CHAPTER FIVE

OSPREY: Protein Design with Ensembles, Flexibility, and Provable Algorithms

Pablo Gainza[*,3], Kyle E. Roberts[*,3], Ivelin Georgiev[*,1], Ryan H. Lilien[†], Daniel A. Keedy[‡], Cheng-Yu Chen[‡], Faisal Reza[§,2], Amy C. Anderson[¶], David C. Richardson[‡], Jane S. Richardson[‡], Bruce R. Donald[*,‡,4]

[*]Department of Computer Science, Duke University, Durham, North Carolina, USA
[†]Department of Computer Science, University of Toronto, Toronto, Ontario, Canada
[‡]Department of Biochemistry, Duke University Medical Center, Durham, North Carolina, USA
[§]Department of Biomedical Engineering, Duke University Medical Center, Durham, North Carolina, USA
[¶]Department of Pharmaceutical Sciences, University of Connecticut, Storrs, Connecticut, USA
[1]Current address: Vaccine Research Center, National Institute of Allergy and Infectious Diseases, National Institutes of Health (NIH), Bethesda, Maryland, USA
[2]Current address: Department of Therapeutic Radiology, Yale University School of Medicine, New Haven, Connecticut, USA
[3]These authors contributed equally to this work.
[4]Corresponding author: e-mail address: brd@cs.duke.edu

Contents

Methods in Enzymology, Volume 523
ISSN 0076-6879
http://dx.doi.org/10.1016/B978-0-12-394292-0.00005-9

Abstract

Summary: We have developed a suite of protein redesign algorithms that improves realistic *in silico* modeling of proteins. These algorithms are based on three characteristics that make them unique: (1) *improved flexibility* of the protein backbone, protein sidechains, and ligand to accurately capture the conformational changes that are induced by mutations to the protein sequence; (2) modeling of proteins and ligands as *ensembles* of low-energy structures to better approximate binding affinity; and (3) a globally optimal protein design search, guaranteeing that the computational predictions are optimal with respect to the input model. Here, we illustrate the importance of these three characteristics. We then describe OSPREY, a protein redesign suite that implements our protein design algorithms. OSPREY has been used prospectively, with experimental validation, in several biomedically relevant settings. We show in detail how OSPREY has been used to predict resistance mutations and explain why improved flexibility, ensembles, and provability are essential for this application.

Availability: OSPREY is free and open source under a Lesser GPL license. The latest version is OSPREY 2.0. The program, user manual, and source code are available at www.cs.duke.edu/donaldlab/software.php. *Contact*: osprey@cs.duke.edu

1. INTRODUCTION

Technological advances in protein redesign could revolutionize therapeutic treatment. With these advances, proteins and other molecules can be designed to act on today's undruggable proteins or tomorrow's drug-resistant diseases. One of the most promising approaches in protein redesign is structure-based computational protein redesign (SCPR). SCPR programs model a protein's three-dimensional structure and predict mutations to the native protein sequence that will have a desired effect on its biochemical properties and function, such as improving the affinity of a drug-like protein for a disease target. In this chapter, we describe OSPREY (*O*pen *S*ource *P*rotein *R***e**design for *Y*ou), a free, open-source SCPR program. We have prospectively used OSPREY, with experimental validation, to redesign enzymes (Chen, Georgiev, Anderson, & Donald, 2009), design new drugs (Gorczynski et al., 2007), predict drug resistance (Frey, Georgiev, Donald, & Anderson, 2010), design peptide inhibitors of protein–protein interactions (Roberts, Cushing, Boisguerin, Madden, & Donald, 2012), and design epitope-specific antibody probes (Georgiev, Acharya, et al., 2012).

Predicting mutations that result in a desired protein structure and enable novel function or new biochemical properties presents four main protein design challenges. First, as the number of mutated residues to the native

sequence increases, the number of unique protein sequences, or the size of *sequence space*, increases exponentially. Second, mutating a protein sequence induces conformational changes to the protein structure. Thus, the most stable, lowest energy conformations of one sequence can differ significantly from those of another sequence. A protein's potential flexibility occurs over many degrees of freedom. This results in an astronomically large, continuous space over which SCPR algorithms must search. A third challenge is that, for each protein sequence, an ensemble of low-energy states exists, which contributes to protein–ligand binding (Gilson, Given, Bush, & McCammon, 1997). Thus, each binding partner's conformational ensemble must be considered to compute the binding energy of the protein and ligand (Donald, 2011; Lilien, Stevens, Anderson, & Donald, 2005). Finally, the fourth challenge in protein design is calculating the energy that drives protein structure and function at the molecular level. The most accurate models would require computationally expensive quantum mechanical simulations of the protein and solvent, which is intractable for SCPR problems.

These challenges require SCPR programs to make approximations in their *input model*. The input model defines (i) the initial protein structure, (ii) the sequence space to which the protein can mutate, (iii) the allowed protein flexibility, and (iv) the energy function to rank the generated conformations. The input model must be carefully chosen to minimize the error that stems from its approximations, while at the same time ensuring that the SCPR algorithm can efficiently search the protein conformational space.

The accuracy of an SCPR program largely depends on how it addresses the protein design challenges. OSPREY's approach is based on three main protein design principles: (1) realistic, yet efficient, models of flexibility; (2) ensembles of low-energy conformations; and (3) provable optimality with respect to the input model. In Section 2, we describe these three principles and their importance. Section 3 details specific design problems where OSPREY has been applied. Section 4 describes the OSPREY program and its input, algorithms, and expected output. In Section 5, we show how to use OSPREY to predict drug resistance-conferring mutations.

2. OSPREY DESIGN PRINCIPLES

2.1. Protein flexibility

Proteins are dynamic and can exist in many different low-energy, relatively near-native conformations at physiological conditions. The ability of a ligand to select or induce protein conformations demonstrates the

requirement of SCPR algorithms to accurately model flexibility (Teague, 2003). However, SCPR algorithms often must limit protein flexibility during the design search in the interest of computational feasibility.

