JOURNAL OF COMPUTATIONAL BIOLOGY
Volume 25, Number 7, 2018

Pp. 726–739
DOI: 10.1089/cmb.2017.0267

BBK (Branch and Bound Over K*):* A Provable and Efficient Ensemble-Based Protein Design Algorithm to Optimize Stability and Binding Affinity Over Large Sequence Spaces

ADEGOKE A. OJEWOLE,[1,2,†] JONATHAN D. JOU,[1,†]
VANCE G. FOWLER,[3] and BRUCE R. DONALD[1,4,*]

ABSTRACT

Computational protein design (CPD) algorithms that compute binding affinity, K_a, search for sequences with an energetically favorable free energy of binding. Recent work shows that three principles improve the biological accuracy of CPD: *ensemble-based design*, *continuous flexibility* of backbone and side-chain conformations, and *provable guarantees of accuracy* with respect to the input. However, previous methods that use all three design principles are *single-sequence* (SS) algorithms, which are very costly: linear in the number of sequences and thus exponential in the number of simultaneously mutable residues. To address this computational challenge, we introduce *BBK, a new CPD algorithm whose key innovation is the *multisequence* (MS) bound: *BBK** efficiently computes a single provable upper bound to approximate K_a for *a combinatorial number of sequences*, and avoids SS computation for all provably suboptimal sequences. Thus, to our knowledge, *BBK** is the first provable, ensemble-based CPD algorithm to run in time *sublinear* in the number of sequences. Computational experiments on 204 protein design problems show that *BBK** finds the tightest binding sequences while approximating K_a for up to 10^5-fold fewer sequences than the previous state-of-the-art algorithms, which require exhaustive enumeration of sequences. Furthermore, for 51 protein–ligand design problems, *BBK** provably approximates K_a up to 1982-fold faster than the previous state-of-the-art iMinDEE/A^*/K^* algorithm. Therefore, *BBK** not only accelerates protein designs that are possible with previous provable algorithms, but also efficiently performs designs that are too large for previous methods.**

Keywords: molecular ensembles, OSPREY, predicting binding affinity, protein design, structural biology, sublinear algorithms.

[1]Department of Computer Science, Duke University, Durham, North Carolina.
[2]Computational Biology and Bioinformatics Program, Duke University, Durham, North Carolina.
[3]Division of Infectious Diseases, Duke University Medical Center, Durham, North Carolina.
[4]Department of Biochemistry, Duke University Medical Center, Durham North Carolina.
†These authors contributed equally to the work.
*Corresponding author.

1. INTRODUCTION

Protein design is the prediction of protein sequences with desired biochemical functions, which often involve binding to a target ligand. Computational protein design (CPD) casts functional design into a structural optimization problem whose goal is to find amino acid sequences that fold into specified three-dimensional structures. Protein design algorithms search a space defined by a biophysical *input model*, which defines the sequence and structural search space (i.e., the input structure, allowed amino acid mutations, and allowed protein flexibility); the optimization objective (e.g., design for binding affinity); and the energy function (Boas and Harbury, 2007). Protein design algorithms (Donald, 2011; Gainza et al., 2016) have successfully predicted protein sequences that fold and bind desired targets *in vitro* and *in vivo*. For example, these algorithms have been used, with experimental validation, to predict drug resistance (Frey et al., 2010; Reeve et al., 2015; Ojewole et al., 2017a) and to design enzymes (Lilien et al., 2005; Stevens et al., 2006; Georgiev and Donald, 2007; Chen et al., 2009), new drugs (Gorczynski et al., 2007), inhibitors of protein–protein interactions (Georgiev et al., 2008; Roberts et al., 2012), epitope-specific antibody probes (Georgiev et al., 2012), and even neutralizing antibodies (Georgiev et al., 2014; Rudicell et al., 2014).

Computational methods can potentially search a large number of sequences to predict the proteins that bind most tightly to a target ligand in less time and with fewer resources than *in vitro* methods such as phage display (Carmen and Jermutus, 2002; Pál et al., 2006). However, four computational challenges have prevented protein design algorithms from realizing this potential. First, for each binding interface, an exponentially large number of conformations in each binding partner's ensemble must be pruned or considered to accurately predict binding affinity (Gilson et al., 1997; Lilien et al., 2005; Georgiev et al., 2008; Donald, 2011). Second, for each sequence, finding the lowest energy conformations that most influence binding affinity is NP-hard (Pierce and Winfree, 2002; Kingsford et al., 2005; Xu, 2005; Xu and Berger, 2006; Peng et al., 2015), making algorithms that guarantee optimality expensive for larger designs. Third, mutating a protein sequence induces conformational changes in the protein structure. Since such conformational changes occur over many continuous degrees of freedom, algorithms that model continuous flexibility must search over a large continuous conformation space. Fourth, the number of protein sequences (i.e., the *sequence space*) grows exponentially with the number of simultaneously mutable residues in the design. Therefore, previous algorithms either focus on accurately modeling smaller designs or attempt larger designs by making simplifications that (1) ignore the ensemble nature of proteins, (2) disregard continuous conformational flexibility, or (3) return heuristic solutions with no guarantees. A discussion of these simplifications and their ramifications for protein design follows [see also Supplementary Information (SI) section 2 (Ojewole et al., 2017b)].

Global Minimum Energy Conformation (GMEC)-based algorithms (Desmet et al., 1992; Gainza et al., 2012; Roberts and Donald, 2015; Hallen and Donald, 2016; Viricel et al., 2016) assume that the lowest energy conformation accurately predicts binding affinity. However, GMEC-based designs cannot accurately model entropic change due to binding (Fleishman et al., 2011) and can disproportionately favor sequences with energetically favorable GMECs over sequences with tight binding affinity (Lilien et al., 2005; Georgiev et al., 2008; Chen et al., 2009; Sciretti et al., 2009; Roberts et al., 2012; Silver et al., 2013). Many protein design algorithms (Leaver-Fay et al., 2011; Traoré et al., 2013, 2016; Simoncini et al., 2015; Viricel et al., 2016) rely on a simplified, discrete model of side-chain flexibility. However, discrete rotamers model a small subset of protein energetics, which are sensitive to small atomic movements not permitted by the discrete model. To overcome this limitation, researchers have developed provable algorithms (Georgiev et al., 2008; Gainza et al., 2012; Roberts et al., 2012; Hallen et al., 2015, 2016; Roberts and Donald, 2015) that incorporate continuous side-chain flexibility (Georgiev et al., 2008; Gainza et al., 2012). Another important aspect of design algorithms is the quality of the computed results. Whereas GMEC-based design is NP-hard (Pierce and Winfree, 2002; Kingsford et al., 2005; Xu, 2005; Xu and Berger, 2006), computation of thermodynamic ensembles and associated partition functions is #P-hard (Valiant, 1979; Nisonoff, 2015; Viricel et al., 2016). Provable protein design algorithms return either the optimal sequences or conformations (Desmet et al., 1992; Leach and Lemon, 1998; Gainza et al., 2012; Hallen et al., 2013; Traoré et al., 2013, 2016; Roberts and Donald, 2015; Simoncini et al., 2015; Viricel et al., 2016) with respect to the input model or return provably good approximate solutions (Lilien et al., 2005; Georgiev et al., 2008; Roberts et al., 2012; Hallen et al., 2015, 2016). Nonprovable algorithms such as Metropolis Monte Carlo methods (Lee and Levitt, 1991; Kuhlman and Baker, 2000; Leaver-Fay et al., 2011) instead use stochastic methods to rapidly sample the space described by the input

model. These algorithms return solutions without any guarantees. Thus, ensemble-based design, a realistic model of structural flexibility, and provable optimality of the computed sequences with respect to the input model improve the predictive power of protein design algorithms. However, each of these design principles also increases the cost of protein design (as a function of the number of sequences).

