Curse and Blessing of Non-proteinogenic Parts in Computational Enzyme Engineering

Korbeld, Kerlen Theo; Fürst, Maximilian Josef Ludwig Johannes

Published in:
ChemBioChem

DOI:
10.1002/cbic.202300192

Publication date:
2023

Review
doi.org/10.1002/cbic.202300192

Chemistry Europe
European Chemical Societies Publishing

Curse and Blessing of Non-Proteinogenic Parts in Computational Enzyme Engineering

Kerlen T. Korbeld[a] and Maximilian J. L. J. Fürst*[a]

Enzyme engineering aims to improve or install a new function in biocatalysts for applications ranging from chemical synthesis to biomedicine. For decades, computational techniques have been developed to predict the effect of protein changes and design new enzymes. However, these techniques may have been optimized to deal with proteins composed of the standard amino acid alphabet, while the function of many enzymes relies on non-proteogenic parts like cofactors, nucleic acids, and post-translational modifications. Enzyme systems containing such molecules might be handled or modeled improperly by computational tools, and thus be unsuitable, or require additional tweaking, parameterization, or preparation. In this review, we give an overview of common and recent tools and workflows available to computational enzyme engineers. We highlight the various pitfalls that come with including non-proteogenic compounds in computations and outline potential ways to address common issues. Finally, we showcase successful examples from the literature that computationally engineered such enzymes.

1. Introduction

Protein engineering is a widely applied technology within many fields of the biological sciences. From its origins as an analytical method used to establish the role of an amino acid or protein part by destruction, protein engineering has matured into an independent research area. It is now commonplace to deliberately alter proteins to establish novel functions, and the field thus overlaps with and shares a design-driven approach with biotechnology and synthetic biology. As protein function is determined by structure, protein engineering has also always been closely aligned with structural biology and (what some already consider its successor) computational biology. A distinction is usually made between "blind" directed evolution – low-success-rate random mutagenesis in conjunction with (high-throughput) functional selection assays – and rational design – the structure- and often computer-aided alteration of residues identified to be of high functional relevance.[1] Although the literature is abundant with examples of entirely blind protein engineering, in this current golden age of protein structure prediction[2] even the most fervent proponents of directed evolution might not refuse to take a structure into account.[3] While modern machine learning (ML) algorithms enabled AlphaFold[4] and its offspring, the underlying training data represent the decades of progress in diverse experimental fields and their coevolved theoretical "wingmen": genomics and bioinformatics, biochemistry and systems biology, as well as biophysics and molecular modeling (sometimes synonymously referred to as "computational biology"). Although the advances in computational approaches have truly been astonishing, even the recent leap toward solving the protein structure problem is only one step on the path to the actual holy grail: the prediction of protein function. While function exerted through protein-protein interaction might come within reach via accurate protein complex predictions, we are still relatively far from predicting the activity of proteins with catalytic activity, which account for approximately half of the protein universe (Figure 1).[5]

Enzymes are powerful and sustainable catalysts, which are applied for example in the pharmaceutical or chemical industry and are at the core of biotech and synthetic biology technologies.[6] One of the main advantages of enzymes over traditional catalysts is their genetic encoding-derived programmability, which facilitates easy engineering.

Judging by the most advanced tools currently at our hands, the availability of a protein structure is not enough to confidently predict the activity of a given enzyme variant. Rather, enzyme engineers also often require accurate predictions and knowledge of protein dynamics or at least conformational ensembles;[7–8] the structural effects of mutations; the binding modes of small-molecule ligands; and the precise molecular mechanisms of catalysis.[9] Unsurprisingly, the catalogue of *in silico* tools aimed at achieving these goals is continuously expanded and improved, as theoretical insights accumulate, and computational capabilities increase. However, we found and outline here that a common pitfall of many techniques is that, out of the box, they only work (reliably) with proteins containing the proteinogenic twenty amino acids (Table 1). Although they are the most important molecules that build up proteins, more than two thirds of all enzymes rely on additional building blocks. Non-covalently, enzymes may stably or transiently bind cofactors, metals, or nucleic acids (Figure 1),

[a] *K. T. Korbeld, Dr. M. J. L. J. Fürst*
Molecular Enzymology Group
University of Groningen
Nijenborgh 4, Groningen 9747AG (The Netherlands)
E-mail: m.j.l.j.furst@rug.nl

This article is part of the Special Collection ChemBioTalents2022. Please see our homepage for more articles in the collection.

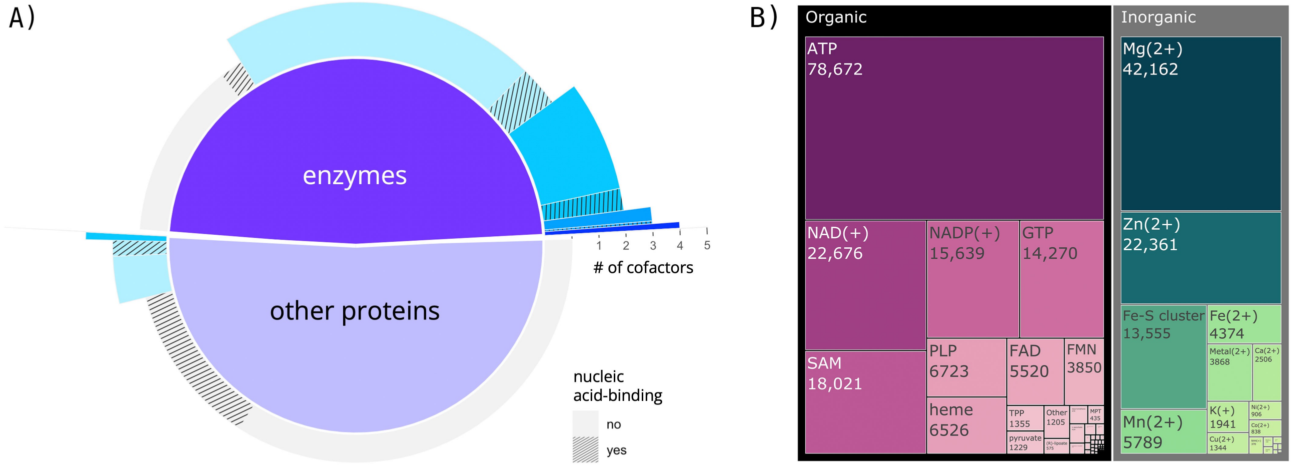

Figure 1. Analysis of the Uniprot database and extracted information about deposited proteins with respect to their catalytic activity and dependency on cofactors and other auxiliary non-proteinogenic molecules, visualized as a mixed pie and doughnut chart. A) Overview of the fractions of proteins that are annotated to have catalytic activity (inner pie chart) and to contain cofactors (second layer/doughnut chart part). The height and color of the doughnut chart piece representing a given protein fraction indicate the number of cofactors, while the fraction of nucleic acid-binding annotated proteins is shaded with stripes. B) Treemap visualizing the distribution of the most common cofactors sub-grouped into organic (shades of purple) and inorganic (shades of green) cofactors. Code and data for reproducing the figures can be found at https://github.com/kt-korbeld/cofactor-figure-data.

and the interaction with these helper molecules is frequently essential for their catalytic mode of action. In addition, naturally occurring enzymes can form covalent intermediates with their substrates, are subject to the over 500 known post-translational modifications (PTMs, e.g. glycosylations, methylations, or phosphorylations),[10] or contain non-canonical amino acids such as selenocysteine and pyrrolysine.[11] Finally, researchers have also expanded the amino acid alphabet with hundreds[12] of non-canonical amino acids (ncAAs),[13] as strategies for incorporation become simpler, more reliable, and directable to multiple sites.[13–15] All these cases represent computational challenges, in particular for molecular modeling. The solutions to overcome them depend on the given task, as well as the diverse roles played by these parts, which we here collectively termed 'non-proteinogenic'.

In this review, we aim to outline the breadth of these challenges, their relevance for common computational tasks, and the variety of approaches that computational biologists have taken to tackle them. Because our objective is to provide guidance to enzyme engineers, we emphasize the computational approaches aimed at manipulating the catalytic functions of existing enzymes, and, in general, excluded *de novo* design (reviewed elsewhere[16–18]). We also do not review literature on the related problem of drug discovery,[19–20] although several of our raised points will apply in that field, too. We will discuss computational tools roughly in the order they occur in many sophisticated computational design workflows (Figure 2), although any given pipeline may diverge from this order, or skip some or even most other steps in this very general scheme. For nearly all methods, a protein sequence is both input (e.g. a wild type) as well as output (the predicted mutant), and in practice, many workflows iterate over cycles of prediction, test, and method adjustment. The in the following discussed pipelines may be purely sequence-based, take the protein structure into account, or model the relevant enzyme-ligand complex for computations (Figure 2). Models and simulations usually require a force field, whose parameters are derived from quantum mechanical (QM) calculations. Predicted mutants are usually selected after an appropriate ranking step, and experimental high-throughput screens can serve as input for machine learning (ML) based methods. Rather than comprehensively listing all available methods, we focus on presenting the most commonly used tools, discuss pipelines that are representative of a given design strategy, or refer to other reviews.

Kerlen Korbeld got his master's degree in Biomolecular Sciences at the University of Groningen and joined the group of Maximilian Fürst as a Ph.D. student in 2022. His research is focused on the integration of bioinformatics computational modeling for protein design.

Maximilian Fürst is an Assistant Professor at the University of Groningen. He studied Biology at the LMU Munich and obtained a Ph.D. in biochemistry under the supervision of Prof. Marco Fraaije in Groningen. For his postdoctoral work, Max moved to Cambridge to work with Dr. Phil Holliger at the MRC LMB. Max's research is focused on protein engineering, and his group aims to combine computational protein design with high-throughput screening platforms.

Table 1. Computational tools for enzyme engineering are mentioned in this review. Each tool is coloured **green** if works readily with (at least one type of) non-proteogenic parts, **orange** if it can work but requires some additional consideration, and **red** if it works poorly or ignores non-proteogenic parts altogether.

Category	Method	Implementation	Ref.
Structure preparation	Structure prediction	AlphaFold2, RoseTTAfold, OpenFold, ESMFold, OmegaFold, RoseTTA-foldNA	[4,43,86,191–193]
	Binding site prediction	ConSurf, PrankWeb, ScanNet, Fpocket, CAVITY, Bsite, Alphafill, NodeCoder, BioMetAll	[40,61–64,67–69,72]
	Molecular modeling software	YASARA, Gromacs, GaudiMM, Gaussian	[37,38,90,139]
	Specialized docking	Haddock, FlexPepDock, Flex-LzerD, LightDock, MELD-DNA, EADock, MpSDock, GOLD	[81,83,85,87,88,194]
Designing catalytic properties	Remodeling the active site	RosettaDesign, Rosetta Enzyme Design, Triad	[195–197]
	Generating mutant libraries	CASCO, FuncLib, HotSpotWizard, DynaComm.py	[198–201]
	Designing Access tunnel	PELE, CAVER, Caverdock, Gpathfinder, MoMa-LigPath, ART-RRT	[202–207]
	Designing metal-binding sites	MetalSearch, HostDesigner, Optgraft	[208–210]
	Machine learning methods	MutCompute, Rossman-toolbox, Cofactory	[211,212,229]
Designing thermostability	Energy functions	Foldx, Rosetta ddg_monomer, CUPSAT, pSTAB, Proteus, ABACUS	[213–218]
	Pipelines using energy functions	Fireprot, HotSpot Wizard, PROSS, FRESCO, GRAPE	[219–223]
	Machine learning methods	mCSM, BayeStab, PoPMuSiC-2.0, Prethermut, i-Mutant 2.0/3.0, DeepDDG, ABACUS-R, MutCompute	[211,224–230]
	Integrated Machine learning methods	iDeepDDG, iStable, Dynamut, DUET	[229,231,232]
	Other methods	SCHEMA, SDM, B-FIT	[233–235]
Force fields	Coarse-Grain MD	MARTINI, oxDNA, oxRNA, AWSEM	[98,99,100,106]
	All-atom MD	AMBER, CHARMM, OPLS-AA, GROMOS	[120–123]
	Polarizable MD	CHARMM Drude, AMOEBA	[135,136]
Obtaining parameters	Automated parameterization	Acpype, CHARMMing, PrATB, LigParGen, AUTOSMILES	[37,149,152,236,237]
	Force field refinements for nucleic acids	Parmbcs0/1, Bcs0/1	[154–156]
	Parameterizing metals	EZAFF, MCPB, VFFDT	[181,185,186]

2. Methods in Computational Enzyme Engineering

2.1. Sequence and static structure-based methods

Any computational protein engineering effort requires basic information about the target. Typically, the primary sequence of a protein is the minimum. Because enzymatic activity can in some cases be influenced by the nucleotide sequence,[21] the today commonly performed procedure of codon optimizing genes[22] for recombinant expression can be considered the first step in computational enzyme engineering. Typically, however, a protein's primary amino acid sequence serves as the starting point to many *in silico* design campaigns. Until 2021 (i.e. before AlphaFold2), amino acid sequence alone was usually not sufficient for reliably deducing a protein's dependence on PTMs,[23] cofactors, or other non-proteinogenic auxiliaries. The sequence similarity to structurally-characterized homologs or the presence of distinct fingerprints could, however, lead to varying degrees of certainty,[24–25] in particular for cofactor[26] and nucleic acid[27] binding. Notably, such sequence-based prediction methods also included early implementations of ML algorithms.[28–29] Tools that use sequence as the sole input do not usually need to be adapted for enzymes with non-proteinogenic parts, because the circumvention of structure means that the processing and the output compute them implicitly. On the other hand, such tools usually do not support the input of modified or non-canonical amino acids (ncAAs) either, perhaps because such an ability is too niche to justify development. Therefore, the protein sequence serves in the majority of cases solely as the input for structure determination, and most mutant prediction workflows actually require a protein structure. Thus, until recently, these approaches depended on the availability of an experimental crystal, NMR, or cryo-EM structure, as homology models were usually considered too inaccurate, except for high-similarity targets.[30] When available, experimental structures always require a preparatory step to address the many issues found in protein data bank entries that prevent realistic *in silico* modeling, such as absence of hydrogens, protonation states, and bond orders, or alternate configurations of or absent side-chains.[31–34] Since these issues are well known to computational biologists, dedicated tools to fix them are available,[35–36] and common modeling software packages often contain built-in features to address them automatically.[37–38] A well-maintained resource for automatically refined and validated protein crystal structures is PDB-REDO,[39] whose atomic models may be significantly improved compared to the original PDB entry. Nevertheless, no foolproof way can currently correct all errors, and it is advisable to manually inspect important regions of original data (e.g. electron density maps) and automatic fixes before structure-based calculations.

