Design of Protein Interfaces Using Computer-Based Methods

Dissertation

zur Erlangung des Doktorgrades der Naturwissenschaften (Dr. rer. nat.) der Fakultät für Biologie und Vorklinische Medizin der Universität Regensburg

vorgelegt von
Julian Simon Nazet
aus München

November 2020

Das Promotionsgesuch wurde eingereicht am: 13.11.2020

Die Arbeit wurde angeleitet von: Prof. Dr. Rainer Merkl

Unterschrift: .....

Julian Nazet

Abstract

Computer-based methods are excellent tools to modify existing proteins. The software suite Rosetta offers a large variety of options to solve many problems of protein design. During the last decades a steadily increasing number of protocols became available and more complex concepts for the modelling of proteins and their function arouse. Among them is multi-state protein design (MSD) that utilizes in parallel several three-dimensional conformations of a protein to increase the chances of a successful design.

In the first part of this work I used MSD to reprogram a protein interface by means of an anchored design approach. The starting point was a pair of glutamine amidotransferase complexes consisting both of homologous pairs of synthase and glutaminase subunits. The goal was to alter the interface of the synthase subunit PabA such that it is no longer able to bind the native glutaminase subunit PabB, but the homolog TrpEx which forms a native complex with the synthase TrpG. The experimental characterization confirmed that a grafting of TrpG-specific interface residues into the PabA interface and a subsequential design by means of Rosetta gave rise to a PabA variant that exclusively bound to TrpEx.

In the second part of this work I present a novel combination of a Rosetta protocol and a neural network (NN) which is used to rapidly score candidate sequences. Generally, protein design protocols require a high computational effort and therefore it is worth to develop time-saving extensions that do not degrade the design performance. My aim was to implement a hybrid combination of an NN and the classical Rosetta approach, with the NN deducing the energy landscape from the Rosetta scores of relatively few candidates. Thereby I employed a hybrid approach of a neural network and Rosetta, whereby the neural network learns the energy landscape from Rosetta and vice versa Rosetta evaluates the predictions of the neural network. Due to its speed, the trained NN allows then the sampling of a much larger region of the vast and design-specific energy landscape spanned by alternative sequences and residue orientations. This approach also facilitates a new way to design MSD protocols, since the outcome of NNs each trained on a different state can be subsequently recombined. A protein benchmark dataset was utilized to test the performance of the new approach named `Rosetta:MSF:NN`. In comparison to a previously described protocol, the new one led to a threefold increase in speed.

References of Published Manuscripts

This thesis is composed of the following published or submitted manuscripts:

A Hertle R.[1], **Nazet, J.**[1], Semmelmann, F., Schlee, S., Funke, F., Merkl, R., & Sterner, R. (2020). Reprogramming the specificity of a protein interface by computational and data-driven design. *Published at Structure*

[1] These authors contributed equally to this work.

B **Nazet, J.**, Lang, E., Merkl, R. (2020). Rosetta:MSF:NN: Boosting multistate computational protein design with a neural network. *Submitted for Publication*

In the course of this work, I contributed to the following seven publications, which are not part of this thesis:

C Schlee, S., Kein, T., Schumacher, M., **Nazet, J.**, Merkl, R., Steinhoff, HJ., Sterner, R. (2018). Relationship of Catalysis and Active Site Loop Dynamics in the $(\beta\alpha)_8$-Barrel Enzyme Indole-3-glycerol Phosphate Synthase. *Biochemistry*, 57(23), 3265-3277.

D Pfab, A., Bruckmann, A., **Nazet, J.**, Merkl, R., Grasser, KD. (2018). The Adaptor Protein ENY2 Is a Component of the Deubiquitination Module of the Arabidopsis SAGA Transcriptional Co-activator Complex but not of the TREX-2 Complex. *Journal of Molecular Biology* 430(10), 1479-1494

E Rustler, K., Mickert, MJ., **Nazet, J.**, Merkl, R.,Gorris, HH., König, B. (2018). Development of photoswitchable inhibitors for β-galactosidase. *Organic and Biomolecular Chemistry* 16(40), 7430-7437

F Hoffmeister, H., Fuchs, A., Strobl, L., Sprenger, F., Gröbner-Ferreira, R., Michaelis, S., Hoffmann, P., **Nazet, J.**, Merkl, R., Längst, G. (2019). Elucidation of the functional roles of the Q and I motifs in the human chromatin-remodeling enzyme BRG1. *Journal of Biological Chemistry* 294(9), 3294-3310

G Semmelmann, F., Straub, K., **Nazet, J.**, Rajendran, C., Merkl, R., Sterner, R. (2019). Mapping the Allosteric Communication Network of Aminodeoxychorismate Synthase. *Journal of Molecular Biology* 431(15), 2718-2728

H Simeth N., Kinateder T., Rajendran C., Zwisele S., **Nazet J.**, Merkl R., Sterner R., Kneuttinger A. and König B. (2020). Towards Photochromic Azobenzene-Based Inhibitors for Tryptophan Synthase. *Accepted at Chemistry - A European Journal*

I Hoffmeister, H., Fuchs, A., Komives, E., Gröbner-Ferreira, R., Strobl L., **Nazet, J.**, Heizinger, L., Merkl, R., Dove S., Längst G. (2020). The ATPase domains of CHD3 and SNF2H are promising targets for selective regulability or drugability. *Submitted for Publication*

Personal Contributions

Publication A

The computational design of the variants PabA-CA and PabA-CAD was performed by myself. Furthermore, I ran all bioinformatical analyses of all variants and all visualization of the variants was done by myself. Additionally, I contributed to the writing of all bioinformatical parts.

Publication B

The machine learning approach was designed by Elmar Lang, Rainer Merkl, and myself. The benchmarking and analysis of `Rosetta:MSF:NN` was performed by Rainer Merkl and myself. Rainer Merkl and myself wrote the paper.

Contents

List of Figures

List of Tables

Chapter 1

General Introduction

1.1 Proteins

Proteins are key elements of all living organisms and are involved in all essential processes within cells. Proteins are required to maintain the structure of cells or to or transmit signals between cells. A special class of proteins are the enzymes that catalyze biochemical reactions. All proteins are composed of 20 naturally occurring amino acids (AAs), which are synthesized by the cells as part of their metabolism. During protein biosynthesis, AAs are covalently bound according to a blueprint, provided by the DNA. The specific AA sequence determines the protein's structure and function. However, there are multiple sequences coding the same function or structure. These proteins are called homologous proteins because they share a common ancestor. Protein homology can be predicted by determining the number of identical residues at corresponding positions. A sequence identity above 40% strongly indicates structural homology (Rost, 1999) and homologous proteins commonly possess highly similar three dimensional structures. Generally, protein structures are classified into folds which specify the localisation of backbone atoms. Typical folds are, e.g. the TIM-barrel or Rossmann-fold. Interestingly, nature uses only a limited number of 1700±400 protein folds (Sadreyev et al., 2009). This limitation helps with predicting a protein's structure from its sequence. However, computational engineered proteins are capable to generate new, unnatural folds (Kuhlman et al., 2003).

In nature, proteins often form complexes with other proteins. For example, only 20% of *Escherichia coli* proteins are monomers, whereas 80% form complexes with other proteins (Levy et al., 2008). These complexes require, for their proteins, to interact exclusively with their binding partner. However, not only the number of folds, but also the number of interface architectures is limited to ~1000 (Marsh and Teichmann, 2015). To avoid cross talk between proteins sharing the same interface topology but belong to different complexes, some interfaces contain so-called add-ons. These are additional structural elements of a protein-protein interface (Plach et al., 2017) to specify the binding of a protein.

Many proteins are part of the metabolism of microorganisms, which can be categorized to either the "primary" (PM) or the "secondary" (SM) metabolism. The PM contains metabolic

pathways producing natural compounds directly involved in growth, development, and reproduction, whereas the SM contains dispensable compounds (Kossel, 1891). Enzymes of the SM are considered to have evolved from the PM through gene duplication and specification (Cavalier-Smith, 1992). Directly after gene duplication the new product is not directly beneficial, but might later, after some modifications, give a selective advantage. A complicated case arises, if the initial gene and its modified copy are elements of different protein complexes. As long as the interfaces are similar, there is a risk of falsely interacting proteins. In Chapter 2, a especially challenging case is studied. There, PabA:PabB and TrpG:TrpEx, two homologous complexes are analyzed in detail. PabA and TrpG are both glutamine amidotransferases and catalyze the same reaction. PabB and TrpEx both utilize the ammonia provided by their partner and only differ in their final product. TrpEx contains an interface add-on to specify its binding to TrpG. I redesigned PabA to exclusively bind TrpEx and to no longer form a stable complex with PabB, its native partner.

1.2 Protein Design

Computational protein design describes the process of manipulating a protein's sequence to change its behavior. Protein design is, for example, useful to improve the thermal stability of an enzyme or to alter or broaden its substrate spectrum. The most ambitious goal is enzyme design with the aim to create a novel catalytic site such that the protein processes a non-natural substrate. Moreover, redesigning proteins can help to understand how a protein adopts the native state, to elucidate details of protein-protein interactions or to identify critical residues of a catalytic or binding site. In order to estimate the effect of changes introduced into a protein, the protein design programs utilize a three-dimensional model of the protein and a scoring function. Creating a three-dimensional model is straightforward, after a protein's structure has been solved experimentally. The protein data bank (PDB) holds more than 165,000 of these structures. However, even the simplest modification of a protein requires an optimization step to answer the following questions depending on the design task. If a residue is replaced by a larger one, how is this effecting the residues in close proximity? Is the new residue even in the optimal orientation? Are there new cavities introduced into the protein? All these questions need to be answered for protein design with the help of force fields or score functions that model the most relevant interactions often on an atomic level; see Figure 1.1. These terms check if the newly introduced residue or an alternative orientation of a residue clashes with neighboring residues and assign a force to move part of the residue out of the way if needed. They score all possible orientations of a residue, choose the optimal one or evaluate the binding of a ligand and determine its binding properties. Score functions are further used to score the interface of interacting proteins to evaluate their binding strength.

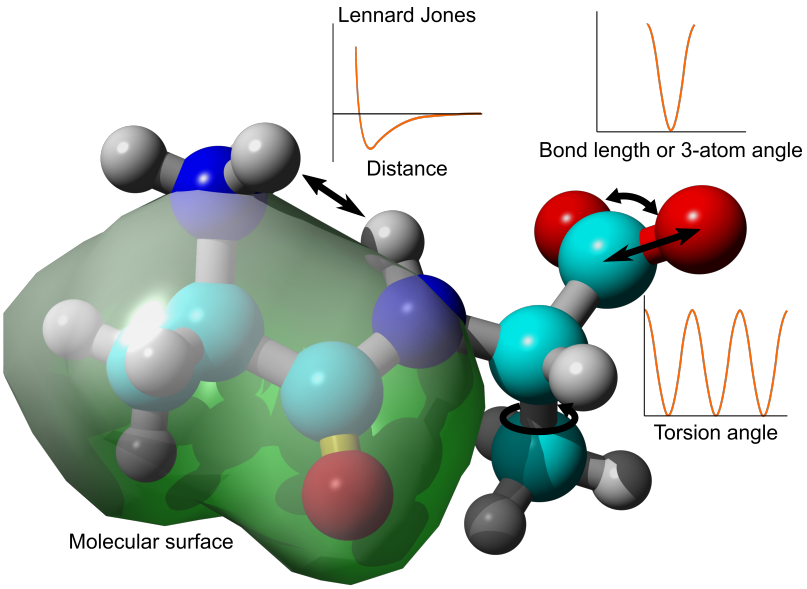

Figure 1.1: Energy based potentials. The scoring function of a protein design algorithm consists of a combination of energy terms that consider physico-chemical properties of proteins. This image illustrates a few of them. The molecular surface of a residue interacts with the solvent, which is usually water and these interactions can be beneficial or unfavorable for hydrophilic or hydrophobic residues. The Lennard-Jones potential approximates the potential energy between two non-covalently bound atoms depending on their distance. For covalently bound atoms, the bond length and three-atom angles may vary according to a potential function. Analogously, the energy of torsion angles is described by a wave-like potential function with distinct minima and maxima.

To build a score function, the software designer has to assess biophysical forces and analyze the structures of native proteins in order to choose the most relevant effects and parametrize them. There are several score functions available and it is up to the user to select a suitable one. The computational assessment of each design solution will contribute a new point in a protein-specific energy landscape, which might be enormously complex. In Chapter 3, a modified Rosetta protocol is presented that enables the user to rapidly scan such energy landscapes for minima via a neural network (see Section 1.4). Rosetta is a state-of-the-art software suite that can be adopted to treat a variety of protein design problems and will be briefly surveyed in the next section.

In order to perform the task to optimize a protein's three-dimensional structure, score function need to know what changes are possible to introduce into the current protein, since not all theoretically possible solutions are naturally occurring. Therefore, score functions need to consider AA side chains, which are the flexible part of an AA compared to the more rigid protein backbone. They are tightly packed and moving or rotating one residue has direct impact on its surrounding positions. The angles between side chain atoms are called torsion angles and they are commonly labeled with $\chi_1, \chi_2, \ldots$. Different torsion angles of one residue are not independent of each other. There is only a discrete number of torsion angle combinations possible for each residue. These conformers are defined by their rotation of their torsion angle and spe-

cific combinations of torsion angles are called rotamers (**rota**tionconfor**mer**). Therefore, a well folded protein consists of a combination of rotamers that allow for a tight packing of residues. Achieving accurate packing of residues is a challenging task in protein design. Protein design algorithms, like Rosetta, utilize so-called rotamer libraries. Rotamer libraries were built containing low energy side chain conformations deduced from solved crystal structures of proteins, with the Dunbrack-library being the most commonly used one (Shapovalov and Dunbrack Jr, 2011). This library contains low energy rotamers of every residue for easy access in a discrete form. This means that all torsion angles are grouped in a specific range of, for example, every five degrees. Therefore, this allows for a simplification of the continues range in real life and reduces computational load. During protein design, an algorithm scans this library for optimal packing. But is it really possible to find the global minimum, i.e. the optimal combination of residue combinations? The number of possible rotamer orientations scales exponentially with the number of amino acid residues, i.e. the length of the protein chain. Therefore, optimizing side chain packing is NP-hard (non-deterministic polynomial-time hardness), which means that finding a optimal solution is nearly impossible. The most common solution to tackle this problem are Monte-Carlo algorithms. They utilize randomized starting points and test different combinations by means of probability amplification. Therefore, multiple repetitions are required to reduce the chance of reaching a supoptimal solution (i.e. local minimum) (Mackay, 1998). Another type of algorithms to solve this task is dead-end elimination. Dead-ends are defined as combinations of variables that can be replaced by a better or equivalent one and therefore are never part of an optimal solution. While there are implementations of dead-end elimination algorithms for protein design (Gainza et al., 2013), they often scale poorly with protein length.

1.3 Rosetta

One of the most commonly used software suites for protein design is Rosetta (Leaver-Fay et al., 2011b). It allows the use of various algorithm and protocols for protein design and engineering. Major fields of application include protein structure prediction (DiMaio et al., 2011), scoring of structures (Alford et al., 2017), docking (Sircar et al., 2010), and design (Guntas et al., 2010). The most important part of Rosetta are score functions which are unique combinations of additive terms, whose contributions are tuned by means of specific weights. During this thesis the talaris and the soft-rep scoring function were used (see Table 1.1). The talaris scoring function is the Rosetta default scoring function and aims to optimize all design tasks at once. Choosing the talaris scoring function is never an error, but there might be better fitting scoring function for a specific design challenge. The soft-rep (soft-repulsion potential) score function is an energy function with shorter Van der Waals distances, so that the optimal interaction distance for two atoms is smaller than regularly. This leads to slight clashes during rotamer packing, since rotamers will be selected that are closer than the optimal Van der Waals distance. Therefore, after each packing step a minimization has to be performed to resolve these clashes. Ultimately

this leads to a tightly packed structure with fewer cavities. The downside is the need for a energy minimization after each packing, which increases computational time and still has a chance to leave residues clashing if no space is available. Which scoring function to choose is up to the user, but to get a first idea a native sequence recovery test could be performed. There all resides subjected to the design are mutated to alanine prior the actual design run. Then the design algorithm tries to recover the native sequence of a protein with different scoring functions. The scoring function giving rise to the largest number of recovered (i.e. native) residues, is the best one for the redesign problem at hand. Score functions of Rosetta are mostly physics-based, but also have statistical terms to favor structures that resemble native like proteins with respect to their sequence. These terms are summed up and yield the Rosetta energy score that is given in Rosetta energy units (REU). Therefore, while a design with a lower score resembles a more native structure, a Rosetta energy score does not represent the energetically state of a protein. Moreover, the resulting values are dependent on the score function and are not comparable between different score functions.

For protein design, residues are classified into two groups in dependency of the distance to the design target (e.g. binding site). The design shell contains all residues in close proximity to the design target and residues that are allowed to mutate to different residues. Second shell residues are commonly the residues surrounding the design shell and therefore are more distant to the design target. These residues are not allowed to mutate, but are kept flexible and thus may only change their rotamer. Second shell residues make sure that a newly introduced residue in the design shell fits nicely into the protein and reduce the occurrence of clashes. While both shells allow for residue rotation and therefore side chain flexibility, none of them enables backbone movement.

During a protein design, the backbone of the protein is kept rigid, since a flexible backbone in combination with flexible residues makes the search for a optimal solution impossible. However, proteins need to change their conformation to function properly, e.g. during catalytic reactions or ligand binding. These different conformation are called “poses” in protein design and a fixed backbone design only optimizes a single pose. To counter this limitation, multi-state design (MSD) algorithms have been developed (Leaver-Fay et al., 2011a). There, multiple states are designed at the same time and a solution must be optimal in all states at the same time. A state, for example, is a pose of a protein and MSD allows the combination of many states. The evaluation of one state could be considered a single fixed backbone design, but the solution of one state must also be optimal for all other states. This allows MSD to find a residue allocation that is optimal for all poses under consideration. A possible way to generate different poses for MSD are molecular dynamics (MD) simulations. MD simulations try to mimic the movement of proteins in solvent and allow for full flexibility. These simulations generate poses of the protein, with different side chain conformers and allow for backbone movement, which are integrated into MSD as states. However, states do not necessarily represent poses only containing conformational differences. They also could be poses of the same protein with different ligands

Feature	Talaris	Soft-rep	Description
fa_atr	1.0	0.8	Lennard-Jones attractive between atoms in different residues
fa_rep	0.55	0.657	Lennard-Jones repulsive between atoms in different residues
fa_sol	0.9375	0.648	Lazaridis-Karplus solvation energy
fa_intra_rep	0.005	0	Lennard-Jones repulsive between atoms in the same residue
fa_elec	0.875	0.875	Coulombic electrostatic potential with a distance-dependent dielectric
pro_close	1.25	1.0	Proline ring closure energy and energy of psi angle of preceding residue
hbond_sr_bb	1.17	0.857	Backbone-backbone hydrogen bonds close in primary sequence
hbond_lr_bb	1.17	0.857	Backbone-backbone hydrogen bonds distant in primary sequence
hbond_bb_sc	1.17	0.857	Sidechain-backbone hydrogen bond energy
hbond_sc	1.1	0.857	Sidechain-sidechain hydrogen bond energy
dslf_fa	1.25	1.25	Disulfide geometry potential
rama	0.25	0.25	Ramachandran preferences
omega	0.625	0.625	Omega dihedral in the backbone
fa_dun	0.7	0.569	Internal energy of sidechain rotamers as derived from Dunbrack's statistics
p_aa_pp	0.4	0.915	Probability of amino acid, given torsion values for phi and psi
yhh_planarity	0.625	0.625	A special torsional potential to keep the tyrosine hydroxyl in the plane of the aromatic ring
ref	1.0	1.0	Reference energy for each amino acid. Balances internal energy of amino acid terms

Table 1.1: Rosetta scoring functions. The first column lists the features of the scoring function. During this thesis the talaris and the soft-rep scoring functions were utilized. They differ within their specific weights, which are listed in column two (talaris) and three (soft-rep). The last column gives a brief description of the scoring feature. Each of this terms is calculated on an atomic level per residue, weighted by their respective weight and later combined to a total score for the whole protein.

bound, if, for example, the design goal is to improve the catalytic efficiency of a new reaction B, while retaining the efficiency of the native reaction A. In this case, MSD aims to optimize a binding site that facilitates both reactions A and B. An even more complicated task is negative protein design. There, the MSD algorithm tries to find a AA sequence that enables reaction B and makes reaction A impossible. This is a challenging task because the design algorithms always try to find a well fitting AA sequence and are conceptually not constructed for negative design. Simply changing the sign of the different scores is not sufficient: This "trick" would favor the selection of clashing residues, which cannot form a stable protein. In Chapter 3 a novel method for negative protein design is presented.

The algorithms for protein design are combined within Rosetta to so-called design protocols and within the framework of this thesis, two major design protocols were used. The first one is anchored design (Lewis and Kuhlman, 2011), a Rosetta protocol to create new protein-protein interactions using information from the native interface. A user-defined element of the original interface is transferred to the new one and serves as an anchor to build up a new interaction. In Chapter 2 this protocol is utilized to design a new protein-protein interface. The second protocol used in Chapter 3 is enzyme design, which allows the user to modify a receptor-ligand interaction (Richter et al., 2011). It is the goal of this protocol to optimize the binding of a possibly non-natural ligand by replacing and/or reorienting residues of the binding site. Both protocols were used as MSD applications and up to now a genetic algorithm (GA) was used to find an optimal solution. GAs are inspired by natural selection and utilize operators such as mutation, crossover, and selection to alter sets of design solutions named population. After the generation of novel design solutions, the surviving population is determined by a user defined fitness function and multiple iterations are necessary to reach an optimum (Mitchell, 1998). The search for optima with a GA is quite time consuming and therefore I replaced the GA with a neural network (see Chapter 3). Neural networks are introduced in the following Section.

1.4 Neural Networks

The human brain consists of approximately 86 billion neuronal cells (Azevedo et al., 2009). They are connected with each other through an even greater number of synapses. Neurons exchange information by means of a chemical signal at the synapses. The chemical signal is then converted to an electrical signal that runs along the membrane from the dendrites to the cell body. There, the electrical signal stemming from many dendrites is processed collectively and if a certain threshold is surpassed, the neuron sends a new spike along the axon to the axon terminal which is connected to dendrites of the following neurons. Signal processing in neurons is highly flexible, because the number of signals and their effects varies between neurons: Some signals may have an inhibitory effect and prevent firing of a neuron, whereas other impulses may be excitatory (Lodish et al., 2000). Schematically, a neuron is shown in Figure 1.2 A.

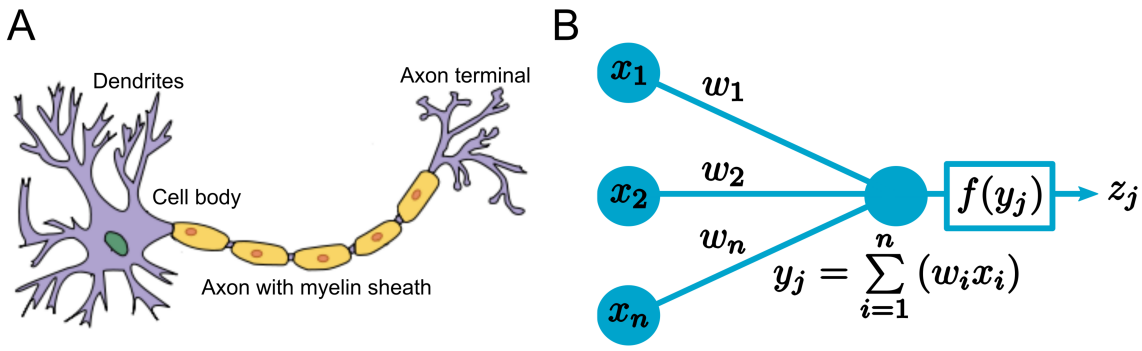

Figure 1.2: Biological and artificial neuron. The left panel (**A**) shows an illustration of a human neuron. The dendrites are connected to many different termini of other neuron's axons and transmit the signal to the cell body. There the signals of many dendrites is combined and if a certain threshold is exceeded, it fires a signal down the axon. An axon is coated with myelin to increase the speed of the signal transmission and ends with the axon terminal, which is connected to dendrites of many different neurons via synapses. The effect of a signal reaching a new neuron can be excitatory (positive) or inhibitory (negative). The right panel (**B**) pictures its artificial counterpart. It has a fixed number of n inputs and one output z_j. To compute the output value, each of the input signals x_i is multiplied with a specific weight w_i and the sum y_i is calculated. An activation function $f()$ is then used to convert y_i to the output signal z_j.

Artificial neural networks (NNs) which were first proposed by McCulloch and Pitts (1943), are computing systems that mimic their biological counterpart. The smallest unit of an NN is called artificial neuron; a typical architecture is depicted in Figure 1.2 B. Analogously to a biological neuron, it receives several input signals. Each input signal is multiplied with a specific weight factor; these values are then summed up and fed into an activation function which determines the output signal. Commonly, neurons are organized in layers which may perform different transformations of their input signals. Usually, several layers constitute an NN which has then an input and an output layer and several (hidden) processing layers in between. NNs are highly flexible signal processing systems because the user can choose different types of neurons, vary the number of inputs, and choose an activation function according to his needs. Moreover, the number of neurons per layer and the number of layers can also be adjusted.

A further and important element of the NNs' flexibility is the mode of parametrizing the weights w_i. Commonly, the signal specific weights are not fixed beforehand, but have to be adjusted during a training phase. NNs can either be trained by supervised or unsupervised learning. The NN used within this thesis is trained via supervised learning and therefore requires for a labeled training data set. The training data consists of a pair of vectors, one containing the input objects and the other containing the desired output. The NN estimates the output, given the input vector and the weights are then adjusted for the error in the output. The output correction is commonly done via an algorithm called backpropagation. Error-backpropagation is a special case of a gradient descent based on a mean squared error estimation of the predicted output compared to the real output of the training data. It then computes the gradient of a loss function with respect to the weights along the NN, which are then adjusted accordingly.

However, a gradient descent with backpropagation will only reach the closest local minimum and not the global minimum. Although there are additional terms added to the gradient descent, like a momentum, it is still not guaranteed to find the global minimum. A momentum term keeps the gradient moving in the same direction to avoid oscillating behavior. The gradient is further affected by the learning rate, which determines the step size. A small step size ensures to even find a narrow minimum, but complicates the crossing of similar scored regions (i.e. saddle points). To make the learning more flexible an additional decay factor is added. The decay reduces the effects of previous updates (Kröse and van der Smagt, 1993).

NNs received an ever-increasing interest over the last two decades in protein bioinformatics (Ke Chen, 2012). A very attractive property of NNs is their high speed after training, thus NNs are frequently used to prediction certain properties (Ke Chen, 2012). Typical applications are the following ones:

1. The prediction of protein characteristics, like their localization in the organism.
2. The identification of ligand binding sites, the specific ligand binding residues, and the characterization of the bound ligands.
3. The prediction of structure-related properties, such as the secondary structure, residue contacts, structural domains, and β-turns.
4. The prediction of the protein three dimensional structure in full detail, i.e. at the atomic level.

A major breakthrough towards the quality of structure prediction was achieved during the 13th Critical Assessment of Techniques for Protein Structure Prediction (CASP). There, an expensively trained network outperformed all existing method for structure prediction (Senior et al., 2020). NNs have been used less successfully for structure prediction earlier, the extraordinary performance of the novel approaches is owing to the utilization of deep neural networks. Meanwhile several alternative algorithms have been developed, among them is trRosetta (Yang et al., 2020) which combined Rosetta and an NNs.

1.5 Aim and Scope of this Work

The aim of the first part of this thesis was to redesign the interface of PabA in such a manner that the binding of the native interaction partner PabB is lost and the exclusive binding of TrpEx is achieved. From previous work the presence of an interface add-on in TrpEx was already known (Plach et al., 2017), which ensures the specific binding of TrpEx to TrpG. My goal was to use `Rosetta:MSF` as an MSD approach and utilize the anchored design protocol to transfer the TrpG region opposite of the TrpEx interface add-on to the design target PabA (Löffler et al., 2017). This creates a novel protein-protein interface, which is then further optimized.

Rosetta:MSF is a framework that allows the use of Rosetta protocols in a multi-state environment (Löffler et al., 2017). It utilizes a GA to optimize sequences to fit multiple states at once and has outperformed the single-state protocols in terms of the native sequence similarity recovery rate (NSSR). The NSSR describes the protocol's ability to recover native residues or similar ones and was benchmarked on an enzyme design benchmark data set. For the benchmark data set, 16 enzymes with bound ligands were extracted from the PDB, their ligand removed and all design shell residues mutated to alanine. Afterwards a MD simulation was performed to sample natural conformations for the MSD approach. Rosetta:MSF was then benchmarked on the recovery of the native design shell residues and on the number of iterations required to achieve this NSSR. The benchmark results already indicated that Rosetta:MSF improved the NSSR rate and produced better scored structures but required a large number of iterations (Löffler et al., 2017). This is mostly due to the GA, since it is quite slow in finding an optimal solution.

To tackle this limitation, I combined NNs with Rosetta:MSF to create the novel protocol Rosetta:MSF:NN during the second part of this thesis. Rosetta:MSF:NN integrates an NN for residue scoring into a Rosetta design protocol. During the training phase the NN learns the problem-specific energy space. For this training, a set of sequences and the corresponding Rosetta scores are needed. After training, the NN replaces the time-consuming Rosetta function and assesses candidate sequences. To validate this new protocol, I benchmarked it again on the *MD_EnzBench* data set already used for Rosetta:MSF. There, it outperformed Rosetta:MSF in speed, produced better scored sequences and scored much more residue combinations.

1.6 Guide to the Following Chapters

Chapter 2 consists of the manuscript entitled **Reprogramming the Specificity of a Protein Interface by Computational and Data-driven Design**. It describes the design and the experimental validation of a novel protein-protein interface by means of Rosetta:MSF:Anchored Design and a data-driven approach. Both approaches produced PabA variants that were able to bind to TrpEx and no longer bind the native partner PabB.

In Chapter 3 the novel protocol Rosetta:MSF:NN is introduced in the manuscript entitled **Rosetta:MSF:NN Boosting Multi-state Computational Protein Design with a Neural Network**. It utilizes an NN to learn the prediction of the Rosetta score based on only the sequence of a design task. This new protocol is flexible and applicable to many Rosetta protocols and allows for MSD. Since every design run produces a fully trained NN, Rosetta:MSF:NN even allows to combine multiple designs after completion of the training phase. This opens a new way for negative protein design, because the design algorithms are used to train the NN with positive designs and the negative design is then performed by the NN. Rosetta:MSF:NN outperformed the GA on all benchmark cases, producing lower scored structures and requir-

ing for fewer iterations. Furthermore, it allows one to predict the Rosetta score of much more sequences.

Chapter 2

Reprogramming the Specificity of a Protein Interface by Computational and Data-driven Design

Regina Hertle[1], Julian Nazet[1], Florian Semmelmann[1], Sandra Schlee, Franziska Funke, Rainer Merkl*, and Reinhard Sterner*

Institute of Biophysics and Physical Biochemistry
University of Regensburg, D-93040 Regensburg, Germany

[1] These authors contributed equally to this work.
* Corresponding authors:
Rainer Merkl: +49-941 943 3086; Rainer.Merkl@ur.de
Reinhard Sterner: +49-941 943 3015; Reinhard.Sterner@ur.de

Lead contact: Reinhard Sterner: Reinhard.Sterner@ur.de

2.1 Summary

The formation of specific protein complexes in a cell is a non-trivial problem given the co-existence of thousands of different polypeptide chains and a limited number of protein interface geometries. A particularly difficult case are two glutamine amidotransferase complexes (anthranilate synthase, AS and aminodeoxychorismate synthase, ADCS) which are composed of homologous pairs of synthase and glutaminase subunits. We have attempted to identify discriminating interface residues of the glutaminase subunit TrpG from AS which are responsible for its specific interaction with the synthase subunit TrpEx and prevent binding to the closely related synthase subunit PabB from ADCS. For this purpose, TrpG-specific interface residues were grafted into the glutaminase subunit PabA from ADCS by two different approaches, namely

a computational and a data-driven one. Both approaches resulted in PabA variants that bound TrpEx with higher affinity than PabB. Hence, we have accomplished a complete reprogramming of protein-protein interaction specificity that provides insights into the evolutionary adaptation of protein interfaces.