One common SCPR approximation is to limit the allowed side-chain conformations to search. Protein amino acid side-chains appear in clusters at low-energy regions of χ-angle space, known as rotamers (Lovell, Word, Richardson, & Richardson, 2000). Many SCPR programs use discrete rotamers to represent each cluster as only a single point in χ-angle space. However, protein energetics are sensitive to small changes in atom coordinates; so, the reduction of a cluster to a single discrete conformation cannot fully describe a continuous region of side-chain conformation space.

To improve upon the limitations of discrete rotamers, OSPREY implements *continuous rotamers* (Gainza, Roberts, & Donald, 2012; Georgiev, Lilien, & Donald, 2008). In contrast to discrete rotamers, each continuous rotamer is a region in χ-angle space that more accurately reflects the empirically discovered side-chain clusters. A large-scale study of protein core designs using continuous rotamers versus discrete rotamers demonstrated the benefits of continuous rotamers in protein design (Gainza et al., 2012). Importantly, continuous rotamers were able to find conformations that were both lower in energy and had different sequences than the conformations found using discrete rotamers, even when more expansive discrete rotamer libraries were used. This means that discrete rotamers do not accurately quantize conformation space and will likely result in less than optimal design predictions. Also, using continuous rotamers improves the biological accuracy of the designs. Specifically, sequences found using continuous rotamers were significantly more similar to native sequences than sequences found with rigid rotamers. The accuracy improvements are comparable to gains achieved when incorporating sophisticated energy terms such as solvation (Hu & Kuhlman, 2006). Therefore, continuous rotamers are likely required to accurately search conformation space to find the true low-energy protein structures.

While the large-scale study in Gainza et al. (2012) was conducted for side-chain flexibility, OSPREY can also be used to search over local backbone flexibility (Georgiev, Lilien, & Donald, 2008), or continuous global backbone flexibility (Georgiev & Donald, 2007). Extrapolating from the benefits obtained by using continuous rotamers, similar benefits were shown (Hallen, Keedy, & Donald, 2013) when using OSPREY's flexible backbone models instead of traditional backbone models (that use only a fixed backbone or discrete backbone conformers). The benefits of continuous rotamers and

continuous backbone flexibility have been experimentally demonstrated by Chen et al. (2009), Frey et al. (2010), and Roberts et al. (2012).

2.2. Ensemble-based design

Traditional protein design methods often focus on finding the single global minimum energy conformation (GMEC) for a design. However, this simplification ignores the reality that proteins in solution exist as a thermodynamic ensemble of conformations, and not just a single low-energy structure (Fig. 5.1). In fact, current nuclear magnetic resonance (NMR) techniques can now estimate relative populations of side-chain rotamers in folded proteins (Chou, Case, & Bax, 2003). It is the nature of this thermodynamic ensemble that governs protein–ligand binding (Gilson et al., 1997). Therefore, if several low-energy conformations contribute to protein–ligand binding, a model that only considers a single GMEC is likely to incorrectly predict binding.

OSPREY uses the K^* algorithm (Donald, 2011; Georgiev, Lilien, et al., 2008) to efficiently approximate the association constant, K_A, of a protein–ligand complex using structural ensembles. K^* considers ensembles of only the most probable low-energy conformations and discards the majority of conformations that are rarely populated by the protein or ligand. K^*'s ability to accurately rank protein sequences by weighting ensembles of low-energy conformations relies heavily on OSPREY's provable guarantees

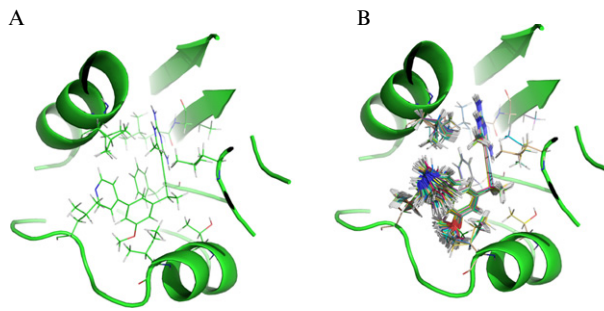

Figure 5.1 *Binding prediction using a single conformation versus using an ensemble.* Dihydrofolate reductase from methicillin-resistant *Staphylococcus aureus* is shown bound to a propargyl-linked anti-folate inhibitor (Frey et al., 2010). (A) Many SCPR algorithms use a single low-energy conformation to model a protein–ligand complex. The GMEC for the protein–ligand complex is shown. (B) OSPREY's MinDEE/*A*/K** pipeline models the most populated conformations in which binding occurs. Members of an ensemble of bound low-energy conformations are superimposed.

(discussed below). Since OSPREY can guarantee that it finds all low-energy conformations for a protein sequence, the generated ensembles do not lack any critical conformations and can be accurately ranked for each sequence. We found K^* to be more accurate and reliable than GMEC-based designs when applying OSPREY to biologically relevant protein design systems (Chen et al., 2009; Roberts et al., 2012).

2.3. Provable guarantees

SCPR requires searching over a very large protein conformation space. Even when searching over a relatively small rotamer library (152 rotamers; Lovell et al., 2000), redesigning 10 residues results in approximately 10^{21} possible rotamer combinations. To handle this large space, heuristic search methods, such as Monte Carlo, are often used. However, when using heuristic methods, it is impossible to know when the design search is complete and how close the computed protein conformation is to the GMEC. Therefore, OSPREY uses provable techniques that guarantee that it finds all low-energy conformations with respect to the input model.

As discussed above, the protein design input model contains many assumptions that can potentially cause errors in the protein design predictions. Ultimately, experimental validation is required to determine whether these assumptions are sufficiently accurate. If the experimentally tested SCPR predictions are successful, the input model is considered sufficiently accurate. However, if the designs fail, it is crucial to ascertain why they did not function as designed. One key advantage of provable SCPR is that there is no error or inaccuracy arising from the search; so, all error can be attributed to the input model. Specifically, if a design prediction fails, one can be confident that improvements should be made to the input model. In contrast, if a heuristic approach were used, it is impossible to disambiguate inaccuracies in the input model from inaccuracies resulting from an insufficient search of the input model.

Misattributing heuristic SCPR search inaccuracies as flaws in the input model could have dire consequences when trying to improve protein energy models. If energy term weights are recalculated or additional terms are added to the energy function based on this misinformation, overfitting is likely to occur. The overfitting is worsened because the energies are not fit to the actual GMECs but rather to the local minima found by the heuristic search. Therefore, training an energy function and improving an input model is more straightforward when using provable SCPR techniques (Roberts et al., 2012).

