dorn et al., 2006), but the general idea is still valid: Despite the knowledge that other factors like post-translational modifications or ligand binding events can impact the structural integrity of a protein, the linear amino acid sequence alone is the single major factor deciding how a protein should look and act like.

## Conformational Complexity

To realize that this sequence-structure relationship is not trivial, it is important to understand how a protein can adopt a range of different conformations: The linear chain of amino acid building blocks comprises several degrees of freedom allowing the molecule to alter the shape. Without breaking any covalent bonds, there are two main contributors to this source of flexibility:

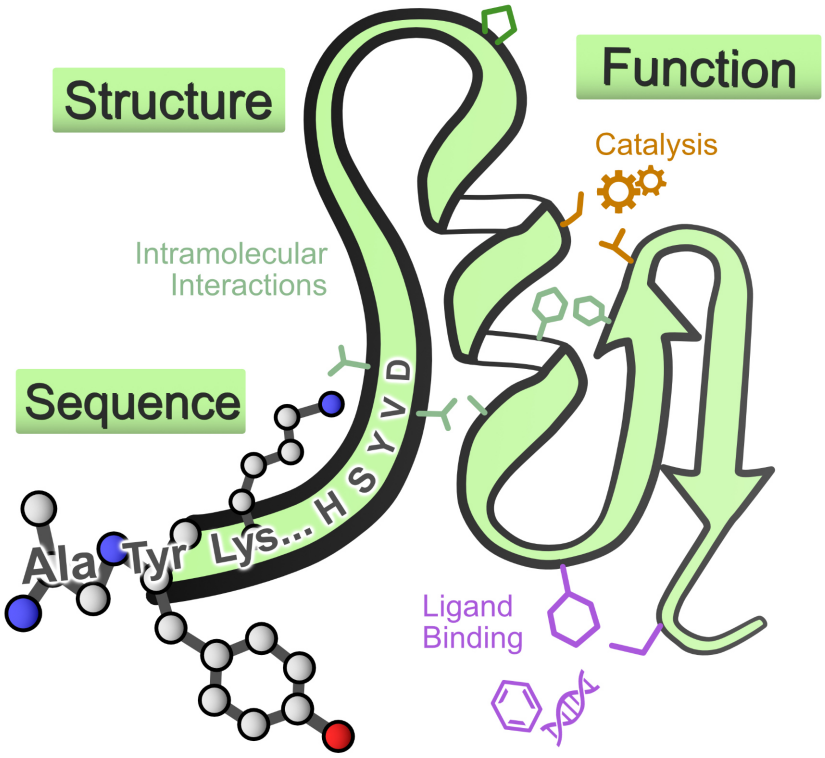



**Figure 1: Visualization of the relationship between protein sequence, structure, and function.** The sequence of amino acids determines structural features by influencing secondary structure formation or forming internal interactions of residues that are distant in sequence. Through forming the structural framework and providing distinct chemical properties at crucial positions, the amino acid sequence also provides function. This function could be to bind small ligands and other proteins or form an active site for a catalytic function.

First, the backbone of the chain is flexible. It harbours nitrogen, the carbon α atom linked to the side chain and the carbonyl carbon in a repetitive manner for each residue of the polypeptide. Thus, there are three dihedral angles – spanning a connection between four backbone atoms - per residue alongside the polypeptide chain. Since the peptide bond is mostly planar due to a double-bond character of the amide, we can find two variable angles per residue. These angles are centred at the bonds between N-Cα or Cα-CO and called phi (ϕ) and psi (ψ). The ϕ-ψ-combination of a residue is restricted to allowed regions based on the amino acid type and the secondary structure. These regions are often defined in a so-called Ramachandran plot where both torsion angles are plotted against each other (Ramachandran et al., 1963).

Second, most amino acid side chains feature rotatable bonds. The associated torsion angles are called chi (χ) and incremented by their number of covalent bonds from the α-carbon. Depending on the amino acid type this adds up to five χ torsion angles per residue as additional degrees of freedom.

Even though not all combinations of torsion angles are biologically relevant or even physically possible, the enormous number of possible torsion angle combinations is far from human imagination – and grows exponentially with each additional protein residue.

According to Anfinsen, there is only one or few native structures of a polypeptide that are formed as the needle in the haystack of all possible torsion angle combinations. Since the folding of proteins usually falls in the time range of micro- or milliseconds nature cannot scan through all conformational possibilities exhaustively. This mismatch is described as Levinthal's paradox of folding speeds and suggests the existence of specific folding pathways (Levinthal, 1969). While Anfinsen describes the thermodynamic element of the sequence-structure relationship, Levinthal handles kinetic considerations of the same problem. This problem is also referred to as the protein folding problem (Nassar et al., 2021).

Despite this conformational complexity, it is clear that intramolecular interactions and interactions with the environment are the main contributions for the formation of protein structures. Intramolecular interactions can be formed (1) between two or more amino acid side chains, (2) between a side chain and the protein backbone, and (3) between different parts of the protein backbone. An obvious example is based on backbone-backbone hydrogen bonds, namely the formation of secondary structure elements - β-sheet and α-helix structures (reviewed in Eisenberg, 2003). Turns and loop structures connect α-helices and β-strands (Sun et al., 2004). While these secondary structure elements are mostly formed between backbone atoms, their appearance is defined by the amino acid sequence. For example, glycines and prolines are known as helix breakers, a helix-induced dipole influences the frequency and positioning of charged amino acids and loop structures are dominated by glycines (Argos & Palau, 1982; Aurora et al., 1994; Kim & Kang, 1999).

Amino acid side chain interactions are manyfold. Charge-charge interactions, hydrogen bonds and π- π or ion- π interactions are crucial influences on stability of super-secondary structures. Moreover, polar, and non-polar residues are unevenly distributed in surface and core regions of protein structures due to the hydrophobic effect. This way hydrophobic side chains are shielded from the polar aqueous solvent. In fact, there is evidence for a stabilizing effect of expanded networks of hydrophobic residues in protein cores (Arunachalam & Gautham, 2008; Kathuria et al., 2016; Romero-Romero et al., 2021). Finally, all these factors contribute to the three-dimensionally folded structure (see Figure 1).

## From Sequence to Function

Since the sequence defines the protein structure, every function that is structurally defined is induced by the sequence-to-structure relationship (see Figure 1). Functionality in the protein space is highly diverse, ranging from purely structural modules to interactions with ligands or even catalysis. Moreover, individual proteins or protein complexes often perform multiple purposes: Microtubules for example not only maintain the cell shape as a major component of the cytoskeleton, but are also involved in mitosis, cell motility and intracellular transport (Akhmanova & Steinmetz, 2008).

In analogy to internal interactions that stabilise a protein fold, the amino acid sequence is the main driver for these functions: On the one hand, the protein sequence defines the three-dimensional structure, dynamics, and stability to form building blocks in living cells. On the other hand, it dictates the placement of crucial amino acids for forming catalytically active sites and binding sites. This relationship is evident in many mutation-derived diseases where single amino acid mutations lead to severe health conditions (Veltman & Brunner, 2012). Understanding the connection of sequence, structure and function helps to target these diseases. Furthermore, it helps to modulate or generate a desired functionality by designing proteins via adaptation or *de novo* design. In order to extend this understanding, a great source of such knowledge can be found in nature.

# 2. Protein Evolution as a Source of Inspiration

The variety of features in today's proteins is based on constant adaptation as described by Darwin's theory of evolution. As a process of slight modifications while being influenced by external factors this often leads to an improved fitness in even the smallest biological niche. Thus, the variety of protein sequences and structures is a consequence of highly diverse environments. As a result, each natural protein carries useful information about its origin or function.

## Evolution of Functionality

Proteins did not evolve in an isolated space but embedded in a heterogeneous environment. Thus, their evolution was always in close coexistence with potential binding partners, substrates, or even structural violators. Proteolytic activity, for example, needs to be tuned down to a beneficial level and improved catalytic activity could be the existential feature in competition with the environment (Attaix et al., 2001). In the case of ligand binding, increasing affinity to the target might help to survive in the case of a shortage period. Equally important, the specificity to distinguish binding partners from each other is an important part of the overall effectivity of ligand binders. Over the course of evolution all this information was accumulated in existing protein sequences and structures. Hence, known protein sequences or structures are a fruitful source of inspiration when looking for new functionality in protein design. Even though it is hidden, the vast amount of available data facilitates the identification of frequently used and potentially valuable patterns.

To find similar functional features in different proteins, the protein sequences can be aligned and compared. However, even very sensitive sequence comparisons using hidden Markov models are less powerful than structural comparisons because structural features are highly conserved. With the use of structural databases based on evolutionary relationships (Fox et al., 2014), common structural features can be revealed and linked to specific functionality (Bordin et al., 2021; Todd et al., 1999).

# 3. Protein Engineering and Design

The more we understand the relationship between protein sequence and structure as well as their implications on function, the better we can use this knowledge for our own purposes. By applying alterations to a protein sequence, we can try to establish or modulate binding capabilities, and change protein stability or enzymatic features. When it comes to engineering a protein to exhibit a desired function, structure-based protein design with computational guidance is certainly one of the most promising applications (Gainza et al., 2016). It ranges from the exploration of new protein folds to repurposing known scaffolds to feature new or improved capabilities (Pan & Kortemme, 2021).

## The Inverse Folding Problem

While modern software sometimes still struggles to correctly predict the structure of a known protein sequence – relating to the protein folding problem described earlier, the inverse process is even more challenging. The definition of a sequence from a given structure – referred to as the inverse folding problem – requires a reliable understanding of the sequence to structure relationship. Additionally, this relationship must be assessed for every potential protein sequence – in contrast to structure prediction from one protein sequence. About four decades ago, the rational design of a protein with the aim to exhibit a certain fold and functionality seemed almost impossible (Korendovych & DeGrado, 2020). This was of course due to the combinatorial explosion of potential amino acid sequences even for a moderately long amino acid sequence. Technology has however been developed even to design proteins from scratch (Pan & Kortemme, 2021).

## *De Novo* Protein Design

Today, there are numerous examples of proteins that have been designed based on physical principles while the amino acid sequences are unrelated to known proteins (Huang, Boyken, et al., 2016). These proteins are also called *de novo* designed proteins and they have been a main driver for knowledge in protein design studies. In a review article by Korendovych and DeGrado this progress in *de novo* design is divided into three waves which were each made possible by technological leaps at their time: First, manual protein design using physical models based on improvements in peptide synthesis starting in the late 1970s. Initiated by groundbreaking advances in structure investigation with crystallographic and NMR techniques,

computer graphics and processing as well as gene editing, the phase of *computational design guided by fundamental physicochemical principles* dominated from the mid-1980s to the early 2000s. Within this era, computational modelling got more and more prominently used. The third phase of *fragment-based and bioinformatically informed computational protein design* combined earlier strategies with sequence and structure information from databases like the Protein Data Bank (PDB) (Berman et al., 2000; Korendovych & DeGrado, 2020).

As described by Korendovych and DeGrado, the design of simple architectures based on β-sheets (Dou et al., 2018; Lim et al., 1998; Quinn et al., 1994; Richardson & Richardson, 1989) and α-helices (Beesley & Woolfson, 2019; DeGrado & Lear, 1985) got more advanced by providing internal parameterization of the structural architecture (Betz & DeGrado, 1996; Emberly et al., 2004; Grigoryan & DeGrado, 2011; Korendovych & DeGrado, 2020; Lasters et al., 1988; Lombardi et al., 2000; North et al., 2001; Offer et al., 2002; Salemme, 1983). Repetitive modules that rely on an internal symmetry reduced the sequence search space and allowed more complex structures (Brunette et al., 2015; Harbury et al., 1998; Huang et al., 2014; Huang, Feldmeier, et al., 2016; Nanda et al., 2005; Thomson et al., 2014; Voet et al., 2014). These and many other studies clearly outlined the capability to design protein structures from physical principles. As a new contender in the field, machine learning-based applications arrived just recently to incorporate the accumulated knowledge for training new design algorithms in a promising manner (Anishchenko et al., 2021; Dauparas et al., 2022; Ferruz et al., 2022; Watson et al., 2023). Time will tell how far these methods can take us, but as for all machine learning applications the limit is the availability and quality of training data.

Despite substantial success with rational design approaches in the last decades and the rise of new machine-learning based software to automate such applications, those approaches remain not trivial and exhibit a low hit rate on narrowly defined or uncommon design tasks (Höcker et al., 2023).

## Protein Engineering

In contrast to the *de novo* design of a structure with desired properties, existing proteins can also be reused and adapted for a new goal. This strategy allows to attempt to delineate the influences responsible for the pure target functionality (Regan, 1993). In such an engineering approach, the design workflow is divided into finding suitable scaffold proteins for the desired application and redesigning these scaffolds to exhibit the feature of interest. Depending on the desired functionality, there are tools to select suitable scaffolds, like the software

ScaffoldSelection (Stiel et al., 2014). Redesigning the chosen scaffolds can be approached with either randomly or systematically applying mutations to find those with beneficial effects towards the design goal. Both results in new variants that need to be evaluated against the wild type. This can be relatively straightforward for a small number of variants and a visual read-out of for example a fluorescent protein. However, since the number of variants that need to be tested often exceeds expression and purification capabilities, an alternative approach is necessary. In this case, directed evolution or library-based techniques can be applied to test a large number of variants with a connected readout like fluorescence-based signal changes on ligand binding. Another way to approach this is to use computational predictions of certain features like thermodynamic stability or binding capability.

## Computational Guidance in Protein Design

As outlined above, protein design has immensely profited from the vast development in technologies for computational support. Besides the obvious improvement in raw computing power and hardware acceleration, the implementation of advanced search algorithms and efficient modelling of protein systems helped to lift computational protein design to the fundamental unit of modern protein design. In this section, I will try to justify this argument on the example of two software suites for protein design, namely Rosetta and OSPREY. Both programs strive to accomplish the same goal of structural modelling and design but with two fundamentally different approaches.

## Rosetta

The heuristics-based Monte Carlo algorithms within the Rosetta software suite have gained extreme popularity and made Rosetta one of the most prominent and most used software for macromolecular modelling and design (Leaver-Fay, Tyka, et al., 2011). One core part of Rosetta are the custom energy functions which have improved over the years. Since macromolecular systems like protein structures or complexes consist of numerous atoms, covalent bonds and non-covalent interactions, the fast and meanwhile accurate description of all energetical effects is challenging. These energy functions consist of weighted energy terms that have a physical or statistical origin. Physical phenomena like electrostatic interactions are implemented as simplified versions of the Coulomb potential. For example, by limiting long range-effects, the cubic scaling effect of three dimensions is pruned and the calculation is sped up drastically. To make up for the lost accuracy of the simplified physical terms, statistical terms

based on empiric observations of known protein structures are included. These include the prevalence of rotational angles in preferred positions like the backbone torsion angles of the protein backbone or preferred conformations of the side chains. While modern energy scoring functions like ref2015 include many energy terms with full-atom resolution, coarse-grained functions for broader energy search focus on effects of the whole structure like secondary structure content (Park et al., 2016).

To improve the energy of a protein structure Rosetta incorporates libraries of low-energy conformations of the side chains that have been shown to dominate the protein cores for tight packing – also referred to as rotamers (Dunbrack & Cohen, 1997; Dunbrack & Karplus, 1993; Janin et al., 1978; McGregor et al., 1987). By exchanging the present rotamers with new ones, the energy difference of both structures can be compared, and a minimum energy structure can be obtained. This is relatively simple for two protein conformations, but the exponential scaling of conformations makes it hard – even for smaller proteins. In fact, even five available rotamers at each position create more than $9 \cdot 10^{11}$ combinations for a protein with 20 amino acids – and at this length, it might not even pass the stage of being a peptide. Rosetta facilitates a heuristic algorithm – namely the Monte Carlo algorithm proposed in the 1950s – to find a balance between computational cost and accuracy (Metropolis et al., 1953). Random exchanges of rotamers are evaluated energetically, and if the total energy decreases, the steps are accepted. In the case of increasing total energy, a random factor is introduced and compared to the energy difference to decide if the step is reverted or not (Leaver-Fay et al., 2005; Leaver-Fay, Jacak, et al., 2011; Leaver-Fay, Tyka, et al., 2011). Due to this heuristic factor, Rosetta has the chance of escaping local energy wells to sample broader regions of the folding energy landscape. Thus, this packing method allows efficient rotamer assignment of protein structures with a fixed backbone conformation (Kuhlman & Baker, 2000). To also sample and evaluate different backbone conformations, Rosetta offers several minimization algorithms that define a vector as the direction to lower the overall energy. While iteratively repeating the definition and application of this vector on the overall structure the total energy decreases and we end up in a local energy minimum deterministically. Both algorithms – heuristic repacking and deterministic minimization – are connected in alternating cycles to compose the default algorithm for energetic preparation of protein structures in Rosetta called relaxation. The iterative alternation allows for finding local energy minima while retaining the possibility to escape higher energy wells in favour of finding the global energy minimum or the native state of the protein (Leaver-Fay, Tyka, et al., 2011).

While there are dozens of protein design protocols in Rosetta, relaxation is also capable of finding lower energy sequences by applying mutations. The most basic design protocol – the

FastDesign protocol – is based on this principle. In fact, the basic algorithm remains similar, but instead of only allowing rotamers of the native amino acid type to be picked by the packing algorithm, non-native amino acid types are randomly introduced. As a result, the decline in total energy is not only due to a more favourable packing of amino acid side chains but also due to their identity and their interactions with their environment (Maguire et al., 2021).

By applying heuristics and using a highly engineered energy function, the Rosetta suite is capable of modelling even very complex macromolecular systems. It is potentially capable of finding global energy minima while utilizing only a relatively small amount of computing power. However, the stochastic approach in many Rosetta algorithms creates the need for a high number of repetitions for complex systems to sample enough conformational space. In fact, Rosetta is not guaranteed to find the global minimum at all. Furthermore, scoring of individual conformations is limited by Rosetta energy functions which are focusing on fast calculations. For example, these scoring functions also rely on empirical data for weighting energy terms or the incorporation of statistically defined terms (Leman et al., 2020).

## OSPREY

In contrast to heuristically sampling the conformational space, the search for a global energy minimum or protein design has been extensively studied as an optimization problem for deterministic, provable algorithms (Allouche et al., 2014). While their end goal is the same as for Rosetta's stochastic methodology, deterministic algorithms guarantee to find an optimal solution. To not get overwhelmed by almost endless combinations, the search space is filtered to remove unfavourable combinations. The software OSPREY makes use of different deterministic algorithms to generate optimal solutions for protein design problems (Gainza et al., 2013; Hallen et al., 2018): The search for the global minimum energy conformation (GMEC) is performed by an adaptation of the A* algorithm – originally designed for pathfinding (Leach & Lemon, 1998). Instead of detecting the optimal path between two points by selecting decent travel nodes, this adaptation uses rotamers as nodes. By constructing a tree-like representation of all rotamer combinations A* guarantees to find the optimal combination within its force field. This derivative of the A* algorithm was further optimized for applications on protein structures. For example, reordering the rotamer sequence helps to decrease the relevant search space by pruning branches with failing rotamer combinations as early as possible (Roberts et al., 2015). Usually, A* is performed in combination with dead-end elimination (DEE) as a preprocessing step for the search for GMEC. DEE is an algorithm to efficiently prune those branches of the search tree that can be proven to not be part of the GMEC (Desmet et al., 1992; Gordon et al.,

2003; Lasters et al., 1995; Leach & Lemon, 1998). Traditional applications of DEE contain rigid rotamers as possible side chain conformations. OSPREY implemented several iterations of DEE to optimize the search performance and enable new functionality. While MinDEE takes into account the minimized backbone structure, iMinDEE and CATS even enable continuous rotation of side chain or backbones to overcome the gaps of rigid rotamers (Gainza et al., 2012; Georgiev et al., 2008; Hallen & Donald, 2017). Furthermore, non-pairwise decomposable energy terms can be included to account for effects like solvation or incorporate basic quantum mechanics calculations (Hallen et al., 2015). Such effects in complex protein modelling and design tasks are getting more and more accessible by OSPREY's search algorithms in combination with GPU-acceleration (Hallen et al., 2018). Protein redesign in OSPREY is approached as in Rosetta by appending other amino acid rotamers at the desired positions and, thus, expanding the search space.

In comparison to heuristic algorithms implemented in Rosetta, deterministic algorithms with provable guarantees to find the global minimum tend to be more computationally expensive. This expense, however, is justified in cases where the exact solution is desired, or repeatability of the calculations matters. The additional options introduced by recent algorithms to include continuous flexibility or even quantum mechanics-based calculations set another unique advantage for OSPREY (Hallen et al., 2018).

Overall, both, stochastic sampling and energetic abstraction of Rosetta as well as the deterministic approach of OSPREY were proven to provide powerful frameworks for modelling and design in protein-based systems. The individual advantages and disadvantages of both software suites lead to diverse applications and a large number of specialized subprotocols as a shared effort of the protein design community (Du et al., 2021; Guerin et al., 2022; Ollikainen et al., 2015; Raveh et al., 2010; Traoré et al., 2013).

# 4. Engineering Function: Ligand Binding

As outlined in the last chapter, modern protein design software is able to model and redesign proteins or protein complexes to find structural energy minima and improve the total energy of such proteins. While there has long been a desire to utilize proteins to perform a function with biomolecular techniques (Craik et al., 1985; Knowles, 1987), software has quickly become a useful tool to guide these engineering approaches. This section will summarize relevant developments and potential targets for computer-aided engineering of proteins for a target function.

Besides the catalysis of enzymatic reactions, binding other proteins or small molecule ligands is the most prominent goal in engineering protein functionality. Enzymatic activity, protein-protein interactions, and ligand binding rely on matching the protein surface with the target – be it a peptide, protein, small molecule ligand, or substrate. This was described in Hellinga's and Richards' work back in 1991 in order to announce their molecular modelling program DEZYMER (Hellinga et al., 1991). Like other early approaches to create new binding functionality, their work aimed at the introduction of metal-binding sites. This early focus on metal-binding can be explained by the diverse functions that can be performed by metals and their well-characterized geometries based on X-ray structures of natural metalloproteins (Regan, 1993). In the following years, designs for binding sites for organic small molecule ligands have been attempted, too (Allert et al., 2004; Looger et al., 2003). Even though these early designs were proven wrong (Schreier et al., 2009), the understanding about the design process got more advanced in general and successful designs were published (Tinberg et al., 2013). As a result, in combination with directed evolution and high-throughput screening, ligand binding design became a crucial tool in molecular biology. Despite numerous success stories, it is clear that engineering molecular binding is far from being a solved problem (Höcker et al., 2023). Computational guidance suffers from additional degrees of freedom by orientation, translation, and conformation of the target ligand. On top, selectivity over antagonist ligands is crucial. For sensors, for example, selectivity for the target keeps noise levels low and guarantees significant detection results.

To limit this complexity of designing a ligand-binding protein, existing features can be reused. To do so, two methods are available: One effective way to approach ligand binding is altering specificity by changing the affinity of involved ligands in an existing binder protein (Yang & Lai, 2017). There are several examples how this can be conducted rationally (Kröger, Shanmugaratnam, Ferruz, et al., 2021; Kröger, Shanmugaratnam, Scheib, et al., 2021), but also

computational tools have emerged to support this task. One such tool is the software called PocketOptimizer which predicts the most promising mutations to stabilize the interaction within a protein-small molecule complex (Malisi et al., 2012; Stiel et al., 2016). This is particularly useful for existing protein scaffolds, since known structures can serve as starting points for this design. It is important to know, however, that PocketOptimizer – like other programs – relies on the ligand being positioned in or close to the binding pocket. This creates a need for supporting software that helps to dock ligands on the protein's surface. Docking software has been developed for drug design, but also for testing binder designs like the software HADDOCK and AutoDock (Goodsell & Olson, 1990; Morris et al., 2009; van Zundert et al., 2016).

A second method for providing a starting point for design is to reuse pre-existing binding interfaces from known protein-ligand complexes in a new context. The iterative improvement of protein-ligand interaction in the course of evolution yielded copious amounts of highly optimized binding interfaces. By extracting the essential features of known binding pockets these pockets can be abstracted to a binding motif. If a suitable scaffold can be found, the motif can be introduced to a new protein. This transfer process is also referred to as grafting and was applied in tools like Optgraft for finding metal binding sites (Fazelinia et al., 2008). One successful example was extracting and planting such a binding motif on a regularly designed armadillo repeat protein (Ernst et al., 2020).

Newly designed binders can also be tested computationally. The software suites Rosetta and OSPREY, for example, include specialized protocols for redesign or pure testing of ligand binding capabilities. This is useful, especially in the case of a comparison of one target ligand and several non-desired ligands. Rosetta implements the flex ddG protocol which calculates the Rosetta energies of protein, ligand, and the protein-ligand complex to obtain a relative energy difference (Barlow et al., 2018). OSPREY approximates the partition functions of bound and unbound states with the algorithm K* as an additional layer over the A* search (Lilien et al., 2005). This algorithm was refined further, for example with a branch and bound over K* (BBK*) and with a combination of minimization-aware enumeration and recursive K* (MARK*) for ensemble-based protein binder design. While BBK* allows to efficiently remove high-energy sequences from the sequence space, MARK* is able to outperform its predecessors by setting tighter energy bounds and a different prioritization in the energy landscape (Jou et al., 2020; Ojewole et al., 2018). Thus, both software suites offer efficient protocols for testing and optimizing designed binders. Another helpful way of investigating binding is to use molecular dynamics (MD) simulations. With innovative approaches, MD simulations are getting faster or enable to estimate binding energies to get a better understanding of the protein-ligand interaction (Doerr et al., 2016; Fu et al., 2022; Jespers et al., 2021). However, it must be mentioned

that most of those techniques still require heavy resources and their ability for a generalized approach has yet to be proven (Mobley & Klimovich, 2012; Sheng et al., 2021).

## Design of a Modular Binding Reagent

Despite recent improvements in engineering ligand binding to create new binding reagents, such a process is costly, time-intensive, and is not always successful. Hence, many binding reagents are not designed individually, but antibodies are produced in an established system. Here, mice are introduced to the target ligand to produce antibodies as an immune response. The corresponding b-cells are extracted from the mice after an incubation time and fused to myeloma cells to generate a hybrid cell line called hybridoma (Bradbury & Plückthun, 2015). While this process is well established and more cost-effective than designing individual binders computationally with subsequent experiment testing, it has several downsides. First, it requires a whole immunization and extraction cycle for each target. Thus, it is time- and cost-intensive compared to purely computational work. Second, for most reagent antibodies the sequence is not determined and thus reproducibility is not given. Third, the generated antibodies often exhibit lower specificity than desired leading to wrong detection in subsequent experiments (A. C. Gray et al., 2020). On top of that, there is an urgent desire to reduce animal testing in the scientific community (Bradbury & Plückthun, 2015; A. Gray et al., 2020).

To address these issues, it would be ideal to find a non-animal derived binder system that needs minimal cost and preparation for each new target. There are attempts to use repeat protein binders which are selected by screening massive DNA libraries and produced recombinantly (Binz et al., 2004; Forrer et al., 2003; Plückthun, 2015). Even though this can limit the necessary time and cost of selection and production, the resource-intensive selection step is still required for every new target. In contrast, the Plückthun lab proposed a modular binding system based on the armadillo repeat protein scaffold which is a natural peptide binder (Gisdon et al., 2022; Parmeggiani et al., 2008). By optimizing the armadillo repeat protein scaffold to form a regular binding groove for stretched peptides, they separated each bound amino acid (see Figure 2 and Hansen et al., 2016, 2018). On this basis, the interacting residues of the armadillo repeat binder can be optimized to specifically detect one amino acid side chain at one position, leading to a strong detection of small polypeptide stretches. If a library of exchangeable modules is designed, it can be reused for each sequential stretch of unfolded protein or peptide. The arrangement of binding modules can be executed *in silico* and, thus, the resulting binder sequence only needs to be expressed in a bacterial expression system. To establish this system, the project *Predictive reagent antibody replacement technology* (Pre-ART) was formed as a shared

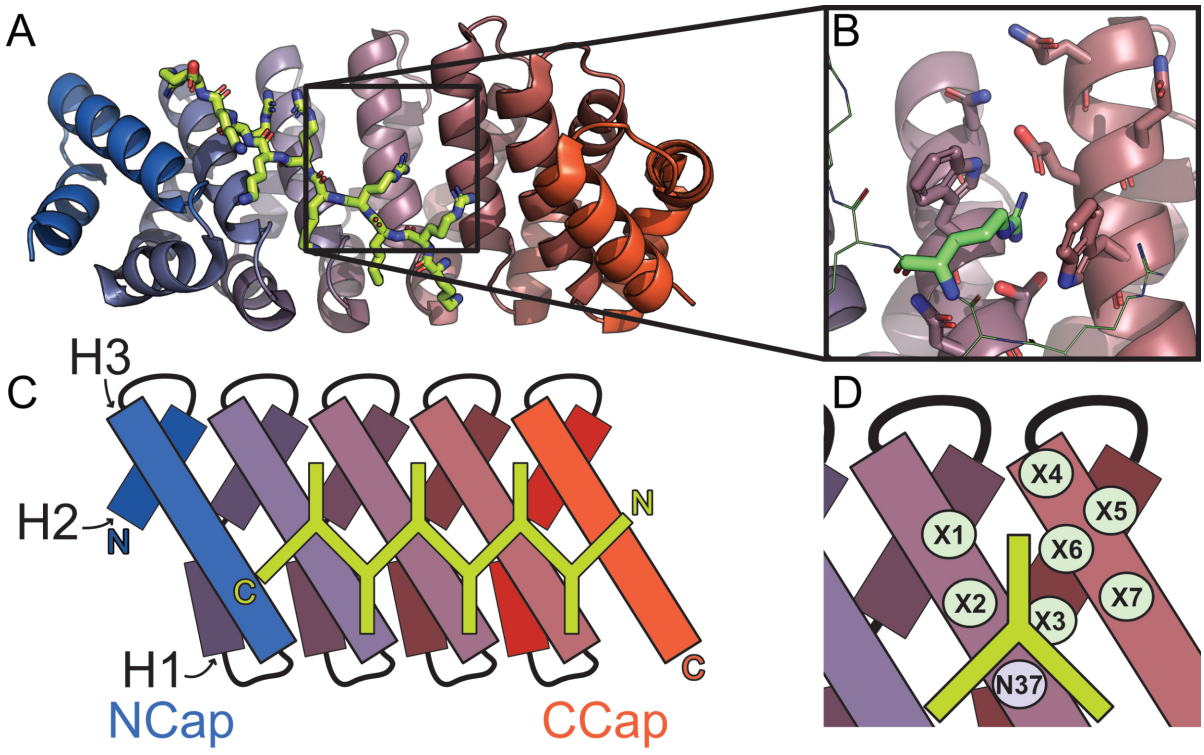



**Figure 2: Armadillo repeat protein binding a $KR_5$ peptide illustrates the regularity of its binding mode.** The designed armadillo repeat protein consisting of repeats of three α helices binds a stretched peptide with alternating arginine and lysine residues. A and C compare the protein-peptide complex in cartoon and sticks representation of the crystal structure (PDB: 5AEI) to a schematic representation. The interaction mode consists of interactions of asparagine 37 to the peptide backbone and specific interactions to the peptide side chains – as shown in detail by B and D. By focusing on one of the peptide amino acids potentially interacting binder residues can be identified in the protein. The PRe-ART project aims to design modules that can specifically target individual peptide amino acids. Those modules are combined by mutating the corresponding residues of helix three ($H_3$) of the designed armadillo repeat protein (D).

effort that combines computational and experimental methods. For a detailed perspective on this approach, I like to refer to the review article in the attached articles (Gisdon et al., 2022).

To approach this idea of a modular binding system based on designed armadillo repeat proteins a variety of binding modules need to be generated. To target every possible peptide sequence, a counterpart for all 20 of the canonical amino acids is needed. Thus, in an ideal case only 20 binding modules would be enough to build a universal tool kit. Since there is a significant difference between both sides of the stretched peptide, each side would need its own set of 20 modules. However, neighbouring binding pockets are tightly connected, and the choice of overlapping pocket residues needs to be flexible, which may compromise specificity. Therefore, alternative pockets or even double pockets are needed for certain combinations of pockets. Additionally, it is possible that suitable pockets cannot be found for the discrimination of all individual amino acid types in singularity, but in a group of two or few amino acids. Having pockets for targeting certain subsets of those groups will be crucial for targeting most peptide

sequences. To account for those factors, the identification of binding pockets is a major limiting factor for building an assorted catalogue to choose from.

As outlined in the last chapters, computational support can guide the design of new binding pockets. In combination with information from known protein structures or protein-peptide complexes and a diversity of tools for the prediction of binding specificity, the design capabilities of existing software might be improved even further.

# V. Aim

The aim of this thesis is to provide new methods for the design of peptide-binding pockets. The new techniques focus on the identification, design, and computational characterization of new binding pockets for the armadillo repeat protein scaffold. These methods are crucial for the construction of a catalogue of binding pockets that function as recognition modules. Thus, this work supports the establishment of a predictive modular binding system as an alternative for reagent antibodies.

# VI. Synopsis

List of publications in this synopsis:

Gisdon FJ*, **Kynast JP***, Ayyildiz M*, Hine AV, Plückthun A and Höcker B.

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

*Biol. Chem.* **2022**; 403(5-6): 535-543

* equal contribution

**Kynast JP**, Schwägerl F, Höcker B.

ATLIGATOR: editing protein interactions with an atlas-based approach.