Keywords: anthranilate synthase, aminodeoxychorismate synthase, glutamine amidotransferases, grafting, interface add-on, protein design, protein-protein interactions, protein specificity switch.

2.2 Introduction

A large fraction of the proteins found in nature belongs to compositionally well-defined homomeric or heteromeric complexes. For example, a survey of the oligomerization state of *Escherichia coli* proteins revealed that only 20% of the polypeptides are existing as monomers whereas 80% are forming defined complexes with other proteins. Approximately 80% of these complexes are homo-oligomers and 20% are hetero-oligomers (Goodsell and Olson, 2000; Levy et al., 2008). Despite the risk of the emergence of millions of non-physiological protein assemblies in the crowded environment of cells, highly specific protein complexes are formed. This is all the more astonishing as the number of different protein interface geometries has shown to be restricted to 1000 (Gao and Skolnick, 2010; Jones and Hore, 1991; Marsh and Teichmann, 2015). It is therefore important to understand how protein interaction specificity is guaranteed and how non-physiological protein-protein cross talk is prevented under such constraints.

The avoidance of non-productive assemblies is most difficult in cases where proteins that (i) share the same fold and (ii) possess similar interface geometries compete for the same interaction partner. To disclose the evolutionary driving forces that confer specificity for one of these difficult cases, we exemplarily analyzed complex formation and specificity in glutamine amidotransferases (GATases). GATases form a large enzyme family whose members are responsible for the incorporation of nitrogen within numerous metabolic pathways (Mouilleron and Golinelli-Pimpaneau, 2007; Raushel et al., 1999). GATases are bi-enzyme complexes, consisting of a synthase and a glutaminase subunit. The activities of the two subunits are mutually coupled in a two-fold manner: (i) Substrate binding to the synthase subunit allosterically induces glutamine hydrolysis to glutamate and nascent ammonia at the glutaminase subunit; (ii) ammonia is then transported through an intermolecular channel to the active site of the synthase subunit where it reacts with the "waiting" substrate to different reaction products. The glutaminases can be categorized into two classes according to their folds and active site compositions: Class I glutaminases show an $\alpha\beta$-hydrolase fold and a catalytic triad Cys-His-Glu (Ollis et al., 1992) whereas class II show an Ntn-hydrolase fold and a catalytic N-terminal Cys (Brannigan et al., 1995).

Compared to the glutaminases, the synthase subunits of the various GATases are structurally quite diverse. One exception, however, are the two GATases 4-amino-4-deoxychorismate synthase (ADCS) and anthranilate synthase (AS) that both use chorismate and glutamine to catalyze the committed steps in the biosynthetic pathways leading to tryptophan and folate, respectively. In ADCS and AS not only the two class I glutaminases but also the two synthases are homologs and possess highly similar three-dimensional structures (Morollo and Eck, 2001; Parsons et al., 2002; Semmelmann et al., 2019b). Hence, ensuring interaction specificity in the ADCS and AS pair of class I GATases is a particular challenge. Remarkably, in most species, the glutaminase PabA functionally interacts with both, the ADCS subunit PabB and the AS subunit TrpE (Plach et al., 2017). Consequently, tryptophan and folate biosynthesis cannot be regulated independently at the glutaminase level. This issue has been resolved in some "modern" γ-proteobacteria by evolving an additional AS glutaminase subunit TrpG, which specifically interacts with a modified version of the AS synthase subunit. We refer to this modified version of TrpE as TrpEx (**x** stands for e**x**tended) (Figure 2.1 A).

TrpEx, in comparison to TrpE and PabB, contains an insertion in the interface towards TrpG (Plach et al., 2017). We hypothesized that the "interface add-on" of TrpEx guarantees specific complex formation with TrpG and avoids the formation of non-physiological complexes with PabA (Figures 2.1 B and 2.1 C). A phylogenetic analysis has indicated that TrpG evolved from PabA by gene duplication and speciation (Plach et al., 2017). In the course of this hypothetical evolutionary pathway from PabA to TrpG, mutations must have occurred in PabA that allowed for its interaction with TrpEx and at the same time prevented its binding to PabB and TrpE. To retrace these putative mutational events in the laboratory, we have replaced residues of PabA with TrpG-specific residues that were assumed to interact with the interface add-on of TrpEx. The resulting variant PabA* formed a strong complex with TrpEx, while it retained significant affinity for its original interaction partner PabB (Figure 2.1 D). Based on these findings, the goal of the present study was to generate PabA variants that exclusively interact with TrpEx and are no longer able to bind to PabB or TrpE (Figure 2.1 E). To achieve this complete conversion of interaction specificity, we used two different approaches, namely interface design based on the Rosetta software suite and, alternatively, a data-driven approach that exploits the differential conservation of interface residues in PabA and TrpG.

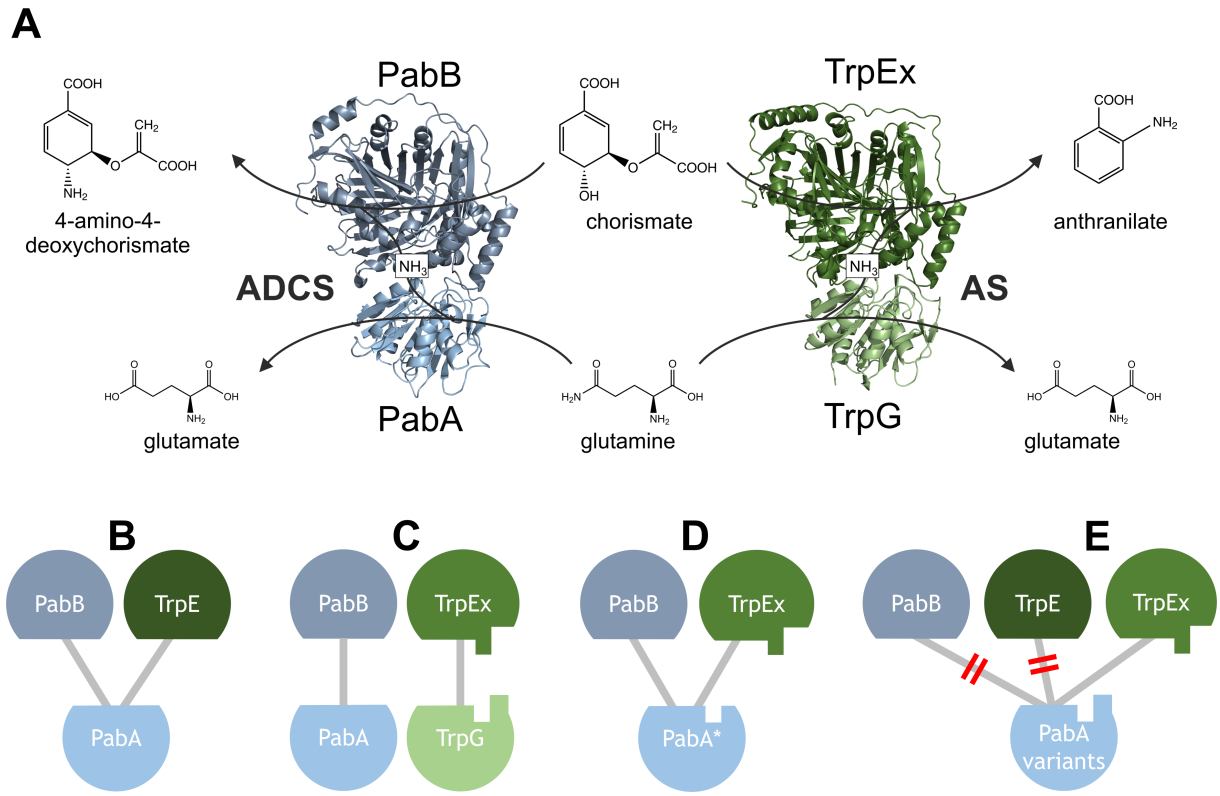

Figure 2.1: Structures, reactions, and subunit interaction specificities within the ADCS and AS complexes. (A) Structures and reactions as observed in γ-proteobacteria. Left: Homology model of ADCS from *Escherichia coli* (Semmelmann et al., 2019b); the glutaminase subunit PabA is shown in light blue, the synthase subunit PabB is shown in dusty blue. Right: Crystal structure of AS from *Salmonella typhimurium* (PDB-ID: 1i1q); the glutaminase subunit TrpG is shown in light green, the synthase subunit TrpEx is shown in dark green. Both ADCS and AS use chorismate and glutamine to produce glutamate and 4-amino-4-deoxychorismate or anthranilate, respectively. (B) In most bacteria, the generalist glutaminase PabA interacts with both PabB and TrpE. (C) In γ-proteobacteria, a dual glutaminase system exists where the glutaminase PabA selectively interacts with PabB whereas the glutaminase TrpG selectively interacts with TrpEx (cf. A). TrpEx contains an additional structural element at the protein-protein-interface (referred to as interface add-on) (Plach et al., 2017). Phylogenetic analysis indicates that the dual glutaminase system shown in (C) has evolved from the single glutaminase system shown in (B). (D) In a data-driven approach, the PabA* variant has been designed, which structurally and functionally interacts with both PabB and TrpEx (Plach et al., 2017). (E) The goal of this work was to generate PabA variants that resembles TrpG by exclusively interacting with TrpEx but not with PabB or TrpE.

2.3 Results

Designing a PabA interface that accommodates the TrpEx add-on by means of computational grafting

We have previously attempted to reconstruct the putative conversion of the generalist glutaminase PabA into the specialized glutaminase TrpG. For this purpose, we have generated the variant PabA* that formed a functional complex with both PabB and TrpEx (Figure 2.1 D). The binding properties of PabA* made us presume that a structural element that is complementary to the TrpEx add-on is required in the PabA interface. This element should prevent binding to PabB and TrpE and ensure exclusive binding to TrpEx (Figure 2.1 E). A suitable method to introduce in silico a natural binding epitope is grafting (Schreiber and Fleishman, 2013). The broad functionality of the well-proven software suite Rosetta (Rohl et al., 2004) assists the user in addressing various state-of-the art research challenges by means of different protocols. A Rosetta protocol that supports grafting is "anchored design" (Lewis and Kuhlman, 2011) and we utilized `Rosetta:MSFAnchoredDesign` (Löffler et al., 2017), because this implementation supports multi-state design. We opted for a multi-state design approach, because we considered the representation of a protein-protein complex by means of alternative poses less susceptible to modeling errors of the design phase. Thus, we applied this protocol in combination with an "anchor" consisting of 25 successive TrpG residues that interact with the TrpEx add-on (Plach et al., 2017). Initially, a chimeric sequence PabA-**CA** was created that contained the TrpG-specific **c**omplement of the TrpEx **a**dd-on at residue positions 6 - 30 of PabA (Figure 2.2).

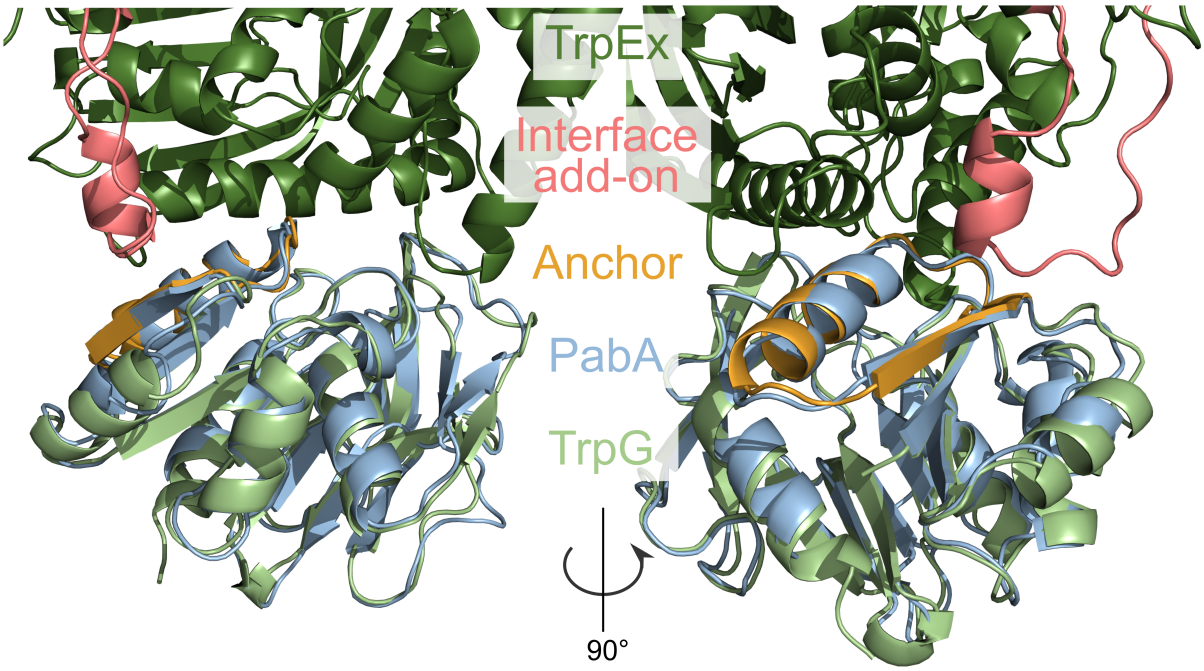

Figure 2.2: Orientation of subunits and localization of critical interface elements. Arrangement of TrpEx (dark green) and TrpG (light green) as observed in the native AS complex (PDB-ID 1i1q), localization of the TrpEx interface add-on (residues 104 - 123, light red) and the TrpG "anchor" (residues 7 - 31, orange). The orientation of PabA (light blue) resulted from a superimposition with TrpG.

This chimeric sequence was subjected to the genetic algorithm of the multi-state design protocol `Rosetta:MSFAnchoredDesign` that evolved a set of 239 sequences by improving their fitness with respect to two states, i.e. PabA:TrpEx configurations. For this design, all PabA residues contributing to the PabA:TrpEx interface were allowed to be replaced by Rosetta. Whereas the initial PabA-CA sequence differed in 13 residues from ecPabA, the **d**esigned sequence PabA-CA**D** contained 17 mutations (Table 2.1). The localization of the mutated interface residues of PabA-CAD is shown in Figures 2.3 A and B.

PabA	PabA*	PabA-CA	PabA-CAD	PabA*+	PabA**
Y8		**I**	K	I	I
D9		**D**	A		
Y16		**A**	A		
Q17	D	**D**	D	D	D
Y18	Q	**Q**	C	Q	Q
F19		**L**	L		
C20	R	**R**	R	R	R
E21		**T**	T	T	T
L22		**N**	N		
A24		**H**	H		
D25		**N**	N		
L27		**V**	V		
V28		**I**			
K29	Y	**Y**	Y	Y	Y
R30		**R**	D		
D32	Q		C	H	H
C54			G	G	G
R95				Q	Q
K98				E	E
M100			E		
T104				A	A
T125					A

Table 2.1: Overview of PabA variants. The first column denotes the wild-type residue in ecPabA, the other columns denote the corresponding residue in the listed PabA variant (white space for unchanged residue). The variant PabA* has been generated and characterized previously (Plach et al., 2017). The bold residues of PabA-CA are those of the anchor taken from stTrpG. PabA residues marked with a dusty blue background form a hydrogen bond across the native ecPabA:ecPabB interface. Residues of the PabA variants marked with a light green (dark green) background form in the specific complex models a hydrogen bond (a hydrogen bond plus a salt bridge) across the interface with stTrpEx. Interface positions possessing different residues in PabA-CAD and PabA** are printed in red.

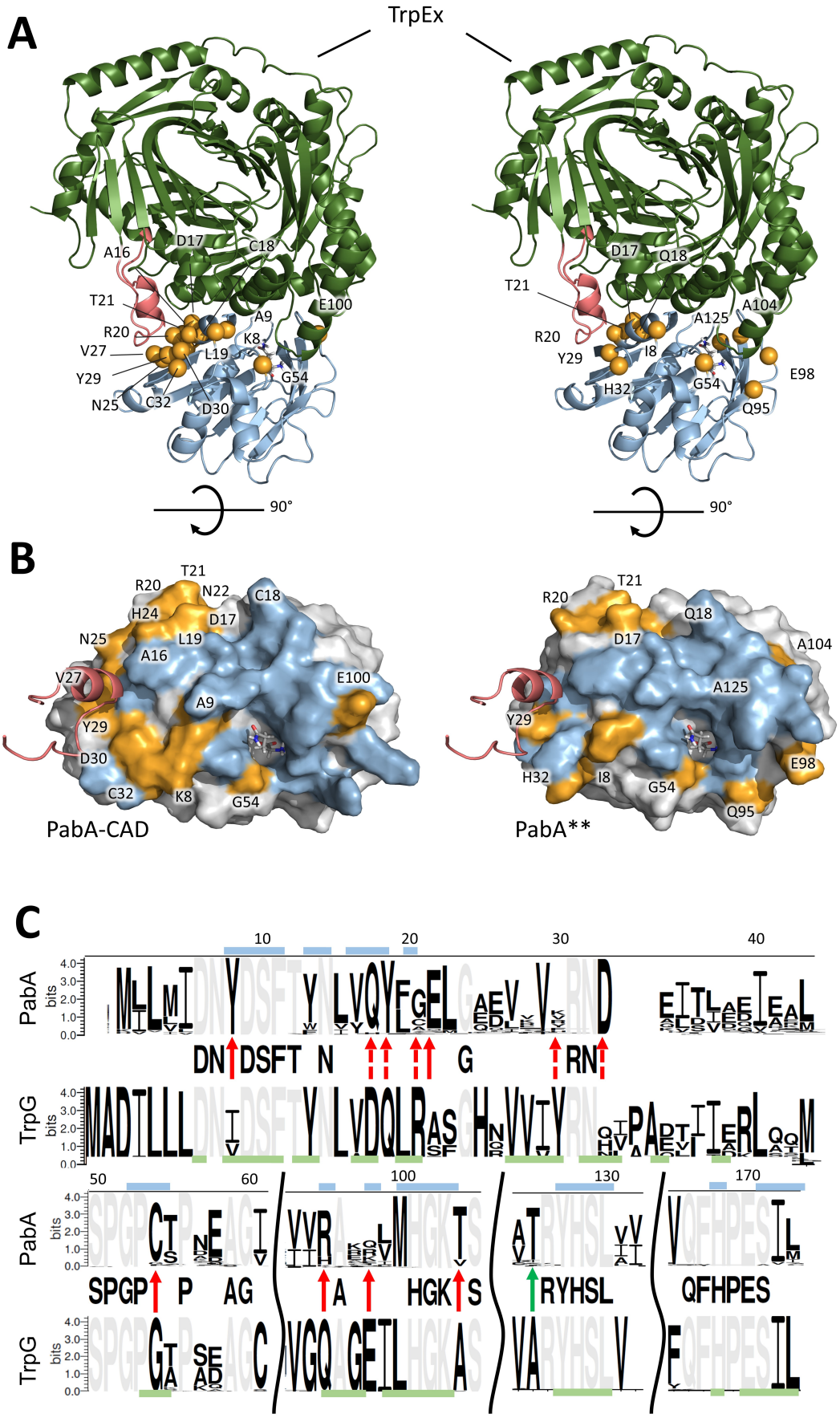
A
TrpEx
A16
D17
C18
T21
A9
E100
R20
K8
V27
L19
Y29
G54
N25
C32
D30
90°
D17
Q18
T21
A125
A104
R20
I8
Y29
E98
H32
G54
Q95
90°
B
T21
R20
N22
C18
H24
D17
N25
L19
A16
V27
E100
A9
Y29
D30
C32
K8
G54
PabA-CAD
T21
R20
Q18
D17
A104
A125
Y29
E98
H32
I8
G54
Q95
PabA**
C
10
20
30
40
PabA
bits
4.0
3.0
2.0
1.0
0.0
DN DSFT N G RN
TrpG
50
60
100
130
170
SPGP P AG
A HGK S
RYHSL
QFHPES

Figure 2.3: Localization of mutated PabA interface residues and design strategy for the conversion of PabA into PabA based on the differential conservation of interface residues.**
(A) Side view of the TrpEx:PabA-CAD complex (left panel) and of the TrpEx:PabA** complex (right panel) in cartoon representation. TrpEx is colored in dark green and the two PabA variants in light blue. The interface add-on of TrpEx is shown in red and mutated PabA residues are shown as orange balls. The structure of PabA and the pose of glutamine shown as a stick model inside the binding pocket were modeled on the basis of a published PabA structure (Semmelmann et al., 2019b).
(B) Top-down view of the PabA-CAD interface (left panel) and of the PabA** interface (right panel) in surface representation. Mutated residues are shown in orange and the remaining residues of the PabA interfaces in light blue. Residue A125 is a buried residue and not located on the PabA** surface.
(C) Differential sequence logo indicating the conservation of PabA and TrpG residues. Residues that are conserved (threshold = 95%) in both PabA and TrpG are indicated in gray and listed on the central line. Red arrows indicate replacements of PabA residues by TrpG interface residues, yielding PabA*+. The five mutations represented by a red dashed line are those of the previously designed variant PabA*. The green arrow indicates the additional mutation present in PabA**. Horizontal bars indicate interface residues of PabA (light blue) and of TrpG (light green). We classified a residue as an interface residue, if its solvent accessible surface is decreasing upon complex formation. Residues are numbered according to ecPabA.

Designing a PabA interface that exclusively binds to TrpEx by means of a data-driven approach

In order to assess the structural similarity of PabA and TrpG, we superposed corresponding Cα atoms from the relaxed conformations of an ecPabA model (based on PDB-ID 6qur) and the stTrpG structure (PDB-ID 1i1q). Remarkably, the root-mean-square-deviation (RMSD) value was not more than 0.7 Å. This low value confirms an excellent match of the PabA and TrpG backbones and, as a consequence, a highly similar arrangement of the corresponding glutaminase residues at the two interfaces. Based on this finding, as an alternative to the fully automated grafting approach of Rosetta, we utilized a data-driven protocol to choose residue substitutions in PabA that might change interaction specificity from PabB to TrpEx. To this end, a differential sequence logo based on 487 PabA and 32 TrpG sequences was created by means of an in-house program (Figure 2.3 C). Our goal was to identify residue positions whose amino acid occupancy differed significantly between the PabA and TrpG interfaces. Starting from ecPabA, we initially created the variant PabA^{*+}: The two strictly conserved PabA interface residues Y8 and D32 were replaced by the most abundant TrpG residues I and H, respectively. Additionally, fully conserved TrpG interface residues replaced the corresponding residues of PabA. It is known that the PabA mutation T125A causes a 10-fold acceleration of the basal glutaminase activity, most likely due to a rearrangement of the hydrogen bond network near the catalytic triad (Semmelmann et al., 2019b). Thus, we created the variant PabA**, which contains an additional mutation T125A compared to PabA^{*+} (Figure 2.3 C; Table 2.1). The localization of the mutated interface residues of PabA** is shown in Figures 2.3 A and 2.3 B.

Production of designed PabA variants

The genes coding for the different PabA variants were introduced into pET21a-*Bsa*I via *Bsa*I-cloning as described (Rohweder et al., 2018). The genes were expressed in *E. coli*, and the resulting proteins were purified from the soluble fraction of the cell extract using immobilized metal ion affinity chromatography (IMAC) and preparative size exclusion chromatography (SEC). The wild-type synthases ecPabB and ecTrpEx from *E. coli* and ppTrpE from *Pseudomonas putida* as well as the wild-type glutaminases ecPabA and stTrpG from *Salmonella typhimurium*, which was used instead of the ecTrpGD fusion protein, were available from previous studies (Plach et al., 2017)(Semmelmann et al., 2019a).

Qualitative evaluation of the interaction specificity of designed PabA variants by analytical SEC

The ability of the different PabA glutaminase variants to form stable complexes with PabB/TrpE and TrpEx synthases was tested qualitatively by analytical SEC. Three SEC runs were performed

for each complex of interest: one with the isolated glutaminase subunit, one with the isolated synthase subunit, and one with an equimolar mixture of glutaminase and synthase subunits. Complex formation was indicated by the appearance of a new, early eluting complex peak and the disappearance of the glutaminase and synthase peaks in the glutaminase-synthase sample.

As expected, complex formation was observed with native interaction partners ecPabA + ecPabB (Figure 2.4 A, left panel) and stTrpG + ecTrpEx (Figure 2.4 A, right panel). No complex formation was observed for the combinations ecPabA + ecTrpEx (Figure 2.4 B, left panel) and stTrpG + ecPabB (Figure 2.4 B, right panel). These samples served as positive or negative controls, respectively. Both PabA-CA and PabA-CAD showed detectable but incomplete

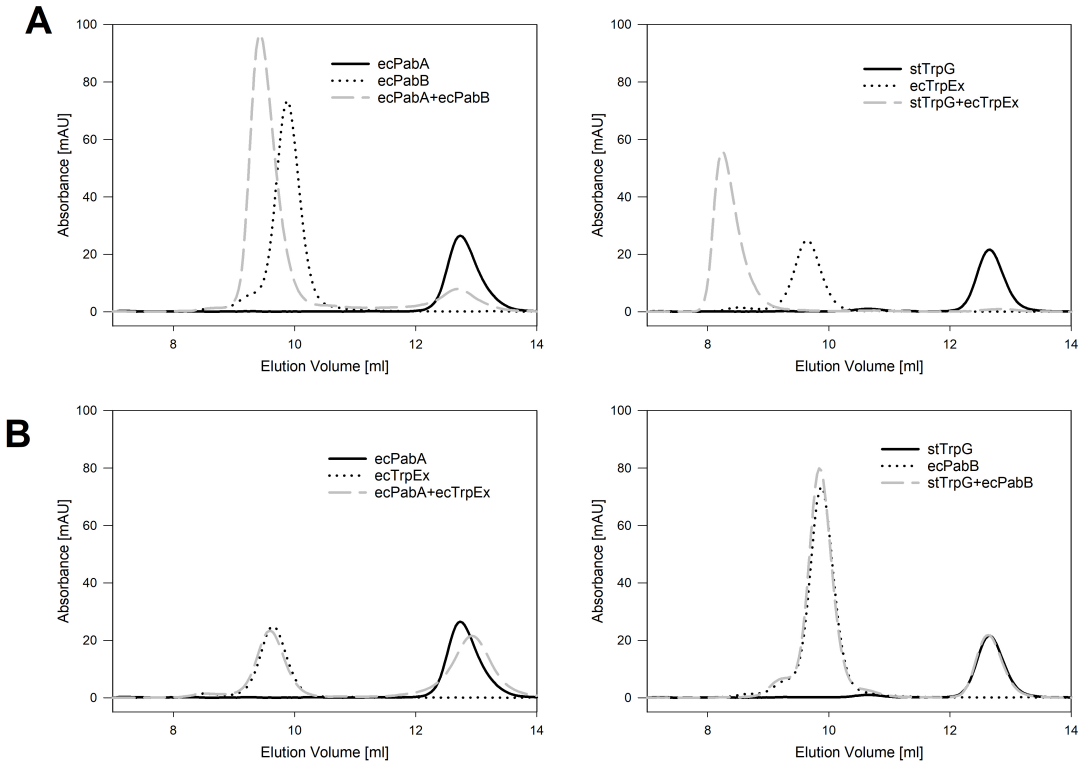

Figure 2.4: SEC analysis of complex formation between ecPabA/stTrpG and ecPabB/ecTrpEx. (A) Positive controls: The elution profiles of naturally interacting glutaminase-synthase pairs, ecPabA:ecPabB and stTrpG:ecTrpEx, are shown. The shifts of the early eluting peak observed with the mixtures indicate complex formation between glutaminases and synthases.
(B) Negative controls. The elution profiles of non-interacting glutaminase-synthase pairs, ecPabA:ecTrpEx and stTrpG:ecPabB are shown. The absence of a shift of the early eluting peak in the mixtures indicates the absence of complex formation.

complex formation with ecTrpEx, as free glutaminase and synthase subunits were detectable in the mixture in addition to the complex peak (Figure 2.5 A and 2.5 B, upper panels). Moreover, neither PabA-CA nor PabA-CAD formed a detectable complex with ecPabB (Figure 2.5 A and 2.5 B, middle panels) or ppTrpE (Figure 2.5 A and 2.5 B, lower panels). Taken together, these results indicate that the synthase interaction specificity of ecPabA has been partially converted

by the interface mutations present in PabA-CA and PabA-CAD. Both PabA*$^{+}$ and PabA**,

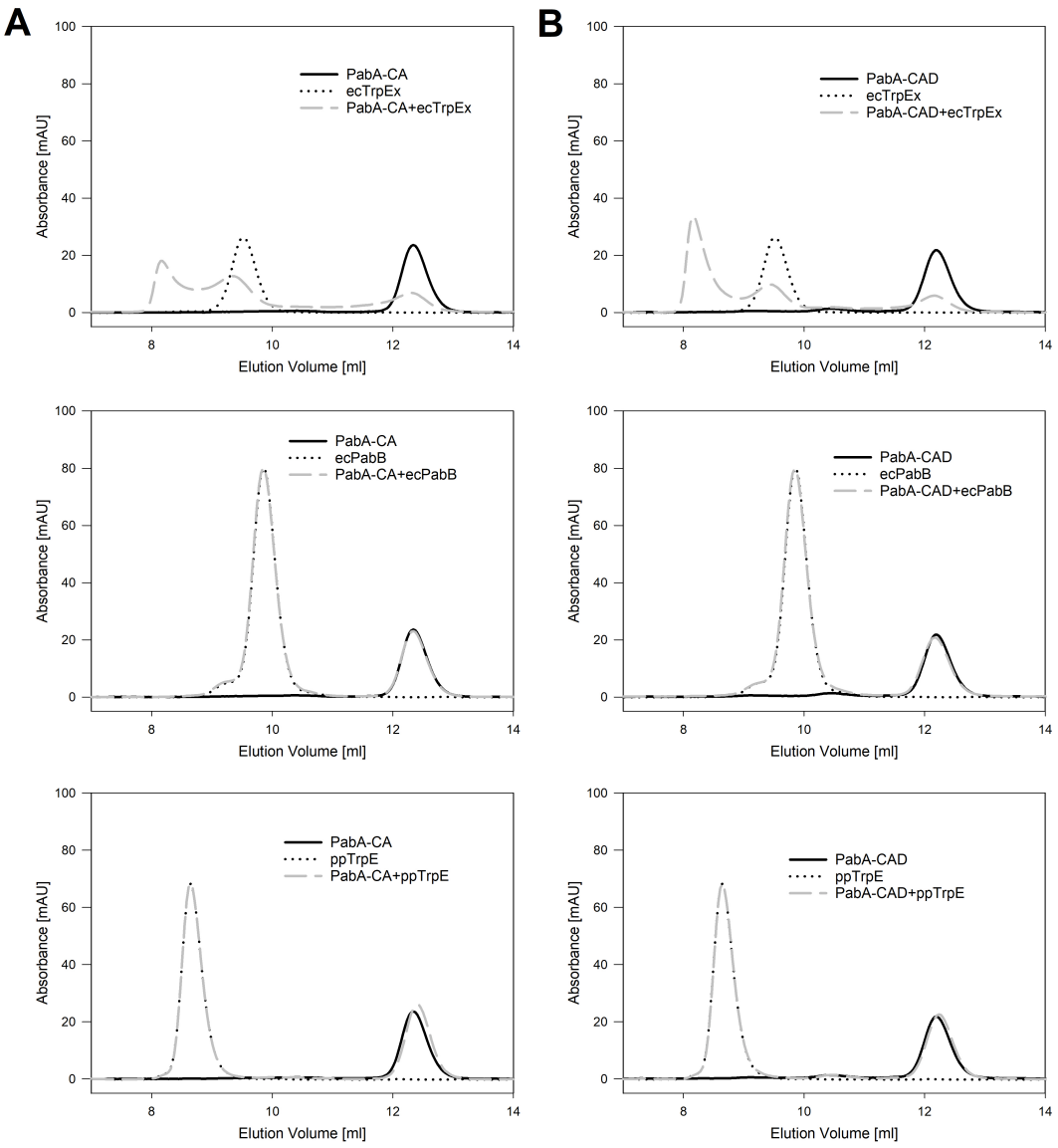

Figure 2.5: SEC analysis of complex formation between PabA-CA (A) and PabA-CAD (B) with ecTrpEx, ecPabB, and ppTrpE. Elution profiles for designed PabA variants PabA-CA (A) and PabA-CAD (B) in combination with the synthases ecTrpEx, ecPabB, and ppTrpE are shown. The profiles indicate weak interactions of PabA-CA and PabA-CAD with ecTrpEx. There is no indication for interactions with ecPabB or ppTrpE.

which were generated on the basis of a differential sequence logo, formed complexes with ecTrpEx, however with slightly different propensities: The elution profile of PabA*$^{+}$ with ecTrpEx indicated incomplete complex formation as small additional peaks for the unbound glutaminase and synthase subunits were detectable in the mixture (Figure 2.6 A, upper panel). In contrast SEC analysis of PabA** combined with ecTrpEx indicated quantitative complex formation (Figure 2.6 B, upper panel). Remarkably neither PabA*$^{+}$ nor PabA** formed a detectable complex

with either ecPabB (Figures 2.6 A and 2.6 B, middle panels) or ppTrpE (Figures 2.6 A and 2.6 B, lower panels). Taken together, these results suggest that the synthase interaction specificity of PabA has been completely converted to a large extent by the interface mutations included in PabA*$^{+}$ and PabA**.