2.4. Significance of design principles in positive/negative design

Most applications of SCPR focus on stabilizing a target protein fold or binding capability (positive design). When trying to design specificity for a single target, it is also important to *prevent* unwanted folds or binding events from occurring (negative design). A successful positive design only requires finding at least one protein sequence with the desired properties. However, in negative design, the SCPR algorithm must be confident that no off-target binding occurs. Therefore, negative design is much more sensitive to false negatives and requires a more thorough search of the conformation space. Missing low-energy conformations is more detrimental to a negative design than to a positive design. All of the main OSPREY design principles focus on accurately and completely searching the low-energy protein conformation space, which will likely be a great advantage for negative design efforts (Donald, 2011; Frey et al., 2010; Georgiev, Acharya, et al., 2012; Roberts et al., 2012). We further explore positive and negative design with OSPREY in Section 5.2.

3. APPLICATIONS OF OSPREY

We have used OSPREY in several successful prospective designs. In this section, we summarize these designs and mention which protein design algorithms were used for each design. All of these algorithms are explained in detail in Section 4.2.

OSPREY was used to switch the specificity of the phenylalanine adenylation domain of the nonribosomal peptide synthetase enzyme gramicidin S synthetase toward a set of substrates for which the wild-type enzyme had little or no specificity (Chen et al., 2009). The K^* algorithm, with both Minimized DEE (MinDEE) and BD, predicted mutations to the catalytic active site that would switch the substrate specificity. The OSPREY self-consistent mean field (SCMF) module was then used to find residue positions distal from the active site that could bolster the stability of the redesigned enzymes. The chosen distal positions were analyzed with MinDEE to determine the most stabilizing mutations. The mutant enzyme with the highest activity toward its noncognate substrate (L-Leu) showed 1/6 of the wild-type protein/substrate activity. This mutant showed a 2168-fold switch in specificity from the cognate (L-Phe) to the noncognate (L-Leu) substrate.

In Georgiev, Acharya, et al. (2012), OSPREY used a positive/negative design approach to design epitope-specific antibody probes. OSPREY predicted HIV-1 gp120 mutations that would eliminate the binding of specific, undesired antibodies. ELISA assays confirmed that the designed probes maintained binding to their target antibodies, and had weak or only moderate binding toward undesired antibodies. A set of these designed probes is currently being used to isolate antibodies that target epitopes of interest. In Section 5, we describe how OSPREY's positive/negative design approach was used to predict bacterial resistance mutations (Frey et al., 2010).

In Roberts et al. (2012), MinDEE and K^* were used to design peptides that inhibited the cystic fibrosis-related interaction between the proteins CAL and cystic fibrosis transmembrane conductance regulator (CFTR). Specifically, OSPREY redesigned the protein–protein interface between the CFTR C-terminus and CAL to produce a competitive peptide inhibitor of the interaction. The top-ranked peptide bound the CAL protein with a sevenfold better affinity than the previous best-known hexamer peptide and 170-fold more tightly than the CFTR C-terminus. The top-ranked peptide was also shown to rescue chloride flux in human airway epithelial cells containing the ΔF508-CFTR mutation.

In Gorczynski et al. (2007), OSPREY screened inhibitors of a leukemia-associated protein–protein interaction. An earlier version of K^* (Georgiev, Lilien, et al., 2008) used ensembles of protein structures from NMR to rank small molecules that could bind to CBF-β and disrupt its interaction with the protein Runx1. The small molecules allosterically inhibited the interaction and prevented proliferation of cancerous cells.

Finally, we have also used OSPREY in combination with sparse NMR data to determine protein structures. In Zeng, Zhou, and Donald (2011), the DEE/A* algorithms enabled the computation of side-chain resonance assignments and backbone structure with less NMR data than traditional structure determination methods. In Zeng, Roberts, Zhou, and Donald (2011), side-chain conformations were inferred using a modified version of OSPREY that incorporates unassigned distance restraints data into side-chain placement optimization.

4. PROTEIN DESIGN IN OSPREY

OSPREY redesigns a protein's function and biochemical properties. To perform a redesign, OSPREY requires as input: (a) a 3D structure of the protein to be redesigned; (b) the sequence space, as defined by the allowed mutations

to the redesigned protein; (c) the allowed protein flexibility, defined by both an empirical database of favored side-chain conformations (a *rotamer library*) and the type of allowed flexibility (e.g., see below and Fig. 5.2); and (d) an all-atom pairwise energy function to score protein conformations. The 3D structure, sequence space, and allowed flexibility define the conformation search space. A suite of algorithms with mathematical guarantees then computes the GMEC and, optionally, a gap-free list of the other lowest energy conformations. Finally, sequences are ranked using either the GMEC for each sequence or a binding constant prediction based on the computed ensemble of low-energy conformations. Here, we give a brief overview of the input and algorithms. Detailed explanations can be found in Donald (2011), Gainza et al. (2012), Georgiev, Lilien, and Donald (2006), Georgiev and Donald (2007), Georgiev, Keedy, Richardson, Richardson, and Donald (2008), Georgiev, Lilien, et al. (2008), and Lilien et al. (2005) and in the OSPREY user manual (Georgiev, Roberts, Gainza, & Donald, 2012).