Single-sequence (SS) algorithms, which explicitly evaluate each possible sequence, are powerful and versatile. Molecular dynamics (Leech et al., 1996; Zheng et al., 2008), for instance, is frequently applied to design for binding affinity. The approach models ensembles and continuous flexibility. Provable algorithms have also been developed to model these two phenomena. The *K** algorithm (Lilien et al., 2005; Georgiev et al., 2008; Roberts et al., 2012) in OSPREY (Lilien et al., 2005; Georgiev et al., 2006, 2008; Georgiev and Donald, 2007; Gainza et al., 2012, 2013; Roberts et al., 2012; Hallen et al., 2013, 2015, 2016; Roberts and Donald, 2015; Hallen and Donald, 2016; Jou et al., 2016) uses a combination of dead-end elimination pruning (Georgiev et al., 2008; Gainza et al., 2012) and *A** (Hart et al., 1968; Leach and Lemon, 1998; Roberts et al., 2015) gap-free conformation enumeration to provably and efficiently approximate K_a. *K** and all previous provable ensemble-based algorithms that model continuous side-chain flexibility (Hallen et al., 2015, 2016) are SS algorithms. The empirical and asymptotic runtime complexity of SS algorithms is linear in the number of possible sequences, and, therefore, exponential in the number of mutable residues. Thus, designs with many mutable residues rapidly become intractable when using SS algorithms. The COMETS algorithm (Hallen and Donald, 2016) in OSPREY is the only provable multistate design algorithm that is more efficient than SS algorithms. However, its binding predictions are GMEC based rather than ensemble based. Additional background and references are located in **SI** section 2 (Ojewole et al., 2017b).

To efficiently search large sequence spaces while retaining all the benefits of provable guarantees, ensemble-based design, and continuous side-chain flexibility, we present a new, provable algorithm: Branch and Bound over *K** (*BBK**). The key innovation of *BBK** is the *multisequence* (MS) bound: *BBK** efficiently computes upper and lower bounds on the binding affinities of *partial sequences*, which are shared by a combinatorial number of full sequences. By avoiding costly SS computation, *BBK** runs in time *sublinear* in the number of sequences. To our knowledge, *BBK** is the first provable, ensemble-based algorithm to do so. *BBK** not only avoids explicitly computing all possible sequences, but also provably and efficiently enumerates sequences in a gap-free decreasing order of binding affinity. Therefore, *BBK** provides a vast performance improvement over the previous state-of-the-art, by not only accelerating protein designs that were possible with previous provable algorithms but also efficiently handling large designs that previous algorithms could not compute in a reasonable amount of time.

By presenting *BBK**, our article makes the following contributions:

(1) A novel, ensemble-based algorithm that provably computes the same results as the previous state of the art (exhaustive search over sequences) but is empirically combinatorially faster, returns a gap-free list of sequences in decreasing order of binding affinity, and runs in time sublinear in the number of sequences.
(2) Proofs of correctness for MS bounds, a key innovation in *BBK**.
(3) A new two-pass bound that more efficiently computes a provable ε-approximation to the desired partition functions.
(4) Totally 255 protein designs showing that *BBK** approximates binding affinity for full sequences up to 1982-fold faster than the best previous algorithm and that *BBK** computes the best binding sequences in a large sequence space up to 10^5-fold more efficiently than exhaustive search.
(5) Support for both continuous side-chain and backbone flexibility, demonstrating the ability of *BBK** to handle multiple modes of protein flexibility in addition to large conformation and sequence spaces.
(6) An implementation of *BBK** in our laboratory's open-source OSPREY (Lilien et al., 2005; Georgiev et al., 2006, 2008; Georgiev and Donald, 2007; Gainza et al., 2012, 2013; Roberts et al., 2012; Hallen et al., 2013, 2015, 2016; Roberts and Donald, 2015; Hallen and Donald, 2016; Jou et al., 2016) protein design software package, available for download as free software.

2. COMPUTING THE PARTITION FUNCTION

To successfully design for improved binding affinity K_a, design algorithms must consider the energy of more than just the GMEC. In particular, all algorithms that design for improved K_a optimize the ratio of partition

function Z for the bound and unbound states of the protein and ligand [Eq. (2)]. Protein design can thus be cast as an optimization problem. For an n-residue protein design problem with at most a amino acids per mutable residue, let P, L, and PL denote the unbound protein, unbound ligand, and bound protein–ligand complex, respectively. For each sequence s, let $\mathbf{Q}(s)$ be the set of discrete rigid rotamer conformations defined by the allowed amino acids for each mutable residue of s. For a rigid rotamer conformation c, let E_X be a pairwise energy function with respect to input structure X, which may be one of P, L, or PL. In particular, we consider the case of design with *continuous rotamers* (Georgiev et al., 2008; Gainza et al., 2012). We define $E_X(c)$ to be the energy of c for structure X after energy minimizing the side chains of mutable residues.

2.1. K*

We define the Boltzmann-weighted partition function $Z_X(s)$ as

$$Z_X(s) = \sum_{c \in \mathbf{Q}(s)} \exp\left(-E_X(c)/RT\right). \tag{1}$$

We define the K^* score, a partition function ratio that approximates binding affinity, K_a, as

$$K^*(s) = \frac{Z_{PL}(s)}{Z_P(s) Z_L(s)}. \tag{2}$$

As stated in Section 1, the K^* approximation and, by extension, the full partition function are #P-hard to compute (Valiant, 1979; Nisonoff, 2015; Viricel et al., 2016). Therefore, researchers have not only developed heuristic algorithms that rapidly compute loose partition function bounds, but also developed efficient, provable algorithms that compute ε-approximations to the partition function. Probabilistic algorithms bound the partition function either provably (Wainwright et al., 2013) or nonprovably (Fromer and Yanover, 2008). An efficient ε-approximation to $Z_X(s)$ is computed in Lilien et al. (2005), Georgiev et al. (2008), and Roberts et al. (2012). However, these methods are designed to compute partition functions for SSs. For an n-residue design with at most t possible amino acids at each residue and q rotamers per amino acid, provable SS methods must compute or bound the partition functions of all t^n sequences, each with q^n conformations. Thus, previous SS algorithms for protein design for binding affinity take time exponential in n ($\mathcal{O}(t^n)$) when computing the sequence with the best predicted binding affinity.

Therefore, to provably find the best binding sequences, new efficient provable algorithms are needed to search over an exponentially large sequence space, in which each sequence represents an exponentially large conformation space. *BBK** addresses this need. *BBK** compares *partial sequences* (for which some mutable residues have not been assigned an amino acid identity) without computing the partition functions for all full sequences (which assign a specific amino acid to each mutable residue). *BBK** computes bounds on the free energies of partial sequences, and avoids enumerating conformations from sequences with poor binding affinity by pruning sequences during search. As we describe in Section 3, pruning these sequences circumvents prohibitive computational costs required to compute many SS K^* scores.