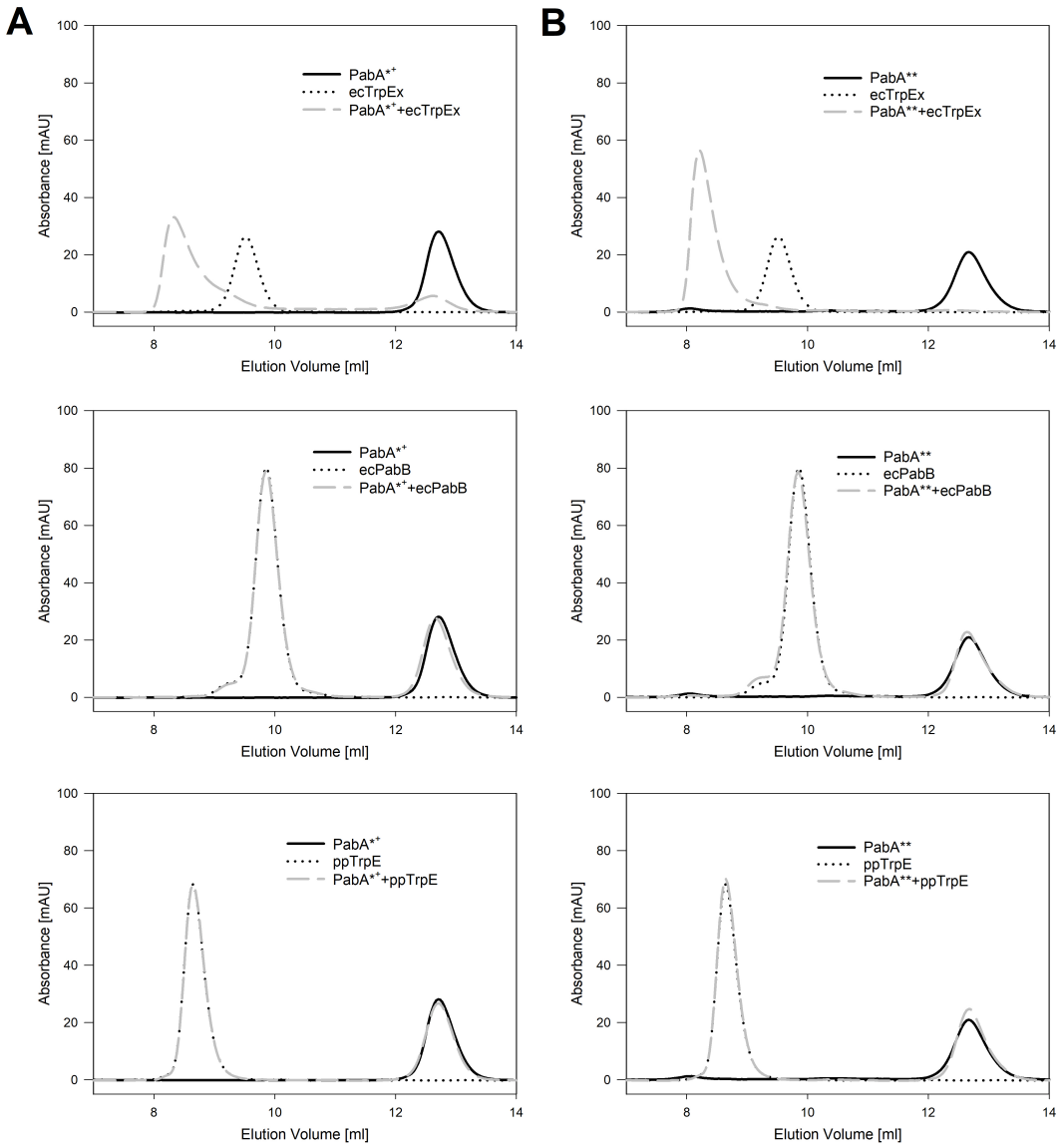

Figure 2.6: SEC analysis of PabA*$^{+}$ (A) and PabA (B) variants with ecTrpEx, ecPabB, and ppTrpE.** Elution profiles for designed PabA variants PabA*+ (A) and PabA** (B) in combination with the synthases ecTrpEx, ecPabB, and ppTrpE are shown. The profiles indicate complex formation of PabA*+ and PabA** with ecTrpEx. There is no indication for interactions with ecPabB or ppTrpE.

Quantification of the interaction specificity of designed PabA variants by activity titrations

The specific structural interactions of PabA with PabB and of TrpG with TrpEx are reflected by the specific stimulation of the glutaminase activities through the respective synthase (Plach et al., 2017). Based on this finding, activity titrations were performed to obtain a quantitative measure for the interaction specificity switch of the different PabA variants. For this purpose, the rates of glutamine hydrolysis by the PabA variants were monitored as a function of varied concentrations of ecPabB or ecTrpEx (Figure 2.7).

The analysis of the activation data allowed for the determination of apparent dissociation constants ($K_D{}^{app}$) and apparent maximum turnover rates ($k_{cat}{}^{app}$), which are listed in Table 2.2. To obtain a measure for the interaction specificities of the different PabA variants, the quotient $K_D{}^{app}$(ecPabB)/$K_D{}^{app}$(ecTrpEx) was calculated. As expected, the quotient was «1 for the glutaminase activity of ecPabA, indicating a high specificity for ecPabB, and »1 for stTrpG, indicating a high specificity for ecTrpEx. With the exception of PabA-CA, for all PabA variants a value of >1 was calculated, corresponding to a marked specificity switch from ecPabB to ecTrpEx. Importantly, the ratios for the newly designed variants Pab-CAD (7.3), PabA*$^{+}$ (9.1) and PabA** (3.9) surpass the value of PabA* (2.3) (Table 2.2). The activity titration experiments were performed in the presence of saturating concentrations of glutamine. Therefore, the obtained maximum rates allow for the determination of apparent turnover numbers ($k_{cat}{}^{app}$) in the presence of saturating concentrations of ecPabB and ecTrpEx, respectively. As shown previously, the wild-type glutaminases are stimulated much stronger by their cognate interaction partners compared to their non-cognate ones. All newly designed variants are stimulated to a similar extent by ecPabB and ecTrpEx, with stimulation levels comparable to cognate glutaminase-synthase pairs. An exception is PabA-CAD where $k_{cat}{}^{app}$-values and thus stimulation levels by both synthases are significantly lower (Table 2.2).

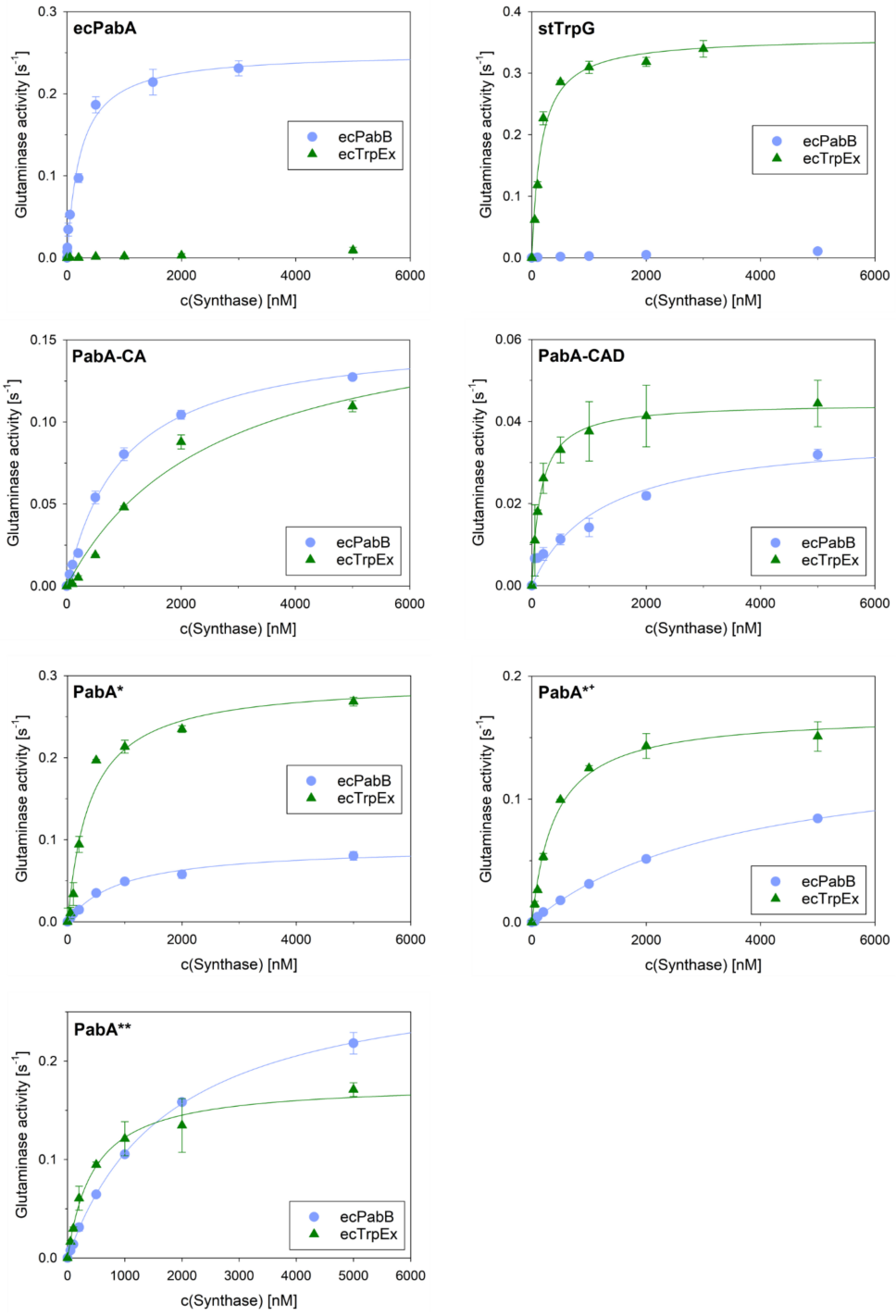

Figure 2.7: Activity titrations of ecPabA and variants with the synthase ecPabB or ecTrpEx. The rate of glutamine hydrolysis by ecPabA and denoted variants is plotted as function of [ecPabB] or [ecTrpEx]. Error bars indicate the standard deviation observed in two separate experiments. Solid lines represent the best fit to hyperbolic equations.

Glutaminase	ecPabB (A)		ecTrpEx (B)		K_D^{app} (A) / K_D^{app} (B)
	K_D^{app} (nM)	k_{cat}^{app} (s^{-1})	K_D^{app} (nM)	k_{cat}^{app} (s^{-1})	
ecPabA	228	0.25	>5000	not determ.	≪1
stTrpG	>5000	not determ.	162	0.36	≫1
PabA*	929	0.092	396	0.29	2.3
PabA-CA	1036	0.16	2602	0.174	0.40
PabA-CAD	1200	0.037	165	0.044	7.3
PabA*+	3669	0.15	403	0.17	9.1
PabA**	1790	0.30	453	0.18	3.9

Table 2.2: Stimulated glutaminase activities of ecPabA, stTrpG, and PabA variants. The first column lists the specific glutaminase subunit. The second and the third column show the K_D^{app} and the k_{cat}^{app} values for the interaction with the ecPabB synthase subunit. In contrast, the fourth and the fifth column show the K_D^{app} and the k_{cat}^{app} values for the interaction with the ecTrpEx synthase subunit. In the sixth column the ratio of the K_D^{app} values of the synthase subunits is displayed. A higher value describes a higher specificity of the glutaminase subunit for ecTrpEx.

Computational analysis of interface properties

To gain insights into the residue interactions that distinguish complex stability of our designs, we utilized the PISA service of EMBL-EBI to analyze the results of our interface designs. After upload, the underlying program determines for a given protein-protein complex parameters that allow one to estimate the stability of macromolecular complexes (Krissinel and Henrick, 2007). We concentrated on the analysis of potential hydrogen bonds and salt bridges formed across the interfaces, because of their substantial contribution to complex stability. To begin with, we built three-dimensional models of all experimentally confirmed complexes (see STAR Methods): the ecPabA:ecPabB complex and the five complexes consisting of a PabA variant and stTrpEx, which is the closest relative of ecTrpEx with known three-dimensional structure. In order to relax the complex structures, they were subjected to a short molecular dynamics simulation and each of the final poses was stored in PDB format. Finally, these six files were uploaded to the PISA server and residues involved in hydrogen bonds or salt bridges across the interfaces were identified. Table 2.1 shows that the four ecPabA residues Y8, D9, Q17, and C54 form hydrogen bonds with residues from ecPabB. As expected, a structural analysis of the complexes consisting of stTrpEx and PabA-CAD or PabA** revealed new interactions not observed in the wild-type complex ecPabA:ecPabB (Figure 2.8). In the PabA-CAD:stTrpEx complex, a hydrogen bond forms between residue G54 and residue G259 and a further one between residue V28 and residue E114, the latter is located in the interface-addon of stTrpEx. Two hydrogen bonds are formed between residue R20 and residue A371; moreover, both a hydrogen bond

and a salt bridge were created between residue D17 and residue R382. The PabA**:stTrpEx complex revealed even more new interactions: Residue G54 also forms a hydrogen bond with residue G259; additionally, residue D17 forms a hydrogen bond with residue G380. Interestingly, residues V28, Y29, and R30 form hydrogen bonds with the residues N115, L111, and P110, which are located in the interface-addon of stTrpEx. Moreover, hydrogen bonds and salt bridges form between residues E98 and R257 and between D17 and R382, respectively. Hence, this analysis suggests that the newly designed complexes between PabA variants and stTrpEx are at least as stable as the complex between ecPabA and ecPabB. This assumption is in agreement with our biochemical characterization (Table 2.2): The designed variants PabA-CAD and PabA** form stable complexes with ecTrpEx and achieve $K_D{}^{app}$ values within the range of the native complexes ecPabA:ecPabB and stTrpG:ecTrpEx. However, this in silico analysis, which was based on homology models of the complexes, did not allow us to explain binding differences in more detail.

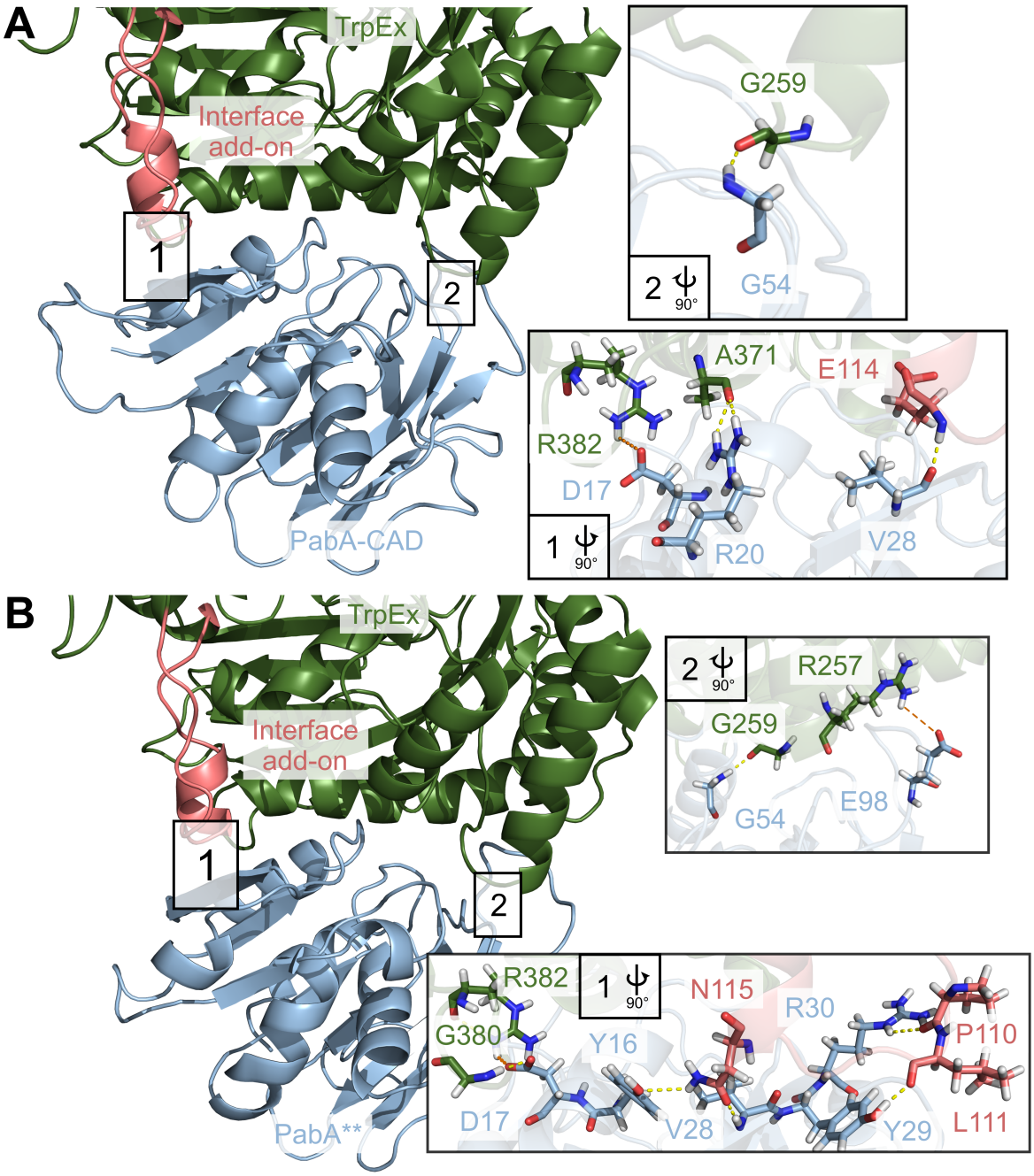

Figure 2.8: Structural representation of the newly designed interactions in PabA-CAD:TrpEx and PabA:TrpEx complexes.** In both design approaches, newly introduced hydrogen bonds (dotted yellow lines) and salt bridges (dotted orange lines) are concentrated in two regions labelled 1 and 2, respectively.
(**A**) Interface of the PabA-CAD:TrpEx complex in cartoon representation. In region 1, four hydrogen bonds and one salt bridge are formed between PabA-CAD and TrpEx residues. Region 2 contains one additional hydrogen bond.
(**B**) Interface of the PabA**:TrpEx complex. In region 1, five PabA** residues form six hydrogen bonds and one salt bridge with TrpEx residues. In region 2, two hydrogen bonds and one salt bridge are formed between interface residues.

2.4 Discussion

Cells contain thousands of different proteins, which in principle can form a myriad of different heteromeric complexes. The formation of the large majority of those potential complexes has to be avoided, because they would be non-physiological and presumably lead to unproductive metabolic cross-talk. But how does nature assure that only physiologically meaningful complexes are formed, given the limited number of different quaternary structure topologies (Ahnert et al., 2015) and interface geometries (Gao and Skolnick, 2010; Garma et al., 2012)? This problem seems to be particularly acute when two or more homologous proteins with similar sequences and highly related structures compete for the same interaction partner (Schreiber and Keating, 2011).

Class I GATases, which form a family of well characterized enzyme complexes (Zalkin, 1993), are suitable model systems for studying the structural basis of protein interaction specificity in such difficult cases. Each member of this family contains a glutaminase subunit with a conserved $\alpha\beta$-hydrolase fold (Ollis et al., 1992) being bound to a specific synthase subunit. In our analysis of protein interaction specificity in class I GATases, we focused on the protein complexes ADCS and AS (Plach et al., 2017) which are involved in folate and tryptophan biosynthesis. These enzymes face the problem that not only the two glutaminases but also the two synthases are homologous proteins, which makes it particularly challenging to assure the exclusive formation of physiological complexes. Indeed, in most bacteria a single glutaminase PabA interacts with the two synthases PabB (forming the ADCS) and TrpE (forming the AS). This situation, however, complicates the independent regulation of folate and tryptophan biosynthesis. γ-proteobacteria avoid this regulatory problem by the formation of the independent ADCS complex PabA:PabB and the newly evolved independent AS complex TrpG:TrpEx. TrpEx has acquired additional structural elements in the interface periphery, which we termed “interface add-on”. We demonstrated the significance of the interface add-on for the specificity of TrpEx for TrpG by showing that its (partial) deletion leads to the binding of PabA (Plach et al., 2017). Vice versa, we also tried to elucidate the structural basis of the specificity of TrpG for TrpEx. For this purpose, the variant PabA* has been generated, which contained several TrpG-specific residues that interact with the interface add-on of TrpEx. Remarkably, PabA* formed a detectable complexes with both TrpEx and PabB (Plach et al., 2017).

The evolution of a highly specific interface required the adaptation of both interaction partners.

PabA* contained only a minor fraction of the TrpG residues that interact with the interface add-on of TrpEx (Plach et al., 2017). We reasoned that the specific binding of a PabA variant to TrpEx required the implementation of the entire TrpG-counterpart of the interface add-on. We used grafting to put this idea into action, because it has been successfully used in several design

experiments that altered the binding properties of antibodies (Adolf-Bryfogle et al., 2018; Jones et al., 1986; Darnell et al., 2000; Riechmann et al., 1988), inhibitors (Guntas et al., 2016), and other protein backbones (Azoitei et al., 2011; Hussain et al., 2018; Netzer et al., 2018). Hence, we grafted 25 consecutive TrpG residues onto PabA and thereby generated the PabA-CA variant. This variant was subsequently optimized by Rosetta, yielding the related variant PabA-CAD. Regardless of the Rosetta results, we also performed a data-driven approach. This approach was relying on the assumption that the co-evolution is reflected in the conservation of interface residues in TrpG and yielded the variants PabA*$^{+}$ and PabA**. Qualitative SEC runs indicated a drastically reduced affinity of all four newly generated PabA variants for ecPabB and, at the same time, significant binding to ecTrpEx (Figures 2.5, 2.6). This interaction specificity switch was quantitatively confirmed by activity titration experiments (Table 2.2).

In TrpEx and all PabA variants, specificity-determining residues are located at the rim of the interfaces.

Commonly, interfaces of globular proteins consist of residue "patches" on the respective protein surface and complex formation does not drastically affect the global conformation of the interaction partners (Zhang et al., 2013). This conformational robustness facilitates the evolution of highly specific interactions via mutations of interface residues. The synthase - glutaminase complexes of ADCS and AS possess typical, relatively flat interfaces of globular proteins and our findings confirm that few mutations are sufficient to change interaction specificity. However, these interfaces do not only mediate complex formation, but are also crucial for ammonia channeling and allosteric communication between the glutaminase and synthase subunits. Most plausibly, these functional constraints necessitate the strict conservation of about two third of the interface residues between PabA and TrpG (Figure 2.3 C), which complicates the design of interface elements that determine binding specificity. We have shown that the TrpEx interface add-on extends the interface in a way that does not compromise the functional properties (Plach et al., 2017). The results presented here indicate that the TrpG-specific residues introduced into the PabA variants that bind TrpEx are also located at the rim of the interface (Figure 2.3 B). Thus, our data suggest that the difference in residue conservation, which is generally higher in core than in rim residues (Hu et al., 2000), is due to stronger functional constraints of core residues. In accordance with this notion, the higher variability of rim residues might be due to the specific interactions that determine the binding of the correct interaction partner. Interestingly, an analysis of the co-variation of amino acid frequencies at residue positions from different proteins is highly efficient to predict residue-residue contacts and protein-protein interfaces (Ovchinnikov et al., 2014). This strong correspondence makes it clear that the occupancy of variably occupied and thus seemingly less important residue positions has to be chosen carefully for a successful interface design.

Interestingly, the residues chosen by Rosetta (based on scoring functions representing biophysical forces) and by evolution (represented by an MSA) for rim positions differ noticeable (Figure 2.3B). As Table 2.1 confirms, 16 positions of PabA-CAD and PabA** are occupied by different residues, but their interplay resulted in approximately similar binding specificities (Table 2.2). Figure 2.8 suggests that even the interactions with the TrpEx add-on distinguish these two PabA variants. The silico analysis of interface properties identified only a small number of strong interactions, which did not allow us to identify key residues. Thus, given this marked variation of rim positions, combinatorial complexity makes it difficult to identify interface residues or residue combinations that might be crucial for binding specificity. As expected, the Rosetta protocol which was optimized to alter interface specificity, fulfilled this task satisfactorily. On the other hand, the protocol was blind to consequences for enzyme activity. Thus, it is not surprising that the data driven approach that beneficially exploits the synopsis of many efficient complexes gave rise to a functionally more active PabA variant (Table 2.2).

2.5 Conclusions

The comprehension of the structural basis of protein-protein interaction specificity is an important goal of biochemistry. This goal can be considered as reached only if the specificity of a given protein for its physiological partner can be redirected to another partner by protein design. Such approaches can be most readily performed with structurally and functionally well-characterized protein complexes. A prominent example is provided by the glutamine amidotransferases (GATases), each of which consists of a glutaminase subunit that specifically binds to a given synthase subunit. Moreover, the synthase subunit strongly stimulates the catalytic turnover of its glutaminase subunit partner. We used protein design for the alteration of interaction specificity in a pair of GATases, namely anthranilate synthase (AS), which is involved in tryptophan biosynthesis, and aminodeoxychorismate synthase (ADCS), which is involved in folate biosynthesis. In γ-proteobacteria, AS consists of the glutaminase TrpG and the synthase TrpEx whereas ADCS consists of the glutaminase PabA and the synthase PabB. Two complementary approaches were performed to generate a non-native complex between PabA and TrpEx, computational design based on the Rosetta program suite and alternatively a data-driven design based on the differential conservation of interface residues in TrpG and PabA. Both approaches were successful insofar the interaction specificities of all generated variants were dramatically shifted from PabB to TrpEx. The presented data show that we have identified crucial residues responsible for the specific formation of the native TrpG:TrpEx complex by generating non-native complexes between designed PabA variants and TrpEx. We anticipate that our approaches will be helpful to identify structural determinants of protein-protein interaction specificity also in other protein complexes.

Acknowledgements

We thank Jeannette Ueckert for excellent technical assistance. This work was supported by a grant of the Deutsche Forschungsgemeinschaft to R.S. (STE 891/9-3).

Author contributions

R.H., F.S., F.F., and S.S. performed the experiments and J.N. conducted the computational analysis. R.M. and R.S. supervised the work and wrote the manuscript with contributions from J.N. and F.S..

Declaration of interests

The authors declare no competing interests.

2.6 Star Methods

Lead contact and materials availability

Further information and requests for resources and reagents should be directed to and will be fulfilled by the Lead Contact, Reinhard Sterner (reinhard.sterner@ur.de).

Data and code availability

The sequences of all proteins studied here can be found in Data S1. The raw data of the PISA analyses can be found in Data S2, Data_1.zip, and Data_2.zip. The scripts and data needed for the Rosetta design can be found in Data S3 and Data S4. The PDB structure file related to Figure 8 can be found in Data S5 and Data_3.zip.

Experimental model and subject details

E. coli NEB Turbo cells were originally purchased from New England Biolabs. The strains were further maintained following the manufacturer's guidelines for the preparation of chemically competent cells. Cells were grown in lysogenic broth (LB) medium at 37 °C, and stored in 80% glycerol at -80 °C.

2.7 Method Details

Purification of chorismate

Chorismate was purified from culture supernatant of *Escherichia coli* KA12 cells (graciously provided by Prof. Dr. Donald Hilvert and Prof. Dr. Peter Kast, ETH Zürich). *E. coli* KA12 cells (Phenotype: F^-, λ^- Δ(*pheA-tyrA-aroF*) *thi*-1 *endA*1 *hsdR*17Δ(*argF – lac*)205(*U*169) *supE*44Δ(*srlR-recA*)306::*Tn*10) carry a deletion of the chorismate mutase, so that chorismate is overproduced and secreted into the culture medium (Grisostomi et al., 1997). The protocol involves inoculation of 5 ml of LB_{Tet} (12.5 µg/ml tetracycline) medium with *E. coli* KA12 and shaking overnight at 37 °C. 500 ml growth medium (2 g/L casamino acids, 2 g/L yeast extract, 41 mg/L tryptophan, 20 mL 50x Vogel-Bonner salts, 1.6 g/L glucose) were inoculated with 5 ml preculture and cells were grown at 30 °C (150 ppm) for 4-6 h to give OD_{600} = 1.6 - 1.9 and then centrifuged at 4000 g for 20 min (4 °C). Cells were resuspended in 500 ml accumulation medium (12.8 g/L Na_2HPO_4, 1.36 g/L KH_2PO_4, 18 g/L glucose, 2.7 g/L NH_4Cl, 20.3 mg/L $MgCl_2$ - $6H_2O$, 2 mg/L L-tryptophan) and the cell suspension was incubated at 30 °C (120 rpm) for 16 h. After removing the cells by centrifugation, the supernatant was acidified with HCl (5 M) to pH 1.5, and subsequently extracted with ethylacetate (2x 180 ml). The combined ethylacetate extracts were washed with saturated brine (360 g/L NaCl). Removal of the solvent in vacuo yielded a small volume of a red oily substance. Crude chorismate was purified in a single step by flash chromatography on C18 reverse phase silica gel. The crude material (150 to 300 mg) was loaded on 50 g C18 reverse phase silica and eluted with 10 mM ammonium acetate (pH 6) under pressure, 10 mM fractions were collected and for each fraction absorption spectra were recorded. Chorismate and the co-elutant 4-hydroxybenzoate feature absorption maxima at 275 nm and 244 nm, respectively (ϵ_{270} (chorismate) = 2630 $M^{-1}cm^{-1}$; (Addadi et al., 1983)), allowing for a rough quantification of the two substances in each fraction. Fractions containing chorismate in sufficient concentration and purity were combined and lyophilized. The resulting lyophilisate was dissolved in a small volume of sterile water, aliquoted, and stored at −80 °C. Sample puritiy was assessed by analytical reverse phase HPLC and enzymatic conversion with anthranilate synthase in an enzymatic assay.

