ChemBioChem

Combining Chemistry and Biology

Chemistry Europe

European Chemical Societies Publishing

Accepted Article

Title: Click, Compute, Create: A Review of Web-based Tools for Enzyme Engineering

Authors: Adrian Tripp, Markus Braun, Florian Wieser, Gustav Oberdorfer, and Horst Lechner

To be cited as: *ChemBioChem* **2024**, e202400092

Link to VoR: https://doi.org/10.1002/cbic.202400092

WILEY-VCH

10.1002/cbic.202400092

Click, Compute, Create: A Review of Web-based Tools for Enzyme Engineering

Adrian Tripp[1], Markus Braun[1], Florian Wieser[1], Gustav Oberdorfer[1,2], Horst Lechner*[1,2]

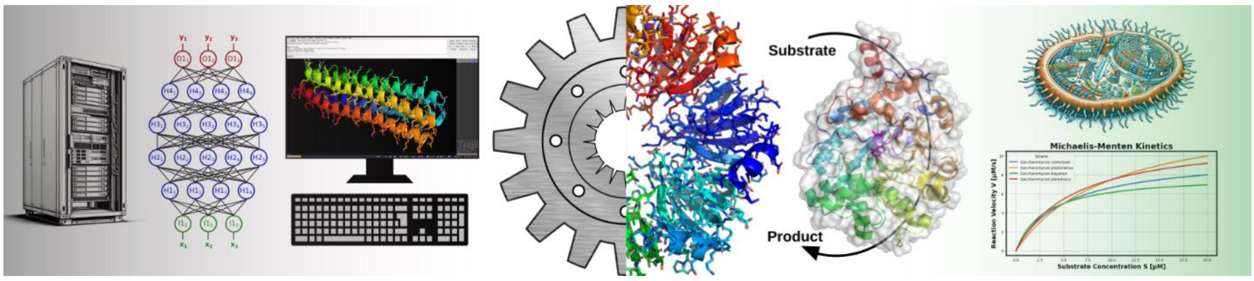
[1] Institute of Biochemistry, Graz University of Technology, Petersgasse 12/2, 8010 Graz, Austria

[2] BioTechMed Graz, Austria

*E-mail: horst.lechner@tugraz.at

TOC:

Reviewing computational methods for Enzyme Engineering in the era of Machine Learning: A concise overview of state-of-the-art methods, ranging from property engineering to the *de novo* design of enzymes, with an emphasis on user-friendly tools that are freely accessible as web-interfaces.


Substrate
Product
Michaelis-Menten Kinetics
Reaction Velocity V [µM/s]
Substrate Concentration S [µM]

ABSTRACT:

Enzyme engineering, though pivotal across various biotechnological domains, is often plagued by its time-consuming and labor-intensive nature. This review aims to offer an overview of supportive *in silico* methodologies for this demanding endeavor. Starting from methods to predict protein structures, to classification of their activity and even the discovery of new enzymes we continue with describing tools used to increase thermostability and production yields of selected targets. Subsequently, we discuss computational methods to modulate both, the activity as well as selectivity of enzymes. Last, we present recent approaches based on cutting-edge machine learning methods to redesign enzymes. With exception of the last chapter, there is a strong focus on methods easily accessible via web-interfaces or simple Python-scripts, therefore readily useable for a diverse and broad community.

frontispiece image:

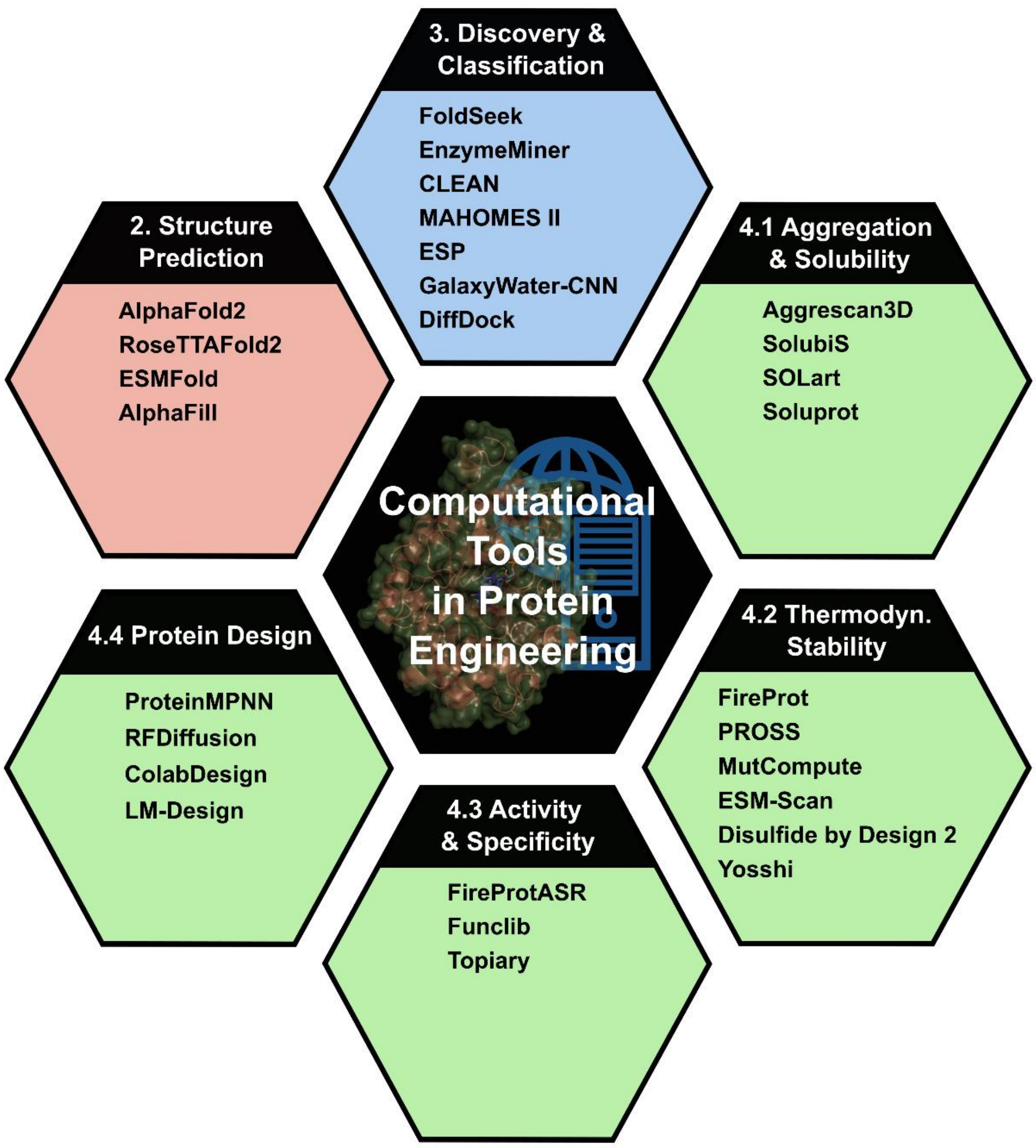


Table 1: Tabular overview of the computational tools for structure predictions

Entry	Name	Link	Ref.	Problems and applications	Input (optimal input)	Output	User-friendliness/rating	Comments
1	AlphaFold2	ColabFold[1]	[2]	Structure prediction	Protein sequence (MSA, template)	Model of protein structure	+++	Preferred, worl best when lots (homologous sequences a available
2	RoseTTAFold2	ColabFold[1]	[3]	Structure prediction	Protein sequence	Model of protein structure	+	
3	ESMFold	ColabFold[1]	[4]	Structure prediction	Protein sequence	Model of protein structure	++	Faster, but le accurate than AF
4	AF_Cluster	https://colab.research.google.com/github/HWaymentSteele/AF_Cluster/blob/main/AFcluster.ipynb	[5]	Prediction of conformational states	Sequence alignment	Models of different states of structures	+	MSA generatic has to be evaluate carefully
5	AlphaFill	https://alphafill.eu	[6]	Addition of ligands to protein models	Structure file	Addition of ligands to given model	+	
6	Metal3D	https://colab.research.google.com/github/lcbc-epfl/metal-site-prediction/blob/main/Metal3D/ColabMetal.ipynb	[7]	Identification of metal ion binding sites and placing them at the correct position	Structure file (residue numbers)	Coordinates of ligands, probability	++	

Table 2. Overview over the tools used for discovery and classification of enzymes

Entry	Name	Link	Ref.	Problems and applications	Input (optimal input)	Output	User-friendliness/rating	Comments
	Online tools							
1	FoldSeek	https://search.foldseek.com	[8]	Find enzymes with similar folds in databases containing experimentally determined as well as predicted structures	Structure file	Sequences of proteins with predicted homologous structures from various databases	+++	
2	EnzymeMiner	https://loschmidt.chemi.muni.cz/enzymeminer/	[9]	Search for new enzymes with known function	Sequence(s) of target enzymes, list of essential residue(s)	Sequences of potential enzymes	+++	
3	CLEAN	https://clean.frontend.mmli1.ncsa.illinois.edu/configuration	[10]	Prediction of EC	Sequence	EC-Number, Confidence level	++	EC number wi low confidenc may is given fo any inp sequence (enzym or not)
4	MAHOMES II	https://mahomes.ku.edu/	[11]	Prediction of enzymatic metal site	Structure file with metal ions	Prediction if metal ions in structure is catalytically active or not	++	
5	Enzyme-Substrate Pair Prediction	https://esp.cs.hhu.de/	[12]	Enzyme-substrate pair prediction	Amino acid sequence and SMILES of ligand	Prediction score, information if ligand was in training set	+	Limited substrates provided list

6	GalaxyWater-CNN	https://galaxy.seoklab.org/cgi-bin/submit.cgi?type=GWCNN_INTRO	[13]	Water prediction	Structure file	Three models with placed water molecules (three different score cutoff values)		
7	DiffDock	https://colab.research.google.com/github/hgbrian/biocolabs/blob/master/DiffDock.ipynb	[14]	Docking	Protein structure, SMILES or PubChem ID	Structures with docked pose(s), confidence score and gnina scores	++	
8	AutoDock Vina	https://durrantlab.pitt.edu/webina/	[15]	Docking	Protein structure, ligand structure	Structures with docked pose(s), Vina scores	+++	Employing loc installation mig be more efficient
9	TurNuP	https://turnup.cs.hhu.de/	[16,17]	k_{cat} prediction, K_M prediction	Protein structure, SMILES or PubChem ID of substrate/product	Predcited value	+	Easy to use, on suited for wildtyp enzyme wi natura substrat results might hav large error due errors in trainin set
	Python-based tools							
10	DeepECtransformer	https://github.com/kaistsystemsbiology/DeepProZyme	[18]	EC prediction			+	Available python an to b run locally only
11	UniKP	https://github.com/Luo-SynBioLab/UniKP	[19]	k_{cat}, K_M and k_{cat}/K_M prediction			+	More precis than web-base tools (TurNuP bit available python an to b run locally only

Table 3. Overview over the tools to engineer enzymes

Entry	Name	Link	Ref.	Problems and applications	Input (optimal input)	Output	User-friendliness/rating	Comments
Aggregation, solubility, thermostability								
1	Aggrescan3D 2.0	https://biocomp.chem.uw.edu.pl/A3D2	[20]	aggregation	Protein structure	Per-residue aggregation propensity plot and color-coded structure, table with suggested variants and the corresponding structures	+++	
2	Disulfide by Design 2.0	http://cptweb.cpt.wayne.edu/DbD2/index.php	[21]	thermostability	Protein structure	Table with suggested variants and the corresponding structures	+++	
3	FireProt	https://loschmidt.chemi.muni.cz/fireprot/	[22]	thermostability	Protein sequence or structure	Table with suggested variants and the corresponding sequences	+++	
4	MutCompute	https://www.mutcompute.com	[23]	thermostability	PDB ID	Color-coded protein structure with all possible variants	+	Only accep structures alread in the PDB
5	PROSS	https://pross.weizmann.ac.il/step/pross-terms/	[24]	thermostability	Protein structure	Table with suggested variants, PyMOL session files for all variants	++	
6	SOLart	http://babylone.ulb.ac.be/SOLART/	[25]	solubility	Protein structure	Scaled solubility value and single feature diagram	+++	
7	SolubiS	https://solubis.switchlab.org/	[26]	aggregation	Protein structure	Plot of TANGO score vs. FoldX dG energy for each APR, structural presentation of APRs, plots of variants for each APR, table of suggested variants with TANGO/FoldX scores	+++	

8	SoluProt	https://loschmidt.chemi.muni.cz/soluprot/	[27]	solubility	Protein sequence(s)	Table with solubility score	+++	Not for varia screening
9	Yosshi	https://biokinet.belozersky.msu.ru/yosshiserver/	[28]	thermostability	Protein structure	Table containing all suggested variants and corresponding PyMOL session file	++	
10	ESM-Scan	https://huggingface.co/spaces/thaidaev/zsp	[29]	thermostability, (soluble) expression	Protein sequence	Heatmap (deep mutational scanning) or list of variants with scores	++	
11	DynaMut	https://biosig.lab.uq.edu.au/dynamut/	[30]	thermostability	Protein structure	Variant score and structural interaction analysis, PyMOL session file, fluctuation plot	+++	No dee mutational scanning optio mutations have be predetermined
Activity, selectivity								
12	FireProtASR	https://loschmidt.chemi.muni.cz/fireprotasr/	[31]	thermostability, activity, specificity	Protein sequence	Ancestral sequences, phylogenetic tree, MSA	++	
13	FuncLib	https://ablift.weizmann.ac.il/step/fl_terms/	[32]	thermostability, activity, specificity	Protein structure	Table containing scores for all variants and corresponding PyMOL session file	+	
14	Topiary	https://topiary-asr.readthedocs.io/en/latest/	[33]	thermostability, activity, specificity	Protein sequence(s)	Ancestral sequences, phylogenetic tree, MSA	+	

Table 4. Recommended tools for enzyme design.