3. *A** SEARCH OVER SEQUENCES, WITH MULTISEQUENCE BOUNDS

It may at first seem counter-intuitive to compute the sequence with optimal binding affinity, along with its predicted K^* score, without explicitly computing the K^* scores of all possible sequences. Indeed, all previous ensemble-based provable methods, as well as many heuristic methods, are SS methods: they must individually evaluate and compare each sequence to provably return the optimal sequence. In contrast, *BBK** bounds the K^* ratios of a combinatorial number of sequences efficiently and can prune these sequences without computing any SS bounds. The key to this improvement is the observation that a partial sequence s' with poor predicted binding affinity adversely affects the K^* score of the combinatorial set of sequences that contain s'. That is, the best possible K^* score consistent with s' limits the K^* ratio of all sequences consistent with s'. Henceforth, we refer to a bound on the binding affinity for a sequence as a *bound on the sequence*. To compute a bound on all sequences consistent with s', *BBK** computes the partition function for an ensemble that contains conformations from multi-sequence. Figure 1 shows the difference between SS and MS ensembles. The K^* ratio of a MS ensemble is a provable upper bound on the best possible K^* ratio of all sequences that contain s'. We show this MS bound is not only cheaper to

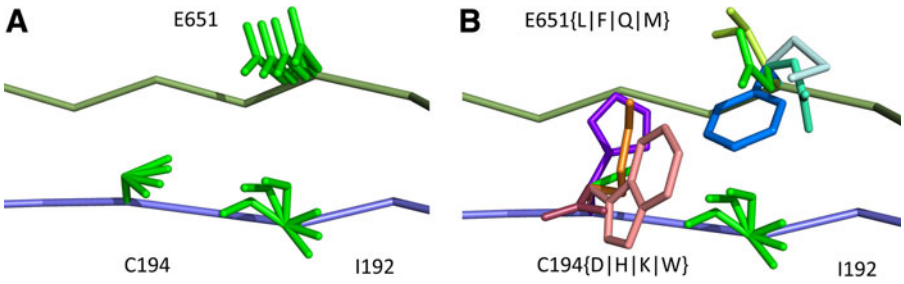

FIG. 1. A toy protein design problem in which conformational ensembles **(A)** and optimal mutations **(B)** must be computed at three residues. Residues of the fibronectin F1 module (Fn, blue ribbon) and of a fragment of *Staphylococcus aureus* fibronectin binding protein A (FNBPA-5, green ribbon) are shown (PDB id: 2RL0). Side-chain conformations, labeled with amino acid identity, are also shown per residue. **(A)** Previous provable methods require a *fully defined sequence* to compute an SS ε-approximation bound on binding affinity [i.e., a K^* score, Eq. (2)]. **(B)** A key innovation in this article is the MS *bound* for binding affinity in protein design. An MS bound is a provable bound on the binding affinity of a *partial sequence*. Unassigned residues, whose amino acid identities are not defined by the partial sequence, adopt side-chain conformations from multiple amino acids, shown as the blue, purple, pink, and light blue ensembles. Thus, an MS bound is a provable upper bound on the binding affinity of all sequences containing that partial sequence, and is obtained without computing any SS bounds. (The full analysis of the Fn:FNBPA-5 design problem is described in Section 4.3). MS, multi-sequence; SS, single sequence.

compute, but it also allows *BBK** to compare a combinatorial number of sequences without computing any SS bounds. By bounding every possible sequence consistent with a partial sequence, *BBK** can provably eliminate those sequences, and prune a combinatorial number of sequences without performing any SS computation. Figure 2 illustrates the combinatorial speedup provided by MS bound pruning, whereby pruning the partial sequence obviates computation of all SSs containing the partial sequence. MS bounds are not limited to pruning based on binding affinity, however. They are also useful for pruning combinations of sequences whose *favorable K** scores are due to energetically *unfavorable* unbound states that cannot fold into ligand-binding poses. SS methods compute binding affinity for each of these sequences. In contrast, *BBK** prunes these partial sequences using MS bounds, thereby avoiding many costly SS computations. Details are provided in section A.7 of the **SI** (Ojewole et al., 2017b).

The improvement of *BBK** over SS methods can be measured using *cost per sequence*. We show the improvement is threefold: *BBK** (a) reduces the cost to compute a bound on a combinatorial number of sequences, (b) eliminates all computational costs once a sequence is pruned, and (c) when it must compute a bound for an SS, computes a bound that is in many cases cheaper than the bounds computed by previous SS algorithms. To guarantee that the first sequence returned is optimal, an algorithm must either compute or bound the partition function for all possible sequences. Previous provable algorithms compute a provable SS bound of the partition function, called an ε-approximation (SS-ε bound), for each sequence (Lilien et al., 2005; Georgiev et al., 2008; Roberts et al., 2012). These SS-ε bounds are guaranteed to be within a user-specified ε of the K^* score for a sequence. *BBK** also provably returns the optimal sequences, but does so without enumerating all possible sequences. Instead of SS-ε bounds, *BBK** computes an MS bound, which is an upper bound on the best possible K^* score of multiple sequences that share a common partial sequence.

We will now compare the cost of bounding sequences with SS algorithms with the cost with *BBK**. Consider an n-residue protein design: we are given an initial partial sequence s', which fixes amino acid identity (but not the rotamer) for a residues, and u residues do not have a fixed amino acid identity ($a+u=n$). If the design problem allows at most t amino acids per unassigned (u) residue and at most q rotamers for any amino acid, there are t^u sequences containing s' and q^a partial conformations defined by s'. A complete sequence would still have q^n conformations, and computing the energy of a conformation takes $\mathcal{O}(n^2)$ time using a pairwise energy function. Thus, an SS algorithm would spend $\mathcal{O}(t^u q^n n^2)$ worst-case time individually computing the K^* scores of all t^u sequences. In contrast, the cost of an MS bound is $\mathcal{O}(q^a(a^2+q^2t^2un))$, which includes $\mathcal{O}(q^a a^2)$ time to compute the pairwise energy of the a assigned residues of all q^a partial conformations, and $\mathcal{O}(q^{a+2}t^2un)$ time to compute a bound on the energy of each partial conformation. By reducing two exponentials from t^u to t^2, and from q^n to q^{a+2}, *BBK** computes an MS bound in time sublinear in the number (t^u) of sequences. The cost to compute a single provable MS bound (that holds for all t^u sequences) is, therefore, significantly smaller than the cost to compute t^u SS bounds.

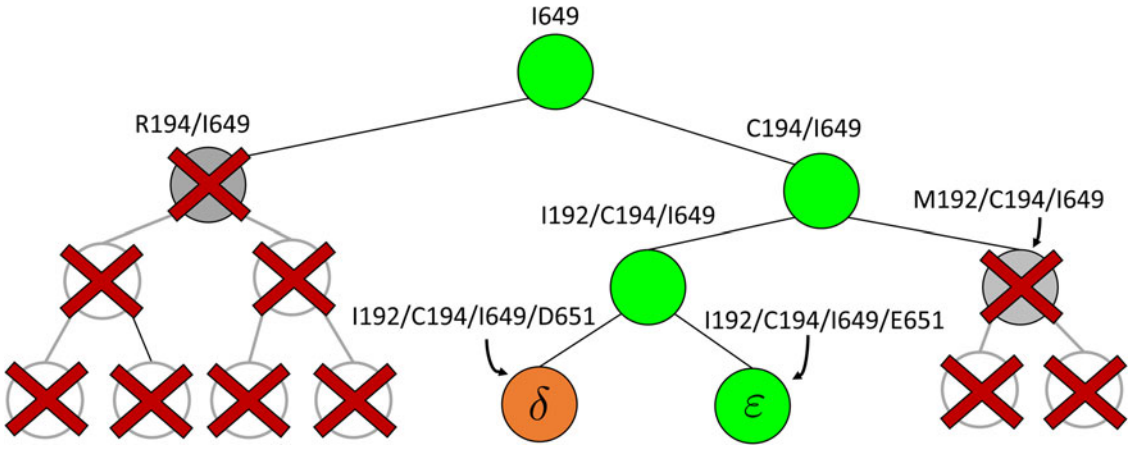

FIG. 2. *BBK** pruning efficiently explores the sequence space. An example design of residues 192 and 194 of the fourth fibronectin F1 module and residues 649 and 651 of a fragment *Staphylococcus aureus* fibronectin binding protein a-5 is shown (Fig. 1, PDB id: 2RL0). As *BBK** searches the sequence space (tree above) its MS bounds provably prunes subtrees from the sequence space. All sequences containing R194/I649 are pruned (red crosses) after computing exactly one MS bound: the bound on the partial sequence R194/I649, which is an upper bound for all sequences containing R194/I649. Sequences containing M192/C194/I649 are pruned (red crosses) after computing only the MS bound for the partial sequence M192/C194/I649. All pruned sequences and partial sequences, shown as empty gray circles, have no additional computation performed. Even though SS bounds are initiated for both I192/C194/I649/E651 and I192/C194/I649/D651, the latter is pruned after computing a mer e δ-approximation bound (orange leaf node), which is cheaper, and not as tight as a ε-approximate bound. A provable ε-approximation bound (green leaf node) is computed for only the optimal sequence, I192/C194/I649/E651. In contrast, SS methods compute separate ε-approximate bounds (which are expensive) for all eight possible sequences, shown as leaf nodes in the tree.