*Bioinformatics* **2022** Nov 30; 38(23): 5199-5205

Noske J, **Kynast JP**, Lemm D, Schmidt S, Höcker B.

PocketOptimizer 2.0: A modular framework for computer-aided ligand-binding design.

*Protein Sci*. **2023** Jan; 32(1): e4516

**Kynast JP**, Höcker B.

Atligator Web: A Graphical User Interface for Analysis and Design of Protein–Peptide Interactions.

*BioDesign Research* **2023**; 5; 0011

The design of specific binding pockets can be approached with traditional computational redesign tools like Rosetta Design, OSPREY, or PocketOptimizer (Hallen et al., 2018; Leaver-Fay, Tyka, et al., 2011; Malisi et al., 2012). However, these programs were originally built to design or redesign single-chain protein structures and only have a limited ability to accurately predict protein-protein or protein-peptide complexes. Despite recent advances in scoring binding energy or sampling flexibility, they are limited by the exponentially growing complexity for design tasks and fast – but less accurate – energy scoring. This task is particularly intricate in systems where the difference in ligand agonist and antagonist is small. Thus, even scoring functions that perform well on big interaction surfaces tend to struggle capturing the small

details within these test cases. With an iterative algorithm like Rosetta FastDesign known biases and shortcomings will be accumulated. Moreover, it does not cover a parallel negative design route for unwanted antagonist-binding designs (Leaver-Fay, Tyka, et al., 2011).

## Learning from Nature

Modern protein redesign tools advanced significantly in recent years, but it is still challenging to create binding pockets directly considering the challenges mentioned above. To circumvent these inabilities of those tools for the creation of pockets directly, we turned around and watched out for already known binding sites. Inspired by the pioneering work of Singh and Thornton (Singh & Thornton, 1992), we searched through structures with a similar structural background to find common interactions in existing protein structures with peptide or protein ligands. With this approach, finding patterns in the interaction partners or geometry in a reoccurring manner would hint at general binding motifs. Those motifs can potentially be reused by transferring them to the armadillo repeat protein binding site. Based on this idea we created ATLIGATOR which is a software package written in the programming language Python. ATLIGATOR is short for ATlas-based LIGAnd-binding ediTOR. It can collect protein structures based on fold classifications and extract pairwise interactions from these structures. These interactions are defined by having a binder residue and a ligand residue. While the binder residue typically belongs to the polypeptide chain of the targeted fold classification, the ligand residue is coming from a bound peptide or protein. The pairwise interactions are grouped by ligand and binder residue type and aligned on the position of the ligand residue as the central point spatially. Thus, all pairwise interactions are formed from the viewpoint of the ligand and thus the spatial orientation of the binder residues can be retrieved independently of the original structure. It also contains information about their origin, like PDB code, chain, or residue identifier, which corresponds to a detailed description of the data source. This extraction procedure generates a data structure called an *atlas* where pairwise interactions are contained (Figure 3).

This collection of interaction data points now creates an intrinsic value on top of the individual data points information. Since all interaction pairs are centred at the ligand residue, they are aligned by definition and can be overlayed with each other in three-dimensional space. Thus, frequent patterns like residue type pairs in a distinct mutual orientation can be observed and described. To account for this potential knowledge gain, ATLIGATOR incorporates several ways to visualize *atlases* three-dimensionally. *Atlases* are collections of pairwise interactions of all ligand amino acid types against all binder amino acid types. However, for the design of a

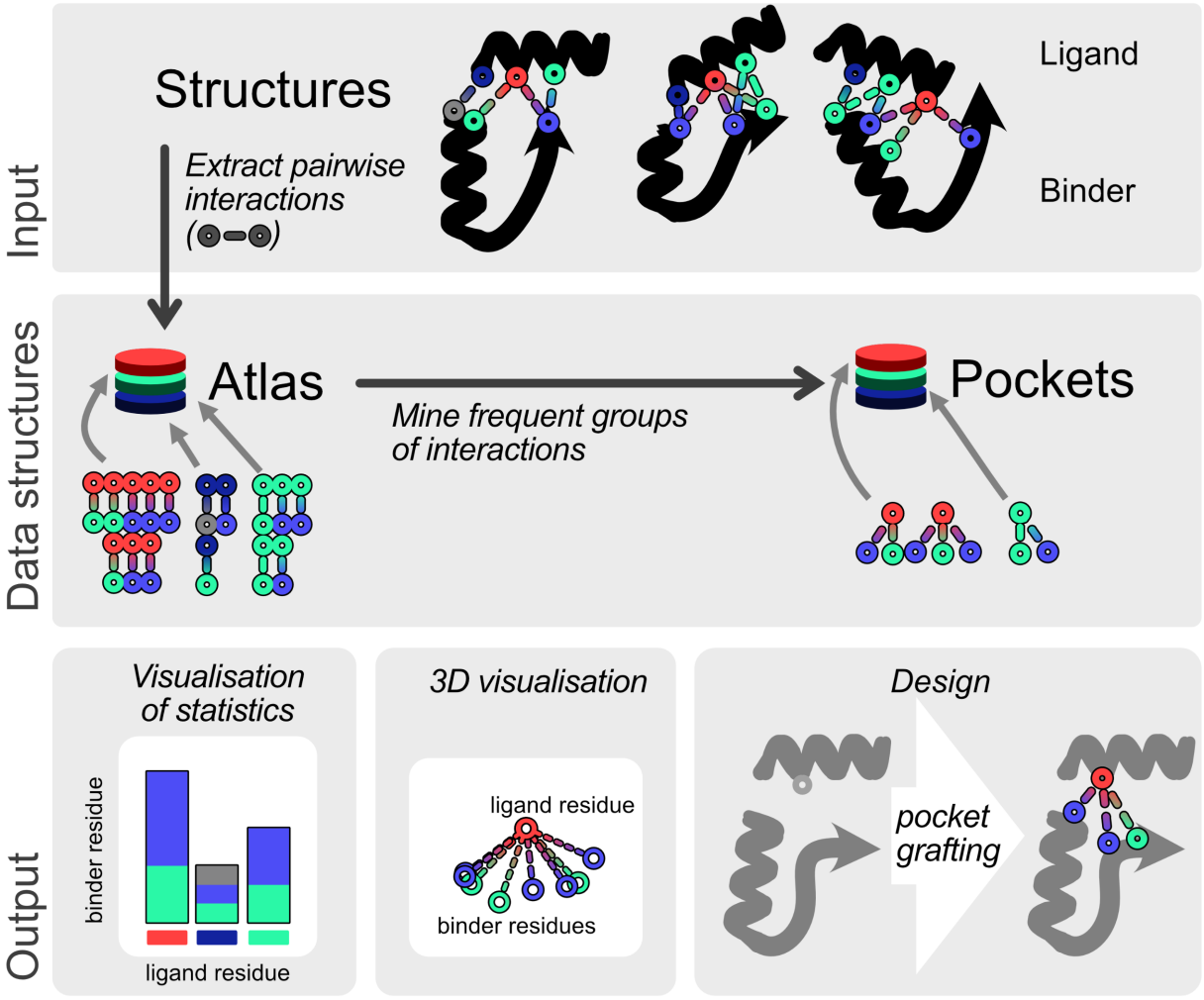



**Figure 3: Schematic overview of the ATLIGATOR components.** The ATLIGATOR python package is based on the data structures *Atlas* and *Pockets* which consist of pairwise interactions extracted from existing protein structures. While *Atlases* are structured based on the amino acid identity of both interacting residues, *Pockets* extend this structure with a one-to-many relationship. ATLIGATOR allows to visualize statistics, but also three-dimensional datapoints. *Pockets* can be grafted to own proteins by providing prepared scaffold structures and selecting the desired pockets. (Figure taken from Kynast et al., 2022)

binder of a certain amino acid type only one ligand amino acid type is important. Thus, ATLIGATOR congregates all interactions based on the amino acid type of the ligand residue. The resulting collections are called *atlas maps*, and their focus on one amino type makes them the largest naturally connected source of information for design purposes. One *atlas map* is divided in *atlas pages* which correspond to one-to-one connections between two amino acid types. The visualization of both *atlas* representations can lead to valuable insights about which interactions are found in nature and can be repurposed for own design ideas.

A specific interaction pattern which recognizes individual polypeptide residues is likely to be formed from several residues on the binder side. An *atlas* contains the information about hot spots of binder residue type positions – usually visualized as clouds of pairwise interaction

points. The combination of several of those interaction clouds could match and construct a potential binding pocket. Nevertheless, it is not guaranteed to find those combinations in known protein structures even once. In fact, two strongly preferred binder residue suggestions might be mutually exclusive in a new binder design. To compensate for this lack of combinatorial information ATLIGATOR offers a functionality to extract frequent groups of interactions. It works similar to what online sellers do when suggesting products to buy as a supplement to what customers are actually aiming for. By watching combinations of products in recent orders one can calculate the best matching additions to targeted goods. In the case of ATLIGATOR this is done with the *a priori* algorithm: Sets containing all residue types around a ligand residue are extracted and matched with all other sets (Agrawal et al., 1993). Only those subsets that are reasonably sized and occur frequently, are highlighted as plausible patterns for binding. We call those subsets *pockets* as they potentially define the crucial parts of a binding pocket.

The combination of *atlases* and *pockets* can give useful insights in which interactions are favoured in nature. These insights can be extracted and transferred to the design of new protein binders. ATLIGATOR even includes a matching algorithm to transfer interaction motifs from *pockets* directly. This combination of analysis and design make ATLIGATOR a helpful assistant for a diverse audience and requires only moderate coding experience. To share this software with the community, ATLIGATOR is available as a python package at the Python packaging index (https://pypi.org/project/atligator/) and published as free open-source software on GitHub (https://github.com/Hoecker-Lab/atligator).

## ATLIGATOR Web Server Enables Easy Usage

ATLIGATOR is a useful tool in gaining an understanding about binding motifs and how they can be used on new scaffolds. However, a programmatic interface is a significant steppingstone for users without coding experience. A graphical user interface would make up for this, and it also improves understanding by connecting visualized data. For this reason, we created ATLIGATOR web – a web server to showcase and extend ATLIGATOR functionality.

ATLIGATOR web contains the sections *Structures*, *Atlases*, *Pockets*, *Scaffolds* and *Designs* (Figure 4). *Structures* contains the input complex structures which are used in the other sections and can be inspected three-dimensionally or downloaded. *Atlases* and *Pockets* contain the data structures known from ATLIGATOR which incorporate pairwise interactions and frequent groups of interactions, respectively. The graphical interface enables linking connected data structures or browsing through different levels of hierarchy. For example, a *pocket*

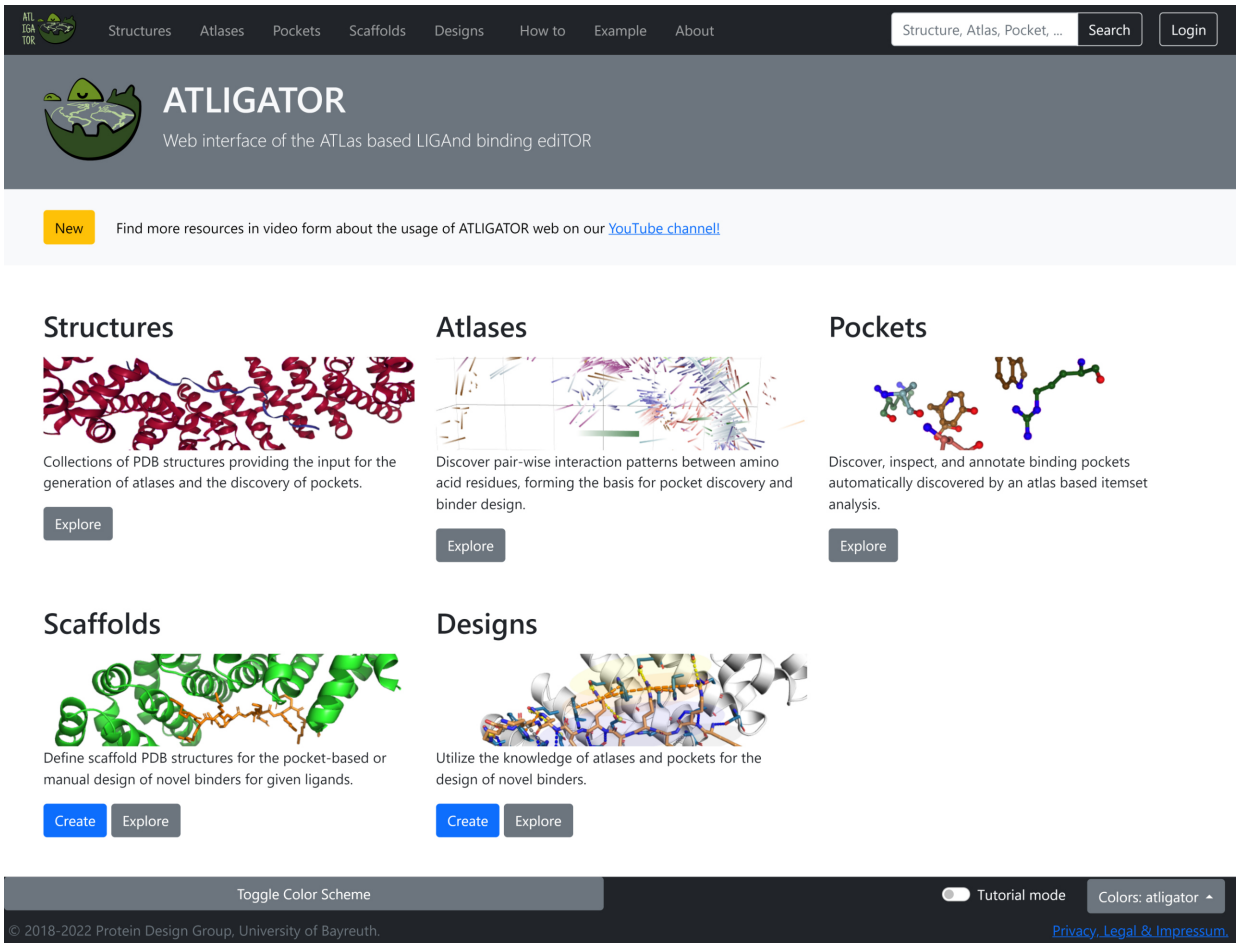



**Figure 4: Screenshot of the ATLIGATOR web landing page at https://atligator.uni-bayreuth.de/.** The sections *Structures*, *Atlases*, *Pockets*, *Scaffolds* and *Designs* offer functions to browse interactions or design new binders. A search and a login function are provided in the header. For extended guidance, a tutorial mode can be activated in the footer, besides the options to change and observe the current colour scheme for individual amino acids.

collection is linked to its underlying *atlas*, and in an *atlas* one can browse through *atlas maps* or *atlas pages* to visualize the contained data. *Atlas* and *pocket* visualization also allows to click on data points to find their origin or detailed full-atom representations. This contributes to a more seamless experience in inspecting ATLIGATOR data.

The sections *Scaffolds* and *Designs* offer functions to upload user-defined scaffolds and graft *pockets* or introduce manual mutations with a visualization of the resulting protein. For a more native representation of the introduced mutations, a repacking function is implemented based on the Rosetta *fixbb* protocol (Leaver-Fay et al., 2005). Thus, ATLIGATOR web does not only help to understand ATLIGATOR better, but also speeds up designing new binding pockets based on known interaction motifs.

Even though ATLIGATOR and ATLIGATOR web were developed with the intention to design binding pockets for the armadillo repeat protein scaffold both tools are not limited to this application. Every potential interaction between two or more amino acids can be extracted with

ATLIGATOR. Hence, every design approach that aims at protein-protein or protein-peptide interactions can benefit from this functionality.

## Prediction of Binder Specificity

While the detection and the placement of potential binding motifs on the armadillo repeat protein can be done with ATLIGATOR, we do not know if our designed binder is preferring one ligand amino acid over all others. As reviewed by Gisdon *et al.*, the heavy lifting of finding the exact binder sequence can be outsourced to directed evolution by screening a focussed DNA library (Gisdon et al., 2022). However, to match ATLIGATOR designs with a suggestion of a focussed library it is useful to test a set of designed binders computationally to finalize the content of the focussed library. For this application, perfect accuracy in the prediction of specificity is not necessary. Rather, finding a trend with several, orthogonal programs with imperfect prediction capabilities provides a reasonable guideline for library design. Protocols like Rosetta flex ddG or OSPREY's BBK* seem to be promising candidates for providing such distinctions within a timeframe of hours to days for single binders (Gisdon *et al.* unpublished - compare with chapter 4 of the introduction). In fact, our lab developed a software program called PocketOptimizer that serves a very similar purpose (Malisi et al., 2012; Stiel et al., 2016). It can find optimal solutions for mutating the binding pocket of a small molecule ligand with a linear programming solving algorithm. It features ligand and side chain flexibility of the binding pocket as well as a modular pipeline for exchanging force fields, scoring functions and more. However, PocketOptimizer depends on the software libraries tinker and BALL as outdated dependencies and does not feature a modern user interface (Hildebrandt et al., 2010; Rackers et al., 2018). This complicates the handling of PocketOptimizer and inhibits the implementation of modern force fields, scoring functions or algorithms. To address these shortcomings, we created PocketOptimizer 2.0 that offers a cleaner python user interface, even more supported modules for its architecture, such as force fields and scoring functions, a backbone-dependent rotamer library, and optimizations of the underlying algorithms. The general way how PocketOptimizer works and improvements of version 2.0 are described below.

The program PocketOptimizer contains four software components that are executed sequentially (compare with Figure 5):

First, ligand and protein structures are prepared in order to generate a correctly protonated and minimized protein-ligand complex. For this complex a binding pocket is defined to direct PocketOptimizer in generating a diverse representation of the interaction.

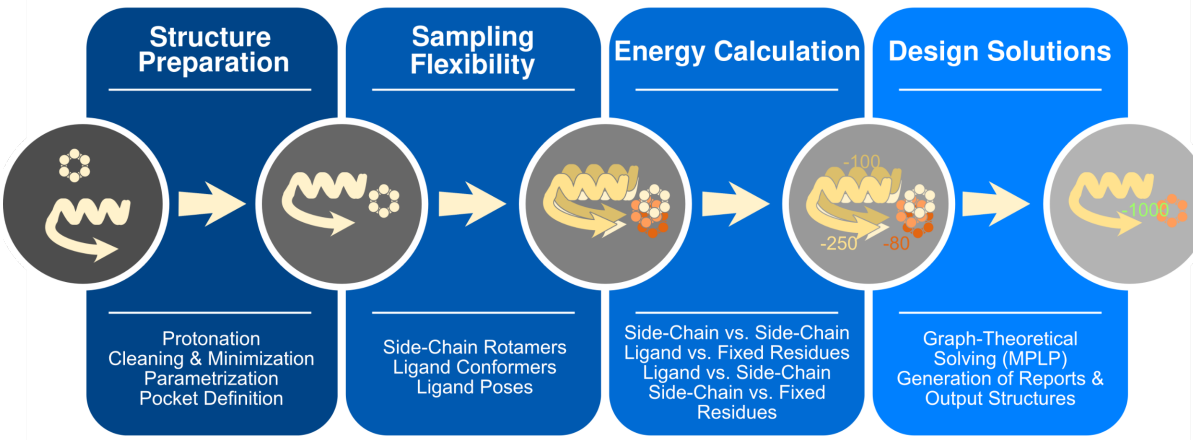



**Figure 5: Workflow of the design of a ligand-binding pocket with PocketOptimizer consisting of four steps.** After the preparation of a protein-ligand complex, the individual flexibility of the binding pocket and the ligand is sampled. Energies of non-covalent interactions between flexible and fixed parts are calculated and fed into a linear solving algorithm. This results in one or more design solutions which represent the lowest energy complex of ligand and optimized binder. This image is edited from Noske et al., 2023.

Second, flexibility of the interaction is sampled by introducing side chain conformers – also known as rotamers – for the binding pocket residues. While PocketOptimizer extracted these rotamers from a hand-crafted rotamer library, PocketOptimizer 2.0 additionally offers to use the backbone-dependent Dunbrack rotamer library (Dunbrack & Karplus, 1993; Shapovalov & Dunbrack, 2011). Version 2.0 also accelerates the rotamer sampling by replacing tinker with a faster program called FFEvaluate as part of the high-throughput molecular dynamics (HTMD) software (Doerr et al., 2016). To introduce ligand flexibility, ligand conformers are generated which are further rotated and translated in the binding pocket. PocketOptimizer 2.0 also offers a new method for the generation of ligand conformers. Conformers and positional variants are bundled up as ligand poses.

Third, the energy of the protein-ligand complex is calculated in the context of previously sampled flexibility. The total energy contains four parts where non-covalent interactions are calculated (Figure 6). To compute those parts, the side chain rotamers are matched with the fixed protein scaffold (1), other side chain rotamers (2) and the ligand poses (3). Beyond that, ligand poses are also matched with the fixed scaffold (4). Those components can be grouped into binding (3 and 4) and packing energies (1 and 2) on the basis of the ligand binding being part of it or not. If the interaction partner is the fixed scaffold, they are also defined as self-interaction energies (1 and 4). Pairwise-interaction energies are those components that include another rotamer or a ligand pose as the interaction partner (2 and 3). PocketOptimizer 2.0 offers more scoring functions for ligand binding than the original version to enable a more tailored usage for specific types of interaction.

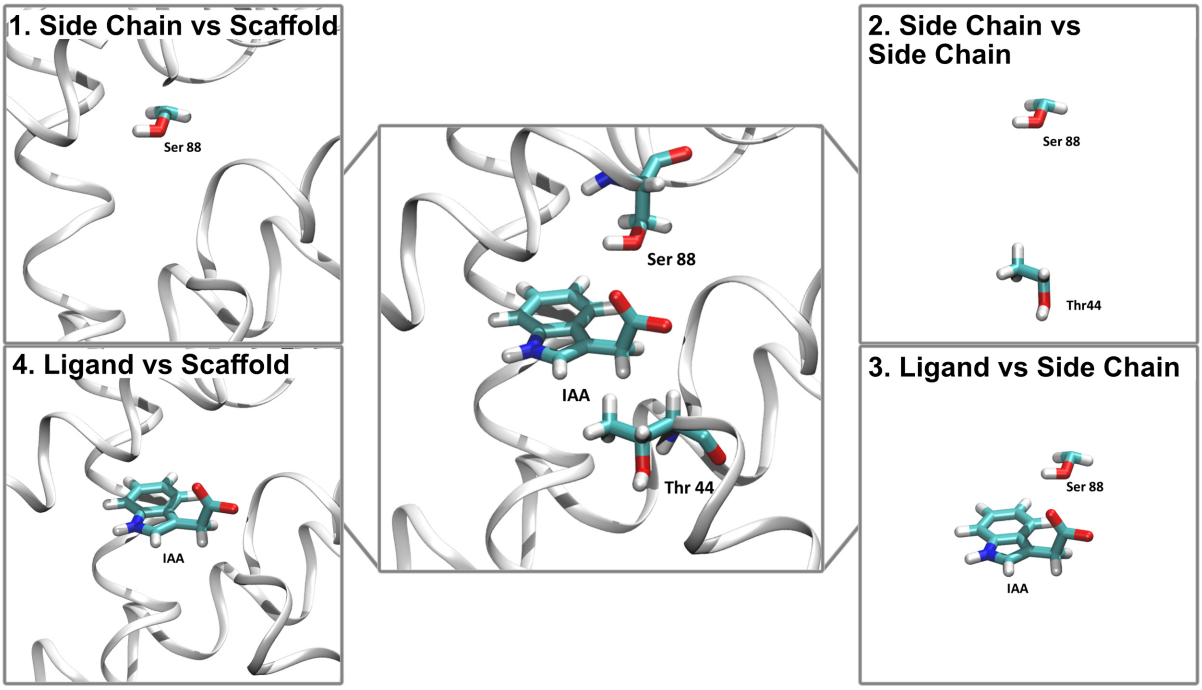



**Figure 6: Energetical components of the ligand binding pocket in PocketOptimizer.** PocketOptimizer aggregates four types of pairwise energy calculations. Flexible side chains are matched with the ligand poses, other flexible positions and the fixed scaffold, while the ligand poses are also matched with the fixed scaffold. This image is edited from Noske et al., 2023.

Fourth, all the sampled interaction energies are passed into a solving algorithm that efficiently alters flexible parts to identify the combination with the lowest total energy.

Overall, PocketOptimizer 2.0 is an improvement over PocketOptimizer in several ways. It is not only faster and easier to use and develop, but it also incorporates more force fields, scoring functions and rotamer libraries which makes it more robust. Its source code is available at https://github.com/Hoecker-Lab/pocketoptimizer.

## Pipeline for Design of Binding Modules

With the development of ATLIGATOR and ATLIGATOR web, the design of initial binding pockets for modular binders can be done more efficiently. Even though these designs might not be highly specific as they are, computational redesign methods like Rosetta flex ddG or OSPREY BBK* can be used to improve specificity later. The development of PocketOptimizer 2.0 also makes it more attractive for this task. In combination with experimental testing this setup seems promising for challenges like creating new binder modules for the modular system proposed with PRe-ART.

In the attached review article, we discussed the approach of creating such a modular binding reagent based on the armadillo repeat protein scaffold. We came up with a strategy to identify, test and implement new binder modules: Based on such computational predictions DNA library compositions have been proposed which are generated non-degenerately with the MAX-strategies (Chembath et al., 2022) by our collaboration partners in the PRe-ART project. These libraries can be screened for promising binder candidates that are characterized and complement the set of known binding modules.

# VII. Author Contributions

Gisdon FJ*, **Kynast JP***, Ayyildiz M*, Hine AV, Plückthun A and Höcker B.

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

*Biol. Chem.* **2022**; 403(5-6): 535-543

* equal contribution

For this review I participated in the concept design with F.J.G., M.A. and B.H.. I wrote parts of the manuscript, especially the section about complementarity of computational and experimental work. F.J.G. and M.A. wrote the initial draft other parts of the manuscript. All authors contributed to the discussion about the work as well as editing and completion of the manuscript.

**Kynast JP**, Schwägerl F, Höcker B.

ATLIGATOR: editing protein interactions with an atlas-based approach.

*Bioinformatics* **2022** Nov 30; 38(23): 5199-5205

In this work F.S, B.H. and I worked on the concept. F.S. and I developed and implemented the methodology. I published the software package. I did the analysis and wrote the initial draft of the manuscript. B.H. edited the manuscript and provided financial support and supervision.

Noske J, **Kynast JP**, Lemm D, Schmidt S, Höcker B.

PocketOptimizer 2.0: A modular framework for computer-aided ligand-binding design.

*Protein Sci*. **2023** Jan; 32(1): e4516

In this work J.N., myself, D.L., and S.S. implemented the methodology and did the validation. J.N. did the major work on implementation and validation. J.N., S.S., and B.H. did the formal analysis. J.N. and B.H. did the writing. All authors contributed in review and editing of the manuscript. B.H. provided the concept, financial support, and supervision.

**Kynast JP**, Höcker B.

Atligator Web: A Graphical User Interface for Analysis and Design of Protein–Peptide Interactions.

*BioDesign Research* **2023**; 5; 0011

In this work B.H. and I did the conceptualization. I designed and implemented the methodology and the web server. I did the analysis wrote the initial draft of the manuscript. B.H. contributed in review and editing of the manuscript, provided administrative and financial support, as well as supervision.

# VIII. Bibliography

Agrawal, R., Imieliński, T., & Swami, A. (1993). Mining Association Rules between Sets of Items in Large Databases. *SIGMOD Rec.*, *22*(2), 207–216. https://doi.org/10.1145/170036.170072

Akhmanova, A., & Steinmetz, M. O. (2008). Tracking the ends: a dynamic protein network controls the fate of microtubule tips. *Nature Reviews Molecular Cell Biology*, *9*(4), 309–322. https://doi.org/10.1038/nrm2369

Allert, M., Rizk, S. S., Looger, L. L., & Hellinga, H. W. (2004). Computational design of receptors for an organophosphate surrogate of the nerve agent soman. *Proceedings of the National Academy of Sciences*, *101*(21), 7907–7912. https://doi.org/10.1073/pnas.0401309101

Allouche, D., André, I., Barbe, S., Davies, J., de Givry, S., Katsirelos, G., O'Sullivan, B., Prestwich, S., Schiex, T., & Traoré, S. (2014). Computational protein design as an optimization problem. *Artificial Intelligence*, *212*, 59–79. https://doi.org/10.1016/j.artint.2014.03.005

Anfinsen, C. B. (1973). Principles that Govern the Folding of Protein Chains. *Science*, *181*(4096), 223–230. https://doi.org/10.1126/science.181.4096.223

Anishchenko, I., Pellock, S. J., Chidyausiku, T. M., Ramelot, T. A., Ovchinnikov, S., Hao, J., Bafna, K., Norn, C., Kang, A., Bera, A. K., DiMaio, F., Carter, L., Chow, C. M., Montelione, G. T., & Baker, D. (2021). De novo protein design by deep network hallucination. *Nature*, *600*(7889), 547–552. https://doi.org/10.1038/s41586-021-04184-w

Argos, P., & Palau, J. (1982). Amino acid distribution in protein secondary structures. *International Journal of Peptide and Protein Research*, *19*(4), 380–393. https://doi.org/10.1111/j.1399-3011.1982.tb02619.x

Arunachalam, J., & Gautham, N. (2008). Hydrophobic clusters in protein structures. *Proteins: Structure, Function and Genetics*, *71*(4), 2012–2025. https://doi.org/10.1002/prot.21881

Attaix, D., Combaret, L., Pouch, M.-N., & Taillandier, D. (2001). Regulation of proteolysis. *Current Opinion in Clinical Nutrition and Metabolic Care*, *4*(1), 45–49. https://doi.org/10.1097/00075197-200101000-00009

Aurora, R., Srinivasan, R., & Rose, G. D. (1994). Rules for α-Helix Termination by Glycine. *Science*, *264*(5162), 1126–1130. https://doi.org/10.1126/science.8178170

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. *The Journal of Physical Chemistry B*, *122*(21), 5389–5399. https://doi.org/10.1021/acs.jpcb.7b11367

Beesley, J. L., & Woolfson, D. N. (2019). The de novo design of α-helical peptides for supramolecular self-assembly. *Current Opinion in Biotechnology*, *58*, 175–182. https://doi.org/10.1016/j.copbio.2019.03.017

Berman, H. M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T. N., Weissig, H., Shindyalov, I. N., & Bourne, P. E. (2000). The Protein Data Bank. In *Nucleic Acids Research* (Vol. 28, Issue 1). http://www.rcsb.org/pdb/status.html

Betz, S. F., & DeGrado, W. F. (1996). Controlling Topology and Native-like Behavior of de Novo-Designed Peptides: Design and Characterization of Antiparallel Four-Stranded Coiled Coils. *Biochemistry*, *35*(21), 6955–6962. https://doi.org/10.1021/bi9600095a

Binz, H. K., Amstutz, P., Kohl, A., Stumpp, M. T., Briand, C., Forrer, P., Grütter, M. G., & 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

Bordin, N., Sillitoe, I., Lees, J. G., & Orengo, C. (2021). Tracing Evolution Through Protein Structures: Nature Captured in a Few Thousand Folds. *Frontiers in Molecular Biosciences*, *8*. https://doi.org/10.3389/fmolb.2021.668184

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

Brunette, T., Parmeggiani, F., Huang, P.-S., Bhabha, G., Ekiert, D. C., Tsutakawa, S. E., Hura, G. L., Tainer, J. A., & Baker, D. (2015). Exploring the repeat protein universe through computational protein design. *Nature*, *528*(7583), 580–584. https://doi.org/10.1038/nature16162

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* (pp. 19–41). https://doi.org/10.1007/978-1-0716-2152-3_3

Cordes, M. H. J., Burton, R. E., Walsh, N. P., McKnight, C. J., & Sauer, R. T. (2000). An evolutionary bridge to a new protein fold. *Nature Structural Biology*, *7*(12), 1129–1132. https://doi.org/10.1038/81985

Craik, C. S., Largman, C., Fletcher, T., Roczniak, S., Barr, P. J., Fletterick, R., & Rutter, W. J. (1985). Redesigning Trypsin: Alteration of Substrate Specificity. *Science*, *228*(4697), 291–297. https://doi.org/10.1126/science.3838593

Dauparas, J., Anishchenko, I., Bennett, N., Bai, H., Ragotte, R. J., Milles, L. F., Wicky, B. I. M., Courbet, A., de Haas, R. J., Bethel, N., Leung, P. J. Y., Huddy, T. F., Pellock, S., Tischer, D., Chan, F., Koepnick, B., Nguyen, H., Kang, A., Sankaran, B., … Baker, D. (2022). Robust deep learning–based protein sequence design using ProteinMPNN. *Science*, *378*(6615), 49–56. https://doi.org/10.1126/science.add2187

David, A., Islam, S., Tankhilevich, E., & Sternberg, M. J. E. (2022). The AlphaFold Database of Protein Structures: A Biologist's Guide. In *Journal of Molecular Biology* (Vol. 434, Issue 2). Academic Press. https://doi.org/10.1016/j.jmb.2021.167336

DeGrado, W. F., & Lear, J. D. (1985). Induction of peptide conformation at apolar water interfaces. 1. A study with model peptides of defined hydrophobic periodicity. *Journal of the American Chemical Society*, *107*(25), 7684–7689. https://doi.org/10.1021/ja00311a076