Entry	Name	Link	Ref.	Problems and applications	Input (optimal input)	Output	User-friendliness/rating	Comments
1	ProteinMPNN	https://huggingface.co/spaces/simonduerr/ProteinMPNN	[34]	thermostability, solubility, expression	Protein Structure	Protein Sequence	+++	
2	ZymCTRL	https://huggingface.co/AI4PD/ZymCTRL	[35]	Structure-free enzyme design	EC number	Protein Sequence	+++	
3	ColabDesign	ColabDesign		protein design, general collection	Any	Any	++	Basic python skil are high recommended
4	AFDesign	ColabDesign Notebook		motif scaffolding	None, Protein Structure	Protein Structure	++	
5	AF2Rank	ColabDesign Notebook		model quality assessment	Protein Structure	Structure quality estimate	+++	
6	RFDiffusion	https://github.com/RosettaCommons/Rfdiffusion	[36]	motif scaffolding	None, Protein Structure	Protein Structure	++	

1. Introduction

Proteins govern essential processes in all living organisms and the intricate ways how they do this has been the focus of biotechnological and biomedical research for decades. However, the quantitative prediction of sequence-function relationships in proteins is still one of the main challenges in biology. Nevertheless, enzymes play a central role when developing green chemistry-based technologies, focused on replacing the existing fossil-based chemicals by bio-derived fine chemicals and sustainable products.[37,38] Industrial biocatalysis relies on both engineered natural enzymes and artificial catalysts targeted to achieve sustainable chemistry, alleviating societal problems related to public health and environmental sustainability. In recent years, this active and highly dynamic area of research has largely been driven by readily available machine learning (ML) models adapted to the protein design and engineering problem. To date, several models of varying complexity have been employed to tackle protein backbone generation, sequence optimization and activity enhancement problems. Some of these already show great promise in transforming enzyme design and engineering from a field in which detailed domain knowledge is essential, into an approach that the broader biotechnological community can apply. However, while this research has lately seen a kind of 'gold rush', it has also highlighted the limited availability of computational resources for the unspecialized lab and the lack of detailed understanding of the published protocols, required for their successful use as prediction tools.

ML has revolutionized *de novo* protein structure prediction and likely will continue to do the same for enzyme design and engineering efforts: AI-based algorithms like AlphaFold2[2] or RoseTTAfold[3] provide high-quality predictions – in many cases, even for proteins with novel folds. Recent advancements in deep neural architectures play a key role in applying AI to biotechnology. Traditional architectures include recurrent neural networks such as long short-term memories (LSTMs)[39] operating on sequence data, e.g. for protein property prediction[40]. Graph neural networks (GNNs)[41,42] are a recent type of architecture operating on graph-typed inputs and output, making them a natural choice for biological structures such as proteins and other molecules and were key in recent successes such as protein function prediction[43]. Applying the transformer technology and neural attention mechanisms[44] combined with gigantic datasets enabled the application of large language models to Biotechnology.[4,45] Equally transformative for protein structure modelling are generative models. The goals of generative models are to represent a probability distribution over a chosen data domain (e.g. molecules) to allow the generation of "new" objects, similar to the training data and often to construct a latent embedding or representation of modeled objects. Successful examples include e.g. *variational autoencoders (VAEs)*[46], and diffusion-based models[36,47].

In this review, we present an extensive list of easy to use and well documented tools for enzyme engineering. The individual tools are discussed and briefly explained, providing the reader with a solid overview on state-of-the-art web-based AI, but also non-AI tools for enzyme engineering. Most importantly, all presented resources herein can be used without prior knowledge of the underlying computational procedures and thus should provide a comprehensive resource for the wet-lab biochemist. The recommended tools are gathered in a table in each chapter linking directly to the webservice. We want to stress the pitfalls of these methods as well. Most models (if based on nonlinear functions) are not reliable when predicting values outside of the values used for the training of the model.[48]

2. Structure Prediction

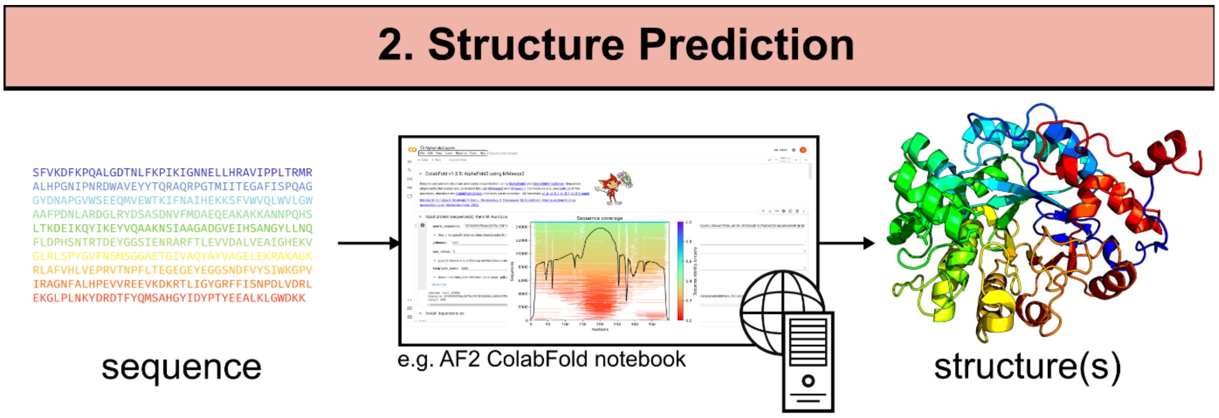

2. Structure Prediction
SFVKDFKPQALGDTNLFKPIKIGNNELLHRAVIPPLTRMR
ALHPGNIPNRDWAVEYYTQRAQRPGTMIITEGAFISPQAG
GYDNAPGVWSEEQMVEWTKIFNAIHEKKSFVWVQLWVLGW
AAFPDNLARDGLRYDSASDNVFMDAEQEAKAKKANNPQHS
LTKDEIKQYIKEYVQAAKNSIAAGADGVEIHSANGYLLNQ
FLDPHSNTRTDEYGGSIENRARFTLEVVDALVEAIGHEKV
GLRLSPYGVFNSMSGGAETGIVAQYAYVAGELEKRAKAGK
RLAFVHLVEPRVTNPFLTEGEGEYEGGSNDFVYSIWKGPV
IRAGNFALHPEVVREEVKDKRTLIGYGRFFISNPDLVDRL
EKGLPLNKYDRDTFYQMSAHGYIDYPTYEEALKLGWDKK
sequence
e.g. AF2 ColabFold notebook
structure(s)

Figure 1. Exemplified concept for structure prediction using current computational methods.

The first set of tools we are describing are programs used to predict structures from given amino acid sequences (Table 1, Figure 1.). Since the team of Google DeepMind developing AlphaFold2[2] (AF2) won the CASP14[49,50] contest (critical assessment of methods of protein structure prediction) in 2021 with an unprecedented increase in accuracy, but also a huge difference to the second-best team, some consider the problem of protein structure prediction from a given sequence as solved. This is not totally true, but the current methods will give in many cases predictions of structures with very high quality. Within a short period of time, several other and in its spirit similar AI-based methods for structure prediction have been developed. Many of them are compiled into a very useful resource called ColabFold provided by Ovchinnikov and Steinegger.[1] The programs are running on resources provided by Google, using their Collaboratory (Colab) Notebooks. These tools are effortlessly accessible and user-friendly for everyone. It's important to note, however, that they are not suited to predict structures on a large scale.

On ColabFold, AF2 is used in combination with MMseqs2[51], allowing the quick generation of a multiple sequence alignment (MSA). MSAs are required by AF2 to gather information about coevolutionary relationships within the sequences. This information is then used to create a distance matrix (distogram) during the progress of prediction, which maps the distances between every amino acid in the given sequence to all others. Using this information and structure templates from the Protein Data Bank (PDB)[52], structures are predicted using a deep and sophisticated convolutional neural network. For multi-chain protein complexes of known stoichiometry, a version of AlphaFold2 (AlphaFold2-multimer) was trained specifically for this purpose and is provided there as well.[53]

Some important remarks about the needed MSAs: General caution is advised if only a very shallow MSA (<30 sequences) can be created. It's essential to acknowledge that single-sequence inputs typically yield lower Template Modeling Scores (TM-score). This metric assesses the similarity between predicted and experimentally determined structures. Recent benchmarks emphasize the inherent challenges associated with achieving high TM-scores for predictions based solely on individual sequences.[54] Using the protein structures of CASP15 (not in the training set for AF2), it was shown that AF2 predictions with an MSA have a median TM-score above 90% while without it, this value drops to 35%. Nevertheless, the score rapidly improves already even if a very shallow MSA containing only a dozen sequences is available.

Another way to tweak the quality of the output is the number of recycles. AF2 refines predicted models by repeatedly feeding the neural network again with the derived MSA, contact map and the gained actual 3D model. The standard value is three recycles before predicting the final structure, but increasing this number (up to 20) can be especially useful for target with a shallow MSA, leading to improved models with a cost in computing time. [1] Additionally templates as structural models might be given as input. But it might have little impact on the quality and output of the prediction if a well-suited MSA was used. If strong coevolutionary signals from an MSA are determined, AlphaFold2 tends to ignore template structures.

Another limitation for the predictions is the size of the memory (RAM) of the graphic processing unit (GPU) used when running ColabFold and thus only sequences possessing less than 2000 amino acids can be predicted on the provided (free-to-use) GPUs.

In the same spirit of AF2 RoseTTAFold2[3] was developed as sequence-to-structure prediction tool, where ideas and considerations of AF2 were independently reproduced and implemented by the group of David Baker. According to a comparison done by the developers of RoseTTAFold2, both systems predicted protein structures equally well. RoseTTAFold2 is available as well *via* ColabFold.

Other methods for structure prediction included in ColabFold are ESMFold[55] and OmegaFold[56]. These methods are not based on MSAs but rather on language models trained on protein sequences. Contrary to the conserved interactions identified in MSAs as done by AF2, the language model was trained with a masked language modeling objective - a random amino acid is covered with a mask and the model has to predict which one fits given the context of the surrounding ones similar to how a text or translation is constructed by various machine-learning text recognition software tools.
The details of these methods are described elsewhere and the interested reader is refereed to reviews explaining either the use of MSA based distograms[57] or language models for proteins[58] in detail.

They outperform AlphaFold2 only when no MSA is available – a rare case in the field of biocatalysis. With an MSA and structural homologs present in the PDB, ESMFold and AF2 perform similarly, although AF2 is usually slightly better.[54] But it seems that the prediction quality is still limited by the number of the sequences of a protein family that the model has seen during training. If no structural homologs but an MSA is present ESMFold performs much worse than AF2.[22] Consequently, "orphan" proteins, possessing sequences hard to get an MSA for, are more difficult to predict for any algorithm. Surprisingly, this is not so much observed for (*de-novo*) designed sequences, where usually MSAs are also hard to obtain. A possible explanation might be that the design process usually aims towards almost perfect foldability to a certain structure with strong and well-defined interactions in the core, possessing no tradeoffs for function in the corresponding sequence.[54]

Another aspect to consider in some cases is speed – important if many structures have to be predicted. ESMFold is roughly 60 times faster than AF2[55], which is useful if hundreds to thousands of structures have to be predicted with the aforementioned tradeoff in accuracy.

Local instances of these tools can be installed using the provided source code, however, all of them need specialized CUDA-encoded GPUs to work properly.

At the end of this chapter, we want to draw your attention to a misuse and misinterpretation of the output of structure prediction programs. The foldability/stability of a given protein upon the introduction of point variants cannot be predicted using any of the presented tools, since they were not designed to evaluate single point variants or variants in general.[59] These concerns also epistatic phenomena, which are not captured within any of these tools.

For tools addressing the topic of the stability of variants, see the next chapter about engineering specific enzyme properties. Depending on the further interpretation and use of the predicted model one should carefully observe the depths of the MSA as well as confidence of the prediction (pLDDT), as only those models with a high pLDDT have a good chance to correctly predict the position of C_α atoms accurately.[60] This is especially important for residues in the region of interest, the active site or an entrance channel. The accuracy of the prediction is further influenced by protein size, amino acid identity and secondary structure. As to expect, AlphaFold2 provides models with high prediction confidence for secondary structure elements like beta-sheet and alpha-helix, but with lower one for unstructured regions.[61] It should be noted that even in models with high pLDDT (> 90) about 10% of residues predicted with very high confidence differ from deposited crystal models by over 2 Å rmsd.[62] Additionally, the pockets or binding sites for cofactors or metal ions are empty and residues might have wrong conformations at these positions.

AlphaFill is a tool to integrate ions or cofactors in the predicted model. But as the authors of the accompanying paper state nicely "*It is good to keep in mind that the AlphaFill models are not very suitable for precise quantification of interactions between the transferred ligand(s) and the protein (for example, hydrogen bonds, π–π or cation–π interactions, van der Waals interactions, hydrophobic interactions, halogen bonds). Namely, this requires coordinate precision that is not provided by either the AlphaFold or the AlphaFill models*."[6]

Metal3D is an option to identify possible coordination sites for selected metal ions within a given protein.[7] It works very well with zinc ions, where the underlying model was trained with, but can be used for many other transition metal ions as well. However, it should be noted that it has its limitations and does not work for example for the identification of coordination site for alkali metals.

AF2 was used to predict multiple conformations at least for some proteins by tweaking the MSA using sequence clustering.[5,63,64] The generation of the MSAs needed is not trivial and might be not feasible for every enzyme/protein of interest. There is a pipeline to using a curated MSA as input yielding potentially conformational states termed AF-Cluster available as Colab notebook.[5] Several other methods to model different conformational states were reviewed recently and might be applicable to some problems.[65]

To sum up – the community has excellent, easily applicable and fast tools in hand, able to predict models with high accuracy. Depending on the application those models are often sufficient and will give great insight into the structural properties of the desired protein. But they have to be evaluated carefully before using them for certain downstream applications and the predicted structures are still models with potential flaws.

