LUTE (Local Unpruned Tuple Expansion): Accurate continuously flexible protein design with general energy functions and rigid-rotamer-like efficiency

Mark A. Hallen[1,†], Jonathan D. Jou[1,††], and Bruce R. Donald[1,2,3,*]

August 12, 2016

Running head: *LUTE: unpruned tuple expansion for protein design*

[1] Department of Computer Science, Levine Science Research Center, 308 Research Dr., Duke University, Durham, NC 27708 USA

[2] Department of Chemistry, Duke University, Durham, NC 27708 USA

[3] Department of Biochemistry, Duke University Medical Center, Durham, NC 27710 USA

*Corresponding author. Phone: 919-660-6583. Fax: 919-660-6519. Email: `brd+jcb16@cs.duke.edu`

† Phone: 919-660-4017. Fax: 919-660-6519. Email: `mark.hallen@duke.edu`

†† Phone: 919-660-4017. Fax: 919-660-6519. Email: `jj@cs.duke.edu`

Abstract

Most protein design algorithms search over discrete conformations and an energy function that is residue-pairwise, i.e., a sum of terms that depend on the sequence and conformation of at most two residues. Although modeling of *continuous flexibility* and of *non-residue-pairwise energies* significantly increases the accuracy of protein design, previous methods to model these phenomena add a significant asymptotic cost to design calculations. We now remove this cost by modeling continuous flexibility and non-residue-pairwise energies in a form suitable for direct input to highly efficient, discrete combinatorial optimization algorithms like DEE/A* or Branch-Width Minimization. Our novel algorithm performs a local unpruned tuple expansion (LUTE), which can efficiently represent both continuous flexibility and general, possibly non-pairwise energy functions to an arbitrary level of accuracy using a discrete energy matrix. We show using 47 design calculation test cases that LUTE provides a dramatic speedup in both single-state and multistate continuously flexible designs.

1 Introduction

Protein design algorithms compute protein sequences that will perform a desired function [Donald, 2011]. They generally do this by minimizing the energy of a desired binding or structural state (or some combination thereof [Hallen and Donald, 2015, Leaver-Fay *et al.*, 2011]) with respect to sequence [Donald, 2011, Kuhlman and Baker, 2000, Desmet *et al.*, 1992, Gainza *et al.*, 2012, Georgiev *et al.*, 2014, Karanicolas and Kuhlman, 2009, Georgiev *et al.*, 2008, Floudas *et al.*, 1999]. Given a model of the conformational space of a protein and its energy function (which maps conformations to their energies), this is a well-defined computational problem [Donald, 2011].

Previously, this minimization problem has been most efficient to solve if two restrictions are imposed on the model. First, the conformational space of the protein is modeled as discrete. Specifically, each residue takes on conformations from a discrete set (typically, experimentally observed sidechain conformations known as *rotamers* [Janin *et al.*, 1978]). Hence, we optimize with respect to the amino-acid type and rotamer of each residue. Second, the energy function is assumed to be residue-pairwise, i.e., it is assumed to be a sum of terms that each depend on the amino-acid types and conformations of at most two residues.

A large body of efficient algorithms has been developed for this restricted case of the protein design problem, many of which offer provable accuracy. In particular, the dead-end elimination (DEE) algorithm [Desmet *et al.*, 1992] removes rotamers that provably cannot be part of the global minimum-energy conformation (GMEC). The A* algorithm from artificial intelligence [Hart *et al.*, 1968] finds the optimal conformation using these unpruned rotamers [Leach and Lemon, 1998]. This DEE/A* framework has been generalized to model free energies for each sequence instead of simply GMECs [Lilien *et al.*, 2005, Georgiev *et al.*, 2008] (the K^* algorithm). It has also been generalized to optimize combinations of stability and specificity by minimizing, with respect to sequence, a linear combination of the conformationally optimized energies of several bound and unbound states of a protein, instead of just the energy of a single state [Hallen and Donald, 2015] (the COMETS algorithm). Several methods in addition to DEE/A* have also been used to address the protein design problem. Some of these, such as Metropolis Monte Carlo and simulated annealing [Lee and Levitt, 1991, Kuhlman and Baker, 2000], lack provable guarantees of accuracy, and thus may miss the optimal conformation significantly. Other algorithms with provable accuracy are also available, largely building on techniques from integer linear programming [Kingsford *et al.*, 2005, Roberts *et al.*, 2015] and weighted constraint satisfaction [Traoré *et al.*, 2016, Traoré *et al.*, 2013, Roberts *et al.*, 2015]. Notably, treewidth- and branch-width-based algorithms, such as TreePack [Xu and Berger, 2006] and BWM* [Jou *et al.*, 2015] solve this problem with provable accuracy in polynomial time for systems whose residue interaction graph has treewidth or branch-width bounded by a constant [Jou *et al.*, 2015].