The now ubiquitous availability of models with near-experimental accuracy thanks to artificial intelligence (AI)

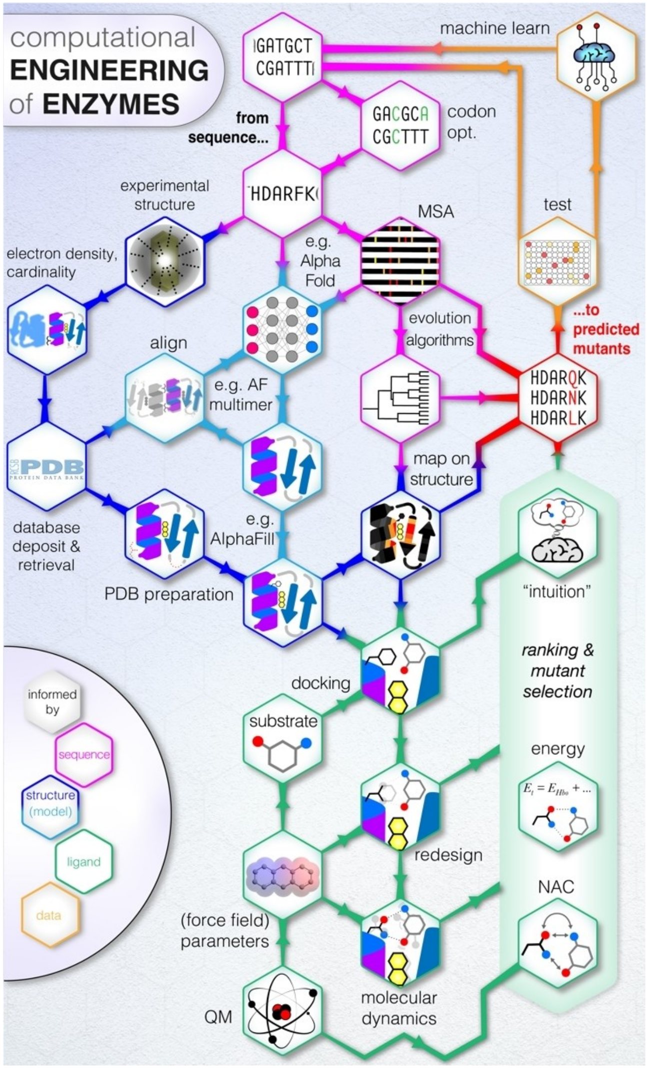

Figure 2. General workflow in computational enzyme engineering. Although any given strategy may deviate from the here presented order and may use only some of these elements, the scheme intents to depict a consensus of computational workflows that aim to predict enzyme variants with improved properties.

algorithms means nothing less than a revolution to the field: in the seldom cases where existing computational tools required PDB IDs as input, the pipelines were (or can be expected to be) updated to incorporate automatic retrieval or generation of such models. This sudden abundance of structural information already reverberates through the field and might make rational protein engineering based solely on sequence information essentially obsolete or at least rare. A promising approach lies, however, in combining both sequence- and structure-based methods. An example is ConSurf,[40] a database and associated web server for evolutionary analysis of an input sequence.[41] The tool natively supports mapping the output on (predicted) structures, and it has also been adapted to identify ligand and cofactor pockets.[42]

Since the (quasi simultaneous) release of AlphaFold2[4] and RoseTTAFold[43] a plethora of other AI structure prediction tools has been released.[44] Given the rapid and ongoing developments in the field, we refrain from attempting a comprehensive list here and recommend the reader to consult a recent review article (at the time of writing, e.g.).[44–45] For computational enzyme engineering, the decision on which prediction tool to use will often be influenced by whether speed or accuracy is the more important factor. While it is relatively simple to estimate or test the former, evaluating the latter is significantly more challenging and thus a community effort.[46]

Some of the currently popular AI structure prediction tools create valuable, sequence-related output. For example, the multiple sequence alignment (MSA) generated by AlphaFold can be manipulated to reveal alternative protein conformations.[47–50] Additional information can also be extracted from the per-residue uncertainty factor called the predicted local distance difference test (pLDDT), which was shown to correlate with protein dynamics.[51–52] While confident structure prediction enabled a plethora of opportunities in protein design, such as the *de novo* design of backbones from random sequences via Monte Carlo "hallucination"[53] or diffusion models,[54] the reliable prediction of the structure of point mutations, lamentably, is still an unmet need.[55–57] Another obvious flaw is that current AI structure models, as in the AlphaFold database,[58] only contain monomers. Thus, an important first step in *in silico* modeling is to first determine and then model the appropriate (i.e. the in solution/the cell occurring) oligomeric state. At the time of writing, there was no dedicated tool available to predict and model the most relevant oligomeric state *ab initio*, and the current standard procedure appears to be to infer cardinality by aligning the monomer model to the closest PDB homolog, followed by remodeling the oligomer (which can, for example, be achieved with AlphaFold multimer).[59]

Another shortcoming of AI structure prediction is the inability to directly infer even obligate non-proteinogenic protein parts or to deal with non-canonical amino acids. General predictions for ligand binding are, however, a standard annotation in Uniprot,[60] and dedicated tools for binding-site prediction such as PrankWeb,[61] ScanNet,[62] Fpocket,[63] and CAVITY,[64–66] or the human proteome-specific HProteome-Bsite[67] exist. BioMetAll can be used to specifically find potential metal binding sites, based on backbone geometries alone.[68] In addition, AlphaFill is a recently released tool and accompanying webserver that allows researchers to directly obtain a structure with cofactors and other common ligands.[69] The method is essentially an automated pipeline that recapitulates the common modeling task of transplanting ligands from homologous structures. Such transplants usually work well for structurally closely related proteins or conserved ligand binding cavities, as is often the case for cofactor and nucleic-acid binding sites.[70–71] However, a more sophisticated approach is needed when no experimental structure template exists. A recent AI method called NodeCoder aims to solve this problem by using a graph convolutional network to perform structure-based predictions for ligand, peptide, ion, and nucleic acid binding sites.[72]

The next instrument in the computational enzyme engineering toolbox is docking – one of the most frequently used *in silico* methods for studying the interactions between organic molecules and biological macromolecules. Docking is a relatively fast computational method and is used extensively in protein engineering and drug discovery. Established algorithms typically sample and rank potential ligand binding modes in a receptor protein structure using an energy function,[73–74] but ML-based methods are also entering the field and await wider benchmarks.[75] Comparisons of docking algorithms occur regularly in the "Critical Assessment of PRedicted Interactions" (CAPRI) competition,[76] or in dedicated studies.[77] Because the success rate of docking is variable, it is sometimes recommended to use a combination of available tools, as none of the algorithms clearly exceed the others on all fronts.[73–74]

While the most commonly used docking tools work best with small molecules, specialized docking algorithms are typically more suited for larger flexible structures like peptides or nucleic acids. Haddock[78] is a widely used macromolecular docking algorithm and webserver with numerous described adaptations and applications.[79] Rosetta FlexPepDock is a dedicated tool of the Rosetta commons software suite[80] to dock peptides with full backbone flexibility. Although it can generally not take non-proteogenic parts into account,[81] it can be adapted to accept ncAAs.[82] For docking nucleic acids, FlexLzerD,[83] LightDock,[84] or the recent MELD-DNA[85] can be used. Furthermore, RoseTTAFoldNA[86] is a variant of the Baker lab's implementation of the AlphaFold framework, RoseTTAfold,[43] for the prediction of nucleic acid-protein complexes.

When docking small-molecule non-proteinogenic parts such as cofactors into apo protein structures, docking programs do not need to be specifically adapted, as the support for a wide range of ligands falls within their core competencies. When docking other ligands, such as a substrate, to a holo enzyme, non-proteinogenic parts can usually also simply be included as part of the receptor structure. The exception to this rule are metals or other metal-containing compounds like heme, as the energy functions of docking algorithms have trouble accurately representing the complex interactions of metals with the surrounding residues, ligands, and water molecules (also called the coordination sphere of a metal).[87–88] Thus, docking algorithms specifically optimized for metalloproteins like EADock,[87] MpSDock,[88] or more computationally expensive docking methods that implement quantum mechanical calculations can be used.[88–89] The molecular modeling platform GaudiMM also includes a functionality for docking bare metals that takes these coordination spheres into account.[90] A comprehensive review discussing metal groups in docking is given by Riccardi et al.[91] It is the common practice to visually inspect docking results to assess the plausibility of the outcome with respect to orientation, shape complementarity, hydrogen bonds, and hydrophobic contacts.[92] Such manual scoring based on "chemical intuition" represents unfortunately an understated hurdle for novices in the field.[43]

2.2. Simulating the protein structure

Once an enzyme structure with docked ligand and cofactor has been obtained, its behavior can be studied by simulating the protein structure. The output from such simulations can help interpreting experimental results, cross-validate mutations predicted by other means, or help investigating conformational dynamics and create ensembles to serve as input for other computational design methods.[8,93] One of the most frequently used methods for simulating protein behavior is molecular dynamics (MD) simulations, which calculate the movement of molecules by applying Newton's equations of motion to a protein's atoms. A set of energy functions called a force field is used to describe all bonded and non-bonded interactions acting on a system. The result is a time trajectory that can give a much more detailed picture of the precise mechanism and behavior of a protein than a static structure.[94–95] Before discussing in detail the most commonly applied all-atom simulations and how to adjust them for enzymes with non-proteinogenic parts, we will first have a look at both a more simplified and a more detailed simulation approach and their use cases.

Coarse grain (CG) force fields streamline the simulation by turning larger functional groups into a single bead. This procedure greatly speeds up calculations of large systems and smoothens the energy landscape, which allows for significantly greater conformational sampling.[96] MARTINI[97] is a popular coarse grain force field due to its generalizability,[96] but more specialized CG force fields like oxDNA[98–99] and oxRNA[99] for nucleic acids, or AWSEM[100] for proteins have also been developed. Although the loss of detail inherent to the coarse-graining process often limits its use in enzyme engineering, coarse grain MDs show their power with complex systems, for example in the investigation of large conformational changes inherent to many enzymes.[101] Due to their vastly decreased calculation time, CG MD can also compensate accuracy with throughput, and has for instance been used to screen candidate sites for protein modifications.[102–103] MARTINI has also been applied as an alternative method to investigate protein-ligand binding, functioning as an intermediate between fast but simplified docking methods, and more informative but computationally expensive all-atom MD simulations.[104] Although finding good mappings from atoms to coarse-grain beads can be challenging for non-proteogenic parts, MARTINI parameters for various nucleotide-based cofactors have been published,[105] and an automated MARTINI pipeline and database for small molecule mappings facilitate future use.[106]

On the opposite side in terms of accuracy for simulating biophysical systems are quantum mechanics (QM) and mixed QM/molecular mechanics (MM) calculations. This approach is especially valuable for studying the transition state of catalytic mechanisms, as classical MD cannot describe the transfer of electrons or covalent bonds. Since QM calculations scale exponentially and are computationally very costly, it is only feasible to simulate a small section of an enzyme.[107] As these restrictions typically don't even allow the simulation of an entire catalytic pocket, multiscale QM/MM methods were developed.

In this approach, a core region of the studied system is simulated with QM, while its surrounding is simulated using MM, i.e. via classical MD parameters.[108–109]

QM calculations generally depend on two factors: the level of theory, which corresponds to the specific approximation of the full Schrödinger equation, and the basis set – system-dependent functions that represent the orbitals of the molecules involved.[110] The level of theory used for QM and QM/MM calculations is usually some form of density functional theory (DFT).[107] The commonly used B3LYP hybrid functional, for instance, combines DFT with a slightly different level of theory called Hartree-Frock.[111]

As QM calculates energies *ab initio*, non-proteinogenic parts within the QM region usually require no special attention, while the same (parameterization) considerations as for MD (see below) apply to parts in the MM region. Although DFT-based methods generally represent the best trade-off between reliability and computational cost for describing metals,[112] their accuracy may be limited for metals with both high- and low-spin states,[113–114] although such systems are uncommon in the pathways of most metalloproteins.[113] Some metalloproteins have also been found to be particularly sensitive to the embedding scheme (the QM and MM connection region),[115] and QM/MM calculations often cannot cover the long timescales required for optimization of the metal coordination spheres.[112,116]

Due to the wide availability of atomic data for biological systems, all-atom MDs are the most frequent type of simulations in literature.[117] A variety of all-atom force fields is available, which primarily differ in their treatment of dihedrals and non-bonded parameters, as they usually aim to reproduce slightly different experimental or QM benchmarks.[118–119] The most commonly used all-atom force fields for biological applications are AMBER,[120] CHARMM,[121] OPLS-AA,[122] and GROMOS.[123] AMBER and CHARMM are specialized for use in biological systems, while OPLS's parameterization is more general.[124] GROMOS models hydrogens implicitly, which reduces detail, but makes it well suited for the study of larger systems. Since it may take a long time for a simulation to sample behaviors such as ligand diffusion[125] or conformational dynamics[93] – a reflection of the discrepancy between the timescales of biological processes and computational simulation capacity – enhanced sampling methods such as accelerated MDs[126] and metadynamics[127] have been developed. These and other strategies to prevent excessive sampling of non-relevant states have been extensively reviewed by Lazim et al.[128]

All-atom force fields generally require pre-defined parameters and fixed partial charges for each compound. Consequently, no bond or charge changes can occur during an MD simulation, and the protonation state of all compounds must carefully be fixed beforehand. To that end, the pKa of each residue is usually estimated via either Poisson-Boltzmann calculations[129–130] or empirical methods.[131–132] For some systems like cofactors[133] or disordered peptides,[134] it should be kept in mind that more than one protonation state might be relevant. While the parameters for amino acids and nucleotides are included in the common force fields, non-proteogenic compounds like ligands, ncAAs, or organic cofactors often are not. Therefore, such compounds must be carefully parameterized before a simulation, a process that can require significant effort and is addressed in the subsequent sections. For the study of electrostatically sensitive systems such as catalytic sites with complex non-proteogenic compounds like large aromatic cofactors or metal ions, fixed partial charges obtained from the parameterization process might not describe the relevant interactions accurately enough. To address these shortcomings, polarizable force fields like CHARMM Drude[135] or AMOEBA[136] were developed, which can shift their charges according to the local environment and thus provide a much more accurate representation of electrostatic interactions in protein cavities.[137] Although their high computational cost has made their use relatively rare, they have proven valuable for studying metalloproteins and nucleic acids in particular.[137]

2.3. Obtaining force field parameters

Since many computational enzyme engineering workflows involve MD simulations, special care must be taken to parameterize non-proteogenic parts. The main effort in the parameterization of new molecules usually consists in the calculation of partial charges.[138] A thorough computational design approach starts with QM calculations (e.g. using software packages like Gaussian)[139] to determine the electrostatic potentials and fit to point charges using a restraining penalty function that prevents unrealistic behavior.[140] Partial charges calculated this way are called restrained electrostatic potential (RESP) charges.[141] Usually, Hartree-Frock (HF) is sufficient as the level of theory for organic molecules, while the hybrid functional B3LYP is common for metal-containing systems, in both cases making use of the 6-31G* basis set.[120,142] Although force field developers often recommend a consistent charge approach,[120] RESP charges should theoretically be force field-independent and compatible between levels of theory.[143]

A computationally cheaper alternative is automatic on-the-fly calculation of charges, often involving semi-empirical approximations that greatly simplify the Schrödinger equation with empirically derived parameters.[144] An approximation called "neglect of diatomic differential overlap" (NDDO) is commonly used, as implemented in AM1 or the slightly more accurate PM3 methods.[145–146] Another fast strategy is employed by CHARMM's general force field protocol CGenFF, which uses libraries of fragments with optimal charges that are pieced together using bond charge increment (BCI) rules to obtain partial charges for the full molecule.[147–148]

Automatic parameterization pipelines are often included in or provided by a particular force field or software developer: for AMBER there is ANTECHAMBER,[120] which was further automated with a python pipeline called Acpype.[149] The software package YASARA uses AUTOSMILES, which parameterizes for AMBER, but also checks a database of pre-parameterized small molecules.[37] CHARMM has a pipeline for parameterization implemented in the CHARMMing modeling package.[148] For

GROMOS, PRODRG,[150] and, since recently, the automated topology builder (ATB), are available.[151] For OPLS-AA there is BOSS[138] and the corresponding automation LigParGen.[152] Except for CHARMMing and PRODRG, which use a BCI-based algorithm, all these pipelines rely on semi-empirical methods.[150] Of note, automatic parameterization tools can produce unreliable results for complex ligands such as cofactors, and it is thus considered best practice to specifically determine and, if possible, validate parameters for important non-proteinogenic parts.[153] In the following section, we discuss available parameters for some common molecules.

2.4. Optimized parameters for non-proteogenic parts

The adequacy of faster semi-empirical charges obtained from automated pipelines can vary depending on the compound. Especially electronically complex systems such as aromatic groups and heavier elements should be treated with caution. For common organic cofactors, more thoroughly optimized and tested parameters are available.