Furthermore, these MS bounds are used to prune *partial sequences* containing *combinations of mutations*: for a pruned partial sequence s', all t^u sequences containing s' are provably eliminated from search without any additional computation. That is, *BBK** provably, *combinatorially* prunes the search space. Since this pruning occurs during search, rather than as a preprocessing step, the pruning performed by *BBK** is distinct from previous pruning techniques such as dead-end elimination (Desmet et al., 1992; Dahiyat and Mayo, 1996). *BBK** pruning is used after dead-end elimination pruning to further improve empirical performance. Finally, MS bounds are in many cases inexpensive to compute when compared with the $\mathcal{O}(q^n n^2)$ complexity of computing an SS-ε bound for an SS. Since there are q^a partial conformation energy bounds to compute, the cost of an MS bound increases exponentially as a increases. Obviously, when $a << n$, $q^{a+2} << q^n$. This is very advantageous for A^* search, because a is initially very small: when *BBK** begins search, $a = 1$, and increases one at a time. Furthermore, $a + u = n$, and a never exceeds n. Thus in many cases $a << n$, and MS bound costs of $\mathcal{O}(q^{a+2} t^2 un)$ are significantly smaller than the SS-ε costs of $\mathcal{O}(q^n n^2)$ for an SS. Use of MS bounds enables *BBK** to efficiently bound and prune sequences that would otherwise require $\mathcal{O}(q^n n^2)$ time *each* to evaluate.

The algorithmic advances that make MS bounds possible are new bounds on partial and full sequences. We denote the design states unbound protein, unbound ligand, and bound complex as P, L, and PL, respectively. The following definitions of these new bounds are sufficient for the theorems provided in the main article—the precise definitions involve some subtleties, which are deferred to section A of the **SI** (Ojewole et al., 2017b). Given a sequence s and a state $X \in \{P, L, PL\}$, the function $L_X(s)$ is a provable lower bound of the partition function for s in state X, and $U_X(s)$ is a provable upper bound on the partition function for s in state X. For a partial sequence s', $L_X(s')$ and $U_X(s')$ are, respectively, partition function lower and upper bounds for the combinatorial number of sequences containing s'. These lower- and upper-bounding functions are combined into an upper-bounding function $K_a^+(s')$ on the partition function ratio of s'.

Definition 1. *Let s be a sequence. $K_a^+(s)$ is defined as follows:*

$$K_a^+(s) = \frac{U_{PL}(s)}{L_P(s) L_L(s)}. \tag{3}$$

The following theorem establishes the relationship between the partition function ratio of a partial sequence and the partition function ratio of any sequence containing the partial sequence:

Theorem 1. *Let s be a partial or complete sequence. For any partial sequence* $s' \subset s$, $K_a^+(s')$ *bounds* $K_a^+(s)$ *from above:*

$$K_a^+(s') \geq K_a^+(s) \geq \frac{Z_{PL}(s)}{Z_P(s)Z_L(s)} = K^*(s). \tag{4}$$

A proof of this theorem is provided in section A.3 of the **SI** (Ojewole et al., 2017b). Theorem 1 shows that the bounds used by *BBK** are *admissible*. That is, they never underestimate the *K** ratio of any partial sequence. Thus, *BBK** uses $K_a^+(s')$ as the optimistic bounding function for *A** search. Previously, *A** search has been used to provably enumerate conformations within some energy window $\boldsymbol{E_w}$ of the GMEC (Leach and Lemon, 1998) and to provably approximate the partition function of single sequences (Lilien et al., 2005; Georgiev et al., 2008; Donald, 2011; Roberts et al., 2012). Since Equation (4) defines an admissible bound over sequences, all of the provable guarantees of *A** apply to *BBK**. With these guarantees, *BBK** provably searches over *sequences* rather than conformations, and is guaranteed to return a gap-free list of sequences in order of decreasing binding affinity.

3.1. Algorithm overview

*BBK** bounds all possible sequences either with the MS bounds described in Section 3, or by computing an SS bound as described in Lilien et al. (2005), Georgiev and Donald (2007), and Roberts et al. (2012). In brief, to bound an SS, *BBK** computes a gap-free list of conformations whose statistical mechanical energies [Eq. (1)] are used to bound the *K** ratio. The algorithm reports an error bound δ such that the computed bound is guaranteed to be no more than a $(1+\delta)$ factor greater than the true *K** ratio. We refer to these SS, δ-approximate bounds (Georgiev et al., 2008; Georgiev, 2009) as *SS-δ bounds*. As the gap-free list of conformations used for an SS-δ bound grows in size, the computed SS bound becomes tighter (δ decreases). Eventually, $\delta \leq \varepsilon$, and the SS bound becomes a provable ε-approximation, which we refer to as an *SS-ε bound*. We refer to an SS-δ bound constructed this way as a *running bound*, which *BBK** incrementally tightens as it enumerates additional conformations (Georgiev et al., 2008).

*BBK** maintains a max heap whose node values correspond to either full or partial sequences and whose node keys are an upper bound on all *K** scores [Eq. (2)] in the sequence space represented by the node. The heap is initialized with a node representing the entire sequence space. *BBK** then repeatedly removes the max node *x* of the heap and performs one of the following operations:

(1) *Branch.* If *x* contains a partial sequence s', then s' is expanded. Expansion creates *t* new child nodes by selecting an unassigned residue *r* in s', and creating a new child node for each allowed amino acid *a* at *r*. Each child node contains s' plus the additional mutation of *a* assigned at *r*. These nodes are bounded with MS bounds or SS-δ bounds and reinserted into the heap.
(2) *Update.* If *x* contains a complete sequence *s*, whose bound is an SS-δ bound, then *BBK** enumerates *m* additional conformations ($m=8$ in our study), tightening the SS-δ bound. *x* is reinserted into the heap, with the updated SS-δ bound as its key. Computing this tighter SS-δ bound is important for pruning sequences, as shown by our computational experiments in Section 4.2.
(3) *Return.* If node *x* contains a full sequence, whose bound is an SS-ε bound, then the sequence in *x* has the best *K** score of all unenumerated sequences (as with *A**, all better sequences are guaranteed to have already been enumerated). The sequence of *x* is returned.

*BBK** terminates when it has enumerated the top *k* sequences (by SS-ε bound), where *k* is a user-specified number. A detailed description of the algorithm is provided in section A.8 of the **SI** (Ojewole et al., 2017b).