3. Discovery and classification

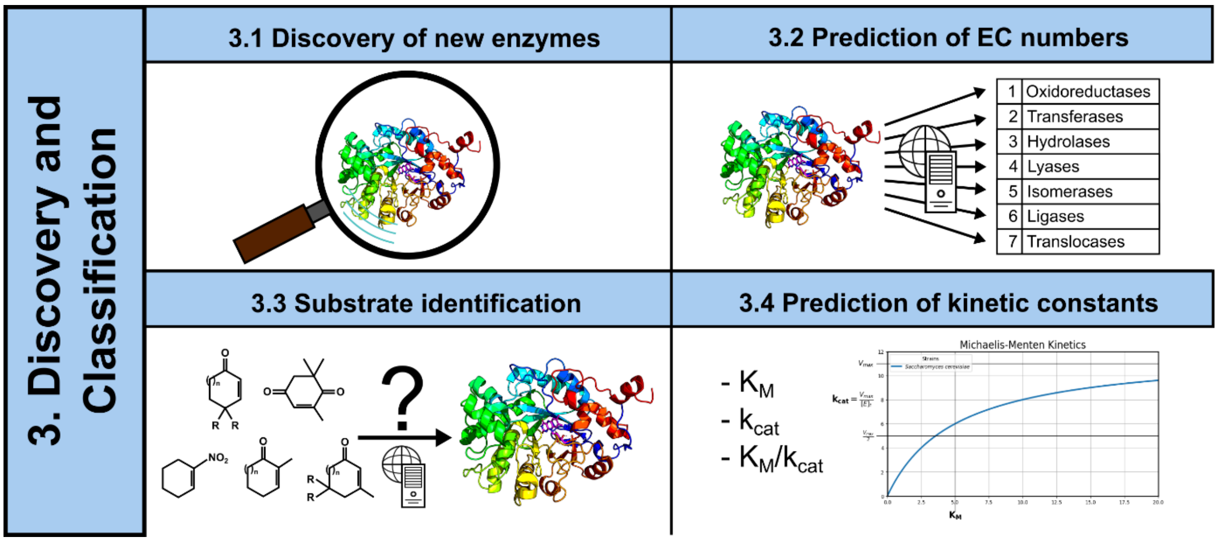

3. Discovery and Classification
3.1 Discovery of new enzymes
3.2 Prediction of EC numbers
1 Oxidoreductases
2 Transferases
3 Hydrolases
4 Lyases
5 Isomerases
6 Ligases
7 Translocases
3.3 Substrate identification
3.4 Prediction of kinetic constants
- K_M
- k_{cat}
- K_M/k_{cat}
Michaelis-Menten Kinetics
Saccharomyces cerevisiae
K_M

Figure 2. Overview about the possible predictions to discovery and classify new enzymes.

The *in-silic*o discovery of new enzymes often starts with exploration of the vast amount of sequencing data from different sources deposited in databases like UniProt[66] and NCBI[67] using methods like the basic local alignment search tool (BLAST).[68,69] Correct annotation of these sequences is still a major problem and relies on comparing protein sequences to organize them into families.[70] Tools to discover new enzymes based on known properties, to predict classifications or substrates and kinetics (Figure 2) are summarized here and in Table 2.

3.1. Discovery of new enzymes

To improve the quality of such queries for the identification of putatively useful members of an enzyme family the Loschmidt laboratories developed a pipeline termed EnzymeMiner[9], using enzyme sequences of interest and a list of essential residues, needed for activity. The pipeline uses BLAST to create an initial alignment of protein sequences. In the next step, the sequence alignments are filtered based on user-defined essential residues, which are necessary for the enzyme's catalytic function. At last, an annotation step is included, where soluble protein expression in *E. coli*, transmembrane regions, the organism of origin, as well as the Pfam-domains are predicted. The obtained sequences are presented and clustered using a sequence similarity network and can be further refined using the self-explanatory provided filter options. The algorithm was successfully applied for the identification of novel, thermostable and highly active haloalkane dehalogenases.[71]

Compared to the databases containing sequence information, databases containing protein structures have been limited in size up until recently. The advent of the structure prediction tools mentioned in the previous chapters remedied that issue. The developers of ESMFold provides a database of 772 million predicted metagenomic protein structures (https://esmatlas.com/)[4] while the team of AF2 collaborated with the European Molecular Biology Laboratory – European Bioinformatics Institute to predict 200 million sequences of the UniProt (https://alphafold.ebi.ac.uk/).[72] Thus it became necessary to develop fast methods like FoldSeek[8] to search those databases for proteins possessing a high structural similarity with a target protein or enzyme. This advancement offers an orthogonal approach for identifying homologous proteins.

3.2. Prediction of enzyme commission numbers

Machine learning has also been employed to address the challenge of predicting the enzymatic activity of sequences that have missing or incorrect annotations. One very promising tool to annotate enzyme commission (EC) numbers is CLEAN (contrastive learning–enabled enzyme annotation).[10] This method was benchmarked by the developers against other state-of-the-art EC-prediction tools including DeepECtransformer[18], a method mentioned below, and showed better prediction performance than all of them. Additionally, it was applied in the identification of an "understudied" enzyme family, the halogenases. Understudied means that the occurrence of these enzymes in databases is low compared to others, making them a more difficult target to annotate correctly. The algorithm was able to identify two enzymes which have not been annotated by any of the compared methods. Furthermore, it could identify a promiscuous enzyme classified with 3 different EC numbers, being able to catalyze three different reactions.

The developers of DeepECtransformer[18], another method for EC number prediction, state that their methods perform slightly worse than CLEAN on some datasets. This program is only available in python and has to be installed and run locally. But they also identified a problem of CLEAN assigning EC numbers for any input amino acid sequences, including even non-enzymatic proteins. When using the webserver, this problem is somehow reflected by the flag indicating a low confidence level for such predictions. They conclude that *"this highlights the importance of considering an ensemble of prediction tools for comprehensive enzyme function characterization*", which is true and applicable for many methods described within this review.

3.3. Substrate identification

Possible substrates of enzymes can be predicted using a webserver for Enzyme-Substrate Pair Prediction (ESP)[12], with the limitation of a substantial drop of model performance for small molecules not included in the library of the training set. This library currently contains ~1400 small molecules and is available for download on their homepage. For molecules that were part of the training set, the model has a very high accuracy of almost 90% to predict substrates for a given enzyme with a sequence identity as low as 40% compared to the ones in the training set.

If the molecule is not included in the library but an educated guess for substrates is possible, docking of those is a viable option as well. There are several new deep-learning based programs under development tackling this task quickly and efficiently. One of those, DiffDock[14], is available as Google Colab notebook, but has to be used with caution. A recent and thorough benchmark study which compares established physics-based docking programs, like AutoDockVina[73] or Gold[74], with several AI based methods, including DiffDock, revealed some problems with those.[75] The authors conclude that deep-learning based docking methods do not generalize well to novel data, struggle with physical plausibility and perform generally worse than the established energy-based methods, both physical plausibility and binding mode RMSD. They speculate this might be caused by overfitting to the protein-substrate pairs they were trained on, a problem which has to be carefully evaluated with every new method based on deep-learning.

The advantage of DiffDock versus AutoDock Vina is the much shorter runtime. Both methods are available as webservices and might be used complementary. If higher accuracy of the results is preferred AutoDock Vina is to prefer.

3.4. Prediction of kinetic constants

Another topic of interest covered by recent machine learning methods is the *in-silico* elucidation of kinetic constants of enzymes like the Michaelis constant (K_M), the turnover number (k_{cat}) and the catalytic efficiency (k_{cat}/K_M).

Most of these methods are provided as python-based repositories on GitHub with either ready-to-use python scripts or tutorials on how to use them. All of them are somewhat hampered from large errors in the dataset used for learning and developing their methods. The authors of two of those models[16,17] highlight that one of the datasets they used (BRENDA) has a mean relative deviation of 3.4-fold between a single K_M measurement and the geometric mean of all measurements, and a mean relative deviation of 5.7-fold for a single k_{cat} measurement compared to the geometric mean of all other measurements for the same enzyme–substrate combination. This leads to problems with the development of an accurate model later on.

The first models were only trained with natural enzyme-substrate pairs appearing in the BRENDA database.[16,17] Therefore, the models are only suitable for predicting the k_{cat} or K_M value if the substrate for a wild-type enzyme is known. They were not trained to predict K_M or k_{cat} values for variants or non-natural substrates. On average their predictions deviate from experimental estimates by 4.1 and 3.3-fold, respectively, also due to the large experimental error as described above. Both models can be employed via a webservice (TurNuP), but we would recommend using them only for the purpose they were developed and to evaluate the results carefully. Nevertheless, both models ignore environmental factors like pH-value and temperature, which can significantly impact enzyme kinetics.

This problem was tackled recently with the development of UniKP[19]. The model behind this program is based on a language model, a different model compared to the previous studies, but still uses the same datasets as all other models for learning. An additional dataset was used to create the EF-UniKP model, where pH values of buffers and temperature of the reactions are included. After careful evaluation and selection of methods, as well as adjustment of the sample weight distributions, their final model performs well on k_{cat}, K_M and even k_{cat}/K_M prediction. It was used in the wetlab to identify tyrosine ammonia lyases (TAL) with higher k_{cat} than a model enzyme. Two out of the top five ranked sequences, in terms of predicted k_{cat}, showed up to 4 times increased turnover numbers. The authors also analyzed *in-silico* designed variants of the model TAL, selecting for the ones with highest predicted k_{cat} and k_{cat}/K_M and identified three more active variants, one with a 3.5-fold increase in catalytic efficiency.[19] The drawback of these method is their non-availability as a no-local resource. They must be installed on a computer with dedicated GPUs and results can be computed easily with the provided documentation.

Other groups have addressed more specialized questions arising during enzyme characterization, achieving high success rates. One question that needs to be answered is whether a protein’s metal binding site has an enzymatic or non-enzymatic function. The answer can be given by a classifier termed MAHOMESII with very high precision and recall.[11] Another aspect of interest might be the placements of water molecules within an active site. GalaxyWater-CNN is a 3D-convolutional network designed for this task.[13]

3.5. Summary

Contrary to structure prediction algorithms, which employ the PDB for learning, the developers of programs for discovery and classification have problems to find suited, well curated datasets. Hence, some of the models have decreased accuracy, while other methods overcame this issue by thoughtful tinkering and adaptions of the algorithms behind their models. Methods for EC classification and kinetic constants predictions are available, ready to use and very helpful, but tend to have a greater uncertainty than other methods described above.

4. Enzyme Engineering

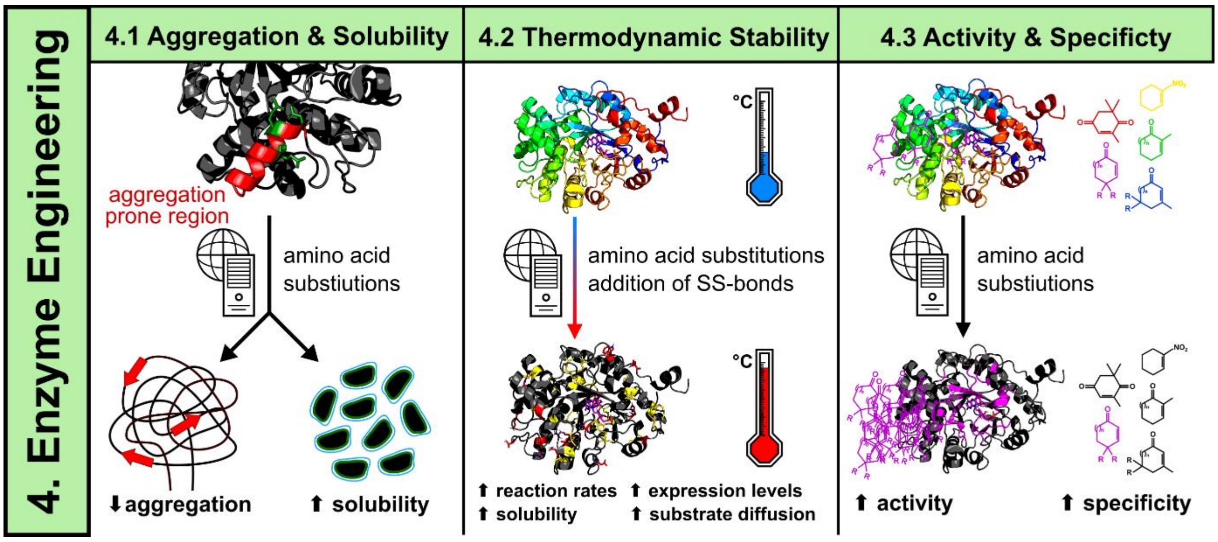


Figure 3. Abilities of current web-based services to engineer given properties of enzymes.

The tools for enzyme engineering shown in Table 3 have been divided groups according to the specific challenge they want to address. In this chapter of our review, we preferentially included tools suggesting tangible amino acid substitutions from sequence or structure information instead of tools evaluating already predetermined variants. The first two parts are focusing on methods to increase the yield of soluble produced protein, reducing aggregation propensity, and improving thermostability while maintaining activity and selectivity of the wild-type enzyme (Figure 3). These tasks are closely related, as both decreasing a protein's aggregation propensity and enhancing thermostability can have a strong positive effect on expression levels. The tools in the third subchapter tackle enzyme-specific properties such as activity and selectivity. The final section covers recent developments in general protein design (Figure 4). These protocols and programs are less easy to use for people outside the protein design field but are increasingly attractive to experimenters. Protein engineering using i*n silico* methods has been a valuable alternative to laborious wet lab screening of vast variant libraries and multiple tools focusing on this topic have been released.

4.1. Aggregation & Solubility

Protein aggregation and insolubility are among the most common reasons obstructing the application of a protein in biotechnology and medicine. Unsurprisingly, a plethora of tools to improve solubility and prevent aggregation have been developed. Typically, these algorithms attempt to identify aggregation prone regions (APRs, sometimes called 'aggregation hotspots') in proteins from either sequence or structural data and suggest variants with reduced aggregation propensity.

Aggrescan3D 2.0[20], a major update of Aggrescan3D[76], performs identification of APRs on a structural level and additionally considers protein flexibility. First, each residue in the protein sequence is assigned an individual aggregation propensity score. This score is calculated using the Aggrescan method[77] (also available *via* webserver[78]), which employs an algorithm based on an experimental aggregation study on a set of amyloid β-peptide variants in *E. coli*. The aggregation propensity score is then mapped on a spherical region centered on the C_α carbon atom of the respective residue in the input structure - this creates a structurally corrected aggregation propensity value (A3D score) by taking into account solvent exposure and the aggregation propensity scores of spatially close residues. If the dynamic mode is selected, Aggrescan3D 2.0 employs the CABS-flex[79] approach, a method to rapidly calculate protein dynamics based on a coarse-grained representation which is also available as an independent webserver (https://biocomp.chem.uw.edu.pl/CABSflex2), to simulate the protein's flexibility, since fluctuations in a protein structure can have an effect on the level of exposure of APRs. The A3D score is calculated for every CABS-flex predicted model in the same way as described above. The model with highest A3D score is returned as the most aggregation prone model in solution. This program has been shown to significantly outperform the sequence-based aggregation propensity prediction algorithms SOLpro[80] and PROSO II[81].