For instance, although AMBER and CHARMM have built-in topologies for nucleotides, they are more accurately represented using the refined force field parameters Parmbsc0/Parmbsc1[154–155] and Bsc0/Bsc1.[156] Sugar moieties in glycosylated residues are well represented using either CHARMM, which only slightly underestimates, or the AMBER refinement GLYCAM,[157] which slightly overestimates the binding strength of carbohydrate-protein interactions.[158] Both AMBER and CHARMM also include parameters for phosphorylated amino acids, which generally exhibit good performance, although discrepancies between the two methods were noted.[159] Various databases with parameters and RESP charges for ncAAs and PTMs are available,[160–161] and a thorough review by Li et al. provides a useful table of parameter sets.[13] Although RESP charges are available for nicotinamide cofactors, modern automated pipelines were found to produce reliable parameters for all major force fields.[162–163] Optimized S-adenosyl-methionine (SAM) parameters have been published for various force fields,[164–165] and PLP RESP charges are available for AMBER.[166] Parameterization of flavins requires extra care because the electron density in the central three-ring structure cannot easily be fragmented into smaller groups.[167–168] Although FAD[169] and FMN[170] have been parametrized with RESP charges for AMBER, a recent comparison shows that automatically generated topologies for newer versions of AMBER, CHARMM, and OPLS are also able to reproduce some of the more subtle electrostatic effects relevant to FMN.[171] While the same work showed that CHARMM force field topologies had trouble reproducing the effects of nearby amino-acid substitutions on flavin binding, newer, more thoroughly derived parameters have since been published.[172]

Metals are widely employed as cofactors in enzymes (Figure 1B) and are some of the most difficult atoms to properly parameterize. Due to their complex electron shells, the distribution of charges and their coordination can change significantly with the specific environment.[173] It is therefore often difficult and sometimes impossible to reproduce all desired properties using a single topology, and the optimal parameterization strategy is system-dependent.[174–175] Simply using the standard or automatic parameters will likely lead to inaccurate energies and geometries.[142]

For metals that are not part of an organic cofactor, three parameterization strategies exist: non-bonded models, which simply keep the metal ion as a separate entity but optimize its parameters; bonded models, which approximate the strongest electrostatic interactions as a covalent link; and lastly models that try to recapitulate the complex electronic structure with multiple "dummy" atoms. A highly comprehensive review of these methods was written by Li et al.[173] For proteins where the metal remains in a relatively fixed position, the bonded approach is most commonly used, as it accurately reproduces the desired interactions and geometries. The resulting (unrealistic) static coordination sphere is acceptable in situations where the metal is tightly bound to its coordinating residues,[142,173–174] whereas non-bonded or dummy-atom models can be considered where a more dynamic behavior of the metal is desired. Even if thoroughly parameterized, accuracy is ultimately limited by the simplification of the complex heterogeneous charge distribution into a single point, which can lead for example to a systematic underestimation of the interactions of divalent cations with water.[174] The cationic dummy model (CADM) attempts to deal with these problems by distributing the charge over several mock atoms with fractional charges that surround a central core,[176] although the unrealistic negative core charge may distort results in some situations.[173] While the dummy-atom model produced highly accurate geometries for double-metal centers, it was outperformed by the classical non-bonded models for single-metal centers.[177–178] A further extension of the dummy model by the Kamerlin group showed a significant improvement compared to the non-bonded model when more complex metal binding sites are involved, confirming the particular strength of the dummy model in describing complex multi-metal sites.[179]

Metal cofactor-specific approaches include the zinc AMBER force field (ZAFF) or its extended version (EZAFF).[180–181] Heme has been parameterized with various coordination spheres,[182] and general topologies for CHARMM and AMBER, as well as for the specific contexts of Photosystem II[183] and the P450 enzyme family are available.[182,184] For metal-containing systems more generally, AMBER's metal center parameter builder (MCPB) has been developed for bonded potentials or metal-containing cofactors. The tool is complimented by the visual force field derivation toolkit (VFFDT), which enables a user-friendly visual assignment of bonded potentials.[185–186]

2.5. Designing catalytic properties

Although simulating the behavior of a protein gives valuable insight into its molecular mechanism, the ultimate aim in protein engineering is finding mutations that positively impact a property of interest. For enzymes, a computational prediction method often involves remodeling the active site. A commonly used tool for this task is RosettaDesign,[187] a part of the Rosetta Commons software suite for redesigning protein structures. The core of RosettaDesign is a fixed backbone algorithm that remodels the amino acid sidechains without altering the backbone. The standard Rosetta energy function[188–189] is used to score each design.[187,190] A Rosetta application more specifically aimed at enzyme design is also available, and can remodel the catalytic site around a given 'Theozyme' – a theoretical model of the desired active site including ligands and cofactors obtained using QM calculations.[196] Although the tool was initially limited to the canonical amino acids, rotamer libraries for non-canonical amino acids were later developed, too.[238] The modular nature of Rosetta means that most of its numerous packages access not only the same main energy function, but also the various "patches" for unusual residues that have been deposited. Various graphical user interfaces to Rosetta[239–240] have been developed that can aid in the discovery of this kind of documented, but potentially unpublished features. In principle, all users can add additional parameters for their specific molecules of interest. For special cases like modified residues, custom protocols are commonly developed. For instance, to engineer a ubiquitin conjugation enzyme, a custom Rosetta protocol was used to model the covalent substrate-enzyme complex, create a rotamer library, and score via docking.[241]

Several more generalized pipelines in the literature utilize one or more algorithms from the Rosetta suite, and over 80 such methods were recently reviewed.[242] The Triad software package[197] also includes several pipelines for enzyme redesign, and, besides Rosetta, includes other scoring functions such as Dreiding[243] or PHOENIX.[244] Another exemplifying enzyme redesign workflow using Rosetta is CASCO, which can be used to engineer enantioselective enzymes.[245] The protocol relies on the definition of a near attack conformation (NAC),[246] i.e. a geometric orientation of the ligand-enzyme complex compatible with catalysis toward the desired enantiomer. CASCO uses RosettaDesign to find mutations that stabilize this conformation and screens the best hits with short MD simulations, in which the percentage of NAC time is used for ranking. Variations of this approach incorporate an initial MD step to allow RosettaDesign to screen mutants in different conformations,[247] or alternatively leave out the MD screening altogether.[248] Some example applications of methods involving Rosetta and MD are discussed in section 3 of this review.

An alternative to designing mutants that affect a single property is the generation of 'smart' mutant libraries likely to affect catalytic properties, which can be screened more generally for new functions. FuncLib, for instance, combines RosettaDesign with evolutionary information to create a small library of promising multipoint mutations that create a new active site.[199] The method involves limiting sequence space via MSAs to filter out improbable mutations. The remaining variants are tested with RosettaDesign by first removing mutations that destabilize the protein, and then scoring all remaining multipoint mutations.

Although MSAs, and thus FuncLib, cannot deal with non-canonical amino acids, it has been verified on proteins with cofactors and catalytic metals.[199] HotSpotWizard is one of several webserver-accessible computational tools of the Damborsky lab for creating smart libraries. It combines MSAs with structural computational tools to find mutational 'hot spots' likely to affect a property of interest.[200,220]

Instead of targeting the active site, several energy-based workflows aim at altering catalytic properties by engineering protein tunnels.[249] A highly popular tool to rapidly identify substrate access or product release channels in static structures or within an MD trajectory is CAVER.[203,250] More complex approaches include steered MDs to pull a ligand along an assumed access channel and concurrent energy calculations.[251] A faster alternative is PELE, a Monte Carlo sampling method that stochastically perturbs a ligand to study its diffusion and allows a much broader sampling of the energy landscape than MD.[202] In either case, classical energy-based force fields apply, so non-standard residues and molecules can be integrated if proper parameters are provided. Several other "approximate"[252] tools for investigating ligand transport also compromise between the accuracy and computational cost of MD simulations.[252] While Caverdock[204] or GPathFinder[205] use Autodock Vina[253] to score the energy of the ligand along the entry path, other examples such as MoMa-LigPath,[206] and ART-RRT[207] use planning algorithms borrowed from robotics to find optimal ligand pathways. Yet, due to the limitations of fixed structures, these methods are often combined with MD to further investigate or verify mutations.[252] Although these tools typically support input of proteins with cofactors and other non-proteogenic parts, it should be kept in mind that these might not be accurately represented by algorithms like MoMa-LigPath or ART-RRT, which use a highly simplified energy function that ignores electrostatics. For methods that rely on docking algorithms, the previously discussed limitations, such as the inaccurate representation of metals, apply.

Besides mutations in obvious functional hotspots such as the active site and tunnels, directed evolution campaigns frequently find that distal mutations can also exhibit significant effects on activity.[254–256] An extensive recent review on the topic elaborates on the challenges in the design of such mutants,[257] which typically are hard to predict and often require long MD or QM/MM simulations to investigate.[254] A promising method for analyzing MD trajectories is a correlation-based technique called the shortest path map (SPM), which identifies hotspots for distal mutations via a graph network of allosteric interactions.[201,258] Being MD-based, the tool is in principle amendable to proteins with non-proteogenic parts, although they will not be included in the graph network itself. Given the inherent difficulty of rationalizing distal mutations *a priori*, machine learning (ML)-based prediction methods represent a promising alternative approach.[259–261]

Also in other contexts, recent years have seen a surge of ML algorithms for predicting promising mutations in enzymes. MutCompute is a neural network trained on structural features of amino acids in natural proteins to predict plausible mutations.[211] Although only amino acids are explicitly addressed by MutCompute, the inclusion of cofactor-containing proteins in the training set might allow it to take such parts implicitly into account. Some ML algorithms are also directly related to cofactor-specific properties. Specifically for protein structures featuring a Rossman fold, a deep learning algorithm called Rossman-toolbox predicts the preference between NADH and NADPH,[212] and Cofactory, also optimized for Rossman folds, predicts a mutant's cofactor specificity and binding affinity for NAD, NADP, FAD, and SAM.[212] A purely sequence-based ML algorithm for ranking mutations that may cause a switch between NAD and NADP specificity is also available, although the creation of a new optimized model for each protein is required.[262] Another promising, although computationally expensive, approach is to use ML to predict energies, by training them on MD or QM/MM trajectories of a protein of interest. At the expense of throughput, such algorithms may lead to more robust predictions than simply using these simulations for rational design.[263–264]

Lastly, algorithms have also been developed for introducing new metal binding sites into a protein. MetalSearch,[265] HostDesigner,[209] and OptGraft[210] scan a protein to find optimal positions to introduce a new metal binding center.

Although redesigning proteins to contain non-canonical amino acids has become an increasingly large field,[13] it is mainly the field of pharmaceutical peptides,[82,266] and artificial metalloenzymes where ncAAs are used in computational design.[267–269] For the latter approach, a protein scaffold is designed to host a synthetic metal-containing cofactor for the catalysis of new-to-nature chemical reactions.[270] Although this strategy is in principle also applicable to naturally found enzymes, the majority of literature reports either *de novo* or non-catalytic proteins as scaffold. Consequently, we refrain from extensively discussing these approaches here and refer the reader to an example discussed in section 3, as well as review articles on the topic in general,[271] and their computational design in particular.[272–273]

2.6. Designing Thermostability

Thermostability represents an important non-catalytic property targeted in enzyme engineering due to its relevance for industrial processes.[274] Computational strategies to identify stabilizing mutations span from evolutionary algorithms to energy function-based predictions.

Evolutionary methods include consensus-guided mutagenesis, which predicts conserved or high-frequency residues in MSAs as stabilizing; ancestral sequence reconstruction, which reconstructs the sequences of enzymes at different nodes of a phylogenetic tree; and the ancestral mutation method.[275] The ability to improve thermostability via the latter two methods is hypothesized to stem from the adaptation of ancestral organisms to a hot earth environment.[276] Specific considerations for cofactors and metal-dependent proteins are usually not necessary for these techniques. However, although binding motifs and catalytic residues are often conserved, the introduced mutations can affect cofactor binding and specificity. For example, ancestral proteins were reported to exhibit a switched preference between nicotinamide cofactors[277] or flavins.[278]

In contrast, energy-based predictions require a protein structure, and typically calculate the $\Delta\Delta G^{\mathrm{Fold}}$, i.e. the difference in the free energy of (un)folding between a mutant and a wild-type structure.

The predictive capability is dictated by the individual terms of the energy function, which include amino acid hydrogen bonds, hydrophobic interactions, and so forth, but often do not contain parameters for non-proteinogenic parts.[279] Consequently, several *in silico* stabilization techniques ignore them, and foresee the rather crude solution of excluding residues around the ligand pocket for predictions.[222] Although the energy functions used to score thermostability are similar to those used in MD force fields, they can be specifically optimized. FoldX[280] is a popular software package[281] in which newer versions introduced support for nucleotides and metals. The current release (FoldX 5.0) also includes a plugin for the molecular modeling software YASARA[37] that allows parameterization of small molecules like cofactors or ncAAs.[213] Another commonly used option is the DDG monomer algorithm included in the Rosetta software suite, which calculates the change in free energy for point mutations. Notably, this Rosetta algorithm is unable to compute non-proteogenic parts.[214]

Various pipelines and web servers for the prediction of stabilizing mutations that integrate these tools are available. FireProt[282] and PROSS,[221] for example, combine MSAs to find consensus mutations with FoldX and/or Rosetta energy calculations, and this ability has also been integrated into HotSpot Wizard 3.0.[220] However, at the time of writing, neither of these workflows supports the latest FoldX integration of cofactors. More manual protocols include FRESCO,[283] which calculates the $\Delta\Delta G^{\mathrm{Fold}}$ of all of a protein's single mutants with FoldX and Rosetta and screens them via high-throughput MDs. Cofactor-dependent enzymes can be and have been[284] accommodated in FRESCO and PROSS by disregarding the ligand-surrounding residues while conducting energy calculations, or incorporating parameterization with FoldX 5.0.[222] Another workflow is GRAPE, which incorporates two additional prediction algorithms to rank mutants – consensus design and the statistical energy function ABACUS[218] to rank mutants. Experimental single variant results are then computationally clustered and combinatorial mutants selected via a greedy algorithm.[223]

In addition, several other energy functions have been developed. CUPSAT[215] uses an empirical energy function specifically parameterized on amino acids, while pSTAB predicts stabilization using ensemble-based statistical mechanics.[216] Proteus tries to introduce new pairs of interactions instead of looking for single point mutations.[217] These algorithms generally do not take non-proteogenic parts into account, although CUPSAT's web service lists residues involved in the binding of small molecules present in the PDB, and so permits avoiding mutating residues important in cofactor binding, although not for catalytic metals. SCHEMA is a protein fragment combination algorithm that uses an energy function to construct chimeric proteins,[233] and has been successfully used to stabilize a metal-containing protein.[285] SDM[234] predicts stabilizing mutations based on environment-specific amino-acid substitution frequencies and therefore does not consider non-proteogenic parts explicitly. Methods such as B-FIT,[235] which identify flexible residues via a crystal structure's B-factors are not limited to proteinogenic atoms in the identification phase, but, depending on the implementation, might be in the redesign phase.[286]

Due to the availability of relatively large and consistent datasets, thermostability engineering is also a popular target for ML methods such as mCSM,[224] BayeStab,[225] PoPMuSiC-2.0,[226] DeepDDG,[229] ABACUS-R,[230] and MutCompute.[211] These algorithms extract structural information like secondary structure, surface area or residue environment from the crystal structure before passing it on to the ML algorithms, during which non-proteogenic molecules are usually ignored. Instead of exclusively relying on structure, more sequence-based ML approaches also exist: i-Mutant[228] includes a purely sequence-based functionality, and Prethermut[227] works on sequence information exclusively. As mentioned before, sequence-based approaches have the advantage of potentially processing non-proteinogenic parts implicitly, but do not support input beyond the standard alphabet of amino acids. ML can also be used as a meta method that integrates the outputs of other predictive algorithms to generate more accurate predictions than they do in isolation. This approach is followed by iDeepDDG,[229] iStable,[231] Dynamut,[232] and DUET;[287] and whether these accurately represent enzymes with non-proteogenic parts depends on the integrated methods.

ML approaches may also not require an explicit functionality for non-proteinogenic parts and can handle them without adaptation, as they are trained on diverse datasets like those provided by ProThermDB[288] or SKEMPI.[289] While information about the viability of mutations near cofactors thus may implicitly be provided via these datasets, the inability to disentangle the internal processing of ML algorithms makes a general assessment difficult and requires empirical validation. Notably, a benchmark study comparing the performance of PoPMuSiC-2.1, i-Mutant 2.0, i-Mutant 3.0, CUPSAT, SDM, and mCSM to predict stabilizing mutations of the heme-containing proteins myoglobin and cytochrome C, found the ML algorithm i-Mutant 2.0 to generally have the best correlation with experimental data.[290]

3. Examples

Despite the difficulties to model enzymes with non-proteinogenic parts, their catalytic value triggered a host of computational engineering campaigns. In this final section, we showcase some of the many literature examples, which highlight both the wide scope of catalytic mechanisms, as well as the various fields in which computational enzyme engineering is important. Perhaps the most significant application of enzymes is biocatalysis and industrial enzyme biotechnology. Frequent research goals in this field are the discovery or design of enzymes that are thermostable, solvent tolerant, highly expressed, and display a broad substrate scope to efficiently and economically produce chemicals valuable for human processes and products.[291–293] As enzymes often compete with cheap chemical catalysts, the main opportunities currently lie in high-value products, e.g. pharmaceuticals,[294] and chemically challenging reactions, which enzymes frequently enable via cofactors[295] or can be catalyzed by artificial metalloenzymes.[296]

Some enzymes are also used as therapeutic drugs, and a rapidly growing field of research is investigating their potential for future biomedical applications.[297] For the following examples of computational enzyme engineering in these fields (Figure 3), we examine the challenges faced by the studied system, and outline the approaches taken by the authors to tackle non-proteinogenic components.

To create an artificial metalloenzyme for the otherwise chemically challenging enantioselective water addition reaction, Roelfes and coworkers incorporated the copper-binding ncAA (2,2′-bipyridin-5yl)alanine (BpyA) into the non-enzymatic protein scaffold LmrR.[298] An extensive computational framework was used in the design: DFT calculations helped optimizing the copper-BpyA interaction, the copper-BpyA complex was parameterized with MCPB, and the ligand docked with GOLD,[194] due to its inbuilt parameters for metals. A subsequent MD simulation then identified promising positions for installing a catalytic acidic residue. In another study, BpyA was incorporated in a serine protease and a luciferase to engineer metal-responsive reversible protein switches.[299] Based on the results

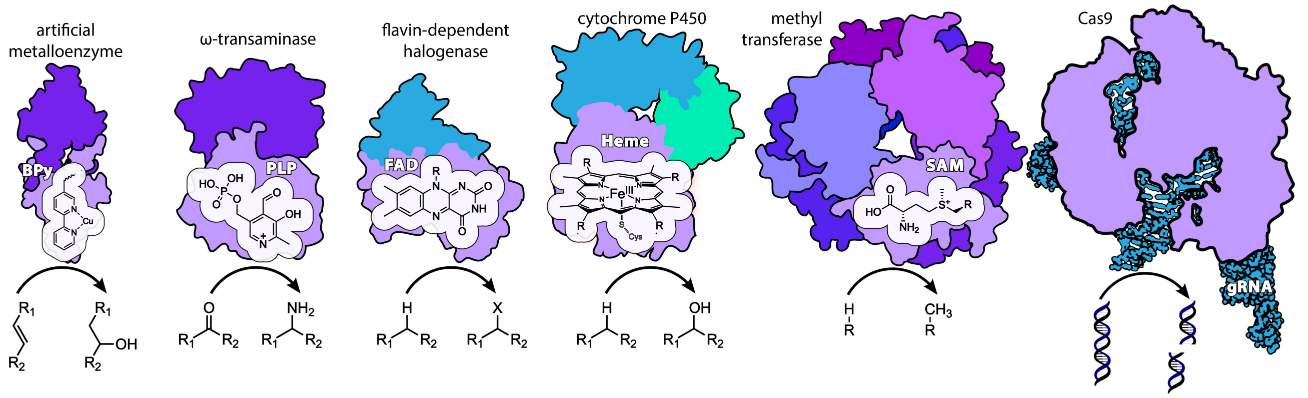

Figure 3. Examples of engineered enzymes containing non-proteogenic parts. Protein parts are represented as surface cross sections, where shades of purple indicate homooligomers and differing colors indicate domain fusions. Non-proteinogenic parts, as well as the canonically catalyzed reactions, are shown as structural formulas in the case of ncAAs and cofactors, or schematically for the nucleic acids involved in Cas9 catalysis.

of extensive MD simulations, BpyA was introduced at two positions observed to undergo maximal changes in distance during the enzymes' natural conformational sampling. Due to the ncAA's high metal affinity, the altered residues fixed the protein in a closed conformation upon addition of divalent cations. While BpyA was parameterized with literature data using CHARMM's CGENFF, the authors derived custom parameters for the BpyA-metal interactions based on experimental binding rates.