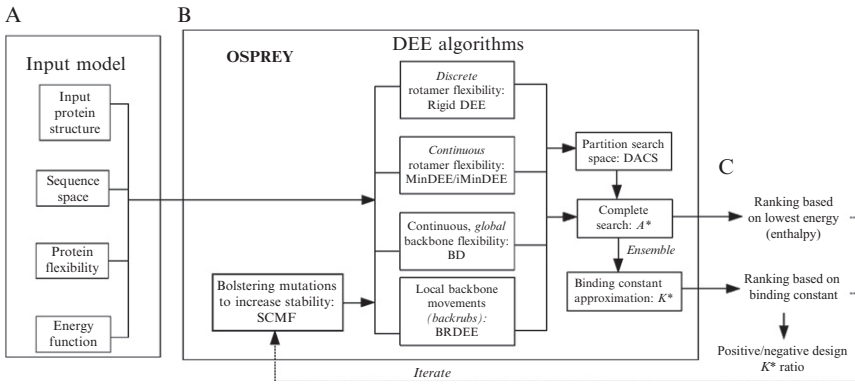

Figure 5.2 *SCPR with OSPREY*. (A) The input model (see Section 4.1). (B) According to the type of flexibility allowed, a specific pruning algorithm is run. The output from the pruning algorithm is directed to either a divide-and-conquer algorithm or directly to the *A** algorithm for a full conformational search. According to the user's selection, the *A** output can then be used to generate a ranking for each sequence based on either the lowest energy structure or on an ensemble of structures generated by the *K** algorithm. (C) If the goal is to find sequences that have a high affinity for one ligand (positive design) while having a low affinity for another (negative design), a ranking can be produced based on the ratio of *K** scores (i.e., positive design score/negative design score; Frey et al., 2010). In addition, if desired, predicted mutants can be improved by finding bolstering mutations that can increase the stability of a mutant. The bolstering mutations can be designed using any of the DEE variants and *A**.

4.1. Input model

OSPREY requires a protonated protein structure in PDB format that has no residues with missing atoms. A single molecule can be modeled as a flexible ligand, while all other water and nonamino acid molecules present in the PDB file must be either specified as rigid bodies or removed. When a ligand is present, OSPREY can use distinct or identical structures for both the protein and ligand's unbound (*apo*) states and for the bound (*holo*) state.

By default, OSPREY includes the Richardsons' Penultimate Rotamer Library (Lovell et al., 2000) and is extensible to other rotamer libraries. Rotamer libraries are available for all natural amino acids, but they are rare for nonamino acid molecules. Thus, in cases where a nonamino acid small molecule is used as the ligand, the user must define its low-energy conformations (called the small molecule's *generalized rotamers*). Within OSPREY, the small molecule's generalized rotamers consist of different conformations of the molecule's flexible dihedral-torsion angles. These conformations are defined by specifying the angle value for each flexible dihedral torsion in the molecule. For a specific example of a generalized rotamer, see Frey et al. (2010) and Georgiev, Roberts, et al. (2012).

OSPREY relies on empirical pairwise-decomposable energy functions to rank protein conformations. OSPREY includes both the Amber96 (Pearlman et al., 1995) and CHARMM (Brooks et al., 2009) energy functions for electrostatics and repulsive–attractive van der Waals (vdW) forces. EEF1 (Lazaridis & Karplus, 1999) is used to score solvation penalties, which are the cost to bury hydrophilic amino acids and/or solvate hydrophobic residues. OSPREY includes charges for some nonamino acid molecules, such as DNA and RNA nucleotides (Reza, 2010), as well as waters. Charges for other organic molecules can be precomputed with a program such as Antechamber (Chen et al., 2009; Frey et al., 2010; Wang, Wang, Kollman, & Case, 2001). All energy parameters, vdW, solvation, and electrostatics, can be scaled and weighted by the user from the defaults provided. Other pairwise-decomposable energy functions can be incorporated into OSPREY; in fact, users are encouraged to improve designs by modifying the energy function.

The user can also specify other design parameters that can significantly improve the accuracy of OSPREY. Amino acid reference energies (Lippow, Wittrup, & Tidor, 2007) account for the energy of a residue in the unfolded state of the protein. These reference energies are important in GMEC-based designs but are not necessary in K^* designs (see following section). Dihedral energy penalties can also be used to prevent continuous flexibility algorithms

from minimizing away from the most frequently observed protein conformations. These and several other design parameters are thoroughly explained in Georgiev, Roberts, et al. (2012).

4.2. Protein design algorithms

The search space in protein design is large, and grows exponentially with the number of protein residues and side-chain rotamers. To search in an efficient manner, OSPREY first reduces the size of the search space through a suite of algorithms based on extensions and generalizations of the dead-end elimination (DEE) algorithm (Desmet, de Maeyer, Hazes, & Lasters, 1992). These algorithms prune the rotamers that, even in the presence of backbone or continuous side-chain flexibility, would not lead to the GMEC or one of the lowest energy conformations (Gainza et al., 2012; Georgiev & Donald, 2007; Georgiev, Keedy, et al., 2008; Georgiev et al., 2006; Georgiev, Lilien, et al., 2008; Lilien et al., 2005). A branch-and-bound algorithm based on A^* (Georgiev, Lilien, et al., 2008; Leach & Lemon, 1998) then traverses the remaining search space and outputs the GMEC and, if desired, a gap-free list of low-energy conformations. The K^* algorithm uses the gap-free list of low-energy conformations to approximate the protein–ligand binding constant. After a design iteration, the results can be reintroduced into OSPREY to search for mutations distal from the active site that will increase the stability of the design (Chen et al., 2009; Fig. 5.2).

Many SCPR algorithms restrict the backbone to a single rigid conformation and the side-chains to discrete, rigid rotameric conformations. The original DEE algorithm (Desmet et al., 1992) falls into this category, and we will refer to it as *rigid DEE* because the rotamers are discrete, rigid geometries that do not include the continuous χ-angle space that immediately surrounds them. OSPREY includes rigid DEE as well as improved variations of DEE that search the continuous χ-angle space that surrounds side-chain rotamers and the continuous ϕ- and ψ-angle space that surrounds the protein backbone (Fig. 5.3, B–E). These continuous-flexibility algorithms compute upper and lower *bounds* on the energy that a backbone or a rotamer could reach after minimization and use these bounds for pruning instead of the rigid energies.