If run in static mode, the Aggrescan3D 2.0 webserver can suggest mutations for residues with high A3D score. The selected residues are exchanged with so-called gatekeeper residues - aspartic acid, glutamic acid, arginine and lysine - that disrupt the APR. The variants are ranked by the energetic effect according to an evaluation with the FoldX force field[82] and the difference in A3D score is compared to the one of the wild-type protein. Additionally, amino acid substitutions can be set manually and are handled in the same way as the automatic suggestions. Aggrescan3D has been used to engineer variants with increased solubility and decreased aggregation propensity for Green Fluorescent Protein (GFP)[83], a human single-domain VH antibody[83], a single-chain variable fragment of anti-PD1 antibody Nivolumab[84] and an alcohol dehydrogenase from *Rhodococcus ruber*[85]. Furthermore, it has been applied in the engineering of nanoscale protein materials[86]. An in-depth review has been published recently[87].

The web server SolubiS[26] operates in a similar fashion as Aggrescan3D 2.0. It uses TANGO[88], a statistical mechanics algorithm for protein aggregation prediction based on the physicochemical principles of beta-sheet formation, to identify APRs on a sequence level and calculates the contribution of each APR to the local stability of the protein using FoldX, a widely employed force field that allows the rapid calculation of the free energy of proteins and nucleic acids. The combination of both scores is called SolubiS score and reflects the lower impact of buried APRs compared to exposed ones in folded, globular proteins. The introduction of gatekeeper residues into exposed APRs is performed in a similar fashion as Aggrescan3D 2.0. However, it does not account for protein flexibility. SolubiS has been used to increase solubility and abundance of protective antigen from *Bacillus anthracis* and human α-galactosidase[89] as well as monoclonal antibodies[90].

Soluprot[27], an ML-based method trained on a preprocessed version of the TargetTrack database[91], specifically addresses the probability of successful soluble (over)expression in *Escherichia coli*. The model characterizes protein sequences in five feature groups, resulting in a total number of 96 features. The groups are single amino acid content, amino acid dimer content, sequence physicochemical features,

10.1002/cbic.202400092

content of amino acids in transmembrane helices as predicted by TMHMM and maximum identity to the *E. coli* PDB subset as calculated using USEARCH. Soluprot achieved the highest accuracy (58%) when tested against several other solubility prediction tools (PROSO II[81], SWI[92], CamSol[93], ESPRESSO[94], rWH[95], DeepSol[96], Protein-Sol[97], SOLpro[80], SKADE[98], ccSOLomics[99], RPSP[100]) on a set of 3100 sequences. The webserver accepts one (or multiple) sequences in FASTA format and returns a solubility score ranging from zero to one. Values above 0.5 indicate soluble expression in *E. coli*, whereas values below 0.5 indicate insoluble expression. One drawback of this approach is that it does not offer recommendations for improvement of the input sequences, however since the results are returned almost instantly it can be used as a complementary method for *in silico* screening of putative enzyme libraries. It has been integrated into EnzymeMiner[9], a webserver for *in silico* screening of enzymes for candidates for wet lab experiments.

SOLart, another solubility predictor, was trained on two solubility datasets from the cell-free expression system PURE[25]. In this case, solubility is defined as the ratio of the protein in the supernatant *versus* the total concentration. The datasets for *E. coli* and *S. cerevisiae* were limited to proteins where a corresponding structure was deposited in the PDB and further filtered by sequence identity and X-ray resolution, resulting in 406 and 59 proteins, respectively. Additionally, 550 (*E. coli*) and 50 (*S. cerevisiae*) homology models obtained from SWISS-MODEL[101] were included in the dataset where no crystal structure was available. 52 features (both sequence- and structure-based) of the *E. coli* dataset containing experimental structures were used to construct the model with a random forest regression algorithm. SOLart significantly outperformed sequence-based solubility predictors as well as the structure-based Aggrescan3D 2.0 when applied to the datasets containing experimental crystal structures for proteins from *S. cerevisiae* and the modeled structures from *E. coli*, showing an average Pearson R correlation coefficient of 0.65. The webserver requires a PDB structure as input and returns a scaled solubility value. Additionally, the effect of the 20 most important features is displayed in a bar chart. It does not, however, suggest variants with increased solubility.

4.2 Thermodynamic Stability

Enhancing protein thermodynamic stability is another common task in enzyme engineering, as natural proteins are often ill-equipped to resist the harsh environments commonly found in industrial applications. Especially for enzymes increased thermostability can be auspicious, as reaction rates, substrate diffusion and solubility are generally improved at higher temperatures. Enhanced thermodynamic stability has also been linked to elevated expression levels[24,102–106]. Typically, single amino acid substitutions lead to only marginal improvement of thermostability[107] necessitating the combination of multiple amino acid substitutions for significant enhancement by adding up individual effects. Non-additive effects described by epistasis, where one mutation depends on the presence of other mutations, may offer higher enhancement at a reduced number of mutations, but are harder to achieve.[108–110] Especially ML-based methods struggle to capture epistatic effects, as these models are typically trained on single amino acid substitution datasets.[111]

FireProt[22], a tool published by the Loschmidt laboratory, combines evolutionary data and structural information to suggest single or multiple position amino acid substitutions on the wild-type protein and is available as a web server[112]. The first step of the pipeline is a BLAST search of the input protein against the UniRef90 database[113] to identify sequence homologs. After clustering by sequence identity and ranking based on the BLAST input query coverage, the best-ranked sequences are selected to create an MSA with Clustal Omega[114]. Based on this MSA, a conservation coefficient for every residue position in the protein is assigned. Additionally, correlated positions are identified and amino acid frequencies at individual positions are analyzed.

The input protein structure is minimized using Rosetta[115]. Positions in the structure, identified to be not conserved or coevolved according to the MSA are subjected to saturation mutagenesis using FoldX and Rosetta. Single amino acid substitutions resulting in a ddG below a given threshold are selected as potential candidates for the generation of variants with multiple amino acid exchanges. A back-to-consensus approach is employed to add additional single amino acid substitution variants to the pool of candidates if the wild-type amino acid differs from the most prevalent one in the MSA based on majority and frequency ratio methods[116].

All pairs of candidates where the amino acid positions are within 10 Å are evaluated separately for the energy- and evolution-based approach *via* Rosetta. The residue pairs are ranked according to ddG and introduced into the wild-type structure until no mutations are left or the stabilizing effect reaches a set threshold. Residue pairs colliding with already accepted pairs are discarded. These steps are repeated for a combination of energy- and evolution-derived variants, resulting in three suggested multiple amino acid substitution variants (one energy-based, one evolution-based, one combination of both). FireProt was experimentally validated on three proteins (haloalkane dehalogenase DhaA, γ-hexachlorocyclohexane dehydrochlorinase LinA, fibroblast growth factor 2) resulting in variants with a shift in melting temperature of 25, 21 and 15°C, respectively. Comparison with PROSS[24] on an experimental haloalkane dehalogenase dataset[117] showed similar results for both methods, correctly identifying 29 (FireProt) and 20 (PROSS) potentially stabilizing positions.

FireProt was recently updated to version 2.0[118], introducing new functionality like focusing on low- or high-risk variants, selecting modifications based on B-factor of the input structure and a pipeline for ancestral sequence reconstruction, a method discussed in more detail below. Additionally, FireProt 2.0 utilizes AlphaFold2 models (if available) or homology models constructed using ProMod3[119] if a sequence instead of a structure is provided as input. The original FireProt version has been used in more than 20 studies – a list including ddG values, shifts in melting temperatures and half-life improvement over the wild-type protein can be found in the supporting information of the FireProt 2.0 publication[118].

Another tool for the improvement of thermodynamic stability is PROSS (Protein Repair One Stop Shop)[24]. Similar to FireProt, its algorithm combines atomistic Rosetta modeling and phylogenetic sequence information. Out of an MSA of the input sequence, a position-specific substitution matrix (PSSM)[120] is calculated containing the log-likelihood values of observing any of the 20 amino acids for each position in the sequence. Possible amino acid substitutions are restricted to the ones with a favorable PSSM score. Each allowed substitution is modeled using Rosetta and ddG values compared to the wild-type input structure are calculated. Single position substitutions below a given ddG threshold are collected and combinations of these are evaluated using Rosetta combinatorial sequence design.

Initially, PROSS was evaluated on an experimental dataset of single point variants of the enzymes fungal endoglucanase Cel5A and yeast triosephosphate isomerase (TIM). It successfully eliminated all severely destabilizing substitutions and 99.6% of all destabilizing substitutions. One third (Cel5A) and two thirds (TIM) of stabilizing substitutions were successfully identified. Since then, PROSS has been extensively benchmarked[121,122]. PROSS has been applied to a plethora of targets, a list of the corresponding studies compiled by the creators of PROSS and FuncLib can be found in this document[123].

An extensive study comparing different stabilization strategies (PROSS, FireProt, *in silico* saturation mutagenesis and rational disulfide bridge design) on the model enzyme haloalkane dehalogenase DhaA115 has been published recently[122]. Both PROSS (11 to 36 substitutions) and FireProt (3 to 8 substitutions) succeeded in generating multiple point variants with increased thermal stability (thermal shifts of the top variants of 5.1°C and 3.6°C, respectively), whereas the introduction of disulfide bridges and *in silico* saturation mutagenesis failed to do so. Interestingly, manual rational selection of suggested single point variants of these methods to create a multiple point variant yielded variants with even higher positive shift in melting temperature. Filtering PROSS single point variants by their MutCompute score (MutCompute is described below) and then combining them to create a multiple point variant resulted in a DhaA variant with a melting temperature of 81.7°C ± 0.9 (8.4°C higher than the original variant Dha115), the most stable variant known to date. Notably, the catalytic activity at 60°C of variants with manual or MutCompute selection steps was increased compared to Dha115.

MutCompute[23] is a three-dimensional self-supervised convolutional neural network trained to associate amino acids with the local chemical microenvironments in a protein structure. If an amino acid in the wild-type structure does not fit its microenvironment, it can suggest variants with improved intermolecular interactions at this position. In contrast to many other ML-based protein engineering tools, MutCompute does not rely on analysis of existing deep mutational scanning studies on the target protein and is not restricted by limited availability of annotated amino acid substitution training datasets. The neural network is based on the architecture published by Torng and Altman[23] and additionally considers information like explicit hydrogens and biophysical attributes like partial charge and solvent accessibility on an atomic level. Together with careful evaluation of the training set, this results in the correct identification of the wild-type residues in 87% of all cases when performed on targets without any possible beneficial amino acid substitutions as proven by deep mutational scanning. MutCompute has been used to identify gain-of-function variants with improved thermodynamic stability for blue fluorescent protein, phosphomannose isomerase, TEM-1 β-lactamase[23], PETase[124] and Bst DNA polymerase[125].

MutCompute is available as a webserver. In its current implementation, account creation is required, which is only possible for academic users. It solely accepts protein structures already uploaded to the PDB. The selected protein structure will be displayed color-coded according to how well each wild-type amino acid fits to its microenvironment, blue indicating residues that show good agreement and red indicating bad agreement, suggesting promising targets for amino acid substitutions[126]. For every position, MutCompute provides a score of the wild-type residue and the amino acid predicted to fit best.

To study the effect of missense mutations on protein stability, DynaMut2[30] was developed. It combines normal mode analysis, a graph-based representation of protein structure and other features to predict changes in thermal stability for single- and multiple-point amino acid substitutions. It was trained on data from the ProTherm database[127], which contains thermodynamic parameters related to protein stability, and requires a protein structure in PDB format as input. DynaMut2 achieved an overall Pearson R correlation coefficient for single amino acid substitutions of 0.68 (root mean square error (RMSE) 1.14 kcal/mol). This value dropped to 0.51 (RMSE 1.02 kcal/mol) and 0.62 (RMSE 0.91 kcal/mol) when considering only stabilizing and destabilizing substitutions, respectively. Slightly lower values were achieved for variants with multiple amino acid substitutions. The webserver outputs a ddG value for each variant and offers a structural representation of residue interactions.

ESM-Scan[29], recently published as a preprint, is a tool for protein optimization based on the ESM family of protein language models. It was trained as a masked language model and assigns each residue in an input protein sequence a probability score based on its sequence context. It can be used in one of four different ways: I) score a single or multiple specific substitutions (e.g. R57K R57E A101M), II) evaluate all possible substitutions at a certain position, III) if provided a second sequence of the same length, evaluate every substitution one by one, IV) perform *in silico* deep mutational scanning of the full sequence. ESM-Scan is available *via* a user-friendly web interface hosted on huggingface.com. Evaluation was performed on three experimental datasets, demonstrating Pearson R correlation coefficients of 0.44 to 0.56 for datasets containing thermostability, expression levels and activity data. Correlation dropped significantly when performed for amino acid substitutions at protein-protein interfaces (Pearson R coefficient of 0.06 to 0.09).

Another method for stabilization of proteins is disulfide engineering. The covalent bond formed between thiol groups of two cysteine residues in close proximity have been shown to stabilize proteins in the range of 2.3 to 5.2 kcal/mol, attributed to the loss of conformational entropy in the unfolded state.[128–130] Typically, disulfide bonds are introduced into regions of medium to high mobility (identified by a high B-factor in the crystal structure) to maximize their stabilization potential. With Disulfide by Design

2.0[21], positions for cysteine substitutions are selected according to the geometric constraints a disulfide bond imposes. However, the success rate of this approach is rather low[130]. In addition to selecting geometrically viable positions, Yosshi (Your webserver for S-S bond harvesting)[28] limits potential positions for disulfide engineering to sites found in structural homologs that also have a disulfide bond. It requires the structure of the protein of interest in PDB format and an MSA of structural homologs in FASTA format. The latter can be easily created using Mustguseal, a webserver created by the same group[131]. The output variants are evaluated according to a 3D motif analysis of how close the identified positions for cysteine substitution resemble existing disulfide-forming cysteines in the PDB. The geometry-based filtering steps of Yosshi were evaluated on a dataset comprised of protein structures containing either true S-S bridges or cysteines in close proximity but not forming disulfide bonds. Using the strictest parameters for disulfide identification, the reported sensitivity and specificity were 78.6% and 100%. Additionally, variants with increased stability were correctly predicted using Yosshi according to existing datasets for subtilisin E, sperm whale myoglobin, lipases from *Proteus mirabilis* and *Penicillium cyclopium*, human carbonic anhydrase and xylanases from *Bacillus circulans* and *Thermomyces lanuginosus*. Homology-based disulfide engineering has been successfully employed to stabilize L-threonine aldolase[132], a lipase[133], immunoglobulin heavy chain variable domains[134] and subtilisin E[135], among others.