In more traditional biocatalysis research, pyridoxal phosphate (PLP) cofactor-dependent transaminases are such a popular target for enzyme engineering that a dedicated database was established.[300] PLP forms a covalent reaction intermediate to facilitate the exchange of an amino with a keto group. ω-Transaminases are of particular interest, due to their potential application in the synthesis of compounds of pharmaceutical and agrochemical relevance.[301] One of the early successes in the computational engineering of ω-transaminases used docking and rational design in conjunction with directed evolution. The resulting enzyme variant was able to efficiently produce sitagliptin,[302] an anti-diabetic medication, whose traditional synthesis requires toxic heavy-metal catalysts.[303] In later work, the enantioselectivity of ω-transaminases was boosted for a host of substrates in a strategy employing RosettaDesign.[304] To that end, the PLP-substrate covalent intermediate was created manually, and its geometry and partial charges derived via semi-empirical QM. A rotamer library created with YASARA then was docked with Rosetta. This strategy produced 38 designs, whose experimental validation showed that the docking energy correlated well with enantiomeric excess. Nevertheless, outliers were observed and could only partially be explained by a detailed analysis of active-site waters. Thermostable ω-transaminases were also created, and a highly active and cosolvent-tolerant variant with an increase in melting temperature (T_m) from 62 °C to 85 °C was achieved using the FRESCO protocol.[305] In a structure-independent engineering approach, an *in silico* resurrected ancestral ω-transaminase yielded a significantly more promiscuous enzymes with up to 10-fold increases in activity and a 10 °C increase in T_m.[306]

Flavin-containing enzymes are also frequently engineered for biocatalysis. Flavin adenine dinucleotide (FAD) dependent halogenases (FDHs) bear promise of industrial application due to their ability to selectively introduce halides at C–H bonds, which may improve bioactivity and -availability of pharmaceuticals[307] or represent synthesis intermediates. As FDHs tend to be catalytically inefficient and highly substrate specific,[308] enzyme engineering was used to improve their synthetic performance. To investigate the catalytic mechanism and selectivity-conferring residues, QM/MM[309] and MD[310] simulations were performed and opened the door for rational design. One group also used a combination of docking, QM, and MD to investigate the mechanism behind the selectivity of an FDH's halogenation site. Using RESP charges for FAD, the authors found that docking in combination with QM calculations to exclude catalytically inactive binding poses provided an effective predictor for site selectivity.[311] Another paper used CAVER to improve the low efficiency of an FDH caused by the leaky transfer of an intermediate. The workflow found three key residues that formed a bottleneck for the diffusion of the intermediate. Site-saturation mutagenesis on these residues identified a mutant that displayed a threefold improvement in catalytic efficiency and a serendipitous increase in T_m by 12 °C due to improved hydrophobic interactions in the core.[312]

Among metal-dependent enzymes, heme-employing cytochrome P450s are a popular enzyme class in synthetic applications due to their ability to catalyze a broad range of synthetically challenging oxidation reactions.[313] A commonly studied enzyme is P450 BM3 (CYP102A1), a natural P450 and P450 reductase fusion that makes it electronically self-sufficient. In addition to heme, the full protein also contains FAD and FMN as prosthetic groups and requires NADPH as electron donor. The mechanism of P450 BM3 has been studied extensively using QM/MM and MD,[314] most commonly to rationalize experimentally found mutations. In this way, the regioselectivity for the malaria drug artemisinin[315] and the regio- and enantioselectivity for the antibacterial compounds thiochromanone and 1-tetralone were elucidated.[316] Computational predictions have, however, also been applied to aid in the rational design of selectivity-altering P450 BM3 mutants.[317] A dedicated review collecting molecular dynamics simulation studies of P450 BM3 is available,[314] illustrating both the popularity of the enzyme and the variety of approaches to model it appropriately.

Rosetta was used to model the lowest energy transition state to rationally design more promiscuous P450 BM3 mutants,[318] and CAVER to aid in the design of mutants that reduce water diffusion for increased coupling efficiency.[319] An ancestral sequence reconstruction of P450 BM3 resulted in a highly active variant and an impressive increase in T_m of 30 °C.[320] Despite these successes, the complex structure and mechanism of P450s have largely prevented the design of mutants fit for industrial applications. As increases in activity can easily come at the cost of desired properties like efficiency or regioselectivity, a more holistic approach in designing these parameters simultaneously has been suggested.[321]

With the number of reports on ML-based protein engineering steadily increasing, examples of ML-engineered enzymes with non-proteogenic parts begin to enter the literature. A ML algorithm specifically trained on the NADPH-dependent family of alcohol-forming fatty acyl reductases (FARs) predicted mutations that resulted in a twofold increase in activity.[322] Another group successfully predicted thermostability and activity-increasing mutations in DNA polymerases with MutCompute.[323] ML models specifically trained to work in this context could also predict the activity of artificial metalloenzymes.[324]

In biomedical research, computational biology manifests mostly in drug design, which largely relies on the same procedures as enzyme engineering, but where the ligand, not the protein is variable. The fields overlap, because – as is the case for P450s – protein classes that make relevant drug targets can also be useful biocatalysts. SAM-dependent methyltransferases (Mtases), for instance, methylate proteins, nucleotides, and small molecules with the help of S-Adenosyl methionine (SAM).[325] While a lot of insight was gained from the computational study of Mtases from pathogens such as SARS-CoV 2,[326–327] these enzymes can also be employed in the synthetic methylation of pharmaceuticals, which can vastly increase a drug's potency (the "magic methyl effect").[328] Although most Mtases are highly specific, a few more promiscuous variants form a promising starting point for the methylation of a variety of compounds,[325] and the origin of their promiscuity was studied via QM/MM and MD simulations.[329] One promiscuous Mtase was engineered to produce cis-α-irone, an important fragrance compound. To increase specificity, a combination of MD and energy calculations was combined with experimental screening to find mutants with an impressive $>$10,000-fold and $>$1000-fold improvement in cis-α-irone activity and selectivity, respectively.[330] Another computational strategy to affect substrate specificity avoided modeling the SAM cofactor by using Rosetta to design an MTase-fused recruitment domain as prey that can capture a peptide-linked designed bait.[331] MTases have also been designed using the FuncLib tool to find variants capable of methylating pharmaceutically useful but biologically uncommon unsaturated N-heterocycles.[332] Funclib has also been applied to transform a promiscuous MTase into a new pyrazole-alkylating enzyme by screening for variants that accept natural SAM analogues (NSAs), which transfer other alkyl groups besides methyl.[333] In both these works, the effect of the discovered mutants was further characterized by DFT and MD, using RESP charges parameterized by ANTECHAMBER for the SAM cofactor.

Besides being catalysts for the production of small molecule drugs, enzymes are also employed as drugs themselves. Use of therapeutic enzymes is an established field with various drugs already commercialized.[297] With the advent of mRNA therapeutics, enzyme-encoding transcripts can also be delivered intracellularly to address metabolic deficiencies in genetic diseases.[334] Computationally engineered were for instance superoxide dismutases (SOD), a family of metalloenzymes that can scavenge oxygen radicals in the treatment of cancer, arthritis, and neurodegenerative diseases.[335] A combination of HotSpot Wizard and CUPSAT identified residues for site-directed mutagenesis and resulted in a mutant with improved kinetics and increased H_2O_2 tolerance in a mechanism elucidated via MD simulations.[336]

A strategy for the potential cure of genetic diseases is gene editing – a technology that in recent years made a leap toward reality by the discovery of sequence-programmable nucleases. To cleave their target DNA, a guide RNA and two catalytic magnesium ions are required by the prototypical Cas9.[337] The enzyme was the target of various computational enzyme engineering efforts, aiming to alter properties such as cleavage specificity or protospacer-adjacent motif (PAM) requirements.[338] Due to its complex and highly dynamic nature, a lot of effort has initially focused on investigating its precise mode of action. The catalytic mechanism[339] and the structural and catalytic role of the magnesium ions[340] were studied using a mixture of MD and QM/MM. MD was also used to gain insight into DNA/RNA binding,[341] enzyme activation via domain rearrangements,[342] and to unravel the effect of mutations in high-fidelity Cas9 variants.[343] The FuncLib tool allowed identification of highly active Cas9 variants,[344] and various ancestral Cas9 variants were created that accept different PAMs and guide RNA backbones.[345] Parameterizing the non-proteogenic parts in Cas9 for MD required some additional care: for MDs, DNA and RNA were generally simulated using either the Parmbsc0 or Parmbsc1 force field refinements, while the magnesium ions were usually modelled with refined non-bonded force field parameters.[339–340,342]

Another pharmaceutical enzyme design study with non-proteogenic compounds engineered cyclic GMP–AMP synthase (cGAS). This enzyme produces the signaling molecule cGAMP upon recognition and binding of cytosolic DNA, and so triggers the cGAS–STING pathway – a part of the innate immune system important for reporting cancer and viral activity.[346] Similarly to Cas9, it binds DNA and requires two catalytic magnesium ions. The employed MD simulations to investigate the mechanism used the Bsc1 force field corrections for DNA, but did not refine the magnesium ions.[347] To facilitate further studies of the pathway, computational enzyme engineering was then used to create a continuously active cGAS variant independent of DNA binding. The workflow employed Rosetta to find mutations that stabilize the active state in the absence of DNA.[348]

4. Summary and Outlook

Enzymes are powerful catalysts and are increasingly used in biocatalytic and biomedical applications.[6] Given the high fraction of enzymes containing non-proteogenic parts, in particular among catalysts for valuable reactions, it is essential that computational enzyme engineering tools can be reliably used with them. While a certain attraction lies in sequence or data-dependent methods that deal with non-proteinogenic parts implicitly, emerging ML tools still lack broader validation, and examples with proteins containing cofactors, ncAAs, or other modifications are rare to date. Although the recent breakthroughs in structure prediction[2] will further push the frontiers of computational enzyme engineering, atomic precision is often paramount for realistic modeling. Protein structure is only one (albeit essential) piece in mutant design workflows, whose success equally depends on accuracy in the subsequent steps of the various paths towards correct predictions. With "empty" and monomeric AI protein models abundant, oligomeric structure generation followed by transplantation and docking become the crucial next bottleneck. As manifested in our inability to reliably design drug candidates *in silico*,[349] docking in particular remains a computational challenge that leaves a lot to be desired in terms of reliability and quantifiable output.[350]

Although thoroughly performed MDs generally are an established and reliable tool, simulations often need to (or are too short to) compensate inaccurately docked complexes. Further, although parameters are available for nucleic acids as well as for common cofactors and non-standard residues, their retrieval and implementation constitute a significant hurdle, while deriving and validating new parameters typically require tools and skills unavailable to non-specialists. In this regard, automation, standardisation, and a central parameter repository might aid. Because computing power remains insufficient for simulations of timescales relevant for many enzyme reactions,[351] methods to increase sampling capacity are of great interest.[128] Larger conformational changes as undergone by many complex enzymes will likely remain unpredictable with MD, however. Dedicated or tweaked AI tools for reliable predictions of protein conformations would therefore represent another major breakthrough; otherwise, experimental protein structures, especially those with non-proteinogenic parts, may remain an important cornerstone of enzyme redesign for longer than some currently forecast.

Because experimental determination of many mutated sidechain conformations is practically impossible, ligand-binding pocket redesign tools will remain essential for many pipelines. While rotamer libraries and energy-based complex predictions are constantly improving, the sword of Damocles in form of generative ML models hangs also over this field. As enzyme *de novo* design is starting to become feasible thanks to implementing deep learning techniques,[352] the future role of biophysical modeling in enzyme design is not certain. Similar to how AI models and experimental protein structures are currently used in parallel, it seems likely that force field and ML-based design methods will also be used side by side and system dependent. Likewise, enzyme design may in the future adopt a hybrid strategy that integrates the current dualism of either *de novo* constructing backbones given a fixed catalytic pocket or sculpting a new active site into a fixed backbone. In such a flexible approach, relevant protein parts may freely be combined from natural proteins and engineered or from scratch designed (/hallucinated[53]/diffusion model-derived[54]) as required.

While examples of enzyme engineering with systems containing important PTMs are rare in literature (perhaps due to a bias toward bacterial enzymes in biotech applications), ncAA-dependent catalysts gain traction.[353] Computational enzyme redesign with these residues is, however, still in its infancy. While adapting existing or developing dedicated machine-learning techniques for this task will be challenging due to the scarcity of data, classical modeling can be more readily adjusted, and the topic thus represents a chance for rational design endeavours.

Acknowledgements

M.J.L.J.F. gratefully acknowledges funding from the Dutch Research Council (Veni grant VI.Veni.212.263).

Conflict of Interests

The authors declare no conflict of interest.

Data Availability Statement

The data that support the findings of this study are openly available in Github at https://github.com/kt-korbeld/cofactor-figure-data, reference number 609586835.

Keywords: Bioinformatics · Cofactors · Computational chemistry · Enzyme catalysis · Enzymes · Metalloenzymes · Molecular dynamics · Molecular modeling · Protein engineering

[1] S. Lutz, *Curr. Opin. Biotechnol.* **2010**, *21*, 734–743.

[2] M. AlQuraishi, *Nature* **2020**, *577*, 627–628.

[3] M. S. Packer, D. R. Liu, *Nat. Rev. Genet.* **2015**, *16*, 379–394.

[4] 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.

[5] M. H. Serres, S. Gopal, L. A. Nahum, P. Liang, T. Gaasterland, M. Riley, *Genome Biol.* **2001**, *2*, Research0035.

[6] A. Vogel, O. May, *Industrial Enzyme Applications*, John Wiley & Sons, **2019**.

[7] E. C. Campbell, G. J. Correy, P. D. Mabbitt, A. M. Buckle, N. Tokuriki, C. J. Jackson, *Curr. Opin. Struct. Biol.* **2018**, *50*, 49–57.

[8] R. M. Crean, J. M. Gardner, S. C. L. Kamerlin, *J. Am. Chem. Soc.* **2020**, *142*, 11324–11342.

[9] X. Zhang, K. N. Houk, *Acc. Chem. Res.* **2005**, *38*, 379–385.

[10] E. K. Keenan, D. K. Zachman, M. D. Hirschey, *Mol. Cell* **2021**, *81*, 1868–1878.

[11] A. Ambrogelly, S. Palioura, D. Söll, *Nat. Chem. Biol.* **2007**, *3*, 29–35.

[12] D. Gfeller, O. Michielin, V. Zoete, *Nucleic Acids Res.* **2021**, *41*, D327–D332.

[13] Y. Li, P. A. Dalby, *Biosci. Rep.* **2022**, *42*, BSR20220168.

[14] A. J. Link, M. L. Mock, D. A. Tirrell, *Curr. Opin. Biotechnol.* **2003**, *14*, 603–609.

[15] I. Drienovská, G. Roelfes, *Nat. Catal.* **2020**, *3*, 193–202.

[16] Y. Cui, J. Sun, B. Wu, *Trends Chem.* **2022**, *4*, 409–419.

[17] X. Pan, T. Kortemme, *J. Biol. Chem.* **2021**, *296*, 100558.

[18] W. M. Dawson, G. G. Rhys, D. N. Woolfson, *Curr. Opin. Chem. Biol.* **2019**, *52*, 102–111.

[19] P. B. Cox, R. Gupta, *ACS Med. Chem. Lett.* **2022**, *13*, 1016–1029.

[20] S. Dara, S. Dhamercherla, S. S. Jadav, C. H. M. Babu, M. J. Ahsan, *Artificial Intelligence Rev.* **2022**, *55*, 1947–1999.