The MinDEE (Georgiev, Lilien, et al., 2008) algorithm extends rigid DEE by including in the search the continuous χ-angle space that immediately surrounds rotamers, and guarantees that no rotamer that can minimize and be part of a minimized GMEC (minGMEC) will be pruned. The *iMinDEE* algorithm (Gainza et al., 2012) improves over MinDEE by

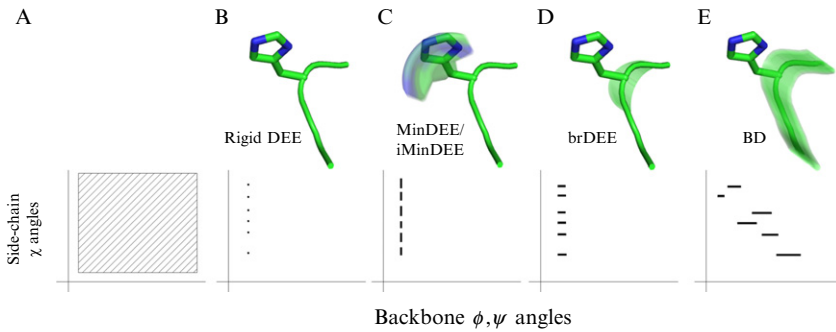

Figure 5.3 *Conceptual illustration of the protein flexibility modeled by* OSPREY*'s algorithms.* Each panel portrays a different algorithm, with the flexibility it models shown above ("blurs" denote movement) and a plot illustrating which dihedral angles it changes shown below. (A) Theoretical "complete" flexibility that could be induced by the introduction of one or more mutations. OSPREY does not yet model "complete" protein flexibility. (B) Rigid DEE: discrete side-chain flexibility with a rigid backbone. (C) MinDEE: continuous side-chain flexibility with a rigid backbone. (D) brDEE: discrete side-chain flexibility and local backbone (backrub) moves. (E) BD: discrete side-chain flexibility and continuous global backbone moves.

pruning orders of magnitude more rotamers with close to the same efficiency as rigid DEE, and also guarantees to find the minGMEC. The backrub DEE (brDEE) algorithm (Georgiev, Keedy, et al., 2008) allows mutants to undergo *backrub motions*, which are entirely local backbone movements that each change the orientation of one C_α–C_β bond vector by performing a small rotation of the surrounding dipeptide (Davis, Arendall, Richardson, & Richardson, 2006). The Backbone DEE (BD) algorithm (Georgiev & Donald, 2007) prunes only rotamers that cannot be part of the GMEC after allowing continuous global backbone movements.

When a ligand is present, it can rotate and translate with respect to the protein (i.e., rigid-body motions), and continuous rotamers can be defined for the ligand. If the ligand is a polypeptide, both the ligand and protein can mutate and rigid DEE, MinDEE, iMinDEE, brDEE, or BD can be used on both molecules.

Each OSPREY algorithm (rigid DEE, MinDEE, iMinDEE, brDEE, and BD) reduces the search space through a stage of DEE-based pruning. In addition, several extensions to the DEE algorithm implemented in OSPREY further improve its pruning capabilities, including generalized DEE (Goldstein, 1994), split flags (Pierce, Spriet, Desmet, & Mayo, 2000),

bounds pruning (Gordon, Hom, Mayo, & Pierce, 2003), and a divide-and-conquer strategy called DACS (Georgiev et al., 2006).

Once the DEE algorithms prune the majority of the conformational space, the remaining space must be searched to find the lowest energy conformation(s). We have implemented a branch-and-bound algorithm based on the A^* algorithm (Georgiev, Lilien, et al., 2008; Leach & Lemon, 1998) that searches conformations in a tree and traverses only the branches that might lead to the lowest energy structure, even in the presence of flexibility. A^* searches the space completely and guarantees to find the optimal answer. When rigid DEE is used, our extension of A^* can also enumerate conformations in order of the lowest energy, which makes it possible to enumerate a gap-free list of low-energy conformations and their sequences. When MinDEE, brDEE, or BD is used, our version of A^* enumerates conformations in order of the lower *bounds* on the energy of each conformation, and can also enumerate a gap-free list of low-energy conformations.

The K^* algorithm (Georgiev, Lilien, et al., 2008; Lilien et al., 2005) uses this list of low-energy conformations to compute a provable ε-approximation to the binding constant (a K^* *score*) with respect to the input model (Fig. 5.2A). A provable ε-approximation algorithm guarantees that the computed binding constant is mathematically accurate up to a user-specified percentage error of ε, with respect to the input model. The ε parameter is specified by the user as the desired accuracy and all computed solutions are guaranteed to be ε-accurate. K^* is efficient because it uses A^* to compute only the reduced set of low-energy conformations that are most likely to be taken on by the protein, the ligand, and the protein–ligand complex. These low-energy conformations are then Boltzmann-weighted and used to approximate the partition function for the unbound and bound states. Because A^* enumerates conformations in order of their low-energy bound, K^* can calculate exactly when each partition function is within an ε-factor of the exact solution and stop the computation. Since the energy of each low-energy conformation is Boltzmann-weighted, K^* must only compute a small percentage of the total number of conformations (Georgiev, Lilien, et al., 2008). Once each partition function is computed, the K^* score is computed by dividing the partition function of the bound state (i.e., the protein–ligand complex) by the partition functions of both unbound states (Georgiev, Lilien, et al., 2008).

After redesigning a protein core, boundary, surface, or active site, it can be beneficial to increase the mutant's stability by further mutating residues that are distal to the redesigned region. However, since proteins can be large, searching

at these distal positions for potential bolstering mutations using algorithms with mathematical guarantees can be an expensive process. To address this issue, OSPREY uses a heuristic SCMF algorithm to find residue positions, that when mutated, might increase the stability of the engineered protein. After SCMF identifies distal residue positions, OSPREY's MinDEE variants and A^* can be used to find mutant residues at those distal positions that stabilize the fold. We have used this approach to increase the stability and catalytic efficiency of a redesigned enzyme using MinDEE/A^* (Chen et al., 2009).

5. EXAMPLE: PREDICTING DRUG RESISTANCE MUTATIONS USING OSPREY

Pharmaceutical companies periodically release new, effective drugs to treat the world's most dangerous infectious diseases. After these drugs are first introduced, pathogens that cause these diseases, such as methicillin-resistant *Staphylococcus aureus* (MRSA), recede temporarily, only to reemerge months or years later as drug-resistant strains. When novel drugs are first discovered, little is known about how pathogens will develop drug resistance. Without that information, drug designers cannot improve existing drugs or develop new ones to target resistant strains until after they spread in the community.