Desmet, J., Maeyer, M. De, Hazes, B., & Lasters, I. (1992). The dead-end elimination theorem and its use in protein side-chain positioning. *Nature*, *356*(6369), 539–542. https://doi.org/10.1038/356539a0

Doerr, S., Harvey, M. J., Noé, F., & De Fabritiis, G. (2016). HTMD: High-Throughput Molecular Dynamics for Molecular Discovery. *Journal of Chemical Theory and Computation*, *12*(4), 1845–1852. https://doi.org/10.1021/acs.jctc.6b00049

Dou, J., Vorobieva, A. A., Sheffler, W., Doyle, L. A., Park, H., Bick, M. J., Mao, B., Foight, G. W., Lee, M. Y., Gagnon, L. A., Carter, L., Sankaran, B., Ovchinnikov, S., Marcos, E., Huang, P.-S., Vaughan, J. C., Stoddard, B. L., & Baker, D. (2018). De novo design of a fluorescence-activating β-barrel. *Nature*, *561*(7724), 485–491. https://doi.org/10.1038/s41586-018-0509-0

Du, Z., Su, H., Wang, W., Ye, L., Wei, H., Peng, Z., Anishchenko, I., Baker, D., & Yang, J. (2021). The trRosetta server for fast and accurate protein structure prediction. *Nature Protocols*, *16*(12), 5634–5651. https://doi.org/10.1038/s41596-021-00628-9

Dunbrack, R. L., & Cohen, F. E. (1997). Bayesian statistical analysis of protein side-chain rotamer preferences. *Protein Science*, *6*(8), 1661–1681. https://doi.org/10.1002/pro.5560060807

Dunbrack, R. L., & Karplus, M. (1993). Backbone-dependent Rotamer Library for Proteins Application to Side-chain Prediction. *Journal of Molecular Biology*, *230*(2), 543–574. https://doi.org/10.1006/jmbi.1993.1170

Eisenberg, D. (2003). The discovery of the α-helix and β-sheet, the principal structural features of proteins. *Proceedings of the National Academy of Sciences*, *100*(20), 11207–11210. https://doi.org/10.1073/pnas.2034522100

Emberly, E. G., Mukhopadhyay, R., Tang, C., & Wingreen, N. S. (2004). Flexibility of β-sheets: Principal component analysis of database protein structures. *Proteins: Structure, Function, and Bioinformatics*, *55*(1), 91–98. https://doi.org/10.1002/prot.10618

Ernst, P., Zosel, F., Reichen, C., Nettels, D., Schuler, B., & Plückthun, A. (2020). Structure-Guided Design of a Peptide Lock for Modular Peptide Binders. *ACS Chemical Biology*, *15*(2), 457–468. https://doi.org/10.1021/acschembio.9b00928

Fazelinia, H., Cirino, P. C., & Maranas, C. D. (2008). OptGraft: A computational procedure for transferring a binding site onto an existing protein scaffold. *Protein Science*, NA-NA. https://doi.org/10.1002/pro.2

Ferruz, N., Schmidt, S., & Höcker, B. (2022). ProtGPT2 is a deep unsupervised language model for protein design. *Nature Communications*, *13*(1), 4348. https://doi.org/10.1038/s41467-022-32007-7

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

Fox, N. K., Brenner, S. E., & Chandonia, J.-M. (2014). SCOPe: Structural Classification of Proteins—extended, integrating SCOP and ASTRAL data and classification of new structures. *Nucleic Acids Research*, *42*(D1), D304–D309. https://doi.org/10.1093/nar/gkt1240

Fu, H., Chen, H., Blazhynska, M., Goulard Coderc de Lacam, E., Szczepaniak, F., Pavlova, A., Shao, X., Gumbart, J. C., Dehez, F., Roux, B., Cai, W., & Chipot, C. (2022). Accurate determination of protein:ligand standard binding free energies from molecular dynamics simulations. *Nature Protocols*, *17*(4), 1114–1141. https://doi.org/10.1038/s41596-021-00676-1

Gainza, P., Nisonoff, H. M., & Donald, B. R. (2016). Algorithms for protein design. *Current Opinion in Structural Biology*, *39*, 16–26. https://doi.org/10.1016/j.sbi.2016.03.006

Gainza, P., Roberts, K. E., & Donald, B. R. (2012). Protein Design Using Continuous Rotamers. *PLoS Computational Biology*, *8*(1), e1002335. https://doi.org/10.1371/journal.pcbi.1002335

Gainza, P., Roberts, K. E., Georgiev, I., Lilien, R. H., Keedy, D. A., Chen, C.-Y., Reza, F., Anderson, A. C., Richardson, D. C., Richardson, J. S., & Donald, B. R. (2013). OSPREY: protein design with ensembles, flexibility, and provable algorithms. *Methods in Enzymology*, *523*, 87–107. https://doi.org/10.1016/B978-0-12-394292-0.00005-9

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/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. In *Biological Chemistry* (Vol. 403, Issues 5–6, pp. 535–543). De Gruyter Open Ltd. 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 Genetics*, *8*(3), 195–202. https://doi.org/10.1002/prot.340080302

Gordon, D. B., Hom, G. K., Mayo, S. L., & Pierce, N. A. (2003). Exact rotamer optimization for protein design. *Journal of Computational Chemistry*, *24*(2), 232–243. https://doi.org/10.1002/jcc.10121

Gray, A., Bradbury, A. R. M., Knappik, A., Plückthun, A., Borrebaeck, C. A. K., & Dübel, S. (2020). Animal-free alternatives and the antibody iceberg. *Nature Biotechnology*, *38*(11), 1234–1239. https://doi.org/10.1038/s41587-020-0687-9

Gray, A. C., Bradbury, A., Dübel, S., Knappik, A., Plückthun, A., & Borrebaeck, C. A. K. (2020). Reproducibility: bypass animals for antibody production. *Nature*, *581*(7808), 262–262. https://doi.org/10.1038/d41586-020-01474-7

Grigoryan, G., & DeGrado, W. F. (2011). Probing Designability via a Generalized Model of Helical Bundle Geometry. *Journal of Molecular Biology*, *405*(4), 1079–1100. https://doi.org/10.1016/j.jmb.2010.08.058

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

Hallen, M. A., & Donald, B. R. (2017). CATS (Coordinates of Atoms by Taylor Series): protein design with backbone flexibility in all locally feasible directions. *Bioinformatics*, *33*(14), i5–i12. https://doi.org/10.1093/bioinformatics/btx277

Hallen, M. A., Gainza, P., & Donald, B. R. (2015). Compact Representation of Continuous Energy Surfaces for More Efficient Protein Design. *Journal of Chemical Theory and Computation*, *11*(5), 2292–2306. https://doi.org/10.1021/ct501031m

Hallen, M. A., Martin, J. W., Ojewole, A., Jou, J. D., Lowegard, A. U., Frenkel, M. S., Gainza, P., Nisonoff, H. M., Mukund, A., Wang, S., Holt, G. T., Zhou, D., Dowd, E., & 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

Hansen, S., Ernst, P., König, S. L. B., Reichen, C., Ewald, C., Nettels, D., Mittl, P. R. E., Schuler, B., & Plückthun, A. (2018). Curvature of designed armadillo repeat proteins allows modular peptide binding. *Journal of Structural Biology*, *201*(2), 108–117. https://doi.org/10.1016/j.jsb.2017.08.009

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

Harbury, P. B., Plecs, J. J., Tidor, B., Alber, T., & Kim, P. S. (1998). High-Resolution Protein Design with Backbone Freedom. *Science*, *282*(5393), 1462–1467. https://doi.org/10.1126/science.282.5393.1462

Hellinga, H. W., Caradonna, J. P., & Richards, F. M. (1991). Construction of new ligand binding sites in proteins of known structure. *Journal of Molecular Biology*, *222*(3), 787–803. https://doi.org/10.1016/0022-2836(91)90511-4

Hildebrandt, A., Dehof, A. K., Rurainski, A., Bertsch, A., Schumann, M., Toussaint, N. C., Moll, A., Stöckel, D., Nickels, S., Mueller, S. C., Lenhof, H.-P., & Kohlbacher, O. (2010). BALL - biochemical algorithms library 1.3. *BMC Bioinformatics*, *11*(1), 531. https://doi.org/10.1186/1471-2105-11-531

Höcker, B., Lu, P., Glasgow, A., Marks, D. S., Chatterjee, P., Slusky, J. S. G., Schueler-Furman, O., & Huang, P. (2023). How can the protein design community best support biologists who want to harness AI tools for protein structure prediction and design? *Cell Systems*, *14*(8), 629–632. https://doi.org/10.1016/j.cels.2023.07.005

Huang, P.-S., Boyken, S. E., & Baker, D. (2016). The coming of age of de novo protein design. *Nature*, *537*(7620), 320–327. https://doi.org/10.1038/nature19946

Huang, P.-S., Feldmeier, K., Parmeggiani, F., Fernandez Velasco, D. A., Höcker, B., & Baker, D. (2016). De novo design of a four-fold symmetric TIM-barrel protein with atomic-level accuracy. *Nature Chemical Biology*, *12*(1), 29–34. https://doi.org/10.1038/nchembio.1966

Huang, P.-S., Oberdorfer, G., Xu, C., Pei, X. Y., Nannenga, B. L., Rogers, J. M., DiMaio, F., Gonen, T., Luisi, B., & Baker, D. (2014). High thermodynamic stability of parametrically designed helical bundles. *Science*, *346*(6208), 481–485. https://doi.org/10.1126/science.1257481

Janin, J., Wodak, S., Levitt, M., & Maigret, B. (1978). Conformation of amino acid side-chains in proteins. *Journal of Molecular Biology*, *125*(3), 357–386. https://doi.org/10.1016/0022-2836(78)90408-4

Jespers, W., Åqvist, J., & Gutiérrez-de-Terán, H. (2021). *Free Energy Calculations for Protein–Ligand Binding Prediction* (pp. 203–226). https://doi.org/10.1007/978-1-0716-1209-5_12

Jou, J. D., Holt, G. T., Lowegard, A. U., & Donald, B. R. (2020). Minimization-Aware Recursive *K*:* A Novel, Provable Algorithm that Accelerates Ensemble-Based Protein Design and Provably Approximates the Energy Landscape. *Journal of Computational Biology*, *27*(4), 550–564. https://doi.org/10.1089/cmb.2019.0315

Kathuria, S. V., Chan, Y. H., Nobrega, R. P., Özen, A., & Matthews, C. R. (2016). Clusters of isoleucine, leucine, and valine side chains define cores of stability in high-energy states of globular proteins: Sequence determinants of structure and stability. *Protein Science*, *25*(3), 662–675. https://doi.org/10.1002/pro.2860

Kim, M. K., & Kang, Y. K. (1999). Positional preference of proline in alpha-helices. *Protein Science : A Publication of the Protein Society*, *8*(7), 1492–1499. https://doi.org/10.1110/ps.8.7.1492

Knowles, J. R. (1987). Tinkering with Enzymes: What Are We Learning? *Science*, *236*(4806), 1252–1258. https://doi.org/10.1126/science.3296192

Korendovych, I. V., & DeGrado, W. F. (2020). De novo protein design, a retrospective. *Quarterly Reviews of Biophysics*, *53*, e3. https://doi.org/10.1017/S0033583519000131

Kröger, P., Shanmugaratnam, S., Ferruz, N., Schweimer, K., & Höcker, B. (2021). A comprehensive binding study illustrates ligand recognition in the periplasmic binding protein PotF. *Structure*, *29*(5), 433-443.e4. https://doi.org/10.1016/j.str.2020.12.005

Kröger, P., Shanmugaratnam, S., Scheib, U., & Höcker, B. (2021). Fine-tuning spermidine binding modes in the putrescine binding protein PotF. *Journal of Biological Chemistry*, *297*(6), 101419. https://doi.org/10.1016/j.jbc.2021.101419

Kuhlman, B., & Baker, D. (2000). Native protein sequences are close to optimal for their structures. *Proceedings of the National Academy of Sciences*, *97*(19), 10383–10388. https://doi.org/10.1073/pnas.97.19.10383

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

Lasters, I., Maeyer, M. De, & Desmet, J. (1995). Enhanced dead-end elimination in the search for the global minimum energy conformation of a collection of protein side chains. *'Protein Engineering, Design and Selection'*, *8*(8), 815–822. https://doi.org/10.1093/protein/8.8.815

Lasters, I., Wodak, S. J., Alard, P., & van Cutsem, E. (1988). Structural principles of parallel beta-barrels in proteins. *Proceedings of the National Academy of Sciences*, *85*(10), 3338–3342. https://doi.org/10.1073/pnas.85.10.3338

Leach, A. R., & Lemon, A. P. (1998). Exploring the conformational space of protein side chains using dead-end elimination and the A* algorithm. *Proteins*, *33*(2), 227–239. https://doi.org/10.1002/(sici)1097-0134(19981101)33:2<227::aid-prot7>3.0.co;2-f

Leaver-Fay, A., Jacak, R., Stranges, P. B., & Kuhlman, B. (2011). A Generic Program for Multistate Protein Design. *PLoS ONE*, *6*(7), e20937. https://doi.org/10.1371/journal.pone.0020937

Leaver-Fay, A., Kuhlman, B., & Snoeyink, J. (2005). An adaptive dynamic programming algorithm for the side chain placement problem. *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, 16–27.

Leaver-Fay, A., Tyka, M., Lewis, S. M., Lange, O. F., Thompson, J., Jacak, R., Kaufman, K. W., Renfrew, P. D., Smith, C. A., Sheffler, W., Davis, I. W., Cooper, S., Treuille, A., Mandell, D. J., Richter, F., Ban, Y.-E. A., Fleishman, S. J., Corn, J. E., Kim, D. E., … Bradley, P. (2011). Rosetta 3: An Object-Oriented Software Suite for the Simulation and Design of Macromolecules. In M. L. Johnson & L. Brand (Eds.), *Methods in Enzymology* (Vol. 487, pp. 545–574). Academic Press. https://doi.org/https://doi.org/10.1016/B978-0-12-381270-4.00019-6

Leman, J. K., Weitzner, B. D., Lewis, S. M., Adolf-Bryfogle, J., Alam, N., Alford, R. F., Aprahamian, M., Baker, D., Barlow, K. A., Barth, P., Basanta, B., Bender, B. J., Blacklock, K., Bonet, J., Boyken, S. E., Bradley, P., Bystroff, C., Conway, P., Cooper, S., … Bonneau, R. (2020). Macromolecular modeling and design in Rosetta: recent methods and frameworks. *Nature Methods*, *17*(7), 665–680. https://doi.org/10.1038/s41592-020-0848-2

Levinthal, C. (1969). How to fold graciously. *Mossbauer Spectrosc. Biol. Syst*, *67*, 22–24.

Lilien, R. H., Stevens, B. W., Anderson, A. C., & Donald, B. R. (2005). A Novel Ensemble-Based Scoring and Search Algorithm for Protein Redesign and Its Application to Modify the Substrate Specificity of the Gramicidin Synthetase A Phenylalanine Adenylation Enzyme. *Journal of Computational Biology*, *12*(6), 740–761. https://doi.org/10.1089/cmb.2005.12.740

Lim, A., Saderholm, M. J., Kroll, M., Yan, Y., Perera, L., Erickson, B. W., Makhov, A. M., & Griffith, J. D. (1998). Engineering of betabellin-15d: A 64 residue beta sheet protein that forms long narrow multimeric fibrils. *Protein Science*, *7*(7), 1545–1554. https://doi.org/10.1002/pro.5560070708

Lin, Z., Akin, H., Rao, R., Hie, B., Zhu, Z., Lu, W., Smetanin, N., Verkuil, R., Kabeli, O., Shmueli, Y., dos Santos Costa, A., Fazel-Zarandi, M., Sercu, T., Candido, S., & Rives, A. (2023). Evolutionary-scale prediction of atomic-level protein structure with a language model. In *Science* (Vol. 379). https://www.science.org

Lombardi, A., Summa, C. M., Geremia, S., Randaccio, L., Pavone, V., & DeGrado, W. F. (2000). Retrostructural analysis of metalloproteins: Application to the design of a minimal model for diiron proteins. *Proceedings of the National Academy of Sciences*, *97*(12), 6298–6305. https://doi.org/10.1073/pnas.97.12.6298

Looger, L. L., Dwyer, M. A., Smith, J. J., & Hellinga, H. W. (2003). Computational design of receptor and sensor proteins with novel functions. *Nature*, *423*(6936), 185–190. https://doi.org/10.1038/nature01556

Maguire, J. B., Haddox, H. K., Strickland, D., Halabiya, S. F., Coventry, B., Griffin, J. R., Pulavarti, S. V. S. R. K., Cummins, M., Thieker, D. F., Klavins, E., Szyperski, T., DiMaio, F., Baker, D., & 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

Malisi, C., Schumann, M., Toussaint, N. C., Kageyama, J., Kohlbacher, O., & Höcker, B. (2012). Binding Pocket Optimization by Computational Protein Design. *PLoS ONE*, *7*(12), e52505. https://doi.org/10.1371/journal.pone.0052505

McGregor, M. J., Islam, S. A., & Sternberg, M. J. E. (1987). Analysis of the relationship between side-chain conformation and secondary structure in globular proteins. *Journal of Molecular Biology*, *198*(2), 295–310. https://doi.org/10.1016/0022-2836(87)90314-7

Meier, S., Jensen, P. R., David, C. N., Chapman, J., Holstein, T. W., Grzesiek, S., & Özbek, S. (2007). Continuous Molecular Evolution of Protein-Domain Structures by Single Amino Acid Changes. *Current Biology*, *17*(2), 173–178. https://doi.org/10.1016/j.cub.2006.10.063

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

Mobley, D. L., & Klimovich, P. V. (2012). Perspective: Alchemical free energy calculations for drug discovery. *The Journal of Chemical Physics*, *137*(23). https://doi.org/10.1063/1.4769292

Morris, G. M., Huey, R., Lindstrom, W., Sanner, M. F., Belew, R. K., Goodsell, D. S., & Olson, A. J. (2009). AutoDock4 and AutoDockTools4: Automated docking with selective receptor flexibility. *Journal of Computational Chemistry*, *30*(16), 2785–2791. https://doi.org/10.1002/jcc.21256

Nanda, V., Rosenblatt, M. M., Osyczka, A., Kono, H., Getahun, Z., Dutton, P. L., Saven, J. G., & DeGrado, W. F. (2005). De Novo Design of a Redox-Active Minimal Rubredoxin Mimic. *Journal of the American Chemical Society*, *127*(16), 5804–5805. https://doi.org/10.1021/ja050553f

Nassar, R., Dignon, G. L., Razban, R. M., & Dill, K. A. (2021). The Protein Folding Problem: The Role of Theory. *Journal of Molecular Biology*, *433*(20), 167126. https://doi.org/https://doi.org/10.1016/j.jmb.2021.167126

North, B., Summa, C. M., Ghirlanda, G., & DeGrado, W. F. (2001). D(n)-symmetrical tertiary templates for the design of tubular proteins. *Journal of Molecular Biology*, *311*(5), 1081–1090. https://doi.org/10.1006/jmbi.2001.4900

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). https://doi.org/10.1002/pro.4516

Nowak, E., Miller, J. T., Bona, M. K., Studnicka, J., Szczepanowski, R. H., Jurkowski, J., Le Grice, S. F. J., & Nowotny, M. (2014). Ty3 reverse transcriptase complexed with an RNA-DNA hybrid shows structural and functional asymmetry. *Nature Structural & Molecular Biology*, *21*(4), 389–396. https://doi.org/10.1038/nsmb.2785

Offer, G., Hicks, M. R., & Woolfson, D. N. (2002). Generalized Crick Equations for Modeling Noncanonical Coiled Coils. *Journal of Structural Biology*, *137*(1–2), 41–53. https://doi.org/10.1006/jsbi.2002.4448

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

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), e1004335. https://doi.org/10.1371/journal.pcbi.1004335

Pan, X., & Kortemme, T. (2021). Recent advances in de novo protein design: Principles, methods, and applications. *Journal of Biological Chemistry*, *296*, 100558. https://doi.org/10.1016/j.jbc.2021.100558

Park, H., Bradley, P., Greisen, P., Liu, Y., Mulligan, V. K., Kim, D. E., Baker, D., & DiMaio, F. (2016). Simultaneous Optimization of Biomolecular Energy Functions on Features from Small Molecules and Macromolecules. *Journal of Chemical Theory and Computation*, *12*(12), 6201–6212. https://doi.org/10.1021/acs.jctc.6b00819

Parmeggiani, F., Pellarin, R., Larsen, A. P., Varadamsetty, G., Stumpp, M. T., Zerbe, O., Caflisch, A., & 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/10.1016/j.jmb.2007.12.014

Plückthun, A. (2015). Designed Ankyrin Repeat Proteins (DARPins): Binding Proteins for Research, Diagnostics, and Therapy. *Annual Review of Pharmacology and Toxicology*, *55*(1), 489–511. https://doi.org/10.1146/annurev-pharmtox-010611-134654

Quinn, T. P., Tweedy, N. B., Williams, R. W., Richardson, J. S., & Richardson, D. C. (1994). Betadoublet: de novo design, synthesis, and characterization of a beta-sandwich protein. *Proceedings of the National Academy of Sciences*, *91*(19), 8747–8751. https://doi.org/10.1073/pnas.91.19.8747

Rackers, J. A., Wang, Z., Lu, C., Laury, M. L., Lagardère, L., Schnieders, M. J., Piquemal, J.-P., Ren, P., & Ponder, J. W. (2018). Tinker 8: Software Tools for Molecular Design. *Journal of Chemical Theory and Computation*, *14*(10), 5273–5289. https://doi.org/10.1021/acs.jctc.8b00529

Ramachandran, G. N., Ramakrishnan, C., & Sasisekharan, V. (1963). Stereochemistry of polypeptide chain configurations. *Journal of Molecular Biology*, *7*, 95–99. https://doi.org/10.1016/s0022-2836(63)80023-6

Raveh, B., London, N., & Schueler-Furman, O. (2010). Sub-angstrom modeling of complexes between flexible peptides and globular proteins. *Proteins: Structure, Function, and Bioinformatics*, *78*(9), 2029–2040. https://doi.org/10.1002/prot.22716

Regan, L. (1993). The Design of Metal-Binding Sites in Proteins. *Annual Review of Biophysics and Biomolecular Structure*, *22*(1), 257–281. https://doi.org/10.1146/annurev.bb.22.060193.001353

Richardson, J. S., & Richardson, D. C. (1989). The de novo design of protein structures. *Trends in Biochemical Sciences*, *14*(7), 304–309. https://doi.org/10.1016/0968-0004(89)90070-4

Roberts, K. E., Gainza, P., Hallen, M. A., & Donald, B. R. (2015). Fast gap-free enumeration of conformations and sequences for protein design. *Proteins: Structure, Function, and Bioinformatics*, *83*(10), 1859–1877. https://doi.org/10.1002/prot.24870

Romero-Romero, S., Costas, M., Silva Manzano, D.-A., Kordes, S., Rojas-Ortega, E., Tapia, C., Guerra, Y., Shanmugaratnam, S., Rodríguez-Romero, A., Baker, D., Höcker, B., & Fernández-Velasco, D. A. (2021). The Stability Landscape of de novo TIM Barrels Explored by a Modular Design Approach. *Journal of Molecular Biology*, *433*(18), 167153. https://doi.org/10.1016/j.jmb.2021.167153

Salemme, F. R. (1983). Structural properties of protein β-sheets. *Progress in Biophysics and Molecular Biology*, *42*, 95–133. https://doi.org/10.1016/0079-6107(83)90005-6

Schreier, B., Stumpp, C., Wiesner, S., & Höcker, B. (2009). Computational design of ligand binding is not a solved problem. *Proceedings of the National Academy of Sciences*, *106*(44), 18491–18496. https://doi.org/10.1073/pnas.0907950106

Shapovalov, M. V., & Dunbrack, R. L. (2011). A Smoothed Backbone-Dependent Rotamer Library for Proteins Derived from Adaptive Kernel Density Estimates and Regressions. *Structure*, *19*(6), 844–858. https://doi.org/10.1016/j.str.2011.03.019

Sheng, Y., Yin, Y., Ma, Y., & Ding, H. (2021). Improving the Performance of MM/PBSA in Protein–Protein Interactions via the Screening Electrostatic Energy. *Journal of Chemical Information and Modeling*, *61*(5), 2454–2462. https://doi.org/10.1021/acs.jcim.1c00410

Singh, J., & Thornton, J. M. (1992). *Atlas of Protein Side-Chain Interactions*. IRL Press at Oxford University Press.

Stiel, A. C., Feldmeier, K., & Höcker, B. (2014). *Identification of Protein Scaffolds for Enzyme Design Using Scaffold Selection* (pp. 183–196). https://doi.org/10.1007/978-1-4939-1486-9_9

Stiel, A. C., Nellen, M., & Höcker, B. (2016). *PocketOptimizer and the Design of Ligand Binding Sites* (pp. 63–75). https://doi.org/10.1007/978-1-4939-3569-7_5

Sun, P. D., Foster, C. E., & Boyington, J. C. (2004). Overview of protein structural and functional folds. *Current Protocols in Protein Science*, *Chapter 17*(1), Unit 17.1. https://doi.org/10.1002/0471140864.ps1701s35

Thomson, A. R., Wood, C. W., Burton, A. J., Bartlett, G. J., Sessions, R. B., Brady, R. L., & Woolfson, D. N. (2014). Computational design of water-soluble α-helical barrels. *Science*, *346*(6208), 485–488. https://doi.org/10.1126/science.1257452

Tinberg, C. E., Khare, S. D., Dou, J., Doyle, L., Nelson, J. W., Schena, A., Jankowski, W., Kalodimos, C. G., Johnsson, K., Stoddard, B. L., & 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

Todd, A. E., Orengo, C. A., & Thornton, J. M. (1999). Evolution of protein function, from a structural perspective. *Current Opinion in Chemical Biology*, *3*(5), 548–556. https://doi.org/10.1016/S1367-5931(99)00007-1

Traoré, S., Allouche, D., André, I., de Givry, S., Katsirelos, G., Schiex, T., & Barbe, S. (2013). A new framework for computational protein design through cost function network optimization. *Bioinformatics*, *29*(17), 2129–2136. https://doi.org/10.1093/bioinformatics/btt374

Van Dorn, L. O., Newlove, T., Chang, S., Ingram, W. M., & Cordes, M. H. J. (2006). Relationship between Sequence Determinants of Stability for Two Natural Homologous Proteins with Different Folds. *Biochemistry*, *45*(35), 10542–10553. https://doi.org/10.1021/bi060853p

van Zundert, G. C. P., Rodrigues, J. P. G. L. M., Trellet, M., Schmitz, C., Kastritis, P. L., Karaca, E., Melquiond, A. S. J., van Dijk, M., de Vries, S. J., & Bonvin, A. M. J. J. (2016). The HADDOCK2.2 Web Server: User-Friendly Integrative Modeling of Biomolecular Complexes. *Journal of Molecular Biology*, *428*(4), 720–725. https://doi.org/10.1016/j.jmb.2015.09.014

Varadi, M., Anyango, S., Deshpande, M., Nair, S., Natassia, C., Yordanova, G., Yuan, D., Stroe, O., Wood, G., Laydon, A., Zídek, A., Green, T., Tunyasuvunakool, K., Petersen, S., Jumper, J., Clancy, E., Green, R., Vora, A., Lutfi, M., … Velankar, S. (2022). AlphaFold Protein Structure Database: Massively expanding the structural coverage of protein-sequence space with high-accuracy models. *Nucleic Acids Research*, *50*(D1), D439–D444. https://doi.org/10.1093/nar/gkab1061

Veltman, J. A., & Brunner, H. G. (2012). De novo mutations in human genetic disease. *Nature Reviews Genetics*, *13*(8), 565–575. https://doi.org/10.1038/nrg3241

Voet, A. R. D., Noguchi, H., Addy, C., Simoncini, D., Terada, D., Unzai, S., Park, S.-Y., Zhang, K. Y. J., & Tame, J. R. H. (2014). Computational design of a self-assembling symmetrical β-propeller protein. *Proceedings of the National Academy of Sciences*, *111*(42), 15102–15107. https://doi.org/10.1073/pnas.1412768111

Watson, J. L., Juergens, D., Bennett, N. R., Trippe, B. L., Yim, J., Eisenach, H. E., Ahern, W., Borst, A. J., Ragotte, R. J., Milles, L. F., Wicky, B. I. M., Hanikel, N., Pellock, S. J., Courbet, A., Sheffler, W., Wang, J., Venkatesh, P., Sappington, I., Torres, S. V., … Baker, D. (2023). De novo design of protein structure and function with RFdiffusion. *Nature*, *620*(7976), 1089–1100. https://doi.org/10.1038/s41586-023-06415-8

Yang, W., & Lai, L. (2017). Computational design of ligand-binding proteins. *Current Opinion in Structural Biology*, *45*, 67–73. https://doi.org/10.1016/j.sbi.2016.11.021

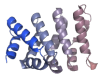
# IX. Research Articles

## 1. Modular Peptide Binders: Development of a Predictive Technology as Alternative for Reagent Antibodies

Florian J. Gisdon*, **Josef P. Kynast***, Merve Ayyildiz*, Anna V. Hine, Andreas Plückthun and Birte Höcker

*Biological Chemistry*. **2022**; 403(5-6): 535-543

* equal contribution



DE GRUYTER Biol. Chem. 2022; 403(5–6): 535–543

Review

Florian J. Gisdon, Josef P. Kynast, Merve Ayyildiz, Anna V. Hine, Andreas Plückthun and Birte Höcker*

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

https://doi.org/10.1515/hsz-2021-0384
Received September 30, 2021; accepted January 11, 2022; published online January 28, 2022

**Abstract:** Current biomedical research and diagnostics critically depend on detection agents for specific recognition and quantification of protein molecules. Monoclonal antibodies have been used for this purpose over decades and facilitated numerous biological and biomedical investigations. Recently, however, it has become apparent that many commercial reagent antibodies lack specificity or do not recognize their target at all. Thus, synthetic alternatives are needed whose complex designs are facilitated by multidisciplinary approaches incorporating experimental protein engineering with computational modeling. Here, we review the status of such an engineering endeavor based on the modular armadillo repeat protein scaffold and discuss challenges in its implementation.

**Keywords:** affinity reagent; armadillo repeat proteins; computational design; directed evolution; library generation; protein-peptide interface.

## Introduction

Current biomedical research relies on the use of reagent antibodies to detect biomolecules in medical diagnostics and basic life science research. The development of a chimeric antibody in 1984 (Morrison et al. 1984; Neuberger et al. 1984) as a first recombinant antibody opened new possibilities for the development of therapeutics and applications as affinity reagents. Recombinant production allows one to define the sequence, which is important as it ensures the reproducibility of experiments and reliability of results. In contrast, the sequence of monoclonal antibodies is not directly known, but can be obtained via protein sequencing, though it is time-consuming and costly. The efficient production and sophisticated technology of monoclonal antibodies that are derived by immunization is certainly a reason for their prevalence as specific binders in biological sciences. But the use of animal-derived antibodies has been more and more brought into question. On the one hand information about monoclonal antibodies, which are derived from hybridoma cell lines, can get lost due to cell line death or gene loss (Bradbury and Plückthun 2015). But apart from that, increasing awareness arose from the observation that animal-derived antibodies are varying between separate batches and often lack distinct specificity, which affects experimental reproducibility (Baker 2015). To address this issue, DNA sequencing can be applied and in fact, a recombinant production should then be possible. While this is technologically feasible, it is unfortunately not routinely done for commercially available reagent antibodies, which is likely due to commercial reasons (Bradbury and Plückthun 2015).

Consequently, in an interdisciplinary meeting in 2019, 35 years after the first recombinant antibody had been engineered, the development and use of animal-free recombinant antibodies were discussed with the objective to foster their increased use in basic research (Groff et al. 2020). Still, conventional antibodies are widely used in research applications, but antibodies with poor specificities or the

Florian J. Gisdon, Josef P. Kynast and Merve Ayyildiz contributed equally to this work.

***Corresponding author: Birte Höcker**, Department of Biochemistry, University of Bayreuth, Universitätsstr. 30, D-95447 Bayreuth, Germany, E-mail: birte.hoecker@uni-bayreuth.de. https://orcid.org/0000-0002-8250-9462
**Florian J. Gisdon, Josef P. Kynast and Merve Ayyildiz,** Department of Biochemistry, University of Bayreuth, D-95447 Bayreuth, Germany, E-mail: florian.gisdon@uni-bayreuth.de (F.J. Gisdon), josef.kynast@uni-bayreuth.de (J.P. Kynast), Merve.Ayyildiz@uni-bayreuth.de (M. Ayyildiz)
**Anna V. Hine,** College of Health and Life Sciences, Aston University, Birmingham B4 7ET, UK, E-mail: a.v.hine@aston.ac.uk
**Andreas Plückthun,** Department of Biochemistry, University of Zurich, CH-8057 Zürich, Switzerland, E-mail: plueckthun@bioc.uzh.ch

lack of reproducibility led to the development of alternative affinity reagents, which can be produced recombinantly and hence ensure their reliability (Groff et al. 2015). Recombinant production requires one to know the sequence of the reagent, and thus makes experimental results transparent and reproducible. Furthermore, recombinantly produced affinity reagents are truly monoclonal but can also be made polyclonal by using exactly defined pools. This procedure even provides knowledge about the full composition of the reagent mixture.