4. COMPUTATIONAL EXPERIMENTS

We implemented *BBK** in our laboratory's open source OSPREY (Gainza et al., 2013) protein design package and compared our algorithm with the previous state-of-the-art SS iMinDEE/A^*/K^* algorithm

(Lilien et al., 2005; Georgiev et al., 2008; Roberts et al., 2012). We computed the five best binding sequences using both *BBK** and iMinDEE/A^*/K^* for 204 different protein design problems from 51 different protein–ligand complexes. For each protein–ligand interface, we created four design problems spanning the wild-type sequence and all sets of single, double, triple, and quadruple mutants, respectively. In each design problem, we modeled either eight or nine residues at the protein–ligand interface as mutable and flexible. Each mutable residue was allowed to assume its wild-type identity or mutate to 13–19 other amino acids. The size of the resulting design problems ranged from 10 to 2.6×10^6 sequences and 10^5 to 10^{11} conformations (over all sequences). In all cases, we modeled continuous side-chain flexibility using continuous rotamers (Gainza et al., 2012; Roberts and Donald, 2015). As in Georgiev et al. (2008) and Gainza et al. (2012), rotamers from the Penultimate Rotamer Library (Lovell et al., 2000) were allowed to minimize to any conformation within 9° of their modal χ-angles. For all design problems, we performed minimized dead-end elimination pruning (minDEE) (Georgiev et al., 2008), followed by either iMinDEE/A^*/K^* or *BBK**. The initial pruning window (Gainza et al., 2012) was 0.1 kcal/mol, and the SS-ε bound accuracy was 0.683 [details are provided in section A.9 of the **SI** (Ojewole et al., 2017b)]. Each design either provably returned the optimal sequences or was terminated after 30 days. A detailed description of the 51 protein–ligand systems in our experiments, the 204 protein design problems based on these systems, and our experimental protocol is provided in section C.1 of the **SI** (Ojewole et al., 2017b).

4.1. Performance comparison

As the size of the sequence space increases, so does the efficiency of *BBK** over iMinDEE/A^*/K^* (Fig. 3), demonstrating that the complexity of *BBK** is in practice sublinear in the number of sequences. Because iMinDEE/A^*/K^* computes individual sequence binding energies as SS-ε bounds, we first measured the efficiency of *BBK** using the number of SS-ε bounds computed. Since K^* scores in iMinDEE/A^*/K^* are based on minimized conformation energies (Lilien et al., 2005; Georgiev et al., 2008; Gainza et al., 2012), whose computation represents a major computational bottleneck, we also measured efficiency using the number of conformation energy minimizations performed. Last, we compared the running times of *BBK** with those of iMinDEE/A^*/K^*.

We divide the design problem sizes into three categories: the smallest problems have between 10 and 10^2 sequences, medium-sized problems contain between 10^2 and 10^4 sequences, and the largest problems contain between 10^4 and 10^7 sequences. After 30 days, *BBK** completed all 204 designs, but iMinDEE/A^*/K^* completed only 107 of 204 designs: all 39 of the smallest designs, 54 of 63 medium-sized designs, and only 14 of the 111 largest designs. We now discuss results for the 107 designs completed by iMinDEE/A^*/K^*. Because iMinDEE/A^*/K^* computes individual sequence binding energies as SS-ε bounds, we first measured the efficiency of *BBK** using the number of SS-ε bounds computed (Fig. 3A). For small, medium, and large designs, respectively, *BBK** was on average 17-fold, 162-fold, and 2568-fold more efficient than iMinDEE/A^*/K^*. Next, we measured efficiency using the number of conformation energy minimizations performed (Fig. 3B). Here, *BBK** minimized, on average, 10-, 43-, and 113-fold fewer conformations for small, medium, and large-sized designs, respectively, compared with iMinDEE/A^*/K^*. Last, we compared empirical running times for both methods (Fig. 3C). On average, *BBK** was 36-, 67-, and 97-fold faster than iMinDEE/A^*/K^* for small, medium, and large-sized designs, respectively.

Based on the data from the 107 designs that iMinDEE/A^*/K^* was able to complete within 30 days, we conclude that *BBK** provides a combinatorial speedup over iMinDEE/A^*/K^*. Crucially, *BBK** is not only more efficient but also retains the provability guarantees and biophysical modeling improvements (i.e., ensemble-based design and continuous flexibility) employed by SS iMinDEE/A^*/K^*. In one large design (2.6×10^6 sequences), involving a camelid single-domain V_HH antibody fragment in complex with RNase A (PDB id: 2P49), *BBK** pruned >99.9% of sequences to provably find the best five sequences. Therefore, *BBK** can design over similarly sized sequence spaces to high-throughput experimental screening methods such as phage display (Carmen and Jermutus, 2002; Pál et al., 2006).

4.2. Sequence space pruning

*BBK** owes its efficiency to two complementary modes of sequence pruning: MS bound pruning and SS-δ bound pruning. Figure 4A shows the efficiency gains in *BBK** due to MS bound pruning. In small, medium, and large design problems, respectively, *BBK** pruned up to 90%, 99%, and 99.9% of the