4.3 Activity & Specificity

Ancestral sequence reconstruction (ASR) has proven to be another highly effective way of protein stability engineering. Moreover, ancestral enzymes have been associated with increased substrate scope compared to extant counterparts[136–138], however a direct link is still under debate[139,140]. This approach requires an MSA, from which a phylogenetic tree is inferred to identify sequences of extinct proteins. An extensive review focusing on ASR as a method for thermostability enhancement has been published recently[141]. FireProtASR[31], created by the Loschmidt group, offers a fully automated workflow for inexperienced users by covering the MSA and phylogenetic tree creation steps as well as ancestral inference[142]. This tool was experimentally validated on haloalkane dehalogenase from *Rhodococcus rhodochrous* by probing expression levels, solubility, yield, melting temperatures and haloalkane dehalogenase activity. Moreover, a screening for luciferase activity, which is not present in the wild-type enzyme, was performed because haloalkane dehalogenases and luciferase enzymes were found to have common ancestors. The ancestral proteins showed similar expression levels, solubility, and yield as the wild-type protein. Melting temperatures were significantly increased in five out of the six characterized enzymes (76.2 ± 0.2 °C for the top performer *versus* 50.6 ± 2.4 °C for the wild-type enzyme) as was HLD activity in three enzymes (0.061 µmol/mg·s for the two top performers *versus* 0.032 µmol/mg·s for the wild-type enzyme). Additionally, three enzymes showed luciferase activity.

An alternative to FireProtASR for ancestral sequence reconstruction is Topiary[143]. It accepts one or multiple input sequences in the form of a spreadsheet to set the scope for the creation of the initial MSA.

All steps are species-aware to optimize the creation of the evolutionary tree considering both protein and organismal evolutionary signals. Topiary is hosted on Google Colab[123] and can alternatively be installed locally and controlled *via* command line or Jupyter Notebooks.

FuncLib[32], an online tool published by the same group that created PROSS, was developed to create multipoint variants at enzyme active sites. It specifically addresses the issues of epistasis and stability threshold effects to create variants with increased activity and novel substrate scopes. Stability threshold effects describe the increased demand for stabilizing interactions in- or outside the enzyme active site to accommodate function-enhancing, but possibly destabilizing amino acid substitutions [144–147].

The first step in the FuncLib engineering approach is the creation of an MSA for the protein of interest. Out of this MSA, a position specific scoring matrix (PSSM) is created that limits the number of amino acids available for each position that should be diversified. These positions have to be set manually and are typically part of the first or second shell in an enzyme active site but should not contain residues essential for catalysis. Each possible substitution is scored individually using Rosetta and removed if it significantly destabilizes the protein. All possible combinations of amino acid substitutions (with an adjustable cutoff for minimum and maximum number of substitutions per variant) are modelled using Rosetta and ranked according to their predicted stability. There are two major ways of running FuncLib: Either with a substrate present in the input crystal structure with the primary goal of increasing activity for this specific substrate, or without substrate to broaden the substrate scope. FuncLib was initially evaluated on the metalloenzyme phosphotriesterase (PTE) from *Pseudomonas diminuta* and *Salmonella enterica* acetyl coenzyme A synthetase (ACS) which catalyzes a multi-step reaction. For both targets, variants with high expression yield and activity with substrates where the wild-type enzyme showed little or no activity at all were found. Since then, FuncLib has been used on many different targets (often in conjunction with PROSS). A list containing the corresponding studies compiled by the creators of PROSS and FuncLib can be found here[148].

Additionally, FuncLib has been employed to stabilize regions that were identified to energetically suboptimal by pSUFER (protein Strain, Unsatisfactoriness, and Frustration findER)[149], another tool created by the Fleishman group specifically developed for *de novo* enzyme design. pSUFER employs the Rosetta energy function to model all single point variants at every position while iterating sidechain packing and whole-protein minimization to calculate ddG values relative to the wildtype protein. Positions with a number of stabilizing amino acid substitutions above a certain threshold are considered potentially suboptimal. The combination of pSUFER and FuncLib was successfully employed to rescue failed designs for glycoside hydrolase 10 (GH10) xylanases generated *via* modular assembly[150] and protein binders designed in a bottom-up approach.[151]

10.1002/cbic.202400092

4.4 Protein Design Tools for Enzyme Engineering

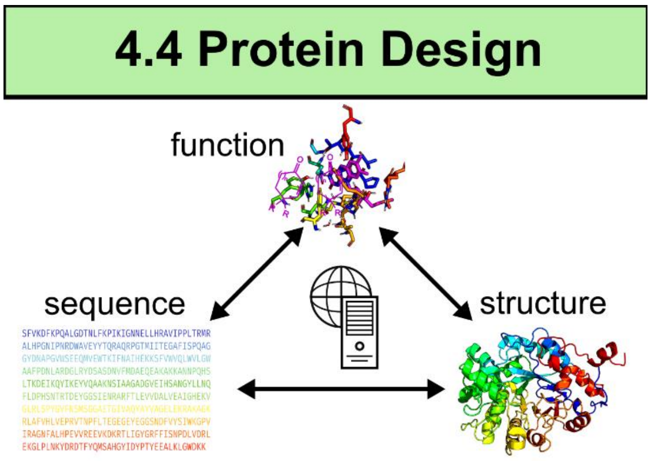


Figure 4: Protein design as a method to engineer function in the context of sequence and structure.

The previous section focused on accessible tools to engineer properties of enzymes. Most of these tools are fully automated and implemented on webservers with graphical user interfaces, while all algorithms are kept in the background. The algorithms and the datasets they were trained on are geared towards one highly specific task, like protein solubilization, stabilization, and activity. This high degree of specialization tends to require little input from the user. Compared to that, protein design tools summarized in Table 4 are typically much more general. Most of them were developed to solve generic problems in macromolecular modeling, like protein backbone generation, energy minimization, fixed-backbone sequence design, or structure prediction. With this level of generality, a large degree of customization is required for their use. This is desired and necessary for the challenges addressed by protein design, but it limits their applicability for "quick" protein engineering purposes. Every changeable parameter in a design protocol can potentially become a source of error in an enzyme engineering project. Outside of integrated workflows established on webservers, protein design (Figure 4) has thus historically not been considered viable for quick and robust engineering of natural enzymes. While we highly encourage our readers to learn about protein design, the focus of this review is applicability in protein engineering. For this reason, we want to direct the readers interested in protein design to other excellent reviews about the topic[152,153]. Nevertheless, we want to highlight one sequence design method that is currently gaining traction for enzyme engineering and briefly outline examples from a more sophisticated collection of design tools with a supportive community and some descriptive examples in the form of Colab notebooks.

Generating amino acid sequences that fold into a pre-specified protein backbone is called fixed-backbone sequence design. Methods for fixed-backbone sequence design have recently become sufficiently robust to be applicable for enzyme engineering. By far the most experimentally validated of them is ProteinMPNN[34], a geometric graph neural network. The network was trained to reproduce native sequences of proteins in the PDB when given their 3D coordinates as input. The final model was able to recover on average roughly 50% native sequence identity, state-of-the art at the time the model was published. In the original publication, the viability of ProteinMPNN was demonstrated by rescuing failed designs of previous methods for sequence generation. Since then, multiple papers corroborated its initial findings. A flavin binding fluorescent protein was stabilized by redesigning all but the amino acids close to flavin. Three out of three tested genes were thermally more stable than the wild-type, while fluorescent properties were still intact.[154] Another group successfully enhanced Rsp5 E3 ligase activity by designing thermally stable Ubiquitin variants that were binding Rsp5 at low micromolar affinity.[155] Finally, ProteinMPNN's effectiveness to engineer enzymes was demonstrated by creating stabilized myoglobin variants and variants of TEV protease with increased catalytic activities.[156] Some of the engineered TEV protease variants had higher catalytic activity than wild-type TEV and also exhibited higher thermal stability and retained their activity for a longer time at room temperature. Their strategy utilized an MSA to identify evolutionarily conserved residues, and then allowed ProteinMPNN to remodel the remaining sequence. This can be easily replicated once an MSA is obtained using an intuitive interface deployed on huggingface that has a verification step with AF2 already built in. Additionally, ProteinMPNN is widely available within Colab notebooks. For more custom design applications, it can be installed directly from the official GitHub repository. The repository is well documented and contains multiple helper scripts and examples for a variety of design problems. In its latest iteration, the ProteinMPNN architecture was extended with the ability to model small molecule interactions and a version of ProteinMPNN that was trained on a subset of the PDB containing only soluble protein structures.[157] The new, highly capable version was published as LigandMPNN [158]and, like its predecessor, can be installed easily from GitHub.

Computational generation of enzyme variants can be quick and cheap, and is not limited to predefined fixed backbones alone. Efficient free generation of enzyme sequences for specified enzyme classes was demonstrated with a network called ZymCTRL which is based on ideas from natural language processing.[35] ZymCtrl designs putative enzyme variants on consumer GPUs within seconds and, remarkably, it creates these sequences with only an EC number as input. The tool is accessible through a huggingface API or can be installed with the transformers python package for larger testing purposes. Although ZymCtrl can efficiently generate novel variants for enzymes within EC numbers, it was validated only *in silico* using a convolutional neural network for protein function prediction (ProteInfer).[159] This leaves experimental confirmation of the ability of ZymCtrl to generate *active* enzymes still pending.

For some natural enzymes, fixed-backbone sequence design and standard protein engineering methods might be ineffective. In such cases, advanced protein design methods can help transferring their enzymatic activity onto stable protein scaffolds. Several methods for this task were developed recently[36,47,160–162], building on a variety of sophisticated deep learning architectures. Effectiveness of these tools varies, and currently, the most reliable and versatile choice for active-site scaffolding is RFDiffusion. RFDiffusion robustly generates protein backbones that hold an active site motif as specified in the input[36]. The diverse capabilities of RFDiffusion are very well documented and the tool can be installed from GitHub.

For experimentation with full protein design workflows, we want to highlight a repository that offers several other essential protein design tools alongside RFDiffusion: ColabDesign. This repository was initialized by one of the authors of ColabFold, and like ColabFold, its code was designed for ease of use and practicality. No designated publication about ColabDesign has been released so far, but methods of several publications link to the repository or are implemented within it. Among the first of them was AFDesign[163], which inverted the AlphaFold2 structure prediction network for structure generation. Their approach for structure generation was later refined by sampling from a continuous sequence space during structure search[164]. This structure search in relaxed sequence space exhibited competitive performance to other protein structure generation algorithms.

AF2Rank is another useful method available within ColabDesign to evaluate the accuracy of computational models of proteins.[165] This method feeds the structure of the protein model as a sequence-masked template to AF2 in single sequence mode. A combination of AF2's confidence metrics after prediction then serves as AF2Rank's accuracy estimate. Their approach exceeded previous methods for estimation of model accuracy (EMA) on the CASP14 EMA dataset. Most protocols implemented in ColabDesign - including protocols for running RFDiffusion, ProteinMPNN, and AlphaFold2 - can be tested within Colab Notebooks which are available as examples in the repository. Their code is continually updated and supported by a strong community on a Discord server linked through the repository. This provides an excellent point of entry for less traditional enzyme engineering projects that focus on solution of problems that are highly specific to a given enzyme. This could involve, for example, remodeling of disordered loop regions or the removal of heterooligomeric interfaces of natural enzymes to drastically alter expression levels and solubility.

CONCLUSION & OUTLOOK

The amount of readily available and simple to use *in silico* tools to discover, characterize and improve enzymes increased rapidly over the last years. This comes as a consequence of AI tools being recognized as methods that can be applied to virtually any challenge in the field of biocatalysis and enzyme engineering. Moreover, it was only through the advent of large, standardizes databases, available for public use, that enabled training of these models. Additionally, many institutions and companies provide the necessary computational resources to host these tools - another prerequisite for their broad applicability.

Scaling laws – the observation that larger amounts of compute during training lead to proportional increases of model capabilities – provide another glimpse into a possible area of large improvement for future models. The limits of emergent capabilities in large language models like the GPT-series for natural language are currently a big area of speculation. Protein language models raise a similar issue. For example, the attention maps of protein language models were found to encode protein structure[166], a property that was exploited using the ESM2 models to build the ESMFold structure prediction networks.[4] The reason for the transformer's encoding of protein structure in its attention maps during training was supposed to be a result of the masked language modelling training objective. Prediction of amino acid probabilities at masked positions was simply more accurate when the network could infer the local structural environment. Similarly, encoding of catalytic features might be the most efficient way to solve the task of catalytic rate prediction for future deep learning models. However, compared to the large genomic databases used to train protein language models, the available datasets to train models on catalytic properties of enzymes are far less refined. This is a result of many factors including the large number of variables that can distort catalytic efficiency and the high cost of obtaining high-quality information for individual enzyme variants. For this reason, we think that future advances into enzyme engineering based on deep learning will not only be driven by algorithmic progress but require coordinated efforts between academia and industry to build standardized, high-quality datasets. This will include datasets linking enzyme variants and function, and datasets that contain large amounts of accurate small-molecule structural data.

In addition to the number of computational tools, also their complexity has increased. The labs developing them often lack the capacities for experimental validation, leaving this task to the community. Yet, experimental data is crucial to validate and subsequently improve the predictive capacity of computational tools, especially when deep learning is involved. We think that the higher numbers of experiments in the biochemical community makes the availability of well documented and evaluated computational tools crucial more than ever. Consequently, we aimed to not only provide an index of useful tools to the community of protein engineers, but also to emphasize how important it is for the developing community to make their tools easily accessible.

The results of the newly developed programs and tools are often astonishing and might be simplifying enzyme engineering pipelines. However, the drawbacks and pitfalls of these models should be considered as well. We point out that predictions and models are *predictions, not the ground truth*. One of the limitations of some of the methods is their confidence in the predicted results. Especially concerning are predicted values outside the of the range used for training. Therefore, we suggest to predict the desired enzyme properties with multiple methods, if available, and comparing the results to check for agreement. Careful evaluation of the predictions and internal benchmarking is advised. In general, wet-lab data already created for similar systems and not included in the training data for AI-based methods serve as good tests for reliability of these methods.

Prediction accuracy for protein structure and function improved by a large margin, as seen with AF2 and other methods like PROSS and FireProt, which were successfully applied to dozens of proteins. We are confident that *in-silico* methods are ready to be used for every enzyme discovery and engineering attempt and that there will be more to come in the near future.

Acknowledgments

10.1002/cbic.202400092

Funding from a FETOPEN project (ARTIBLED, H2020-FETOPEN-2018-2019-2020-01ID 863170), an ERC StartingGrant (HelixMold, EU-ERCID 802217) and from BioTechMed Graz for a YoungResearcherGroup is gratefully acknowledged.