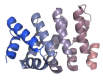
The first recombinant affinity reagents have been immunoglobulin derivatives. Immunoglobulins consist of a tail region, the Fc fragment, which interacts with cellular receptors, and a Fab fragment, which binds to antigens. In 1989 the first Fc-fusion protein was described, a fusion of the Fc fragment with the cell-surface glycoprotein CD4 (Capon et al. 1989). In the meantime, Fc-fusion proteins have been used as reagents for immunotherapy (to harvest their long half-lives) and laboratory research (to exploit detection reagents against the Fc part) (Duivelshof et al. 2021; Flanagan et al. 2007; Liu and Yu 2016), and the Fab regions have been utilized as recombinant affinity reagents (Conroy et al. 2017; Shih et al. 2012). However, the structure of antibodies entails technical challenges such as production in eukaryotic cells to obtain the required disulfide pattern and/or glycosylation (Gebauer and Skerra 2020). Such considerations have supported the development of alternative binding reagents which are not based on the immunoglobulin fold. First alternatives were e.g., based on natural folds such as fibronectin (Koide et al. 1998) or lipocalin (Beste et al. 1999), leading to monobodies or anticalins as designed affinity reagents. For both affinity reagents loop regions can be randomized to generate different variants, which can be selected for specific targets. However, a change of the scaffold has just been the first step. The mentioned affinity reagents are restricted by their size and variability in their binding mode. For a better adjustability and for control of the binding properties designed repeat proteins have been considered as scaffolds. Designed ankyrin repeat proteins (DARPins), for example, have the advantage to be fully characterized, and their size can be adapted by addition of further repeats (Binz et al. 2004; Forrer et al. 2003; Plückthun 2015). The repeats can also be easily randomized, which allows for a great variability that can be screened (see below) to find good binding reagents. Further, the binding site is slightly concave, which is favorable for binding large epitopes. However, DARPins have to be devel-oped anew for every target, nevertheless, they are used as innovative affinity reagents (Schilling et al. 2021).

A fundamentally different concept, which is also based on a repeat protein scaffold, is currently investigated within the collaborative 'Predictive Reagent Antibody Replacement Technology' (for PRe-ART) project. Here, the alternative affinity reagent is a designed armadillo repeat protein (dArmRP, see Figure 1), which can be varied in length of the concave binding surface, analogous to DARPins. However, the modularity of dArmRPs gives them an additional unique feature, as each internal repeat harbors binding capabilities for exactly two adjacent amino acid residues of a target. Furthermore, the distance between the repeats is optimized to match the periodicity of a peptide chain, so that the dArmRP can be applied to bind linear epitopes (Reichen et al. 2014). By designing different repeat modules, with specificities for all individual amino acids, a universal toolkit will be created from which desired binders can easily be assembled. This idea radically rethinks the established concept of affinity reagents and will affect a broad user base, as the recognition of linear epitopes is fundamental in many research applications, for instance protein purification with affinity tags or the recognition of unstructured regions such as found on western blots or intrinsically disordered proteins. Further, such unstructured regions are often targets for post-translational modifications such as phosphorylation and play an important role in the function of proteins (Dyson and Wright 2005; Liu et al. 2020).

This growing number of applications strengthens the need for robust and well-defined affinity reagents, which are less cost- and time-consuming in their production compared to commonly used reagent antibodies. This is especially important since many commercial reagent antibodies lack specificity or do not recognize their target at all. The modular dArmRPs define an innovative technology that fully reexamines the concept of existing affinity reagents and promises to revolutionize their applications.

## Armadillo repeat proteins are modular scaffolds for peptide recognition

The natural armadillo repeat protein (ArmRP) scaffold harbors unique and useful features necessary for its development into recombinant affinity reagents. It is comprised of homologous structural units that stack to form an elongated, rigid structure. Crystal structures show that natural ArmRPs bind stretched peptides of up to six amino acids (Conti and Kuriyan 2000; Conti et al. 1998; Graham et al. 2000). This binding of peptides in extended conformation reveals a conserved modular recognition mechanism which is a key feature of the ArmRP scaffold. Every second main



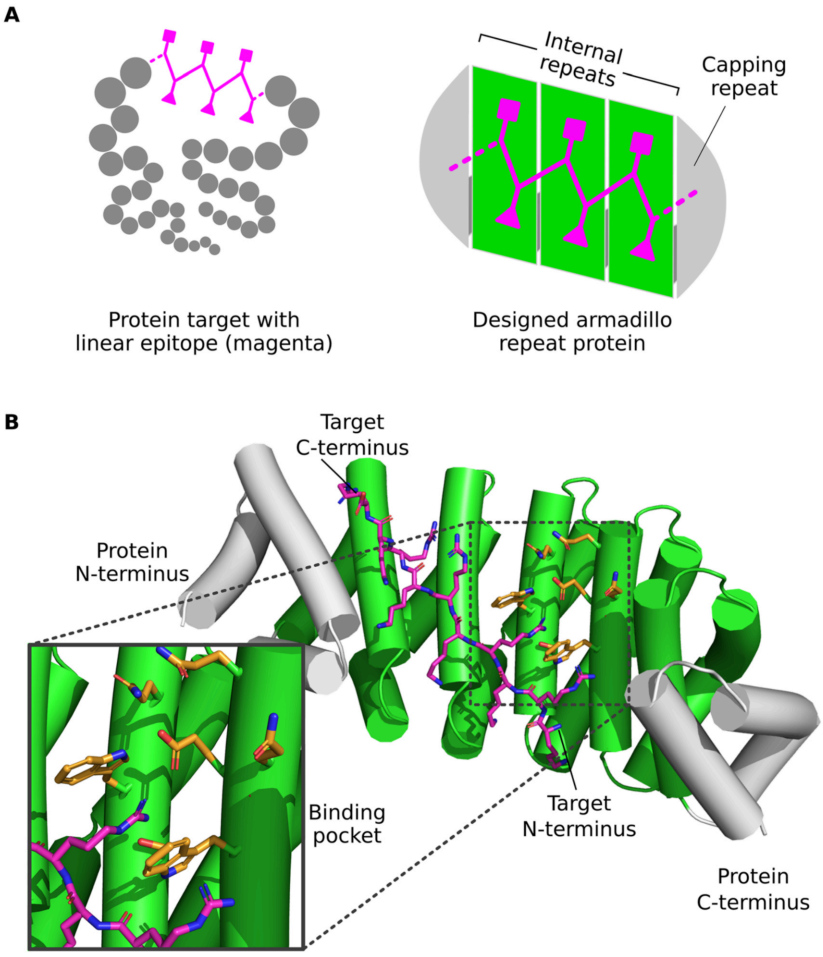



**Figure 1:** The modular nature of designed armadillo repeat proteins (dArmRPs). (A) Dipeptide units of a linear peptide stretch are bound in a modular fashion by dArmRPs. (B) The crystal structure of a dArmRP in complex with its target (PDB-ID: 5AEI) shows the modularity of binding of the extended peptide (magenta, as sticks) to the repeat modules of the protein (green, as cartoon). The residues of one arginine binding pocket are highlighted (orange, as sticks).

chain peptide bond of the target is held in place by a conserved asparagine residue on every ArmRP repeat. These interactions provide a general affinity and secure the regularity of the binding interactions. Each ArmRP repeat unit further binds two adjacent amino acid side chains in the target sequence in a specific manner (Figure 1).

These features were enhanced and regularized in iterative rounds of engineering. Using a consensus approach followed by computational and structural engineering for stability yielded a highly stable dArmRP, which consists of perfectly stackable repeats and optimized cap structures (Alfarano et al. 2012; Madhurantakam et al. 2012; Parmeggiani et al. 2008). Each repeat is 42 amino acids long and forms three alpha-helices. The assembled repeats again form an extended superhelical structure. Reichen et al. (2016) analyzed the variation in curvature of natural ArmRPs and identified a repeat pair in yeast importin-alpha with the ideal curvature geometry for optimal binding of an extended peptide. Based on binding pockets from importin-alpha, a dArmRP could be built that has picomolar affinity to its target peptide of alternating lysine and arginine residues (Hansen et al. 2016). The crystal structure of this protein, built from five identical repeats and N- and C-terminal caps in complex with a $(KR)_5$ peptide, confirmed the regular binding mode (Figure 1B). It lays the groundwork for the design of tailored binders with specific affinities by the assembly of dipeptide-specific dArmRP modules.

For the development of a diverse set of binding modules for different amino acids, a consistent design and testing approach is crucial for success as we discuss below. Furthermore, it is important that other binding modes are eliminated as it had been observed that repetitive sequences lead to register shifts and flipping of peptides during selections from libraries, which affects the investigation of binding specificities (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. 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. The lock could further be moved from the N-terminus of the dArmRP to its middle nicely highlighting the modularity of the system (Ernst et al. 2020).

With stability and modular binding of dArmRPs established and with an efficient locking system in place, the main goal is now to develop modules that can bind any other amino acid including negatively charged or even phosphorylated ones. Clearly, further adjustments of the dArmRP scaffold will also be necessary as neighboring binding pockets and combination of modules might have effects on the overall binder. However, the current challenge is to identify sequences that form binding pockets for other amino acids and thereby design new binding modules. Here, a consistent strategy to reduce the number of theoretical binding pocket sequences to an experimentally testable level is the key to success.

## Experimental strategies in the design of specific dArmRP modules

The repeat units of dArmRPs bind two adjacent amino acids in an alternating orientation (Figure 1). Originating from the importin-alpha framework one binding pocket is specific for arginine and the other one is specific for lysine (Hansen et al. 2016). The specificity of each pocket has to be adjusted to recognize other amino acids by mutating binding pocket residues. For an efficient search of specific binding pockets, DNA library selection technologies play a major role. These techniques allow to rapidly screen large numbers of DNA sequences encoding for the target protein that are randomized in regions responsible for the desired interaction. A complete randomization of a dArmRP module, however, is not useful. First, only a small fraction of residues is in direct contact with the ligand side chain of the target peptide. Second, uncontrolled randomization will incorporate unwanted termination codons. And third, due to the assignment of 64 codons to 20 canonical amino acids and termination codons, the distribution of amino acids will heavily differ at each position of randomization. Hence, the probability for certain amino acids to occur will be drastically reduced and create a bias. Additionally, the total number of sequences necessary to exhaustively screen a library will exponentially increase per randomized amino acid position.

A solution to these difficulties and to reduce the number of DNA sequences necessary for exhaustive screening is the use of MAX randomization as a non-degenerate saturation mutagenesis technology (Hughes et al. 2003). This technology allows one to build libraries with exactly 20 codons (one for each amino acid) or a desired subset of those for the randomized position. As a related technique ProxiMAX even allows to saturate multiple contiguous codons in a non-degenerate manner (Ashraf et al. 2013). Both methods require no specialized chemistry, reagents, or equipment. Ultimately, the use of the MAX techniques allows to generate DNA libraries without amino acid bias, termination codons, and degeneracy. Limiting both library size and degeneracy is critical to maximizing the output from the applied screening technology.

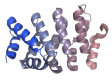
Three main selection technologies exist that could be used for the selection of dArmRP libraries: phage display, ribosome display, and yeast display. Because of the starting consensus scaffold being dominated by importin-alpha, the libraries are heavily biased to bind positively charged peptides, which creates difficulties during panning. As ribosome display uses highly negatively charged mRNA molecules and filamentous phages are equally negatively charged, it is not possible to select specific binding to the positively charged peptides. In contrast, selections by yeast display can be successfully performed, as the yeast surface is apparently not as negatively charged.

During selections of pockets for individual amino acids it is key that the peptides bind specifically and efficiently to the dArmRPs. Due to the repetitive nature of the dArmRP binding pockets the target peptide can bind in different registers. To avoid flipping or sliding of the peptide it is important to provide a binding pocket that locks the peptide into place. This was achieved by grafting a binding site from β-catenin into the dArmRP as described above (Ernst et al. 2020). The lock allows that selections can now be focused onto the binding pocket residues to the new target side chain to which specificity should be achieved.

Selection by yeast display is a very powerful technique and many different variants can be sorted in a high-throughput manner. Nonetheless, even with this technique only a library of a certain size can be screened. While library design by MAX randomization is a huge advantage as it allows particular residue types in predefined positions, screening of these libraries is still time-consuming. Therefore, it is useful to focus the libraries further to the most likely variants. Here, computational techniques can help to predict precise mixtures of amino acids for each position of randomization.

## Computational strategies in the design of specific dArmRP modules

The modularity of the dArmRP scaffold allows for the individual design of a single pocket at a time. However, there is still an enormous number of residue combinations and degrees of freedom that need to be sampled. Therefore, a



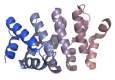
computational pre-selection of possible binding modes is useful and necessary to enable efficient experimental screening as described above.

The computational sampling of a very large number of combinations and degrees of freedom is challenging as well, though the past decade has seen significant improvements in the development and application of computational methods for protein design (Lechner et al. 2018). With algorithmic improvements and technological progress in computer hardware, new protein design approaches yielded increased accuracy and efficiency by allowing more flexibility and by applying simultaneous sampling of multiple sequences (Friedland et al. 2008; Murphy et al. 2012; Saunders and Baker 2005; Yin et al. 2007). In addition to the applied flexibility, different design objectives like creating single-state, multi-state, or ensemble-based designs influence the quality of the computational predictions.

One powerful method for sampling flexibility in computational protein design is Molecular Dynamics (MD) that has proven to provide valuable insights on protein stability, dynamics, and macromolecular interactions (Simonson et al. 2020). Technological advances such as parallelization on graphics processing units (GPU) have significantly accelerated MD calculations. Although the high computational cost is still a limiting factor, MD simulations of microseconds on a single GPU for protein systems such as dArmRPs are achievable within several days (Lazim et al. 2020). However, the design of binding pockets requires to sample many different combinations of amino acids, which spans an enormous combinatorial search space. For the evaluation of such a large amount of different variants, provable algorithms, including Branch and Bound, Dead-End Elimination, and Dynamic Programming, that have been successfully applied to protein design problems with backbone flexibility are promising developments for efficient calculations (Desmet et al. 1992; Gordon and Mayo 1999; Jou et al. 2016; Leaver-Fay et al. 2005; Ojewole et al. 2018). Also, deep learning techniques have experienced a large gain in interest in protein redesign since novel deep learning architectures achieve extraordinary prediction results in various fields due to clever model design and effective pattern recognition (Jumper et al. 2021; Krizhevsky et al. 2017). Thus, prediction of protein design features might be applicable on multiple sequences in a drastically smaller timescale. However, many state-of-the-art machine learning models, especially deep learning models, have not been extensively explored for protein design applications so far (Gao et al. 2020; Wang et al. 2018; Xu et al. 2020).

Nonetheless, computational strategies are often used as a complementary approach to experimental methods since experimental work is time-consuming and expensive (Chen and Keating 2012; Ernst et al. 2020; Liang et al. 2021). Within the multidisciplinary approach of PRe-ART, computational tools with diverse features help to characterize existing and to design new binding modules. For the characterization of new or existing binding pockets, different computational options are available. Tools can be used to screen the possible sequence space with methods such as the non-exhaustive screening and scoring protocols FastDesign (Loshbaugh and Kortemme 2020; Maguire et al. 2021) and coupled moves (Ollikainen et al. 2015) included in the software suite Rosetta. FastDesign performs iterations of side chain repacking and global minimization to find energy minima while exchanging predefined residues within the sequence. Coupled moves, however, alters backbone and sidechain conformations as well as the sequence at a time, to allow for more effective sampling. Further, several well-established computational methods, including *flex ddG* and *Branch and Bound Over K** (*BBK**) algorithms implemented in the Rosetta and Osprey protein design suites, respectively, allow to specifically target single binder sequences with exchanges in one residue position (Barlow et al. 2018; Ojewole et al. 2018). The *flex ddG* protocol incorporates backrub motion to accurately calculate binding affinity changes upon mutation. The *BBK** algorithm efficiently evaluates the partition function to calculate the binding affinity, while additionally allowing for continuous flexibility. Complementing these algorithms, MD simulations can support the analysis of the influence of mutations on the dynamics and the protein-ligand interactions of the system.

To predict promising mutations in a binding pocket in the first place that potentially develops a specific binding ability for the desired peptide, the software suite ATLIGATOR has been developed (Kynast et al. 2022). It is based on a knowledge-based approach that extracts pairwise interactions from existing structures to be used in the design of new binding pockets. Furthermore, it incorporates the detection of frequent interaction groups for specific amino acid side chains. Subsequent evaluation of the suggested binding pockets from ATLIGATOR can be performed by algorithms such as *flex ddG* or *BBK**, which can be complemented with MD simulations. The combination of the described methods results in a detailed understanding of the new binding pocket candidates. Hence, even if the computational prediction of exact binder sequences is not entirely possible, the multidisciplinary PRe-ART approach established a feedback loop to use the findings from computational modeling for the design of focused libraries for experimental screening (Figure 2).



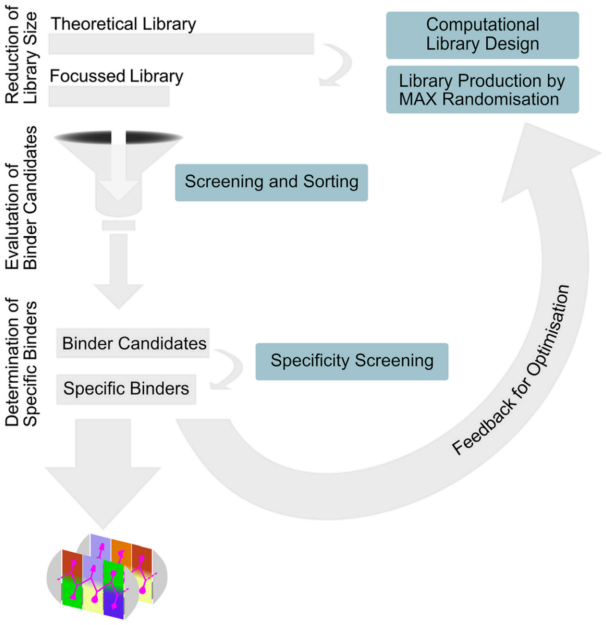



**Figure 2:** Workflow in the engineering of binding modules. Libraries are designed, synthesized, screened, and evaluated, providing feedback to the input techniques. The overall loop creates an ensemble of binding modules that can later be assembled to recognize predefined target peptides.

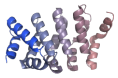
## Complementarity of experimental and computational design

The design of specific protein-protein or protein-peptide interactions with experimental screening and selection methods as well as computational modeling and prediction tools has progressed significantly. Experimental screening of DNA libraries with molecular display technologies (Levin and Weiss 2006) allows one to sample millions of sequences at once. When combining fluorescence activated cell sorting (FACS) with bacterial or yeast display approaches, cells can be sorted according to desired features. However, experimental screening methods suffer from exponentially growing complexity, the more residue positions are randomized. The use of techniques such as MAX randomization optimizes the codon selection to the minimum required and allows to define limited sets of amino acids for randomized positions (Hughes et al. 2003). Thus, the total number of sequences to screen for a complete coverage of desired amino acid sequences is minimized and the effective screening capacity is drastically increased.

Still, a theoretical library for complete randomization of a binding pocket quickly exceeds screening capacities. Thus, sequence space has to be reduced to a relevant set of sequences in the randomization. To screen only the relevant sequence space computational modeling can be used to exclude noninvolved positions and unfavorable mutations. An early attempt was by Voigt et al. (2001) who computationally focused a library and successfully selected proteins with increased stability. Also, in protein-protein interaction engineering, several groups have used computational design to focus libraries to select sequences compatible with the target fold that were screened for function later-on (Guntas et al. 2010; Hayes et al. 2002; Treynor et al. 2007). With increasing computational power and new protein modeling and design algorithms in the fields of deterministic (reviewed in Gainza et al. [2016]) or heuristic solving (reviewed for the Rosetta Suite in Kuhlman [2019]) as well as machine learning (reviewed in AlQuraishi [2021]) the potential to computationally focus libraries increased heavily.

The prediction of protein structures and stability are used successfully as a less cost- and labor-intense alternative to experimental methods. Even though the prediction of protein complex structures and their binding free energy is still not feasible for “bigger systems” in many cases, current software protocols can give crucial insights into those events (Barlow et al. 2018; Ojewole et al. 2018). Thus, functionally important positions can be identified or amino acid properties with potentially positive effects can be defined to reduce the size of the relevant search space. An incorporation of this knowledge into a designed library for experimental screening allows one to screen a bigger part of potentially advantageous sequences and to sort out disfavored sequences with a higher probability. Hence, the interplay of computational and experimental techniques leads to a higher likelihood to find variants with an improvement of the desired functionality. In the case of the PRe-ART project individual binding pockets are designed in dArmRPs, which detect and discriminate single amino acid side chains with high specificity. Randomization of all possible interacting positions would lead to a search space that largely exceeds screening capacities, which is why complementation with computational methods to design focused libraries is highly beneficial.

The precise objective of such a library design process for subsequent experimental screening is not immediately obvious. Possible priorities in creating such a library can be the inclusion of the best predicted sequences, the most frequently predicted sequences or a preferably high sequence diversity (Chen and Keating 2012), as well as sequences with highest affinity versus specificity. A reasonable choice would be to focus on *affinity* with computational selection and on *specificity* with subsequent experimental screening. Additionally, a library can be designed by scanning and scoring relevant shares of



the sequence space (Barlow et al. 2018; Gainza et al. 2013) or by considering interaction motifs found in natural proteins (Kynast et al. 2022). The results of screening such a focused library will potentially detect more desired binder sequences.

These variants of the binder sequences can be further characterized for their binding specificity as well as the structure of the protein-peptide complex. Such specific binding affinity information is crucial for the establishment and improvement of computational prediction tools (as seen in Barlow et al. [2018], Kadukova et al. [2021], and Spiliotopoulos et al. [2016]) to enable effective evaluation of methodological parameters. Furthermore, computational approaches can complement or explain experimental findings by simulations of the dynamic behavior of the dArmRP-peptide complex. Additionally, the selection rounds during experimental screening can be sequenced with next-generation sequencing techniques. By that strategy, a gigantic amount of sequencing data is generated whose analysis can lead to an even more sophisticated design of focused libraries or selection methods.

## Conclusions

Most affinity reagents for scientific research applications are still monoclonal antibodies derived from immunization, which either already exist and thus can be ordered from a supplier (catalog antibody), or they do not exist and have to be produced by immunization of animals (custom antibody). In fact, for most targets, epitopes and applications no suitable catalog antibody exists. And even if they exist, catalog antibodies frequently do not perform for reasons of cross-reactivity or low affinity, and the production of custom antibodies is costly in terms of time and money.

A major issue for common catalog or custom antibodies is that their genetic information is not available unless the antibody is sequenced in a labor-intensive step. However, applications with fusion proteins or the expression on cell or virus surfaces require the knowledge of the protein sequence to produce the binder recombinantly. Therefore, many catalog or custom antibodies are not suitable for such applications (Bradbury and Plückthun 2015). Additionally, common recombinant antibodies also have to be created anew for every new target sequence.

The collaborative PRe-ART project addresses these issues. A modular affinity reagent has been built based on the Armadillo repeat scaffold, where the modularity of the binder matches the target peptide architecture. Now, individual binding pockets are being designed to be specific for individual amino acids on the target that can later be combined. Thus, with an existing set of binding pockets in place it will be possible to assemble an affinity reagent for a specific target sequence in a very short time. Apart from slight adaptations at the pocket interfaces no further experimental selections and computational optimizations will be necessary during the assembly of new sequence-specific binding proteins.

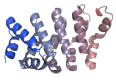
This fundamentally new concept allows one to bind linear target sequences in an unfolded state. Such stretches are often available at the termini of proteins or in linker regions, or they can be obtained by denaturation of the target protein as in SDS-PAGE or western blots. Unstructured targets of great interest are also the tails of receptors or regions of signal transduction molecules which are phosphorylated, or intrinsically disordered. Since unstructured regions are often post-translationally modified, these modular affinity reagents could be used to specifically target and investigate post-translational modifications. It would also be highly interesting to build pairs of binders for phosphorylated and unphosphorylated targets to visualize effects of candidate drugs on signaling pathways. Such an approach could accelerate mass spectrometry detection by orders of magnitude, circumvent labeling and thus permit to incorporate such a workflow into drug discovery. Because of the modular nature, “calibration” binders could be added that detect constant parts of the proteins in question, which would further add to the robustness of the concept.

Overall, the application of modular affinity reagents that can be assembled from predefined binding pockets has enormous potential for a wide range of applications. Because of the sequence-specific binding nature, these applications are completely out of reach of monoclonal antibodies or other conventional affinity reagent scaffolds.

**Author contributions:** All the authors have accepted responsibility for the entire content of this submitted manuscript and approved submission.
**Research funding:** This work was supported by H2020-FETopen-RIA grant agreement 764434 (‘PRe-ART’)
**Conflict of interest statement:** The authors declare no conflicts of interest regarding this article.

## References

Alfarano, P., Varadamsetty, G., Ewald, C., Parmeggiani, F., Pellarin, R., Zerbe, O., Plückthun, A., and Caflisch, A. (2012). Optimization of designed armadillo repeat proteins by molecular dynamics simulations and NMR spectroscopy. Protein Sci. 21: 1298–1314.



AlQuraishi, M. (2021). Machine learning in protein structure prediction. Curr. Opin. Chem. Biol. 65: 1–8.

Ashraf, M., Frigotto, L., Smith, M.E., Patel, S., Hughes, M.D., Poole, A.J., Hebaishi, H.R.M., Ullman, C.G., and Hine, A.V. (2013). ProxiMAX randomization: a new technology for non-degenerate saturation mutagenesis of contiguous codons. Biochem. Soc. Trans. 41: 1189–1194.

Baker, M. (2015). Blame it on the antibodies. Nature 521: 274–276.

Barlow, K.A., Ó Conchúir, S., Thompson, S., Suresh, P., Lucas, J.E., Heinonen, M., and Kortemme, T. (2018). Flex ddG: Rosetta ensemble-based estimation of changes in protein-protein binding affinity upon mutation. J. Phys. Chem. B 122: 5389–5399.

Beste, G., Schmidt, F.S., Stibora, T., and Skerra, A. (1999). Small antibody-like proteins with prescribed ligand specificities derived from the lipocalin fold. Proc. Natl. Acad. Sci. U.S.A. 96: 1898–1903.

Binz, H.K., Amstutz, P., Kohl, A., Stumpp, M.T., Briand, C., Forrer, P., Grütter, M.G., and Plückthun, A. (2004). High-affinity binders selected from designed ankyrin repeat protein libraries. Nat. Biotechnol. 22: 575–582.

Bradbury, A. and Plückthun, A. (2015). Reproducibility: standardize antibodies used in research. Nature 518: 27–29.

Capon, D.J., Chamow, S.M., Mordenti, J., Marsters, S.A., Gregory, T., Mitsuya, H., Byrn, R.A., Lucas, C., Wurm, F.M., Groopman, J.E., et al. (1989). Designing CD4 immunoadhesins for AIDS therapy. Nature 337: 525–531.

Chen, T.S. and Keating, A.E. (2012). Designing specific protein-protein interactions using computation, experimental library screening, or integrated methods. Protein Sci. 21: 949–963.

Conroy, P.J., Law, R.H.P., Caradoc-Davies, T.T., and Whisstock, J.C. (2017). Antibodies: from novel repertoires to defining and refining the structure of biologically important targets. Methods 116: 12–22.

Conti, E. and Kuriyan, J. (2000). Crystallographic analysis of the specific yet versatile recognition of distinct nuclear localization signals by karyopherin α. Structure 8: 329–338.

Conti, E., Uy, M., Leighton, L., Blobel, G., and Kuriyan, J. (1998). Crystallographic analysis of the recognition of a nuclear localization signal by the nuclear import factor karyopherin α. Cell 94: 193–204.

Desmet, J., De Maeyer, M., Hazes, B., and Lasters, I. (1992). The dead-end elimination theorem and its use in protein side-chain positioning. Nature 356: 539–542.

Duivelshof, B.L., Murisier, A., Camperi, J., Fekete, S., Beck, A., Guillarme, D., and D'Atri, V. (2021). Therapeutic Fc-fusion proteins: current analytical strategies. J. Separ. Sci. 44: 35–62.

Dyson, H.J. and Wright, P.E. (2005). Intrinsically unstructured proteins and their functions. Nat. Rev. Mol. Cell Biol. 6: 197–208.

Ernst, P., Zosel, F., Reichen, C., Nettels, D., Schuler, B., and Plückthun, A. (2020). Structure-guided design of a peptide lock for modular peptide binders. ACS Chem. Biol. 15: 457–468.

Flanagan, M.L., Arias, R.S., Hu, P., Khawli, L.A., and Epstein, A.L. (2007). Soluble Fc fusion proteins for biomedical research. In: Albitar, M. (Ed.), *Monoclonal antibodies: methods and protocols*. Humana Press, Totowa, pp. 33–52.

Forrer, P., Stumpp, M.T., Binz, H.K., and Plückthun, A. (2003). A novel strategy to design binding molecules harnessing the modular nature of repeat proteins. FEBS Lett. 539: 2–6.

Friedland, G.D., Linares, A.J., Smith, C.A., and Kortemme, T. (2008). A simple model of backbone flexibility improves modeling of side-chain conformational variability. J. Mol. Biol. 380: 757–774.

Gainza, P., Nisonoff, H.M., and Donald, B.R. (2016). Algorithms for protein design. Curr. Opin. Struct. Biol. 39: 16–26.

Gainza, P., Roberts, K.E., Georgiev, I., Lilien, R.H., Keedy, D.A., Chen, C.-Y., Reza, F., Anderson, A.C., Richardson, D.C., Richardson, J.S., et al. (2013). OSPREY: protein design with ensembles, flexibility, and provable algorithms. Methods Enzymol. 523: 87–107.

Gao, W., Mahajan, S.P., Sulam, J., and Gray, J.J. (2020). Deep Learning in protein structural modeling and design. Patterns 1: 100142.

Gebauer, M. and Skerra, A. (2020). Engineered protein scaffolds as next-generation therapeutics. Annu. Rev. Pharmacol. Toxicol. 60: 391–415.

Gordon, D.B. and Mayo, S.L. (1999). Branch-and-terminate: a combinatorial optimization algorithm for protein design. Structure 7: 1089–1098.

Graham, T.A., Weaver, C., Mao, F., Kimelman, D., and Xu, W. (2000). Crystal structure of a β-catenin/Tcf complex. Cell 103: 885–896.

Groff, K., Allen, D., Casey, W., and Clippinger, A.J. (2020). Increasing the use of animal-free recombinant antibodies. ALTEX, https://doi.org/10.14573/altex.2001071 (Epub ahead of print).

Groff, K., Brown, J., and Clippinger, A.J. (2015). Modern affinity reagents: recombinant antibodies and aptamers. Biotechnol. Adv. 33: 1787–1798.

Guntas, G., Purbeck, C., and Kuhlman, B. (2010). Engineering a protein-protein interface using a computationally designed library. Proc. Natl. Acad. Sci. U.S.A. 107: 19296–19301.

Hansen, S., Tremmel, D., Madhurantakam, C., Reichen, C., Mittl, P.R.E., and Plückthun, A. (2016). Structure and energetic contributions of a designed modular peptide-binding protein with picomolar affinity. J. Am. Chem. Soc. 138: 3526–3532.

Hayes, R.J., Bentzien, J., Ary, M.L., Hwang, M.Y., Jacinto, J.M., Vielmetter, J., Kundu, A., and Dahiyat, B.I. (2002). Combining computational and experimental screening for rapid optimization of protein properties. Proc. Natl. Acad. Sci. U.S.A. 99: 15926–15931.

Hughes, M.D., Nagel, D.A., Santos, A.F., Sutherland, A.J., and Hine, A.V. (2003). Removing the redundancy from randomised gene libraries. J. Mol. Biol. 331: 973–979.

Jou, J.D., Jain, S., Georgiev, I.S., and Donald, B.R. (2016). BWM*: a novel, provable, ensemble-based dynamic programming algorithm for sparse approximations of computational protein design. J. Comput. Biol. 23: 413–424.

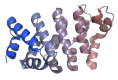
Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., et al. (2021). Highly accurate protein structure prediction with AlphaFold. Nature 596: 583–589.

Kadukova, M., Machado, K.D.S., Chacón, P., and Grudinin, S. (2021). KORP-PL: a coarse-grained knowledge-based scoring function for protein-ligand interactions. Bioinformatics 37: 943–950.

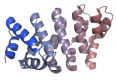
Koide, A., Bailey, C.W., Huang, X., and Koide, S. (1998). The fibronectin type III domain as a scaffold for novel binding proteins. J. Mol. Biol. 284: 1141–1151.

Krizhevsky, A., Sutskever, I., and Hinton, G.E. (2017). ImageNet classification with deep convolutional neural networks. Commun. ACM 60: 84–90.

Kuhlman, B. (2019). Designing protein structures and complexes with the molecular modeling program Rosetta. J. Biol. Chem. 294: 19436–19443.