[21] Y. Jiang, S. S. Neti, I. Sitarik, P. Pradhan, P. To, Y. Xia, S. D. Fried, S. J. Booker, E. P. O'Brien, *Nat. Chem.* **2023**, *15*, 308–318.

[22] Y. Liu, Q. Yang, F. Zhao, *Annu. Rev. Biochem.* **2021**, *90*, 375–401.

[23] K.-C. Chou, *Int. J. Pept. Res. Ther.* **2020**, *26*, 873–888.

[24] M. Punta, Y. Ofran, *PLoS Comput. Biol.* **2008**, *4*, e1000160.

[25] P. Radivojac, W. T. Clark, T. R. Oron, A. M. Schnoes, T. Wittkop, A. Sokolov, K. Graim, C. Funk, K. Verspoor, A. Ben-Hur, G. Pandey, J. M. Yunes, A. S. Talwalkar, S. Repo, M. L. Souza, D. Piovesan, R. Casadio, Z. Wang, J. Cheng, H. Fang, J. Gough, P. Koskinen, P. Törönen, J. Nokso-Koivisto, L. Holm, D. Cozzetto, D. W. A. Buchan, K. Bryson, D. T. Jones, B. Limaye, H. Inamdar, A. Datta, S. K. Manjari, R. Joshi, M. Chitale, D. Kihara, A. M. Lisewski, S. Erdin, E. Venner, O. Lichtarge, R. Rentzsch, H. Yang, A. E. Romero, P. Bhat, A. Paccanaro, T. Hamp, R. Kaßner, S. Seemayer, E. Vicedo, C. Schaefer, D. Achten, F. Auer, A. Boehm, T. Braun, M. Hecht, M. Heron, P. Hönigschmid, T. A. Hopf, S. Kaufmann, M. Kiening, D. Krompass, C. Landerer, Y. Mahlich, M. Roos, J. Björne, T. Salakoski, A. Wong, H. Shatkay, F. Gatzmann, I. Sommer, M. N. Wass, M. J. E. Sternberg, N. Škunca, F. Supek, M. Bošnjak, P. Panov, S. Džeroski, T. Šmuc, Y. A. I. Kourmpetis, A. D. J. van Dijk, C. J. F. t Braak, Y. Zhou, Q. Gong, X. Dong, W. Tian, M. Falda, P. Fontana, E. Lavezzo, B. Di Camillo, S. Toppo, L. Lan, N. Djuric, Y. Guo, S. Vucetic, A. Bairoch, M. Linial, P. C. Babbitt, S. E. Brenner, C. Orengo, B. Rost, et al., *Nat. Methods* **2013**, *10*, 221–227.

[26] C. Zhang, P. L. Freddolino, Y. Zhang, *Nucleic Acids Res.* **2017**, *45*, W291–W299.

[27] L. Xu, S. Jiang, J. Wu, Q. Zou, *Briefings Bioinf.* **2021**, *22*, 1–7.

[28] H. H. Lin, L. Y. Han, H. L. Zhang, C. J. Zheng, B. Xie, Z. W. Cao, Y. Z. Chen, *BMC Bioinf.* **2006**, *7*, S13.

[29] H. M. Geertz-Hansen, N. Blom, A. M. Feist, S. Brunak, T. N. Petersen, *Proteins Struct. Funct. Bioinf.* **2014**, *82*, 1819–1828.

[30] J. P. G. L. M. Rodrigues, A. S. J. Melquiond, E. Karaca, M. Trellet, M. van Dijk, G. C. P. van Zundert, C. Schmitz, S. J. de Vries, A. Bordogna, L. Bonati, P. L. Kastritis, A. M. J. J. Bonvin, *Proteins Struct. Funct. Bioinf.* **2013**, *81*, 2119–2128.

[31] Y. Hashem, P. Auffinger, *Methods* **2009**, *47*, 187–197.

[32] K. R. Acharya, M. D. Lloyd, *Trends Pharmacol. Sci.* **2005**, *26*, 10–14.

[33] J. Kirchmair, P. Markt, S. Distinto, D. Schuster, G. M. Spitzer, K. R. Liedl, T. Langer, G. Wolber, *J. Med. Chem.* **2008**, *51*, 7021–7040.

[34] A. Wlodawer, Z. Dauter, P. J. Porebski, W. Minor, R. Stanfield, M. Jaskolski, E. Pozharski, C. X. Weichenberger, B. Rupp, *FEBS J.* **2018**, *285*, 444–466.

[35] H. Liu, Y. Jin, H. Ding, *Briefings Bioinf.* **2023**, *24*, bbad057.

[36] G. Martínez-Rosell, T. Giorgino, G. De Fabritiis, *J. Chem. Inf. Model.* **2017**, *57*, 1511–1516.

[37] H. Land, M. S. Humble, *Protein Engineering: Methods Protocols* **2018**, 43–67.

[38] M. J. Abraham, T. Murtola, R. Schulz, S. Páll, J. C. Smith, B. Hess, E. Lindahl, *SoftwareX* **2015**, *1–2*, 19–25.

[39] R. P. Joosten, F. Long, G. N. Murshudov, A. Perrakis, *IUCrJ* **2014**, *1*, 213–220.

[40] H. Ashkenazy, S. Abadi, E. Martz, O. Chay, I. Mayrose, T. Pupko, N. Ben-Tal, *Nucleic Acids Res.* **2016**, *44*, W344–W350.

[41] A. Ben Chorin, G. Masrati, A. Kessel, A. Narunsky, J. Sprinzak, S. Lahav, H. Ashkenazy, N. Ben-Tal, *Protein Sci.* **2020**, *29*, 258–267.

[42] F. Glaser, R. J. Morris, R. J. Najmanovich, R. A. Laskowski, J. M. Thornton, *Proteins Struct. Funct. Bioinf.* **2006**, *62*, 479–488.

[43] M. Baek, F. DiMaio, I. Anishchenko, J. Dauparas, S. Ovchinnikov, G. R. Lee, J. Wang, Q. Cong, L. N. Kinch, R. D. Schaeffer, C. Millán, H. Park, C. Adams, C. R. Glassman, A. DeGiovanni, J. H. Pereira, A. V. Rodrigues, A. A. van Dijk, A. C. Ebrecht, D. J. Opperman, T. Sagmeister, C. Buhlheller, T. Pavkov-Keller, M. K. Rathinaswamy, U. Dalwadi, C. K. Yip, J. E. Burke, K. C. Garcia, N. V. Grishin, P. D. Adams, R. J. Read, D. Baker, *Science* **2021**, *373*, 871–876.

[44] A. Elofsson, *Curr. Opin. Struct. Biol.* **2023**, *80*, 102594.

[45] P. Bryant, *Curr. Opin. Struct. Biol.* **2023**, *79*, 102529.

[46] M. Akdel, D. E. V. Pires, E. P. Pardo, J. Jänes, A. O. Zalevsky, B. Mészáros, P. Bryant, L. L. Good, R. A. Laskowski, G. Pozzati, A. Shenoy, W. Zhu, P. Kundrotas, V. R. Serra, C. H. M. Rodrigues, A. S. Dunham, D. Burke, N. Borkakoti, S. Velankar, A. Frost, J. Basquin, K. Lindorff-Larsen, A. Bateman, A. V. Kajava, A. Valencia, S. Ovchinnikov, J. Durairaj, D. B. Ascher, J. M. Thornton, N. E. Davey, A. Stein, A. Elofsson, T. I. Croll, P. Beltrao, *Nat. Struct. Mol. Biol.* **2022**, *29*, 1056–1067.

[47] D. del Alamo, D. Sala, H. S. McHaourab, J. Meiler, *eLife* **2022**, *11*, e75751.

[48] R. A. Stein, H. S. McHaourab, *PLoS Comput. Biol.* **2022**, *18*, e1010483.

[49] G. Casadevall, C. Duran, M. Estévez-Gay, S. Osuna, *Protein Sci.* **2022**, *31*, e4426.

[50] H. K. Wayment-Steele, S. Ovchinnikov, L. Colwell, D. Kern, *bioRxiv preprint* **2022**, 10.1101/2022.1110.1117.512570.

[51] H.-B. Guo, A. Perminov, S. Bekele, G. Kedziora, S. Farajollahi, V. Varaljay, K. Hinkle, V. Molinero, K. Meister, C. Hung, P. Dennis, N. Kelley-Loughnane, R. Berry, *Sci. Rep.* **2022**, *12*, 10696.

[52] P. Ma, D.-W. Li, R. Brüschweiler, *Proteins Struct. Funct. Bioinf.* **2023**, *91*, 847–855.

[53] I. Anishchenko, S. J. Pellock, T. M. Chidyausiku, T. A. Ramelot, S. Ovchinnikov, J. Hao, K. Bafna, C. Norn, A. Kang, A. K. Bera, F. DiMaio, L. Carter, C. M. Chow, G. T. Montelione, D. Baker, *Nature* **2021**, *600*, 547–552.

[54] S. V. Torres, P. J. Y. Leung, I. D. Lutz, P. Venkatesh, J. L. Watson, F. Hink, H.-H. Huynh, A. H.-W. Yeh, D. Juergens, N. R. Bennett, A. N. Hoofnagle, E. Huang, M. J. MacCoss, M. Expòsit, G. R. Lee, P. M. Levine, X. Li, M. Lamb, E. N. Korkmaz, J. Nivala, L. Stewart, J. M. Rogers, D. Baker, *bioRxiv Preprint* **2022**, 10.1101/2022.1112.1110.519862.

[55] Y. Zhang, P. Li, F. Pan, H. Liu, P. Hong, X. Liu, J. Zhang, *bioRxiv Preprint* **2021**, 10.1101/2021.1111.1103.467194.

[56] 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.

[57] G. R. Buel, K. J. Walters, *Nat. Struct. Mol. Biol.* **2022**, *29*, 1–2.

[58] M. Varadi, S. Anyango, M. Deshpande, S. Nair, C. Natassia, G. Yordanova, D. Yuan, O. Stroe, G. Wood, A. Laydon, A. Žídek, 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.* **2021**, *50*, D439–D444.

[59] 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 preprint* **2022**, 10.1101/2021.1110.1104.463034.

[60] E. Coudert, S. Gehant, E. de Castro, M. Pozzato, D. Baratin, T. Neto, C. J. A. Sigrist, N. Redaschi, A. Bridge, T. U. Consortium, *Bioinformatics* **2022**, *39*, btac793.

[61] D. Jakubec, P. Skoda, R. Krivak, M. Novotny, D. Hoksza, *Nucleic Acids Res.* **2022**, *50*, W593-W597.

[62] J. Tubiana, D. Schneidman-Duhovny, H. J. Wolfson, *Nat. Methods* **2022**, *19*, 730–739.

[63] V. Le Guilloux, P. Schmidtke, P. Tuffery, *BMC Bioinf.* **2009**, *10*, 168.

[64] Y. Yuan, J. Pei, L. Lai, *Curr. Pharm. Des.* **2013**, *19*, 2326–2333.

[65] Y. Xu, S. Wang, Q. Hu, S. Gao, X. Ma, W. Zhang, Y. Shen, F. Chen, L. Lai, J. Pei, *Nucleic Acids Res.* **2018**, *46*, W374–W379.

[66] S. Wang, H. Lin, Z. Huang, Y. He, X. Deng, Y. Xu, J. Pei, L. Lai, *Biomol. Eng.* **2022**, *12*, 967.

[67] J. Sim, S. Kwon, C. Seok, *Nucleic Acids Res.* **2022**, *51*, D403–D408.

[68] J.-E. Sánchez-Aparicio, L. Tiessler-Sala, L. Velasco-Carneros, L. Roldán-Martín, G. Sciortino, J.-D. Maréchal, *J. Chem. Inf. Model.* **2021**, *61*, 311–323.

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

[70] M. L. Mascotti, M. Juri Ayub, N. Furnham, J. M. Thornton, R. A. Laskowski, *J. Mol. Biol.* **2016**, *428*, 3131–3146.

[71] I. B. Kuznetsov, Z. Gou, R. Li, S. Hwang, *Proteins Struct. Funct. Bioinf.* **2006**, *64*, 19–27.

[72] N. Abdollahi, S. A. M. Tonekaboni, J. Huang, B. Wang, S. MacKinnon, *arXiv preprint* **2023**, 10.48550/arXiv.41711.00533.

[73] N. S. Pagadala, K. Syed, J. Tuszynski, *Biophys. Rev. Lett.* **2017**, *9*, 91–102.

[74] Z. Wang, H. Sun, X. Yao, D. Li, L. Xu, Y. Li, S. Tian, T. Hou, *Phys. Chem. Chem. Phys.* **2016**, *18*, 12964–12975.

[75] G. Corso, H. Stärk, B. Jing, R. Barzilay, T. Jaakkola, *arXiv preprint* **2022**, 10.48550/arXiv.42210.01776.

[76] S. J. Wodak, S. Vajda, M. F. Lensink, D. Kozakov, P. A. Bates, *Annu. Rev. Biophys.* **2023**, *52*, 7.1–7.24.

[77] G. Weng, J. Gao, Z. Wang, E. Wang, X. Hu, X. Yao, D. Cao, T. Hou, *J. Chem. Theory Comput.* **2020**, *16*, 3959–3969.

[78] C. Dominguez, R. Boelens, A. M. J. J. Bonvin, *J. Am. Chem. Soc.* **2003**, *125*, 1731–1737.

[79] P. I. Koukos, J. Roel-Touris, F. Ambrosetti, C. Geng, J. Schaarschmidt, M. E. Trellet, A. S. J. Melquiond, L. C. Xue, R. V. Honorato, I. Moreira, Z. Kurkcuoglu, A. Vangone, A. M. J. J. Bonvin, *Proteins Struct. Funct. Bioinf.* **2020**, *88*, 1029–1036.

[80] J. K. Leman, B. D. Weitzner, P. D. Renfrew, S. M. Lewis, R. Moretti, A. M. Watkins, V. K. Mulligan, S. Lyskov, J. Adolf-Bryfogle, J. W. Labonte, *PLoS Comput. Biol.* **2020**, *16*, e1007507.

[81] N. London, B. Raveh, E. Cohen, G. Fathi, O. Schueler-Furman, *Nucleic Acids Res.* **2011**, *39*, W249–W253.

[82] N. Bloodworth, N. R. Barbaro, R. Moretti, D. G. Harrison, J. Meiler, *PLoS One* **2022**, *17*, e0275759.

[83] C. Christoffer, D. Kihara, *Proteomics* **2022**, e2200322.

[84] B. Jiménez-García, J. Roel-Touris, M. Romero-Durana, M. Vidal, D. Jiménez-González, J. Fernández-Recio, *Bioinformatics* **2017**, *34*, 49–55.

[85] R. Esmaeeli, A. Bauzá, A. Perez, *Nucleic Acids Res.* **2023**, *51*, 1625–1636.

[86] M. Baek, R. McHugh, I. Anishchenko, D. Baker, F. DiMaio, *bioRxiv preprint* **2022**, 10.1101/2022.1109.1109.507333.

[87] U. F. Röhrig, A. Grosdidier, V. Zoete, O. Michielin, *J. Comput. Chem.* **2009**, *30*, 2305–2315.

[88] F. Bai, S. Liao, J. Gu, H. Jiang, X. Wang, H. Li, *J. Chem. Inf. Model.* **2015**, *55*, 833–847.

[89] A. A. Adeniyi, M. E. S. Soliman, *Drug Discovery Today* **2017**, *22*, 1216–1223.

[90] J. Rodríguez-Guerra Pedregal, G. Sciortino, J. Guasp, M. Municoy, J.-D. Maréchal, *J. Comput. Chem.* **2017**, *38*, 2118–2126.

[91] L. Riccardi, V. Genna, M. D. Vivo, *Nat. Chem. Rev.* **2018**, *2*, 100–112.

[92] A. Fischer, M. Smieško, M. Sellner, M. A. Lill, *J. Med. Chem.* **2021**, *64*, 2489–2500.

[93] M. A. Maria-Solano, E. Serrano-Hervás, A. Romero-Rivera, J. Iglesias-Fernández, S. Osuna, *Chem. Commun.* **2018**, *54*, 6622–6634.

[94] M. C. Childers, V. Daggett, *Mol. Syst. Des. Eng.* **2017**, *2*, 9–33.

[95] M. Rouhani, F. Khodabakhsh, D. Norouzian, R. A. Cohan, V. Valizadeh, *J. Mol. Graphics Modell.* **2018**, *84*, 43–53.

[96] A. Liwo, C. Czaplewski, A. K. Sieradzan, A. G. Lipska, S. A. Samsonov, R. K. Murarka, *Biomol. Eng.* **2021**, *11*, 1347.

[97] P. C. T. Souza, R. Alessandri, J. Barnoud, S. Thallmair, I. Faustino, F. Grünewald, I. Patmanidis, H. Abdizadeh, B. M. H. Bruininks, T. A. Wassenaar, P. C. Kroon, J. Melcr, V. Nieto, V. Corradi, H. M. Khan, J. Domański, M. Javanainen, H. Martinez-Seara, N. Reuter, R. B. Best, I. Vattulainen, L. Monticelli, X. Periole, P. Tieleman, A. H. de Vries, S. J. Marrink, *Nat. Methods* **2021**, *18*, 382–388.