However, proteins are actually continuously flexible, and continuous flexibility both in the sidechains [Gainza *et al.*, 2012]

and backbone [Hallen *et al.*, 2013] has been shown to result in significantly lower energies and biologically better sequences [Gainza *et al.*, 2012, Hallen *et al.*, 2013]. Although a residue sidechain will usually be found in the *vicinity* of the modal conformation for a rotamer, its dihedral angles will often differ from this mode by 10° or more [Janin *et al.*, 1978]. These continuous adjustments are often critical for determining what conformations are sterically feasible [Gainza *et al.*, 2012]. Thus, incorporation of continuous flexibility modeling substantially increases the accuracy of designs. The minDEE and iMinDEE methods [Georgiev *et al.*, 2008, Gainza *et al.*, 2012] do this for continuous sidechain flexibility, and DEEPer [Hallen *et al.*, 2013] for simultaneous continuous sidechain and backbone flexibility. These methods replace the traditional discrete rotamers used in DEE/A* with voxels in the conformation space of each residue, called *residue conformations (RCs)*. An RC is defined as an amino acid type together with bounds on each of the conformational degrees of freedom of the residue (e.g., sidechain dihedrals) [Hallen *et al.*, 2013]. The modal conformation for a rotamer is usually found at the center of this voxel. In this model, the conformation space of a entire protein is a union of voxels, each of which is constructed as the cross-product of single-residue voxels. Thus, each voxel in the conformation space of the entire protein is represented by a list of RCs, one for each residue being modeled as flexible. RCs are constructed to be small enough that we can use local minimization to find the optimal energy within the voxel. This applies to both the single-residue and entire-protein voxels.

However, the global minimum energy in this model could not previously be computed directly by DEE/A*. Instead, DEE/A* was used to enumerate *RC lists* (protein conformational voxels) in order of a lower bound on minimized energy [Georgiev *et al.*, 2008, Gainza *et al.*, 2012, Hallen *et al.*, 2013]. Subsequently, the optimal energy for each RC list with a sufficiently low-energy lower bound was computed by minimization. The lower bound was computed from minimized pairwise interaction energies [Georgiev *et al.*, 2008]. This minimization was accelerated significantly by precomputing polynomials to approximate the energy landscape, using the EPIC algorithm [Hallen *et al.*, 2015]. However, minimization was still the bottleneck in continuously flexible designs and prevented them from approaching the efficiency of designs with discrete flexibility. **In essence, these previous methods modeled continuously flexibility by modifying DEE/A* and making it do much more work. In contrast, LUTE achieves much greater efficiency by representing continuous flexibility in a form suitable for direct input into DEE/A***.

We must also address the question of the energy function. The energy landscape of a real protein is not residue-pairwise, or otherwise exactly described solely as the sum of local terms. There is, however, ample evidence that protein interactions are local in a more general sense [Flocke and Bartlett, 2004, Vizcarra *et al.*, 2008, Hallen *et al.*, 2015, Zhang and Zhang, 2003]—i.e., that the cross-derivative of the energy with respect to conformational degrees of freedom of two residues will tend to zero fairly quickly as the distance between the residues increases. These properties are also observed for more realistic energy func-

tions that return an energy for the entire protein, rather than breaking the energy into terms as molecular mechanics does. For example, the Poisson-Boltzmann model for implicit solvation [Sitkoff *et al.*, 1994] and quantum-chemical models return an energy for the entire system on which they are run. Thus, a viable approach to modeling protein energies more realistically is to infer local terms from full-protein energies. Vizcarra et al. [Vizcarra *et al.*, 2008] apply this approach to the Poisson-Boltzmann model, calculating pairwise energies from differences in full-protein conformational states and achieving a pairwise energy matrix that quite accurately matches the Poisson-Boltzmann energies of full conformations. However, their method can only accommodate rotamer pairs, does not support continuous flexibility, and can only be used when substituting a single rotamer into a conformation is possible while maintaining the conformation of the other residues. This is impossible when residues share conformational degrees of freedom, which is typically needed for backbone flexibility [Georgiev and Donald, 2007, Hallen *et al.*, 2013], and may also cause problems in the case of steric clashes. Also, DEE/A* has been generalized to accommodate higher-than-pairwise energy terms if these terms are computed explicitly for particular tuples, e.g., triples of residues [LuCore *et al.*, 2015]. However, most energy functions modeling higher-than-pairwise effects, including Poisson-Boltzmann, return a single energy for the entire system, rather than a sum of explicit local terms as required by algorithms such as those in [LuCore *et al.*, 2015].