Fortunately, SCPR programs can be used to predict drug resistance that could arise through mutations in enzyme active sites as soon as a drug is developed. This type of application exemplifies the next frontier in SCPR: the design of proteins not only for activity but for specificity. We have used OSPREY to predict resistance mutations in enzyme active sites that confer resistance to competitive inhibitors. Competitive inhibitors are drugs that compete with the natural substrate for binding, and thus inhibit a critical enzyme in a pathogen. Organisms can often evolve active site mutations that maintain catalysis of the substrate but reduce the affinity for the competitive inhibitor. In effect, this resistance mechanism allows the substrate to outcompete the inhibitor for binding to the enzyme. To perform this kind of design, OSPREY must design specificity for the natural substrate over the drug, which is a relatively new and attractive goal for SCPR algorithms.

We have developed a methodology that uses OSPREY to accurately predict resistance-conferring mutations. In Frey et al. (2010), we showed that this methodology can successfully predict resistance mutations to a new antibiotic, UCP111D26M (termed D26M, and shown in Fig. 5.4) that inhibits the MRSA DHFR enzyme. In this section, we describe in depth the methods, empirical rationale, and experimental validation used in Frey et al. (2010).

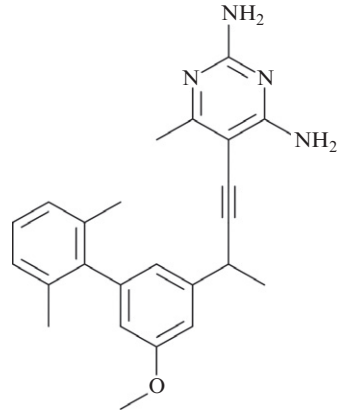

Figure 5.4 *D26M compound.* This compound belongs to a new class of propargyl-linked antifolates. D26M is an effective antibiotic against MRSA.

5.1. Input model in a resistance prediction problem

5.1.1 Initial protein structure

Our approach assumes that mutant sequences that bind the natural substrate well (*positive design*) while binding the competitive inhibitor poorly (*negative design*) will confer drug resistance. This positive/negative design approach needs both a structure of the wild-type enzyme bound to its natural substrate and a structure of the wild-type enzyme bound to the competitive inhibitor drug. In general, higher quality input structures should lead to more accurate results. However, if the relevant high-quality structure is not available, modeling based on a related structure can be used. In the case of MRSA DHFR resistance prediction, a structure of the wild-type enzyme bound to D26M (Fig. 5.4) was determined (PDB ID: 3F0Q) for negative design, and the structure of the F98Y DHFR mutant bound to folate was used to generate a model of the wild type for positive design.

5.1.2 Protein flexibility

The structural changes caused by resistance mutations must preserve the catalytic activity of the enzyme. Thus, we expect that successful resistance mutations to the active site of MRSA DHFR will cause small conformational changes in the protein structure. The BD, brDEE, and MinDEE algorithms can model these conformational changes. However, in the case of DHFR, we chose to model continuous side-chain flexibility (MinDEE) in addition to ligand flexibility (described below) because we expected that

the protein backbone surrounding the active site would remain relatively rigid after the introduction of resistance mutations to maintain catalytic activity. Thus, we assumed that the protein's side-chain interactions with the ligand would determine resistance.

The ligand (both D26M and dihydrofolate) can potentially bind DHFR in a large number of conformations (see Section 4.1). However, choosing the D26M negative design rotamers is especially important because missing low-energy binding conformations between D26M and DHFR could result in a failure of the negative design (see Section 2.4). We chose 512 generalized rotamers for D26M over four rotatable dihedrals. Based on structural information, in addition to the reasons described in Section 2.4, we believe that the catalytic activity of DHFR is highly optimized for a few specific binding conformations of dihydrofolate. We chose 12 generalized rotamers over 10 rotatable dihedrals for dihydrofolate. Each of these generalized rotamers was treated as a continuous rotamer, meaning that each flexible dihedral was allowed to minimize by rotating its torsional dihedrals $\pm 9°$. Additionally, continuous rigid-body motions (rotation and translation) were allowed for each ligand within the active site. The allowed conformation space, C_I, for the inhibitor D26M is completely described by these three modeling choices. Namely, C_I is defined by the chosen generalized rotamers, the allowed continuous minimization around these rotamers, and the allowed rigid-body rotation and translation. The conformation space for DHFR and dihydrofolate, C_P, and C_S, respectively, were similarly defined.

5.1.3 Sequence space

For this study, we only allowed residues in the active site with direct contact to the drug to mutate. Specifically, only residues L5, V6, L20, D27, L28, V31, T46, I50, L54, and F92 were allowed to mutate and/or change conformation. The allowed amino acid mutations were selected based on the wild-type amino acid and correlation with other DHFR species. Residues 5, 6, 20, 28, 31, 50, and 92 were allowed to mutate to Ala, Val, Leu, Ile, Met, Phe, Trp, and Tyr. Residue D27 was allowed to maintain its identity or mutate to Glu, while residues T46 and L54 were allowed to change their conformation but not mutate. Only mutant sequences representing single- or double-point amino acid mutations were allowed to mimic resistance mutations that could evolve naturally. This resulted in a total sequence space of 1173 mutants to the wild type.

5.2. Results

OSPREY computed the K^* score for the 1173 DHFR single- and double-point mutants bound to dihydrofolate (positive design) or D26M (negative design). Positive and negative design computations were performed separately and then combined.

Each sequence was ranked by its K^* ratio: the K^* score of the positive design divided by the K^* score of the negative design. Sequences with a negative design score of zero were ranked solely by the positive design score, namely by their binding affinity for dihydrofolate. Since the K^* score approximates a K_A for the protein–ligand complex, a higher K^* score represents better binding. Therefore, a mutant DHFR sequence with a high positive design score and negative design score of zero is predicted to destabilize binding to D26M versus dihydrofolate. All of the top 10 mutations were predicted to bind dihydrofolate, while disrupting the binding of D26M in the conformations specified by C_I for negative design. Four of the top 10 predicted resistance mutants were tested experimentally: (1) V31Y/F92I, (3) V31I/F92S, (7) V31F/F92L, and (9) I50W/F92S. Mutants (1), (3), and (7) yielded biologically successful results. Specifically, these mutants maintained catalytic activity and had a lower affinity for the D26M drug. The top-ranked mutant sequence, V31Y/F92I, conferred the greatest decrease in binding to D26M, an 18-fold loss.