[98] T. E. Ouldridge, A. A. Louis, J. P. Doye, *Phys. Rev. Lett.* **2010**, *104*, 178101.

[99] P. Sulc, F. Romano, T. Ouldridge, J. Doye, A. Louis, *Biophys. J.* **2017**, *112*, 369a.

[100] A. Davtyan, N. P. Schafer, W. Zheng, C. Clementi, P. G. Wolynes, G. A. Papoian, *J. Phys. Chem. B* **2012**, *116*, 8494–8503.

[101] S. Sacquin-Mora, *Mol. Simul.* **2014**, *40*, 229–236.

[102] K. M. Wilding, A. K. Smith, J. W. Wilkerson, D. B. Bush, T. A. I. V. Knotts, B. C. Bundy, *ACS Synth. Biol.* **2018**, *7*, 510–521.

[103] A. M. Saleh, K. M. Wilding, S. Calve, B. C. Bundy, T. L. Kinzer-Ursem, *J. Biol. Engineering* **2019**, *13*, 43.

[104] P. C. T. Souza, S. Thallmair, P. Conflitti, C. Ramírez-Palacios, R. Alessandri, S. Raniolo, V. Limongelli, S. J. Marrink, *Nat. Commun.* **2020**, *11*, 3714.

[105] F. M. Sousa, L. M. P. Lima, C. Arnarez, M. M. Pereira, M. N. Melo, *J. Chem. Inf. Model.* **2021**, *61*, 335–346.

[106] R. Alessandri, J. Barnoud, A. S. Gertsen, I. Patmanidis, A. H. de Vries, P. C. T. Souza, S. J. Marrink, *Adv. Theory Simulations* **2022**, *5*.

[107] V. Vennelakanti, A. Nazemi, R. Mehmood, A. H. Steeves, H. J. Kulik, *Curr. Opin. Struct. Biol.* **2022**, *72*, 9–17.

[108] H. Lin, D. G. Truhlar, *Theor. Chem. Acc.* **2007**, *117*, 185–199.

[109] H. M. Senn, W. Thiel, *Angew. Chem. Int. Ed.* **2009**, *48*, 1198–1229.

[110] U. Ryde, in *Methods in Enzymology*, Vol. 577 (Ed.: G. A. Voth), Academic Press, **2016**, pp. 119–158.

[111] R. P. Magalhães, H. S. Fernandes, S. F. Sousa, *Isr. J. Chem.* **2020**, *60*, 655–666.

[112] D. Berta, P. J. Buigues, M. Badaoui, E. Rosta, *Curr. Opin. Struct. Biol.* **2020**, *61*, 198–206.

[113] A. Ghosh, *J. Biol. Inorg. Chem.* **2006**, *11*, 712–724.

[114] M. Radoń, *Phys. Chem. Chem. Phys.* **2019**, *21*, 4854–4870.

[115] T. Borowski, M. Quesne, M. Szaleniec, in *Advances in Protein Chemistry and Structural Biology*, Vol. 100 (Ed.: T. Karabencheva-Christova), Academic Press, **2015**, pp. 187–224.

[116] S. F. Sousa, P. A. Fernandes, M. J. Ramos, *Phys. Chem. Chem. Phys.* **2012**, *14*, 12431–12441.

[117] W. F. van Gunsteren, H. J. C. Berendsen, *Angew. Chem. Int. Ed.* **1990**, *29*, 992–1023.

[118] O. Guvench, A. D. J. MacKerell, in *Molecular Modeling of Proteins*, Vol. 443 (Ed.: A. Kukol), Humana Press, **2008**, pp. 63–88.

[119] S. Riniker, *J. Chem. Inf. Model.* **2018**, *58*, 565–578.

[120] J. Wang, R. M. Wolf, J. W. Caldwell, P. A. Kollman, D. A. Case, *J. Comput. Chem.* **2004**, *25*, 1157–1174.

[121] K. Vanommeslaeghe, E. Hatcher, C. Acharya, S. Kundu, S. Zhong, J. Shim, E. Darian, O. Guvench, P. Lopes, I. Vorobyov, A. D. Mackerel, *J. Comput. Chem.* **2009**, *31*, 671–690.

[122] G. A. Kaminski, R. A. Friesner, *J. Phys. Chem. B* **2001**, *105*, 6474–6487.

[123] N. Schmid, A. P. Eichenberger, A. Choutko, S. Riniker, M. Winger, A. E. Mark, W. F. van Gunsteren, *Eur. Biophys. J.* **2011**, *40*, 843–856.

[124] J. W. Ponder, D. A. Case, in *Advances in Protein Chemistry: Protein Simulations*, Vol. 66 (Ed.: V. Daggett), Academic Press, Cambridge, Massachusetts, **2003**, pp. 27–85.

[125] J. Rydzewski, W. Nowak, *Phys. Life Rev.* **2017**, *22–23*, 58–74.

[126] D. Perez, B. P. Uberuaga, Y. Shim, J. G. Amar, A. F. Voter, in *Annual Reports in Computational Chemistry*, Vol. 5 (Ed.: R. A. Wheeler), Elsevier, **2009**, pp. 79–98.

[127] A. Barducci, M. Bonomi, M. Parrinello, *WIREs Comput. Mol. Sci.* **2011**, *1*, 826–843.

[128] R. Lazim, D. Suh, S. Choi, *Int. J. Mol. Sci.* **2020**, *21*, 6339.

[129] E. Jurrus, D. Engel, K. Star, K. Monson, J. Brandi, L. E. Felberg, D. H. Brookes, L. Wilson, J. Chen, K. Liles, M. Chun, P. Li, D. W. Gohara, T. Dolinsky, R. Konecny, D. R. Koes, J. E. Nielsen, T. Head-Gordon, W. Geng, R. Krasny, G. W. Wei, M. J. Holst, J. A. McCammon, N. A. Baker, *Protein Sci.* **2018**, *27*, 112–128.

[130] M. Fixman, *J. Chem. Phys.* **1979**, *70*, 4995–5005.

[131] E. Krieger, J. E. Nielsen, C. A. E. M. Spronk, G. Vriend, *J. Mol. Graphics Modell.* **2006**, *25*, 481–486.

[132] C. R. Søndergaard, M. H. M. Olsson, M. Rostkowski, J. H. Jensen, *J. Chem. Theory Comput.* **2011**, *7*, 2284–2295.

[133] Y.-m. M Huang, W. You, B. G. Caulkins, M. F. Dunn, L. J. Mueller, C.-e. A Chang, *Protein Sci.* **2016**, *25*, 166–183.

[134] A. B. Shimon, D. E. Shalev, M. Y. Niv, *Curr. Pharm. Des.* **2013**, *19*, 4173–4181.

[135] F.-Y. Lin, J. Huang, P. Pandey, C. Rupakheti, J. Li, B. t Roux, A. D. MacKerell Jr., *J. Chem. Theory Comput.* **2020**, *16*, 3221–3239.

[136] Y. Shi, Z. Xia, J. Zhang, R. Best, C. Wu, J. W. Ponder, P. Ren, *J. Chem. Theory Comput.* **2013**, *9*, 4046–4063.

[137] Z. Jing, C. Liu, S. Y. Cheng, R. Qi, B. D. Walker, J.-P. Piquemal, P. Ren, *Annu. Rev. Biophys.* **2019**, *48*, 371–394.

[138] W. L. Jorgensen, J. Tirado-Rives, *J. Comput. Chem.* **2005**, *26*, 1689–1700.

[139] M. J. Frisch, G. W. Trucks, H. B. Schlegel, G. E. Scuseria, M. A. Robb, J. R. Cheeseman, G. Scalmani, V. Barone, G. A. Petersson, H. Nakatsuji, X. Li, M. Caricato, A. V. Marenich, J. Bloino, B. G. Janesko, R. Gomperts, B. Mennucci, H. P. Hratchian, J. V. Ortiz, A. F. Izmaylov, J. L. Sonnenberg, Williams, F. Ding, F. Lipparini, F. Egidi, J. Goings, B. Peng, A. Petrone, T. Henderson, D. Ranasinghe, V. G. Zakrzewski, J. Gao, N. Rega, G. Zheng, W. Liang, M. Hada, M. Ehara, K. Toyota, R. Fukuda, J. Hasegawa, M. Ishida, T. Nakajima, Y. Honda, O. Kitao, H. Nakai, T. Vreven, K. Throssell, J. A. Montgomery Jr., J. E. Peralta, F. Ogliaro, M. J. Bearpark, J. J. Heyd, E. N. Brothers, K. N. Kudin, V. N. Staroverov, T. A. Keith, R. Kobayashi, J. Normand, K. Raghavachari, A. P. Rendell, J. C. Burant, S. S. Iyengar, J. Tomasi, M. Cossi, J. M. Millam, M. Klene, C. Adamo, R. Cammi, J. W. Ochterski, R. L. Martin, K. Morokuma, O. Farkas, J. B. Foresman, D. J. Fox, Wallingford, CT, **2016**.

[140] C. I. Bayly, P. Cieplak, W. D. Cornell, P. A. Kollman, *J. Phys. Chem.* **1993**, *97*, 10269–10280.

[141] D. van der Spoel, *Curr. Opin. Struct. Biol.* **2021**, *67*, 18–24.

[142] L. Hu, U. Ryde, *J. Chem. Theory Comput.* **2011**, *7*, 2452–2463.

[143] F.-Y. Dupradeau, C. Cezard, R. Lelong, E. Stanislawiak, J. Pecher, J. C. Delepine, P. Cieplak, *Nucleic Acids Res.* **2007**, *36*, D360–D367.

[144] W. Thiel, *WIREs Comput. Mol. Sci.* **2013**, *4*, 91–168.

[145] J. J. P. Stewart, *J. Mol. Model.* **2012**, *19*, 1–32.

[146] H. Zhou, E. Tajkhorshid, T. Frauenheim, S. Suhai, M. Elstner, *Chem. Phys.* **2002**, *277*, 91–103.

[147] J. L. Knight, C. L. Brooks, *J. Chem. Theory Comput.* **2009**, *5*, 1680–1691.

[148] K. Vanommeslaeghe, E. P. Raman, A. D. J. MacKerell, *J. Chem. Inf. Model.* **2012**, *52*, 3155–3168.

[149] A. W. S. da Silva, W. F. Vranken, *BMC Res. Notes* **2012**, *5*, 367.

[150] A. W. Schuttelkopf, D. M. F. van Aalten, *Acta Crystallogr. Sect. D* **2004**, *60*, 1355–1363.

[151] K. B. Koziara, M. Stroet, A. K. Malde, A. E. Mark, *J. Comput.-Aided Mol. Des.* **2014**, *28*, 221–233.

[152] L. S. Dodda, I. C. de Vaca, J. Tirado-Rives, W. L. Jorgensen, *Nucleic Acids Res.* **2017**, *45*, W331–W336.

[153] J. A. Lemkul, W. J. Allen, D. R. Bevan, *J. Chem. Inf. Model.* **2010**, *50*, 2221–2235.

[154] A. Perez, I. Marchan, D. Svozil, J. Sponer, T. E. Cheatham, C. A. Laughton, M. Orozco, *Biophys. J.* **2007**, *92*, 3817–3829.

[155] I. Ivani, P. D. Dans, A. Noy, A. Pérez, I. Faustino, A. Hospital, J. Walther, P. Andrio, R. Goñi, A. Balaceanu, G. Portella, F. Battistini, J. L. Gelpí, C. González, M. Vendruscolo, C. A. Laughton, S. A. Harris, D. A. Case, M. Orozco, *Nature Methods*, **2016**, *13*, 55–58.

[156] R. Galindo-Murillo, J. C. Robertson, M. Zgarbová, J. I. Šponer, M. Otyepka, P. Jurečka, T. E. Cheatha, *J. Chem. Theory Comput.* **2016**, *12*, 4114–4127.

[157] K. N. Kirschner, A. B. Yongye, S. M. Tschampel, J. González-Outeiriño, C. R. Daniels, B. L. Foley, R. J. Woods, *J. Comput. Chem.* **2008**, *29*, 622–655.

[158] A. Plazinska, W. Plazinski, *J. Chem. Theory Comput.* **2021**, *17*, 2575–2585.

[159] J. I. Vymětal, V. Jurásková, J. I. Vondrás, *J. Chem. Theory Comput.* **2019**, *15*, 665–679.

[160] A. Croitoru, S.-J. Park, A. Kumar, J. Lee, W. Im, A. D. J. A. MacKerell, *J. Chem. Theory Comput.* **2021**, *17*, 3554–3570.

[161] G. A. Khoury, J. P. Thompson, J. Smadbeck, C. A. Kieslich, C. A. Floudas, *J. Chem. Theory Comput.* **2013**, *9*, 5653–5674.