Hence, today's protein and drug designers are faced with a choice. They can neglect continuous flexibility and energy terms that aren't explicitly local (e.g., explicitly pairwise), thus incurring significant error. Or they can pay a massive overhead to incorporate them—by enumerating many conformations (for continuous flexibility) or searching exhaustively (for non-pairwise energy functions). We now offer a way around this dilemma. We construct an energy function that is an explicit, discrete sum of local energy terms, which are associated with tuples of RCs. This function maps RC lists, which represent voxels in the conformation space of a protein, to energies. But it will approximate, to arbitrary accuracy, the minimized voxel energy, which can be computed with *any* energy function: no need for residue-pairwiseness or any other local representation. Computing this approximation is a machine learning problem, and we attack it with a least-squares method. Our approach has some resemblance to cluster expansion methods, which have previously been used in quantum mechanics [Čížek, 2009] and to represent optimized energies for protein sequences [Grigoryan *et al.*, 2006, Grigoryan *et al.*, 2009]. However, as discussed in [Hallen *et al.*, 2015], approximations of energy surfaces can be much more compact if unrealistically high-energy regions of conformational space are excluded from the approximation (and from the subsequent conformational search). Thus, unlike cluster expansion methods, we exclude *pruned* tuples of RCs, making our derived energy function a *local unpruned tuple expansion*, or LUTE. Because conformational and sequence search using the LUTE energy function is a discrete optimization problem of the type solved by DEE/A*, BWM*, and other

very efficient algorithms, it allows designs to run quickly using these algorithms, while still approximating continuous flexibility and highly realistic energy functions to a high level of accuracy.

We have implemented LUTE in the OSPREY [Gainza *et al.*, 2013, Georgiev *et al.*, 2008, Georgiev *et al.*, 2009] open-source protein design package, which has yielded many designs that performed well experimentally—*in vitro* [Rudicell *et al.*, 2014, Roberts *et al.*, 2012, Gorczynski *et al.*, 2007, Chen *et al.*, 2009, Frey *et al.*, 2010, Stevens *et al.*, 2006, Georgiev *et al.*, 2012] and *in vivo* [Rudicell *et al.*, 2014, Roberts *et al.*, 2012, Gorczynski *et al.*, 2007, Frey *et al.*, 2010] as well as in non-human primates [Rudicell *et al.*, 2014]. OSPREY contains a wide array of flexibility modeling options and provably accurate design algorithms [Gainza *et al.*, 2013, Georgiev *et al.*, 2009], allowing LUTE to be used for many types of designs.

By presenting LUTE, this paper makes the following contributions:

1. A method to represent continuous flexibility and general energy functions to arbitrary accuracy in a local unpruned tuple expansion (LUTE) that can be used directly as input to discrete combinatorial search algorithms like DEE/A*.

2. A free implementation of LUTE in our laboratory's open-source OSPREY protein-design software package [Georgiev *et al.*, 2009, Frey *et al.*, 2010, Chen *et al.*, 2009, Georgiev *et al.*, 2008], available for download [Georgiev *et al.*, 2009] upon publication as free software [Georgiev *et al.*, 2009], supporting representation of both continuous sidechain and backbone flexibility and of molecular-mechanics and Poisson-Boltzmann energy functions.

3. Integration of LUTE with the DEE/A* [Leach and Lemon, 1998], iMinDEE [Gainza *et al.*, 2012], BWM* [Jou *et al.*, 2015], and COMETS [Hallen and Donald, 2015] algorithms for sequence and conformational search.

4. Bounds on the time and space complexity of protein design calculations that model continuous flexibility and/or use energy functions with non-local terms. The time and space complexity are exponential merely in the branch-width w of the residue interaction graph, and thus the designs can be done in polynomial time for systems whose branch-width is bounded by a constant.

5. Experimental results for 47 computational design calculations on 36 protein structures using LUTE, which demonstrate its accuracy and efficiency in single-state designs, multistate designs and for both n-body Poisson-Boltzmann and pairwise energy functions.

2 Methods

The basic strategy of LUTE is to create a discrete, quick-to-evaluate energy matrix that tells us everything we need to know for design purposes about the continuous energy landscape of a protein. We will now

describe this energy matrix and how it works.

Our goals in protein design (both GMEC [Leach and Lemon, 1998, Gainza *et al.*, 2012] and binding/partition function [Lilien *et al.*, 2005, Georgiev *et al.*, 2008] calculations) can be posed in terms of a discrete function $E(\mathbf{r})$ that maps an ordered list $\mathbf{r}$ of RCs to an energy. The list $\mathbf{r}$ contains exactly one RC per residue and thus represents a voxel $V(\mathbf{r})$ in conformation space, where a vector $\mathbf{x}$ of sequence and conformational degrees of freedom satisfies $\mathbf{x} \in V(\mathbf{r})$ if the degree-of-freedom bounds defined by each RC in $\mathbf{r}$ are respected by every degree of freedom in $\mathbf{x}$. The conformational degrees of freedom in $\mathbf{x}$ will generally be continuous internal coordinates, e.g., sidechain dihedrals. We let $E'(\mathbf{x})$ denote the energy of the protein system, as a function of all its degrees of freedom.