Kynast, J.P., Schwägerl, F., and Höcker, B. (2022). ATLIGATOR: editing protein interactions with an atlas-based approach. bioRxiv, https://doi.org/10.1101/2022.01.19.476980.
Lazim, R., Suh, D., and Choi, S. (2020). Advances in molecular dynamics simulations and enhanced sampling methods for the study of protein systems. Int. J. Mol. Sci. 21: 1–20.
Leaver-Fay, A., Kuhlman, B., and Snoeyink, J. (2005). An adaptive dynamic programming algorithm for the side chain placement problem. Proc. Pacific Symp. Biocomput. 27: 16–27.
Lechner, H., Ferruz, N., and Höcker, B. (2018). Strategies for designing non-natural enzymes and binders. Curr. Opin. Chem. Biol. 47: 67–76.
Levin, A.M. and Weiss, G.A. (2006). Optimizing the affinity and specificity of proteins with molecular display. Mol. Biosyst. 2: 49–57.
Liang, T., Chen, H., Yuan, J., Jiang, C., Hao, Y., Wang, Y., Feng, Z., and Xie, X.-Q. (2021). IsAb: a computational protocol for antibody design. Briefings Bioinf. 22: 1–14.
Liu, N., Guo, Y., Ning, S., and Duan, M. (2020). Phosphorylation regulates the binding of intrinsically disordered proteins via a flexible conformation selection mechanism. Commun. Chem. 3: 1–9.
Liu, Y. and Yu, J. (2016). Oriented immobilization of proteins on solid supports for use in biosensors and biochips: a review. Microchim. Acta 183: 1–19.
Loshbaugh, A.L. and Kortemme, T. (2020). Comparison of Rosetta flexible-backbone computational protein design methods on binding interactions. Proteins Struct. Funct. Bioinf. 88: 206–226.
Madhurantakam, C., Varadamsetty, G., Grütter, M.G., Plückthun, A., and Mittl, P.R.E. (2012). Structure-based optimization of designed Armadillo-repeat proteins. Protein Sci. 21: 1015–1028.
Maguire, J.B., Haddox, H.K., Strickland, D., Halabiya, S.F., Coventry, B., Griffin, J.R., Pulavarti, S.V.S.R.K., Cummins, M., Thieker, D.F., Klavins, E., et al. (2021). Perturbing the energy landscape for improved packing during computational protein design. Proteins Struct. Funct. Bioinf. 89: 436–449.
Morrison, S.L., Johnson, M.J., Herzenberg, L.A., and Oi, V.T. (1984). Chimeric human antibody molecules: mouse antigen-binding domains with human constant region domains. Proc. Natl. Acad. Sci. U.S.A. 81: 6851–6855.
Murphy, G.S., Mills, J.L., Miley, M.J., Machius, M., Szyperski, T., and Kuhlman, B. (2012). Increasing sequence diversity with flexible backbone protein design: the complete redesign of a protein hydrophobic core. Structure 20: 1086–1096.
Neuberger, M.S., Williams, G.T., and Fox, R.O. (1984). Recombinant antibodies possessing novel effector functions. Nature 312: 604–608.
Ojewole, A.A., Jou, J.D., Fowler, V.G., and 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. J. Comput. Biol. 25: 726–739.
Ollikainen, N., de Jong, R.M., and Kortemme, T. (2015). Coupling protein side-chain and backbone flexibility improves the re-design of protein-ligand specificity. PLoS Comput. Biol. 11: 1–22.
Parmeggiani, F., Pellarin, R., Larsen, A.P., Varadamsetty, G., Stumpp, M.T., Zerbe, O., Caflisch, A., and Plückthun, A. (2008). Designed armadillo repeat proteins as general peptide-binding scaffolds: consensus design and computational optimization of the hydrophobic core. J. Mol. Biol. 376: 1282–1304.
Plückthun, A. (2015). Designed ankyrin repeat proteins (DARPins): binding proteins for research, diagnostics, and therapy. Annu. Rev. Pharmacol. Toxicol. 55: 489–511.
Reichen, C., Hansen, S., Forzani, C., Honegger, A., Fleishman, S.J., Zhou, T., Parmeggiani, F., Ernst, P., Madhurantakam, C., Ewald, C., et al. (2016). Computationally designed armadillo repeat proteins for modular peptide recognition. J. Mol. Biol. 428: 4467–4489.
Reichen, C., Hansen, S., and Plückthun, A. (2014). Modular peptide binding: from a comparison of natural binders to designed armadillo repeat proteins. J. Struct. Biol. 185: 147–162.
Saunders, C.T. and Baker, D. (2005). Recapitulation of protein family divergence using flexible backbone protein design. J. Mol. Biol. 346: 631–644.
Schilling, J., Jost, C., Mariuca Ilie, I., Schnabl, J., Buechi, O., Eapen, R.S., Truffer, R., Caflisch, A., and Forrer, P. (2021). Thermostable designed ankyrin repeat proteins (DARPins) as building blocks for innovative drugs. J. Biol. Chem. 298: 1–12.
Shih, H.H., Tu, C., Cao, W., Klein, A., Ramsey, R., Fennell, B.J., Lambert, M., Ní Shúilleabháin, D., Autin, B., Kouranova, E., et al. (2012). An ultra-specific avian antibody to phosphorylated tau protein reveals a unique mechanism for phosphoepitope recognition. J. Biol. Chem. 287: s44425–s44434.
Simonson, T., Mignon, D., Druart, K., Michael, E., Opuu, V., Polydorides, S., Villa, F., Gaillard, T., Panel, N., and Archontis, G. (2020). Physics-based computational protein design: an update. J. Phys. Chem. 124: 10637–10648.
Spiliotopoulos, D., Kastritis, P.L., Melquiond, A.S.J., Bonvin, A.M.J.J., Musco, G., Rocchia, W., and Spitaleri, A. (2016). dMM-PBSA: a new HADDOCK scoring function for protein-peptide docking. Front. Mol. Biosci. 3: 46.
Treynor, T.P., Vizcarra, C.L., Nedelcu, D., and Mayo, S.L. (2007). Computationally designed libraries of fluorescent proteins evaluated by preservation and diversity of function. Proc. Natl. Acad. Sci. U.S.A. 104: 48–53.
Voigt, C.A., Mayo, S.L., Arnold, F.H., and Wang, Z.G. (2001). Computational method to reduce the search space for directed protein evolution. Proc. Natl. Acad. Sci. U.S.A. 98: 3778–3783.
Wang, J., Cao, H., Zhang, J.Z.H., and Qi, Y. (2018). Computational protein design with deep learning neural networks. Sci. Rep. 8: 1–9.
Xu, C., Lu, P., Gamal El-Din, T.M., Pei, X.Y., Johnson, M.C., Uyeda, A., Bick, M.J., Xu, Q., Jiang, D., Bai, H., et al. (2020). Computational design of transmembrane pores. Nature 585: 129–134.
Yin, S., Ding, F., and Dokholyan, N.V. (2007). Modeling backbone flexibility improves protein stability estimation. Structure 15: 1567–1576.



# 2. ATLIGATOR: Editing Protein Interactions with an Atlas-Based Approach

**Josef P. Kynast**, Felix Schwägerl and Birte Höcker

*Bioinformatics* **2022** Nov 30; 38(23): 5199-5205

*Bioinformatics*, 38(23), 2022, 5199–5205
https://doi.org/10.1093/bioinformatics/btac685
Advance Access Publication Date: 19 October 2022
Original Paper

Structural bioinformatics

# ATLIGATOR: editing protein interactions with an atlas-based approach

**Josef Paul Kynast , Felix Schwägerl and Birte Höcker ***

Department of Biochemistry, University of Bayreuth, 95447 Bayreuth, Germany

*To whom correspondence should be addressed.

Associate Editor: Pier Luigi Martelli

Received on February 1, 2022; revised on September 24, 2022; editorial decision on October 9, 2022; accepted on October 17, 2022

## Abstract

**Motivation:** Recognition of specific molecules by proteins is a fundamental cellular mechanism and relevant for many applications. Being able to modify binding is a key interest and can be achieved by repurposing established interaction motifs. We were specifically interested in a methodology for the design of peptide binding modules. By leveraging interaction data from known protein structures, we plan to accelerate the design of novel protein or peptide binders.

**Results:** We developed ATLIGATOR—a computational method to support the analysis and design of a protein's interaction with a single side chain. Our program enables the building of interaction atlases based on structures from the PDB. From these atlases pocket definitions are extracted that can be searched for frequent interactions. These searches can reveal similarities in unrelated proteins as we show here for one example. Such frequent interactions can then be grafted onto a new protein scaffold as a starting point of the design process. The ATLIGATOR tool is made accessible through a python API as well as a CLI with python scripts.

**Availability and implementation:** Source code can be downloaded at github (https://www.github.com/Hoecker-Lab/atligator), installed from PyPI ('atligator') and is implemented in Python 3.

**Contact:** birte.hoecker@uni-bayreuth.de

## 1 Introduction

For protein design it is crucial to understand how proteins form interactions. Interactions can be formed intramolecularly to define stability or function as well as intermolecularly with various interaction partners such as solvent, small molecules, peptides or full proteins. Thus, the choice of a particular amino acid at a certain position in a protein is crucial to establish such favorable interactions between two or more amino acid residues. Hence, understanding how specific residue types interact with each other is of particular interest when creating newly designed proteins.

A description of the conformational space that is occupied by interacting amino acid side chains in known protein structures as well as relative positioning of both interaction partners can provide powerful information for protein design. Singh and Thornton (Singh and Thornton, 1992) already classified interactions between pairs of distinct residue types. Moreover, they described clusters of orientation and position combinations within these pairs of amino acids. In a similar approach, Vondrasek and colleagues investigated interaction energies of amino acid combinations calculated in gas-phase (Berka *et al.*, 2009, 2010; Galgonek *et al.*, 2017). For the analysis of enzymatic active sites, groups of amino acid residues from three-dimensional structures were categorized, based on sequence alignments (Porter *et al.*, 2004). While these studies led to a better understanding of amino acid interactions, their focus was more on analysis rather than design applications.

Recent developments, clearly moved toward designing protein structures as well as interaction surfaces. By extending amino acid pairs with information about their structural environment, two independent approaches successfully improved the quality of interaction data extracted from existing protein structures (Holland and Grigoryan, 2022; Jha *et al.*, 2010). In particular, the software dTERMen incorporates structural elements called tertiary structural motifs (TERMs) into the redesign of protein structures (MacKenzie *et al.*, 2016; Zhou *et al.*, 2020). TERMs were also recently used as surface-complementary fragments during protein design for peptide-binding (Swanson *et al.*, 2022).

Another important point was investigated by focusing next on positional and orientational information within amino acid-based interactions. For example, Polizzi and DeGrado generalized pairwise interactions by describing connections between amino acids and functional groups in so-called van der Mers (Polizzi and Degrado, 2020), while Liu *et al* developed the neighborhood-sensitive program NEPRE which is able to assess the quality of protein structures based on amino acid identities (Liu *et al.*, 2020).

We started looking into similar amino acid interaction groups for a specific design problem, namely the construction of custom-made modular peptide-binders based on armadillo repeat proteins.

Armadillo repeat proteins comprise a natural binding interface for elongated peptide stretches which was further refined to exhibit peptide binding in a regularized fashion (Hansen *et al.*, 2018, 2016). Thus, the transfer of existing motifs of known structures on the binding interface of a single repeat—also referred to as grafting—would be a crucial step to design new modules that can be assembled or incorporated in an existing peptide binder (Gisdon *et al.*, 2022).

So, we extended the atlas idea of Singh and Thornton to be applicable for design. By now, much more structural data is available that can be leveraged and made searchable for specific design applications. Some interaction modes can be found more frequently in nature and thus appear more favorable than others. Our aim was to make such natural interaction motifs explorable so that they can be searched and incorporated into the context of a protein scaffold. Such information allows to modify not only internal interactions within a protein, but also interactions with a different peptide or protein binding partner. Furthermore, the identification of frequently interacting residue groups plus their favored conformations opens the possibility to graft specialized binding pockets to specifically bind peptide or protein targets of interest.

To enable such rational design, we now present the software tool ATLIGATOR, short for ATlas-based LIGAnd binding site ediTOR. It allows the user to analyze frequent interaction modes of two or more amino acids and to directly apply this information to rational design approaches (Fig. 1). The program relies on data structures called *atlases* that contain descriptions of pairwise interactions from protein structures. A collection of structures that builds up such an *atlas* is a subgroup of all structures in the Protein Data Bank (Berman *et al.*, 2000) and can for example represent a certain type of fold based on classifiers of the SCOPe database (Fox *et al.*, 2014). Moreover, the ATLIGATOR tool also incorporates association rule learning in the form of frequent itemset mining to extract frequent groups of pairwise interactions based on single ligand residues from the *atlas*. These groups are called *pockets* and represent starting points for protein interface design tasks. This representation is based on the assumption that favorable interaction groups have been established during the evolution of the proteins of choice and are thus detected as *pockets*. A major key functionality of ATLIGATOR is the ability to visualize each individual step of the ATLIGATOR toolchain interactively. Furthermore, ATLIGATOR *atlas* and *pocket* datapoints can not only be browsed for individual amino acid combinations but can additionally be used in an integrated tool called Manual Design. Manual Design allows to use a protein–peptide complex structure of choice and alter the interaction surface by binding pocket grafting or manual mutations with recommendations based on *pocket* data. Hence, ATLIGATOR acts as a framework that offers a multitude of possible workflows. Besides the use of single parts for analysis or design, the setup offers a complete workflow from the analysis of interaction modes in protein structures all the way to the interactive application of protein interface design by leveraging previously accumulated knowledge.

## 2 System and methods

### 2.1 Algorithms

ATLIGATOR is a versatile toolkit for the analysis and the design of protein interactions. It focusses on single side chains of one interacting partner (the ligand) and its relation with multiple residues at the surface of the other interacting partner (the binder) that form a binding pocket for the single residue. *Atlases* are generated for all of these interactions within a user-specified set of complex structures. Through this focus on the single residue interaction level, the tool allows to detect promising interaction features. This knowledge can directly be applied to specific design problems of protein complex interfaces. The toolkit contains the following parts.

#### 2.1.1 Structure selection and preprocessing

The information gathered by ATLIGATOR is extracted from existing protein structures derived from the Protein Data Bank (PDB). The PDB contains an abundant collection of protein structures, which have been derived mainly from experimental methods. It is useful to be able to select the qualitatively best structures as well as the most fitting structures, e.g. from the same protein family, fold or class. Therefore, we provide the option to select structures based on one's own rationale or on identifiers of the SCOPe database, thereby creating sets of structures with shared structural or evolutionary background.

Furthermore, we allow to additionally filter structures for certain properties and quality criteria using a pre-selection and processing utility. This utility within ATLIGATOR is capable of applying the following filter criteria:

1. Specific protein families (e.g. by SCOPe query).
2. Minimum/maximum length of binder and ligand sequences.
3. Maximum distance between ligand and any binder residue.
4. Secondary structure content.

The underlying routine produces a directory of pre-processed pdb files, each containing one ligand-binding complex where ligand and binder are located in individual chains, removing unnecessary parts of ligand chains. These files are then used for *atlas* generation after an optional filtering step.

**Algorithm 1:** Atlas Datapoint Extraction as Simplified Python Code.

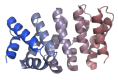
```
for structure in all_structures:
    ligands = find_ligands(structure)
    for ligand_residue in ligands:
        for binder_residue in residues_within_radius(ligand, structure):
            icoor = get_internal_coordinates_from(ligand_residue)
            for atom in ligand_residue:
                atom = icoor.external_to_internal(atom)
            for atom in binder_residue:
                atom = icoor.external_to_internal(atom)
            yield AtlasDatapoint(ligand_residue, binder_residue.re-
sidue_type, ligand_residue.origin, binder_residue.origin, ligan-
d_residue.atoms, binder_residue.atoms)
```

#### 2.1.2 Atlas generation

The pre-selected input structures contain external coordinates for the atoms of different ligand residues and binder proteins, respectively. An *atlas* is a collection of filtered and transformed datapoints, each describing an interaction between one residue of the ligand and one residue of a binder. The following algorithm describes on a coarse level how *atlas* datapoints are obtained from the input structures: For determining whether any pair of ligand and binder atoms are considered as interacting, we define specific interaction distances. These distances depend on the type of interactions between ligand and binder atom:

- Ionic: interactions between positively and negatively charged atoms (default: 8.0 Å).
- Aromatic: interactions between carbon atoms of aromatic rings (default: 6.0 Å).
- Hydrogen bonds: interactions between donor and acceptor atoms (default: 6.0 Å).
- All other interactions, e.g. hydrophobic (default: 4.0 Å).

The interacting residues are transformed into an internal coordinate system, which allows to detect patterns in pairwise interactions, seen from the perspective of ligand residues. It is defined similarly to Liu *et al* as follows (Liu *et al.*, 2020):



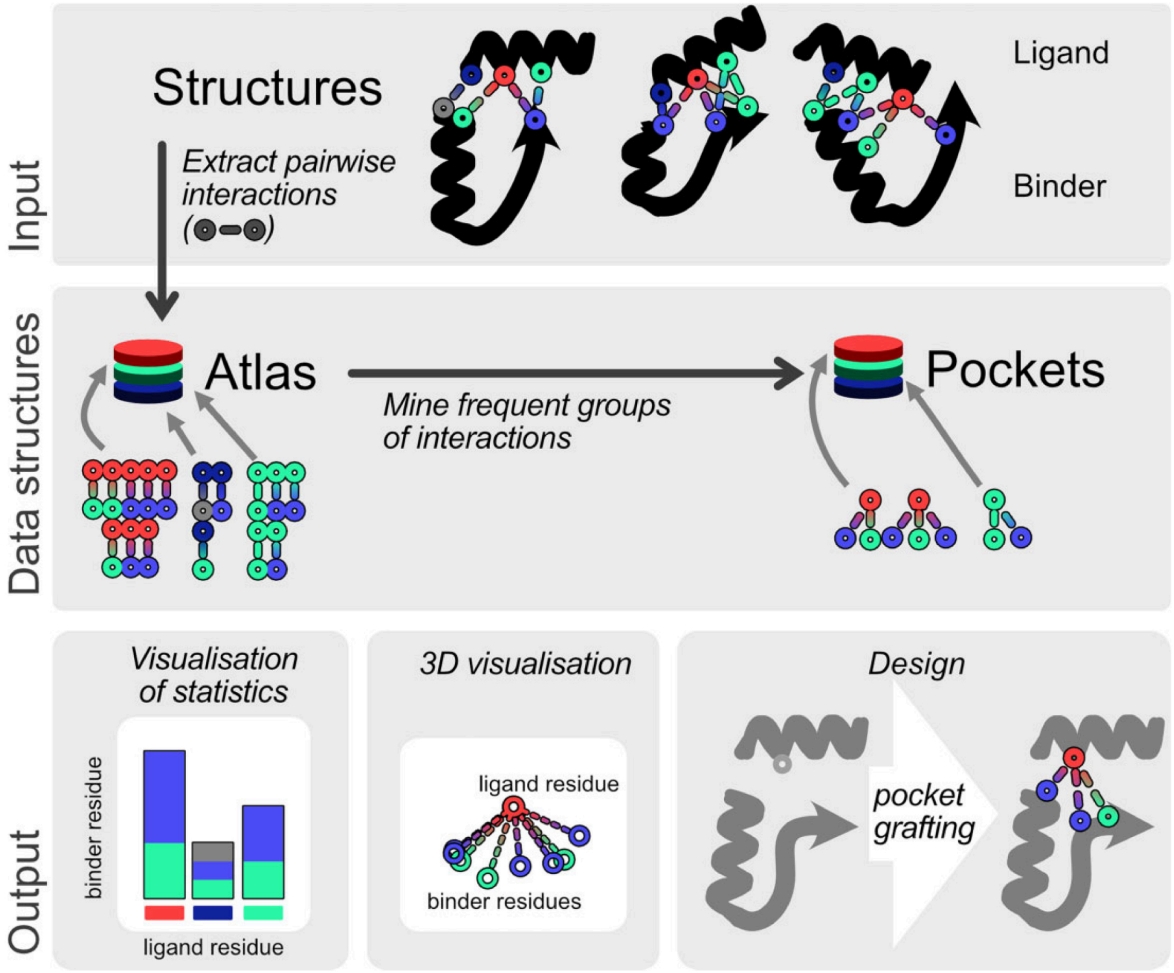



**Fig. 1.** Overview of the ATLIGATOR toolchain. The python-based tools of ATLIGATOR include the extraction of pairwise interactions from a *structure collection* as well as mining of frequent groups of interactions. Those tools as well as the input and output data can be accessed via a python API, meaning the source code as well as predefined scripts. Both types of interfaces can be used to analyze extracted interactions to find patterns which can be employed for new designs. This can be achieved by visualizing *atlas* statistics or 3D plotting of *atlas* and *pockets*. Moreover, ATLIGATOR includes the option to design new interaction sites based on binding pocket grafting

- The ligand residue's $C_\alpha$ atom is the origin.
- The ligand residue's $C_\beta$ atom is located on the $x$-axis of the internal coordinate system. (For glycine, we simulate a virtual $C_\beta$ atom for this purpose.)
- The ligand residue's C atom (carbonyl carbon) lies within the $xy$-plane.
- The ligand residue's N atom is defined with a negative $z$-value.

Every *atlas* is composed of datapoints storing individual interactions between two residues—a ligand and a binder residue. This collection of datapoints is grouped into *atlas pages* including all datapoints of a certain ligand residue type. *Atlas pages* are partitioned further into *atlas maps* including all datapoints of a combination of one ligand residue amino acid type interacting with one binder residue amino acid type (Fig. 2).

#### 2.1.3 Spatial similarity function

To compare *atlas* datapoints with each other or with designable binder residues we created a distance-orientation function to describe the spatial similarity of two residues $R_1$ and $R_2$. Assuming that they are both represented in the same, internal or external, coordinate system, their distance $|R_1 - R_2|$ is defined as follows:

$$|R_1 - R_2| = f_d \left| \overrightarrow{C_1^\alpha} - \overrightarrow{C_2^\alpha} \right| + f_o \angle \left( \overrightarrow{C_1^\beta} - \overrightarrow{C_1^\alpha}, \overrightarrow{C_2^\beta} - \overrightarrow{C_2^\alpha} \right) + f_s \angle \left( \overrightarrow{C_1^O} - \overrightarrow{C_1^\alpha}, \overrightarrow{C_2^O} - \overrightarrow{C_2^\alpha} \right) \quad (1)$$

The equation considers positions of Cα atoms of both interacting residues (where $C_1^\alpha$ denotes the $C_\alpha$ atom of $R_1$, etc.). Furthermore, the angles between two characteristic orientation vectors, namely those between $C_\alpha$ and $C_\beta$ (referred to as primary orientation below) as well as $C_\alpha$ and the carbonyl C (secondary orientation) of the residues are compared. The weight factors $f_d$, $f_o$ and $f_s$ can be adjusted by the user; the default values are $1.0\text{Å}^{-1}$ for $f_d$ and 2.0 for both $f_o$ and $f_s$.

#### 2.1.4 Pocket mining

Ligand–binder interactions as shown in the *atlas* do not have a purely pairwise nature. Several binder residues can instead contribute to binding one ligand residue. If similar binder residue groups form interactions to ligand residues in various structures, interaction patterns can be extracted and generalized. We call such a frequently occurring interaction pattern a *pocket*. Such *pockets* can be detected and extracted from an *atlas* database which is described below.

**Itemset extraction.** In its first step, the algorithm exploits the fact that datapoints of the *atlas* include their origin. Hence, we group all datapoints originating from the same ligand residue and call this a natural pocket. To detect which *pockets* are frequent, we reduce the

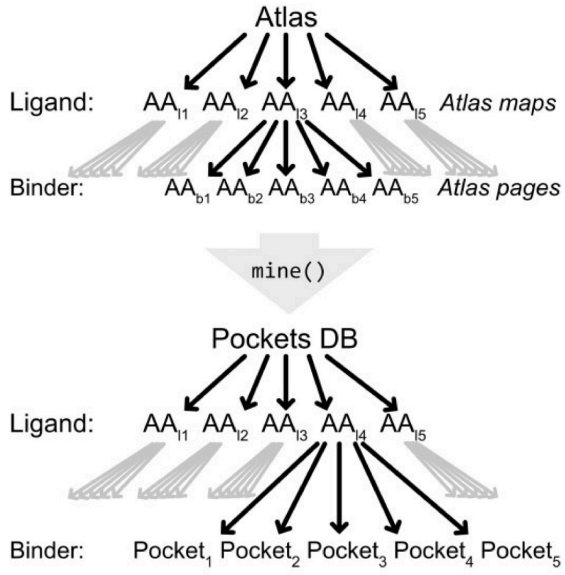



**Fig. 2.** Structure of atlas and pocket database. An *atlas* consists of *atlas maps* defining the ligand residue types that can be further subdivided into *atlas pages*. *Atlas pages* are defined by a specific ligand–binder residue type combination. *Pockets* are subgroups of the underlying *atlas* and can be structured by their ligand residue type as well. One *pocket* is defined by the ligand residue type in combination with a group of binder residue types. Both, *atlas pages* and *pockets* contain several *datapoints* that store exactly one pairwise interaction between two residues

information stored in these groups into natural itemsets, which are mere enumerations of binder residues that interact with the same ligand residue.

**Frequent itemset mining**. Depending on the size of the *atlas*, we obtain a large number of itemsets for every specific ligand residue type in this way. In the field of business intelligence, the so-called a-priori algorithm (Agrawal *et al.*, 1993) has been established. It guides customers to products frequently bought in combination with their product of interest by finding representative subsets of products in previous purchases. We apply this procedure in order to find representative subsets that are contained in a relevant share of natural itemsets extracted from the *atlas*. As a result, these subsets are groups of binder residue types that are found to interact frequently with a ligand residue type.

**Pocket extraction**. Frequent itemsets indicate which residues are part of a *pocket*, but ignore their structure. This information in turn is added during *pocket* extraction where the coordinates of the underlying *atlas* datapoints play a major role. In this step, natural pockets of the *atlas* are matched with frequent itemsets to identify and extract those pockets that represent the itemset (i.e. they include the same collection of residues or more). This adds the structural component of each pairwise interaction from the *atlas datapoint* to the group of amino acid types within the itemset. The resulting *pocket* stores the *datapoints* of the natural pockets in a superimposed way; technically, every superimposed *pocket* is a subset of the original *atlas*.

**Clustering, noise reduction and selection of representative**. Last, the information stored in every superimposed *pocket* is clustered. To this end, we employ a modified variant of the k-means algorithm (Chen *et al.*, 2004), utilizing the spatial similarity function shown in Section 2.1.3 for the calculation of cluster centroids and variances (i.e. the mean deviation of clustered *pocket* residues from the cluster centroid).

In order to reduce noise, the specific algorithm utilized here additionally ensures that the variance of a cluster is kept below a user-defined threshold (default value: 5.0 Å). By removing the most distant members from the cluster until the threshold is met, the noise present in superimposed *pockets* is reduced.

For every cluster, we ultimately select as the most representative element (also called medioid) the natural pocket with the least distance from the cluster centroid. This is an instance of the so-called assignment problem, which can be solved using the Hungarian Algorithm (Kuhn, 1955).

#### 2.1.5 Pocket grafting

*Pocket* grafting is a simple method that directly exploits the information available from the *pockets* for the creation of designed ligand–binder complexes based on a scaffold. It takes the best-matching *pocket* residues of a selected *pocket* according to the spatial similarity function (see Section 2.1.3) and applies corresponding mutations to binder residues. The details of the procedure are described by the following algorithm:The algorithm contains an additional adjustable parameter, the distance threshold $\theta$d (default: 12.0), which prevents the alignment of bad-matching *pocket* residues.

**Algorithm 2:** Pocket Grafting as Simplified Python Code.

```
icoor = get_internal_coordinates_from(ligand_residue)
for atom in mutable_binder_residues:
  atom = icoor.external_to_internal(atom)
for mutable_positions in mutable_binder_residues:
  for pocket_residue in pocket.binder_residues:
    score = calc_spatial_similarity(mutable_position, pocket_
    residue)
    scores[mutable_residue, pocket_residue] = score
occupied_positions = []
occupied_residues = []
for mutable_position, pocket_residue, score in sorted(scores):
  if score > threshold:
  break
  if mutable_position not in occupied_positions:
  if pocket residue not in occupied_residues:
    mutable_position.mutate_to(pocket_residue)
    occupied_positions.append(mutable_position)
    occupied_residues.append(pocket_residue)
```

The algorithm contains an additional adjustable parameter, the distance threshold $\theta$d (default: 12.0), which prevents the alignment of badmatching *pocket* residues.

#### 2.1.6 Quick graft

Pocket mining usually results in several *pockets* for each ligand residue type. To overcome the need to graft each *pocket* onto the scaffold individually and to select the best graft manually the quick graft protocol includes automatic grafting of the best matching pocket. To select the best pocket quick graft picks the pocket graft resulting in the best cumulative spatial similarity (see 2.1.3).

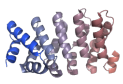
In the process of redesigning an interaction interface more than one ligand residue might be mutated. In this case, the grafted binding pockets need to complement each other to create the best fit for all exchanged ligand side chains. As a solution, quick graft detects conflicting grafts and finds the optimal set of pocket grafts with mutually exclusive positions to mutate. In addition, the best n grafted designs can be generated to give the user the option to compare and select the best grafts.

### 2.2 Implementation

The algorithms discussed in Section 2.1 are implemented in Python 3. To access the functionality, we deliver scripts with user argument parsing as well as a raw python API (see Fig. 1).

#### 2.2.1 Python API and CLI

The API as well as the supplemented python CLI (command-line interface) allow to generate and access all parts of the ATLIGATOR toolchain, visualize *atlases*, *pockets* and additional statistics and follow preprocessing steps. Furthermore, within the API the visualization of



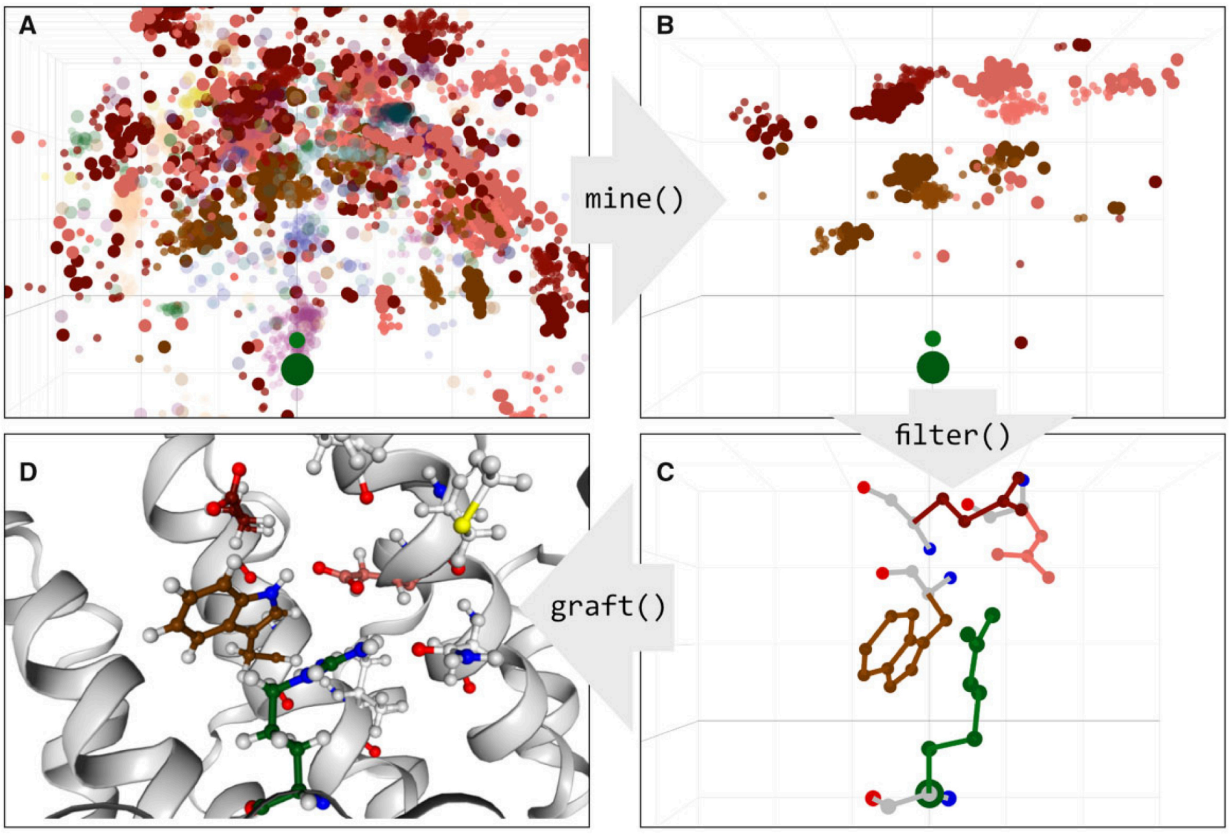



**Fig. 3.** Essential steps during design process with ATLIGATOR. (**A**) Three-dimensional visualization of *Atlas* datapoints for an Arg ligand residue. For reasons of visibility all datapoints with a positive z-value were discarded and only Asp (D), Glu (E) and Trp (W) binder residues are shown with full opacity. (**B**) All datapoints of the DEW *pocket* of an Arg ligand residue derived by mining (A). (**C**) Detailed view on a single instance of the DEW *pocket* including all side chain atoms. (**D**) DEW *pocket* grafted onto an armadillo repeat protein scaffold. The mutated residues are highlighted by coloring their carbon atoms according to the ATLIGATOR scheme. In A, B and C the axes are defined using the standard *x*, *y* and *z* coordinate definitions. The C$\alpha$ atom of the Arg ligand residue is highlighted by a bigger radius as the center of view. In A and B every binder residue as well as the ligand residue consists of a C$\alpha$ atom (stronger color, big radius) and a C$\beta$ atom (lighter color, small radius) colored according to the ATLIGATOR color scheme

single pockets and pocket grafting functions are available. The following paragraphs will guide through a typical workflow of the different tools.