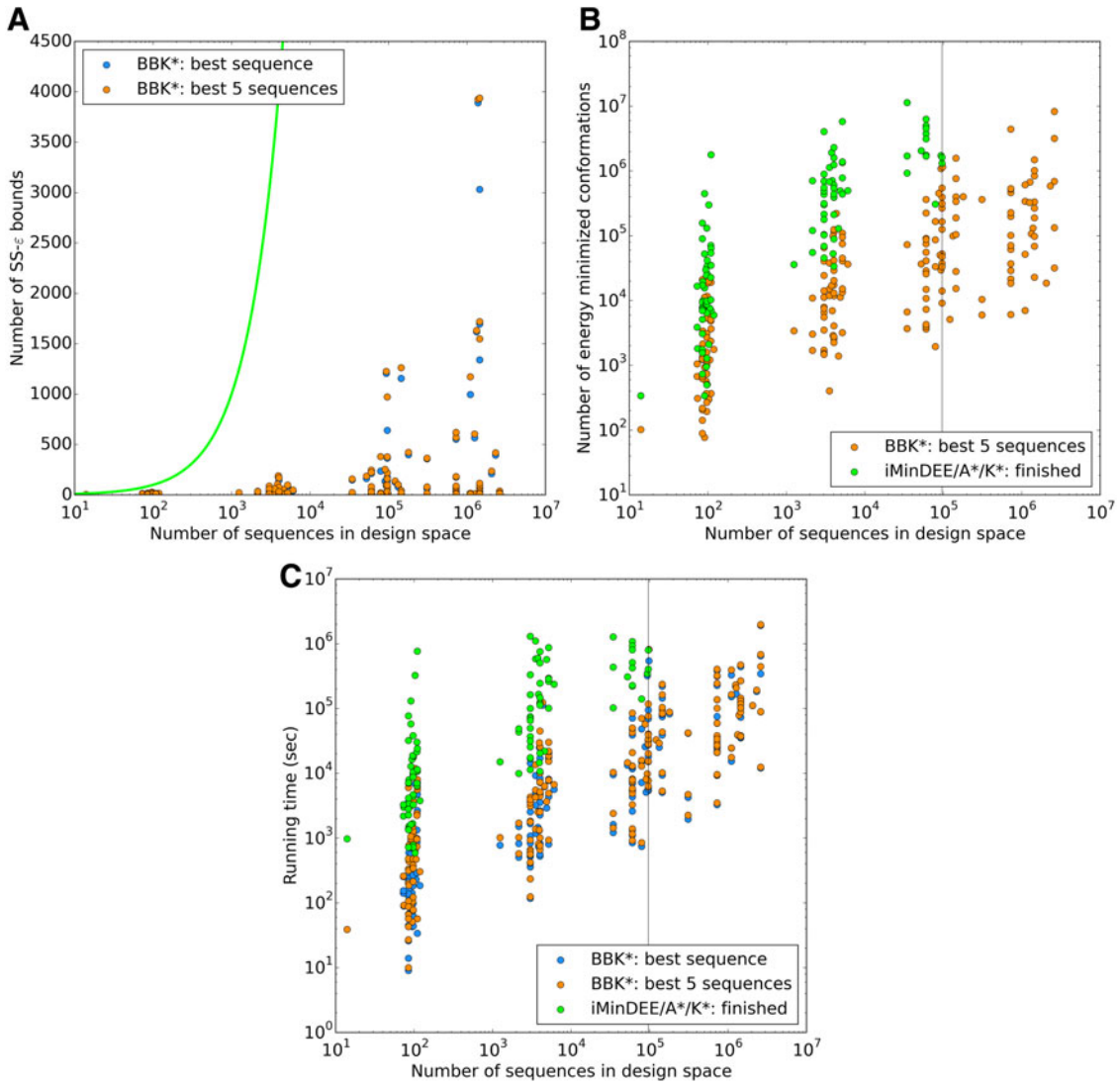

FIG. 3. *BBK** is up to five orders of magnitude more efficient than iMinDEE/A^*/K^***.** *BBK** completed all 204 designs within a 30-day limit, whereas iMinDEE/A^*/K^* completed only 107. **(A)** The number of SS-ε bounds performed versus the number of sequences in the design space. Results are shown for computing only the best sequence (blue) and computing the best five sequences (orange). SS algorithms, including the best previous algorithm iMinDEE/A^*/K^*, must compute binding affinity for all possible sequences (green curve). *BBK** required up to 6×10^5-fold fewer SS-ε bounds to find the best sequences. **(B)** The number of energy-minimized conformations by *BBK** and iMinDEE/A^*/K^* versus the number of sequences in the design space. iMinDEE/A^*/K^* completed only 107 of 204 designs (left of the vertical line) before the 30-day timeout. For these designs, *BBK** was up to 1700-fold more efficient. **(C)** *BBK** and iMinDEE/A^*/K^* running times versus the number of sequences in the design space. For the 107 designs completed by iMinDEE/A^*/K^* within 30 days (left of the vertical line), *BBK** was up to 800-fold more efficient than iMinDEE/A^*/K^*.

sequence design space using MS bound pruning. These data show that the amount of MS pruning increased significantly with the size of the design space. Figure 4B illustrates the efficiency gains in *BBK** due to SS-δ bound pruning. In small, medium, and large design problems, respectively, SS-δ pruning eliminates up to 98%, 99.9%, and 99.99% of the sequences not pruned by MS pruning. These data show that the amount of SS-δ pruning increased with the size of the design problem. Further details are provided in sections A.10 and B.1 of the **SI** (Ojewole et al., 2017b).

Importantly, MS bound pruning and SS-δ bound pruning have multiplicative synergy, producing a combined pruning effect of up to 99.99999% of the original sequence space while provably finding the five

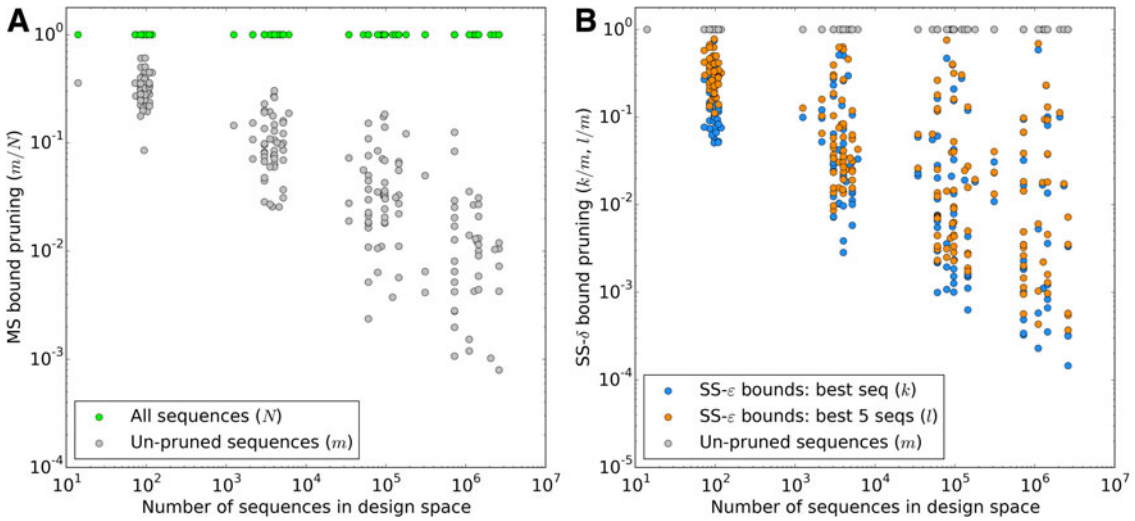

FIG. 4. *BBK** used MS and SS-δ bounds to prune up to 99.99999% of the sequence space. **(A)** Sequence space reduction due to MS pruning. The fraction of unpruned sequences (gray) normalized to the total number of sequences (green). *BBK** used MS bounds to provably prune up to 99.9% of the sequence space. *BBK** does not compute SS-ε bounds for pruned sequences. **(B)** The fraction of *BBK** SS-ε bounds (blue for the best sequence and orange for the best five sequences) normalized to the number of sequences not pruned by MS bound pruning (gray). To compute the best sequences, *BBK** calculated SS-ε bounds for as few as 0.01% of the unpruned sequences. The remaining 99.99% of these sequences were pruned at mere SS-δ accuracy.

best binding sequences. In one example, we redesigned the protein–protein interface (PPI) of a camelid affinity-matured single-domain V_HH antibody fragment (PDB id: 2P4A). The sequence space, 2.6×10^6 sequences, consisted of all quadruple mutants in the 9-residue PPI. *BBK** pruned all but 2078 sequences using MS pruning and then pruned 2071 sequences from these remaining 2078 sequences using SS-δ bound pruning. These data show how *BBK** prunes a combinatorial number of sequences from the design space, producing dramatic efficiency gains over SS methods. See **SI** section 5.2 for details (Ojewole et al., 2017b).

4.3. Design with coupled continuous side-chain and backbone flexibility

To determine whether design with a *fixed* backbone and continuous rotamers predicts tight binding in the same sequences as does a model with both *local backbone flexibility* and continuous rotamers, we used *BBK** to redesign the human fibronectin F1:*Staphylococcus aureus* FNBPA-5 interface (Lower et al., 2011) (PDB id: 2RL0) for binding affinity. As we discuss below, the flexible backbone model favors binding in different sequences than the fixed backbone model does. Details of our experimental protocol are provided in section C.3 of the **SI** (Ojewole et al., 2017b).

In the first experiment, we redesigned the fibronectin F1:FNBPA-5 interface for binding affinity over the wild-type sequence and 15 single amino acid polymorphisms. Our results showed that using the flexible backbone model versus the fixed backbone model increased the size of the design conformation space by 1417-fold but only increased the running time by 4-fold in *BBK**. By comparison, iMinDEE/A^*/K^* required 48-fold more time than *BBK** to complete the flexible backbone design. Our results also showed that the *BBK** sequence rankings between the two input models had a Spearman correlation coefficient of only $\rho=0.53$. Thus, the flexible backbone model favors binding in different sequences than the fixed backbone model does. For instance, the FNBPA-5 D650E mutant is predicted to bind less tightly than the wild-type in the fixed backbone model (Fig. 5A) but more tightly than WT in the flexible model (Fig. 5B). In our second experiment, the sequence design space consisted of the wild-type sequence and 25 single amino acid polymorphisms. The *BBK** sequence rankings produced by the two input models had a Spearman correlation coefficient of $\rho=0.82$ [additional details are provided in section B.2 of the **SI** (Ojewole et al., 2017b)]. Relative to the fixed backbone model, the flexible backbone model increased the size of the design conformation space by 8447-fold but only increased the running time by only 1.7-fold in *BBK**. iMinDEE/A^*/K^* required 89-fold more time than *BBK** to complete the design using the flexible backbone model.