Cloning and mutagenesis

The genes for PabA*, PabA-CA, and PabA-CAD variants as well as a gene for a PabA variant containing nine mutations (Y8I, Q17D, C20R, E21T, K29Y, D32H, C54G, R95Q, R98E) were synthesized (Life Technologies) and cloned into pET21a-*Bsa*I (Rohweder et al., 2018). Additional mutations that were necessary to generate the PabA*$^+$ and PabA** variants were then introduced using a modified version of the NEB Q5 site-directed mutagenesis protocol (New England Biolabs) and also cloned into pET21a-*Bsa*I. The sequences of the oligonucleotides used for site-directed mutagenesis are available upon request.

Expression and purification of proteins

The pET21a-*Bsa*I plasmids containing the genes for PabA-CA, PabA-CAD, PabA*$^{+}$, and PabA**, were used to transform *E. coli* NEB Turbo cells. Gene expression and protein purification were performed as described (Semmelmann et al., 2019b). The purity of all other proteins was > 90%, as judged by SDS-PAGE. Following purification, proteins were immediately used or frozen in liquid nitrogen and stored at −80 °C. Protein concentrations were determined by measuring the absorbance at 280 nm, using the molar extinction coefficient calculated via ExPASy ProtParam (https://web.expasy.org/protparam/).

Analytical size exclusion chromatography (SEC)

Structural complex formation between glutaminases and synthases was examined by analytical SEC using a Superdex 75 10/300 GL column (volume: 23.5 ml; void volume: 7.6 ml) operated on an ÄKTAmicro system (GE Healthcare), essentially as described (Semmelmann et al., 2019a,b). Individual synthases and glutaminases were assayed at subunit concentrations of 50 µM (A volume of 50 µl was applied to a 100 µl loading loop). For analysis of complex formation, synthases and glutaminases were equimolarly mixed to final concentrations of 50 µM each. Elution at a flow rate of 0.5 ml/min was performed with 50 mM Tris/HCl, pH 7.5, 150 mM KCl, 5 mM $MgCl_2$ at 25 °C and monitored by measuring the absorbance at 280 nm. The column was calibrated with proteins from the GE Healthcare SEC low-molecular-weight and SEC high-molecular-weight calibration kits.

Activity titration

Glutaminase activities were measured under steady-state conditions with a coupled assay in a 96-well titer plate, essentially as described (Semmelmann et al., 2019a). Glutamate formed by the glutaminases was converted to α-ketoglutarate by glutamate-dehydrogenase (GDH) with simultaneous reduction of NAD^+ to NADH. A standard assay contained 50 mM tricine-KOH buffer, pH 8.0, 150 mM KCl, 5 mM $MgCl_2$, 1 mM DTT, 10 mM NAD^+, 1 mg/ml GDH, and 20 mM glutamine. After determining the unstimulated glutaminase activity with 0.5 or 1.0 µM glutaminase at 340 nm and 25 °C, different concentrations of the respective synthase (50 - 5000 nM) were added. Rates of glutamine hydrolysis corrected for unstimulated glutaminase activity were plotted as a function of synthase concentration. Hydrolysis rates were normalized to 1 µM glutaminase. Enzymatic constants ($K_D{}^{app}$ and $k_{cat}{}^{app}$) were determined by fitting a hyperbolic equation to the saturation curves.

$$k_{app} = \frac{k_{cat}{}^{app} * [Synthase]}{K_D{}^{app} + [Synthase]} \tag{2.1}$$

Interface design based on grafting

A sequence named PabA-CA was created by replacing ecPabA residues 6 - 30 with the residues 7 - 31 of stTrpG (PDB-ID 1i1q chain B) that interact with the stTrpEx add-on. To determine initial PabA:TrpEx complex structures, stTrpEx (PDB-ID 1i1q chain A) served as target and ecPabA (a homology model based on PDB-ID 6qur chain A) as scaffold. For PabA-CA, 48 alternative structures were created, each by means of 100,000 cycles of `Rosetta:AnchoredPDBCreator` (Lewis and Kuhlman, 2011). These structures were sorted according to their LAM-score that assesses the quality of anchor integration and the two most successful grafts gr_1 and gr_2 were identified. These two structures were relaxed by means of `Rosetta:FastRelax` and subsequently taken as input for `Rosetta:MSFAnchoredDesign` (Löffler et al., 2017) to perform a multi-state design with two states. During the `AnchoredDesign` protocol, Rosetta was free to mutate arbitrarily all residues of the ecPabA interface and to repack all sidechains with a maximal distance of 6 Å to interface residues. In stTrpEx, no mutations were allowed, but interface residues were subjected to repacking. The native PabA-CA sequence served as seed for the design of novel interfaces by means of the genetic algorithm implemented in `Rosetta:MSFAnchoredDesign`. Each generation consisted of 239 sequences; for each sequence, the fitness was determined by adding the two `soft-rep-design` scores reached for gr_1 and gr_2. The algorithm was stopped after 46 generations and the final sequences were ranked according to their fitness. From the top ten best scoring sequences, a sequence (named PabA-CAD) was chosen that was free of mutations we considered unfavorable for solubility or interaction stability (Semmelmann et al., 2019b). Additional information files can be found in Data S3 and Data S4.

Computational analysis of protein-protein interfaces

Three-dimensional models of protein complexes were built by means of `YASARA` (Krieger and Vriend, 2014) (version 18.4.24). To begin with, the native TrpG:TrpEx complex (PDB-ID 1i1q) and the model of the PabA:PabB complex were aligned (superimposed) by means of the `YASARA` implementation of `MUSTANG` (Konagurthu et al., 2006) and PabB and TrpG were removed subsequently. The poses of the remaining PabA and TrpEx enzymes were combined to a model of the PabA:TrpEx complex. Utilizing the swap method of `YASARA`, the mutations specific for the different PabA variants were introduced and for each variant, a three-dimensional model was saved for further analysis. A `YASARA` script to introduce these mutations can be found in Data S5.

For each model, a relaxed three-dimensional structure was created by means of `GROMACS` (Berendsen et al., 1995) (version 5.1.2) and the AMBER03 (Wang et al., 2004) force field. The complex structures were placed in a rectangular water box; the simulation cell was at least 5 Å larger than the solute along each axis and the system was neutralized by adding NaCl ions. To prepare a production run, the solvated system was at first energy minimized

until reaching convergence followed by a two-part equilibration phase (each lasted 50 ps) that began with an NVT simulation followed by an NPT simulation (Andersen, 1980). The last snapshot of the production runs (each lasting 1 ns) were uploaded to the PDBePISA service (https://www.ebi.ac.uk/pdbe/pisa/) in order to identify hydrogen bonds and salt bridges (Krissinel and Henrick, 2007) with a cutoff value of 4 Å. The PISA output was added to Data S2.

Quantification and statistical analysis

Data points in activity titrations are the mean ± SEM (standard error of mean) of at least two technical replicates. Analyses and data representations were performed and assembled in Origin 2020. Fitting of datapoints to hyperbolic equations resulted in values for enzymatic constants ($K_D{}^{app}$, $k_{cat}{}^{app}$). Detailed description of the used fitting functions can be found in the method details section.

2.8 Supplemental Data

Data S1

Sequences of ecPabA, PabA*, PabA-CA, PabA-CAD, PabA*$^{+}$, and PabA** in FASTA format.

>ecPabA
MILLIDNYDSFTWNLYQYFCELGADVLVKRNDALTLADIDALKPQKIVISPGPCTPDEA
GISLDVIRHYAGRLPILGVCLGHQAMAQAFGGKVVRAAKVMHGKTSPITHNGEGVFRG
LANPLTVTRYHSLVVEPDSLPACFDVTAWSETREIMGIRHRQWDLEGVQFHPESILSEQ
GHQLLANFLHR
>PabA*
MILLIDNYDSFTWNLYDQFRELGADVLVYRNQALTLADIDALKPQKIVISPGPCTPDEA
GISLDVIRHYAGRLPILGVCLGHQAMAQAFGGKVVRAAKVMHGKTSPITHNGEGVFRG
LANPLTVTRYHSLVVEPDSLPACFDVTAWSETREIMGIRHRQWDLEGVQFHPESILSEQ
GHQLLANFLHR
>PabA-CA
MILLIDNIDSFTWNLADQLRTNGHNVVIYRNDALTLADIDALKPQKIVISPGPCTPDEAG
ISLDVIRHYAGRLPILGVCLGHQAMAQAFGGKVVRAAKVMHGKTSPITHNGEGVFRGL
ANPLTVTRYHSLVVEPDSLPACFDVTAWSETREIMGIRHRQWDLEGVQFHPESILSEQG
HQLLANFLHR
>PabA-CAD
MILLIDNKASFTWNLADCLRTNGHNVVVYDNCALTLADIDALKPQKIVISPGPGTPDEA
GISLDVIRHYAGRLPILGVCLGHQAMAQAFGGKVVRAAKVEHGKTSPITHNGEGVFRG
LANPLTVTRYHSLVVEPDSLPACFDVTAWSETREIMGIRHRQWDLEGVQFHPESILSEQ
GHQLLANFLHR
>PabA*$^{+}$
MILLIDNIDSFTWNLYDQFRTLGADVLVYRNHALTLADIDALKPQKIVISPGPGTPDEAG
ISLDVIRHYAGRLPILGVCLGHQAMAQAFGGKVVQAAEVMHGKASPITHNGEGVFRGL
ANPLTVTRYHSLVVEPDSLPACFDVTAWSETREIMGIRHRQWDLEGVQFHPESILSEQG
HQLLANFLHR
>PabA**
MILLIDNIDSFTWNLYDQFRTLGADVLVYRNHALTLADIDALKPQKIVISPGPGTPDEAG
ISLDVIRHYAGRLPILGVCLGHQAMAQAFGGKVVQAAEVMHGKASPITHNGEGVFRGL
ANPLTVARYHSLVVEPDSLPACFDVTAWSETREIMGIRHRQWDLEGVQFHPESILSEQG
HQLLANFLHR

Data S2

Relaxed models of the six complexes ecPabA:ecPabB, PabA-CA:stTrpEx, PabA-CAD:stTrpEx, PabA*:stTrpEx, PabA*$^{+}$:stTrpEx, and PabA**:stTrpEx in PDB format are included in the Data_1_PDB_Files.zip. Additionally, the output files of the corresponding six PISA analyses are included as Data_2_PISA_Files.zip. The data is related to Table 2.1 and Figure 2.3.

Data S3

The following two file contents are required for the Rosetta:AnchoredPDBCreator run. The "Loop_PDBC" file determines the region for the stTrpG anchor in ecPabA and the "Options_PDBC" file contains the general settings for the run. Data and scripts are related to Figure 2.2, Figure 2.8, and STAR Methods.

File - Loop_PDBC:
A 4 5 29
File - Options_PDBC:
#Flagsfile for Design
-nstruct 48
-anchor_pdb Anchor_TrpG.pdb
-target_pdb Target_TrpE.pdb
-scaffold_pdb Scaffold_PabA.pdb
-scaffold_loop Loop_PDBC
-APDBC_cycles 100000

Data S4

The following file contents contain the general settings to run the design. The "Anchor" file determines the anchor region. The fitness function used for this design is defined in the "Fitness.daf" file. First shell residues are listed in the "GenEntity.resfile" file and mapped to specific residues in the "GenPos.corr" file. Second shell residues are defined in the "GenPos.resfile2" file. The "Loop" file determines the flexible loop region. Further, general options required for the MSD run are listen in the "MSF_Options" file. Data and scripts are related to Figure 2.2, Figure 2.8, and STAR Methods.

File - Anchor:
A 6 30

File - Fitness.daf:

STATE_VECTOR state1 /Path/to/State_1
STATE_VECTOR state2 /Path/to/State_2
SCALAR_EXPRESSION best_state1 = vmin(state1)
SCALAR_EXPRESSION best_state2 = vmin(state2)
SCALAR_EXPRESSION best_sum = best_state1 + best_state2
FITNESS best_sum

File - GenEntity.resfile:

35
ALLAA EX 1 EX 2

File - GenPos.corr:

1 8 A
2 9 A
3 10 A
4 11 A
5 13 A
6 14 A
7 17 A
8 18 A
9 20 A
10 21 A
11 27 A
12 28 A
13 29 A
14 30 A
15 32 A
16 33 A
17 34 A
18 52 A
19 53 A
20 54 A
21 95 A
22 98 A
23 100 A
24 101 A
25 102 A
26 103 A
27 127 A
28 128 A

29 129 A
30 130 A
31 168 A
32 170 A
33 171 A
34 172 A
35 173 A

File - GenPos.resfile2:

NATRO
START
1 A NATAA EX 1 EX 2
2 A NATAA EX 1 EX 2
3 A NATAA EX 1 EX 2
4 A NATAA EX 1 EX 2
5 A NATAA EX 1 EX 2
6 A NATAA EX 1 EX 2
7 A NATAA EX 1 EX 2
12 A NATAA EX 1 EX 2
15 A NATAA EX 1 EX 2
16 A NATAA EX 1 EX 2
19 A NATAA EX 1 EX 2
22 A NATAA EX 1 EX 2
23 A NATAA EX 1 EX 2
24 A NATAA EX 1 EX 2
25 A NATAA EX 1 EX 2
26 A NATAA EX 1 EX 2
31 A NATAA EX 1 EX 2
35 A NATAA EX 1 EX 2
36 A NATAA EX 1 EX 2
37 A NATAA EX 1 EX 2
38 A NATAA EX 1 EX 2
39 A NATAA EX 1 EX 2
50 A NATAA EX 1 EX 2
51 A NATAA EX 1 EX 2
55 A NATAA EX 1 EX 2
56 A NATAA EX 1 EX 2
58 A NATAA EX 1 EX 2
60 A NATAA EX 1 EX 2
61 A NATAA EX 1 EX 2

78 A NATAA EX 1 EX 2
79 A NATAA EX 1 EX 2
80 A NATAA EX 1 EX 2
93 A NATAA EX 1 EX 2
94 A NATAA EX 1 EX 2
96 A NATAA EX 1 EX 2
97 A NATAA EX 1 EX 2
99 A NATAA EX 1 EX 2
104 A NATAA EX 1 EX 2
105 A NATAA EX 1 EX 2
123 A NATAA EX 1 EX 2
124 A NATAA EX 1 EX 2
125 A NATAA EX 1 EX 2
126 A NATAA EX 1 EX 2
131 A NATAA EX 1 EX 2
166 A NATAA EX 1 EX 2
167 A NATAA EX 1 EX 2
169 A NATAA EX 1 EX 2
174 A NATAA EX 1 EX 2
175 A NATAA EX 1 EX 2
177 A NATAA EX 1 EX 2
178 A NATAA EX 1 EX 2
297 B NATAA EX 1 EX 2
298 B NATAA EX 1 EX 2
299 B NATAA EX 1 EX 2
300 B NATAA EX 1 EX 2
301 B NATAA EX 1 EX 2
302 B NATAA EX 1 EX 2
304 B NATAA EX 1 EX 2
346 B NATAA EX 1 EX 2
347 B NATAA EX 1 EX 2
443 B NATAA EX 1 EX 2
444 B NATAA EX 1 EX 2
445 B NATAA EX 1 EX 2
446 B NATAA EX 1 EX 2
447 B NATAA EX 1 EX 2
449 B NATAA EX 1 EX 2
543 B NATAA EX 1 EX 2
544 B NATAA EX 1 EX 2

547 B NATAA EX 1 EX 2
550 B NATAA EX 1 EX 2
551 B NATAA EX 1 EX 2
553 B NATAA EX 1 EX 2
554 B NATAA EX 1 EX 2
555 B NATAA EX 1 EX 2
557 B NATAA EX 1 EX 2
558 B NATAA EX 1 EX 2
561 B NATAA EX 1 EX 2
562 B NATAA EX 1 EX 2
564 B NATAA EX 1 EX 2
566 B NATAA EX 1 EX 2
567 B NATAA EX 1 EX 2
569 B NATAA EX 1 EX 2
571 B NATAA EX 1 EX 2
574 B NATAA EX 1 EX 2
595 B NATAA EX 1 EX 2
616 B NATAA EX 1 EX 2
617 B NATAA EX 1 EX 2
618 B NATAA EX 1 EX 2
619 B NATAA EX 1 EX 2
620 B NATAA EX 1 EX 2
623 B NATAA EX 1 EX 2
674 B NATAA EX 1 EX 2
675 B NATAA EX 1 EX 2
676 B NATAA EX 1 EX 2

File - Loop:

LOOP 5 47 0 0 0

File - MSF_Options:

#packing options
-ex1
-ex2
-use_input_sc
-extrachi_cutoff 8
-linmem_ig 42
#minimization options
-run::min_type dfpmin_armijo
-nblist_autoupdate
#loops options

```
-loops::vicinity_sampling true
-loops::loop_file ../Loop
#score
-score:weights /Path/to/Rosetta/main/database/scoring/weights/soft_rep_design
#AnchoredDesign options
-AnchoredDesign
    -anchor /Path/to/Anchor
    -refine_only false
    -show_extended false
    -perturb_temp 0.8
    -perturb_show false
    -debug false
    -refine_temp 0.8
    -refine_repack_cycles 50 #Perform repack/minimize every N cycles of refine mode
    -allow_anchor_repack true
    -vary_cutpoints false
#Loopmodeling
    -perturb_CCD_off false
    -perturb_KIC_off false
    -refine_CCD_off false
    -refine_KIC_off false
#
    -rmsd false
    -unbound_mode false
    -no_frags false
    -chainbreak_weight 2.0
    -testing::VDW_weight 2
#sample-size command
-AnchoredDesign::perturb_cycles 2500
-AnchoredDesign::refine_cycles 5000
-nstruct 1
#9999
#MSF
-msf::entity_resfile /Path/to/GenEntity.resfile
-msf::fitness_file /Path/to/Fitness.daf
-msf::pop_size 239
-msf::generations 500
-msf::fraction_by_recombination 0.05
-msf::seed_sequence_from_input_pdb Path/to/S_0023_min.pdb
```

-msf::resfile_tmpdir /Path/to/Tempres
-msf::checkpoint_write_interval 1
-msf::checkpoint_prefix /Path/to/Checkpoints/Checkpoint
-msf::seed_sequence_using_correspondence_file /Path/to/GenPos.corr
#-msf::darwin_resume true

Chapter 3

Rosetta:MSF:NN Boosting Multi-state Computational Protein Design with a Neural Network

Julian Nazet, Elmar Lang, Rainer Merkl*

Institute of Biophysics and Physical Biochemistry
University of Regensburg, D-93040 Regensburg, Germany

* Corresponding author:
Rainer Merkl: +49-941 943 3086; Rainer.Merkl@ur.de

Short title: Rosetta:MSF:NN boosts performance of MSD with a neural network

3.1 Abstract

Rational protein design aims at the targeted modification of existing proteins. To reach this goal, software suites like Rosetta propose sequences to introduce the desired properties. Challenging design problems necessitate the representation of a protein by means of a structural ensemble. Thus, Rosetta multi-state design (MSD) protocols were developed wherein each state represents one conformation of the protein. Computational demands of MSD protocols are high, because for each of the candidate sequences a costly three-dimensional (3D) model has to be created and assessed for each of the states. Each of these scores contributes one data point to a complex, design specific energy landscape. As neural networks (NN) proved well-suited to learn such solution spaces, we integrated one into the previously proposed framework `Rosetta:MSF` instead of the genetic algorithm with the aim to reduce computational costs.

As its predecessor, `Rosetta:MSF:NN` administers a set of candidate sequences and their scores and scans sequence space iteratively. During each iteration, the union of all candidate sequences

and their Rosetta scores are used to re-train NNs that possess a design-specific architecture. The enormous speed of the NNs allows an extensive assessment of alternative sequences, which are ranked on the scores predicted by the NN. Costly 3D models are computed only for a small fraction of best scoring sequences; these and the corresponding 3D-based scores replace half of the candidate sequences during each iteration.

The analysis of 48,588 candidate sequences generated for a specific design problem confirmed that the NN predicted 3D-based scores quite well; the Pearson correlation coefficient was > 0.9 for up to three mutations. Applying `Rosetta:MSF:NN:enzdes` to a benchmark consisting of 16 ligand binding problems showed that this protocol converges three-times faster than the genetic algorithm and finds sequences with better scores. Moreover, the biochemical characterization of a newly designed HisB-N enzyme confirmed that our approach can be utilized for MSD and negative design problems.

3.2 Author Summary

An important goal of many computational protein design programs is to find a protein sequence that i) fits to a given three-dimensional structure and ii) fulfills certain requirements that arise from the desired design goal. Due to its versatility, Rosetta is the "swiss army knife" of protein design and several protocols exist for the identification of optimal sequences. To tackle challenging design problems, one has to represent a protein with several three-dimensional structures; their analysis slows down the design process. In order to improve the performance of these protocols, we have integrated a neural network and demonstrated that it can learn design specific energy landscapes. Thus, the costly generation and assessment of three-dimensional models needed to score candidate sequences can be reduced such that the novel protocol converges three times faster than our previous one. These findings were derived from an analysis of 16 protein design problems that served as benchmark.

3.3 Introduction

Computational protein design has become an important tool in molecular biology (Gainza et al., 2016). Different approaches and protocols have proven their reliability for a broad range of applications. To name just a few design problems for a protein under study, the informed user can increase thermostability (Shah et al., 2007; Goldenzweig et al., 2016), alter the binding of ligands (Looger et al., 2003; Shifman and Mayo, 2003), redesign interactions with other proteins (Fleishman et al., 2011; Procko et al., 2014) or design novel catalytic sites (Richter et al., 2011). Moreover, the *de novo* design of catalytically active proteins is feasible (Kaplan and DeGrado, 2004; Röthlisberger et al., 2008) as well as antibody redesign (Adolf-Bryfogle et al., 2018; Lippow et al., 2007).

For challenging design problems that require the modelling of structural flexibility, the traditional, single-state design (SSD) strategy that is optimal for finding a sequence for a structurally fixed backbone is not sufficient. For example, enzymes adopt different conformations during a catalytic cycle and more generally, all important biological effects are best represented by an ensemble of conformational states; for a review see (Wrabl et al., 2011). This is why multi-state design (MSD) protocols have been introduced, which score a single sequence with respect to conformationally different backbones modelled as states (Leaver-Fay et al., 2011a; Löffler et al., 2017; Yanover et al., 2007; Davey and Chica, 2012; Negron and Keating, 2013; Allen and Mayo, 2010; Harbury et al., 1998; Fromer et al., 2010; Karimi and Shen, 2018; Vucinic et al., 2020). This methodology is useful, if the design target is represented by an ensemble of structures, or if the protein has to adopt several distinct conformations, for example during a catalytic cycle. Moreover, MSD allows for negative design, i. e., the computation of sequences that destabilize certain states related to misfolded conformations or an undesired binding interaction (Pokala and Handel, 2005). However, the more precise MSD approach has its price due to the demands for higher computing efforts needed for the identification of appropriate sequences: Considering m states requires scoring each candidate sequence m times and combining these scores to a global "fitness" value in order to identify sequences that are optimal for all states. MSD approaches have shown their superiority in applications like the prediction of mutational tolerance in enzymes (Humphris-Narayanan et al., 2012), the understanding of thermal adaptation of enzymes (Howell et al., 2014), the design of influenza antibodies (Sevy et al., 2019), multi-specific interfaces (Humphris and Kortemme, 2007), and multi-substrate enzymes (St-Jacques et al., 2019).

A well-proven and highly flexible software suite supporting many problems of protein design is Rosetta (Leaver-Fay et al., 2011b) and several Rosetta-based MSD protocols have been implemented (Leaver-Fay et al., 2011a; Löffler et al., 2017; Sevy et al., 2019). To determine the fitness of a sequence during the search phase, the 3D residue positions, whose occupancy can be varied by the software protocol, are decorated with the considered amino acid side chains. A key element of this assessment is to find an optimal combination of side chain orientations (Kaufmann et al., 2010). Building these optimized 3D ($3D_{opt}$) models consumes most of the computational time needed for the whole protein design protocol. If the occupancy of n residue positions is unconstrained for a design task, optimal rotamer combinations have to be found and assessed for 20^n different $3D_{opt}$ models in a SSD and for $m * 20^n$ $3D_{opt}$ models in an MSD experiment. Thus, even for design problems of moderate complexity, a hybrid method that does not require the computation of a costly $3D_{opt}$ model to score each of the candidate sequences might drastically reduce computation time of Rosetta's protein design protocols.

This search for optimal sequences can be considered a problem of multi-dimensional regression, were every combination of amino acid residues yields one data point of the design-specific energy landscape. Neural networks (NNs) proved well-suited to solve complex classification and regression problems of computational biology (Rost, 1996; Kuhlman and Bradley, 2019). Thus,

we wanted to know, whether we can utilize an NN in a hybrid approach to rapidly sample candidate sequences during Rosetta protocols. More specifically, we wanted to test whether the existence of a moderate number of $3D_{opt}$ models is sufficient to teach an NN the energy landscape of a specific design problem. We implemented `Rosetta:MSF:NN` and used two benchmark data sets to confirm that an NN can learn the design-specific energy landscape of a protein design. We found that `Rosetta:MSF:NN` converges 3-times faster than our previous protocol and samples alternative areas of sequence space.

3.4 Results and Discussion

A 4-layer NN is able to deduce the energy landscape of a design problem from a feature-based representation of amino acid side chains.

Our basic assumption was that an NN can learn the energy landscape of a design problem, if it is trained with a small number of sequences, whose design-specific Rosetta scores are known. In order to test this hypothesis, we had to choose a suitable representation of amino acid residues, the number of residues fed into the NN, and a network topology. We decided to represent each amino acid residue aa_i by means of the five features (aaf_i) "volume", "polarity", "isoelectric point", "hydrophobicity", and "mean solvent accessibility" (Bogardt et al., 1980). The amino acid specific feature vectors are listed in Table 3.5. We restricted the number of residue positions fed into the NN to those n ones that are subjected to the design process, i.e. belong to the design shell. To deduce a prediction of the Rosetta score from these $5 * n$ features, we opted for a 4-layered, fully connected, feed-forward network. We chose two hidden layers, because these feed-forward networks generalize better than those with only one hidden layer (Thomas et al., 2017). As we expected mutual dependencies in the occupancy of the structurally adjacent positions of the design shell, we opted for a fully connected architecture, because it is are capable of learning any function (Ramsundar and Zadeh, 2018). The representation of the candidate sequences proposed an input layer consisting of $5 * n$ neurons and an output layer consisting of one neuron that presents the predicted score as a real value. Based on a grid search for the number of hidden neurons (Figure 3.6), we chose a first hidden layer with $10 + 5 * n/2$ neurons and a second hidden layer with $5 + 5 * n/4$ neurons (Figure 3.1 A).

To assess the predictive power of the NN, we utilized the outcome of a comprehensive `Rosetta:MSF:GA:enzdes` run, which was based on a genetic algorithm (GA). Briefly, a GA imitates the process of natural selection by maintaining a population of design sequences that are evolving for a number of generations. The MSD protocol Rosetta:MSF:GA generates candidate sequences by using the well-proven GA of Rosetta and computes their Rosetta total score averaged over all chosen states; for details see (Löffler et al., 2017) and references therein. MSD is superior over SSD even for seemingly simpler design task: Compared to SSD, a positive MSD

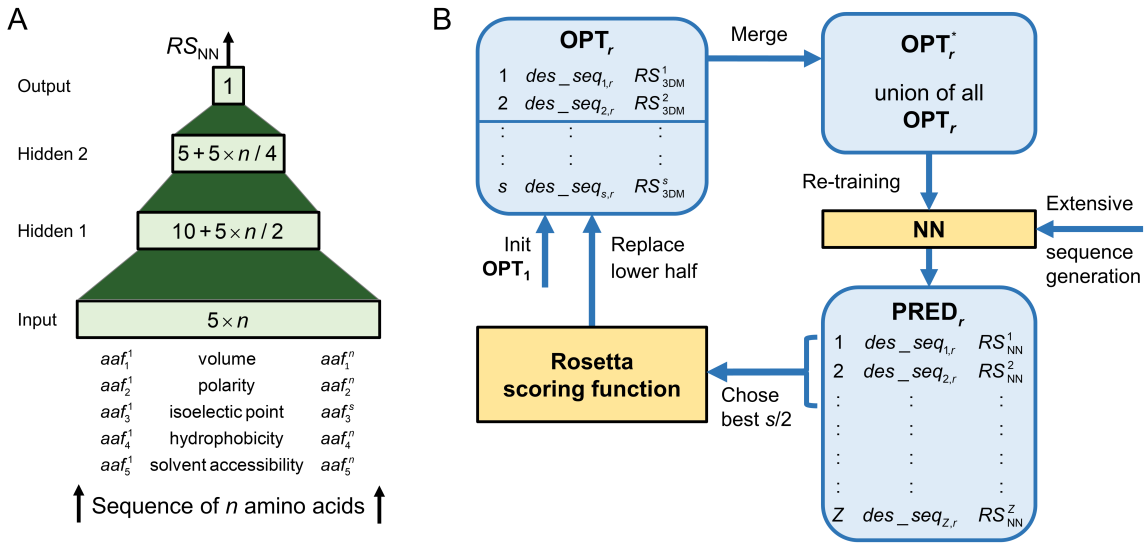

Figure 3.1: Architecture and data generation within `Rosetta:MSF:NN`. (A) The input layer of the NN contains $5 * n$ neurons that process the five-dimensional vectors representing the features aaf_i of n amino acid residues under study. The output layer consists of one neuron that determines the RS_{NN} value. The first hidden layer consists of $10 + 5 * n/2$ neurons and the second hidden layer of $5 + 5 * n/4$ neurons. All neurons of different layers are fully connected. (B) During all iterations of sequence optimization, `Rosetta:MSF:NN` administers a pool OPT_r of s sequences, whose RS_{3DM} value is known. The union OPT_r^* of these sequences is growing iteration-wise and used to re-train the NN that has an architecture as in (A). The re-trained NN is utilized for an extensive sequence scan that computes the RS_{NN} values for $Z = 2{,}000{,}000$ randomly generates sequences. For $s/2$ sequences possessing highest RS_{NN} values, the (costly) RS_{3DM} is computed. These sequences and their RS_{3DM} values constitute the lower half of the updated set OPT_{r+1}. For the first iteration, OPT_1 is initialized with s datasets.

approach similar to the one used here has more clearly identified for a dataset of 15 design problems the sequences that are optimal for all states (Vucinic et al., 2020).

For this on-going design project, we wanted to accommodate a new ligand in the binding site of the enzyme HisB by altering the occupancy of 14 residue positions. During 500 iterations of the GA, the Rosetta scores RS_{3DM} of 48,588 candidate sequences were computed, each based on a sequence-specific $3D_{opt}$ model. Thus, this dataset HisB_GA consisted of 48,588 tuples $des\_seq_j^+ = (aa_1^j, .., aa_i^j, ..aa_{14}^j, RS_{3DM}^j)$, where each 5-dimensional feature vector aa_i^j represents one amino acid i of a candidate sequence des_seq_j and RS_{3DM}^j is the normalized score (Eq 3.1) deduced by Rosetta from a specific $3D_{opt}$ model. We randomly selected 66% of the HisB_GA tuples and used them to train the NN; for details see Materials and Methods. After training, we utilized this NN to determine predicted Rosetta scores RS_{NN} for the remaining 33% of the feature vectors $des\_seq_j = (aa_1^j, .., aa_{14}^j)$. In Figure 3.2 A, RS_{NN}^j values are plotted versus the corresponding RS_{3DM}^j values. The clearly visible correspondence of both scores is evidenced by the high value of the Pearson correlation coefficient, which was 0.92 ($p \ll$ 1E-100) for the set of all test data. The average error determined for the test data set was not larger than 1.14 Rosetta Energy Units (REU). In order to confirm that the NN can efficiently approximate the energy landscape also for sequences differing in several mutations from test sequences, the test set was divided into subsets. A comparison of each test set sequence with the closest training set sequence was used to split the test set into subsets containing sequences with 1 - 11 mutations. As Table 3.1 indicates, the Pearson correlation coefficient decreased from 0.9 to 0.79, if the number of mutations increases from 1 to 11. In parallel, the error increased from 1.24 REU to 3.51 REU.