**Structure selection.** The PDB is a rich resource for protein structures. Due to the large amount of data, but also due to biases in structures, scanning the whole PDB for ligand–binder interaction information is not recommended. Rather, the user can select structures from the PDB based on own rationale or on identifiers of the SCOPe database, creating sets of structures with shared structural or evolutionary background. Furthermore, we allow to use preprocessing and filtering structure files (see Section 2.1.1). Those sets of structures are called *structure collections*.

**Atlas visualization and usage.** *Atlases* can be obtained with the *atlas* generation algorithm (see Section 2.1.2) from *structure collections*. *Atlases* do not only serve as input for further analysis and design but visualizing them directly also provides insights into the collected data. Of particular interest are the 20 different *atlas maps* encoded in every *atlas*, which show frequent interactions for given ligand residue types. ATLIGATOR offers a three-dimensional visualization of single ligand amino acid types against one or all other binder amino acid types, corresponding to *atlas maps* or *pages*. These plots contain C$\alpha$ and C$\beta$ atoms of the centered ligand residue as well as C$\alpha$ and C$\beta$ atoms of the binder residues of each included datapoint. Thus, information about the relative position as well as orientation of both interaction partners is provided. Furthermore, it provides statistical insights into the composition of the *atlas* in terms of pair-wise interactions such as frequency of detected interaction pairs.

**Pocket visualization and general usage.** *Pockets* can be mined directly from *atlases*. ATLIGATOR can visualize and export into pdb format both superimposed and representative *pockets* (see Section 2.1.4). To present a more detailed point of view *pockets* can also be plotted as a collection of all included datapoints, representing the *pocket atlas* as a filtered instance of the corresponding *atlas* page (see Fig. 3A and B). Also, single *pocket* instances can be visualized, they contain exactly one ligand rotamer as well as all binder residue rotamers interacting with this exact ligand in the source structure as a part of this *pocket*. Thus, only those residues included in the *pocket* itemset that were not filtered during *pocket* generation will be present (see Fig. 3C). *Pockets* constitute a useful information *per se*, but they are also utilized in an automated grafting algorithm.

**Pocket grafting and quick graft.** Gathered insights and ideas from *atlas* and *pockets* can be applied to a protein of interest to craft designs with new binding features. Pocket grafting and the quick graft protocol can help fulfill this task. By supplying a structure of the protein–protein or protein–peptide complex of interest as a scaffold and selecting a previously mined collection of pockets such a task can be started. After defining mutable groups of peptide or protein ligand and binder within the scaffold this can be fed into a new design. Here, *pockets* of the assigned *pocket* collection can be selected and grafted automatically onto the binder protein (see Sections 2.1.5 and 2.1.6). If *pockets* are chosen for neighboring ligand residues and the same binder residue is mutated multiple times conflicts may occur. Such conflicts are internally solved based on cumulative similarity scoring (see Sections 2.1.5 and 2.1.6) and provide the optimal grafting solutions. Nevertheless, the mutations are based on natural pockets in the input structures and the side-chain rotamers will not fit perfectly into the new backbone. Thus, we recommend minimizing these rotamers subsequently, e.g. with the Rosetta fixbb side-chain packing protocol (Leaver-Fay *et al.*, 2011),

**Table 1.** Parameters for preprocessing, *atlas* generation and pocket mining

| Preprocessing parameter | Value |
|---|---|
| Distance | 8 A |
| Threshold binder | 40 aa |
| Min ligand | 3 aa |
| Hydrogen interactions | No |
| Atlas generation parameter | Value |
| Min length ligand | 3 aa |
| Max length ligand | 20 aa |
| Interaction radius (default) | 4.0 |
| Interaction radius (h-bond) | 6.0 |
| Interaction radius (aromatic) | 6.0 |
| Interaction radius (Ionic) | 8.0 |
| Skip backbone atoms | True |
| Allow intramolecular interactions | False |
| Pocket mining parameter | Value |
| Max p per Lig Res | 10 aa |
| Minimum pocket size | 3 aa |
| Confidence threshold | 0.02 |
| Support threshold | 0.01 |
| Cardinality base | 1.21 |
| Distance factor | 2.0 |
| Orientation factor | 1.0 |
| Secondary orientation factor | 1.0 |
| Variance threshold | 9999 |

to receive a self-consistent representation of all mutable residues. Designs can be written into a pdb file.

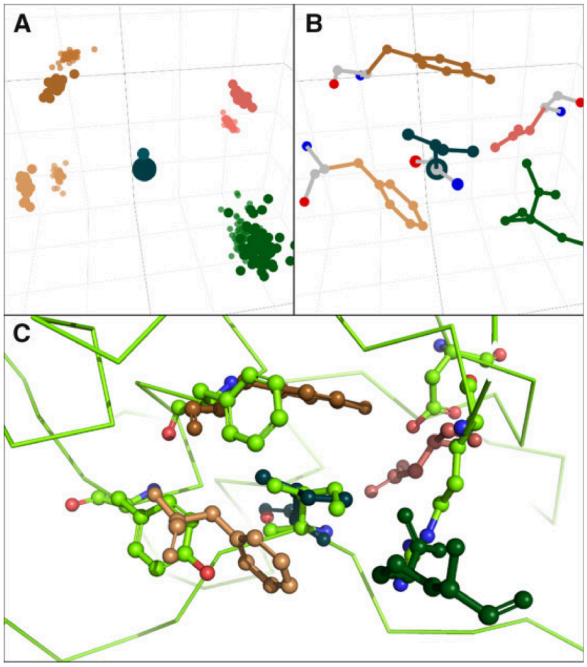



**Fig. 4.** Ile—RDFY *pocket* in comparison to a natural binding pocket found in an unrelated peptide-binding protein. (**A**) Cα and Cβ data points of Ile-binding pocket containing Arg, Asp, Phe and Tyr (see Fig. 3B). This *pocket* is based on an *atlas* of structures assigned to SCOPe classification a.118 (alpha–alpha superhelix). (**B**) Single RDFY *pocket* with complete side-chain configuration, originating from pdb structure 2ein. (**C**) Overlay of *pocket* in B (same coloring as before) and an Ile—RDFY interaction (Cα trace and side chains) found in an ankyrin repeat protein (d.211.1.0)

## 3 Results and discussion

There has always been an interest in computational structural biology to describe and classify protein side-chain interactions. The introduction of such descriptions established by Singh & Thornton (Singh and Thornton, 1992) led to improved understanding, but so far, the utilization of this data did not combine individual exploration as well as higher order interactions with a focus on protein design approaches. To do so, we created ATLIGATOR to more automatically detect naturally occurring interaction patterns and feed them into the design of protein–protein and protein–peptide interactions.

Now that *atlases* can be generated and designs can be created based on this data, it is interesting to look at the main functions of ATLIGATOR in the context of a typical workflow e.g. when designing a peptide binding interface. As an example, we chose the redesign of the binding interface of a designed armadillo repeat protein (dArmRP) that binds a peptide with the sequence $[KR]_5$ (Hansen *et al.*, 2016). We will focus on the third arginine in the peptide. On the one hand we aim to improve binding to arginine by pocket grafting and on the other hand we would like to alter the binding preference to isoleucine.

For redesigning the dArmRP binding site we decided to use structures assigned to SCOPe identifier a.118 (alpha–alpha superhelix) as our data source based on their structural similarity to our target protein. We processed all corresponding structures (parameters shown in Table 1). Hereby, we selected 907 structure files from the Protein Data Bank, leading to 2584 processed substructure complexes.

The *atlas* was generated from all structures obtained in the last step using parameters shown in Table 1. The *atlas* includes 20 pages containing 400 maps (see Fig. 2) in total—respective to every combination of canonical amino acids—comprising 43 752 datapoints. Of these, 4869 datapoints contain Arg as ligand residue (Fig. 3A), with the most frequent interaction partners of Arg being Asp (1055), Glu (998) and Thr (482). To get a better understanding of these interactions we analyzed frequent groups of interactions by *pocket* mining using the parameters shown in Table 1. One prevalent motif is the DEW *pocket*, which contains the residues Asp, Glu and Trp (see Fig. 3B). An example single *pocket* instance is shown in Figure 3C. These *pockets* are then used for grafting onto the scaffold of choice as shown exemplarily in Figure 3D for a DEW *pocket* grafted on the dArmRP scaffold. Such designs can now serve as starting points for further calculations or experimental testing.

For our second objective of altering the binding preference to isoleucine, we used the same atlas and pockets as above. Here, 1447 datapoints contained Ile as the ligand residue. The most frequent interaction partners of Ile were Tyr (157), Asp (150) and Met (148). For the transfer of an Ile binding pocket, we want to highlight an Ile—RDFY *pocket*, which was found in 8% of all Ile ligand residues. The RDFY interaction groups that were extracted from proteins that contain the alpha–alpha superhelix fold (a.118) are shown in Figure 4A. The single pocket instance shown in Figure 4B visualizes the interactions of isoleucine with the members of this pocket. Apart from using this pocket for grafting, it can also be used to search for similar binding pockets present in different folds. In fact, when we did this, we found a motif with remarkable similarity in the ankyrin repeat cluster domain 4 of human Tankyrase 2 (Guettler *et al.*, 2011), which is unrelated to our input *structure collection*. Interestingly, this motif is interacting with an Ile side chain of a bound peptide in a similar way (see Fig. 4C), supporting the idea that general descriptions of binding pockets can exist in different folds. This encourages potential transferability of pockets from one protein to another.

Surprisingly, the individual *pocket* instance in Figure 4B does not originate from the alpha–alpha superhelix (a.118) domain, but other parts within the larger multidomain protein. In fact, this binding pocket is formed by three subsections of two polypeptide chains, classified as f.17.2.1, b.6.1.2 and f.23.3.1. This is due to the fact that the original polyprotein complex contains just one a.118 subunit and no additional filtering was applied to input structures for

*atlas* generation. Even though the *pocket*'s origin is not the same fold as our scaffold protein, this is a very strong hint that interaction motifs found with ATLIGATOR can be generalized to other folds—even if more than one chain is forming such a binding pocket. Thus, analyzing *atlases* or *pockets* from different origins will help understand relationships of yet uncovered binding motifs.

In fact, ATLIGATOR is a versatile, data-driven methodology to analyze protein–protein and protein–peptide interactions in a variety of protein folds. In contrast to other tools, it focusses on local interactions, basically focusing the problem onto the side-chain level while incorporating higher-order interactions and intuitive design options. Moreover, it opens the opportunity to compare binding motifs from different sources to answer questions about generalizability of such motifs. Hence, fold-specific motifs can be detected and compared. ATLIGATOR also features statistical tools which can be utilized for analyzing interactions within the context of an *atlas*, *atlas map*, *atlas page* or *pockets*.

Despite these possible applications of ATLIGATOR, the main focus is to analyze the interaction in *atlas* and *pockets* for further use in a specific design task. To this end, it includes multiple ways to visualize and use data stored in the *atlas* and *pockets* and provides pocket grafting and quick graft options enabling a unique use of the interactions leveraged from the input structures.

## Acknowledgements

The authors thank members of the Höcker lab and the PReART research team for discussions and feedback, in particular Florian Gisdon, Julian Beck, Merve Ayyildiz, Steffen Schmidt, Pascal Kröger, Noelia Ferruz, Dominik Lemm and Abhishek Anan Jalan.

## Funding

This work was supported by the European Research Council [H2020-FETopen-RIA grant 764434 'PRe-ART'].

*Conflict of Interest*: none declared.

## References

Agrawal,R. *et al.* (1993) Mining association rules between sets of items in large databases. *SIGMOD Rec.*, **22**, 207–216.

Berka,K. *et al.* (2009) Representative amino acid side chain interactions in proteins. A comparison of highly accurate correlated ab initio quantum chemical and empirical potential procedures. *J. Chem. Theory Comput.*, **5**, 982–992.

Berka,K. *et al.* (2010) Energy matrix of structurally important side-chain/side-chain interactions in proteins. *J. Chem. Theory Comput.*, **6**, 2191–2203.

Berman,H.M. *et al.* (2000) The protein data bank. *Nucleic Acids Res.*, **28**, 235–242.

Chen,J.-S. *et al.* (2004) An extended study of the K-means algorithm for data clustering and its applications. *J. Oper. Res. Soc.*, **55**, 976–987.

Fox,N.K. *et al.* (2014) SCOPe: structural classification of proteins—extended, integrating SCOP and ASTRAL data and classification of new structures. *Nucleic Acids Res.*, **42**, D304–D309.

Galgonek,J. *et al.* (2017) Amino acid interaction (INTAA) web server. *Nucleic Acids Res.*, **45**, W388–W392.

Gisdon,F.J. *et al.* (2022) Modular peptide binders-development of a predictive technology as alternative for reagent antibodies. *Biol. Chem.*, **403**, 535–543.

Guettler,S. *et al.* (2011) Structural basis and sequence rules for substrate recognition by Tankyrase explain the basis for Cherubism disease. *Cell*, **147**, 1340–1354.

Hansen,S. *et al.* (2016) Structure and energetic contributions of a designed modular peptide-binding protein with picomolar affinity. *J. Am. Chem. Soc.*, **138**, 3526–3532.

Hansen,S. *et al.* (2018) Curvature of designed armadillo repeat proteins allows modular peptide binding. *J. Struct. Biol.*, **201**, 108–117.

Holland,J. and Grigoryan,G. (2022) Structure-conditioned amino-acid couplings: how contact geometry affects pairwise sequence preferences. *Protein Sci.*, **31**, 900–917.

Jha,A.N. *et al.* (2010) Amino acid interaction preferences in proteins. *Protein Sci.*, **19**, 603–616.

Kuhn,H.W. (1955) The hungarian method for the assignment problem. *Naval Res. Logistics Q.*, **2**, 83–97.

Leaver-Fay,A. *et al.* (2011) Chapter nineteen—Rosetta3: an object-oriented software suite for the simulation and design of macromolecules. In: Johnson M.L. and Brand L. (eds.) *Methods in Enzymology*. Academic Press (Elsevier), Cambridge, Massachusetts, pp. 545–574.

Liu,S. *et al.* (2020) Neighborhood preference of amino acids in protein structures and its applications in protein structure assessment. *Sci. Rep.*, **10**, 4371.

MacKenzie,C.O. *et al.* (2016) Tertiary alphabet for the observable protein structural universe. *Proc. Natl. Acad. Sci. USA*, **113**, E7438–E7447.

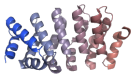
Polizzi,N.F. and Degrado,W.F. (2020) A defined structural unit enables de novo design of small-molecule-binding proteins. *Science*, **369**, 1227–1233.

Porter,C.T. *et al.* (2004) The Catalytic Site Atlas: a resource of catalytic sites and residues identified in enzymes using structural data. *Nucleic Acids Res.*, **32**, D129–33.

Singh,J. and Thornton,J.M. (1992) Atlas of protein side-chain interactions. IRL Press at Oxford University Press, Oxford.

Swanson,S. *et al.* (2022) Tertiary motifs as building blocks for the design of protein-binding peptides. *Protein Sci.*, **31**, e4322.

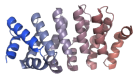
Zhou,J. *et al.* (2020) A general-purpose protein design framework based on mining sequence-structure relationships in known protein structures. *Proc. Natl. Acad. Sci. USA*, **117**, 1059–1068.



# 3. Atligator Web: A Graphical User Interface for Analysis and Design of Protein–Peptide Interactions

**Josef P. Kynast** and Birte Höcker

*BioDesign Research* **2023**; 5; 0011



BioDesign Research
A SCIENCE PARTNER JOURNAL

RESEARCH ARTICLE

# Atligator Web: A Graphical User Interface for Analysis and Design of Protein–Peptide Interactions

**Josef Paul Kynast and Birte Höcker***

Department of Biochemistry, University of Bayreuth, Bayreuth, Germany.

*Address correspondence to: birte.hoecker@uni-bayreuth.de

**Citation:** Kynast JP, Höcker B. Atligator Web: A Graphical User Interface for Analysis and Design of Protein–Peptide Interactions. *BioDesign Res.* 2023;5:Article 0011. https://doi.org/10.34133/bdr.0011

Submitted 23 December 2022
Accepted 14 April 2023
Published 4 May 2023



A key functionality of proteins is based on their ability to form interactions with other proteins or peptides. These interactions are neither easily described nor fully understood, which is why the design of specific protein–protein interaction interfaces remains a challenge for protein engineering. We recently developed the software ATLIGATOR to extract common interaction patterns between different types of amino acids and store them in a database. The tool enables the user to better understand frequent interaction patterns and find groups of interactions. Furthermore, frequent motifs can be directly transferred from the database to a user-defined scaffold as a starting point for the engineering of new binding capabilities. Since three-dimensional visualization is a crucial part of ATLIGATOR, we created ATLIGATOR web—a web server offering an intuitive graphical user interface (GUI) available at https://atligator.uni-bayreuth.de. This new interface empowers users to apply ATLIGATOR by providing easy access with having all parts directly connected. Moreover, we extended the web by a design functionality so that, overall, ATLIGATOR web facilitates the use of ATLIGATOR with a more intuitive UI and advanced design options.

## Introduction

The specific recognition of binding partners in protein–protein or protein–peptide interactions is established by mutual interactions of amino acid residues. While each residue's contribution shapes the binding affinity and specificity to agonist or antagonist binders, certain residue–residue interactions are more crucial than others. With this in mind, it is likely that in the context of optimized binding partners, pairwise interactions with a higher influence will be found more often than others. Following this hypothesis, Singh and Thornton [1] already identified some frequent interaction residue pairs and defined their spatial arrangement. Amino acid pairs have also been investigated energetically [2–4] or in a generalized form with focus on functional groups [5]. Within densely packed interaction surfaces, however, a pairwise residue–residue interaction is affected by its context. This structural context and identities of neighboring residues were successfully incorporated in recent analyses [6–9].

Another idea is to investigate groups of residues that act as a binding partner for single residues. For this approach, we created the software package ATLIGATOR [10]. It extracts pairwise interactions from protein structures to find patterns in frequent residue–residue pairs and stores these in a data structure called *atlas*. By varying the input structures, the structural and evolutionary context of underlying data can be modulated and compared. Moreover, by mining the *atlas* for frequent interaction groups, interaction motifs can be extracted, visualized, and analyzed. Additionally, ATLIGATOR enables direct grafting of frequent interaction motifs (*pockets*) onto a user-defined scaffold protein. Since working with three-dimensional interaction motifs or designing binding pockets is a highly visually demanding task, it also allows to plot *atlas* or *pocket* data.

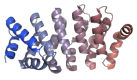
To enable everyone to try and use ATLIGATOR instantly without installation, we developed a web interface. This graphical interface connects all data structures visually and even extends the design options of ATLIGATOR. It is freely accessible at https://atligator.uni-bayreuth.de and creates an intuitive starting point for working with the software.

## Results and Discussion

ATLIGATOR web expands the functionality of the ATLIGATOR python package by a user-friendly interface, thereby providing easy access and supporting users to utilize its features instantly. For the analysis of pairwise interactions, we provide a list of pre-generated *atlases* and *pocket collections* that can be explored in detail. All data can be visualized similarly to the python API. However, a unique aspect of the web interface is the connection between the different sections, i.e., an *atlas* to its input structures, *pockets* to its underlying *atlas*, etc. This connection is accomplished mainly by hyperlinks to superior data structures. Additionally, the representation of the plotted data points clearly highlights these connections in several ways: The original structural environment of data points is directly displayed by clicking on data points and, in this visualization, all atoms are labeled with their origin.

While the focus of ATLIGATOR web is the representation of data structures in a more intuitive way, we additionally provide the opportunity to design protein interaction sites. This is





useful, because ATLIGATOR data can comprise potential starting points for shaping binding surfaces. Hence, by supplying a protein–protein or protein–peptide complex and defining ligand and binder chains, *pockets* can be grafted directly onto the binder chain. Additionally, manual mutations can be applied to extend or fine-tune grafts with new mutations.

## Analysis of binding interfaces

Below, we will elaborate the web interface and each section in more detail to showcase potential usage and highlight special features. To do so, we will guide through the data structures of ATLIGATOR web with the same example as examined before [10]. Namely, we will look for interaction motifs to use in a designed armadillo repeat protein to change specificity from an arginine to a leucine as a residue of the native peptide binding partner [11,12].

### *The interface*

The ATLIGATOR web interface is built around five main sections, namely, *Structures*, *Atlases*, *Pockets*, *Scaffolds*, and *Designs* (see Fig. 1). The landing page gives access to all sections as well as frequently asked questions (FAQ) and an example page. The footer of each web page includes the current color scheme and a switch to activate the tutorial mode. If activated, info boxes that explain function and handling of applications are shown.

We could start our redesign example in the section *Atlases*, but to understand where the *atlas* data originate, we will visit the section *Structures* first.

### *Structure collections*

Interaction data stored in *atlases* or *pockets* originate from protein structures. In contrast to the ATLIGATOR python package, ATLIGATOR web integrates these structures as an explorable section. Within this section, input structures are grouped in *collections* that typically consist of all structures assigned to a SCOPe database identifier. Our example protein scaffold [Protein Data Bank (PDB) identifier 5AEI] is not classified in the SCOPe database (version 2.08) because of its synthetic nature. However, since it originates from natural armadillo repeat proteins (a.118.1.1 in SCOPe), we can utilize the structures attributed to a.118 (α-α-superhelix). The *structure collection* comprises 2,584 structures, each hosting a binder chain and all potentially interacting ligand chains. Each three-dimensional structure can be observed and downloaded individually.

### *Atlases*

*Atlases* are based on pairwise interactions derived from single *structure collections*. Thus, *atlases* in the web interface contain a link to the corresponding *structure collection*. As a landing

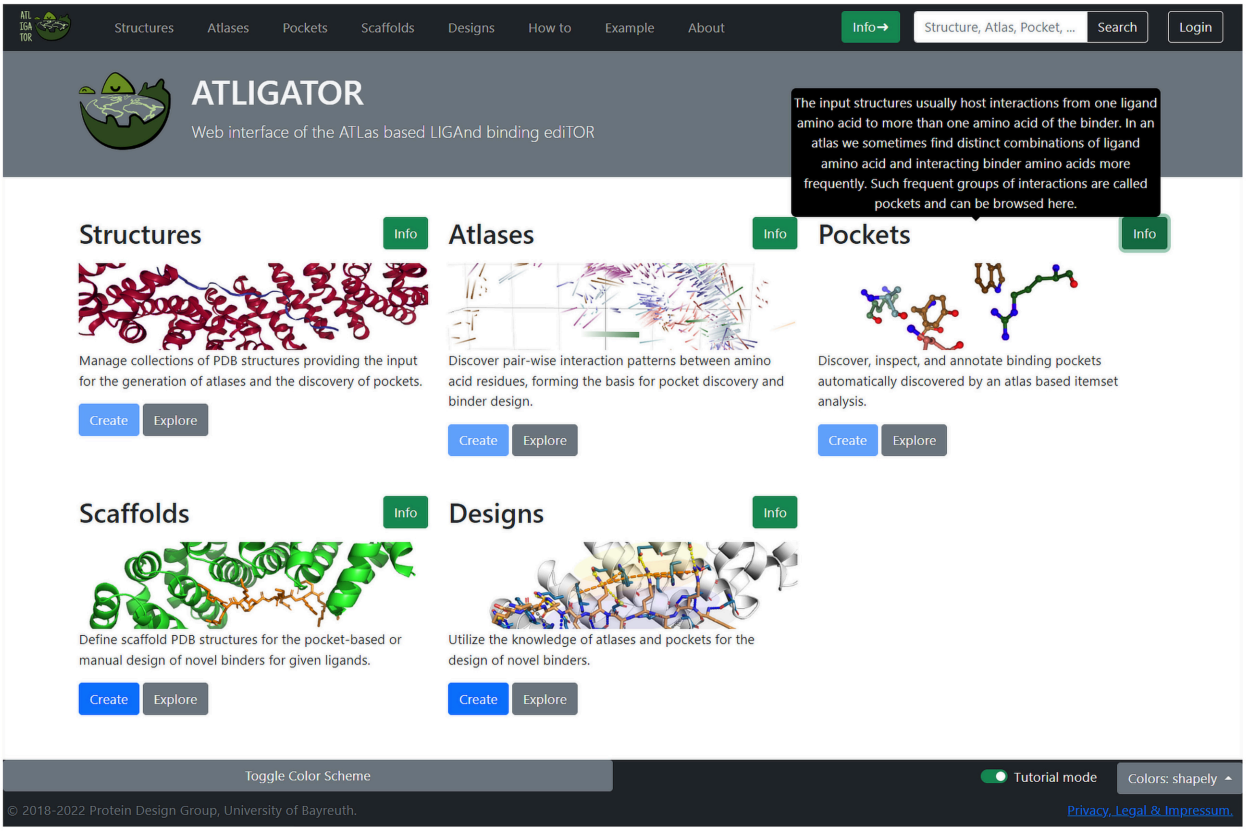



**Fig. 1.** Landing page of ATLIGATOR web. It comprises links to the five main sections, namely, *Structures*, *Atlases*, *Pockets*, *Scaffolds*, and *Designs*. The header includes a navigation bar as well as a search box and a context box for user accounts. The footer features a switch for tutorial mode that enables one to switch on info boxes with explanations to guide new users as shown. The common color scheme can be selected and an overview of the color scheme can also be activated in the footer.

page, *atlas* statistics visualize the frequency of the data points. The a.118-based *atlas* contains 43,645 data points, i.e., pairwise interactions. Our design target amino acid leucine (Leu) comprises 3,577 interactions—with other Leu (705), histidine (His; 517), and aspartate (Asp; 325) being the most frequent binding partners (Fig. 2A). The *atlas* can be browsed for *atlas* maps and *atlas* pages where the number of data points is described and the data points can be viewed in three-dimensional plots. The plot of leucine's *atlas map* reveals point clouds with high density of those Leu/His/Asp interactions that can be revisited, e.g., for the leucine-to-leucine interactions in the corresponding *atlas* page (Fig. 2B). Clicking on individual data points exhibits the entire side chains of both interaction partners of the underlying pair (Fig. 2B—full-atom view). Furthermore, pairwise distances of all displayed atoms can be measured. The positions in combination with their Cα to Cβ orientation can already provide first ideas for the design of such interactions.

### Pockets

Frequent groups of interactions found in an *atlas* that result from mining all atlas data points for recurring motifs are defined as *pockets* [10]. On the web interface, *pockets* from the same

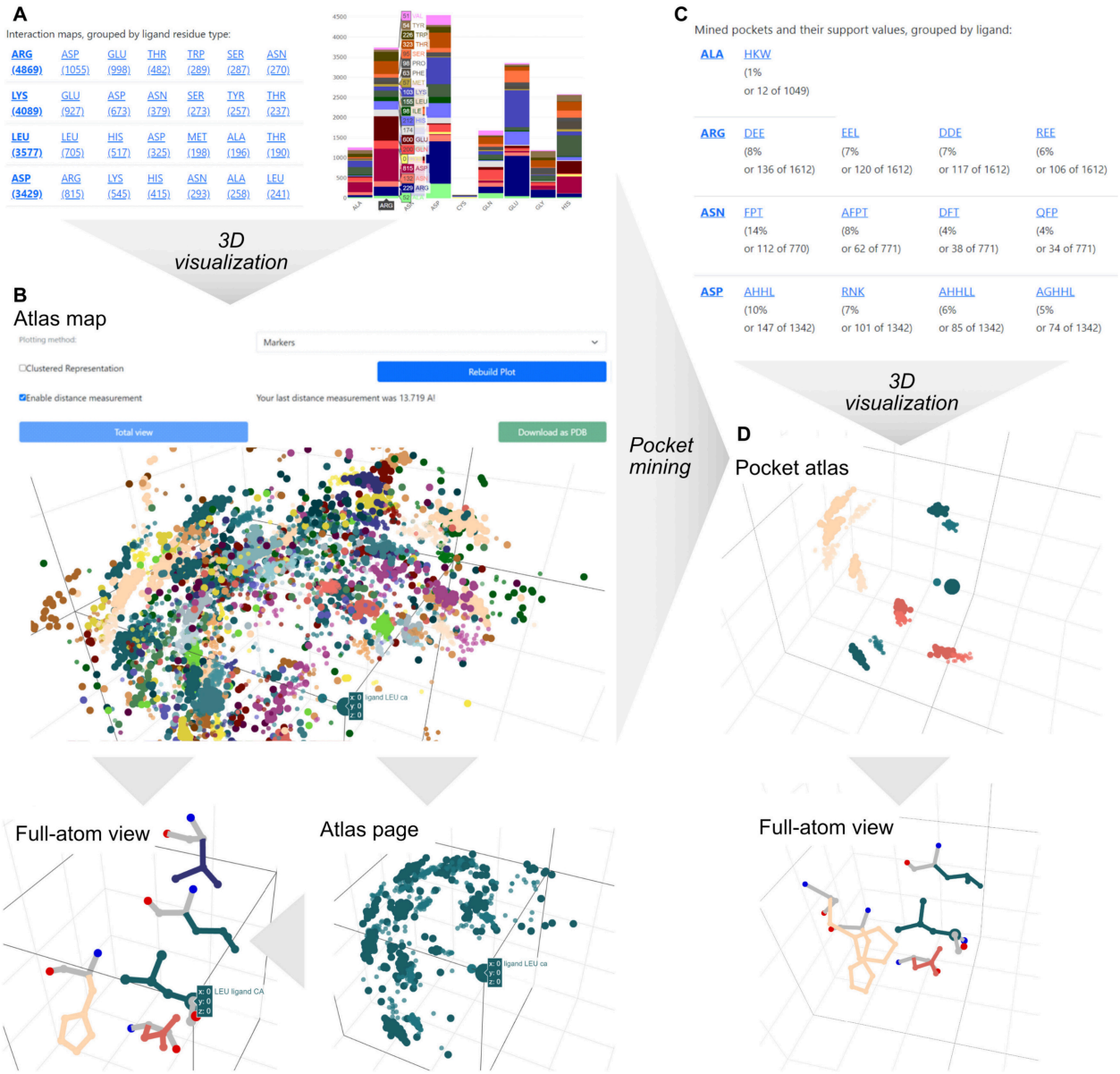



**Fig. 2.** Visualization of *atlas* and *pockets*. Starting at an *atlas*, the ATLIGATOR web gives an overview of the *atlas'* content (A), and further offers the option to plot and visualize its content in three dimensions (B). *Atlas maps* and *pages* can be inspected as Cα–Cβ plots as well as in a full-atom view while concentrating on just one ligand origin. *Pockets* derived from mining the *atlas* are grouped in *pocket collections*. In addition to an overview of the contained pockets (C), all pockets can be plotted in Cα–Cβ plots as well as in full-atom view (D).

*atlas* are grouped in *pocket collections*. After opening the a.118 *pocket collection*, statistics are displayed for all ligand amino acid types (Fig. 2C) of which a DHL *pocket* was the most abundant Leu pocket. This *pocket*, being composed of the three amino acids Asp (D), His (H), and Leu (L), was observed in interaction with 10% of all Leu ligand residues of the initial *atlas*.

The *pocket* and its data points are visualized in different representations: First, the most representative pocket, which corresponds to the pocket instance with the lowest deviation from the item set clusters' centroids [10], is visualized in a full-atom stick representation. Second, the clustered data points (centroids of clusters) and all data points of the pocket *atlas* are initially shown in *atlas* representation with ligand and binder Cα and Cβ atoms as bubbles (Fig. 2D). The pocket *atlas* contains all data points and represents a filtered instance of the original *atlas*. It exposes three spatially restricted clusters for His Cα positions as well as two clusters for each Leu and Asp (see Fig. 2D) and provides two additional types of representations. On the one hand, all ligand residue atoms can be displayed separately to see the distribution of ligand side chain conformations. On the other hand, after clicking on a ligand or binder atom, the corresponding single pocket instance is plotted with all residues in full-atom representation (Fig. 2D). The full-atom representation of our example reveals two interaction motifs: First, Leu being trapped between 2 His, 1 Asp, and 1 Leu, where the Leu–Leu interaction is dominated by side chain–backbone interactions. Second, Leu is attacked by Leu and His roughly at a 90° angle with the Asp facing its carboxy group to the backbone of the target Leu. These single *pocket* instances provide promising starting points for our redesign and can be downloaded as .pdb files or used directly for grafting onto a user-defined scaffold.

### Redesign of binding interfaces

While the analysis of frequent motifs helps to extract ideas for the redesign of interaction interfaces, we additionally included design tools to directly use this information on specific binding interfaces. Those tools will be briefly introduced below.

#### *Scaffolds*

Proteins on which ATLIGATOR *pockets* shall be grafted are called scaffolds. Users can upload their own protein structures, hosting two or more polypeptide chains. One of these chains has to be defined as the ligand chain, comprising the residue on which basis the pocket should be selected. The chain has to be defined as the binder chain that is mutated eventually. Afterwards, mutable positions must be selected to finalize the scaffold preparation.