This research was funded in whole, or in part, by the Austrian Science Fund (FWF) [grant DOI 10.55776/DOC130]. For the purpose of open access, the author has applied a CC BY public copyright licence to any Author Accepted Manuscript version arising from this submission.

[1] M. Mirdita, K. Schütze, Y. Moriwaki, L. Heo, S. Ovchinnikov, M. Steinegger, *Nat Methods* **2022**, *19*, 679–682.

[2] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko, A. Bridgland, C. Meyer, S. A. A. Kohl, A. J. Ballard, A. Cowie, B. Romera-Paredes, S. Nikolov, R. Jain, J. Adler, T. Back, S. Petersen, D. Reiman, E. Clancy, M. Zielinski, M. Steinegger, M. Pacholska, T. Berghammer, S. Bodenstein, D. Silver, O. Vinyals, A. W. Senior, K. Kavukcuoglu, P. Kohli, D. Hassabis, *Nature* **2021**, *596*, 583–589.

[3] M. Baek, I. Anishchenko, I. R. Humphreys, Q. Cong, D. Baker, F. DiMaio, *bioRxiv* **2023**, 2023.05.24.542179.

[4] Z. Lin, H. Akin, R. Rao, B. Hie, Z. Zhu, W. Lu, N. Smetanin, R. Verkuil, O. Kabeli, Y. Shmueli, A. dos Santos Costa, M. Fazel-Zarandi, T. Sercu, S. Candido, A. Rives, *Science (1979)* **2023**, *379*, 1123–1130.

[5] H. K. Wayment-Steele, A. Ojoawo, R. Otten, J. M. Apitz, W. Pitsawong, M. Hömberger, S. Ovchinnikov, L. Colwell, D. Kern, *Nature 2023* **2023**, 1–8.

[6] M. L. Hekkelman, I. de Vries, R. P. Joosten, A. Perrakis, *Nat Methods* **2023**, *20*, 205–213.

[7] S. L. Dürr, A. Levy, U. Rothlisberger, *Nature Communications 2023 14:1* **2023**, *14*, 1–14.

[8] M. van Kempen, S. S. Kim, C. Tumescheit, M. Mirdita, J. Lee, C. L. M. Gilchrist, J. Söding, M. Steinegger, *Nat Biotechnol* **2023**, 1–4.

[9] J. Hon, S. Borko, J. Stourac, Z. Prokop, J. Zendulka, D. Bednar, T. Martinek, J. Damborsky, *Nucleic Acids Res* **2021**, *48*, W104–W109.

[10] T. Yu, H. Cui, J. C. Li, Y. Luo, G. Jiang, H. Zhao, *Science (1979)* **2023**, *379*, 1358–1363.

[11] R. Feehan, M. Copeland, M. W. Franklin, J. S. G. Slusky, *Protein Science* **2023**, *32*, e4626.

[12] A. Kroll, S. Ranjan, M. K. M. Engqvist, M. J. Lercher, *Nat Commun* **2023**, *14*, DOI 10.1038/s41467-023-38347-2.

[13] S. Park, C. Seok, *J Chem Inf Model* **2022**, *62*, 3157–3168.

[14] G. Corso, H. Stärk, B. Jing, R. Barzilay, T. Jaakkola, “DiffDock: Diffusion Steps, Twists, and Turns for Molecular Docking,” can be found under https://arxiv.org/abs/2210.01776v2, **2022**.

[15] Y. Kochnev, E. Hellemann, K. C. Cassidy, J. D. Durrant, *Bioinformatics* **2020**, *36*, 4513–4515.

[16] A. Kroll, Y. Rousset, X. P. Hu, N. A. Liebrand, M. J. Lercher, *Nat Commun* **2023**, *14*, 1–14.

[17] A. Kroll, M. K. M. Engqvist, D. Heckmann, M. J. Lercher, *PLoS Biol* **2021**, *19*, e3001402.

[18] G. B. Kim, J. Y. Kim, J. A. Lee, C. J. Norsigian, B. O. Palsson, S. Y. Lee, *Nat Commun* **2023**, *14*, 1–11.

[19] H. Yu, H. Deng, J. He, J. D. Keasling, X. Luo, *Nat Commun* **2023**, *14*, 1–13.

[20] A. Kuriata, V. Iglesias, J. Pujols, M. Kurcinski, S. Kmiecik, S. Ventura, *Nucleic Acids Res* **2019**, *47*, W300–W307.

[21] D. B. Craig, A. A. Dombkowski, *BMC Bioinformatics* **2013**, *14*, 0–6.

[22] D. Bednar, K. Beerens, E. Sebestova, J. Bendl, S. Khare, R. Chaloupkova, Z. Prokop, J. Brezovsky, D. Baker, J. Damborsky, *PLoS Comput Biol* **2015**, *11*, 1–20.

[23] R. Shroff, A. W. Cole, D. J. Diaz, B. R. Morrow, I. Donnell, A. Annapareddy, J. Gollihar, A. D. Ellington, R. Thyer, *ACS Synth Biol* **2020**, *9*, 2927–2935.

[24] 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, S. J. Fleishman, *Mol Cell* **2016**, *63*, 337–346.

[25] Y. Shimizu, T. Kanamori, T. Ueda, *Methods* **2005**, *36*, 299–304.

[26] J. Van Durme, G. De Baets, R. Van Der Kant, M. Ramakers, A. Ganesan, H. Wilkinson, R. Gallardo, F. Rousseau, J. Schymkowitz, *Protein Engineering, Design and Selection* **2016**, *29*, 285–289.

[27] J. Hon, M. Marusiak, T. Martinek, A. Kunka, J. Zendulka, D. Bednar, J. Damborsky, *Bioinformatics* **2021**, *37*, 23–28.

[28] D. Suplatov, D. Timonina, Y. Sharapova, V. Švedas, *Nucleic Acids Res* **2019**, *47*, W308–W314.

[29] M. G. Totaro, U. Vide, R. Zausinger, A. Winkler, *bioRxiv* **2023**, DOI 10.1101/2023.12.12.571273.

[30] C. H. M. Rodrigues, D. E. V. Pires, D. B. Ascher, *Protein Science* **2021**, *30*, 60–69.

[31] M. Musil, R. T. Khan, A. Beier, J. Stourac, H. Konegger, J. Damborsky, D. Bednar, *Brief Bioinform* **2021**, *22*, 1–11.

[32] O. Khersonsky, R. Lipsh, Z. Avizemer, Y. Ashani, M. Goldsmith, H. Leader, O. Dym, S. Rogotner, D. L. Trudeau, J. Prilusky, P. Amengual-Rigo, V. Guallar, D. S. Tawfik, S. J. Fleishman, *Mol Cell* **2018**, *72*, 178-186.e5.

[33] K. N. Orlandi, S. R. Phillips, Z. R. Sailer, J. L. Harman, M. J. Harms, *Protein Science* **2023**, *32*, e4551.

[34] J. Dauparas, I. Anishchenko, N. Bennett, H. Bai, R. J. Ragotte, L. F. Milles, B. I. M. Wicky, A. Courbet, R. J. de Haas, N. Bethel, P. J. Y. Leung, T. F. Huddy, S. Pellock, D. Tischer, F. Chan, B. Koepnick, H. Nguyen, A. Kang, B. Sankaran, A. K. Bera, N. P. King, D. Baker, *Science (1979)* **2022**, *378*, 49–56.

[35] G. Munsamy, S. Lindner, P. Lorenz, N. Ferruz, in *NeurIPS*, **2022**.

[36] J. L. Watson, D. Juergens, N. R. Bennett, B. L. Trippe, J. Yim, H. E. Eisenach, W. Ahern, A. J. Borst, R. J. Ragotte, L. F. Milles, B. I. M. Wicky, N. Hanikel, S. J. Pellock, A. Courbet, W. Sheffler, J. Wang, P. Venkatesh, I. Sappington, S. V. Torres, A. Lauko, V. De Bortoli, E. Mathieu, S. Ovchinnikov, R. Barzilay, T. S. Jaakkola, F. DiMaio, M. Baek, D. Baker, *Nature 2023 620:7976* **2023**, *620*, 1089–1100.

[37] S. Simić, E. Zukić, L. Schmermund, K. Faber, C. K. Winkler, W. Kroutil, *Chem Rev* **2022**, *122*, 1052–1126.

[38] K. Faber, W. Kroutil, *Curr Opin Chem Biol* **2005**, *9*, 181–187.

[39] S. Hochreiter, J. Schmidhuber, *Neural Comput* **1997**, *9*, 1735–1780.

[40] M. AlQuraishi, *BMC Bioinformatics* **2019**, *20*, 1–10.

[41] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, G. Monfardini, *IEEE Trans Neural Netw* **2009**, *20*, 61–80.

[42] P. W. Battaglia, J. B. Hamrick, V. Bapst, A. Sanchez-Gonzalez, V. Zambaldi, M. Malinowski, A. Tacchetti, D. Raposo, A. Santoro, R. Faulkner, C. Gulcehre, F. Song, A. Ballard, J. Gilmer, G. Dahl, A. Vaswani, K. Allen, C. Nash, V. Langston, C. Dyer, N. Heess, D. Wierstra, P. Kohli, M. Botvinick, O. Vinyals, Y. Li, R. Pascanu, *ArXiv* **2018**.

[43] V. Gligorijević, P. D. Renfrew, T. Kosciolek, J. K. Leman, D. Berenberg, T. Vatanen, C. Chandler, B. C. Taylor, I. M. Fisk, H. Vlamakis, R. J. Xavier, R. Knight, K. Cho, R. Bonneau, *Nature Communications 2021 12:1* **2021**, *12*, 1–14.

[44] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, *Adv Neural Inf Process Syst* **2017**, *2017-December*, 5999–6009.

[45] N. Ferruz, S. Schmidt, B. Höcker, *Nature Communications 2022 13:1* **2022**, *13*, 1–10.

[46] D. P. Kingma, M. Welling, *2nd International Conference on Learning Representations, ICLR 2014 - Conference Track Proceedings* **2013**.

[47] J. B. Ingraham, M. Baranov, Z. Costello, K. W. Barber, W. Wang, A. Ismail, V. Frappier, D. M. Lord, C. Ng-Thow-Hing, E. R. Van Vlack, S. Tie, V. Xue, S. C. Cowles, A. Leung, J. V. Rodrigues, C. L. Morales-Perez, A. M. Ayoub, R. Green, K. Puentes,

F. Oplinger, N. V. Panwar, F. Obermeyer, A. R. Root, A. L. Beam, F. J. Poelwijk, G. Grigoryan, *Nature 2023 623:7989* **2023**, *623*, 1070–1078.

[48] A. Trask, F. Hill, S. Reed, J. Rae, C. Dyer, P. Blunsom, *Adv Neural Inf Process Syst* **2018**, *2018-December*, 8035–8044.

[49] A. Kryshtafovych, T. Schwede, M. Topf, K. Fidelis, J. Moult, *Proteins: Structure, Function, and Bioinformatics* **2021**, *89*, 1607–1617.

[50] J. Pereira, A. J. Simpkin, M. D. Hartmann, D. J. Rigden, R. M. Keegan, A. N. Lupas, *Proteins: Structure, Function and Bioinformatics* **2021**, *89*, 1687–1699.

[51] M. Steinegger, J. Söding, *Nat Biotechnol* **2017**, *35*, 1026–1028.

[52] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, P. E. Bourne, *Nucleic Acids Res* **2000**, *28*, 235–242.

[53] R. Evans, M. O'Neill, A. Pritzel, N. Antropova, A. Senior, T. Green, A. Žídek, R. Bates, S. Blackwell, J. Yim, O. Ronneberger, S. Bodenstein, M. Zielinski, A. Bridgland, A. Potapenko, A. Cowie, K. Tunyasuvunakool, R. Jain, E. Clancy, P. Kohli, J. Jumper, D. Hassabis, *bioRxiv* **2022**, 2021.10.04.463034.

[54] S. M. Kandathil, A. M. Lau, D. T. Jones, *Curr Opin Struct Biol* **2023**, *81*, 102627.

[55] Z. Lin, H. Akin, R. Rao, B. Hie, Z. Zhu, W. Lu, N. Smetanin, R. Verkuil, O. Kabeli, Y. Shmueli, A. dos Santos Costa, M. Fazel-Zarandi, T. Sercu, S. Candido, A. Rives, *Science (1979)* **2023**, *379*, 1123–1130.

[56] R. Wu, F. Ding, R. Wang, R. Shen, X. Zhang, S. Luo, C. Su, Z. Wu, Q. Xie, B. Berger, J. Ma, J. Peng, *bioRxiv* **2022**, 2022.07.21.500999.

[57] M. AlQuraishi, *Curr Opin Chem Biol* **2021**, *65*, 1–8.

[58] N. Ferruz, B. Höcker, *Nature Machine Intelligence 2022 4:6* **2022**, *4*, 521–532.

[59] M. A. Pak, K. A. Markhieva, M. S. Novikova, D. S. Petrov, I. S. Vorobyev, E. S. Maksimova, F. A. Kondrashov, D. N. Ivankov, *PLoS One* **2023**, *18*, e0282689.

[60] T. C. Terwilliger, D. Liebschner, T. I. Croll, C. J. Williams, A. J. McCoy, B. K. Poon, P. V. Afonine, R. D. Oeffner, J. S. Richardson, R. J. Read, P. D. Adams, *Nat Methods* **2023**, 1–7.

[61] U. Abbas, J. Chen, Q. Shao, *bioRxiv* **2023**, 2023.05.23.542006.

[62] T. C. Terwilliger, D. Liebschner, T. I. Croll, C. J. Williams, A. J. McCoy, B. K. Poon, P. V. Afonine, R. D. Oeffner, J. S. Richardson, R. J. Read, P. D. Adams, *Nat Methods* **2023**, 1–7.

[63] D. Del Alamo, D. Sala, H. S. McHaourab, J. Meiler, *Elife* **2022**, *11*, DOI 10.7554/ELIFE.75751.

[64] D. Sala, P. W. Hildebrand, J. Meiler, *Front Mol Biosci* **2023**, *10*, 1121962.

[65] D. Sala, F. Engelberger, H. S. Mchaourab, J. Meiler, *Curr Opin Struct Biol* **2023**, *81*, 102645.