In order to assess the suitability of the chosen NN architecture for this regression problem, the performance was also determined for the training data. The resulting Pearson correlation coefficient of 0.94 (p $\ll$ 1E-100) and the average error of 1.21 REU are indicative of proper NN architecture and training. Moreover, the small differences of the corresponding performance values determined for training and test data indicates that the NN generalized quite well.

To generate the plot shown in Figure 3.2 B, we exclusively utilized the 239 HisB_GA tuples generated during the first iteration of the genetic algorithm. For the 80 test data, the Pearson correlation coefficient dropped to 0.31 (p = 3.7E-4) and the average error was 3.31 REU. These values indicate that one iteration of the GA is not sufficient to sample the complex energy landscape of this design problem in an adequate manner. In summary, we concluded that the feature-based representation of amino acid side chains and a trained NN are appropriate to model that part of the energy landscape (Rosetta scores) sampled by a GA during this problem-specific enzdes calculation.

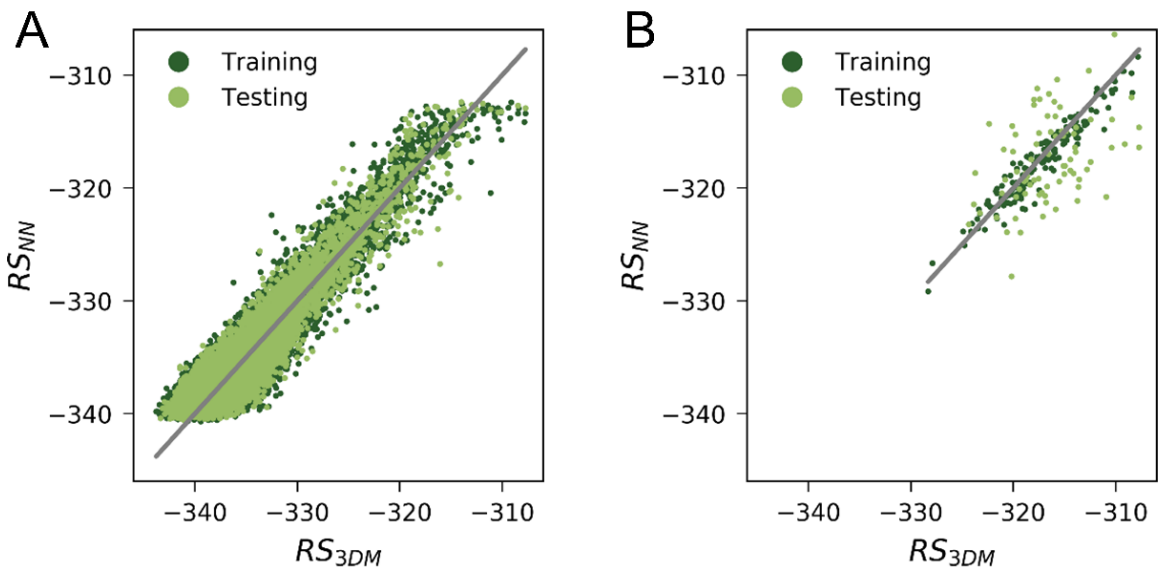

Figure 3.2: Performance of NNs. Plots of Rosetta scores RS^j_{3DM} deduced from $3D_{opt}$ models versus the Rosetta scores RS^j_{NN} predicted by an NN for an enzyme design. Scores are given in REUs and training and test data are represented as dark and light green dots, respectively. The gray lines indicate the diagonal, i.e. the position of perfect predictions. (A) Performance after the training with 32,392 sequences $des\_seq^+_j$ from 500 generations; the Pearson correlation coefficient was 0.92. (B) Performance after a training with the 158 $des\_seq^+_j$ generated during the first iteration of the GA; the Pearson correlation coefficient was 0.31.

Mutations	Sequences	Error [REU]	Correlation
1	14888	1.25	0.9
2	1198	1.38	0.9
3	45	1.19	0.96
8	2	ND	ND
9	12	2.75	0.82
10	46	4.36	0.37
11	5	3.51	0.79

Table 3.1: Performance of the NN for test sequences differing in 1 – 11 mutations from the closest training sequence. Each row characterizes a subset of test sequences each containing the number of mutations indicated in the first column. The number of test sequences belonging to this subset is listed in the second column. The error (difference of RS^j_{3DM} and RS^j_{NN}) is given in REU and the correlation is the value of the Pearson correlation coefficient. ND: Value not determined.

`Rosetta:MSF:NN`, a hybrid framework for MSD

The training with the HisB_GA tuples generated during the first GA iteration was too sparse for the complex energy landscape to be learnt by the NN. Figure 3.2 makes plausible that the RS^j_{3DM} values of several rounds of sequence optimization are required for an adequate representation of this energy landscape. However, one cannot predefine the number of iterations needed to find optimal sequences for a given design problem, as convergence speed is problem-specific. Guided by this constraint, we designed a novel protocol for sequence search (Figure 3.1 B). We decided to replace the genetic algorithm of `Rosetta:MSF:GA` with an NN-based sequence selection, but to continue keeping a specific set $\mathrm{OPT}_r = des\_seq^+_{j,r} | 1 \leq j \leq s$ of s optimal sequences for each iteration r. `Rosetta:MSF:NN` computes the RS^j_{3DM} score based on sequence-specific $3D_{opt}$ models only for newly generated elements of OPT_r. The union of all OPT_t tuples determined during the preceding iterations $t = 1, ..., r-1$ constitutes the continuously growing training set OPT^*_r, which is used to re-train the NN for iteration r.

The re-trained NN is then used to predict the scores RS^j_{NN} of an extensive number Z of randomly generated sequences; the default value is $Z = 2{,}000{,}000$. For those $s/2$ sequences with highest RS^j_{NN} values, `Rosetta:MSF:NN` computes the RS^j_{3DM} value based on sequence-specific $3D_{opt}$ models; for details see Materials and Methods. To compile the next set OPT_{r+1}, half of the $des\_seq_{j,r}$ tuples are replaced with newly generated ones and the corresponding RS^j_{3DM} values. Analogously to `Rosetta:MSF:GA`, the user has to execute the novel algorithm for several iterations until convergence is reached. This hybrid algorithm of sequence selection combines two major advantages:

1. Due to the high speed of the NN in computing RS^j_{NN} values, an extensive number of candidate sequences can be assessed, and the RS^j_{NN} values approximate the RS^j_{3DM} values quite well. The determination of RS^j_{3DM} values for all possible candidate sequences is currently not feasible due to the computational costs of the $3D_{opt}$ models.
2. By merging the iteratively generated OPT_r sets that consist of all hitherto found well-scoring sequences, the prediction quality of the NN increases continuously due to the denser sampling of the problem-specific energy landscape.

Encouraged by these promising initial results, we wanted to answer the following four questions in order to assess the potential benefit of integrating an NN into `Rosetta:MSF`:

1. Does the use of NNs reduce the number of iterations needed to identify optimal candidate sequences?
2. How robust is this approach with respect to the chosen features and scoring functions?
3. Does the NN based approach find sequences with better Rosetta scores?
4. Does the extensive sampling of the sequence space lead to candidate sequences not found by the GA?

`Rosetta:MSF:NN` converges 3-times faster than `Rosetta:MSF:GA` and enumerates better scoring sequences

For a comprehensive comparison of `Rosetta:MSF:NN` with the older `Rosetta:MSF:GA` protocol, we utilized the previously introduced benchmark MD_EnzBench (Löffler et al., 2017). This set has been compiled to test the ability of protocols to rebuild the ligand-binding site of 16 proteins $prot_k$. For each $prot_k$, 10 specifically prepared conformations that served as states of an MSD protocol have been deduced by means of molecular dynamics simulations. Each conformation contains the bound ligand and in order to increase the difficulty of the design task, all residues of the design shells were replaced with alanines; for details see Materials and Methods.

For these 16 design problems, `Rosetta:MSF:GA:enzdes` and `Rosetta:MSF:NN:enzdes` were executed as 10-state MSD protocols for 100 iterations and the NNs were re-trained during each iteration as described. In order to follow the convergence of the two design processes, the mean Rosetta score $RS(\mathrm{OPT}_r)$ (Eq 3.2) was determined for each iteration r. $RS(\mathrm{OPT}_r)$ is the mean of the RS^j_{3DM} scores of all sequences related to the iteration-specific set OPT_r. In Figure 3.3 the protocol-specific convergence is shown for four examples and all 16 designs are documented in Figure 3.7. As expected, both protocols find sequences that score considerably better than the native ones. Most interestingly, the $RS_{\mathrm{NN}}(\mathrm{OPT}_r)$ values dropped more rapidly than the $RS_{\mathrm{GA}}(\mathrm{OPT}_r)$ values. In order to assess the kind of convergence numerically, we determined the normalized area above (NAA), the $RS_{\mathrm{GA}}(\mathrm{OPT}_r)$, and the $RS_{\mathrm{NN}}(\mathrm{OPT}_r)$ values in analogy to a ROC curve (Davis and Goadrich, 2006). A NAA value gets close to 1.0 if the curve drops vertically to its minimum during the first few iterations. In Table 3.2 the $\mathrm{NAA}_{\mathrm{NN}}$ and $\mathrm{NAA}_{\mathrm{GA}}$ values are listed for all 16 designs as well as the numbers n and ss of design shell residues and second shell residues; for their definition see (Richter et al., 2011). The comparison of the design-specific $\mathrm{NAA}_{\mathrm{NN}}$ and $\mathrm{NAA}_{\mathrm{GA}}$ values confirmed that `Rosetta:MSF:NN:enzdes` converged much faster than the GA based protocol: The mean $RS_{\mathrm{NN}}(\mathrm{OPT}_r)$ value was 0.91, whereas the mean $RS_{\mathrm{GA}}(\mathrm{OPT}_r)$ value was 0.70. Although trained with relatively few examples during the first iterations, `Rosetta:MSF:NN:enzdes` found more rapidly sequences with low Rosetta scores than the GA based protocol did.

To estimate the performance gain, we determined for each design k the first iteration r^*_k, whose $RS_{\mathrm{NN}}(\mathrm{OPT}_r)$ value reached the $RS_{\mathrm{GA}}(\mathrm{OPT}_{100})$ value of iteration 100 of `Rosetta:MSF:GA:enzdes`. As Table 3.2 shows, the r^*_k values varied between 11 and 99, which most likely reflects the differing complexity of the design-specific energy landscapes. However, on average 33.5 iterations of the NN based protocol were sufficient to find sequences that scored as good as those created by the GA based protocol in iteration 100. We conclude that `Rosetta:MSF:NN:enzdes` converges on average three times faster than the GA based protocol. A fourfold gain results, if one compares for both approaches the first iterations $r^{native}_{k,\mathrm{GA}}$ and $r^{native}_{k,\mathrm{NN}}$ that generated a sequence set OPT_r having native-like Rosetta energies. On average, the GA needed 17, and the NN approach not more than 4.2 iterations; compare Table 3.2. Interestingly, convergence speed

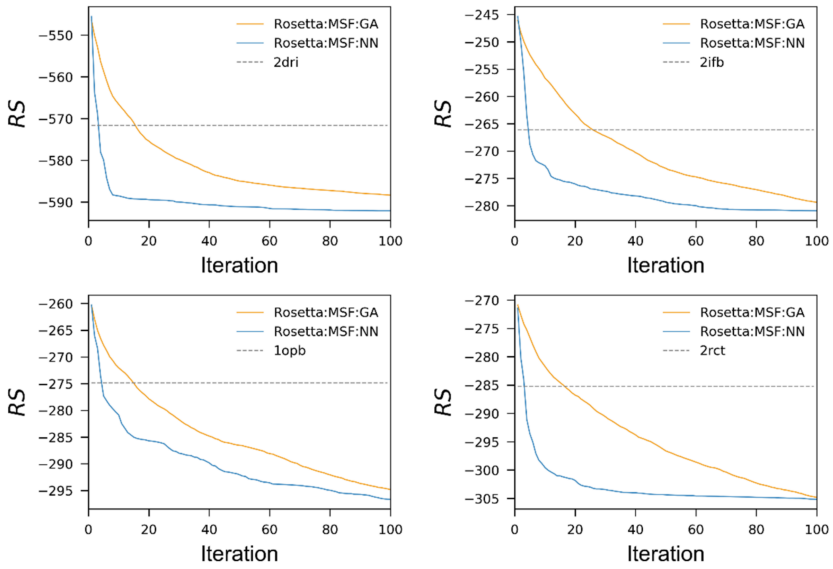

Figure 3.3: Convergence of `Rosetta:MSF:GA:enzdes` and of `Rosetta:MSF:NN:enzdes`. For each of the four designs, the mean Rosetta scores $RS(\mathrm{OPT}_r)$ were determined in REUs for each iteration r = 1 - 100 and plotted. The blue lines represent the $RS_{\mathrm{NN}}(\mathrm{OPT}_r)$ and the orange lines the $RS_{\mathrm{GA}}(\mathrm{OPT}_r)$ values. The dashed, horizontal line marks the score of the relaxed native protein; the PDB-ID of the corresponding protein is indicated. The plots for all 16 designs are shown in Figure 3.7.

seems uncorrelated with the number of the design shell residues; none of the three Spearman rank order correlation tests of n with $\mathrm{NAA}_{\mathrm{NN}}$, $\mathrm{NAA}_{\mathrm{GA}}$, and r_k^* gave a statistically significant result.

The comparison of the $\mathrm{NAA}_{\mathrm{NN}}$ and $\mathrm{NAA}_{\mathrm{GA}}$ values and the plots shown in Figure 3.7 indicate that `Rosetta:MSF:NN:enzdes` finds for a given number of iterations better-scoring sequences than the GA based protocol. If both protocols were executed for 100 iterations, the $RS_{\mathrm{NN}}(\mathrm{OPT}_{100})$ value was superior to the $RS_{\mathrm{GA}}(\mathrm{OPT}_{100})$ value in 14 of the 16 designs; compare Figure 3.7. These findings confirm that the NN can beneficially exploit the extensive sampling to identify sequences with superior Rosetta scores.

The performance of `Rosetta:MSF:NN` does not depend on a specific representation of amino acid residues or a specific scoring function

In order to assess the robustness and the applicability of `Rosetta:MSF:NN`, we performed additional redesign experiments of the four proteins with PDB-IDs 2dri, 2ifb, 1opb, and 2rct. First, we wanted to make plausible that the performance of `Rosetta:MSF:NN` does not critically depend on the chosen residue representation (Table 3.5) and is robust to limitations of feature tables. For the experiment named NN_4, only the features “volume”, “polarity”, “isoelectric point”, and “hydrophobicity” of the chosen residue representation were used. For the experiment NN_AT, an alternative feature table representing each residue with the six properties

PDB-ID	n	ss	$\mathbf{NAA_{GA}}$	$\mathbf{NAA_{NN}}$	r_k^*	$r_{k,\mathrm{GA}}^{native}$	$r_{k,\mathrm{NN}}^{native}$
2uyi	23	35	0.71	0.94	13	28	4
2rde	20	35	0.66	0.93	21	10	3
2rct	22	51	0.68	0.93	74	17	4
2qo4	22	40	0.66	0.89	17	14	5
2q2y	23	34	0.78	0.91	99	24	5
2ifb	22	54	0.70	0.90	50	26	5
2dri	19	42	0.76	0.95	9	16	4
2b3b	17	46	0.71	0.94	14	22	4
1urg	19	40	0.67	0.88	36	21	5
1pot	19	41	0.73	0.93	11	3	2
1opb	22	50	0.67	0.81	78	15	5
1nq7	28	57	0.70	0.92	24	30	7
1n4h	25	51	0.69	0.91	26	15	5
1j6z	27	45	0.77	0.93	33	11	3
1hsl	19	42	0.66	0.92	15	8	2
1fzq	20	29	0.70	0.93	16	17	4
Average	**22**	**43**	**0.70**	**0.91**	**33.5**	**17.3**	**4.2**

Table 3.2: Performance comparison of the GA and NN based protocols. $\mathrm{NAA_{GA}}$, $\mathrm{NAA_{NN}}$, r_k^* , $r_{k,\mathrm{GA}}^{native}$, $r_{k,\mathrm{NN}}^{native}$ and values were determined for each of the 16 proteins (indicated by their PDB-ID) from the MD_EnzBench set. n is the number of design shell residues and ss the number of second shell residues. For details, see Materials and Methods.

"polarity", "accessible surface area", "hydrophilicity", "polarizability", "hydrophobicity", and "solvation free energy" (Yousef and Charkari, 2015) was utilized. The analysis of the plots shown in Figure 3.8 confirms that the two alternative representations of residues have no drastic effect on the performance. In all cases the NN based algorithm converged faster than the GA based one. Alternatively, we used a one-hot encoding of residues, i.e. a 20-column vector having a "1" at the position representing the current residue and "0" for the other 19 ones. This representation requires a different NN architecture; see Text 3.6 for details. As Figure 3.9 indicates, this residue representation leads to lower Rosetta scores than the feature-based approach. However, this network has far more parameters than the feature-based NNs and is thus more prone to overfitting if not trained with special care. It is beyond the scope of this manuscript to identify the optimal training procedure of this NN, which would be required for a fair comparison of the performance values.

For all designs presented here, we utilized the *soft-rep* scoring function. In order to make plausible that the convergence of `Rosetta:MSF:NN` is not dependent on a specific scoring system, another eight redesign experiments were performed by using two alternative scoring functions: A clear convergence gain of the NN- over the GA-based protocol is confirmed by Figure 3.10 for the *talaris*, and by Figure 3.11 for the *ref2015* Rosetta scoring function (Alford et al., 2017).

In summary, these results emphasize the capability of the NNs to learn a problem-specific energy landscape modelled by means of different residue representations and scoring functions.

The extensive sampling of the sequence space finds alternative minima

With default values, our NN based approach allows per iteration the assessment of 2,000,000 alternative sequences; in contrast, the GA based approach generates less than 120 novel candidates per generation. We were interested to find out, whether this rigorous widening of search space sampling had a pronounced effect on the composition of the `enzdes` outcome. It is difficult to compare precisely the composition of two sequence sets, thus we opted for an approximate approach, namely the comparison of amino acid frequency tables.

To characterize trends, we wanted to compare for each design $prot_k$ of MD_EnzBench the composition of the sequence sets $\mathrm{OPT}^{\mathrm{NN}}_{r,k}$ ($r = 1$ - 100) generated by the NN based protocol with a fixed reference sequence set $\mathrm{OPT}^{\mathrm{GA}}_{r*,k}$ consisting of sequences generated by the GA based protocol during iteration $r*$; compare Table 3.2. If both protocols sample the same area of sequence space, the amino acid composition of the NN based sequence outcome should iteratively approach the amino acid frequencies of the GA based reference.

To begin with, we determined for each design $prot_k$ of MD_EnzBench the reference $\mathrm{OPT}^{\mathrm{GA}}_{r*,k}$ that served as a reference sequence set because the score was comparable to that of the value of $\mathrm{OPT}^{\mathrm{NN}}_{100,k}$. In order to approximate compositional differences, we deduced for each of the n design shell residues pos a reference frequency table $ft^{\mathrm{GA}}_{k,pos} = f^{\mathrm{GA}}_{k,pos}(aa_i)(i = 1 - 20)$ from

$\mathrm{OPT}^{\mathrm{GA}}_{r*,k}$ and further frequency tables $ft^{\mathrm{NN}}_{r,k,pos}$ from all of the 100 $\mathrm{OPT}^{\mathrm{NN}}_{r,k}$ sets. Euclidean distances $euk(ft^{\mathrm{NN}}_{r,k,pos}, ft^{\mathrm{GA}}_{k,pos})(r = 1 - 100)$ were computed and their mean $dist_{r,k}$ determined according to Eq 3.4. In order to assess the progression of compositional differences, these were divided into two groups $\mathrm{NN}^k_1 = \{\mathrm{dist}_{r,k}|r = 1 - 50\}$ and $\mathrm{NN}^k_2 = \{\mathrm{dist}_{r,k}|r = 51 - 100\}$ consisting of the outcome of the first and second half of the iterations. In order to generate "null model" distributions R^k_1 and R^k_2, the $ft^{\mathrm{NN}}_{r,k}$ frequencies were shuffled prior to the computation of their Euclidean distances to ft^{GA}_k. For each design $prot_k$ these four distributions of distances were visualized by means of a box plot. Figure 3.4 shows that in all four cases presented also in Figure 3.3, the mean of the distances sampled with the two null models R^k_1 and R^k_2 was close to the maximally possible distance of $\sqrt{2}$. All NN^k_1 and NN^k_2 distributions had lower, but markedly distances to ft^{GA}_k (mean > 0.4). This suggests that the GA and the NN based protocols concentrate on different regions of sequence space. Except for the design 1opb, the NN^k_1 spread (first half of iterations) of the distances was much larger than the NN^k_2 spread, which indicates the convergence to a minimum. For 2dri and 2ifb, the GA and NN based protocols found sequences that resemble each other more than those found for the designs 1opb and 2rct. The broad spread of amino acid frequencies observed in $\mathrm{NN}^{1\mathrm{opb}}_2$ of design 1opb indicates that `Rosetta:MSF:NN:enzdes` did not find a minimum during 100 iterations in contrast to the other three cases; compare 1opb plot in Figure 3.3. The remaining 12 designs have similar variations of their box plots patterns (Figure 3.12): In seven of the 16 designs, the mean of the NN^k_2 distances was > 75. In summary, we concluded that the GA and the NN protocols find different minima of sequence space in approximately half of the designs.

Epistatic effects may cause alternative minima

The region of sequence space optimal for a given protein backbone seems relatively limited to the neighborhood of the native sequence (Kuhlman and Baker, 2000). Although scoring better than native ones, the sequences generated by the GA and the NN for the design shells were markedly different, which prompted us to find an explanation.

One reason could be that the width of sequence sampling differs markedly between the GA and the NN protocol. In order to compare the sampled sequences space, we determined sequence variability of the $s/2$ sequences ($cand^{\mathrm{GA}}_r$, $cand^{\mathrm{NN}}_r$; see Materials and Methods) that were generated by the GA or the NN and used to replace the bottom half of OPT_r during each iteration r. Each sequence from a $cand^*_r$ set was compared to the most similar one from OPT_r to determine the number of newly introduced mutations. Sequence variability deduced from 100 iterations is shown in Table 3.3 and that of the first 10 iterations in Table 3.4, because the initial iterations are crucial for NN training. Both distributions demonstrate that the GA generated the $cand^{\mathrm{GA}}_r$ sequences mainly by introducing single point mutations, whereas the NN introduced up to 18 mutations to generate a $cand^{\mathrm{NN}}_r$ sequence. These findings indicate that the NN approach samples sequence space much broader than the GA. Moreover, these results make

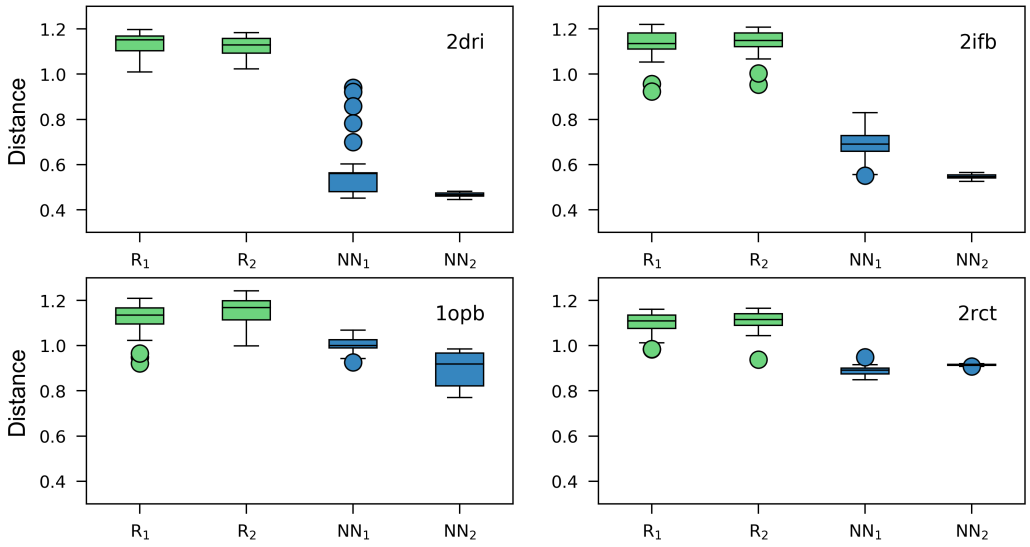

Figure 3.4: Amino acid frequency distributions for the outcome of the first and second half of design protocols. The boxplots represent the distributions of mean distances between amino acid frequencies tables $ft_{r,k,pos}$ related to the iterations of the NN based protocol and a reference table $ft^{\mathrm{GA}}_{k,pos}$ from the GA based protocol; compare Eq 3.3 and Eq 3.4. The group NN^k_1 contains the distributions related to the first 50 and NN^k_2 the distributions of the second 50 iterations. For the computation of R^k_1 and R^k_2, the values of each table $ft^{\mathrm{NN}}_{r,k,pos}$ were shuffled; for details see Materials and Methods. The boxplots show the results for the designs k = 2dri, 2ifb, 1opb, and 2rct.

clear that the NN is trained with a broad sequence set, because the RS^{j}_{3DM} values of these novel and diverse $cand^{\mathrm{NN}}_{r}$ sequences are used to train the NN in the next iteration $r+1$.

It is known that mutual dependencies in the occupancy of residue positions drastically affect the fitness landscape as has been confirmed *in silico* (Weinreich et al., 2013) and experimentally (Lunzer et al., 2010; Tamer et al., 2019). As a consequence, high-order epistasis constrains the adaptive pathways that can be followed by evolution (Yang et al., 2019), because each given occupancy of residue positions severely restrains the subsequently tolerated mutations. The design shells used here consist of a small number of highly constrained residue positions and the above findings strongly suggest that their occupancy is mutually dependent. Analogously to the evolution of native proteins, each design protocol induces a specific chronological order of residue substitutions and a trajectory that is – under these circumstances – most likely inaccessible, if the order of mutations is different. Thus, if GA and NN choose during the first iterations different residues for some key positions, the sequence space available to subsequent candidates will be different, if epistasis is dominant. In order to illustrate the existence of epistatic effects, we analyzed mutual dependencies of residue pairs in four design shells. Algorithms identifying correlated mutations need relatively large MSAs, require a certain variability in residue frequencies and are thus not capable to assess conserved residues (Martin et al., 2005), which are key to the outcome of design protocols. This is why we manually compared sequence logos resulting from GA and NN generation 100 and searched for positions that were occupied by strikingly different residues. By inspecting the 3D structure of corresponding design candidates, consequences, i.e. pairwise mutual dependencies, of these choices were made plausible. Figure 3.5 illustrates differences in the orchestration of four design shells generated by GA and NN protocols, which are all valid with respect to orientation and Rosetta scores: In the design shell of 2dri, the GA chose at position 15 preferentially Phe and at position 235 His. In contrast, the NN selected at position 15 Trp and at position 235 Thr. In the design shell of 2ifb, the occupancies of residue positions 70 and 72 are mutually dependent: The GA prefers for position 70 Tyr and for position 72 Ser. The NN chose at position 70 Trp and at position 72 Asp. In the design shell of 1opb, the GA selected two medium sized residues, namely at positions 53 Ile and at position 40 Leu. In contrast, the NN introduced at position 53 a bulky Trp or Phe residue, which was accompanied at position 40 by a small Val residue. In the design shell of 2rct, the GA chose for position 37 Leu or Tyr, and for position 61 Phe or Leu, i.e. combinations of a large and a small residue. In contrast, the NN preferred at position 37 Trp and at position 61 Ala or Gly.

In summary, these analyses illustrate strong mutual dependencies in the occupancies of design shell residues. As design shells are generated by mutating residues in a randomly chosen chronology, epistasis directs the protocols to different regions of sequence space.

Mutations	2dri		2ifb		1opb		2rct	
	GA	NN	GA	NN	GA	NN	GA	NN
1	11581	7072	11532	5944	11624	3677	11640	6663
2		3722		4106		4806		3362
3		669		931		1948		1070
4		127		301		653		326
5		66		116		249		147
6		44		64		122		66
7		23		41		70		37
8		21		30		41		27
9		7		22		31		18
10		11		24		16		14
11		12		17		23		10
12		1		18		12		10
13		1		18		15		10
14		3		11		12		5
15		2		12		14		5
16				3		30		4
17				4		47		2
18						15		1

Table 3.3: Comparison of sequence heterogeneity in candidate sequences added to OPT$_r$ during 100 iterations. The table lists numbers of $cand_r^{\mathrm{GA}}$ and $cand_r^{\mathrm{NN}}$ sequences grouped according to their differences (number of mutations) to the most similar sequence from OPT$_r$. For this analysis, 100 iterations of the designs for 2dri, 2ifb, 1opb, and 2rct were analyzed.

Mutations	2dri		2ifb		1opb		2rct	
	GA	**NN**	**GA**	**NN**	**GA**	**NN**	**GA**	**NN**
1	1067	177	1063	15	1064	0	1059	15
2		391		224		163		418
3		224		280		266		247
4		91		191		149		127
5		64		99		120		79
6		43		63		67		49
7		23		40		58		32
8		21		30		35		26
9		7		22		29		17
10		11		24		16		14
11		12		17		23		10
12		1		18		12		10
13		1		18		15		10
14		3		11		12		5
15		2		12		14		5
16				3		30		4
17				4		47		2
18						15		1

Table 3.4: Comparison of sequence heterogeneity in candidate sequences added to OPT$_r$ during the first 10 iterations. The table lists numbers of $cand_r^{\mathrm{GA}}$ and $cand_r^{\mathrm{NN}}$ sequences grouped according to their differences (number of mutations) to the most similar sequence from OPT$_r$. For this analysis, the first 10 iterations of the designs for 2dri, 2ifb, 1opb, and 2rct were analyzed.