#### *Designs*

A design is defined by the scaffold and the residue types that will be used in mutable ligand positions. Each design can harbor multiple design tasks where the binder can be altered by applying different mutations. After selecting a pocket collection for the design task, the user can start designing. Within the tool *manual graft*, two ways of mutating the scaffold are implemented. The first method is based on choosing a pocket from the pocket collection for each ligand residue that will be grafted automatically onto the scaffold (Fig. 3). The grafting applies mutations based on the selected pocket onto the best fitting mutable binder positions as in ATLIGATOR [10]. However, the selection of mutated positions heavily depends on the underlying pocket data, which might not fit flawlessly with the scaffold if the geometry of the interaction is not perfectly alignable. Moreover, the best graft is not guaranteed to fit better than the next best grafts due to the lack of scoring for resulting shape complementarity or actual interactions. Thus, as the second mutation method, we implemented an option to choose manual mutations. Mutations can be added independently for all mutable binder residues and override mutations from pocket grafting.

Since grafted mutations incorporate the side chain conformer of the pocket data point and the manual mutations do not include a side chain conformer at all, the rotamers do not resemble realistic conformations. Thus, a refinement of the designed interface might be necessary to judge the design's quality. Manual graft offers to repack the mutable interface using Rosetta's fixbb protocol to generate more realistic side chain conformations [13].

## Conclusion

The development of ATLIGATOR opens up the possibility to collect and learn from protein–protein interactions in a streamlined and automated fashion. ATLIGATOR web empowers users to leverage its functionality in a user-friendly and easily accessible environment. Additionally, the web interface extends ATLIGATOR tools with novel functions like connecting *atlas* data points to the underlying structural data and an advanced design tool for pocket grafting and rational design. In summary, the easy access via this graphical interface enables a broader user base to apply ATLIGATOR and helps to understand its principles. Thus, ATLIGATOR web might also be a starting point and encourage users to directly apply the python package for extended analysis and design.

## Implementation

### Code implementation

The web interface for ATLIGATOR is implemented in the Python 3 web framework Django [14] in combination with uWSGI as a reverse proxy and javascript for client code handling. We use jQuery and bootstrap 5 as css and javascript frameworks for web design and celery in combination with rabbitMQ for asynchronous task handling.

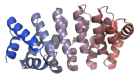
### Visualization

The JavaScript library plotly.js is used for responsive plotting of statistics data. Three-dimensional plotting is performed by two JavaScript libraries. Plotly.js is handling *atlas* and *pockets* as well as full atom plotting of these [15]. NGL viewer is implemented for visualization of proteins and protein complexes [16].

### Availability

The sections *structures*, *atlases*, and *pockets* are openly accessible for discovering the data in the pre-built data structures. The protein structures were collected by searching the SCOPe database [17] with the corresponding queries. The starting *structure collections* were chosen based on known peptide binding capabilities, a diverse classification in SCOPe and multitude of available structures in the PDB. The protein structure files were preprocessed to generate files with one protein chain and





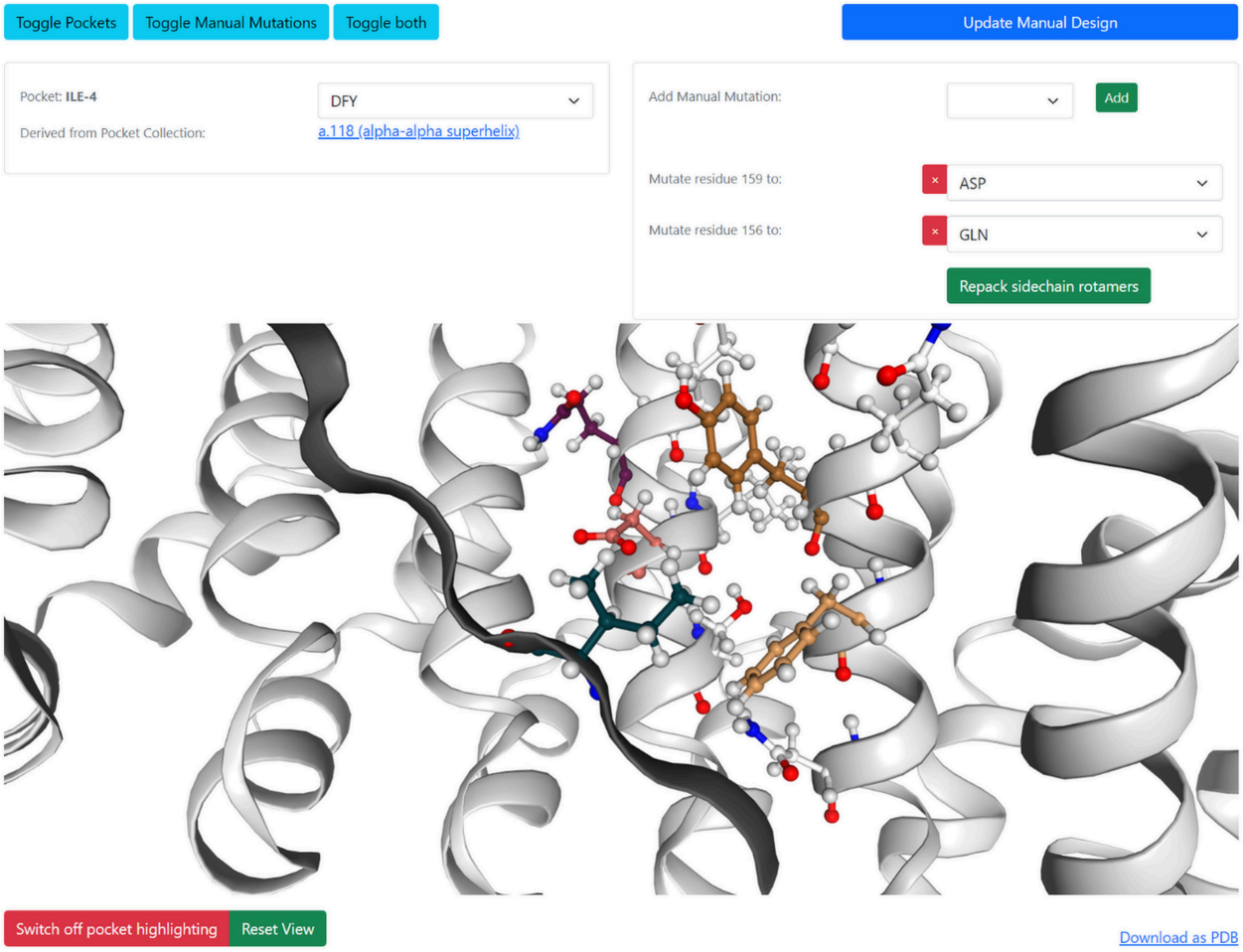



**Fig. 3.** Design interface including grafting and manual mutations. Two tiles provide dropdown menus to design the scaffold protein by grafting a *pocket* of choice or applying manual mutations. Changes are applied after clicking the "Update Manual Design" button or the "Repack sidechain rotamers" button, while the latter also applies the Rosetta repacking protocol to the mutable side chain rotamers. Changes can be observed directly in the preview tile with an option to colorfully highlight the mutated positions or by downloading the .pdb file of the designed protein.

cropped ligand chains. The resulting structure files were used for *atlas* generation and subsequent *pocket* mining. All processing parameters were used as previously described [10]. The design sections including *scaffolds* and *designs* can be fully explored via a user account that only requires a username, email address, and password. We also offer a scaffold and design example that correspond to the ones discussed above.

### Documentation

The web server includes a tutorial mode that offers comments and explanations for the different features and sections directly on the web pages. It also links to videos for showcasing sections of ATLIGATOR web to enable an easier start.

## Acknowledgments

We thank Felix Schwägerl for his help with the web interface source code, Steffen Schmidt and Noelia Ferruz for support with the computational infrastructure, and all other members of the Höcker lab and the PReART research team for discussions and feedback on Atligator web. **Funding**: This work was supported by the European Research Council H2020-FETopen-RIA grant 764434 "PRe-ART" and the EIC Transition grant 10105802 "PRe-ART-2T". **Author contributions**: J.P.K.: Conceptualization, methodology and implementation, and writing. B.H.: Conceptualization, funding acquisition, and writing. **Competing interests**: The authors declare that they have no competing interests.

## Data Availability

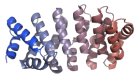
ATLIGATOR web is available at https://atligator.uni-bayreuth.de. It is based on the ATLIGATOR code, which is available at https://github.com/Hoecker-Lab/atligator.

## References

1. Singh J, Thornton JM. *Atlas of protein side-chain interactions*. Oxford (UK): IRL Press at Oxford University Press; 1992.

2. Berka K, Laskowski RA, Hobza P, Vondrášek J. Energy matrix of structurally important side-chain/side-chain interactions in proteins. *J Chem Theor Comput*. 2010;6(7):2191–2203.
3. Berka K, Laskowski R, Riley KE, Hobza P, Vondrášek J. Representative amino acid side chain interactions in proteins. A comparison of highly accurate correlated ab initio quantum chemical and empirical potential procedures. *J Chem Theor Comput*. 2009;5(4):982–992.
4. Galgonek J, Vymětal J, Jakubec D, Vondrášek J. Amino acid interaction (INTAA) web server. *Nucleic Acids Res*. 2017;45(W1):W388–W392.
5. Polizzi NF, Degrado WF. A defined structural unit enables de novo design of small-molecule-binding proteins. *Science*. 2020;369(6508):1227–1233.
6. Liu S, Xiang X, Gao X, Liu H. Neighborhood preference of amino acids in protein structures and its applications in protein structure assessment. *Sci Rep*. 2020;10(1): Article 4371.
7. MacKenzie CO, Zhou J, Grigoryan G. Tertiary alphabet for the observable protein structural universe. *Proc Natl Acad Sci USA*. 2016;113(47):E7438–E7447.
8. Swanson S, Sivaraman V, Grigoryan G, Keating AE. Tertiary motifs as building blocks for the design of protein-binding peptides. *Protein Sci*. 2022;31(6):Article e4322.
9. Zhou J, Panaitiu AE, Grigoryan G. A general-purpose protein design framework based on mining sequence–structure relationships in known protein structures. *Proc Natl Acad Sci USA*. 2019;117(2):1059–1068.
10. Kynast JP, Schwägerl F, Höcker B. ATLIGATOR: Editing protein interactions with an atlas-based approach. *Bioinformatics*. 2022;38(23):5199–5205.
11. Gisdon FJ, Kynast JP, Ayyildiz M, Hine AV, Plückthun A, Höcker B. Modular peptide binders—Development of a predictive technology as alternative for reagent antibodies. *Biol Chem*. 2010;403(5–6):535–543.
12. Hansen S, Tremmel D, Madhurantakam C, Reichen C, Mittl PRE, Plückthun A. Structure and energetic contributions of a designed modular peptide-binding protein with picomolar affinity. *J Am Chem Soc*. 2016;138(10):3526–3532.
13. Kuhlman B, Baker D. Native protein sequences are close to optimal for their structures. *Proc Natl Acad Sci*. 2000;97(19): 10383–10388.
14. Django. Django (Version 4.0). Lawrence (KS): Django Software Foundation; 2022. https://www.djangoproject.com/.
15. Plotly Technologies Inc. Collaborative data science. Montreal (Canada): Plotly Technologies Inc.; 2015.
16. Rose AS, Hildebrand PW. NGL viewer: A web application for molecular visualization. *Nucleic Acids Res*. 2015;43(W1):W576–W579.
17. Fox NK, Brenner SE, Chandonia J-M. SCOPe: Structural classification of proteins—Extended, integrating SCOP and ASTRAL data and classification of new structures. *Nucleic Acids Res*. 2014;42(Database issue):D304–D309.





# 4. PocketOptimizer 2.0: A modular framework for computer-aided ligand-binding design

Jakob Noske, **Josef P. Kynast**, Dominik Lemm, Steffen Schmidt and Birte Höcker

*Protein Science* **2023** Jan; 32(1): e4516

Received: 31 July 2022 | Revised: 12 November 2022 | Accepted: 14 November 2022

DOI: 10.1002/pro.4516

TOOLS FOR PROTEIN SCIENCE

# PocketOptimizer 2.0: A modular framework for computer-aided ligand-binding design

**Jakob Noske**[1] | **Josef Paul Kynast**[1] | **Dominik Lemm**[1] | **Steffen Schmidt**[2] | **Birte Höcker**[1]

[1]Department of Biochemistry, University of Bayreuth, Bayreuth, Germany

[2]Computational Biochemistry, University of Bayreuth, Bayreuth, Germany

**Correspondence**
Birte Höcker, Department of Biochemistry, University of Bayreuth, Universitätsstr. 30, 95447 Bayreuth, Germany.
Email: birte.hoecker@uni-bayreuth.de

**Present address**
Dominik Lemm, Department of Physics, University of Vienna, Vienna, Austria.

**Funding information**
Deutsche Forschungsgemeinschaft, Grant/Award Number: Grant HO 4022/2-3; European Research Council H2020-FETopen RIA, Grant/Award Number: 764434

**Review Editor:** Nir Ben-Tal

**Abstract**

The ability to design customized proteins to perform specific tasks is of great interest. We are particularly interested in the design of sensitive and specific small molecule ligand-binding proteins for biotechnological or biomedical applications. Computational methods can narrow down the immense combinatorial space to find the best solution and thus provide starting points for experimental procedures. However, success rates strongly depend on accurate modeling and energetic evaluation. Not only intra- but also intermolecular interactions have to be considered. To address this problem, we developed PocketOptimizer, a modular computational protein design pipeline, that predicts mutations in the binding pockets of proteins to increase affinity for a specific ligand. Its modularity enables users to compare different combinations of force fields, rotamer libraries, and scoring functions. Here, we present a much-improved version––PocketOptimizer 2.0. We implemented a cleaner user interface, an extended architecture with more supported tools, such as force fields and scoring functions, a backbone-dependent rotamer library, as well as different improvements in the underlying algorithms. Version 2.0 was tested against a benchmark of design cases and assessed in comparison to the first version. Our results show how newly implemented features such as the new rotamer library can lead to improved prediction accuracy. Therefore, we believe that PocketOptimizer 2.0, with its many new and improved functionalities, provides a robust and versatile environment for the design of small molecule-binding pockets in proteins. It is widely applicable and extendible due to its modular framework. PocketOptimizer 2.0 can be downloaded at https://github.com/Hoecker-Lab/pocketoptimizer.

## 1 | INTRODUCTION

Ligand binding is essential in most biological processes, for example, enzyme catalysis, immune recognition, regulation of metabolism, cellular signal transduction, or control of gene expression. The ability to design such interactions will help us address many of society's current challenges. Computational tools for

the design of small molecule-binding pockets in proteins are of great interest for the design of tailored enzymes that can catalyze reactions for which no natural catalyst exists[1–4] or for the development of specific biosensors that can detect small molecules in vitro and in vivo.[5,6]

From an energetic point of view, the recognition of small molecules by proteins relies on the cooperative formation of a set of weak, non-bonded interactions, primarily van der Waals (vdW), and electrostatic attraction, as well as the formation of hydrogen bonds. These interactions can be estimated based on a variety of receptor ligand scoring functions[7–9] and can be used to identify specific mutations that lead to increased binding affinity of a protein to its ligand. Additionally, solvent effects have been discussed to play a major role but are not always included in the scoring functions.[10–12] Apart from protein–ligand interactions, internal protein interactions must also be considered upon mutation to minimize destabilizing effects on the protein structure. To this end, we developed a modular pipeline called PocketOptimizer that accounts for both packing energies and binding-related energies and that can include different scoring functions to allow adaptation to specific design problems.[13] In this design pipeline, we address side chain flexibility via rotamer libraries and ligand flexibility by using stochastic or systematic search algorithms.[13,14] In addition, discrepancies between designs and experimental results can be more easily determined because all sampled conformations, together with the computed interaction energies, are written to user-inspectable files. Finally, a deterministic solving procedure is applied to extract the optimum from the sampled search space.[15]

Due to the significance of protein–ligand binding, several tools have been developed to computationally score and (re)design protein binders. Commonly, these techniques attempt to approximate binding free energy changes and binding constants based on ensembles of bound and unbound states.[16–19] While most programs use only one way of designing and scoring, the PocketOptimizer framework, which only evaluates the bound state, is set up to use different modules. This way, different approaches or scoring functions can be compared, and a tailored method can be created for the design problem at hand. However, PocketOptimizer became outdated, making the addition of new functions difficult. Here, we present a new version, PocketOptimizer 2.0. Its new user interface is much more accessible, and the modular architecture has been improved and extended to provide more options for modeling and scoring within the computer-aided design process.

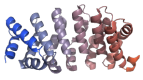
## 2 | RESULTS

### 2.1 | Design pipeline

The design pipeline can be divided into four main steps: structure preparation, flexibility sampling, energy calculations, and computation of design solutions (see Figure 1). As input for the pipeline, the structures of a protein and a ligand are needed. The ligand has to be placed manually inside the binding pocket since its initial position influences the design results and can therefore hardly be automated. Before the actual design process can start, the protein undergoes a cleaning procedure to remove unwanted ions, water molecules, small molecules, and protein chains. Next, all amino acid side chains are protonated according to a pH value defined by the user. Afterwards, an initial minimization step is performed to resolve potential clashes that may occur in the process of model building. During minimization, backbone atoms are typically constrained to maintain the backbone conformation. Once scaffold preparation is complete, the binding pocket can be defined by selecting flexible residues at certain design positions. Thus, all non-selected residues are fixed along with the backbone. Similar to the protein, the ligand is protonated. This is then followed by a parameterization step in which atom types, force field parameters, and partial charges are assigned for both structures.

In the second step, the flexibility sampling step of the pipeline (Figure 1), rotamers for residues at all defined design positions and ligand poses can be sampled. PocketOptimizer 2.0 includes two rotamer libraries: a smaller, backbone-independent rotamer library compiled from high-resolution protein crystal structures named CMLib[20] and a larger, backbone-dependent rotamer library known as the Dunbrack rotamer library.[21] For ligand pose sampling, ligand conformations can be generated using different algorithms.[13,14] All generated conformers are then systematically translated and rotated along a user-defined grid to create an ensemble of poses within the binding pocket. To reduce computational overhead, rotamers and poses are subsequently pruned from the search tree.

Interaction energies for rotamers and ligand poses are calculated in the third step of the pipeline. For this purpose, the binding pocket needs to be decomposed into self- and pairwise interaction energies. Whereas self-interaction energies describe the interaction between either rotamers or ligand poses and the fixed scaffold, pairwise-interaction energies describe the interaction between rotamers or ligand poses and other rotamers. This decomposition of energies allows solving the design



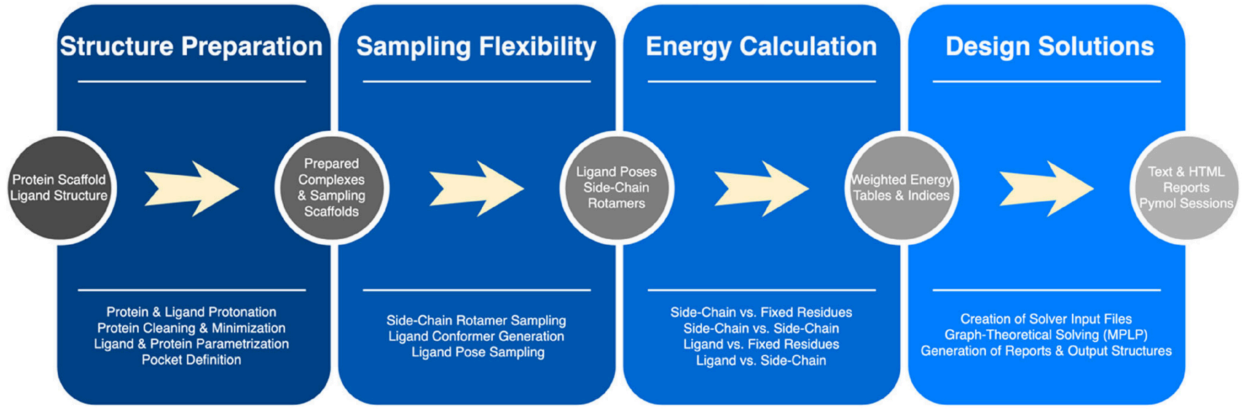



**FIGURE 1** The different steps of the PocketOptimizer pipeline. For each section, the required input, the included steps, and the obtained output are listed. The workflow starts with a protein and a ligand structure. These are processed in a preparation step (first box). To account for flexibility, rotamers and ligand poses are sampled (second box). Next, the interaction energies for each rotamer and each ligand pose against the fixed scaffold and against each other are calculated (third box). Finally, the best design solutions are identified using an integer linear programming solving algorithm (fourth box)

problem at a later stage. The computed energies can be further subdivided into those representing interactions within the protein or interactions between protein and ligand. While the so-called packing-related energies represent changes in protein stability, the binding-related energies represent changes in binding affinity and are therefore particularly important. Hence, they can be scaled according to the packing energies and calculated based on a variety of receptor-ligand scoring functions.[8,9,22]

In the last step of the pipeline, PocketOptimizer uses a solver algorithm based on integer linear programming (ILP) to identify the best design solutions.[15] The algorithm requires weighted energy tables and indices of all rotamers and ligand poses. Once executed, it can provide indices that minimize the total energy. The corresponding rotamers and ligand poses represent the global minimum energy conformation (GMEC) of our system, where the ligand binding energy can again be extracted. According to these rotamers and ligand poses, output files of the resulting energies and design structures can be generated. This includes the designed structures in PyMOL sessions[23] as well as the text and HTML files containing the generated energy tables.

## 2.2 | New features in PocketOptimizer 2.0

The original version of PocketOptimizer 1.0[20] was mainly a collection of binaries and Python scripts that interconnected the various parts of the design pipeline. It was then extended with a command-line interface to allow for easier interaction with the framework.[24] Nonetheless, source code and software dependencies remained unchanged. As, these are now a decade out of date, we fundamentally rewrote the software and implemented a range of new functionalities to extend it further (Table 1). This resulted in version 2.0 of PocketOptimizer, which will be presented in a comparative manner in the following section.

The first version of PocketOptimizer was written in Python 2.7, which lost maintenance support in the beginning of 2020. Since most Python libraries are no longer supporting Python 2.7, PocketOptimizer was rewritten in Python 3.9. Additionally, we implemented a Python application programming interface that allows not only to use specific functionalities of the design pipeline, but also permits a more user-friendly and flexible interaction with the framework. PocketOptimizer 2.0 now also offers multi-core processing, making it faster and scaling better on a larger number of CPUs. Moreover, progress bars have been added to monitor computation progress. Additionally, parts that have already been computed can be now reused when varying a design task.

Previously, in PocketOptimizer 1.0, the user had to prepare the input protein structures, often using external software such as Chimera.[25] In the new version of our software, HTMD's protein preparation pipeline *systemPrepare* has been implemented[26] for this. It also comes with the possibility to assign specific protonation states according to calculated empirical pKa values (PROPKA[27]) and user-defined pH values. After preparation, minimization is now also available using the molecular dynamics framework OpenMM,[28] which provides

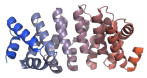


TABLE 1 Comparison of PocketOptimizer 1.0 and 2.0, listing the main differences between both versions

| | Version 1.0 | Version 2.0 |
|---|---|---|
| Language | Python 2.7/C++ | Python 3.9 |
| UI | CLI | API/CLI |
| Processing | Single core | Multi core |
| Scaffold preparation | External (Chimera) | Internal (*systemPrepare*) |
| Ligand preparation | External | Internal (OpenBabel/antechamber/MATCH) |
| Minimization | External | Internal (OpenMM) |
| Rotamer sampling | TINKER | *FFEvaluate* |
| Rotamer library | CMLib | CMLib/Dunbrack |
| Energy computation | BALL | *FFEvaluate* |
| Force field | AMBER96 | AMBER ff14SB/CHARMM36 |
| Scoring options | CADDSuite/Vina | Smina/*FFEvaluate* |
| Compute detection | Re-computation | Detection of computed elements |
| Time estimation | None | Progress bars |

Abbreviations: API, application programming interface; CLI, command-line interface.

GPU-accelerated minimization. In addition, we implemented a small molecule preparation interface that uses the OpenBabel chemical toolbox[29] for protonation and the Antechamber software,[30] or MATCH[31] for parameterization.

Rotamer sampling previously relied on the molecular modeling software TINKER.[32] During the procedure, clashing rotamers were minimized to induce a better fit. Since this minimization can distort the resulting rotamers and is based on an older force field version, we replaced TINKER with the force field evaluation tool *FFEvaluate*.[33] For the same reason and to further limit external dependencies, the Biochemical Algorithms Library (BALL),[34] previously used for all energy calculations, was replaced by *FFEvaluate*. Whereas for BALL, all atom types had to be manually predefined for the AMBER96 force field, *FFEvaluate* handles them through a Python library called ParmEd,[35] allowing the usage of newer force fields such as AMBER ff14SB or CHARMM36. In addition, the scoring function for ligand interactions has been adapted. While version 1.0 included CADDSuite[36] and AutoDock Vina,[8] CADDSuite has been removed due to its dependency on the BALL library. AutoDock Vina, on the other hand, is now included in Smina,[22] which is a new fork and includes other scoring functions such as Vinardo (Vina RaDii Optimized).[9] These scoring functions differ in their compilation of scoring terms describing effects such as vdW interactions, electrostatics, and solvation. Besides, *FFEvaluate* has been implemented for binding-related energy calculations based on force fields that are also used to evaluate internal protein interactions. Accordingly, the enhancements and improvements not only make the pipeline more consistent, but also make it less reliant on the use of external software.

## 2.3 | Benchmarking

PocketOptimizer 1.0 was validated against a benchmark compiled from the 2010 version of the PDBbind database.[37,38] Complexes were selected based on the availability of a high-quality crystal structure with no mutations outside of the binding pocket, only minor conformational differences of the backbone in the binding pocket, with less than seven potential binding water molecules, and with less than 15 rotatable bonds in the ligand. According to these selection criteria, a benchmark set consisting of 12 differently folded proteins had been compiled.[20] For each protein, at least two mutational variants with a corresponding affinity measure for the same ligand were included. To validate the new version of PocketOptimizer, we compiled a subset based on this original benchmark. Pairs of mutational variants with at least a 50-fold difference in binding affinity were selected. This difference in binding affinity was considered to be well outside of experimental error and should be predicted by our design pipeline. In addition, we extended the benchmark set with new structures from the 2020 version of the PDBbind database using the same selection criteria. Overall, our new benchmark set consists of 13 different proteins and 33 protein crystal structures (see Table S1). Skeletal representations of all ligands included in the compiled benchmark set are shown in Figure S1.

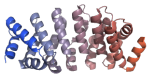


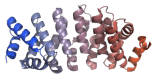
**TABLE 2** Correctly ranked design mutation pairs

| | Original sampling procedure and library (PocketOptimizer 2.0) | | New sampling procedure and library (PocketOptimizer 2.0) | | Original data (PocketOptimizer 1.0) | |
|---|---|---|---|---|---|---|
| **Test case** | **Total** | **Binding** | **Total** | **Binding** | **Total** | **Binding** |
| D7r4 amine-binding protein | 1/1 | 1/1 | 1/1 | 1/1 | 1/1 | 1/1 |
| ABC transporter alpha-glycoside-binding protein | 0/2 | 0/2 | 1/2 | 1/2 | −/− | −/− |
| Estrogen receptor α | 1/1 | 1/1 | 1/1 | 1/1 | 1/1 | 1/1 |
| FimH Fimbrial adhesin | 2/2 | 2/2 | 2/2 | 2/2 | −/− | −/− |
| HIV-1 protease | 5/5 | 5/5 | 4/5 | 5/5 | 5/5 | 5/5 |
| Ketosteroid isomerase | 2/2 | 1/2 | 2/2 | 2/2 | 2/2 | 2/2 |
| Lysine-, arginine-, ornithine-binding periplasmic protein | 7/10 | 9/10 | 7/10 | 9/10 | −/− | −/− |
| Neuroamidase N1 | 3/4 | 2/4 | 2/4 | 2/4 | 1/4 | 0/4 |
| Nopaline-binding periplasmic protein | 1/2 | 1/2 | 0/2 | 1/2 | −/− | −/− |
| Purine nucleoside phosphorylase (PNP) | 1/4 | 0/4 | 4/4 | 4/4 | 7/8 | 6/8 |
| Streptavidin | 5/5 | 4/5 | 5/5 | 5/5 | 5/5 | 5/5 |
| Thymidylate synthase (TS) | 0/4 | 3/4 | 1/4 | 2/4 | 1/6 | 0/6 |
| Anionic trypsin 2 | 1/2 | 2/2 | 1/2 | 2/2 | 1/2 | 1/2 |
| Mean | 65.9% | 70.5% | 70.5% | 84.1% | 70.6% | 61.8% |

*Note*: This is shown for two different versions of PocketOptimizer 2.0 using two rotamer sampling procedures in combination with two different rotamer libraries and for the original data from benchmarking with PocketOptimizer 1.0 (Vina). For PNP and TS, the number of pairs differs since we were more stringent in applying the cutoff of a 50-fold affinity change for each pair. For each test case, the total number of design mutation pairs and the number of correctly ranked pairs by total energy or by binding energy are indicated. The mean value refers to the number of correct predictions in relation to the total number of predictions made.

## 2.4 | Backbone-dependent rotamers lead to improved prediction accuracy

We tested PocketOptimizer 2.0 against the extended benchmark set to compare both versions of the software. The results indicate a similar performance, with a mean prediction accuracy of about 66% compared to about 71% in the first version (see Table 2). Significant differences were found only in two test cases, namely, neuroamidase N1, where the new version gave significantly better predictions, and purine nucleoside phosphorylase, where it made significantly worse predictions. In both test cases, this has been attributed to the fact that minimizing rotamers with TINKER led to a general preference for larger amino acids, as they can engage in more favorable interactions. We can largely overcome this bias by using a backbone-dependent rotamer library and performing no subsequent rotamer minimization, which leads to a prediction accuracy of 70% (see Table 2). Looking only at the cases tested with both versions, PocketOptimizer 2.0 with the new rotamer sampling method achieves a higher overall prediction accuracy of 75% according to the total energies. If only the binding-related energies are considered, this trend becomes even clearer, with the original rotamer library and sampling procedure achieving a prediction accuracy of about 71%, while the new sampling method and library lead to a higher prediction accuracy of about 84%. This is particularly evident in the case of purine nucleoside phosphorylase, where the original rotamer sampling procedure and library were only able to correctly predict one out of four cases, whereas our new rotamer sampling procedure in combination with the Dunbrack rotamer library leads to correct predictions in all cases. eroid isomerase, LAOBP: lys In this test case, the relevant design position is at the entrance of the binding pocket and assumed to influence binding dynamics, as it also has a high temperature factor.[39] Three different variants were tested with PocketOptimizer 2.0: Histidine, aspartate, and phenylalanine, with the histidine showing significantly higher binding affinity. Like the first, the new version correctly predicts hydrogen bonds between ligand and aspartate. For histidine, this is the case only when we use our new sampling procedure and backbone-dependent rotamers. Nonetheless,



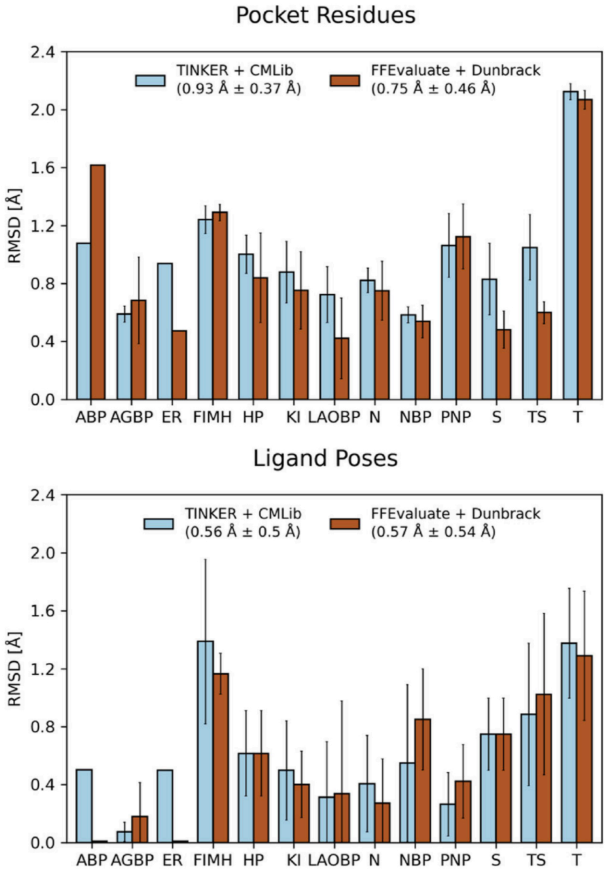



**FIGURE 2** Pocket residue and ligand pose RMSD values between experimentally determined and designed structures. RMSD values were calculated after superimposing the structures using their backbone atoms. Only heavy atoms were considered in all calculations, and only residues that were allowed to change conformations during the designs were included. For each protein test case that included more than one crystal structure, the average RMSD and standard deviation were calculated. Protein test cases are ABP: D7r4 amine-binding protein, AGBP: ABC transporter alpha-glycoside-binding protein, ER: estrogen receptor α, FIMH: fimH fimbrial adhesin, HP: HIV-1 protease, KI: ketosteroid isomerase, LAOBP: lysine-, arginine-, ornithine-binding periplasmic protein, N: neuroamidase N1, NBP: nopaline-binding periplasmic protein, PNP: purine nucleoside phosphorylase, S: streptavidin, TS: thymidylate synthase, T: anionic trypsin 2

phenylalanine is rotated toward the ligand, regardless of rotamer sampling, and forms favorable vdW interactions, whereas it points away from the ligand in the crystal structure (see Figure S2).