[66] A. Bateman, M. J. Martin, S. Orchard, M. Magrane, S. Ahmad, E. Alpi, E. H. Bowler-Barnett, R. Britto, H. Bye-A-Jee, A. Cukura, P. Denny, T. Dogan, T. G. Ebenezer, J. Fan, P. Garmiri, L. J. da Costa Gonzales, E. Hatton-Ellis, A. Hussein, A. Ignatchenko, G. Insana, R. Ishtiaq, V. Joshi, D. Jyothi, S. Kandasaamy, A. Lock, A. Luciani, M. Lugaric, J. Luo, Y. Lussi, A. MacDougall, F. Madeira, M. Mahmoudy, A. Mishra, K. Moulang, A. Nightingale, S. Pundir, G. Qi, S. Raj, P. Raposo, D. L. Rice, R. Saidi, R. Santos, E. Speretta, J. Stephenson, P. Totoo, E. Turner, N. Tyagi, P. Vasudev, K. Warner, X. Watkins, R. Zaru, H. Zellner, A. J. Bridge, L. Aimo, G. Argoud-Puy, A. H. Auchincloss, K. B. Axelsen, P. Bansal, D. Baratin, T. M. Batista Neto, M. C. Blatter, J. T. Bolleman, E. Boutet, L. Breuza, B. C. Gil, C. Casals-Casas, K. C. Echioukh, E. Coudert, B. Cuche, E. de Castro, A. Estreicher, M. L. Famiglietti, M. Feuermann, E. Gasteiger, P. Gaudet, S. Gehant, V. Gerritsen, A. Gos, N. Gruaz, C. Hulo, N. Hyka-Nouspikel, F. Jungo, A. Kerhornou, P. Le Mercier, D. Lieberherr, P. Masson, A. Morgat, V. Muthukrishnan, S. Paesano, I. Pedruzzi, S. Pilbout, L. Pourcel, S. Poux, M. Pozzato, M. Pruess, N. Redaschi, C. Rivoire, C. J. A. Sigrist, K. Sonesson, S. Sundaram, C. H. Wu, C. N. Arighi, L. Arminski, C. Chen, Y. Chen, H. Huang, K. Laiho, P. McGarvey, D. A. Natale, K. Ross, C. R. Vinayaka, Q. Wang, Y. Wang, J. Zhang, *Nucleic Acids Res* **2023**, *51*, D523–D531.

[67] E. W. Sayers, E. E. Bolton, J. R. Brister, K. Canese, J. Chan, D. C. Comeau, R. Connor, K. Funk, C. Kelly, S. Kim, T. Madej, A. Marchler-Bauer, C. Lanczycki, S. Lathrop, Z. Lu, F. Thibaud-Nissen, T. Murphy, L. Phan, Y. Skripchenko, T. Tse, J. Wang, R. Williams, B. W. Trawick, K. D. Pruitt, S. T. Sherry, *Nucleic Acids Res* **2022**, *50*, D20–D26.

[68] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, D. J. Lipman, *J Mol Biol* **1990**, *215*, 403–410.

[69] C. Camacho, G. Coulouris, V. Avagyan, N. Ma, J. Papadopoulos, K. Bealer, T. L. Madden, *BMC Bioinformatics* **2009**, *10*, 1–9.

[70] A. M. Schnoes, S. D. Brown, I. Dodevski, P. C. Babbitt, *PLoS Comput Biol* **2009**, *5*, e1000605.

[71] P. Vanacek, E. Sebestova, P. Babkova, S. Bidmanova, L. Daniel, P. Dvorak, V. Stepankova, R. Chaloupkova, J. Brezovsky, Z. Prokop, J. Damborsky, *ACS Catal* **2018**, *8*, 2402–2412.

[72] M. Varadi, S. Anyango, M. Deshpande, S. Nair, C. Natassia, G. Yordanova, D. Yuan, O. Stroe, G. Wood, A. Laydon, A. Zídek, T. Green, K. Tunyasuvunakool, S. Petersen, J. Jumper, E. Clancy, R. Green, A. Vora, M. Lutfi, M. Figurnov, A. Cowie, N. Hobbs, P. Kohli, G. Kleywegt, E. Birney, D. Hassabis, S. Velankar, *Nucleic Acids Res* **2022**, *50*, D439–D444.

[73] O. Trott, A. J. Olson, *J Comput Chem* **2010**, *31*, 455–461.

[74] M. L. Verdonk, J. C. Cole, M. J. Hartshorn, C. W. Murray, R. D. Taylor, *Proteins: Structure, Function, and Bioinformatics* **2003**, *52*, 609–623.

10.1002/cbic.202400092

[75] M. Buttenschoen, G. M. Morris, C. M. Deane, *Chem Sci* **2023**, DOI 10.1039/d3sc04185a.

[76] R. Zambrano, M. Jamroz, A. Szczasiuk, J. Pujols, S. Kmiecik, S. Ventura, *Nucleic Acids Res* **2015**, *43*, W306–W313.

[77] N. S. de Groot, V. Castillo, R. Graña-Montes, S. Ventura, in *LR Lloyd's Register*, **2012**, pp. 199–220.

[78] O. Conchillo-Solé, N. S. de Groot, F. X. Avilés, J. Vendrell, X. Daura, S. Ventura, *BMC Bioinformatics* **2007**, *8*, DOI 10.1186/1471-2105-8-65.

[79] A. Kuriata, A. M. Gierut, T. Oleniecki, M. P. Ciemny, A. Kolinski, M. Kurcinski, S. Kmiecik, *Nucleic Acids Res* **2018**, *46*, W338–W343.

[80] C. N. Magnan, A. Randall, P. Baldi, *Bioinformatics* **2009**, *25*, 2200–2207.

[81] P. Smialowski, G. Doose, P. Torkler, S. Kaufmann, D. Frishman, *FEBS Journal* **2012**, *279*, 2192–2200.

[82] J. Delgado, L. G. Radusky, D. Cianferoni, L. Serrano, *Bioinformatics* **2019**, *35*, 4168–4169.

[83] M. Gil-Garcia, M. Banó-Polo, N. Varejao, M. Jamroz, A. Kuriata, M. Díaz-Caballero, J. Lascorz, B. Morel, S. Navarro, D. Reverter, S. Kmiecik, S. Ventura, *Mol Pharm* **2018**, *15*, 3846–3859.

[84] J. Shin, S. Raissi, P. Phelan, P. A. Bullock, *Protein Expr Purif* **2023**, *202*, 106196.

[85] A. Minich, J. Šarkanová, Z. Levarski, S. Stuchlík, *World J Microbiol Biotechnol* **2022**, *38*, 1–12.

[86] E. Parladé, E. Voltà-Durán, O. Cano-Garrido, J. M. Sánchez, U. Unzueta, H. López-Laguna, N. Serna, M. Cano, M. Rodríguez-Mariscal, E. Vazquez, A. Villaverde, *Int J Mol Sci* **2022**, *23*, DOI 10.3390/ijms23094958.

[87] C. Pintado-Grima, O. Bárcenas, A. Bartolomé-Nafría, M. Fornt-Suñé, V. Iglesias, J. Garcia-Pardo, S. Ventura, *Biophysica* **2023**, *3*, 1–20.

[88] A. M. Fernandez-Escamilla, F. Rousseau, J. Schymkowitz, L. Serrano, *Nat Biotechnol* **2004**, *22*, 1302–1306.

[89] A. Ganesan, A. Siekierska, J. Beerten, M. Brams, J. Van Durme, G. De Baets, R. Van Der Kant, R. Gallardo, M. Ramakers, T. Langenberg, H. Wilkinson, F. De Smet, C. Ulens, F. Rousseau, J. Schymkowitz, *Nat Commun* **2016**, *7*, DOI 10.1038/ncomms10816.

[90] R. van der Kant, A. R. Karow-Zwick, J. Van Durme, M. Blech, R. Gallardo, D. Seeliger, K. Aßfalg, P. Baatsen, G. Compernolle, A. Gils, J. M. Studts, P. Schulz, P. Garidel, J. Schymkowitz, F. Rousseau, *J Mol Biol* **2017**, *429*, 1244–1261.

[91] H. M. Berman, M. J. Gabanyi, A. Kouranov, D. I. Micallef, J. Westbrook, Protein Structure Initiative network of Investigators, **2017**, DOI 10.5281/zenodo.821653.

[92] B. K. Bhandari, P. P. Gardner, C. S. Lim, *Bioinformatics* **2020**, *36*, 4691–4698.

[93] P. Sormanni, F. A. Aprile, M. Vendruscolo, *J Mol Biol* **2015**, *427*, 478–490.

[94] S. Hirose, T. Noguchi, *Proteomics* **2013**, *13*, 1444–1456.

[95] D. L. Wilkinson, R. G. Harrison, *Nat Biotechnol* **1991**, *9*, 443–448.

[96] S. Khurana, R. Rawi, K. Kunji, G. Y. Chuang, H. Bensmail, R. Mall, *Bioinformatics* **2018**, *34*, 2605–2613.

[97] M. Hebditch, M. A. Carballo-Amador, S. Charonis, R. Curtis, J. Warwicker, *Bioinformatics* **2017**, *33*, 3098–3100.

[98] D. Raimondi, G. Orlando, P. Fariselli, Y. Moreau, *PLoS Comput Biol* **2020**, *16*, 1–15.

[99] F. Agostini, D. Cirillo, C. M. aria Livi, R. Delli Ponti, G. G. aetano Tartaglia, *Bioinformatics* **2014**, *30*, 2975–2977.

[100] A. A. Diaz, E. Tomba, R. Lennarson, R. Richard, M. J. Bagajewicz, R. G. Harrison, *Biotechnol Bioeng* **2010**, *105*, 374–383.

[101] A. Waterhouse, M. Bertoni, S. Bienert, G. Studer, G. Tauriello, R. Gumienny, F. T. Heer, T. A. P. De Beer, C. Rempfer, L. Bordoli, R. Lepore, T. Schwede, *Nucleic Acids Res* **2018**, *46*, W296–W303.

[102] L. Foit, G. J. Morgan, M. J. Kern, L. R. Steimer, A. A. von Hacht, J. Titchmarsh, S. L. Warriner, S. E. Radford, J. C. A. Bardwell, *Mol Cell* **2009**, *36*, 861–871.

[103] D. A. Parsell, R. T. Sauer, *Journal of Biological Chemistry* **1989**, *264*, 7590–7595.

[104] J. Wada, H. Miyazaki, R. Kamada, K. Sakaguchi, *Chem Lett* **2016**, *45*, 185–187.

[105] V. Sieber, A. Plückthun, F. X. Schmid, *Nat Biotechnol* **1998**, *16*, 955–960.

[106] W. S. Kwon, N. A. Da Silva, J. T. Kellis, *Protein Eng* **1996**, *9*, 1197–1202.

[107] H. Zhao, F. H. Arnold, *Protein Engineering, Design and Selection* **1999**, *12*, 47–53.

[108] M. T. Reetz, *Angewandte Chemie International Edition* **2013**, *52*, 2658–2666.

[109] S. Bershtein, M. Segal, R. Bekerman, N. Tokuriki, D. S. Tawfik, *Nature 2006 444:7121* **2006**, *444*, 929–932.

[110] C. M. Miton, N. Tokuriki, *Protein Science* **2016**, *25*, 1260–1272.

[111] X. F. Cadet, J. C. Gelly, A. van Noord, F. Cadet, C. G. Acevedo-Rocha, *Methods in Molecular Biology* **2022**, *2461*, 225–275.

[112] M. Musil, J. Stourac, J. Bendl, J. Brezovsky, Z. Prokop, J. Zendulka, T. Martinek, D. Bednar, J. Damborsky, *Nucleic Acids Res* **2017**, *45*, W393–W399.

[113] B. E. Suzek, Y. Wang, H. Huang, P. B. McGarvey, C. H. Wu, *Bioinformatics* **2015**, *31*, 926–932.

[114] F. Sievers, A. Wilm, D. Dineen, T. J. Gibson, K. Karplus, W. Li, R. Lopez, H. McWilliam, M. Remmert, J. Söding, J. D. Thompson, D. G. Higgins, *Mol Syst Biol* **2011**, *7*, 1–6.

[115] A. Leaver-Fay, M. Tyka, S. M. Lewis, O. F. Lange, J. Thompson, R. Jacak, K. W. Kaufman, P. D. Renfrew, C. A. Smith, W. Sheffler, I. W. Davis, S. Cooper, A. Treuille, D. J. Mandell, F. Richter, Y.-E. A. Ban, S. J. Fleishman, J. E. Corn, D. E. Kim, S. Lyskov, M. Berrondo, S. Mentzer, Z. Popović, J. J. Havranek, J. Karanicolas, R. Das, J. Meiler, T. Kortemme, J. J. Gray, B. Kuhlman, D. Baker, P. Bradley, in *Bone*, **2011**, pp. 545–574.

[116] J. Bendl, J. Stourac, E. Sebestova, O. Vavra, M. Musil, J. Brezovsky, J. Damborsky, *Nucleic Acids Res* **2016**, *44*, W479–W487.

[117] R. J. Floor, H. J. Wijma, D. I. Colpa, A. Ramos-Silva, P. A. Jekel, W. Szymański, B. L. Feringa, S. J. Marrink, D. B. Janssen, *ChemBioChem* **2014**, *15*, 1660–1672.

[118] M. Musil, A. Jezik, J. Horackova, S. Borko, P. Kabourek, J. Damborsky, D. Bednar, *Brief Bioinform* **2024**, *25*, 1–10.

[119] G. Studer, G. Tauriello, S. Bienert, M. Biasini, N. Johner, T. Schwede, *PLoS Comput Biol* **2021**, *17*, 1–18.

[120] S. F. Altschul, E. M. Gertz, R. Agarwala, A. A. Schäffer, Y. K. Yu, *Nucleic Acids Res* **2009**, *37*, 815–824.

[121] Y. Peleg, R. Vincentelli, B. M. Collins, K. E. Chen, E. K. Livingstone, S. Weeratunga, N. Leneva, Q. Guo, K. Remans, K. Perez, G. E. K. Bjerga, Ø. Larsen, O. Vaněk, O. Skořepa, S. Jacquemin, A. Poterszman, S. Kjær, E. Christodoulou, S. Albeck, O. Dym, E. Ainbinder, T. Unger, A. Schuetz, S. Matthes, M. Bader, A. de Marco, P. Storici, M. S. Semrau, P. Stolt-Bergner, C. Aigner, S. Suppmann, A. Goldenzweig, S. J. Fleishman, *J Mol Biol* **2021**, *433*, DOI 10.1016/j.jmb.2021.166964.