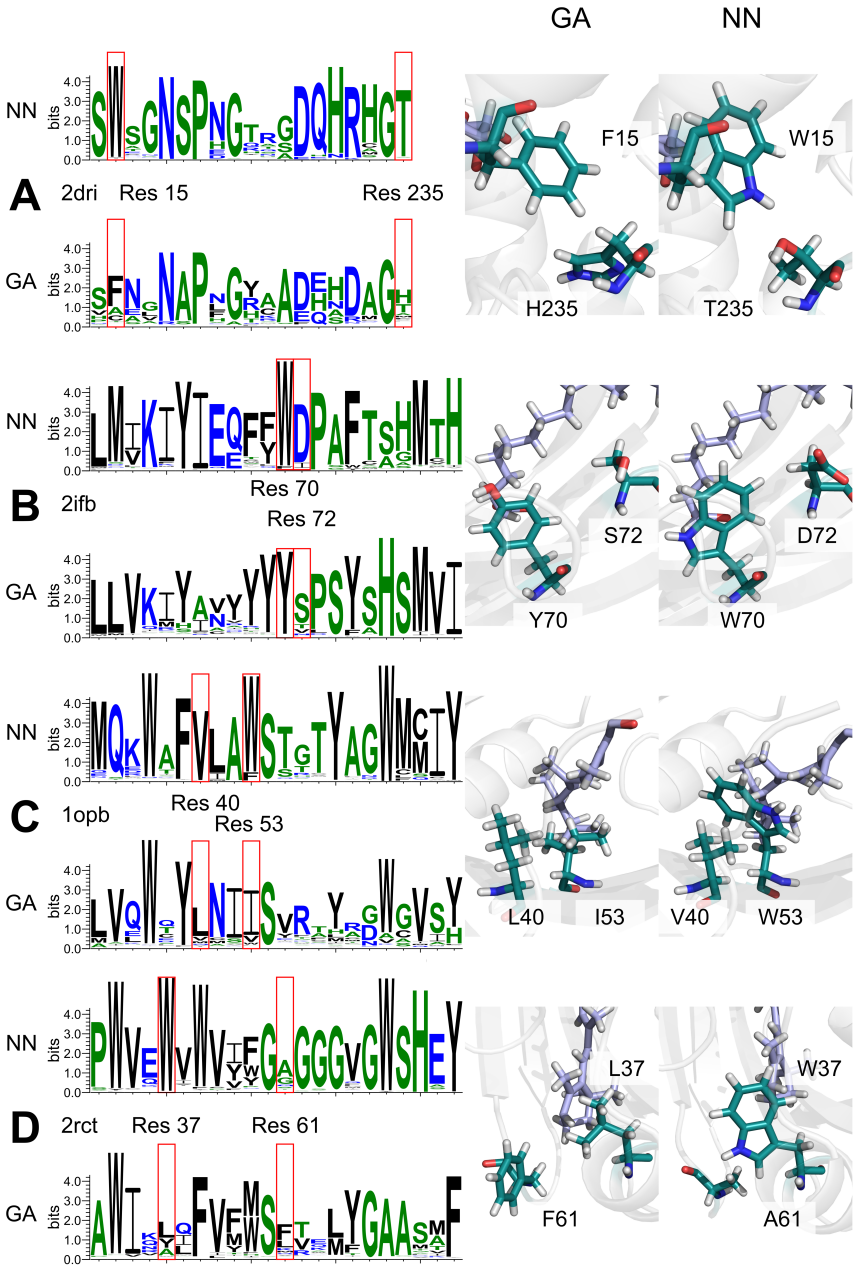

Figure 3.5: Residue occupancies generated by `Rosetta:MSF:GA` and `Rosetta:MSF:NN` for four design shells. For each design shell, the GA- and NN-specific logos resulting from the 239 sequences of generation 100 are given. In each case, one residue pair that is most likely affected by epistatic effects is indicated with red boxes and the 3D orientation of the residues is shown as sticks for the GA (left) and the NN (right) design solution. Four designs were analyzed: (A) 2dri, (B) 2ifb, (C) 1opd, (D) 2rct. The chosen residues are shown in turquoise and the ligand in pale blue; the rest of the protein is represented as a white cartoon. For this analysis, the *talaris* scoring function was used.

For NNs, a limited number of samples map the complexity of a problem-specific energy landscape in sufficient detail

Generally, the reconstruction of an all-atom model from an incomplete representation of a protein is a challenging problem (Badaczewska-Dawid et al., 2020). Thus, at first glance it seems surprising that a feature-based representation of candidate sequences suffices to predict the correct RS-values with an average error of less than 1.5 REUs; compare Figure 3.2. To determine an RS_{3DM}-value, Rosetta has to build a 3D model and to find a combination of rotamers that is suitable for all residues of the design shell. Why is an NN that lacks the assessment of a three-dimensional representation, so successful in an `enzdes` protocol? Although the NN does not explicitly process three-dimensional information, it can learn the energy landscape, because the RS_{3DM}-values of the training data implicitly transfer three-dimensional information into the scoring function learned by the NN.

Another example for the beneficial mapping of structure space by means of an NN is `refineD`, which is aimed at protein structure refinement. This program utilizes additional restraints integrated into a Rosetta all-atom energy function that have been deduced by means of a deep convolutional neural field from the starting structure. `refineD` outperformed unrestrained relaxation strategies, most likely because these restraints guide conformational sampling (Bhattacharya, 2019).

Furthermore, the performance of our NN and our analysis of residue pairs suggests that the orchestration of the design shells with amino acid residues and their orientation is severely biased. This assumption is supported by recent statistical findings related to residue preferences. The algorithm `NEPRE` assesses successfully the quality of three-dimensional protein models. It is based on a scoring system for residue neighborhood preferences deduced from 14,647 PDB structures. It turned out that certain residues exhibit strong preferences for their neighboring residues and their relative positions (Liu et al., 2020). `trRosetta` generates with high quality the three-dimensional structure of proteins based on predictions for inter-residue contacts, distances, and residue orientations. These predictions originate from a deep residual network that analyzes protein-specific MSAs. For the 31 FM targets of the CASP13 contest, the mean TM-score, which signals the correspondence with the native structure, has been 0.625. The score had dropped only marginally to 0.592, if residue orientation has been ignored during structure prediction (Yang et al., 2020). Along this line, a restricted number of residue orientations has been observed at certain positions in more than 2600 antibody structures (Leem et al., 2018). Taken together, these findings support the idea that `Rosetta:MSF:NN:enzdes` performs well, because the orchestration of the design shell with a restricted number of amino acid residues has a stronger effect on scores than the choice of rotamers.

3.5 Materials and Methods

Dataset HisB_GA used for the initial performance test

The dataset HisB_GA consisted of 48,588 tuples $des\_seq_j^+ = (aa_1^j, ..., aa_{14}^j, RS_{3DM}^j)$ that were generated by means of `Rosetta:MSF:GA:enzdes` during 500 GA generations of an ongoing design project. Each 5-dimensional feature vector aa_i^j represented one amino acid i of a candidate sequence. This experiment was aimed at the redesign of the ligand-binding site of the bifunctional enzyme HisB-N from *Escherichia coli* that hydrolyses L-histidinol phosphate to L-histidinol and phosphate as well as O-phospho-L-serine to L-serine and phosphate. For this MSD approach, 11 states were used that represented slightly different conformations generated by means of a short (1 ns) molecular dynamic simulation seeded with the structure of the N-terminal domain of *Escherichia coli* HisB (chain A of PDB-ID 2fpu).

Benchmark data set MD_EnzBench

The data set MD_EnzBench has been generated previously for benchmarking ligand binding design based on Rosetta:MSF (Goldenzweig et al., 2016). It has been deduced from molecular dynamics (MD) simulations of length 10 ns generated with YASARA (version 14.7.17) and the YAMBER3 force field that has been parameterized to produce crystal structure-like protein coordinates (Krieger et al., 2004). MD_EnzBench consists of 16 proteins $prot_k$ with bound ligand taken from the scientific sequence recovery benchmark of Rosetta (Nivón et al., 2014). To introduce conformational flexibility during the MD simulations, the ligand has been removed and for each of the 16 apoproteins, 1000 conformations have been sampled at an interval of 10 ps. As a structural basis for the subsequent MSD protocol, protein conformations have been saved every 1 ns and used as states. After sampling, the native ligands have been re-introduced in all conformations of the respective apoproteins by means of `PyMOL:superpose` (Schrödinger, LLC, 2015). The design and repack shells of all enzymes are listed in (Shah et al., 2007). All design shell residues have been replaced with alanine and prior to design, all conformations have been energy-minimized by means of `Rosetta:fastrelax` with backbone constraints.

Design and implementation of the NN

The 4-layered architecture of the NN is shown in Figure 3.1 A. The input layer consists of $5 * n$ neurons, each of which is supplied with one of five features aa_i^j of the n design shell residues whose composition constitutes the candidate sequence under study. The first hidden layer consists of $10 + 5 * n/2$ neurons and the second hidden layer of $5 + 5 * n/4$ neurons. The output layer consists of one neuron that computes the score RS_{NN} as a real value. Each layer is fully connected with the previous layer and no bias is used in any layer.

The NN was created using the python package *Keras* version 2.2.4 and *TensorFlow* 1.12.0 as back-end. For initialization, a `RandomNormal` kernel was used in all layers. Both hidden layers utilize a `tanh` and the output layer a `linear` activation function. The network was optimized by means of a stochastic gradient descent with momentum. Prior to training, the RS_{3DM} values were converted to z-scores. The model was trained for 100 epochs in incremental mode.

Design and data flow of `Rosetta:MSF:NN`

In order to allow for a fair comparison with the previously introduced GA-based approach `Rosetta:MSF:GA`, the novel NN-based approach `Rosetta:MSF:NN` administers also a set OPT_r of $s = 239$ sequences, whose RS_{3DM} values are used for their ranking; see Figure 3.1 B. These des_seq_j sequences represent the amino acid residues chosen for the positions of the design shell under study and for each des_seq_j, the RS_{3DM} value is computed by means of the chosen Rosetta scoring function. Initially, `Rosetta:MSF:NN` generates the set OPT_1 consisting of the given seed sequences and mutants, each with a randomly introduced single point mutation (Löffler et al., 2017). During each iteration r, OPT_r is added to OPT_r^*, which contains the shuffled union of all so far chosen des_seq_j data. The updated set OPT_r^* is used to re-train the NN, which is then utilized to assess an extensive set of Z novel sequences. The seed for these sequences is the iteration-specific best scoring sequence, which gets mutated at a random number of positions to randomly chosen amino acid residues. Per default, Z = 2,000,000 random sequences are generated, which get mutated at a random number of positions (1 – number of design shell residues) to randomly chosen amino acid residues and are fed into the NN to compute their score RS_{NN}. The dataset PRED_r contains the Z sequences $des\_seq_{j,r}$ ranked according to their RS_{NN}^j values. For the $s/2$ best scoring candidate sequences $cand_r$, the chosen Rosetta scoring function is used to compute their RS_{3DM}^j values. To prepare the set OPT_{r+1} utilized in the next iteration, `Rosetta:MSF:NN` replaces the bottom half of the OPT_r sequences with the $s/2$ candidates $cand_r$. These iterations continue until a user-defined stopping criterion is satisfied. As `Rosetta:MSF:NN` was utilized for an MSD protocol, a separate NN was trained for each individual state of each protein.

Assessing design performance

To determine the score of a candidate sequence des_seq_j for an MSD protocol, the mean Rosetta score was computed:

$$RS_{\mathrm{3DM}}^j(des\_seq_j) = \frac{1}{m}\sum_{i=1}^{m} ts_i(des\_seq_j) \tag{3.1}$$

Here, m is the number of states and $ts_i(des\_seq_j)$ is the Rosetta total score for a sequence given a state i. In all equations, Rosetta scores are given in REUs.

To assess the fitness of a sequence set OPT_r of an iteration r, the mean of all s $RS^j_{\mathrm{3DM}}(des\_seq_j)$ values was determined:

$$RS(OPT_r) = \frac{1}{s}\sum_{j=1}^{s} RS^j_{\mathrm{3DM}}(des\_seq_{j,r}) \tag{3.2}$$

To distinguish the values related to the NN and GA based protocol, they were designated $RS_{NN}(\mathrm{OPT}_r)$ and $RS_{GA}(\mathrm{OPT}_r)$, respectively.

For the determination of the design-specific normalized areas above (NAA) the $RS_{GA}(\mathrm{OPT}_r)$ and the $RS_{NN}(\mathrm{OPT}_r)$ values, the areas flanked by the specific $RS(\mathrm{OPT}_1)$ and $RS(\mathrm{OPT}_{100})$ values were calculated. For their normalization between 0.0 and 1.0, the lowest value reached after 100 generations by either of the two protocols was used.

Following trends of sequence sampling

To characterize trends of sequence sampling, for each design $prot_k$ of MD_EnzBench the composition of the sequence sets $\mathrm{OPT}^{\mathrm{NN}}_{r,k}$ (r = 1 - 100) generated by the NN based protocol was compared with a well-defined reference sequence set $\mathrm{OPT}^{\mathrm{GA}}_{r*,k}$. It consists of sequences generated by the GA based protocol during iteration r^*_k; compare Table 3.2.

To begin with, for each design $prot_k$ the reference set $\mathrm{OPT}^{\mathrm{GA}}_{r*,k}$ was identified. This set represents the earliest generation r whose score value was most similar to $\mathrm{RS_{NN}}(\mathrm{OPT}^{\mathrm{NN}}_{100,k})$, which was generated by `Rosetta:MSF:NN:enzdes` during the last iteration of the NN based protocol. Utilizing the amino acid composition of the related 239 sequences $des\_seq_{j,r}$, a normalized frequency table $ft^{\mathrm{GA}}_{k,pos} = f^{\mathrm{GA}}_{k,pos}(aa_i)(i = 1-20)$ was deduced for each of the n residue positions pos of the design shells. Analogously, frequency tables $ft^{\mathrm{NN}}_{r,k,pos}$ were derived from the 100 $\mathrm{OPT}^{\mathrm{NN}}_{r,k}$ sets created during all iterations of the NN based protocol. Position-specific Euclidean distances $euk_r(ft^{\mathrm{NN}}_{r,k,pos}, ft^{\mathrm{GA}}_{k,pos})$ were computed according to:

$$euk_r(ft^{\mathrm{NN}}_{r,k,pos}, ft^{\mathrm{GA}}_{k,pos}) = \sqrt{\sum_{i_1}^{20}(ft^{\mathrm{NN}}_{r,k,pos}(aa_i) - ft^{\mathrm{GA}}_{k,pos}(aa_i))^2} \tag{3.3}$$

To assess the mean amino acid variation for each iteration r and each design k, the Euclidean distances were averaged according to:

$$dist_{r,k} = \frac{1}{n}\sum_{pos=1}^{n} euk_r(ft^{\mathrm{NN}}_{r,k,pos}, ft^{\mathrm{GA}}_{k,pos}) \tag{3.4}$$

$dist_{r,k}$ values were divided in two groups $\mathrm{NN}^k_1 = \{\mathrm{dist}_{r,k} \mid r = \text{1-50}\}$ and $\mathrm{NN}^k_2 = \{\mathrm{dist}_{r,k} \mid r = \text{51-100}\}$ to distinguish the composition of the sequences generated during the first and second half of the iterations.

To compute "null model" distributions R_1^k and R_2^k that served as references, the frequencies were shuffled table-wise for each of the $100 * n$ $ft_{r,k,pos}^{NN}$ sets. Afterwards, Euclidean distances to $ft_{k,pos}^{GA}$ were computed according to Eq 3.3 and their mean was determined according to Eq 3.4. The four $prot_k$ specific distributions of $dist_{r,k}$ values were visualized by means of a box plot.

Acknowledgments

This work was supported by a grant of the Deutsche Forschungsgemeinschaft to RM (ME 2259/4-1).

Author Contributions

Conceptualization: JN EL RM.
Data curation: JN.
Formal analysis: JN EL RM.
Funding acquisition: RM.
Investigation: JN.
Methodology: JN.
Project administration: RM.
Resources: RM.
Software: JN.
Supervision: RM.
Validation: JN EL RM.
Visualization: JN RM.
Writing ± original draft: JN RM.
Writing ± review & editing: JN EL RM.

3.6 Supporting Information

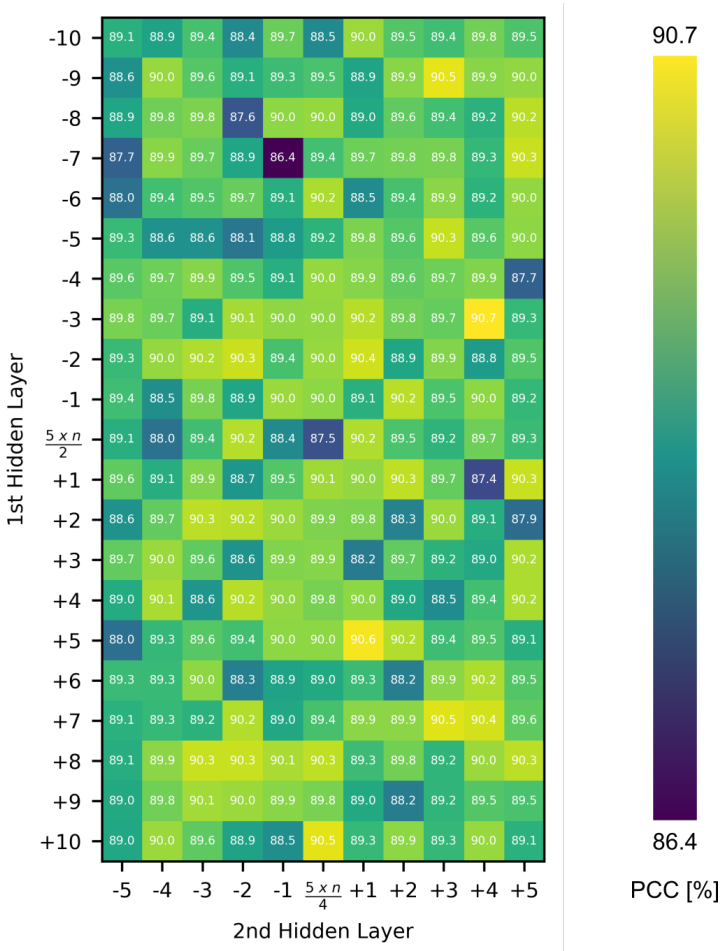

Figure 3.6: Grid search varying the number of neurons of the hidden layers. Each cell of this heat map lists the Pearson correlation coefficient (PCC, in percent) resulting from a comparison of the 16,196 pairs of RS_{NN}- and $RS_{3\mathrm{DM}}$-values for the test sequences from HisB_GA; see Materials and Methods for a detailed description of the data. For the computation of the RS_{NN}-values, the number of neurons of the two hidden layers was altered systematically based on the number $n = 14$ of residue positions constituting the design shell. The y-axis denotes the number of neurons used for the first hidden layer and the x-axis the number of neurons used for the second hidden layer. To determine the background colors of the cells, a color scheme violet (lowest value) - yellow (largest value) was used to map the PCC values. The performance did not critically depend on the number of NNs and architectures of the central grid area represented a broad optimum.

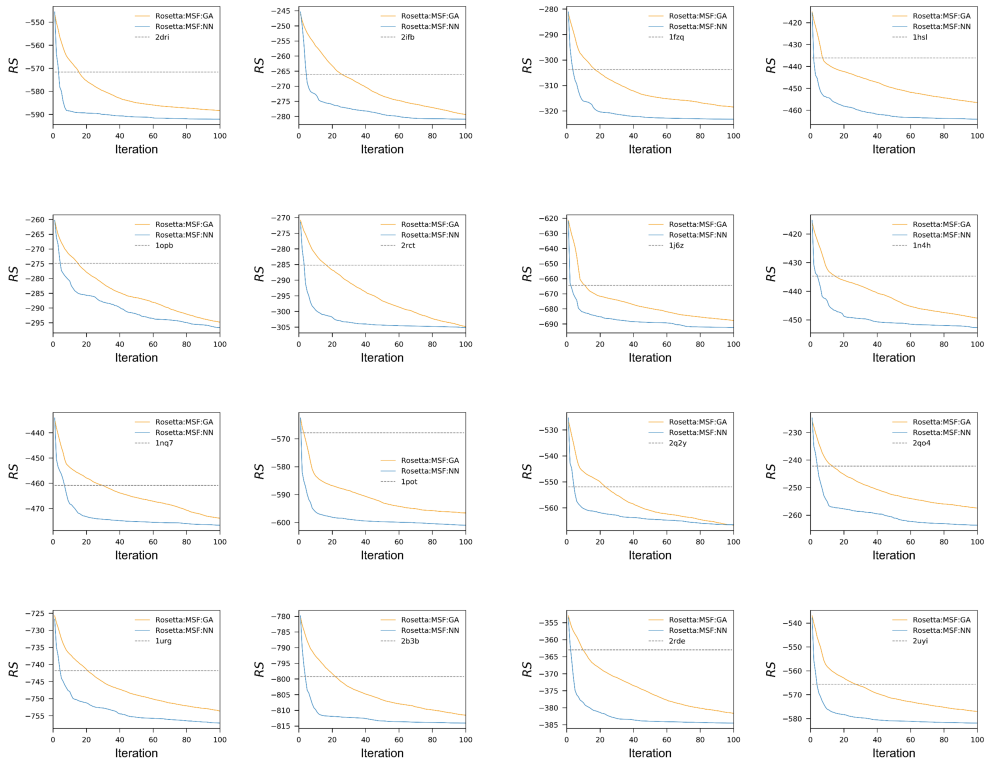

Figure 3.7: Convergence of `Rosetta:MSF:GA:enzdes` and of `Rosetta:MSF:NN:enzdes`. For each of the 16 designs, the mean Rosetta scores $RS(\mathrm{OPT}_r)$ were determined in REUs for each iteration $r = 1$ - 100 and plotted. $RS(\mathrm{OPT}_r)$ is the mean of the Rosetta scores RS^j_{3DM} of all sequences related to the iteration-specific sequence set OPT_r; compare Eq 3.2. The blue lines represent the $RS_{\mathrm{NN}}(\mathrm{OPT}_r)$ and the orange lines the $RS_{\mathrm{GA}}(\mathrm{OPT}_r)$ values. The dashed, horizontal line marks the total relaxed Rosetta score of the native sequence; the PDB-ID of the corresponding protein is indicated.

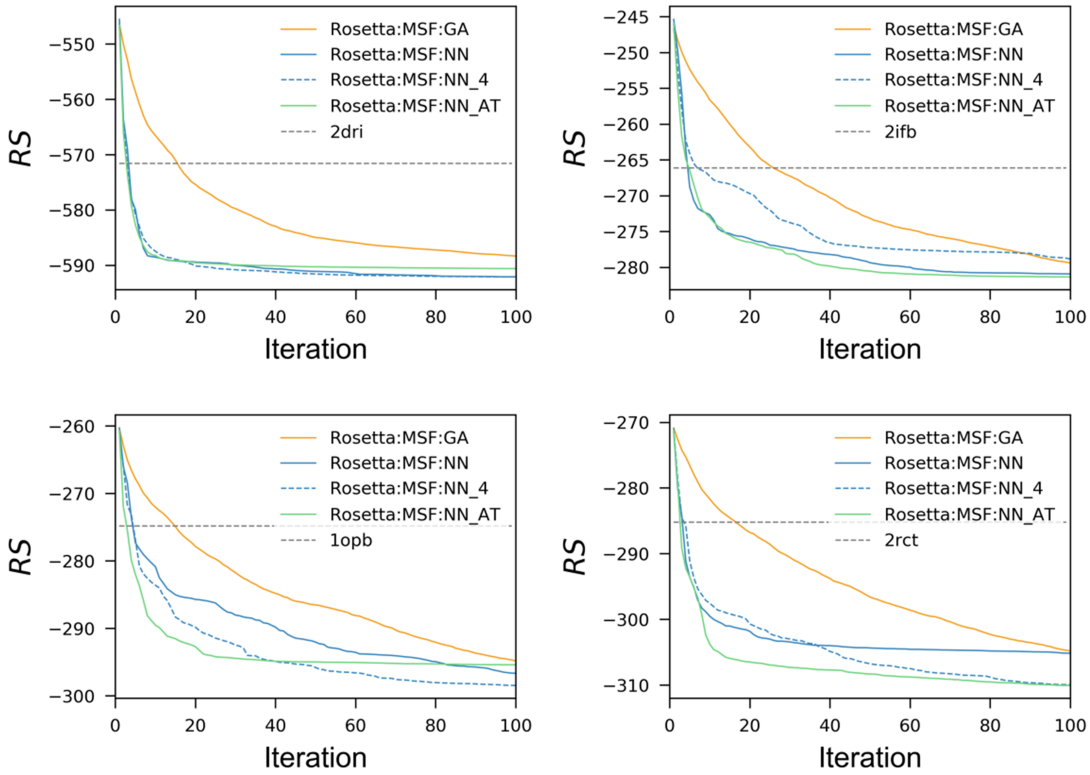

Figure 3.8: Design performance of three different feature tables. For each of the four proteins with PDB-IDs 2dri, 2ifb, 1opb, and 2rct, `Rosetta:MSF:NN` was rerun using alternative feature tables. During the `MSF:NN_4` runs, the first four features of Table 3.5, but not "mean solvent accessibility" was used. For the `Rosetta:MSF:NN_AT` runs, an alternative feature table (AT) representing each residue with the six properties "polarity", "accessible surface area", "hydrophilicity", "polarizability", "hydrophobicity", and "solvation free energy" (Yousef and Charkari, 2015) was utilized. The plots of `Rosetta:MSF:GA` and `Rosetta:MSF:NN` served as control and the dotted horizontal line represents the score of the relaxed native protein. All Rosetta scores (RS) are given in Rosetta Energy Units (REU).

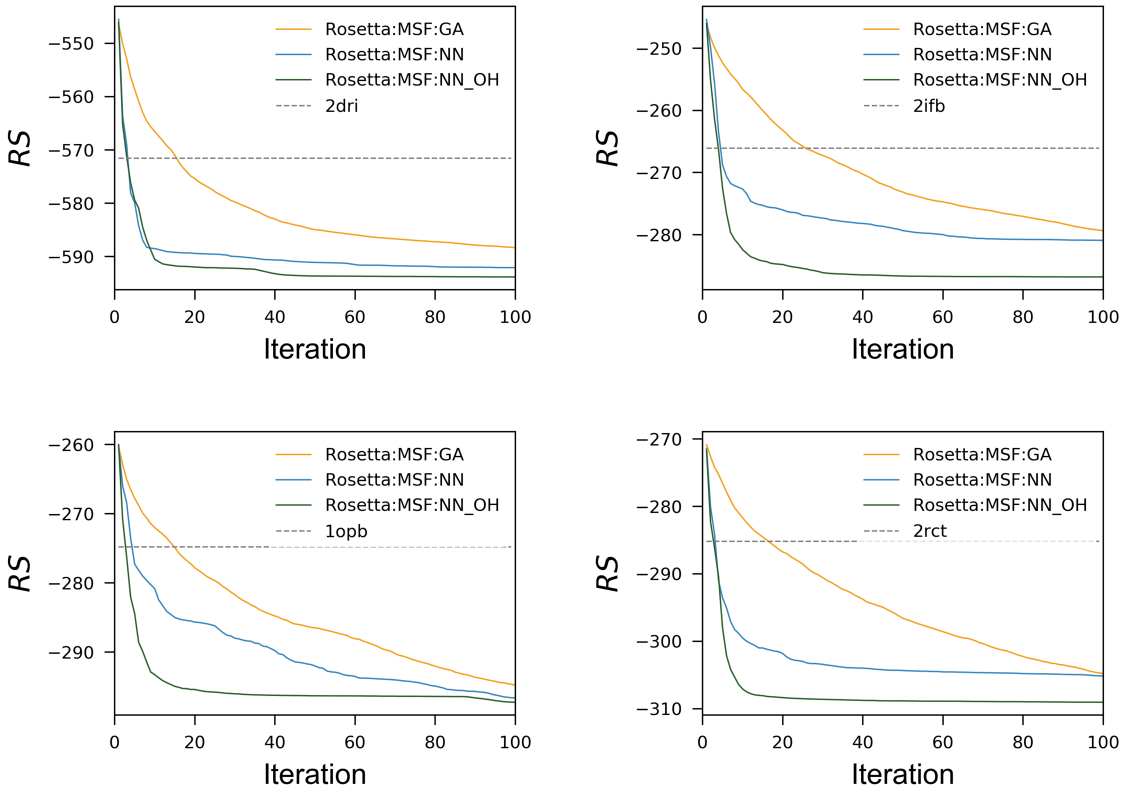

Figure 3.9: Design performance of one-hot encoding. For each of the four proteins with PDB-IDs 2dri, 2ifb, 1opb, and 2rct, `Rosetta:MSF:NN_OH` (one-hot encoding of residues) was used for redesigns. The plots of `Rosetta:MSF:GA` and `Rosetta:MSF:NN` served as control and the dotted horizontal line represents the score of the relaxed native protein. All Rosetta scores (RS) are given in Rosetta Energy Units (REU).

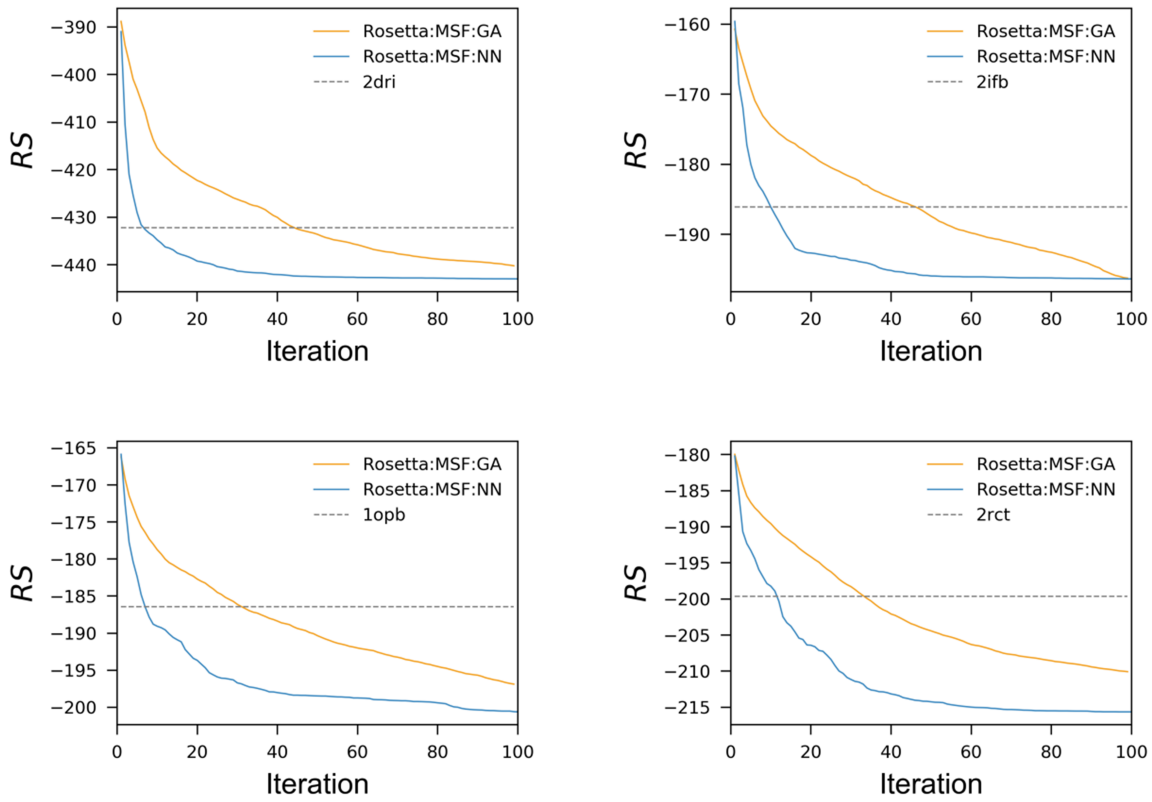

Figure 3.10: Design performance with the talaris scoring function. For each of the four proteins with PDB-IDs 2dri, 2ifb, 1opb, and 2rct, `Rosetta:MSF:NN` was rerun using the scoring function *talaris*. The plots of Rosetta:MSF:GA served as control and the dotted horizontal line represents the score of the relaxed native protein. All Rosetta scores (RS) are given in Rosetta Energy Units (REU).