In addition to confirming that the top mutants were biologically successful, it is important to evaluate the success of the underlying computational predictions. The success of the computational prediction relies entirely on the accuracy of the input model's definition of conformation space and energy function because OSPREY guarantees to find the optimal conformation(s) given the input model. Moreover, the goal of the computational negative design was to find protein sequences that cannot bind any conformation in the D26M ligand's conformational space, C_I. Note that if, for example, C_I does not accurately represent the conformational energy landscape, it is possible that the computational prediction would successfully exclude binding in C_I but result in biologically failed designs because the protein could bind to a D26M conformation outside of C_I. Of course, the issue of defining the input conformational space C_I arises in any protein design algorithm. But since K^*'s search guarantees completeness, we can rule out failures of optimization and attribute any discrepancies between predictions and experimental measurements exclusively to the input model, which includes C_I. This guarantee is crucial because if any low-energy

conformation was missed, there would likely be a low-energy D26M binding mode resulting in designs that fail both computationally and biologically.

To analyze the binding mechanism of the top resistant DHFR mutant, V31Y/F92I, the crystal structure of this mutant bound to D26M was determined (PDB ID: 3LG4). The structure showed that D26M occupies the active site of mutant V31Y/F92I with 50% occupancy. In contrast, wild-type DHFR binds D26M with full occupancy, which suggests that poor occupancy in V31Y/F92I is caused by reduced ligand binding. D26M binds the V31Y/F92I mutant weakly in a conformation that was not in C_I, which demonstrates the difficulty for the user to determine *a priori* all conformations of a drug to input into the design protocol. However, the predicted energy of this new conformation bound to DHFR V31Y/F92I was very similar to the lowest energy conformations in the predicted K^* ensemble. This demonstrates that C_I accurately covered the energy landscape and OSPREY successfully found a mutant protein sequence that could destabilize binding to D26M. Since OSPREY uses provable algorithms, it can guarantee that no conformations in C_I would bind with a better energy than what was found in the K^* ensemble. This is confirmed by the conformation of D26M in the V31Y/F92I crystal structure.

5.2.1 Effect of limiting flexibility on mutation predictions

We have argued that ensembles, flexibility, and provability are essential for both positive design and negative design (see Section 2). We have also shown that using ensembles can have important consequences on binding affinity rankings (Roberts et al., 2012), and that provably modeling continuous flexibility in protein core redesign is critical for accuracy (Gainza et al., 2012). Similarly, we now show the importance of improving flexibility in an additional example. For this example, we performed predictions almost identical to those in Frey et al. (2010) with one crucial change: we limited protein flexibility to discrete rotamers for both the rotamer and ligands, and disabled continuous rigid-body motions. Using such a discrete, rigid model is very common in the protein design field.

Rigid DEE/A^*/K^* was used to compute a positive and a negative design K^* score for all 1173 DHFR mutant sequences. We found that all of the top 10 ranked mutants predicted by MinDEE/K^* in Frey et al. (2010) now received radically different scores. They all had a positive design score of 0 in the rigid, discrete designs because rigid DEE/A^*/K^* could not find any low-energy binding conformation of dihydrofolate for these sequences. Thus, the rigid model incorrectly predicted that the experimentally tested mutants of Frey et al. (2010) would not bind dihydrofolate. These results suggest that rigid rotamers not only fail to cover the entire rotamer space

but also are sensitive to small changes in torsional dihedrals and rigid-body motions. Consequently, few mutants are predicted by rigid DEE/A^*/K^* to bind dihydrofolate or D26M, which is manifestly wrong in light of the MinDEE/A^*/K^* results and our experimental validation.

6. FUTURE DIRECTIONS AND AVAILABILITY

We have presented an overview of OSPREY, a comprehensive open-source SCPR suite. OSPREY has been in continuous development over the last decade and both the algorithms and functionality will continue to improve. The variety of the prospective designs where OSPREY has been applied, and the suite of sophisticated algorithms with which they were created, suggest that OSPREY can be adapted to facilitate protein engineering in a number of settings. Several enhancements of OSPREY are planned, including support for explicit water-mediated hydrogen bonds, concerted backbone and side-chain continuous flexibility (Hallen, Keedy, & Donald, 2013), protein loop modeling (Tripathy, Zeng, Zhou, & Donald, 2012), and RNA rotamers.

OSPREY is available under a GNU Lesser General Public License. As such, the source code is provided as part of the distribution. We encourage users to customize and/or improve it. Specifically, OSPREY provides a platform for the development of new algorithms and new protein design methodology, beyond the features we have presented here. All software is implemented in Java, with parallel computing capabilities provided by mpiJava (Baker, Carpenter, Hoon Ko, & Li, 1998). OSPREY can run on any operating system that supports Java, but we recommend a computing cluster to run OSPREY to distribute and reduce the computation time.

ACKNOWLEDGMENTS

This work was supported by the following NIH grants to B. R. D.: R01 GM-78031, R01 GM-65982, and T32 GM-71340. D. A. K., J. S. R., and D. C. R. were supported by grant NIH R01-GM073930 to D. C. R.

REFERENCES

Baker, M., Carpenter, B., Hoon Ko, S., & Li, X. (1998). mpiJava: A Java interface to MPI. First UK Workshop on Java for High Performance Network Computing.

Brooks, B. R., Brooks, C. L., III, Mackerell, A. D., Jr., Nilsson, L., Petrella, R. J., Roux, B., et al. (2009). CHARMM: The biomolecular simulation program. *Journal of Computational Chemistry*, *30*, 1545–1614.

Chen, C. Y., Georgiev, I., Anderson, A. C., & Donald, B. R. (2009). Computational structure-based redesign of enzyme activity. *Proceedings of the National Academy of Sciences of the United States of America*, *106*, 3764–3769.

Chou, J. J., Case, D. A., & Bax, A. (2003). Insights into the mobility of methyl-bearing side chains in proteins from (3)J(CC) and (3)J(CN) couplings. *Journal of the American Chemical Society*, *125*, 8959–8966.