[162] J. J. Pavelites, J. Gao, P. A. Bash, A. D. J. MacKerell, *J. Comput. Chem.* **1997**, *18*, 221–239.
[163] S. Zhu, *J. Chem. Inf. Model.* **2019**, *59*, 4239–4247.
[164] A. Bunker, P. T. Männistö, J.-F. St Pierre, T. Róg, P. Pomorski, L. Stimson, M. Karttune, *SAR QSAR Environ. Res.* **2008**, *19*, 179–189.
[165] D. A. Saez, E. Vohringer-Martinez, *J. Comput.-Aided Mol. Des.* **2015**, *29*, 951–961.
[166] D. F. A. R. Dourado, S. Pohle, A. T. P. Carvalho, D. S. Dheeman, J. M. Caswell, T. Skvortsov, I. Miskelly, R. T. Brown, D. J. Quinn, C. C. R. Allen, K. Leonid, M. Huang, T. S. Moody, *ACS Catal.* **2016**, *6*, 7749–7759.
[167] E. Sjulstok, I. A. Solovyov, P. L. Freddolino, in *Methods in Enzymology*, Vol. 602 (Ed.: B. A. Palfey), Academic Press, Cambridge, Massachusetts, **2019**, pp. 277–314.
[168] R. K. Kar, A.-F. Miller, M.-A. Mroginski, *WIREs Comput. Mol. Sci.* **2022**, *12*, e1541.
[169] J. Antony, D. M. Medvedev, A. A. Stuchebrukhov, *J. Am. Chem. Soc.* **2000**, *122*, 1057–1065.
[170] C. Schneider, J. Suhnel, *Biopolymers* **1999**, *50*, 287–302.
[171] M. J. Robertson, J. Tirado-Rives, W. L. Jorgensen, *J. Phys. Chem. Lett.* **2016**, *7*, 3032–3036.
[172] A. Aleksandrov, *J. Comput. Chem.* **2019**, *40*, 2834–2842.
[173] P. Li, K. M. J. Merz, *Chem. Rev.* **2017**, *117*, 1564–1686.
[174] P. Li, B. P. Roberts, D. K. Chakravorty, K. M. J. Merz, *J. Chem. Theory Comput.* **2013**, *9*, 2733–2748.
[175] J. Andrys, J. Heider, T. Borowski, *Theor. Chem. Acc.* **2021**, *140*, 115.
[176] F. Duarte, P. Bauer, A. Barrozo, B. A. Amrein, M. Purg, J. Åqvist, S. C. L. Kamerlin, *J. Phys. Chem. B* **2014**, *118*, 4351–4362.
[177] S.-Y. Lu, Z.-M. Huang, W.-K. Huang, X.-Y. Liu, Y.-Y. Chen, T. Shi, J. Zhang, *Proteins Struct. Funct. Bioinf.* **2012**, *81*, 729–917.
[178] Y. Jiang, H. Zhang, W. Feng, T. Tan, *J. Chem. Inf. Model.* **2015**, *55*, 2575–2586.
[179] Q. Liao, A. Pabis, B. Strodel, S. C. L. Kamerlin, *J. Phys. Chem. Lett.* **2017**, *8*, 5408–5414.
[180] M. B. Peters, Y. Yang, B. Wang, L. Fusti-Molnar, M. N. Weaver, K. M. J. Merz, *J. Chem. Theory Comput.* **2010**, *6*, 2935–2947.
[181] Z. Yu, P. Li, K. M. J. Merz, *J. Chem. Theory Comput.* **2018**, *14*, 242–254.
[182] A. Oda, N. Yamaotsu, S. Hirono, *J. Comput. Chem.* **2005**, *26*, 755–863.
[183] S. Adam, M. Knapp-Mohammady, J. Yi, A.-N. Bondar, *J. Comput. Chem.* **2018**, *39*, 7–20.
[184] K. Shahrokh, A. Orendt, G. S. Yost, T. E. Cheatham, *J. Comput. Chem.* **2011**, *33*, 119–133.
[185] P. Li, K. M. J. Merz, *J. Chem. Inf. Model.* **2015**, *56*, 599–604.
[186] S. Zheng, Q. Tang, J. He, S. Du, S. Xu, C. Wang, Y. Xu, F. Lin, *J. Chem. Inf. Model.* **2016**, *56*, 811–818.
[187] B. Kuhlman, G. Dantas, G. C. Ireton, G. Varani, B. L. Stoddard, D. Baker, *Science* **2003**, *302*, 1364–1368.
[188] R. F. Alford, A. Leaver-Fay, J. R. Jeliazkov, M. J. O'Meara, F. P. DiMaio, H. Park, M. V. Shapovalov, P. D. Renfrew, V. K. Mulligan, K. Kappel, J. W. Labonte, M. S. Pacella, R. Bonneau, P. Bradley, R. L. Dunbrack Jr., R. Das, D. Baker, B. Kuhlman, T. Kortemme, J. J. Gray, *J. Chem. Theory Comput.* **2017**, *13*, 3031–3048.
[189] R. F. Alford, A. Leaver-Fay, J. R. Jeliazkov, M. J. O'Meara, F. P. DiMaio, H. Park, M. V. Shapovalov, P. D. Renfrew, V. K. Mulligan, K. Kappel, J. W. Labonte, M. S. Pacella, R. Bonneau, P. Bradley, R. L. Dunbrack Jr., R. Das, D. Baker, B. Kuhlman, T. Kortemme, J. J. Gray, *J. Chem. Theory Comput.* **2022**, *18*, 4594–4594.
[190] G. Dantas, M. Wong, B. Kuhlman, D. Baker, *J. Mol. Biol.* **2003**, *332*, 449–460.
[191] G. Ahdritz, N. Bouatta, S. Kadyan, Q. Xia, W. Gerecke, T. J. O'Donnell, D. Berenberg, I. Fisk, N. Zanichelli, B. Zhang, A. Nowaczynski, B. Wang, M. M. Stepniewska-Dziubinska, S. Zhang, A. Ojewole, M. E. Guney, S. Biderman, A. M. Watkins, S. Ra, P. R. Lorenzo, L. Nivon, B. Weitzner, Y.-E. A. Ban, P. K. Sorger, E. Mostaque, Z. Zhang, R. Bonneau, M. AlQuraishi, *bioRxiv preprint* **2022**, 10.1101/2022.1111.1120.517210.
[192] 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* **2023**, *379*, 1123–1130.
[193] 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 preprint* **2022**, 10.1101/2022.1107.1121.500999.
[194] M. L. Verdonk, J. C. Cole, M. J. Hartshorn, C. W. Murray, R. D. Taylor, *Proteins Struct. Funct. Bioinf.* **2003**, *52*, 609–623.
[195] Y. Liu, B. Kuhlman, *Nucleic Acids Res.* **2006**, *34*, W235–W238.
[196] F. Richter, A. Leaver-Fay, S. D. Khare, S. Bjelic, D. Baker, *PLoS One* **2011**, *6*, e19230.
[197] A. Broom, R. V. Rakotoharisoa, M. C. Thompson, N. Zarifi, E. Nguyen, N. Mukhametzhanov, L. Liu, J. S. Fraser, R. A. Chica, *Nat. Commun.* **2020**, *11*, 4808.
[198] H. J. Wijma, R. J. Floor, S. Bjelic, S. J. Marrink, D. Baker, D. B. Janssen, *Angew. Chem. Int. Ed.* **2015**, *54*, 3726–3730.
[199] 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.
[200] J. Bendl, J. Stourac, E. Sebestova, O. Vavra, M. Musil, J. Brezovsky, J. Damborsky, *Nucleic Acids Res.* **2016**, *44*, W479–W487.
[201] A. Romero-Rivera, M. Garcia-Borràs, S. Osuna, *ACS Catal.* **2017**, *7*, 8524–8532.
[202] K. W. Borrelli, A. Vitalis, R. Alcantara, V. Guallar, *J. Chem. Theory Comput.* **2005**, *1*, 1304–1311.
[203] A. Jurcik, D. Bednar, J. Byska, S. M. Marques, K. Furmanova, L. Daniel, P. Kokkonen, J. Brezovsky, O. Strnad, J. Stourac, A. Pavelka, M. Manak, J. Damborsky, B. Kozlikova, *Bioinformatics* **2018**, *34*, 3586–3588.
[204] O. Vavra, J. Filipovic, J. Plhak, D. Bednar, S. M. Marques, J. Brezovsky, J. Stourac, L. Matyska, J. Damborsky, *Bioinformatics* **2019**, *35*, 4986–4993.
[205] J.-E. Sánchez-Aparicio, G. Sciortino, D. V. Herrmannsdoerfer, P. Orenes Chueca, J. R.-G. Pedregal, J.-D. Maréchal, *Int. J. Mol. Sci.* **2019**, *20*, 3155.
[206] D. Devaurs, L. Bouard, M. Vaisset, C. Zanon, I. Al-Bluwi, R. Iehl, T. Simeon, J. Cortes, *Nucleic Acids Res.* **2013**, *41*, W297–W302.
[207] M. K. Nguyen, L. Jaillet, S. Redon, *J. Comput. Chem.* **2018**, *39*, 665–678.
[208] N. D. Clarke, S.-M. Yuan, *Proteins Struct. Funct. Genet.* **1995**, *23*, 256–263.
[209] B. P. Hay, T. K. Firman, *Inorg. Chem.* **2002**, *41*, 5502–5512.
[210] H. Fazelinia, P. C. Cirino, C. D. Maranas, *Protein Sci.* **2008**, *18*, 180–195.
[211] 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.
[212] K. Kaminski, J. Ludwiczak, M. Jasinski, A. Bukala, R. Madaj, K. Szczepaniak, S. Dunin-Horkawicz, *Briefings Bioinf.* **2022**, *23*, 1–11.
[213] J. Delgado, L. G. Radusky, D. Cianferoni, L. Serrano, *Struct. Bioinform.* **2019**, *35*, 4168–4169.
[214] E. H. Kellogg, A. Leaver-Fay, D. Baker, *Proteins* **2011**, *79*, 830–838.
[215] V. Parthiban, M. M. Gromiha, D. Schomburg, *Nucleic Acids Res.* **2006**, *36*, W239–W242.
[216] S. Gopi, D. Devanshu, P. Krishna, A. N. Naganathan, *Struct. Bioinform.* **2018**, *34*, 875–877.
[217] J. R. M. S. Barroso, D. Mariano, S. R. Dias, R. E. O. Rocha, L. H. Santos, R. A. P. Nagem, R. C. de Melo-Minardi, *BMC Bioinf.* **2020**, *21*, 275.
[218] P. Xiong, M. Wang, X. Zhou, T. Zhang, J. Zhang, Q. Chen, H. Liu, *Nat. Commun.* **2014**, *5*, 5330.
[219] S. Mazurenko, *ChemCatChem* **2020**, *12*, 5590–5598.
[220] L. Sumbalova, J. Stourac, T. Martinek, D. Bednar, J. Damborsky, *Nucleic Acids Res.* **2018**, *46*, W356–W362.
[221] A. Goldenzweig, M. Goldsmith, S. E. Hill, O. Gertman, P. Laurino, Y. Ashani, O. Dym, T. Unge, 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.
[222] H. J. Wijma, M. J. Fürst, D. B. Janssen, in *Protein Engineering*, Springer, **2018**, pp. 69–85.
[223] Y. Cui, Y. Chen, X. Liu, S. Dong, Y. Tian, Y. Qiao, R. Mitra, J. Han, C. Li, X. Han, W. Liu, Q. Chen, W. Wei, X. Wang, W. Du, S. Tang, H. Xiang, H. Liu, Y. Liang, K. N. Houk, B. Wu, *ACS Catal.* **2021**, *11*, 1340–1350.
[224] D. E. V. Pires, D. B. Ascher, T. L. Blundell, *Bioinformatics* **2014**, *30*, 335–342.
[225] S. Wang, H. Tang, Y. Zhao, L. Zuo, *Protein Sci.* **2022**, *31*.
[226] Y. Dehouck, A. Grosfils, B. Folch, D. Gilis, P. Bogaerts, M. Rooman, *Struct. Bioinform.* **2009**, *25*, 2537–2543.
[227] J. Tian, N. Wu, X. Chu, Y. Fan, *BMC Bioinf.* **2010**, *11*.
[228] E. Capriotti, P. Fariselli, R. Casadio, *Nucleic Acids Res.* **2005**, *33*, W306–W310.
[229] H. Cao, J. Wang, L. He, Y. Qi, J. Z. Zhang, *J. Chem. Inf. Model.* **2019**, *59*, 1508–1514.
[230] Y. Liu, L. Zhang, W. Wang, M. Zhu, C. Wang, F. Li, J. Zhang, H. Li, Q. Chen, H. Liu, *Nat. Computational Sci.* **2022**, *2*, 451–462.
[231] C.-W. Chen, J. Lin, Y.-W. Chu, *BMC Bioinf.* **2013**, *14*, S5.
[232] C. H. Rodrigues, D. E. Pires, D. B. Ascher, *Nucleic Acids Res.* **2018**, *46*, W350–W355.
[233] C. A. Voigt, C. Martinez, Z.-G. Wang, S. L. Mayo, F. H. Arnold, *Nat. Struct. Biol.* **2002**, *9*, 553–558.

[234] C. L. Worth, R. Preissner, T. L. Blundell, *Nucleic Acids Res.* **2011**, *39*, W215–222.
[235] M. T. Reetz, J. D. Carballeira, A. Vogel, *Angew. Chem. Int. Ed.* **2006**, *45*, 7745–7751.
[236] B. T. Miller, B. R. Brooks, R. P. Singh, J. B. Klauda, M. Hodoscek, H. L. Woodcock, *J. Chem. Inf. Model.* **2008**, *48*, 1920–1929.
[237] A. K. Malde, L. Zuo, M. Breeze, M. Stroet, D. Poger, P. C. Nair, C. Oostenbrink, A. E. Mark, *J. Chem. Theory Comput.* **2010**, *7*, 4026–4037.
[238] P. D. Renfrew, E. J. Choi, R. Bonneau, B. Kuhlman, *PLoS One* **2012**, *7*, e32637.
[239] J. Adolf-Bryfogle, R. L. Dunbrack Jr, *PLoS One* **2013**, *8*, e66856.
[240] C. D. Schenkelberg, C. Bystroff, *Bioinformatics* **2015**, *31*, 4023–4025.
[241] A. Saha, S. Lewis, G. Kleiger, B. Kuhlman, R. J. Deshaies, *Mol. Cell* **2011**, *42*, 75–83.
[242] J. K. Leman, B. D. Weitzner, S. M. Lewis, J. Adolf-Bryfogle, N. Alam, R. F. Alford, M. Aprahamian, D. Baker, K. A. Barlow, P. Barth, B. Basanta, B. J. Bender, K. Blacklock, J. Bonet, S. E. Boyken, P. Bradley, C. Bystroff, P. Conway, S. Cooper, B. E. Correia, B. Coventry, R. Das, R. M. De Jong, F. Di Maio, L. Dsilva, R. Dunbrack, A. S. Ford, B. Frenz, D. Y. Fu, C. Geniesse, L. Goldschmidt, R. Gowthaman, J. J. Gray, D. Gront, S. Guffy, S. Horowitz, P.-S. Huang, T. Huber, T. M. Jacobs, J. R. Jeliazkov, D. K. Johnson, K. Kappel, J. Karanicolas, H. Khakzad, K. R. Khar, S. D. Khare, F. Khatib, A. Khramushin, I. C. King, R. Kleffner, B. Koepnick, T. Kortemme, G. Kuenze, B. Kuhlman, D. Kuroda, J. W. Labonte, J. K. Lai, G. Lapidoth, A. Leaver-Fay, S. Lindert, T. Linsky, N. London, J. H. Lubin, S. Lyskov, J. Maguire, L. Malmström, E. Marcos, O. Marcu, N. A. Marze, J. Meiler, R. Moretti, V. K. Mulligan, S. Nerli, C. Norn, S. Ó'Conchúir, N. Ollikainen, S. Ovchinnikov, M. S. Pacella, X. Pan, H. Park, R. E. Pavlovicz, M. Pethe, B. G. Pierce, K. B. Pilla, B. Raveh, P. D. Renfrew, S. S. R. Burman, A. Rubenstein, M. F. Sauer, A. Scheck, W. Schief, O. Schueler-Furman, Y. Sedan, A. M. Sevy, N. G. Sgourakis, L. Shi, J. B. Siegel, D.-A. Silva, S. Smith, Y. Song, et al., *Nat. Methods* **2020**, *17*, 665–680.
[243] S. L. Mayo, B. D. Olafson, W. A. Goddard, *J. Phys. Chem.* **1990**, *94*, 8897–8909.
[244] H. K. Privett, G. Kiss, T. M. Lee, R. Blomberg, R. A. Chica, L. M. Thomas, D. Hilvert, K. N. Houk, S. L. Mayo, *Proc. Natl. Acad. Sci. USA* **2012**, *109*, 3790–3795.
[245] H. J. Wijma, R. J. Floor, S. Bjelic, S. J. Marrink, D. Baker, D. B. Janssen, *Angew. Chem. Int. Ed.* **2015**, *54*, 3523–3825.
[246] S. Hur, T. C. Bruice, *Proc. Natl. Acad. Sci. USA* **2003**, *100*, 12015–12020.
[247] Y. Cui, Y. Wang, W. Tian, Y. Bu, T. Li, X. Cui, T. Zhu, R. Li, B. Wu, *Nature Catalysis* **2021**, *4*, 364–373.
[248] R. Li, H. J. Wijma, L. Song, Y. Cui, M. Otzen, Y. Tian, J. Du, T. Li, D. Niu, Y. Chen, J. Feng, J. Han, H. Chen, Y. Tao, D. B. Janssen, B. Wu, *Nat. Chem. Biol.* **2021**, *14*, 664–670.
[249] P. Kokkonena, D. Bednara, G. Pintob, Z. Prokopa, J. Damborsky, *Biotechnol. Adv.* **2019**, *27*, 107386.
[250] E. Chovancova, A. Pavelka, P. Benes, O. Strnad, J. Brezovsky, B. Kozlikova, A. Gora, V. Sustr, M. Klvana, P. Medek, *PLoS Comput. Biol.* **2012**, *8*, e1002708.
[251] B. Isralewitz, M. Gao, K. Schulten, *Curr. Opin. Struct. Biol.* **2001**, *11*, 224–230.
[252] O. Vavra, J. Damborsky, D. Bednar, *Biotechnol. Adv.* **2022**, *60*, 108009.
[253] O. Trott, A. J. Olson, *J. Comput. Chem.* **2009**, *31*, 249–461.
[254] S. Osuna, *WIREs Comput. Mol. Sci.* **2021**, *11*, e1502.
[255] G. Jiménez-Osés, S. Osuna, X. Gao, M. R. Sawaya, L. Gilson, S. J. Collier, G. W. Huisman, T. O. Yeates, Y. Tang, K. N. Houk, *Nat. Chem. Biol.* **2014**, *10*, 431–436.
[256] M. Wilding, N. Hong, M. Spence, A. M. Buckle, C. J. Jackson, *Biochem. Soc. Trans.* **2019**, *47*, 701–711.
[257] J. Gu, Y. Xu, Y. Nie, *Biotechnol. Adv.* **2023**, *63*, 108094.
[258] M. A. Maria-Solano, T. Kinateder, J. Iglesias-Fernández, R. Sterner, S. Osuna, *ACS Catal.* **2021**, *11*, 13733–13743.
[259] S. Somarowthu, H. Yang, D. G. C. Hildebrand, M. J. Ondrechen, *Biopolymers* **2011**, *95*, 390–400.
[260] R. Parasuram, T. A. Coulther, J. M. Hollander, E. Keston-Smith, M. J. Ondrechen, P. J. Beuning, *Biochemistry* **2018**, *57*, 1063–1072.
[261] E. J. Ma, E. Siirola, C. Moore, A. Kummer, M. Stoeckli, M. Faller, C. Bouquet, F. Eggimann, M. Ligibel, D. Huynh, *ACS Catal.* **2021**, *11*, 12433–12445.
[262] S. Sugiki, T. Niide, Y. Toya, H. Shimizu, *ACS Synth. Biol.* **2022**, *11*, 3973–3985.
[263] B. M. Bonk, J. W. Weis, B. Tidor, *J. Am. Chem. Soc.* **2018**, *141*, 4108–4118.
[264] D. Platero-Rochart, T. Krivobokova, M. Gastegger, G. Reibnegger, P. A. Sanchez-Murcia, *ChemRxiv. preprint* **2023**, 10.26434/chemrxiv-22023-v26430rj26436.
[265] N. D. Clarke, S.-M. Yuan, *Proteins Struct. Funct. Genet.* **1995**, *23*, 256–263.
[266] V. K. Mulligan, S. Workman, T. Sun, S. Rettie, X. Li, L. J. Worrall, T. W. Craven, D. T. King, P. Hosseinzadeh, A. M. Watkins, P. D. Renfrew, S. Guffy, J. W. Labontef, R. Moretti, R. Bonneau, N. C. J. Strynadka, D. Baker, *Proc. Natl. Acad. Sci. USA* **2021**, *118*, e2012800118.
[267] J. H. Mills, S. D. Khare, J. M. Bolduc, F. Forouhar, V. K. Mulligan, S. Lew, J. Seetharaman, L. Tong, B. L. Stoddard, D. Baker, *J. Am. Chem. Soc.* **2013**, *135*, 13393–13399.
[268] T. Heinisch, M. Pellizzoni, M. Dürrenberger, C. E. Tinberg, V. Köhler, J. Klehr, D. Häussinger, D. Baker, T. R. Ward, *J. Am. Chem. Soc.* **2015**, *137*, 10414–10419.
[269] J. H. Mills, W. Sheffler, M. E. Ener, P. J. Almhjell, G. Oberdorfer, J. H. Pereira, F. Parmeggiani, B. Sankaran, P. H. Zwart, D. Baker, *Proc. Natl. Acad. Sci. USA* **2016**, *113*, 15012–15017.
[270] G. B. Akcapinar, O. U. Sezerman, *Biosci. Rep.* **2017**, *37*, BSR20160179.
[271] H. J. Davis, T. R. Ward, *ACS Cent. Sci.* **2019**, *5*, 1120–1136.
[272] L. Alonso-Cotchico, J. Rodrıguez-Guerra, A. Lledós, J.-D. Maréchal, *Acc. Chem. Res.* **2020**, *53*, 896–905.
[273] V. Muñoz Robles, E. Ortega-Carrasco, L. Alonso-Cotchico, J. Rodriguez-Guerra, A. Lledós, J.-D. Maréchal, *ACS Catal.* **2015**, *5*, 2469–2480.
[274] A. Sharma, G. Gupta, T. Ahmad, S. Mansoor, B. Kaur, *Food Rev. Int.* **2021**, *37*, 121–154.
[275] M. F. Cole, E. A. Gaucher, *Curr. Opin. Chem. Biol.* **2011**, *15*, 399–406.
[276] L. C. Wheeler, S. A. Lim, S. Marqusee, M. J. Harms, *Curr. Opin. Struct. Biol.* **2016**, *38*, 37–43.
[277] X. Chen, Z. Dou, T. Luo, Z. Sun, H. Ma, G. Xu, Y. Ni, *Bioresour. Technol.* **2022**, *363*.
[278] M. L. Mascotti, H. Kumar, Q.-T. Nguyen, M. J. Ayub, M. W. Fraaije, *Sci. Rep.* **2018**, *8*.
[279] Z. Xu, Y.-K. Cen, S.-P. Zou, Y.-P. Xue, Y.-G. Zheng, *Crit. Rev. Biotechnol.* **2020**, *40*, 83–98.
[280] R. Guerois, J. E. Nielsen, L. Serrano, *J. Mol. Biol.* **2002**, *320*, 369–387.
[281] O. Buß, J. Rudat, K. Ochsenreither, *Comput. Struct. Biotechnol. J.* **2018**, *16*, 25–33.
[282] 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.
[283] H. J. Wijma, R. J. Floor, P. A. Jekel, D. Baker, S. J. Marrink, D. B. Janssen, *Protein Eng. Des. Sel.* **2014**, *27*, 49–58.
[284] M. J. L. J. Fürst, M. Boonstra, S. Bandstra, M. W. Fraaije, *Biotechnol. Bioeng.* **2019**, *116*, 2167–2177.
[285] I. Mateljak, A. Rice, K. Yang, T. Tron, M. Alcalde, *ACS Synth. Biol.* **2019**, *8*, 833–843.
[286] Z. Sun, Q. Liu, G. Qu, Y. Feng, M. T. Reetz, *Chem. Rev.* **2019**, *119*, 1626–1665.
[287] D. E. Pires, D. B. Ascher, T. L. Blundell, *Nucleic Acids Res.* **2014**, *42*, W314–W319.
[288] R. Nikam, A. Kulandaisamy, K. Harini, D. Sharma, M. M. Gromiha, *Nucleic Acids Res.* **2021**, *49*, D420–D424.
[289] I. H. Moal, J. Fernandez-Recio, *Bioinformatics* **2012**, *28*, 2600–2607.
[290] K. P. Kepp, *Biochim. Biophys. Acta Proteins Proteomics* **2015**, *1854*, 1239–1248.
[291] A. Schmid, J. S. Dordick, B. Hauer, A. Kiener, M. Wubbolts, B. Witholt, *Nature* **2001**, *409*, 258–268.
[292] M. T. Reetz, *J. Am. Chem. Soc.* **2013**, *135*, 12480–12496.
[293] H. E. Schoemaker, D. Mink, M. G. Wubbolts, *Science* **2003**, *299*, 1694–1697.
[294] J. M. Woodley, *Appl. Microbiol. Biotechnol.* **2019**, *103*, 4733–4739.
[295] J. B. Broderick, in *Encyclopedia of life sciences, Wiley*, **2001**, pp. 1–7.
[296] F. Schwizer, Y. Okamoto, T. Heinisch, Y. Gu, M. M. Pellizzoni, V. Lebrun, R. Reuter, V. Köhler, J. C. Lewis, T. R. Ward, *Chem. Rev.* **2018**, *118*, 142–231.
[297] M. Vasina, J. Velecký, J. Planas-Iglesias, S. M. Marques, J. Skarupova, J. Damborsky, D. Bednar, S. Mazurenko, Z. Prokop, *Adv. Drug Delivery Rev.* **2022**, *183*, 114143.
[298] I. Drienovská, L. Alonso-Cotchico, P. Vidossich, A. Lledós, J.-D. Maréchal, G. Roelfes, *Chem. Sci.* **2017**, *8*, 7228–7235.
[299] Y. S. Zubi, K. Seki, Y. Li, A. C. Hunt, B. Liu, B. Roux, M. C. Jewett, J. C. Lewis, *Nat. Commun.* **2022**, *13*, 1864.
[300] O. Buß, P. C. F. Buchholz, M. Gräff, P. Klausmann, J. Rudat, J. Pleiss, *Proteins Struct. Funct. Bioinf.* **2018**, *86*, 566–580.