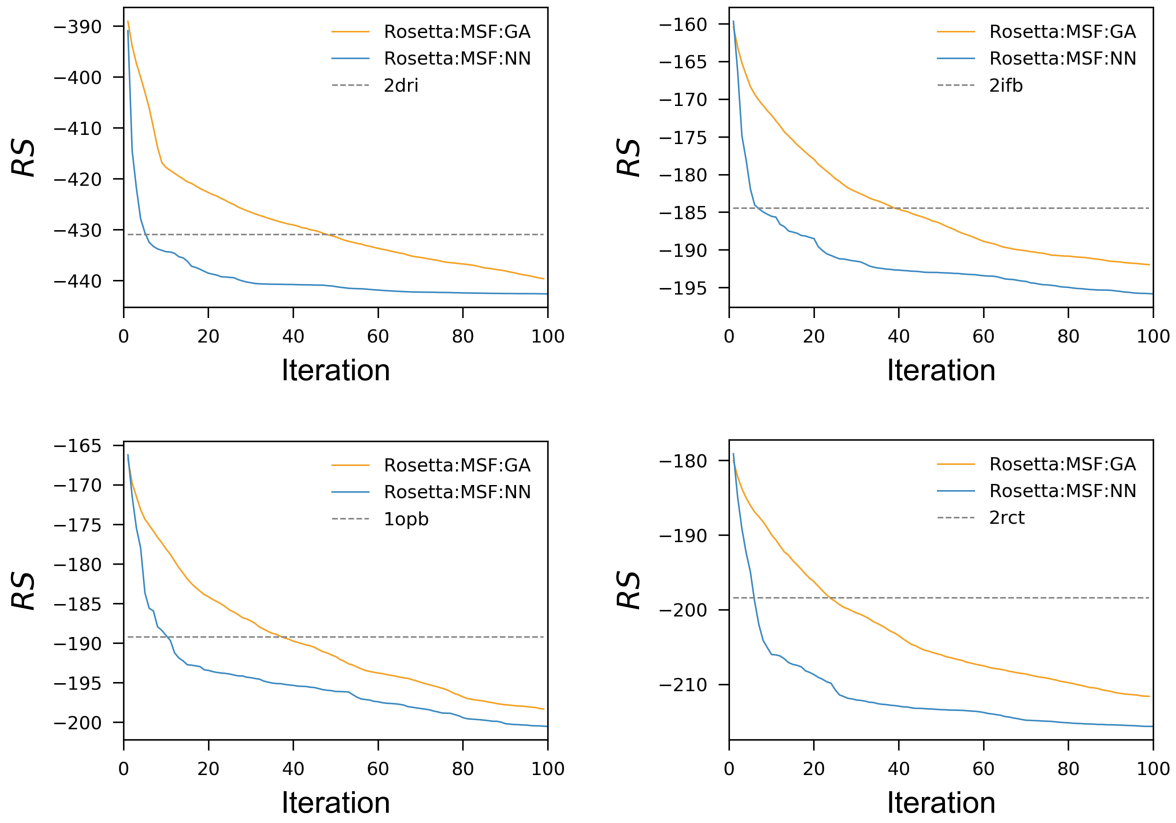

Figure 3.11: Design performance with the ref2015 scoring function. For each of the four proteins with PDB-IDs 2dri, 2ifb, 1opb, and 2rct, `Rosetta:MSF:NN` was rerun using the scoring function *ref2015*. The plots of Rosetta:MSF:GA served as control and the dotted horizontal line represents the score of the relaxed native protein. All Rosetta scores (RS) are given in Rosetta Energy Units (REU).

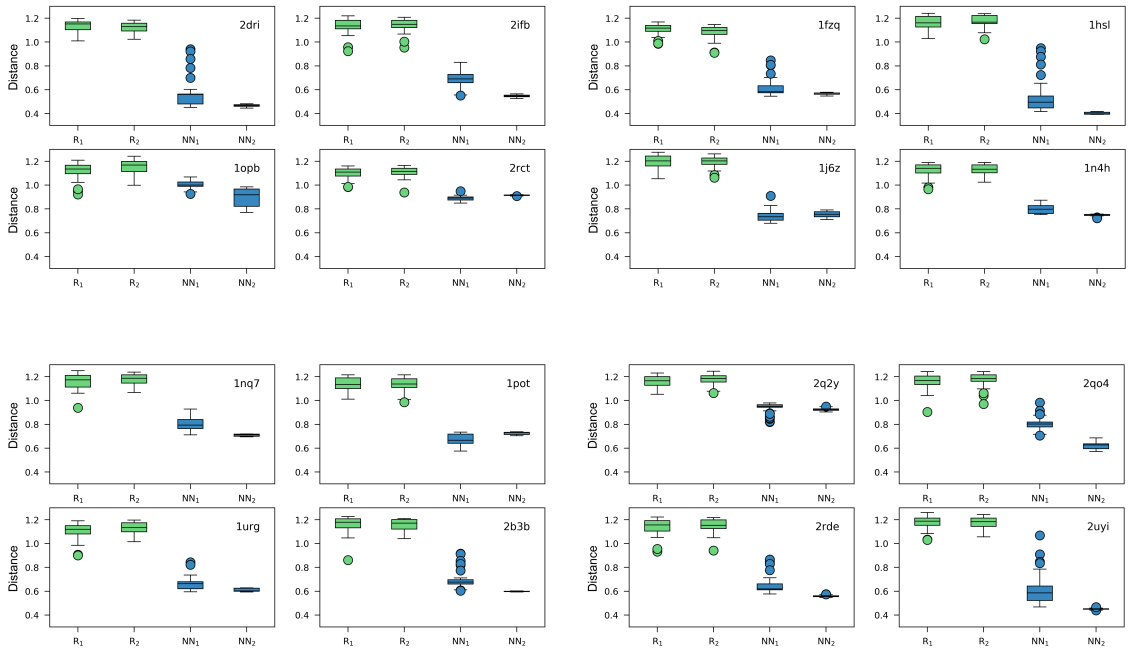

Figure 3.12: Amino acid frequency distributions for the outcome of the first and second half of design protocols. The boxplots represent the distributions of Euclidean distances between amino acid frequencies tables $ft_{r.k}$ related to the iterations of the NN based protocol and a reference table ft_k^{GA} from the GA based protocol for the design k. The group NN_1^k contains the distributions related to the first 50 and NN_2^k the distributions of the second 50 iterations. For R_1^k and R_2^k the values of each table $ft_{r,k}^{\mathrm{NN}}$ were shuffled; for details see Materials and Methods. The boxplots show the results for the 16 designs of MD_EnzBench.

Amino acid	Volume	Polarity	Isoelectric point	Hydrophobicity	Mean solvent accessibility
Lys	68.0	64.2	86.9	43.5	54.3
His	49.2	43.2	59.2	23.1	28.1
Arg	70.8	51.9	100.0	22.6	50.1
Asp	31.3	100.0	0.0	17.5	45.0
Glu	47.2	93.8	3.2	17.8	48.6
Asn	35.4	63.0	31.3	2.4	46.1
Gln	51.3	45.7	34.4	0.0	43.6
Ser	18.1	32.1	34.8	1.9	40.5
Thr	34.0	21.0	45.7	1.9	35.3
Cys	28.0	7.4	26.3	40.3	7.4
Gly	0.0	37.0	38.5	2.7	54.0
Ala	15.9	25.9	39.2	23.1	37.4
Pro	41.0	21.0	40.2	73.5	66.2
Val	47.7	8.6	38.5	49.6	19.6
Met	62.8	4.9	35.7	44.3	3.9
Ile	63.6	0.0	39.2	83.6	7.5
Leu	63.6	0.0	38.6	57.6	10.1
Tyr	78.5	9.9	34.4	70.8	30.1
Phe	77.2	1.2	38.6	76.1	5.5
Trp	100.0	4.9	37.7	100.0	13.8

Table 3.5: Feature vectors of the 20 amino acid residues. Values are taken from (Bogardt et al., 1980).

Structure of the benchmark data set

All required data for the benchmark are available at https://github.com/JulianNazet/Benchmark_Rosetta-MSF-NN. For each protein of the benchmark dataset, ten different PDB files representing the states are included as well as the parameter file of the ligand. Furthermore, it contains all flag files to run the benchmark. As an example, the files related to one design (PDB-ID 1fzq) are listed. The correspondence file (filename 1fzq.corr) maps each mutation to one residue position of the protein. Accordingly, the first resfile (filename 1fzq.resfile) defines which amino acids to use, whereas the second resfile (filename 1fzq.resfile2) describes the repack shell. The fitness function is included in the fitness file (filename fitness.daf). Each state needs to be defined and linked to its PDB file. Finally the default options used for the Rosetta:MSF (Löffler et al., 2017) run are included in the options file (filename MSF_Options).

1fzq.corr:

```
1 23 A
2 25 A
...
20 160 A
```

1fzq.resfile:

```
20
ALLAA EX 1 EX 2
```

1fzq.resfile2:

```
NATRO
START
21 A NATAA EX 1 EX 2
22 A NATAA EX 1 EX 2
...
165 A NATAA EX 1 EX 2
```

fitness.daf:

```
STATE_VECTOR state1 /States/State_1
SCALAR_EXPRESSION best_state1 = vmin( state1 )
STATE_VECTOR state2 /States/State_2
SCALAR_EXPRESSION best_state2 = vmin( state2 )
...
STATE_VECTOR state10 /States/State_10
SCALAR_EXPRESSION best_state10 = vmin( state10 )
SCALAR_EXPRESSION best_sum = best_state1 + best_state2 + best_state3
+ best_state4 + best_state5 + best_state6 + best_state7
```

+ best_state8 + best_state9 + best_state10
FITNESS best_sum

MSF_Options:

#packing options
-ex1
-ex2
-use_input_sc
-extrachi_cutoff 8
-soft_rep_design
-linmem_ig 42
#minimization options
-run::min_type dfpmin_armijo
-nblist_autoupdate
#score
-score:weights /path/to/rosetta/main/database/scoring/weights/soft_rep_design
#Enzdes
-enzdes::cst_design
-enzdes::design_min_cycles 2
-enzdes::cst_min
-enzdes::lig_packer_weight 1
-enzdes:cst_opt
-enzdes:bb_min
-enzdes:chi_min
-out::nstruct 1
#Params
-extra_res_fa /path/to/specific/params/pdb.params
#MSF
-msf::entity_resfile /path/to/pdb.resfile
-msf::fitness_file /path/to/Fitness.daf
-msf::pop_size YYY
-msf::generations XXX
-msf::fraction_by_recombination 0.05
-msf::seed_sequence_from_input_pdb /path/to/one/state/pdb/.pdb
-msf::resfile_tmpdir /path/to/specific/temp/Tempres
-msf::checkpoint_write_interval 1
-msf::checkpoint_prefix path/to/Checkpoints/Checkpoint
-msf::seed_sequence_using_correspondence_file path/to/pdb.corr
#-msf::darwin_resume true

Additional redesigns to assess the robustness of Rosetta:MSF:NN

The specific representation of amino acid properties has little effect on the performance of Rosetta:MSF:NN:enzdes

By applying Table 3.5, each amino acid is represented by means of five features. Two of them, namely "volume" and "mean solvent accessibility", represent correlated properties of the residues. We wanted to make plausible that the performance of Rosetta:MSF:NN does not critically depend on this specific residue representation and is robust to limitations of feature tables. Thus, we examined the performance of our program during two alternative recapitulation experiments of the four proteins with PDB-IDs 2dri, 2ifb, 1opb, and 2rct. During the first experiment named NN_4, only the four features "volume", "polarity", "isoelectric point", and "hydrophobicity" of Table 3.5 were used. For the second experiment named NN_AT, a different feature table representing each residue with the six properties "polarity", "accessible surface area", "hydrophilicity", "polarizability", "hydrophobicity", and "solvation free energy" (Yousef and Charkari, 2015) was utilized. Figure 3.8 shows for each of the four proteins the plot of the Rosetta scores during 100 iterations and in comparison to Rosetta:MSF:GA and Rosetta:MSF:NN, which served as references. The analysis of the plots resulting from the alternative experiments Rosetta:MSF:NN_4 and Rosetta:MSF:NN_AT indicates that the specific representation of residues has no drastic effect on the performance of the NN. In all cases, the NN-based algorithm converges faster than the GA-based one and similar score levels are reached with all feature tables. Note that the user can easily replace the feature table in accordance with his requirements.

Performance of Rosetta:MSF:NN with one-hot encoding

A common method used to encode input data for an NN is one-hot encoding. Thus, we used a 20-column vector having a "1" at the position representing the observed residue and "0" for the other 19 ones. Compared to the default feature table for which a five-column input vector is sufficient, a 20-column-vector is now required to encode residues. Consequently, the architecture of the NN was changed to $20 * n$ neurons for the input layer, $10 + 20 * n/2$ nodes for the first hidden layer and $5 + 20 * n/4$ nodes for the second hidden layer. The output layer still contained one single neuron.

One-hot encoding results in a specific weight for each residue-type irrespective of similar biophysical properties shared by several residues. This allows the NN to distinguish exactly between different residues but this lack of deducing more general properties might increase the risk of being trapped in local minima of sequence space. Moreover, the four-times larger input vector increases the risk of underfitting, because weights will not be adjusted, if residues were not presented during training. This problem does not occur with our default five-feature

representation of residues. Figure 3.9 shows that lower energies can be reached with the one-hot encoded representation of residues compared to the representation of residues by means of five biophysical properties.

`Rosetta:MSF:NN` performs well regardless of the scoring function

We wanted to make plausible that the performance boost of `Rosetta:MSF:NN` does not depend on the chosen scoring function. Therefore, we replaced the soft-rep design function which we utilized for all design and benchmark runs, with the *talaris* scoring function. Again, recapitulation experiments were performed for the four proteins with PDB-IDs 2dri, 2ifb, 1opb, and 2rct. Figure 3.10 shows for each of the four proteins the plot of the Rosetta scores during 100 iterations of `Rosetta:MSF:GA` and `Rosetta:MSF:NN`. The analysis of the plots indicates that in all cases the NN approach outperforms the GA also with the *talaris* scoring function.

Chapter 4

Comprehensive Summary, Discussion, and Outlook

4.1 Comprehensive Summary

The focus of this thesis was the application and improvement of in silico methods of protein design. As an application of an protein design protocol, a protein-protein interface was completely redesigned in Chapter 2. The aim was a reprogramming of the PabA interface in such a manner that it does no longer bind the native partner PabB, but TrpEx. TrpEx possesses a so-called interface add-on that allows TrpEx to exclusively bind TrpG. The native interaction partner of TrpEx is TrpG and the pairs TrpEx and TrpG as well as PabB and PabA are homologous enzymes. Both native complexes are glutamine amidotransferases, which are composed of a glutaminase subunits (PabA and TrpG) and a synthase subunits (PabB and TrpEx). TrpEx proteins without the interface add-on are called TrpE and PabA is able to bind TrpE, but not TrpEx. Therefore, the specification of TrpEx to TrpG is due to the interface add-on (Plach et al., 2017). To enable PabA the binding of TrpEx, I identified a patch of interface residues in TrpG, called anchor, consisting of residues interacting with the interface add-on of TrpEx. The anchor region starts with residue 7 and continues until residue 31 spanning a large consecutive part of the interface in TrpG directly in contact with the interface add-on of TrpEx (Figure 4.1). This anchor region serves as a starting point for the following design step that already introduces a large part of the native interface of TrpG to PabA. Interestingly, transferring only the interface patch from TrpG to PabA, resulted in the design variant PabA-CA that exclusively binds to TrpEx. Additional mutations introduced during the design phase strengthened the binding to TrpEx. The resulting variant PabA-CAD forms a stable complex exclusively with TrpEx and no longer interacts with its original partner PabB. For the first part of this protocol, the anchor is crafted into the new scaffold protein PabA. This process is repeated multiple times and scored to ensure optimal placement of the anchor into the protein. The two top ranking structures served as states for the MSD afterwards.

Computational protein design still requires a lot of processing time and therefore, any speed up is well appreciated. Rosetta needs a three-dimensional structure of atomic level to score a

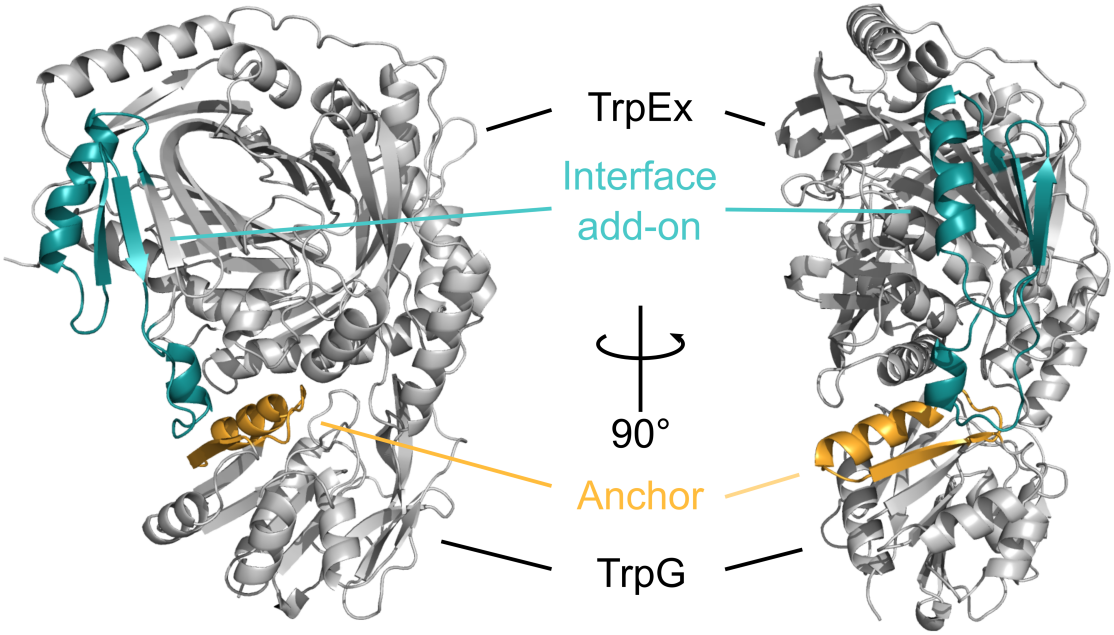

Figure 4.1: Complex of TrpEx with TrpG. The complex of TrpEx with TrpG is shown in cartoon representation (gray). The interface add-on of TrpEx from residue 71-121 is colored dark cyan (Plach et al., 2017). The corresponding region in TrpG from residue 7-31, used as an anchor, is colored orange.

proposed sequences for design. Each protein design problem is characterized by a vast energy landscape and for an approximation it is necessary to enumerate alternative design solutions. This is computationally expensive, because the number of possible sequences grows exponentially with the number of mutable residues. After deciding which residue to incorporate into the protein, the optimal rotamer must be chosen. Considering the high number of possible rotamers available in rotamer libraries, it is no surprise that this step is quite time consuming. Further computational time is needed for minimizing a structure for scoring, since all residues need to be optimized in consideration of each other. One successfully scored structure contributes one point to a vast energy landscape. This whole process is further slowed down by MSD, since a three-dimensional structure needs to be created for every state. All the optimization steps described above need to be done considering all states at once.

NNs generally work well for regression tasks (Specht et al., 1991). Therefore, I decided to use an NN to train on already scored sequences, to then rapidly predict the score of new sequences and only build the costly three-dimensional structure of high scoring predictions. To achieve this, `Rosetta:MSF:NN` was created that utilized a four layer NN to learn a problem-specific energy landscape. As shown with a comprehensive benchmark, this program reduces the computational time needed for enzyme-design problems by a factor of three. I could demonstrate that the NN can be used with a variety of Rosetta scoring functions and is capable, after a reasonable amount of training, to score unknown sequences with high accuracy (Chapter 3). This gain in speed allows for, per default, 2 million sequences to be scored per iteration in only a fraction of time

required, compared to scoring them within Rosetta. The 100 iterations of `Rosetta:MSF:NN` scored 200 million sequences for the benchmark cases, whereas the GA only scored 24.000 sequences. As a consequence, this vast search of sequence space allows the algorithm to find lower scoring solutions with fewer iterations than `Rosetta:MSF` at the benchmark cases. On average `Rosetta:MSF:NN` required for only 33.5 iterations to reach the same energy level as 100 GA iterations. A further improvement of `Rosetta:MSF:NN` is the new way it opens up for MSD. After training, NNs can be combined in any way to score new sequences for any combination of states. This feature facilitates the integration of negative design by penalizing the predictions of NNs related to negative states.

4.2 Comprehensive Discussion

Rosetta score function selection by native sequence recovery rates

A very important step for a successful computational redesign is choosing a suitable scoring function. Rosetta offers a large variety of scoring functions, each focusing on a different part of the energy terms. It is up to the user to choose the best fitting one. For my work on the interface redesign of PabA in Chapter 2, I applied multiple native sequence recovery tests to find a suitable scoring function. Initially, the Rosetta default scoring function, *talaris*, was utilized, but the resulting sequences of the redesign showed a preference for smaller residues. However, an accumulation of smaller residue is not desirable, because this could lead to cavities and a loosely packed interface. Although a protein interface can tolerate some cavities, in sum they weaken the interaction. This is why the scoring function *talaris* is not suitable for this interface design. In order to counter such behavior, Rosetta offers so called soft-repulsion scoring functions. They soften the repulsion term of the scoring function and therefore allow for a tighter packing of residues. This in turn allows for larger residues to be crafted into the protein. To test which scoring function produced the best results, I let Rosetta redesign the native PabA interface. The native sequence recovery then measures the success in recovering the native amino acids at the interface. There, the soft-rep-design scoring function performed best and I chose it for the design process.

The anchored design protocol for protein design

The `Rosetta:MSF:AnchoredDesign` allows the user to predefine a region of the native complex's interface. This region, called anchor, should transfer positive binding properties of its residues to the new protein. The anchor optimally allows the binding of the protein on its own and the later redesign step only enhanced this effect. This worked perfectly for the variants PabA-CA and PabA-CAD, since the anchor alone enables PabA-CA to exclusively bind TrpEx. The mutations introduced in PabA-CAD further improved the binding of TrpEx. However, this protocol has

some major limitations: First of all, the anchor region needs to be one single consecutive block of residues. Protein interface are commonly not arranged in succession at primary structure level. Only after folding are the interface residues in close proximity of each other. Therefore, I chose the longest consecutive part of residues in the TrpEx:TrpG interface of TrpG as anchor region, with consideration of the interface add-on. Simply replacing larger parts of a protein is the second limitation of `Rosetta:MSF:AnchoredDesign`. Introducing the anchor into the scaffold protein produces tension between the anchor and all surrounding residues. A short MD run may not be able to fit the anchor nicely into the scaffold protein. Another shortcoming is the introduction of many mutations a once. The variant PabA-CA contained 15 mutated residues of PabA, which makes it challenging to biochemically characterize each mutation. Not all of these mutations are beneficial for the desired complex of PabA and TrpEx, since they are not located at the interface. Many of them are only present, because of the anchor region needs to be a consecutive part of the protein. These mutation may clash with their surrounding residues and therefore weaken the complex. A third variant, not described in Chapter 2, was designed by allowing only the region 3 Å around the anchor to mutate. This variant should allow the anchor to better fit into the PabA scaffold and indeed showed an intermediate complex strength, by improving the binding of TrpEx compared to PabA-CA, but not reaching the level of PabA-CAD. So, while the intended use of the anchor is to serve as a starting point for the newly generated interface of scaffold (PabA) and target (TrpEx), it may also weaken this interface via introduction of clashes that need to be addressed.

Protein Design by a data-driven approach

In Chapter 2 the variant PabA** was designed by a data-driven approach. The mutations introduced in PabA** were based on PabA* (Plach et al., 2017) and a multiple sequence alignment (MSA). The MSA revealed residues that are conserved in either TrpG or PabA, but different when comparing them. Therefore, the PabA** variant introduced only seven additional mutations to PabA*, whereas PabA-CAD implemented 12 new mutations compared to PabA*. Even with fewer mutations, PabA** outperforms the computational designed variant PabA-CAD in terms of binding of TrpEx and enzyme activity. So, is the data-driven approach the better solution to protein design compared to the computational approach? While the data-driven approach succeeded for this task, it requires for biochemical in-depth knowledge of the design targets. The homology of TrpG and PabA makes is easier to spot corresponding residues and detect differences in conserved regions. Furthermore the previously characterized variant PabA* revealed even more important residues for binding of TrpEx (Plach et al., 2017). The computational design does not rely on such information and therefore performs equally well whether more information is available or not.

Experimental validation of `Rosetta:MSF:NN`

In Chapter 3 I show that `Rosetta:MSF:NN` generates better scored structures for MSD cases compared to its predecessor `Rosetta:MSF`. For all benchmark cases it outperformed the GA, but a experimental validation is missing. So, while generally better scored structures should be preferable, biochemical experiments, for example in Chapter 2, revealed that the best scoring ones are not always optimal. Therefore, a experimental validation of the prediction of `Rosetta:MSF:NN` would be valuable. In Chapter 3 I used the results of a `Rosetta:MSF` design run for the initial validation of the NN. There it was tried to transfer the activity of SerB to its homologous enzyme HisB. HisB from *Escherichia coli* catalyzes the hydrolysis of L-histidinol phosphate to L-histidinol and phosphate. HisB also has a bifunctional activity for the SerB reaction of O-phospho-L-serine to L-serine and phosphate. My goal was an experimental validation of `Rosetta:MSF:NN` by further enhancing the bifunctional activity of HisB. This design resulted in better scored structures compared to the initial design. Out of the ten top scoring sequences I selected three for biochemical validation. While all three of them produced well folded proteins, none of them showed a significant increase in SerB activity. These results do not indicate a failure of `Rosetta:MSF:NN`, because improving the catalytic activity was not part of this interface design The score functions of Rosetta only score the structures regarding their stability and all predicted sequences produced stable proteins.

Native sequences with `Rosetta:MSF:NN`

Native protein sequences are considered to be close to optimal for their structures (Kuhlman and Baker, 2000) and therefore it is desirable to design native sequences or similar ones. The evaluation of the benchmark cases in Chapter 3 indicated that the native sequences scored worse than the top scoring sequences. The native sequence similarity recovery (NSSR) describes the similarity of the designed sequence compared to the native sequence for a recapitulation task. `Rosetta:MSF` increases the NSSR compared to the SSD algorithms for the benchmark data set (Löffler et al., 2017). Estimating the NSSR values of `Rosetta:MSF:NN` considering the benchmark data set resulted in on average lower NSSR values compared to `Rosetta:MSF`. This may be due to the fact that native-like sequences score poorer than the optimal ones. On average, the NN finds sequences that score similar to native sequences after 4.2 iterations and finds better scoring sequences thereafter. However, the scores of sequences depend on the Rosetta score function and not on the NN, since the NN learns the scoring from Rosetta. Therefore, it is impossible for the NN of `Rosetta:MSF:NN` to improve the score of native-like sequences. Changing how sequences are scored to increase the fraction of native-like sequences, one would have to modify the Rosetta score functions.

4.3 Comprehensive Outlook

A new restricting interface element in TrpG

`Rosetta:MSF:AnchoredDesign` is a protocol for MSD with Rosetta and requires the user to select the so-called anchor region (Lewis and Kuhlman, 2011). The anchor region is part of the interface of a protein A that is intended to be transferred to a second protein B. The underlying assumption is that the transfer of the anchor establishes for protein B the binding of the interaction partner of A. From the work on interface add-ons (Plach et al., 2017) it was already known, that the interface of TrpEx has an insertion that determines the specific binding of TrpG. The proteins without this insertion are called TrpE and native TrpE is able to bind PabA. PabA:PabB is a homologous complex of TrpG:TrpE and only the interface add-on of TrpEx prohibits the binding of PabA to TrpEx. Thus to enable PabA to exclusively bind TrpEx, this specificity must be designed. Therefore, I chose the interface patch of TrpG, that is in close proximity to the interface-addon of TrpEx as an anchor. The design variant PabA-CA that contains as the only modification the TrpG anchor crafted into the PabA interface, exclusively forms a complex with TrpEx, but no longer with PabB. This property hints to a new restricting interface element in TrpG, since the transfer of only the anchor region alone blocks PabB from binding PabA-CA similar to how the interface add-on of TrpEx specifies its binding to TrpG. This interaction takes place although the TrpEx interface add-on is missing in PabB and suggests that the specific binding of TrpEx and TrpG requires a modification in the interfaces of both binding partners. Interestingly, the additional mutations introduced in PabA-CAD further strengthen the binding to TrpEx. The anchor region ranging from residue 7 to 31 should contain all necessary residues of this new interface element. Whether the anchor contains the whole new interface element or just the major part of it remains uncertain.

This idea of a new interface element in TrpG could be further analyzed with the help of another complex: HisH and HisF. These enzymes are part of the L-histidine biosynthesis pathway and catalyze the reaction of PRFAR and glutamine to IGP, AICAR, and glutamate (Rieder et al., 1994). HisH is homolog to TrpG and PabA and performs the same reaction of glutamine to glutamate that provides ammonia to HisF. Grafting the anchor region from TrpG to PabA for the PabA-CA variant was sufficient to allow the binding of TrpEx. Therefore, it is tempting to assume that the analogous grafting of the anchor into the HisH interface enables HisH to bind TrpEx. This finding would be a strong conformation for the existence of a new interface element in TrpG, if it also blocks the binding to HisF (see Figure 4.2).

`Rosetta:MSF:NN` a hybrid method for protein design

The sequence space of a computational design is huge as the number of possible residue combinations grows by the factor of 20 for each position allowed to mutate. Even for smaller design

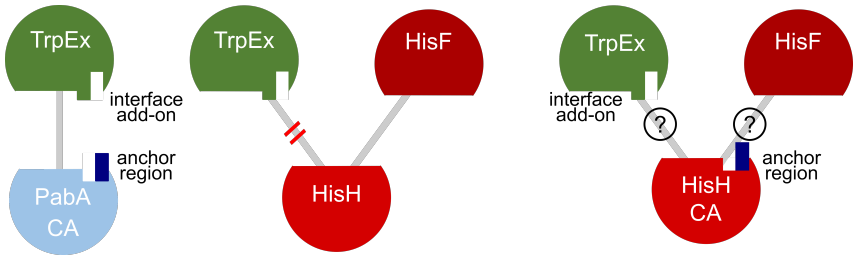

Figure 4.2: Subunit interaction specificity of the imidazole glycerol phosphate synthase. The designed variant PabA-CA already interacts with TrpEx. The native HisH subunit of the imidazol glycerol phosphate synthase complex does not form a complex with TrpEx. If the grafted anchor of TrpG contains indeed a new interface element, a newly designed variant of HisH with the region implemented, called HisH-CA should not only allow for binding to TrpEx, but also block binding to HisF.

tasks of e.g. only five positions to mutate, the number of possible sequences is $20^5 = 3,200,000$. To adequately score a sequence, Rosetta requires a three-dimensional model. This generation and optimization of a three-dimensional structure is very time consuming, because each newly mutated residue needs to be optimized regarding its rotamer. Therefore, it is necessary to scan the rotamer libraries, not only for the first shell mutations but also for each residue in the second shell that is required to move for optimal side chain packing. It follows that an alternative approach which does not require a three-dimensional structure to score a sequence, but reaches the same level of accuracy would greatly speed-up the design process. NNs proved to be a good candidate for this task, since they produce the prediction quite fast and are able to learn a large variety of problems. The hybrid protocol `Rosetta:MSF:NN` combines the best of both sides. It utilizes the scoring of Rosetta to train NNs and allows for the prediction of scores of a vast amount of sequences in a fraction of time compared to the standard Rosetta protocol. To train the NN, sequences and the corresponding Rosetta scores are required. Therefore, this approach is problem-specific and needs to be trained for each design; however, a small number of training data is sufficient. As I could show, after training the NNs are able to determine scores that are similar to those based on the Rosetta score function that assesses the three-dimensional structure. In addition, `Rosetta:MSF:NN` opens new ways for multi-state protein design. Since a specific NN is trained for each state of the design, the output of the NNs, i.e. the predicted scores can later be recombined in any manner. This feature even allows for negative protein design, without the use of the design algorithms not created for negative design, because for training, all designs are positive designs. Furthermore, this feature allows the scoring of sequences for any combination of states after training is complete.

Another great advantage of Rosetta:MSF:NN is the possibility to analyze the NNs. For example, one could alter the input at every single input neuron and record the reaction of a NN to it (Figure 4.3). To demonstrate such an analysis, I used a design run of HisB as described in Chapter 3. Each input feature at every position was varied from 0% to 100% and the resulting error of the prediction recorded. Positions, where changes result in large errors, could be more important for the protein than those with low error. A residue with close to zero changes in error

is position 58. Since all features produce similar scores, regardless of their characteristics leads to the assumption that residue position 58 is of lower interest for the design. In contrast to this is position 65. There, changing the "hydrophobicity" results in a increase of the prediction error of up to 7 REU (Figure 4.3). Therefore, one could assume that position 65 requires a hydrophilic AA, since increasing the "hydrophobicity" feature also increases the prediction error.