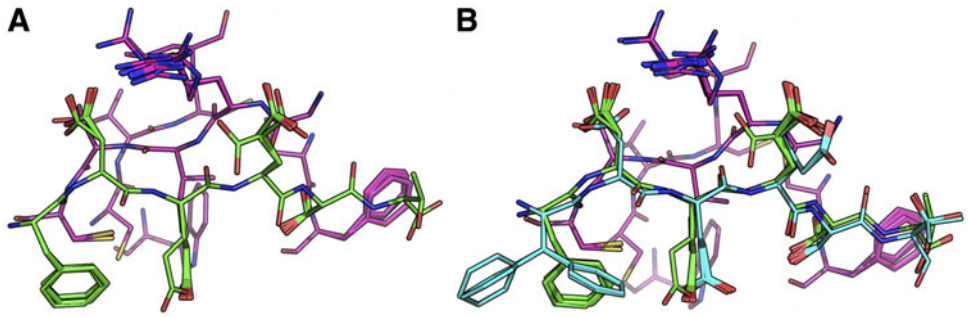

FIG. 5. *BBK** efficiently handles coupled continuous side-chain and local backbone flexibility. Selected residues from ensembles, computed by *BBK**, of human fibronectin F1 modules 4–5 (magenta) in complex with a fragment of *Staphylococcus aureus* fibronectin binding protein A 5 [FNBPA-5, PDB id: 2RL0, (Lower et al., 2011)]. The design space consisted of the wild-type sequence and either 15 or 25 single amino acid mutants. **(A)** Ensemble of the wild-type sequence based on the original crystal structure. The design used a fixed FNBPA-5 backbone (green) and continuous side-chain flexibility. **(B)** Ensemble of the wild-type sequence using two backbones: the original FNBPA-5 backbone (green) and a second backbone (PDB id: 2RKY, cyan) with RMSD 1.3 Å from the original [found using the MASTER program (Zhou and Grigoryan, 2015)]. The sequence rankings [by K^* score, Eq. (2)] from the fixed and flexible backbone models had Spearman correlation coefficients of $\rho = 0.53$ and $\rho = 0.82$ in the 15 and 25 mutant designs, respectively. This shows that the flexible backbone model favors binding in very different sequences than the fixed backbone model does.

It is important to note that these experiments are only possible with provable algorithms. Without the provable guarantees of *BBK**, it would be difficult and perhaps unsound to compare the results of CPD with and without coupled continuous side-chain and backbone flexibility, since differences induced by the fixed backbone and rotamer model cannot be deconvolved from differences stemming from undersampling or inadequate stochastic optimization. Thus, *BBK** provides provable methods to analyze the difference in predicted sequences between different models of side-chain and backbone flexibility.

5. CONCLUSION

*BBK** fills an important *lacuna* in protein design: we presented a novel algorithm that can search efficiently not over the energies of single conformations, but over the binding affinity of sequences. *BBK** is, to our knowledge, the first provable, ensemble-based algorithm to search in order of binding affinity and run in time *sublinear* in the number of sequences. Previously, protein designers either employed heuristic algorithms to compute locally optimal sequences, or computed provably accurate approximations of binding affinity for each sequence individually. *BBK** not only computes the globally optimal sequences, but it also does so while combinatorially pruning the search space. Our experiments show that *BBK** can search over sequence spaces of up to 2.6×10^6 sequences, a capacity comparable with high-throughput experimental screening methods such as phage display. Thus, *BBK** liberates binding affinity-based protein design from the efficiency barrier imposed by exhaustive search. Ensemble-based design for affinity over large sequence spaces was previously possible only with heuristic algorithms (with no guarantees), or using high-throughput wet-bench experiments. *BBK** enables CPD by providing new K_a algorithms, with provable guarantees, for these large-scale protein designs.

ACKNOWLEDGMENTS

We would like to thank Drs. Mark Hallen and Pablo Gainza for helpful discussions and for providing useful protein–ligand binding problems; Dr. Jeffrey Martin for assisting with software optimizations; Jack Holland, Dr. Gevorg Grigoryan, Hunter Nisonoff, Anna Lowegard, and all members of the Donald laboratory for helpful discussions; and the NSF (GRFP DGF 1106401 to A.A.O.) and NIH (R01-GM78031 and R01-GM118543 to BRD, R01-HL119648 to V.G.F.) for funding.

AUTHOR DISCLOSURE STATEMENT

No competing financial interests exist.

REFERENCES

Boas, F.E., and Harbury, P.B. 2007. Potential energy functions for protein design. *Curr. Opin. Struct. Biol.* 17, 199–204.

Carmen, S., and Jermutus, L. 2002. Concepts in antibody phage display. *Brief Funct. Genomic Proteomic.* 1, 189–203.

Chen, C.-Y., Georgiev, I., Anderson, A.C., et al. 2009. Computational structure-based redesign of enzyme activity. *Proc. Natl Acad. Sci. U. S. A.* 106, 3764–3769.

Dahiyat, B.I., and Mayo, S.L. 1996. Protein design automation. *Protein Sci.* 5, 895–903.

Desmet, J., De Maeyer, M., Hazes, B., et al. 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*. MIT Press, Cambridge, MA.

Fleishman, S.J., Khare, S.D., Koga, N., et al. 2011. Restricted sidechain plasticity in the structures of native proteins and complexes. *Protein Sci.* 20, 753–757.

Frey, K.M., Georgiev, I., Donald, B.R., et al. 2010. Predicting resistance mutations using protein design algorithms. *Proc. Natl Acad. Sci. U. S. A.* 107, 13707–13712.

Fromer, M., and Yanover, C. 2008. A computational framework to empower probabilistic protein design. *Bioinformatics* 24, i214–i222.

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

Gainza, P., Roberts, K.E., and Donald, B.R. 2012. Protein design using continuous rotamers. *PLoS Comput. Biol.* 8, e1002335.

Gainza, P., Roberts, K.E., Georgiev, I., et al. 2013. Program, user manual, and source code are available at www.cs.duke.edu/donaldlab/software.php. OSPREY: Protein design with ensembles, flexibility, and provable algorithms. *Methods Enzymol.* 523, 87–107.

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

Georgiev, I., Lilien, R.H., and 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., and 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. *J. Comput. Chem.* 29, 1527–1542.

Georgiev, I., Schmidt, S., Li, Y., et al. 2012. Design of epitope-specific probes for sera analysis and antibody isolation. *Retrovirology* 9, P50.

Georgiev, I.S. 2009. Novel algorithms for computational protein design, with applications to enzyme redesign and small-molecule inhibitor design [Ph.D. thesis]. Duke University, Durham, NC. Retrieved from http://hdl.handle.net/10161/1113.

Georgiev, I.S., Rudicell, R.S., Saunders, K.O., et al. 2014. Antibodies VRC01 and 10E8 neutralize HIV-1 with high breadth and potency even with IG-framework regions substantially reverted to germline. *J. Immunol.* 192, 1100–1106.

Gilson, M.K., Given, J.A., Bush, B.L., et al. 1997. The statistical-thermodynamic basis for computation of binding affinities: A critical review. *Biophys. J.* 72, 1047–1069.

Gorczynski, M.J., Grembecka, J., Zhou, Y., et al. 2007. Allosteric inhibition of the protein-protein interaction between the leukemia-associated proteins Runx1 and CBFbeta. *Chem. Biol.* 14, 1186–1197.

Hallen, M.A., and Donald, B.R. 2016. Comets (Constrained optimization of multistate energies by tree search): A provable and efficient protein design algorithm to optimize binding affinity and specificity with respect to sequence. *J. Comput. Biol.* 23, 311–321.