Davis, I. W., Arendall, W. B., Richardson, D. C., & Richardson, J. S. (2006). The backrub motion: How protein backbone shrugs when a sidechain dances. *Structure*, *14*, 265–274.

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

Donald, B. R. (2011). *Algorithms in structural molecular biology*. Cambridge, MA: MIT Press.

Frey, K. M., Georgiev, I., Donald, B. R., & Anderson, A. C. (2010). Predicting resistance mutations using protein design algorithms. *Proceedings of the National Academy of Sciences of the United States of America*, *107*, 13707–13712.

Gainza, P., Roberts, K. E., & Donald, B. R. (2012). Protein design using continuous rotamers. *PLoS Computational Biology*, *8*(1), e1002335.

Georgiev, I., Acharya, P., Schmidt, S. D., Li, Y., Wycuff, D., Ofek, G., et al. (2012). Design of epitope-specific probes for sera analysis and antibody isolation. *Retrovirology*, *9* (Suppl. 2):P50. PMC id: PMC3442034.

Georgiev, I., & Donald, B. R. (2007). Dead-end elimination with backbone flexibility. *Bioinformatics*, *23*, i185–i194.

Georgiev, I., Keedy, D., Richardson, J., Richardson, D., & Donald, B. R. (2008). Algorithm for backrub motions in protein design. *Bioinformatics*, *24*, i196–i1204.

Georgiev, I., Lilien, R. H., & Donald, B. R. (2006). Improved pruning algorithms and divide-and-conquer strategies for dead-end elimination, with application to protein design. *Bioinformatics*, *22*, e174–e183.

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

Georgiev, I., Roberts, K.E., Gainza, P., & Donald, B.R. (2012). OSPREY v2.0 manual. Dept. of Computer Science, Duke University http://www.cs.duke.edu/donaldlab/software/osprey/osprey2.0.pdf.

Gilson, M. K., Given, J. A., Bush, B. L., & McCammon, J. A. (1997). The statistical-thermodynamic basis for computation of binding affinities: A critical review. *Biophysical Journal*, *72*, 1047–1069.

Goldstein, R. (1994). Efficient rotamer elimination applied to protein side-chains and related spin glasses. *Biophysical Journal*, *66*, 1335–1340.

Gorczynski, M. J., Grembecka, J., Zhou, Y., Kong, Y., Roudaia, L., Douvas, M. G., et al. (2007). Allosteric inhibition of the protein–protein interaction between the leukemia-associated proteins RUNX1 and CBF β. *Chemistry & Biology*, *14*, 1186–1197.

Gordon, D. B., Hom, G. K., Mayo, S. L., & Pierce, N. A. (2003). Exact rotamer optimization for protein design. *Journal of Computational Chemistry*, *24*, 232–243.

Hallen, M. A., Keedy, D. A., & Donald, B. R. (2013). Dead-End Elimination with Perturbations ('DEEPer'): A provable protein design algorithm with continuous sidechain and backbone flexibility. *Proteins*, *81*(1), 18–39.

Hu, X., & Kuhlman, B. (2006). Protein design simulations suggest that side-chain conformational entropy is not a strong determinant of amino acid environmental preferences. *Proteins*, *62*, 739–748.

Lazaridis, T., & Karplus, M. (1999). Discrimination of the native from misfolded protein models with an energy function including implicit solvation. *Journal of Molecular Biology*, *288*, 477–487.

Leach, A. R., & Lemon, A. P. (1998). Exploring the conformational space of protein side chains using dead-end elimination and the A* algorithm. *Proteins*, *33*, 227–239.

Lilien, R. H., Stevens, B. W., Anderson, A. C., & Donald, B. R. (2005). A novel ensemble-based scoring and search algorithm for protein redesign and its application to modify the substrate specificity of the gramicidin synthetase a phenylalanine adenylation enzyme. *Journal of Computational Biology*, *12*, 740–761.

Lippow, S. M., Wittrup, K. D., & Tidor, B. (2007). Computational design of antibody-affinity improvement beyond in vivo maturation. *Nature Biotechnology*, *25*, 1171–1176.

Lovell, S. C., Word, J. M., Richardson, J. S., & Richardson, D. C. (2000). The penultimate rotamer library. *Proteins*, *40*, 389–408.

Pearlman, D. A., Case, D. A., Caldwell, J. W., Ross, W. S., Cheatham, T. E., DeBolt, S., et al. (1995). AMBER, a package of computer programs for applying molecular mechanics, normal mode analysis, molecular dynamics and free energy calculations to simulate the structural and energetic properties of molecules. *Computer Physics Communications*, *91*, 1–41.

Pierce, N. A., Spriet, J. A., Desmet, J., & Mayo, S. L. (2000). Conformational splitting: A more powerful criterion for dead-end elimination. *Journal of Computational Chemistry*, *21*, 999–1009.

Reza, F. (2010). Computational molecular engineering of nucleic acid binding proteins and enzymes. Doctoral Dissertation, Duke University.

Roberts, K. E., Cushing, P. R., Boisguerin, P., Madden, D. R., & Donald, B. R. (2012). Design of a PDZ domain peptide inhibitor that rescues CFTR activity. *PLoS Computational Biology*, *8*(4), e1002477. http://dx.doi.org/10.1371/journal.pcbi.1002477.

Teague, S. J. (2003). Implications of protein flexibility for drug discovery. *Nature Reviews Drug Discovery*, *2*, 527–541.

Tripathy, C., Zeng, J., Zhou, P., & Donald, B. R. (2012). Protein loop closure using orientational restraints from NMR data. *Proteins*, *80*, 433–453.

Wang, J., Wang, W., Kollman, P., & Case, D. (2001). Antechamber, an accessory software package for molecular mechanical calculations. *Abstracts of Papers of the American Chemical Society*, *222*, U403.

Zeng, J., Roberts, K. E., Zhou, P., & Donald, B. R. (2011). A Bayesian approach for determining protein side-chain rotamer conformations using unassigned NOE data. *Journal of Computational Biology*, *18*, 1661–1679.

Zeng, J., Zhou, P., & Donald, B. R. (2011). Protein side-chain resonance assignment and NOE assignment using RDC-defined backbones without TOCSY data. *Journal of Biomolecular NMR*, *50*, 371–395.