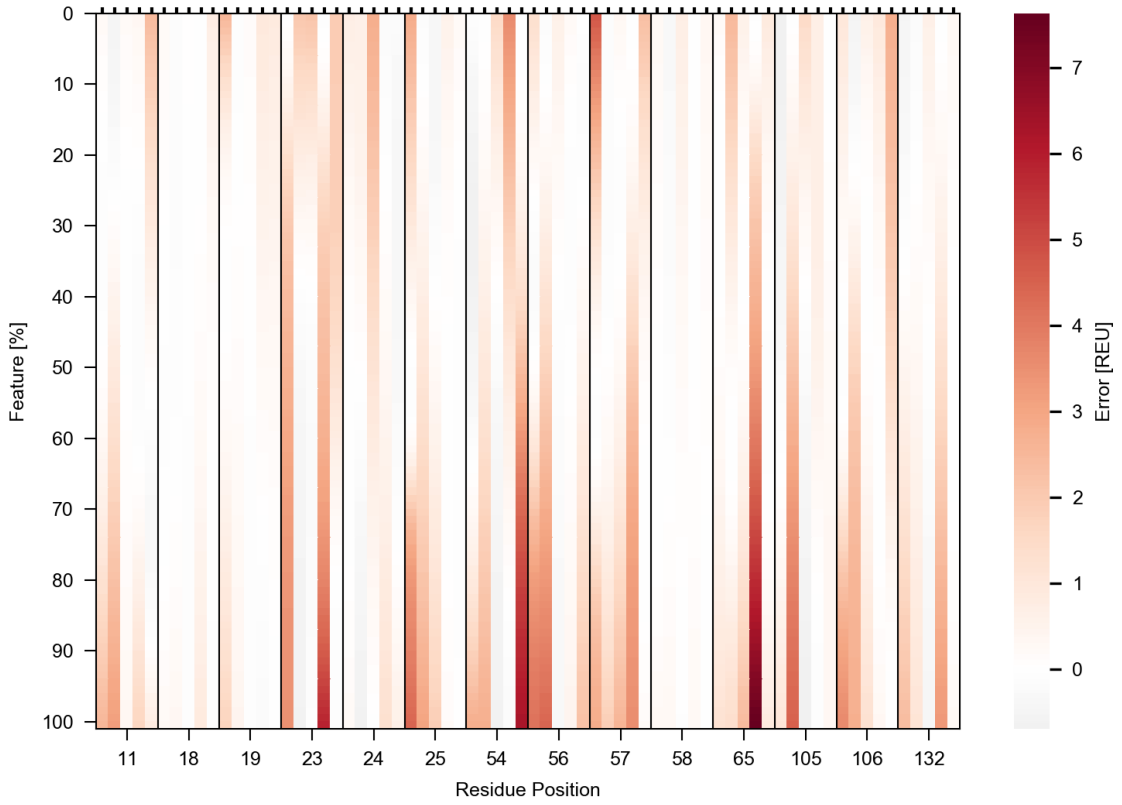

Figure 4.3: Heatmap of feature importance. At every residue position each feature was varied from 0% to 100% and the resulting score was then compared to the original prediction. The error was measured in REUs and color coded from gray (better or same performance) to red (worse performance). Each residue is described by five features. The series of features is "volume", "polarity", "isoelectric point", "hydrophobicity", and "mean solvent accessibility".

To allow the comparison with its predecessor, `Rosetta:MSF:NN` is implemented to run several iterations during the training phase. This could be changed to a longer, initial training phase that could be ended after reaching a low overall prediction error. Since training an NN for each state takes longer than the subsequent scoring phase, this part should be reduced to a minimum. Furthermore, the representation of the input to `Rosetta:MSF:NN` could be further optimized. I already shortly tested the one hot encoding (see Section 3.6) of the input instead of the feature representation. There, each residue is decoded as a 20-column vector having a 1 at the position representing the observed residue and 0 for the other 19 ones. This encoding

removes any inter-connecting features of the residues and introduces a specific weight for each residue. A short test (see Figure 3.9) revealed an improvement of `Rosetta:MSF:NN` compared to the originally used feature table that represents residues by means of biophysical properties. It is known that the specific representation of the input has great impact on speed and accuracy of NNs (Raimondi et al., 2019).

In summary, the current implementation of `Rosetta:MSF:NN` already reduces the computational load by approximately a factor of three, if compared to `Rosetta:MSF`. Moreover, a further gain in speed is feasible by modifying the course of the training phase. Assessing in more detail the effects of encoding amino acid residues for the input layer and analyzing trained networks are further promising avenues than can be followed to improve the performance of this fascinating software.

Digital Supplemental Data

The following Digital Supplemental Data files can be found online at:
https://github.com/JulianNazet/Dissertation_Nazet

Chapter 2:	Data_1_PDB_Files
	Data_2_Pisa_Files
	Data_3_Design_Files
Chapter 3:	Benchmark-Data

Abbreviations

AA	Amino Acid
ADCS	AminoDeoxyChorismate Synthase
AS	Anthranilate Synthase
AT	Alternative feature Table
GA	Genetic Algorithm
GAT	Glutamine AmidoTransferase
IMAC	Immobilized Metal ion Affinity Chromatography
MD	Molecular Dynamics
MSA	Multiple Sequence Alignment
MSD	Mulit-State Design
NAA	Normalized Area Above
NN	Neural Network
NSSR	Native Sequence Similarity Recovery
PCC	Pearson Correlation Coefficient
PDB	Protein Data Bank
PM	Primary Metabolism
REU	Rosetta Energy Unit
RS	Rosetta Score
RMSD	Root Mean Square Deviation
SM	Secondary Metabolism
SEC	Size-Exclusion Chromatography
SSD	Single-State Design

Bibliography

L. Addadi, E. K. Jaffe and J. R. Knowles. Secondary tritium isotope effects as probes of the enzymic and nonenzymic conversion of chorismate to prephenate. *Biochemistry*, 22(19):4494–4501,

J. Adolf-Bryfogle, O. Kalyuzhniy, M. Kubitz, B. D. Weitzner, X. Hu, Y. Adachi, W. R. Schief and R. L. Dunbrack Jr. Rosettaantibodydesign (rabd): A general framework for computational antibody design. *PLoS computational biology*, 14(4):e1006112,

S. E. Ahnert, J. A. Marsh, H. Hernandez, C. V. Robinson and S. A. Teichmann. Principles of assembly reveal a periodic table of protein complexes. *Science*, 350(6266):aaa2245, doi: 10.1126/science.aaa2245.

R. F. Alford, A. Leaver-Fay, J. R. Jeliazkov, M. J. O'Meara, F. P. DiMaio, H. Park, M. V. Shapovalov, P. D. Renfrew, V. K. Mulligan, K. Kappel, J. W. Labonte, M. S. Pacella, R. Bonneau, P. Bradley, R. L. Dunbrack, R. Das, D. Baker, B. Kuhlman, T. Kortemme and J. J. Gray. The rosetta all-atom energy function for macromolecular modeling and design. *Journal of chemical theory and computation*, 13:3031–3048, doi: 10.1021/acs.jctc.7b00125.

B. D. Allen and S. L. Mayo. An efficient algorithm for multistate protein design based on FASTER. *Journal of Computational Chemistry*, 31(5):904–16, doi: 10.1002/jcc.21375.

H. C. Andersen. Molecular dynamics simulations at constant pressure and/or temperature. *The Journal of chemical physics*, 72(4):2384–2393,

F. A. Azevedo, L. R. Carvalho, L. T. Grinberg, J. M. Farfel, R. E. Ferretti, R. E. Leite, W. J. Filho, R. Lent and S. Herculano-Houzel. Equal numbers of neuronal and nonneuronal cells make the human brain an isometrically scaled-up primate brain. *Journal of Comparative Neurology*, 513(5):532–541,

M. L. Azoitei, B. E. Correia, Y.-E. A. Ban, C. Carrico, O. Kalyuzhniy, L. Chen, A. Schroeter, P.-S. Huang, J. S. McLellan, P. D. Kwong et al. Computation-guided backbone grafting of a discontinuous motif onto a protein scaffold. *Science*, 334(6054):373–376,

A. E. Badaczewska-Dawid, A. Kolinski and S. Kmiecik. Computational reconstruction of atomistic protein structures from coarse-grained models. *Computational and Structural Biotechnology Journal*, 18:162–176,

H. J. Berendsen, D. van der Spoel and R. van Drunen. Gromacs: a message-passing parallel molecular dynamics implementation. *Computer physics communications*, 91(1-3):43–56,

D. Bhattacharya. refined: Improved protein structure refinement using machine learning based restrained relaxation. *Bioinformatics*, 35(18):3320–3328,

R. A. Bogardt, B. N. Jones, F. E. Dwulet, W. H. Garner, L. D. Lehman and F. R. Gurd. Evolution of the amino acid substitution in the mammalian myoglobin gene. *Journal of molecular evolution*, 15(3):197–218,

J. A. Brannigan, G. Dodson, H. J. Duggleby, P. C. Moody, J. L. Smith, D. R. Tomchick and A. G. Murzin. A protein catalytic framework with an n-terminal nucleophile is capable of self-activation. *Nature*, 378(6555):416–9,

T. Cavalier-Smith. Origins of secondary metabolism. *Secondary metabolites: their function and evolution*, S. 64–87,

J. Darnell, H. Lodish, A. Berk, L. Zipursky, P. Matsudaira and D. Baltimore. Overview of neuron structure and function. In *Molecular cell biology*. WH Freeman and Company, 2000.

J. A. Davey and R. A. Chica. Multistate approaches in computational protein design. *Protein Science*, 21(9):1241–52, doi: 10.1002/pro.2128.

J. Davis and M. Goadrich. The relationship between precision-recall and roc curves. In *Proceedings of the 23rd international conference on Machine learning*, S. 233–240, 2006.

F. DiMaio, A. Leaver-Fay, P. Bradley, D. Baker and I. André. Modeling symmetric macromolecular structures in rosetta3. *PLoS One*, 6(6),

S. J. Fleishman, T. A. Whitehead, D. C. Ekiert, C. Dreyfus, J. E. Corn, E. M. Strauch, I. A. Wilson and D. Baker. Computational design of proteins targeting the conserved stem region of influenza hemagglutinin. *Science*, 332(6031):816–21, doi: 10.1126/science.1202617.

M. Fromer, C. Yanover, A. Harel, O. Shachar, Y. Weiss and M. Linial. Sprint: side-chain prediction inference toolbox for multistate protein design. *Bioinformatics*, 26(19):2466–2467,

P. Gainza, K. E. Roberts, I. Georgiev, R. H. Lilien, D. A. Keedy, C.-Y. Chen, F. Reza, A. C. Anderson, D. C. Richardson, J. S. Richardson et al. Osprey: protein design with ensembles, flexibility, and provable algorithms. In *Methods in enzymology*, volume 523, S. 87–107. Elsevier, 2013.

P. Gainza, H. M. Nisonoff and B. R. Donald. Algorithms for protein design. *Current opinion in structural biology*, 39:16–26,

M. Gao and J. Skolnick. Structural space of protein-protein interfaces is degenerate, close to complete, and highly connected. *Proceedings of the National Academy of Sciences of the United States of America*, 107(52):22517–22, doi: 10.1073/pnas.1012820107.

L. Garma, S. Mukherjee, P. Mitra and Y. Zhang. How many protein-protein interactions types exist in nature? *PLoS One*, 7(6):e38913, doi: 10.1371/journal.pone.0038913.

A. Goldenzweig, M. Goldsmith, S. E. Hill, O. Gertman, P. Laurino, Y. Ashani, O. Dym, T. Unger, S. Albeck, J. Prilusky, R. L. Lieberman, A. Aharoni, I. Silman, J. L. Sussman, D. S. Tawfik and S. J. Fleishman. Automated structure- and sequence-based design of proteins for high bacterial expression and stability. *Molecular Cell*, 63(2):337–46, doi: 10.1016/j.molcel.2016.06.012.

D. S. Goodsell and A. J. Olson. Structural symmetry and protein function. *Annual Review of Biophysics and Biomolecular Structure*, 29:105–53, doi: 29/1/105[pii]10.1146/annurev.biophys.29.1.105.

C. Grisostomi, P. Kast, R. Pulido, J. Huynh and D. Hilvert. Efficient *in vivo* synthesis and rapid purification of chorismic acid using an engineered *Escherichia coli* strain. *Bioorganic Chemistry*, 25(5-6):297–305,

G. Guntas, C. Purbeck and B. Kuhlman. Engineering a protein–protein interface using a computationally designed library. *Proceedings of the National Academy of Sciences*, 107(45):19296–19301,

G. Guntas, S. M. Lewis, K. M. Mulvaney, E. W. Cloer, A. Tripathy, T. R. Lane, M. B. Major and B. Kuhlman. Engineering a genetically encoded competitive inhibitor of the keap1–nrf2 interaction via structure-based design and phage display. *Protein Engineering, Design & Selection*, 29(1):1–9,

P. B. Harbury, J. J. Plecs, B. Tidor, T. Alber and P. S. Kim. High-resolution protein design with backbone freedom. *Science*, 282(5393):1462–7,

S. C. Howell, K. K. Inampudi, D. P. Bean and C. J. Wilson. Understanding thermal adaptation of enzymes through the multistate rational design and stability prediction of 100 adenylate kinases. *Structure*, 22(2):218–229,

Z. Hu, B. Ma, H. Wolfson and R. Nussinov. Conservation of polar residues as hot spots at protein interfaces. *Proteins*, 39(4):331–42,

E. L. Humphris and T. Kortemme. Design of multi-specificity in protein interfaces. *PLoS Computational Biology*, 3(8):e164, doi: 10.1371/journal.pcbi.0030164.

E. Humphris-Narayanan, E. Akiva, R. Varela, S. Ã“ ConchÃºir and T. Kortemme. Prediction of mutational tolerance in hiv-1 protease and reverse transcriptase using flexible backbone protein design. *PLoS computational biology*, 8:e1002639, doi: 10.1371/journal.pcbi.1002639.

M. Hussain, S. P. Angus and B. Kuhlman. Engineering a protein binder specific for p38α with interface expansion. *Biochemistry*, 57(30):4526–4535,

J. A. Jones and P. J. Hore. The maximum entropy method. Appearance and reality. *Journal of Magnetic Resonance*, 92:363–376,

P. T. Jones, P. H. Dear, J. Foote, M. S. Neuberger and G. Winter. Replacing the complementarity-determining regions in a human antibody with those from a mouse. *Nature*, 321(6069):522,

J. Kaplan and W. F. DeGrado. De novo design of catalytic proteins. *Proceedings of the National Academy of Sciences of the United States of America*, 101(32):11566–70,

M. Karimi and Y. Shen. icfn: an efficient exact algorithm for multistate protein design. *Bioinformatics*, 34(17):i811–i820,

K. W. Kaufmann, G. H. Lemmon, S. L. Deluca, J. H. Sheehan and J. Meiler. Practically useful: what the rosetta protein modeling suite can do for you. *Biochemistry*, 49(14):2987–98, doi: 10.1021/bi902153g.

L. A. K. Ke Chen. *Neural Networks in Bioinformatics*. Springer, 2012. ISBN 978-3-540-92909-3.

A. S. Konagurthu, J. C. Whisstock, P. J. Stuckey and A. M. Lesk. Mustang: a multiple structural alignment algorithm. *Proteins: Structure, Function, and Bioinformatics*, 64(3):559–574,

A. Kossel. Über die chemische Zusammensetzung der Zelle. *Du Bois-Reymond's Archiv/Arch Anat Physiol Physiol Abt*, 8:181–189,

E. Krieger and G. Vriend. Yasara view - molecular graphics for all devices - from smartphones to workstations. *Bioinformatics*, 30(20):2981–2982,

E. Krieger, T. Darden, S. B. Nabuurs, A. Finkelstein and G. Vriend. Making optimal use of empirical energy functions: force-field parameterization in crystal space. *Proteins*, 57(4): 678–83, doi: 10.1002/prot.20251.

E. Krissinel and K. Henrick. Inference of macromolecular assemblies from crystalline state. *Journal of Molecular Biology*, 372(3):774–97, doi: 10.1016/j.jmb.2007.05.022.

B. Kröse and P. van der Smagt. An introduction to neural networks.

B. Kuhlman and D. Baker. Native protein sequences are close to optimal for their structures. *Proceedings of the National Academy of Sciences*, 97(19):10383–8,

B. Kuhlman and P. Bradley. Advances in protein structure prediction and design. *Nature Reviews Molecular Cell Biology*, 20(11):681–697,

B. Kuhlman, G. Dantas, G. C. Ireton, G. Varani, B. L. Stoddard and D. Baker. Design of a novel globular protein fold with atomic-level accuracy. *Science*, 302(5649):1364–8,

A. Leaver-Fay, R. Jacak, P. B. Stranges and B. Kuhlman. A generic program for multistate protein design. *PLoS One*, 6(7):e20937, doi: 10.1371/journal.pone.0020937.

A. Leaver-Fay, M. Tyka, S. M. Lewis, O. F. Lange, J. Thompson, R. Jacak, K. Kaufman, P. D. Renfrew, C. A. Smith, W. Sheffler, I. W. Davis, S. Cooper, A. Treuille, D. J. Mandell, F. Richter, Y. E. Ban, S. J. Fleishman, J. E. Corn, D. E. Kim, S. Lyskov, M. Berrondo, S. Mentzer, Z. Popovic, J. J. Havranek, J. Karanicolas, R. Das, J. Meiler, T. Kortemme, J. J. Gray, B. Kuhlman, D. Baker and P. Bradley. ROSETTA3: an object-oriented software suite for the simulation and design of macromolecules. *Methods in Enzymology*, 487:545–74, doi: 10.1016/B978-0-12-381270-4.00019-6.

J. Leem, G. Georges, J. Shi and C. M. Deane. Antibody side chain conformations are position-dependent. *Proteins: Structure, Function, and Bioinformatics*, 86(4):383–392,

E. D. Levy, E. Boeri Erba, C. V. Robinson and S. A. Teichmann. Assembly reflects evolution of protein complexes. *Nature*, 453(7199):1262–5, doi: 10.1038/nature06942.

S. M. Lewis and B. A. Kuhlman. Anchored design of protein-protein interfaces. *PLoS One*, 6 (6):e20872, doi: 10.1371/journal.pone.0020872.

S. M. Lippow, K. D. Wittrup and B. Tidor. Computational design of antibody-affinity improvement beyond in vivo maturation. *Nature biotechnology*, 25(10):1171–1176,

S. Liu, X. Xiang, X. Gao and H. Liu. neighborhood preference of amino acids in protein structures and its applications in protein structure assessment. *Scientific reports*, 10(1):1–11,

H. Lodish, A. Berk and S. Zipursky. Overview of neuron structure and function. *Molecular Cell Biology*,

P. Löffler, S. Schmitz, E. Hupfeld, R. Sterner and R. Merkl. Rosetta:MSF: a modular framework for multi-state computational protein design. *PLoS Computational Biology*, 13(6):e1005600, doi: 10.1371/journal.pcbi.1005600.

L. L. Looger, M. A. Dwyer, J. J. Smith and H. W. Hellinga. Computational design of receptor and sensor proteins with novel functions. *Nature*, 423(6936):185–90,

M. Lunzer, G. B. Golding and A. M. Dean. Pervasive cryptic epistasis in molecular evolution. *PLoS Genetics*, 6(10):e1001162, doi: 10.1371/journal.pgen.1001162.

D. J. C. Mackay. Introduction to monte carlo methods. In *Learning in graphical models*, S. 175–204. Springer, 1998.

J. A. Marsh and S. A. Teichmann. Structure, dynamics, assembly, and evolution of protein complexes. *Annual review of biochemistry*, 84:551–575, doi: 10.1146/annurev-biochem-060614-034142.

L. C. Martin, G. B. Gloor, S. D. Dunn and L. M. Wahl. Using information theory to search for co-evolving residues in proteins. *Bioinformatics*, 21(22):4116–24,

W. S. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133,

M. Mitchell. *An introduction to genetic algorithms*. MIT press, 1998.

A. A. Morollo and M. J. Eck. Structure of the cooperative allosteric anthranilate synthase from *Salmonella typhimurium*. *Nature Structural Biology*, 8(3):243–7,

S. Mouilleron and B. Golinelli-Pimpaneau. Conformational changes in ammonia-channeling glutamine amidotransferases. *Current Opinion in Structural Biology*, 17(6):653–64, doi: 10.1016/j.sbi.2007.09.003.

C. Negron and A. E. Keating. Multistate protein design using CLEVER and CLASSY. *Methods in Enzymology*, 523:171–90, doi: 10.1016/B978-0-12-394292-0.00008-4.

R. Netzer, D. Listov, R. Lipsh, O. Dym, S. Albeck, O. Knop, C. Kleanthous and S. J. Fleishman. Ultrahigh specificity in a network of computationally designed protein-interaction pairs. *Nature communications*, 9(1):1–13,

L. G. Nivón, S. Bjelic, C. King and D. Baker. Automating human intuition for protein design. *Proteins*, 82(5):858–66, doi: 10.1002/prot.24463.

D. L. Ollis, E. Cheah, M. Cygler, B. Dijkstra, F. Frolow, S. M. Franken, M. Harel, S. J. Remington, I. Silman, J. Schrag and et al. The α/β hydrolase fold. *Protein Engineering*, 5 (3):197–211,

S. Ovchinnikov, H. Kamisetty and D. Baker. Robust and accurate prediction of residue–residue interactions across protein interfaces using evolutionary information. *Elife*, 3:e02030,

J. F. Parsons, P. Y. Jensen, A. S. Pachikara, A. J. Howard, E. Eisenstein and J. E. Ladner. Structure of *Escherichia coli* aminodeoxychorismate synthase: architectural conservation and diversity in chorismate-utilizing enzymes. *Biochemistry*, 41(7):2198–208,

M. G. Plach, F. Semmelmann, F. Busch, M. Busch, L. Heizinger, V. H. Wysocki, R. Merkl and R. Sterner. Evolutionary diversification of protein–protein interactions by interface add-ons. *Proceedings of the National Academy of Sciences*, 114(40):E8333–E8342,

N. Pokala and T. M. Handel. Energy functions for protein design: adjustment with protein-protein complex affinities, models for the unfolded state, and negative design of solubility and specificity. *Journal of Molecular Biology*, 347(1):203–27,

E. Procko, G. Y. Berguig, B. W. Shen, Y. Song, S. Frayo, A. J. Convertine, D. Margineantu, G. Booth, B. E. Correia, Y. Cheng, W. R. Schief, D. M. Hockenbery, O. W. Press, B. L. Stoddard, P. S. Stayton and D. Baker. A computationally designed inhibitor of an epstein-barr viral bcl-2 protein induces apoptosis in infected cells. *Cell*, 157(7):1644–56, doi: 10.1016/j.cell.2014.04.034.

D. Raimondi, G. Orlando, W. F. Vranken and Y. Moreau. Exploring the limitations of biophysical propensity scales coupled with machine learning for protein sequence analysis. *Scientific reports*, 9(1):1–11,

B. Ramsundar and R. B. Zadeh. TensorFlow for deep learning: from linear regression to reinforcement learning. *O'Reilly Media, Inc.*,

F. M. Raushel, J. B. Thoden and H. M. Holden. The amidotransferase family of enzymes: molecular machines for the production and delivery of ammonia. *Biochemistry*, 38(25):7891–9,

F. Richter, A. Leaver-Fay, S. D. Khare, S. Bjelic and D. Baker. De novo enzyme design using rosetta3. *PLoS One*, 6(5):e19230, doi: 10.1371/journal.pone.0019230.

L. Riechmann, M. Clark, H. Waldmann and G. Winter. Reshaping human antibodies for therapy. *Nature*, 332(6162):323–327,

G. Rieder, M. J. Merrick, H. Castorph and D. Kleiner. Function of hisF and hisH gene products in histidine biosynthesis. *Journal of Biological Chemistry*, 269(20):14386–14390,

C. A. Rohl, C. E. Strauss, K. M. Misura and D. Baker. Protein structure prediction using rosetta. *Methods in Enzymology*, 383:66–93,

B. Rohweder, F. Semmelmann, C. Endres and R. Sterner. Standardized cloning vectors for protein production and generation of large gene libraries in *Escherichia coli*. *BioTechniques*, 64(1):24–26,

B. Rost. Twilight zone of protein sequence alignments. *Protein engineering*, 12(2):85–94,

B. Rost. Phd: predicting one-dimensional protein structure by profile-based neural networks. *Methods in Enzymology*, 266:525–539,

D. Röthlisberger, O. Khersonsky, A. M. Wollacott, L. Jiang, J. Dechancie, J. Betker, J. L. Gallaher, E. A. Althoff, A. Zanghellini, O. Dym, S. Albeck, K. N. Houk, D. S. Tawfik and D. Baker. Kemp elimination catalysts by computational enzyme design. *Nature*, 453(7192): 164–6,

R. I. Sadreyev, B.-H. Kim and N. V. Grishin. Discrete–continuous duality of protein structure space. *Current opinion in structural biology*, 19(3):321–328,

G. Schreiber and S. J. Fleishman. Computational design of protein–protein interactions. *Current opinion in structural biology*, 23(6):903–910,

G. Schreiber and A. E. Keating. Protein binding specificity versus promiscuity. *Current opinion in structural biology*, 21(1):50–61,

Schrödinger, LLC. The PyMOL molecular graphics system, version 1.7. November 2015.

F. Semmelmann, E. Hupfeld, L. Heizinger, R. Merkl and R. Sterner. A fold-independent interface residue is crucial for complex formation and allosteric signaling in class i glutamine amidotransferases. *Biochemistry*, 58(22):2584–2588,

F. Semmelmann, K. Straub, J. Nazet, C. Rajendran, R. Merkl and R. Sterner. Mapping the allosteric communication network of aminodeoxychorismate synthase. *Journal of molecular biology*, 431:2718–2728, doi: 10.1016/j.jmb.2019.05.021.

A. W. Senior, R. Evans, J. Jumper, J. Kirkpatrick, L. Sifre, T. Green, C. Qin, A. Žídek, A. W. Nelson, A. Bridgland et al. Improved protein structure prediction using potentials from deep learning. *Nature*, S. 1–5,

A. M. Sevy, N. C. Wu, I. M. Gilchuk, E. H. Parrish, S. Burger, D. Yousif, M. B. Nagel, K. L. Schey, I. A. Wilson, J. E. Crowe et al. Multistate design of influenza antibodies improves affinity and breadth against seasonal viruses. *Proceedings of the National Academy of Sciences*, 116(5):1597–1602,

P. S. Shah, G. K. Hom, S. A. Ross, J. K. Lassila, K. A. Crowhurst and S. L. Mayo. Full-sequence computational design and solution structure of a thermostable protein variant. *Journal of molecular biology*, 372(1):1–6,

M. V. Shapovalov and R. L. Dunbrack Jr. A smoothed backbone-dependent rotamer library for proteins derived from adaptive kernel density estimates and regressions. *Structure*, 19(6): 844–858,

J. M. Shifman and S. L. Mayo. Exploring the origins of binding specificity through the computational redesign of calmodulin. *Proceedings of the National Academy of Sciences*, 100(23): 13274–9, doi: 10.1073/pnas.2234277100.

A. Sircar, S. Chaudhury, K. P. Kilambi, M. Berrondo and J. J. Gray. A generalized approach to sampling backbone conformations with RosettaDock for CAPRI rounds 13–19. *Proteins: Structure, Function, and Bioinformatics*, 78(15):3115–3123,

D. F. Specht et al. A general regression neural network. *IEEE transactions on neural networks*, 2(6):568–576,

A. D. St-Jacques, M.-E. C. Eyahpaise and R. A. Chica. Computational design of multisubstrate enzyme specificity. *ACS Catalysis*, 9(6):5480–5485,

Y. T. Tamer, I. K. Gaszek, H. Abdizadeh, T. A. Batur, K. A. Reynolds, A. R. Atilgan, C. Atilgan and E. Toprak. High-order epistasis in catalytic power of dihydrofolate reductase gives rise to a rugged fitness landscape in the presence of trimethoprim selection. *Molecular biology and evolution*, 36(7):1533–1550,

A. J. Thomas, M. Petridis, S. D. Walters, S. M. Gheytassi and R. E. Morgan. Two hidden layers are usually better than one. S. 279–290,

J. Vucinic, D. Simoncini, M. Ruffini, S. Barbe and T. Schiex. Positive multistate protein design. *Bioinformatics*, 36(1):122–130,

J. Wang, R. M. Wolf, J. W. Caldwell, P. A. Kollman and D. A. Case. Development and testing of a general Amber force field. *Journal of computational chemistry*, 25(9):1157–1174,

D. M. Weinreich, Y. Lan, C. S. Wylie and R. B. Heckendorn. Should evolutionary geneticists worry about higher-order epistasis? *Current opinion in genetics & development*, 23(6):700–707,

J. O. Wrabl, J. Gu, T. Liu, T. P. Schrank, S. T. Whitten and V. J. Hilser. The role of protein conformational fluctuations in allostery, function, and evolution. *Biophysical chemistry*, 159: 129–141, doi: 10.1016/j.bpc.2011.05.020.

G. Yang, D. W. Anderson, F. Baier, E. Dohmen, N. Hong, P. D. Carr, S. C. L. Kamerlin, C. J. Jackson, E. Bornberg-Bauer and N. Tokuriki. Higher-order epistasis shapes the fitness landscape of a xenobiotic-degrading enzyme. *Nature chemical biology*, 15(11):1120–1128,

J. Yang, I. Anishchenko, H. Park, Z. Peng, S. Ovchinnikov and D. Baker. Improved protein structure prediction using predicted interresidue orientations. *Proceedings of the National Academy of Sciences*,

C. Yanover, M. Fromer and J. M. Shifman. Dead-end elimination for multistate protein design. *Journal of Computational Chemistry*, 28(13):2122–9, doi: 10.1002/jcc.20661.

A. Yousef and N. M. Charkari. A novel method based on physicochemical properties of amino acids and one class classification algorithm for disease gene identification. *Journal of biomedical informatics*, 56:300–306,

H. Zalkin. *The amidotransferases*, volume 66. Wiley, New Jersey, 1993.

X. Zhang, T. Perica and S. A. Teichmann. Evolution of protein structures and interactions from the perspective of residue contact networks. *Current Opinion in Structural Biology*, 23(6): 954–63, doi: 10.1016/j.sbi.2013.07.004.

Acknowledgement

First of all, I would like to thank Prof. Dr. Rainer Merkl for his supervision of my thesis. It was great to discuss all topics and work together on possible solutions. Thank you, for supporting me, whenever it was needed. Additionally, I want to thank you for not relenting, even if I was not convinced of the approach.

Furthermore, a big thank-you to Prof. Dr. Jens Meiler for mentoring this thesis and for your support. Especially for your ideas and input, without your support, this thesis would not have been possible.

My special thanks go not least to Prof. Dr. Reinhard Sterner for his mentoring. Every biochemical analysis and experiments would not have been possible without you. You also managed to make biochemistry understandable for me.

I also would like to thank Prof. Dr. Elmar Lang for his support in machine learning. Your work on the neural network was amazing. Also, a special thank-you goes to you for setting me up on this road during my bachelor and master thesis.

In addition, I would like to thank all colleagues of the Sterner and Merkl groups. Without the great atmosphere you created, none of this would have been possible Especially, a big thank-you to Dr. Florian Semmelmann for your collaboration, dedication, and deep knowledge on the paper of the glutaminase transferases. Furthermore, I would like to thank Thomas Kinateder for your biochemical support and analysis of the new designs.

Also many thanks to my colleagues, Dr. Leonhard Heizinger and Dr. Kristina Straub, for making this time incredibly entertaining. We had so much fun at work and your company made the conferences fun-filled.

I also would like to thank my family for supporting me during the whole time of my studies.

My biggest hugs and thanks I have to give to my wife Ute. Thank you for all the joyful time we spent together and your support not only during this thesis, but also during my studies. I am really grateful for having you!