To gain further insight, we calculated structural deviations between experimentally determined and designed structures. On the one hand, we focused on the designed pocket residues, and on the other, on the predicted ligand poses (see Figure 2). We found that the pocket side chains deviate by 0.93 Å on average when designed with the original rotamer sampling method and library, while they deviate by only 0.75 Å with our new method and library. Not only the affinity predictions are more accurate overall, but also the pocket side chains are better reproduced on average. The ligand poses, on the other hand, are more comparable, differing by 0.56 Å on average with the original procedure and by 0.57 Å when *FFEvaluate* and Dunbrack are used. Nevertheless, significantly better pose predictions are observed for two test cases (ABP and ER). This indicates an overall good prediction of poses by PocketOptimizer. However, since the ligand starting poses were taken from the initial structures (see calculations) and often differ only slightly between mutants, the structural deviations may be higher than the suggested values.

## 3 | CONCLUSION

PocketOptimizer 2.0 has been updated and refined to predict affinity-improving mutations and to design protein–small molecule interactions. Different functions, such as scoring, can be easily compared, and approaches can be optimized for a specific design task. The program provides a clean user interface. Its compute times have been significantly improved by adapting the pipeline to multicore processing. The preparation of the protein scaffold and the ligand are now included in the pipeline, as well as a minimization step. To extend the modularity of the pipeline, we added the options for rotamer libraries, scoring functions, and force fields. In addition, rotamer sampling and energy calculations have been updated with newer tools. This improved version of PocketOptimizer performs as good or even better than its predecessor on an extended benchmark set. Overall, the affinity predictions appear to be more accurate, and also the pocket side chains are better reproduced on average. Thus, PocketOptimizer 2.0 provides a robust and versatile framework for the design of small molecule-binding pockets in proteins.

## 4 | MATERIALS AND METHODS

Protein and ligand structures were taken from the PDB, and ligand starting poses were assumed to be the same as in the crystal structures. Protonation states were adjusted according to the pH values reported in the literature for affinity measurements (see Table S1). Side chains were minimized with the AMBER ff14SB force field and allowed to change conformations during designs if they were within 4 Å of the ligand or a Cα atom of a mutation

position. Residues located at the end of protein segments or involved in disulfide bridges were kept static. The number of ligand conformations was selected according to the number of rotatable bonds a ligand contains. Ligand poses were then created by rotating all generated conformations by ±20° around each axis and translating them by ±0.5 Å in each direction. Rotamer sampling was performed using two different procedures. First, TINKER in combination with the CMLib rotamer library and the AMBER96 force field was used, and second, *FFEvaluate* in combination with the Dunbrack rotamer library, and the AMBER ff14SB force field was used. Of the rotamers and ligand poses generated, only those with a vdW energy of less than 100 kcal/mol in the scaffold were kept. Protein–protein interactions were assessed based on the AMBER ff14SB force field, while protein–ligand interactions were evaluated using the Autodock Vina scoring function and were upscaled by a factor of 50. According to the objective of the design, predictions were considered to be correct if, after the identification of the GMEC, the binding energy of the mutant that experimentally shows higher binding affinity is lower.

## AUTHOR CONTRIBUTIONS

**Jakob Noske:** Formal analysis (equal); methodology (equal); validation (equal); writing – original draft (equal); writing – review and editing (equal). **Josef Paul Kynast:** Methodology (equal); validation (equal); writing – review and editing (equal). **Dominik Lemm:** Methodology (equal); validation (equal). **Steffen Schmidt:** Formal analysis (equal); methodology (equal); validation (equal); writing – review and editing (equal). **Birte Höcker:** Conceptualization (equal); formal analysis (equal); funding acquisition (equal); writing – original draft (equal); writing – review and editing (equal).

## ACKNOWLEDGMENTS

We thank Jan Bodenschlägel and Florian Gisdon for bugfixes; Celina Seidl and Pascal Kröger for application and testing; and the AG Höcker for discussions. Work on PocketOptimizer has been supported by Deutsche Forschungsgemeinschaft Grant HO 4022/2-3 and the European Research Council H2020 -FETOpen RIA grant. Open Access funding enabled and organized by Projekt DEAL.

## CONFLICT OF INTEREST

The authors declare no potential conflict of interest.

## DATA AVAILABILITY STATEMENT

PocketOptimizer 2.0 is available under the GNU general public license v3.0 under the following URL: https://github.com/Hoecker-Lab/pocketoptimizer.

## ORCID

*Jakob Noske* https://orcid.org/0000-0001-7763-2645
*Josef Paul Kynast* https://orcid.org/0000-0003-1412-1797
*Dominik Lemm* https://orcid.org/0000-0002-8075-1765
*Steffen Schmidt* https://orcid.org/0000-0001-9077-6010
*Birte Höcker* https://orcid.org/0000-0002-8250-9462

## REFERENCES

1. Jiang L, Althoff EA, Clemente FR, et al. De novo computational design of retro-aldol enzymes. Science. 2008;319(5868):1387–1391.
2. Röthlisberger D, Khersonsky O, Wollacott AM, et al. Kemp elimination catalysts by computational enzyme design. Nature. 2008;453(7192):190–195.
3. Siegel JB, Zanghellini A, Lovick HM, et al. Computational design of an enzyme catalyst for a stereoselective bimolecular Diels-Alder reaction. Science. 2010;329(5989):309–313.
4. Liu DS, Nivón LG, Richter F, et al. Computational design of a red fluorophore ligase for site-specific protein labeling in living cells. Proc Natl Acad Sci U S A. 2014;111(43):E4551–E4559.
5. De Los Santos ELC, Meyerowitz JT, Mayo SL, Murray RM. Engineering transcriptional regulator effector specificity using computational design and in vitro rapid prototyping: Developing a vanillin sensor. ACS Synthetic Biology. 2016;5(4):287–295.
6. Herud-Sikimić O, Stiel AC, Kolb M, et al. A biosensor for the direct visualization of auxin. Nature. 2021;592(7856):768–772.
7. Morris GM, Huey R, Lindstrom W, et al. AutoDock4 and AutoDockTools4: Automated docking with selective receptor flexibility. J Comput Chem. 2009;30(16):2785–2791.
8. Trott O, Olson AJ. AutoDock Vina: Improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. J Comput Chem. 2010;31(2):455–461.
9. Quiroga R, Villarreal MA. Vinardo: A scoring function based on autodock vina improves scoring, docking, and virtual screening. PLoS One. 2016;11(5):e0155183.
10. Katkova EV, Onufriev AV, Aguilar B, Sulimov VB. Accuracy comparison of several common implicit solvent models and their implementations in the context of protein-ligand binding. J Mol Graph Model. 2017;72:70–80.
11. Gopal SM, Klumpers F, Herrmann C, Schäfer LV. Solvent effects on ligand binding to a serine protease. Phys Chem Chem Phys. 2017;19(17):10753–10766.
12. Nguyen NT, Nguyen TH, Pham TNH, et al. Autodock Vina adopts more accurate binding poses but Autodock4 forms better binding affinity. J Chem Inf Model. 2020;60(1):204–211.
13. Obabel. Generate multiple conformers—Open Babel 3.0.1 documentation. [cited 2021 Jul 2]. Available from: https://openbabel.readthedocs.io/en/latest/3DStructureGen/multipleconformers.html
14. O'Boyle NM, Vandermeersch T, Flynn CJ, Maguire AR, Hutchison GR. Confab––Systematic generation of diverse low-energy conformers. J Chem. 2011;3(1):8.
15. Sontag D, Meltzer T, Globerson A, Jaakkola T, Weiss Y. Tightening LP relaxations for MAP using message passing.

Proceedings of the 24th Conference on Uncertainty in Artificial Intelligence, UAI 2008. 2008 p. 503–510.
16. Traoré S, Allouche D, André I, et al. A new framework for computational protein design through cost function network optimization. Bioinformatics. 2013;29(17):2129–2136.
17. Barlow KA, Ó Conchúir S, Thompson S, et al. Flex ddG: Rosetta ensemble-based estimation of changes in protein–protein binding affinity upon mutation. J Phys Chem B. 2018;122(21):5389–5399.
18. Hallen MA, Martin JW, Ojewole A, et al. OSPREY 3.0: Open-source protein redesign for you, with powerful new features. J Comput Chem. 2018;39(30):2494–2507.
19. Panel N, Villa F, Opuu V, Mignon D, Simonson T. Computational design of PDZ-peptide binding. Methods in Molecular Biology. 2021;2256:237–255.
20. Malisi C, Schumann M, Toussaint NC, Kageyama J, Kohlbacher O, Höcker B. Binding pocket optimization by computational protein design. PLoS One. 2012;7(12):e52505.
21. Shapovalov MV, Dunbrack RL. A smoothed backbone-dependent rotamer library for proteins derived from adaptive kernel density estimates and regressions. Structure. 2011;19(6):844–858.
22. Koes DR, Baumgartner MP, Camacho CJ. Lessons learned in empirical scoring with smina from the CSAR 2011 benchmarking exercise. J Chem Inf Model. 2013;53(8):1893–1904.
23. Schroedinger. The PyMOL Molecular Graphics System, Version 1.2r3pre, Schrödinger, LLC.
24. Stiel AC, Nellen M, Höcker B. PocketOptimizer and the design of ligand binding sites. Methods in Molecular Biology. 1414. Humana Press Inc.; 2016. 63–75.
25. Pettersen EF, Goddard TD, Huang CC, et al. UCSF chimera––A visualization system for exploratory research and analysis. J Comput Chem. 2004;25(13):1605–1612.
26. Doerr S, Harvey MJ, Noé F, De Fabritiis G. HTMD: High-throughput molecular dynamics for molecular discovery. J Chem Theory Comput. 2016;12(4):1845–1852.
27. Søndergaard CR, Olsson MHM, Rostkowski M, Jensen JH. Improved treatment of ligands and coupling effects in empirical calculation and rationalization of pKa values. J Chem Theory Comput. 2011;7(7):2284–2295.
28. Eastman P, Swails J, Chodera JD, et al. OpenMM 7: Rapid development of high performance algorithms for molecular dynamics. PLoS Comput Biol. 2017;13(7):e1005659.
29. O'Boyle NM, Banck M, James CA, Morley C, Vandermeersch T, Hutchison GR. Open babel: An open chemical toolbox. J Chem. 2011;3:33.
30. Wang J, Wang W, Kollman PA, Case DA. Automatic atom type and bond type perception in molecular mechanical calculations. J Mol Graph Model. 2006;25(2):247–260.
31. Yesselman JD, Price DJ, Knight JL, Brooks CL. MATCH: An atom-typing toolset for molecular mechanics force fields. J Comput Chem. 2012;33(2):189–202.
32. Rackers JA, Wang Z, Lu C, et al. Tinker 8: Software tools for molecular design. J Chem Theory Comput. 2018;14(10):5273–5289.
33. Ffev. HTMD FFEvaluate–Easy MM force-field evaluation—HTMD 1.24.8 documentation. [cited 2021 Jul 2]. Available from: https://software.acellera.com/htmd/tutorials/FFEvaluate.html
34. Hildebrandt A, Dehof AK, Rurainski A, et al. BALL––Biochemical Algorithms Library 1.3. BMC Bioinformatics. 2010;11(1):531.
35. Parmed. ParmEd/ParmEd: Parameter/topology editor and molecular simulator. [cited 2021 Jul 1]. Available from: https://github.com/ParmEd/ParmEd
36. Kohlbacher O. CADDSuite––A workflow-enabled suite of open-source tools for drug discovery. J Chem. 2012;4(S1):1.
37. Liu Z, Li Y, Han L, et al. PDB-wide collection of binding data: Current status of the PDBbind database. Bioinformatics. 2015;31(3):405–412.
38. Liu Z, Su M, Han L, et al. Forging the basis for developing protein–ligand interaction scoring functions. Acc Chem Res. 2017;50(2):302–309.
39. Murkin AS, Birck MR, Rinaldo-Matthis A, et al. Neighboring group participation in the transition state of human purine nucleoside phosphorylase. Biochemistry. 2007;46(17):5038–5049.

# Supplementary Information for

## PocketOptimizer 2.0: A modular framework for computer-aided ligand-binding design

by Jakob Noske, Josef Paul Kynast, Dominik Lemm, Steffen Schmidt, and Birte Höcker

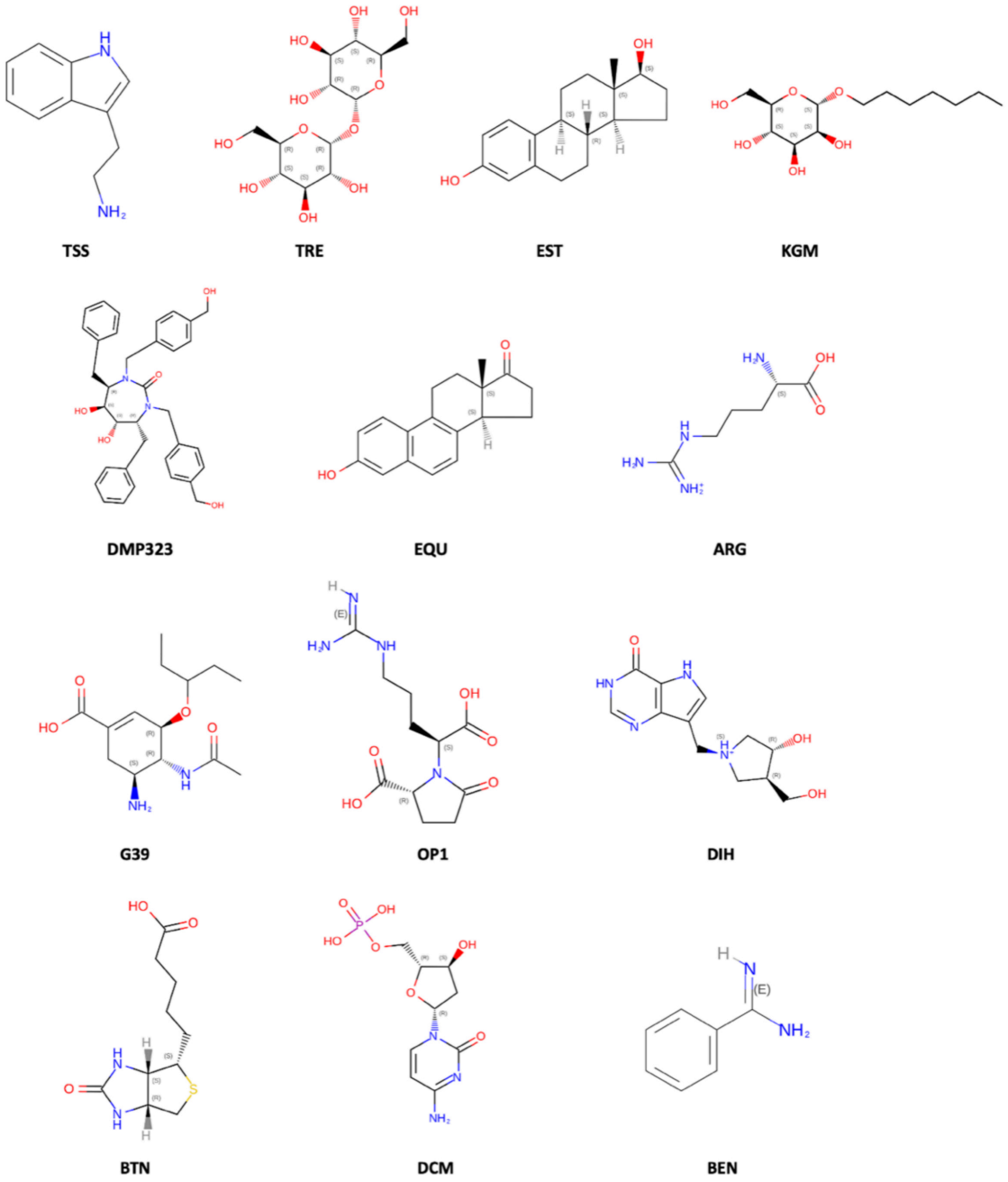



**Figure S1:** Skeletal representation of ligands included in the benchmark set. The ligand identifiers are shown below, structures were taken from: https://www.rcsb.org.

**Table S1:** Benchmark used to evaluate PocketOptimizer 2.0 in comparison. It is a subset of the one used to test PocketOptimizer 1.0 and includes all pairs of mutational variants with an affinity change of at least 50-fold. Protein-ligand complexes are sorted by protein names and ligands are listed by their ligand identifiers. Wildtype structures are indicated and all mutations listed. In case of HIV-1 protease, where the binding pocket is formed by two chains, all indicated mutations are present in both chains. The experimentally determined binding affinities for each complex are listed along with the PDB identifier of the experimentally solved structures, if available.

| Protein | Ligand | Mutation(s) | Affinity [nM] | PDB |
|---|---|---|---|---|
| **D7r4 Amine Binding Protein**[1] | TSS | WT | Inf | - |
| | | L111D | 53 | 2pql |
| **ABC Transporter Alpha-Glycoside-Binding Protein**[2] | TRE | WT | 7460 | 6j9w |
| | | W287A | 58 | 6jb0 |
| **Estrogen Receptor a**[3,4] | EST | WT | 0.29 | 1gwr |
| | | E353A | 60 | - |
| **FimH Fimbrial Adhesin**[5] | KGM | WT | 1.1 | 4xo8 |
| | | Y137A | 206.4 | 5fs5 |
| **HIV-1 Protease**[6] | DMP323 | WT | 0.8 | - |
| | | V82F | 0.4 | 1met |
| | | I84V | 20 | 1mes |
| | | V82F, I84V | 800 | 1meu |
| **Ketosteroid Isomerase**[7] | EQU | WT | 45750 | 1oh0 |
| | | D40N | 810 | 1ogx |
| **Lysine-, Arginine-, Ornithine-Binding Periplasmic Protein**[8] | ARG | WT | 1.0 | 6mle |
| | | D11A | 70 | 6mku |
| | | Y14A | 800 | 6mlo |
| | | D30A | 5000 | 6ml9 |
| | | R77A | 9000 | 6mlg |
| | | D161A | 42000 | 6mla |
| **Neuroamidase N1**[9] | G39 | WT | 0.32 | 2hu4 |
| | | H274Y | 84.8 | 3cl0 |
| | | N294S | 25.9 | 3cl2 |
| **Nopaline-Binding Periplasmic Protein**[10] | OP1 | WT | 0.5 | 4pow |
| | | M117N | 39.9 | 4pp0 |

| | | | | |
|---|---|---|---|---|
| **Purine Nucleoside Phosphorylase**[11] | DIH | WT | 0.01 | 1rsz |
| | | H257D | 0.9 | 2a0y |
| | | H257F | 0.95 | 2a0x |
| **Streptavidin**[12] | BTN | WT | 0.0001 | 1swe |
| | | N23A | 0.028 | 1n43 |
| | | N23E | 0.0069 | - |
| | | S27A | 0.011 | 1n9m |
| **Thymidylate Synthase**[13] | DCM | WT | 160000 | 1nje |
| | | N229C | 490 | 1nja |
| | | N229D | 2800 | 1njc |
| **Anionic Trypsin 2**[14] | BEN | WT | 12000 | 1ane |
| | | D189G, G226D | 15000000 | 1bra |

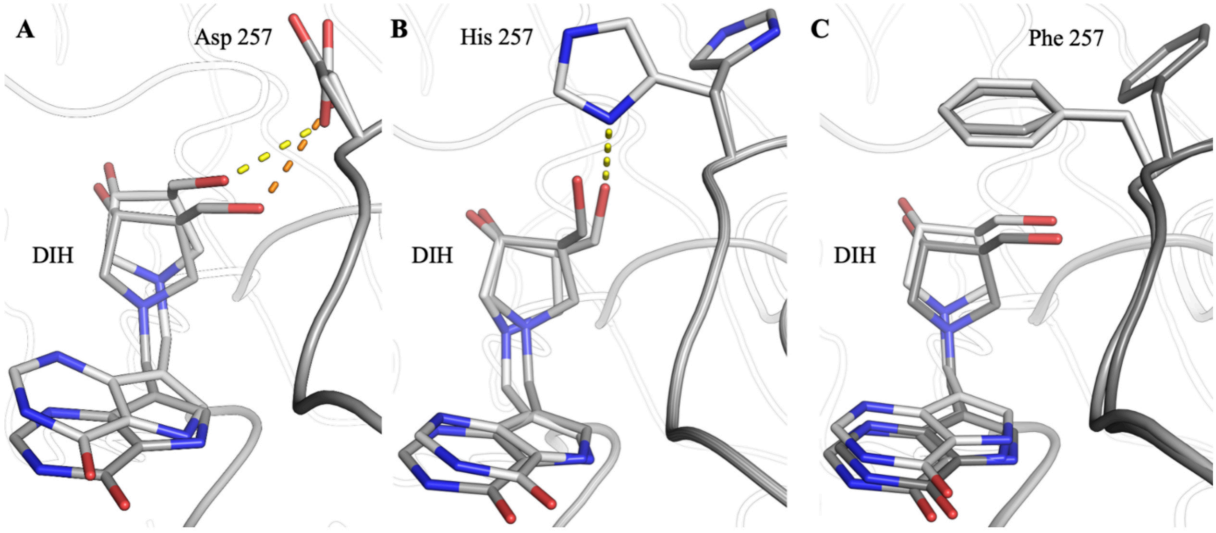



**Figure S2**: Binding pocket of purine nucleoside phosphorylase with the ligand DIH. Designs based on the original rotamer sampling method and library are shown in light gray, while designs based on the newly implemented rotamer sampling method and library are shown in white. In all designs and crystal structures the conformation of the side chain at mutation position 257 is highlighted. A): Aspartate mutations calculated based on the wildtype crystal structure: 1rsz, hydrogen bonds between aspartate and the ligand are depicted in yellow and orange. B) Histidine mutations calculated based on the mutant crystal structure: 2a0x, hydrogen bonds between histidine and the ligand are depicted in yellow. C) Phenylalanine mutations based on the wildtype crystal structure: 1rsz, the crystal structure of the mutated protein (2a0x) is depicted in dark gray.

## References

1. Mans BJ, Calvo E, Ribeiro JMC, Andersen JF. The crystal structure of D7r4, a salivary biogenic amine-binding protein from the malaria mosquito Anopheles gambiae. The Journal of biological chemistry. 2007;282(50):36626–36633.

2. Chandravanshi M, Gogoi P, Kanaujia SP. Structural and thermodynamic correlation illuminates the selective transport mechanism of disaccharide α-glycosides through ABC transporter. The FEBS journal. 2020;287(8):1576–1597.

3. Chen Z, Katzenellenbogen BS, Katzenellenbogen JA, Zhao H. Directed evolution of human estrogen receptor variants with significantly enhanced androgen specificity and affinity. The Journal of biological chemistry. 2004;279(32):33855–33864.

4. Shi Y, Koh JT. Selective regulation of gene expression by an orthogonal estrogen receptor-ligand pair created by polar-group exchange. Chemistry & biology. 2001;8(5):501–510.

5. Rabbani S, Krammer EM, Roos G, Zalewski A, Preston R, Eid S, Zihlmann P, Prévost M, Lensink MF, Thompson A, et al. Mutation of Tyr137 of the universal Escherichia coli fimbrial adhesin FimH relaxes the tyrosine gate prior to mannose binding. IUCrJ. 2017;4(Pt 1):7–23.

6. Ala PJ, Huston EE, Klabe RM, McCabe DD, Duke JL, Rizzo CJ, Korant BD, DeLoskey RJ, Lam PYS, Nicholas Hodge C, et al. Molecular basis of HIV-1 protease drug resistance: structural analysis of mutant proteases complexed with cyclic urea inhibitors. Biochemistry. 1997;36(7):1573–1580.

7. Ha NC, Kim MS, Lee W, Choi KY, Oh BH. Detection of Large pKa Perturbations of an Inhibitor and a Catalytic Group at an Enzyme Active Site, a Mechanistic Basis for Catalytic Power of Many Enzymes. Journal of Biological Chemistry. 2000;275(52):41100–41106.

8. Vergara R, Romero-Romero S, Velázquez-López I, Espinoza-Pérez G, Rodríguez-Hernández A, Pulido NO, Sosa-Peinado A, Rodríguez-Romero A, Fernández-Velasco DA. The interplay of protein-ligand and water-mediated interactions shape affinity and selectivity in the LAO binding protein. The FEBS journal. 2020;287(4):763–782.

9. Collins PJ, Haire LF, Lin YP, Liu J, Russell RJ, Walker PA, Skehel JJ, Martin SR, Hay AJ, Gamblin SJ. Crystal structures of oseltamivir-resistant influenza virus neuraminidase mutants. Nature. 2008;453(7199):1258–1261.

10. Lang J, Vigouroux A, Planamente S, El Sahili A, Blin P, Aumont-Nicaise M, Dessaux Y, Moréra S, Faure D. Agrobacterium uses a unique ligand-binding mode for trapping opines and acquiring a competitive advantage in the niche construction on plant host. PLoS pathogens. 2014;10(10):e1004444.

11. Murkin AS, Birck MR, Rinaldo-Matthis A, Shi W, Taylor EA, Almo SC, Schramm VL. Neighboring group participation in the transition state of human purine nucleoside phosphorylase. Biochemistry. 2007;46(17):5038–5049.

12. Le Trong I, Freitag S, Klumb LA, Chu V, Stayton PS, Stenkamp RE. Structural studies of hydrogen bonds in the high-affinity streptavidin-biotin complex: mutations of amino acids interacting with the ureido oxygen of biotin. Acta crystallographica. Section D, Biological crystallography. 2003;59(Pt 9):1567–1573.

13. Finer-Moore JS, Liu L, Schafmeister CE, Birdsall DL, Mau T, Santi DV, Stroud RM. Partitioning roles of side chains in affinity, orientation, and catalysis with structures for mutant complexes: asparagine-229 in thymidylate synthase. Biochemistry. 1996;35(16):5125–5136.

14. Perona JJ, Tsu CA, McGrath ME, Craik CS, Fletterick RJ. Relocating a negative charge in the binding pocket of trypsin. Journal of molecular biology. 1993;230(3):934–949.

# X. List of Publications

** Elings W*, Tassoni R*, van der Schoot SA, Luu W, **Kynast JP**, Dai L, Blok AJ, Timmer M, Florea BI, Pannu NS, Ubbink M.

Phosphate Promotes the Recovery of Mycobacterium tuberculosis β-Lactamase from Clavulanic Acid Inhibition

*Biochemistry* **2017**, 56, 47, 6257–6267

** **Kynast JP**, Schwägerl F, Höcker B.

ATLIGATOR: editing protein interactions with an atlas-based approach.

*bioRxiv* **2022**, https://doi.org/10.1101/2022.01.19.476980

Gisdon FJ*, **Kynast JP***, Ayyildiz M*, Hine AV, Plückthun A, Höcker B.

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

*Biol. Chem.* **2022**; 403(5-6): 535-543

**Kynast JP**, Schwägerl F, Höcker B.

ATLIGATOR: editing protein interactions with an atlas-based approach.

*Bioinformatics* **2022** Nov 30; 38(23): 5199-5205

Noske J, **Kynast JP**, Lemm D, Schmidt S, Höcker B.

PocketOptimizer 2.0: A modular framework for computer-aided ligand-binding design.

*Protein Sci.* **2023** Jan; 32(1): e4516

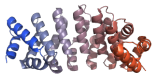
**Kynast JP**, Höcker B.

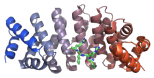
Atligator Web: A Graphical User Interface for Analysis and Design of Protein–Peptide Interactions.

*BioDesign Research* **2023**; 5; 0011

* equal contribution

** not part of this thesis



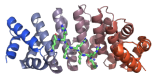


# Acknowledgements

This work would not be possible without the support of many special people.

Thank you.

```
# Armadillo repeat protein progress bar
import pathlib
from pymol import cmd

def armadillo_progress_bar(fraction, bar_width=315, bar_height=189, file="progress_bar.png"):
    n_chain_a = 286
    offset_a = 7
    n_chain_d = 10
    threshold_only_a = n_chain_a / (n_chain_a + n_chain_d)

    cmd.hide("all")
    cmd.show("cartoon", "chain A")
    cmd.show("sticks", "chain D")

    if fraction <= threshold_only_a:
        cmd.hide("sticks", "chain D")
        cmd.hide(f"cartoon",
                 f"chain A and resi {int(fraction*(n_chain_a + n_chain_d) + offset_a)}" \
                  "-{n_chain_a + offset_a}")
    else:
        hidden_fraction_of_d = (1-fraction)/(1-threshold_only_a)
        cmd.hide("sticks",
                 f"chain D and resi {n_chain_d - round(hidden_fraction_of_d*n_chain_d) + 1}-{n_chain_d}")
    cmd.select("n_cap", "resi 8-41 and chain A")
    cmd.select("c_cap", "resi 252-293 and chain A")
    cmd.select("r1", "resi 42-83 and chain A")
    cmd.select("r2", "resi 84-125 and chain A")
    cmd.select("r3", "resi 126-167 and chain A")
    cmd.select("r4", "resi 168-209 and chain A")
    cmd.select("r5", "resi 210-251 and chain A")
    if fraction == 1.0:
        cmd.select("pocket", "resi 155,159,162,194,197,198,201")
        cmd.show("sticks", "pocket")
    cmd.select("peptide", "chain D")
    cmd.color("0x335FFF", "n_cap")
    cmd.color("0x707AB3", "r1")
    cmd.color("0x88769E", "r2")
    cmd.color("0xA07188", "r3")
    cmd.color("0xB86D73", "r4")
    cmd.color("0xD0685E", "r5")
    cmd.color("0xFF5F33", "c_cap")
    cmd.color("lime", "peptide")
    cmd.color("atomic", "(not elem C)")
    cmd.bg_color("white")
    cmd.set("stick_radius", "0.5")

    cmd.set_view("-0.283387840, -0.611639082, -0.738634765, 0.378934324, 0.636115015, -0.672128260," \
                 "0.880965889, -0.470369905, 0.051503245, 0.000000000, 0.000000000, -122.575881958," \
                 "8.341945648, 53.976295471, -13.899051666, 55.680236816, 189.471694946, -20.000000000")
    cmd.png(file, width=bar_width, height=bar_height, dpi=300)

# Fetch the PDB structure 5AEI
cmd.fetch("5AEI")
# Remove unnecessary atoms
cmd.remove("resn HOH or (not chain A and not chain D)")
pages = 93
dir = pathlib.Path("progress_bar")
dir.mkdir(exist_ok=True)
for page in range(1, pages + 1):
    armadillo_progress_bar(
        page/pages, file=str(dir/f"progress_{str(page).zfill(len(str(pages)))}.png")
    )
```

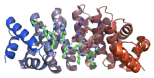


I am deeply grateful to my supervising PI, Prof. Dr. Birte Höcker. I thank her for giving me the opportunity to start a purely computational project in the light of my different background. I highly appreciate her ongoing belief in me and her mentorship throughout my doctoral journey.

I want to thank my amazing colleagues, especially (random order):
Felix, for starting this journey with me and providing the guidance I needed in the beginning.
Noelia, for every discussion and feedback as well as a pleasant welcome.
Flo, for being my partner in crime and always having an open ear for me.
Merve and Jakob, for all the valuable exchange and all your support.
Emmy for supporting me during her bachelor's thesis.
Steffen, for guidance and help on everything - from the clusters to the web servers.

Thank you to everyone who shared this journey with me. All of my colleagues helped me in so many ways. They have been a great source of joy and inspiration, and I am grateful to be part of this group. In alphabetical order: Abhishek, Andi, Andreas, Anke, Anna, Anna, Anu, Basti, Ben, Bruce, Dominik, Erich, Flo M., Francisco, Gabi, Guto, Gwen, Horst, Jeli, Johanna, Jonas, Julian, Katha, Lukas, Mo, Nicteh, Olivier, Onur, Pascal, Sabrina, Sergio, Sina, Sooruban, Surbhi, Yvi, … and everyone who I cannot mention here.

Furthermore, I would like to acknowledge my family. I want to thank my parents and my brothers for their constant support.
Finally, my most sincere gratitude deserve my wife Lena and my daughters Romina and Amelie (& Simba). Thank you for your patience, love, and support through all the highs and lows. *I love you!*



# (Eidesstattliche) Versicherungen und Erklärungen

*(§ 8 Satz 2 Nr. 3 PromO Fakultät)*

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 PromO Fakultät)*

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. 4 PromO Fakultät)*

Hiermit erkläre ich, dass ich Hilfe von gewerblichen Promotionsberatern bzw. -vermittlern oder ähnlichen Dienstleistern weder bisher in Anspruch genommen habe noch künftig in Anspruch nehmen werde.

*(§ 8 Satz 2 Nr. 7 PromO Fakultät)*

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. 8 PromO Fakultät)*

Hiermit erkläre ich mein Einverständnis, dass bei Verdacht wissenschaftlichen Fehlverhaltens Ermittlungen durch universitätsinterne Organe der wissenschaftlichen Selbstkontrolle stattfinden können.

(Ort, Datum) Unterschrift