Hallen, M.A., Gainza, P., and Donald, B.R. 2015. Compact representation of continuous energy surfaces for more efficient protein design. *J. Chem. Theory Comput.* 11, 2292–2306.

Hallen, M.A., Jou, J.D., and Donald, B.R. 2017. LUTE (Local unpruned tuple expansion): Accurate continuously flexible protein design with general energy functions and rigid Rotamer-Like efficiency. *J. Comput. Biol.* 24(6), 536–546.

Hallen, M.A., Keedy, D.A., and Donald, B.R. 2013. Dead-end elimination with perturbations (DEEPer): A provable protein design algorithm with continuous sidechain and backbone flexibility. *Proteins* 81, 18–39.

Hart, P., N.J., N., and Raphael, B. 1968. A formal basis for the heuristic determination of minimum cost paths. *IEEE Trans. SSC.* 4, 100–114.

Jou, J.D., Jain, S., Georgiev, I.S., et al. 2016. BWM*: A novel, provable, ensemble-based dynamic programming algorithm for sparse approximations of computational protein design. *J. Comput. Biol.* 23, 413–424.

Kingsford, C.L., Chazelle, B., and Singh, M. 2005. Solving and analyzing side-chain positioning problems using linear and integer programming. *Bioinformatics* 21, 1028–1039.

Kuhlman, B., and Baker, D. 2000. Native protein sequences are close to optimal for their structures. *Proc. Natl Acad. Sci. U. S. A.* 97, 10383–10388.

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

Leaver-Fay, A., Tyka, M., Lewis, S.M., et al. 2011. Rosetta3: An object-oriented software suite for the simulation and design of macromolecules. *Methods Enzymol.* 487, 545–574.

Lee, C., and Levitt, M. 1991. Accurate prediction of the stability and activity effects of site-directed mutagenesis on a protein core. *Nature* 352, 448–451.

Leech, J., Prins, J.F., and Hermans, J. 1996. Smd: Visual steering of molecular dynamics for protein design. *Comput. Sci. Eng.* 3, 38–45.

Lilien, R.H., Stevens, B.W., Anderson, A.C., et al. 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. *J. Comput. Biol.* 12, 740–761.

Lovell, S.C., Word, J.M., Richardson, J.S., et al. 2000. The penultimate rotamer library. *Proteins* 40, 389–408.

Lower, S.K., Lamlertthon, S., Casillas-Ituarte, N.N., et al. 2011. Polymorphisms in fibronectin binding protein A of *Staphylococcus aureus* are associated with infection of cardiovascular devices. *Proc. Natl. Acad. Sci. U. S. A.* 108, 18372–18377.

Nisonoff, H. 2015. Efficient partition function estimation in computational protein design: Probabalistic guarantees and characterization of a novel algorithm [B.S. Thesis]. Department of Mathematics, Duke University.

Ojewole, A., Lowegard, A., Gainza, P., et al. 2017a. OSPREY predicts resistance mutations using positive and negative computational protein design. *Methods Mol. Biol.* 1529, 291–306.

Ojewole, A.A., Jou, J.D., Fowler, V.G., et al. 2017b. Supplementary information: BBK* (Branch and Bound over K*): A provable and efficient ensemble-based protein design algorithm to optimize stability and binding affinity over large sequence spaces for sparse approximations of computational protein design. Available at: www.cs.duke.edu/donaldlab/Supplementary/jcb17/bbkstar. Last accessed on February 23, 2018.

Pál, G., Kouadio, J.-L.K., Artis, D.R., et al. 2006. Comprehensive and quantitative mapping of energy landscapes for protein-protein interactions by rapid combinatorial scanning. *J. Biol. Chem.* 281, 22378–22385.

Peng, J., Hosur, R., Berger, B., et al. 2015. iTreePack: Protein complex Side-Chain packing by dual decomposition. *arXiv:1504.05467 [q-bio.BM].*

Pierce, N.A., and Winfree, E. 2002. Protein design is NP-hard. *Protein Eng.* 15, 779–782.

Reeve, S.M., Gainza, P., Frey, K.M., et al. 2015. Protein design algorithms predict viable resistance to an experimental antifolate. *Proc. Natl Acad. Sci. U. S. A.* 112, 749–754.

Roberts, K.E., Cushing, P.R., Boisguerin, P., et al. 2012. Computational design of a PDZ domain peptide inhibitor that rescues CFTR activity. *PLoS Comput. Biol.* 8, e1002477.

Roberts, K.E., and Donald, B. R. 2015. Improved energy bound accuracy enhances the efficiency of continuous protein design. *Proteins* 83, 1151–1164.

Roberts, K.E., Gainza, P., Hallen, M.A., et al. 2015. Fast gap-free enumeration of conformations and sequences for protein design. *Proteins* 83, 1859–1877.

Rudicell, R.S., Kwon, Y.D., Ko, S.-Y., et al. 2014. Enhanced potency of a broadly neutralizing HIV-1 antibody in vitro improves protection against lentiviral infection in vivo. *J Virol* 88, 12669–12682.

Sciretti, D., Bruscolini, P., Pelizzola, A., et al. 2009. Computational protein design with side-chain conformational entropy. *Proteins* 74, 176–191.

Silver, N.W., King, B.M., Nalam, M.N.L., et al. 2013. Efficient computation of small-molecule configurational binding entropy and free energy changes by ensemble enumeration. *J. Chem. Theory Comput.* 9, 5098–5115.

Simoncini, D., Allouche, D., de Givry, S., et al. 2015. Guaranteed discrete energy optimization on large protein design problems. *J. Chem. Theory Comput.* 11, 5980–5989.

Stevens, B.W., Lilien, R.H., Georgiev, I., et al. 2006. Redesigning the PheA domain of gramicidin synthetase leads to a new understanding of the enzyme's mechanism and selectivity. *Biochemistry* 45, 15495–15504.

Traoré, S., Allouche, D., André, I., et al. 2013. A new framework for computational protein design through cost function network optimization. *Bioinformatics* 29, 2129–2136.

Traoré, S., Roberts, K.E., Allouche, D., et al. 2016. Fast search algorithms for computational protein design. *J. Comput. Chem.* 37, 1048–1058.

Valiant, L.G. 1979. The complexity of computing the permanent. *Theor. Comput. Sci.* 8, 189–201.

Viricel, C., Simoncini, D., Barbe S., Schiex, T. 2016. Guaranteed Weighted Counting for Affinity Computation: Beyond Determinism and Structure. In: Rueher M. (eds) Principles and Practice of Constraint Programming. CP 2016. Lecture Notes in Computer Science, vol 9892. Springer, Cham. DOI https://doi.org/10.1007/978-3- 319-44953- 1_46.

Wainwright, M.J., Jaakkola, T.S., and Willsky, A.S. 2013. A new class of upper bounds on the log partition function. *CoRR* abs/1301.0610.

Xu, J. 2005. Rapid protein side-chain packing via tree decomposition. *9th Annual International Conference, RECOMB* 3500, 423–439.

Xu, J., and Berger, B. 2006. Fast and accurate algorithms for protein side-chain packing. *J. ACM* 53, 533–557.

Zheng, F., Yang, W., Ko, M.-C., et al. 2008. Most efficient cocaine hydrolase designed by virtual screening of transition states. *J. Am. Chem. Soc.* 130, 12148–12155.

Zhou, J., and Grigoryan, G. 2015. Rapid search for tertiary fragments reveals protein sequence-structure relationships. *Protein Sci.* 24, 508–524.

Address correspondence to:
Bruce R. Donald
Professor of Computer Science, Duke University
Professor of Biochemistry, Duke University Medical Center
P.O. Box 90129
Department of Computer Science
D101 Levine Science Research Center (LSRC)
Research Drive
Duke University
Durham, NC 27708-0129 USA

E-mail: brd+jcb17@cs.duke.edu