[301] I. Slabu, J. L. Galman, R. C. Lloyd, N. J. Turner, *ACS Catal.* **2017**, *7*, 8263–8284.
[302] C. K. Savile, J. M. Janey, E. C. Mundorff, J. C. Moore, S. Tam, W. R. Jarvis, J. C. Colbeck, A. Krebber, F. J. Fleitz, J. Brands, P. N. Devine, G. W. Huisman, G. J. Hughes, *Science* **2010**, *329*, 305–309.
[303] S. A. Kelly, S. Pohle, S. Wharry, S. Mix, C. C. R. Allen, T. S. Moody, B. F. Gilmore, *Chem. Rev.* **2018**, *118*, 349–367.
[304] Q. Meng, C. Ramírez-Palacios, N. Capra, M. E. Hooghwinkel, S. Thallmair, H. J. Rozeboom, A.-M. W. H. Thunnissen, H. J. Wijma, S. J. Marrink, D. B. Janssen, *ACS Catal.* **2021**, *11*, 10733–10747.
[305] Q. Meng, N. Capra, C. M. Palacio, E. Lanfranchi, M. Otzen, L. Z. Van Schie, H. t J Rozeboom, A.-M. W. H. Thunnissen, H. J. Wijma, D. B. Janssen, *ACS Catal.* **2020**, *10*, 2915–2928.
[306] M. Wilding, T. S. Peat, S. Kalyaanamoorthy, J. Newman, C. Scott, L. S. Jermiin, *Green Chem.* **2017**, *19*, 5375–5380.
[307] H. Minges, N. Sewald, *ChemCatChem* **2020**, *12*, 4446–4697.
[308] V. Weichold, D. Milbredt, K.-H. van Pie, *Angew. Chem. Int. Ed.* **2017**, *55*, 6374–6389.
[309] T. G. Karabencheva-Christova, J. Torras, A. J. Mulholland, A. Lodola, C. Z. Christov, *Sci. Rep.* **2017**, *7*, 17395.
[310] J. Ainsley, A. J. Mulholland, G. W. Black, O. Sparagano, C. Z. Christov, T. G. Karabencheva-Christova, *ACS Omega* **2018**, *3*, 4847–4859.
[311] M. C. Andorfer, D. Evans, S. Yang, C. Q. He, A. M. Girlich, J. Vergara-Coll, N. Sukumar, K. N. Houk, J. C. Lewis, *Chem. Catal.* **2022**, *2*, 2658–2674.
[312] K. Prakinee, A. Phintha, S. Visitsatthawong, N. Lawan, J. Sucharitakul, C. Kantiwiriyawanitch, J. Damborsky, P. Chitnumsub, K.-H. van Pée, P. Chaiyen, *Nat. Catal.* **2022**, *5*, 534–544.
[313] K. D. Dubey, S. Shaik, *Acc. Chem. Res.* **2019**, *52*, 389–399.
[314] D. Roccatano, *J. Phys. Condens. Matter* **2015**, *27*, 273102.
[315] C. Hui, W. Singh, D. J. Quinn, C. Li, T. S. Moody, M. Huang, *Phys. Chem. Chem. Phys.* **2020**, *22*, 21696–21706.
[316] J.-b. Wang, Q. Huang, W. Peng, P. Wu, D. Yu, B. Chen, B. Wang, M. T. Reetz, *J. Am. Chem. Soc.* **2020**, *142*, 2068–2073.
[317] J. Pleiss, *ChemCatChem* **2014**, *6*, 944–950.
[318] J. A. Dietrich, Y. Yoshikuni, K. J. Fisher, F. X. Woolard, D. Ockey, D. J. McPhee, N. S. Renninger, M. C. Y. Chang, D. Baker, J. D. Keasling, *ACS Chem. Biol.* **2009**, *4*, 261–267.
[319] S. Meng, Y. Ji, L. Liu, M. D. Davari, U. Schwaneberg, *ChemSusChem* **2021**, *15*, e202102434.
[320] Y. Gumulya, J.-M. Baek, S.-J. Wun, R. E. S. Thomson, K. L. Harris, D. J. B. Hunter, J. B. Y. H. Behrendorff, J. Kulig, S. Zheng, X. Wu, B. Wu, J. E. Stok, J. J. De Voss, G. Schenk, U. Jurva, S. Andersson, E. M. Isin, M. Bodén, L. Guddat, E. M. J. Gillam, *Nat. Catal.* **2018**, *1*, 878–888.
[321] S. Meng, Y. Ji, L. Zhu, G. V. Dhoke, M. D. Davari, U. Schwaneberg, *Biotechnol. Adv.* **2022**, *61*.
[322] J. C. Greenhalgh, S. A. Fahlberg, B. F. Pfleger, P. A. Romero, *Nat. Commun.* **2021**, *12*, 5825.
[323] I. Paik, S. Bhadra, A. D. Ellington, *ACS Synth. Biol.* **2022**, *11*, 1488–1496.
[324] T. Vornholt, F. Christoffel, M. M. Pellizzoni, S. Panke, T. R. Ward, M. Jeschek, *Sci. Adv.* **2021**, *7*, eabe4208.
[325] M. R. Bennett, S. A. Shepherd, V. A. Cronin, J. Micklefield, *Curr. Opin. Chem. Biol.* **2017**, *37*, 97–106.
[326] K. Sharma, S. Morla, A. Goyal, S. Kumar, *Life Sci.* **2020**, *259*, 118169.
[327] T. Otava, M. Šála, F. Li, J. i Fanfrlík, K. Devkota, S. Perveen, I. Chau, P. Pakarian, P. Hobza, M. Vedadi, E. Boura, R. Nencka, *ACS Infect. Dis.* **2021**, *7*, 2214–2220.
[328] H. Schonherr, T. Cernak, *Angew. Chem. Int. Ed.* **2013**, *52*, 12256–12267.
[329] T. Lankau, H. C. Ken, H. M. Chang, C. H. Yu, *ACS Omega* **2022**, *7*, 12753–12764.
[330] X. Chen, R. T. J. Esque, C. Zhang, S. Shukal, C. C. Lim, L. Ong, D. Smith, I. André, *Nat. Commun.* **2022**, *13*, 7421.
[331] R. Park, C. Ongpipattanakul, S. K. Nair, A. A. Bowers, B. Kuhlman, *Nat. Chem. Biol.* **2022**.
[332] F. Ospina, K. H. Schülke, J. Soler, A. Klein, B. Prosenc, M. Garcia-Borràs, S. C. Hammer, *Angew. Chem. Int. Ed.* **2022**, *61*, e202213056.
[333] L. L. Bengel, B. Aberle, A.-N. Egler-Kemmerer, S. Kienzle, B. Hauer, S. C. Hammer, *Angew. Chem. Int. Ed.* **2021**, *60*, 5554–5560.
[334] A. Giessel, A. Dousis, K. Ravichandran, K. Smith, S. Sur, I. McFadyen, W. Zheng, S. Licht, *Sci. Rep.* **2022**, *12*, 1536.
[335] H. Younus, *Int. J. Health Sci. (Qassim)* **2018**, *12*, 88–93.
[336] S. Kumar, V. K. Bhardwaj, A. Kaachra, S. Guleria, A. Kumar, R. Purohit, S. Kumar, *Plant Physiol. Biochem.* **2021**, *168*, 221–229.
[337] J. A. Doudna, E. Charpentier, *Science* **2014**, *346*, 1258096.
[338] I. M. Slaymaker, N. M. Gaudelli, *Curr. Opin. Struct. Biol.* **2021**, *69*, 86–98.
[339] A. Saha, P. R. Arantes, G. Palermo, *Curr. Opin. Struct. Biol.* **2022**, *75*, 102400.
[340] Ł. Nierzwicki, K. W. East, J. M. Binz, R. V. Hsu, M. Ahsan, P. R. Arantes, E. Skeens, M. Pacesa, M. Jinek, G. P. Lisi, G. Palermo, *Nat. Catal.* **2022**, *5*, 912–922.
[341] G. Palermo, Y. Miao, R. C. Walker, M. Jinek, J. A. McCammon, *Proc. Natl. Acad. Sci. USA* **2017**, *114*, 7260–7265.
[342] G. Palermo, Y. Miao, R. C. Walker, M. Jinek, J. A. McCammon, *ACS Cent. Sci.* **2016**, *2*, 756–763.
[343] L. Zheng, J. Shib, Y. Mu, *Phys. Chem. Chem. Phys.* **2018**, *20*, 27439–27448.
[344] P. D. Vos, G. Rossetti, J. L. Mantegna, S. J. Siira, A. P. Gandadireja, M. Bruce, S. A. Raven, O. Khersonsky, S. J. Fleishman, A. Filipovska, O. Rackham, *Nat. Commun.* **2022**, *13*, 3023.
[345] B. Alonso-Lerma, Y. Jabaler, S. Samperio, M. Morin, A. Fernandez, L. T. Hille, R. A. Silverstein, A. Quesada-Ganuza, A. Reifs, S. Fernández-Peñalver, Y. Benitez, L. Soletto, J. A. Gavira, A. Diaz, W. F. Vranken, A. Sanchez-Mejias, M. Güell, F. J. M. Mojica, B. P. Kleinstiver, M. A. Moreno-Pelayo, L. Montoliu, R. Perez-Jimenez, *Nat. Microbiol.* **2023**, *8*, 77–90.
[346] K.-P. Hopfner, V. Hornung, *Nat. Rev. Mol. Cell Biol.* **2020**, *21*, 501–521.
[347] J. Soler, P. Paiva, M. J. Ramos, P. A. Fernandes, M. Brut, *Phys. Chem. Chem. Phys.* **2021**, *23*, 9524.
[348] Q. M. Dowling, H. E. Volkman, E. E. Gray, S. Ovchinnikov, S. Cambier, A. K. Bera, B. Sankaran, M. R. Johnson, M. J. Bick, A. Kang, D. B. Stetson, N. P. King, *Nat. Struct. Mol. Biol.* **2022**, *30*, 72–80.
[349] A. Shah, M. Jain, in *Computer Aided Drug Design (CADD): From Ligand-Based Methods to Structure-Based Approaches* (Eds.: M. Rudrapal, C. Egbuna), Elsevier, **2022**, pp. 283–297.
[350] J. G. Pedreira, L. S. Franco, E. J. Barreiro, *Curr. Top. Med. Chem.* **2019**, *19*, 1679–1693.
[351] M. C. Zwier, L. T. Chong, *Curr. Opin. Pharmacol.* **2010**, *10*, 745–752.
[352] A. H.-W. Yeh, C. Norn, Y. Kipnis, D. Tischer, S. J. Pellock, D. Evans, P. Ma, G. R. Lee, J. Z. Zhang, I. Anishchenko, B. Coventry, L. Cao, J. Dauparas, S. Halabiya, M. DeWitt, L. Carter, K. N. Houk, D. Baker, *Nature* **2023**, *614*, 774–780.
[353] F. Agostini, J.-S. Völler, B. Koksch, C. G. Acevedo-Rocha, V. Kubyshkin, N. Budisa, *Angew. Chem. Int. Ed.* **2017**, *56*, 9680–9703.

Manuscript received: March 10, 2023
Revised manuscript received: May 4, 2023
Accepted manuscript online: May 7, 2023
Version of record online: May 24, 2023