[122] A. Kunka, S. M. Marques, M. Havlasek, M. Vasina, N. Velatova, L. Cengelova, D. Kovar, J. Damborsky, M. Marek, D. Bednar, Z. Prokop, *ACS Catal* **2023**, 12506–12518.

[123] M. J. H. K. N. Orlandi, S. R. Phillips, Z. R. Sailer, J. L. Harman, “Topiary Colab,” can be found under https://colab.research.google.com/github/harmslab/topiary-examples/blob/main/notebooks/seed-to-alignment.ipynb, **2023**.

[124] H. Lu, D. J. Diaz, N. J. Czarnecki, C. Zhu, W. Kim, R. Shroff, D. J. Acosta, B. R. Alexander, H. O. Cole, Y. Zhang, N. A. Lynd, A. D. Ellington, H. S. Alper, *Nature* **2022**, *604*, 662–667.

[125] I. Paik, P. H. T. Ngo, R. Shroff, D. J. Diaz, A. C. Maranhao, D. J. F. Walker, S. Bhadra, A. D. Ellington, *Biochemistry* **2023**, *62*, 410–418.

[126] A. V. Kulikova, D. J. Diaz, J. M. Loy, A. D. Ellington, C. O. Wilke, *J Biol Phys* **2021**, *47*, 435–454.

[127] R. Nikam, A. Kulandaisamy, K. Harini, D. Sharma, M. Michael Gromiha, *Nucleic Acids Res* **2021**, *49*, D420–D424.

[128] B. Tidor, M. Karplus, *Proteins: Structure, Function, and Bioinformatics* **1993**, *15*, 71–79.

[129] C. N. Pace, G. R. Grimsley, J. A. Thomson, B. J. Barnett, *Journal of Biological Chemistry* **1988**, *263*, 11820–11825.

[130] A. A. Dombkowski, K. Z. Sultana, D. B. Craig, *FEBS Lett* **2014**, *588*, 206–212.

[131] D. A. Suplatov, K. E. Kopylov, N. N. Popova, V. V. Voevodin, V. K. Švedas, *Bioinformatics* **2018**, *34*, 1583–1585.

[132] L. Wieteska, M. Ionov, J. Szemraj, C. Feller, A. Kolinski, D. Gront, *J Biotechnol* **2015**, *199*, 69–76.

[133] T. P. Korman, B. Sahachartsiri, D. M. Charbonneau, G. L. Huang, M. Beauregard, J. U. Bowie, *Biotechnol Biofuels* **2013**, *6*, 1–13.

[134] D. Saerens, K. Conrath, J. Govaert, S. Muyldermans, *J Mol Biol* **2008**, *377*, 478–488.

[135] H. Takagi, T. Takahashi, H. Momose, M. Inouye, Y. Maeda, H. Matsuzawa, T. Ohta, *Journal of Biological Chemistry* **1990**, *265*, 6874–6878.

[136] V. A. Risso, J. A. Gavira, J. M. Sanchez-Ruiz, *Environ Microbiol* **2014**, *16*, 1485–1489.

[137] V. A. Risso, J. A. Gavira, D. F. Mejia-Carmona, E. A. Gaucher, J. M. Sanchez-Ruiz, *J Am Chem Soc* **2013**, *135*, 2899–2902.

[138] P. Babkova, E. Sebestova, J. Brezovsky, R. Chaloupkova, J. Damborsky, *ChemBioChem* **2017**, *18*, 1448–1456.

[139] L. C. Wheeler, S. A. Lim, S. Marqusee, M. J. Harms, *Curr Opin Struct Biol* **2016**, *38*, 37–43.

[140] R. Merkl, R. Sterner, *Biol Chem* **2016**, *397*, 1–21.

[141] R. E. S. Thomson, S. E. Carrera-Pacheco, E. M. J. Gillam, *Journal of Biological Chemistry* **2022**, *298*, 102435.

[142] R. T. Khan, M. Musil, J. Stourac, J. Damborsky, D. Bednar, *Curr Protoc* **2021**, *1*, 1–13.

[143] K. N. Orlandi, S. R. Phillips, Z. R. Sailer, J. L. Harman, M. J. Harms, *Protein Science* **2023**, *32*, 1–19.

[144] S. Bershtein, M. Segal, R. Bekerman, N. Tokuriki, D. S. Tawfik, *Nature* **2006**, *444*, 929–932.

[145] J. D. Bloom, S. T. Labthavikul, C. R. Otey, F. H. Arnold, *Proc Natl Acad Sci U S A* **2006**, *103*, 5869–5874.

[146] X. Wang, G. Minasov, B. K. Shoichet, *J Mol Biol* **2002**, *320*, 85–95.

[147] M. Goldsmith, N. Aggarwal, Y. Ashani, H. Jubran, P. Greisen, S. Ovchinnikov, H. Leader, D. Baker, J. L. Sussman, A. Goldenzweig, S. J. Fleishman, D. S. Tawfik, *Protein Engineering, Design and Selection* **2017**, *30*, 333–345.

[148] Fleishman Lab, “Peer-reviewed papers that use PROSS designs,” can be found under https://docs.google.com/document/d/1Z7PINtB4-Eoz61DHKhodLUVZ-rwGoSSYi53Tub27WkU/edit, **2024**.

[149] D. Listov, R. Lipsh-Sokolik, S. Rosset, C. Yang, B. E. Correia, S. J. Fleishman, *Protein Science* **2022**, *31*, e4400.

[150] G. Lapidoth, O. Khersonsky, R. Lipsh, O. Dym, S. Albeck, S. Rogotner, S. J. Fleishman, *Nature Communications 2018 9:1* **2018**, *9*, 1–9.

[151] C. Yang, F. Sesterhenn, J. Bonet, E. A. van Aalen, L. Scheller, L. A. Abriata, J. T. Cramer, X. Wen, S. Rosset, S. Georgeon, T. Jardetzky, T. Krey, M. Fussenegger, M. Merkx, B. E. Correia, *Nature Chemical Biology 2021 17:4* **2021**, *17*, 492–500.

[152] X. Pan, T. Kortemme, *Journal of Biological Chemistry* **2021**, *296*, 100558.

[153] H. Khakzad, I. Igashov, A. Schneuing, C. Goverde, M. Bronstein, B. Correia, *Cell Syst* **2023**, *14*, 925–939.

[154] A. Nikolaev, A. Kuzmin, E. Markeeva, E. Kuznetsova, O. Semenov, A. Anuchina, A. Remeeva, I. Gushchin, *bioRxiv* **2023**, 2023.08.25.554855.

[155] H. W. Kao, W. L. Lu, M. R. Ho, Y. F. Lin, Y. J. Hsieh, T. P. Ko, S. Te Danny Hsu, K. P. Wu, *ACS Synth Biol* **2023**, *12*, 2310–2319.

[156] K. H. Sumida, R. Núñnúñ Ez-Franco, I. Kalvet, S. J. Pellock, B. I. M. Wicky, L. F. Milles, J. Dauparas, J. Wang, Y. Kipnis, N. Jameson, A. Kang, J. De, L. Cruz, B. Sankaran, A. K. Bera, G. Jiménez-Osés, D. Baker, *J Am Chem Soc* **2024**, *146*, DOI 10.1021/JACS.3C10941.

[157] C. A. Goverde, M. Pacesa, N. Goldbach, L. J. Dornfeld, P. E. M. Balbi, S. Georgeon, S. Rosset, S. Kapoor, J. Choudhury, J. Dauparas, C. Schellhaas, S. Kozlov, D. Baker, S. Ovchinnikov, A. J. Vecchio, B. E. Correia, *bioRxiv* **2024**, 2023.05.09.540044.

[158] J. Dauparas, G. R. Lee, R. Pecoraro, L. An, I. Anishchenko, C. Glasscock, D. Baker, *bioRxiv* **2023**, 2023.12.22.573103.

[159] T. Sanderson, M. L. Bileschi, D. Belanger, L. J. Colwell, *Elife* **2023**, *12*, DOI 10.7554/ELIFE.80942.

[160] J. R. Jeliazkov, D. del Alamo, J. D. Karpiak, *bioRxiv* **2023**, 2023.05.23.541774.

[161] J. Yim, B. L. Trippe, V. De Bortoli, E. Mathieu, A. Doucet, R. Barzilay, T. Jaakkola, *Proc Mach Learn Res* **2023**, *202*, 40001–40039.

[162] J. Yim, A. Campbell, A. Y. K. Foong, M. Gastegger, J. Jiménez-Luna, S. Lewis, V. G. Satorras, B. S. Veeling, R. Barzilay, T. Jaakkola, F. Noé, *ArXiv* **2023**, DOI https://doi.org/10.48550/arXiv.2310.05297.

[163] J. Wang, S. Lisanza, D. Juergens, D. Tischer, J. L. Watson, K. M. Castro, R. Ragotte, A. Saragovi, L. F. Milles, M. Baek, I. Anishchenko, W. Yang, D. R. Hicks, M. Expòsit, T. Schlichthaerle, J. H. Chun, J. Dauparas, N. Bennett, B. I. M. Wicky, A. Muenks, F. DiMaio, B. Correia, S. Ovchinnikov, D. Baker, *Science (1979)* **2022**, *377*, 387–394.

[164] C. Frank, A. Khoshouei, Y. de Stigter, D. Schiewitz, S. Feng, S. Ovchinnikov, H. Dietz, *bioRxiv* **2023**, 2023.02.24.529906.

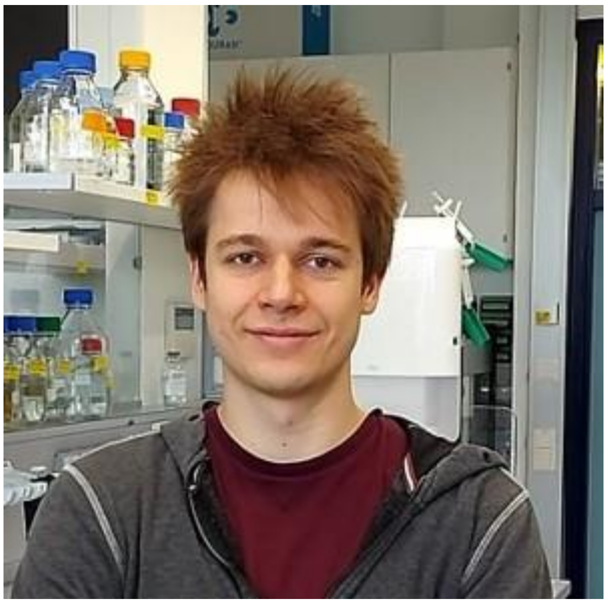
[165] J. P. Roney, S. Ovchinnikov, *Phys Rev Lett* **2022**, *129*, 238101.

[166] J. Vig, A. Madani, L. R. Varshney, C. Xiong, R. Socher, N. F. Rajani, in *ICLR 2021 - 9th International Conference on Learning Representations*, **2021**.

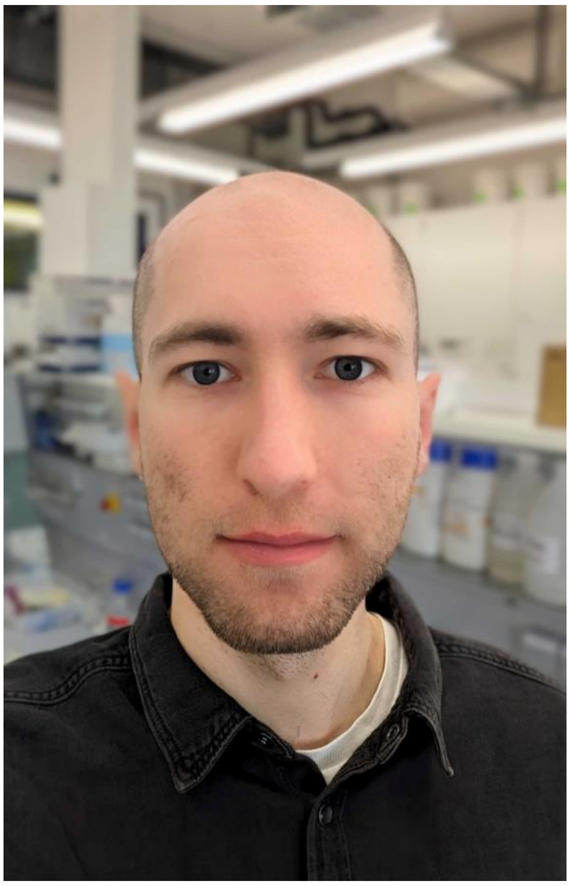
Biographical Information:

Horst Lechner is a PostDoc and "BioTechMed Graz YoungResearcherGroup" leader at the Graz University of Technology (Austria) interested in (re)designing enzymes and protein/ligand interactions. He *studied chemistry and biochemistry at the University of Graz (Austria) where he received his PhD in biocatalysis in 2015 in the group of Prof. Kroutil. Subsequently he moved to Prof. Höcker's laboratory (University of Bayreuth, Germany) as postdoctoral fellow before moving back to Graz.*

Markus Braun studied chemistry and biochemistry at Graz University of Technology where he is currently a PhD student in the lab of Gustav Oberdorfer. His PhD is centered on the design *of enzymes by a combination of recent protein design algorithms. In addition to enzyme design, he contributes his knowledge in multiple collaborations with other academic labs engineering natural enzymes and biologically active proteins.*



Adrian Tripp studied chemistry and biochemistry at Graz University of Technology, where he is currently working on his PhD project in the group of Gustav Oberdorfer at the Institute of Biochemistry. He is interested in computational de novo design involving enzymes and small molecule binders, with a focus on enzymes employing metal ligands for catalysis.



Florian Wieser is a fourth-year PhD student at the Institute of Biochemistry at Graz University of Technology. He earned his Master's degree in Biomedical Engineering, with a specialization in Bioinformatics and Molecular Bioengineering, from the same university. Currently, he is a member of the Oberdorfer lab, dedicating his research efforts to developing computational methods in the field of Protein Structure Prediction and Protein Design. His academic interests are diverse, ranging from molecular dynamics simulations to the cutting-edge field of Deep Learning.

10.1002/cbic.202400092

Gustav Oberdorfer obtained his PhD in biochemistry and molecular biology from the University of Graz in 2011. He then moved to the University of Washington as a postdoctoral fellow in 2012. In 2020 he joined the faculty of the Institute of Biochemistry at the Graz University of Technology as Assistant Professor. The primary research interests of his group revolve around enzymology and the computational design of novel de novo ligand binding and catalytically active proteins.

10.1002/cbic.202400092