For calculation of the GMEC energy E_g, we wish to minimize $E'(\mathbf{x})$ with respect to $\mathbf{x}$. Letting R be the set of all possible voxels, the domain over which we minimize is a finite union of voxels $\bigcup_{\mathbf{r}\in R} V(\mathbf{r})$:

$$E_g = \min_{\mathbf{x}\in \bigcup_{\mathbf{r}\in R} V(\mathbf{r})} E'(\mathbf{x}) = \min_{\mathbf{r}\in R} \min_{\mathbf{x}\in V(\mathbf{r})} E'(\mathbf{x}), \tag{1}$$

which can be expressed in the form $\min_{\mathbf{r}\in R} E(\mathbf{r})$ where

$$E(\mathbf{r}) = \min_{\mathbf{x}\in V(\mathbf{r})} E'(\mathbf{x}). \tag{2}$$

Similarly, partition function calculations seek to calculate the partition function

$$q = \int_{\bigcup_{\mathbf{r}\in R} V(\mathbf{r})} \exp\left(-\frac{E'(\mathbf{x})}{RT}\right) d\mathbf{x} = \sum_{\mathbf{r}\in R} \int_{V(\mathbf{r})} \exp\left(-\frac{E'(\mathbf{x})}{RT}\right) d\mathbf{x} \tag{3}$$

where R is the gas constant and T is the temperature. Letting

$$E(\mathbf{r}) = -RT \ln\left(\int_{V(\mathbf{r})} \exp\left(-\frac{E'(\mathbf{x})}{RT}\right) d\mathbf{x}\right) \tag{4}$$

we have a formulation of q in terms of the discrete free energy function $E(\mathbf{r})$:

$$q = \sum_{\mathbf{r}\in R} \exp\left(-\frac{E(\mathbf{r})}{RT}\right) \tag{5}$$

Alternately, if we use the definition Eq. (2) to define $E(\mathbf{r})$, then Eq. (5) gives us the approximation used in [Lilien *et al.*, 2005] and [Georgiev *et al.*, 2008] for the partition function.

Because $\mathbf{r}$ is a discrete variable, the energy $E(\mathbf{r})$ can be decomposed as a sum of energies associated with tuples of RCs (Fig. 1). If all the RCs in a tuple are in the list $\mathbf{r}$, then that tuple's energy will contribute to $E(\mathbf{r})$. Most higher-order tuples of RCs consist of residues too far apart to have higher-order interactions,

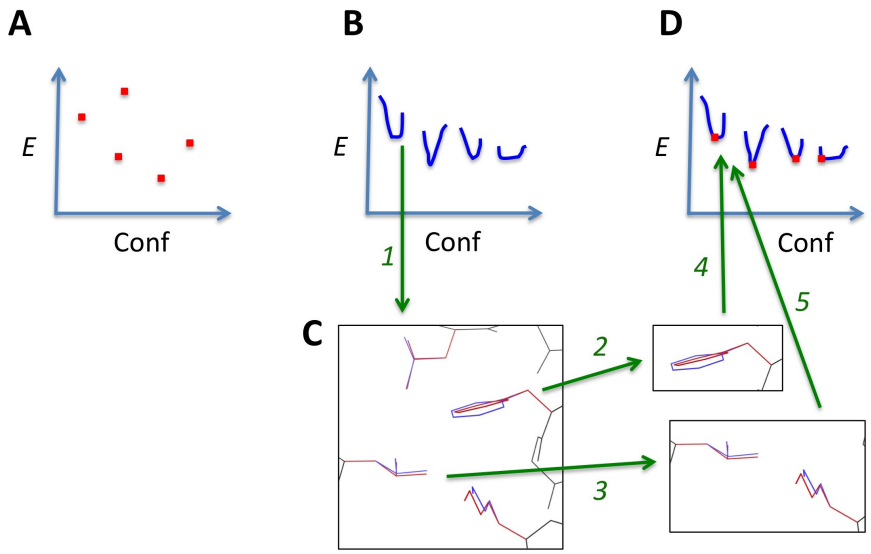

Figure 1: **LUTE makes continuously flexible design efficient by representing continuous flexibility using local, discrete energy terms.** (A) Protein design with discrete flexibility searches over a discrete (albeit large) conformational space ("Conf"), looking for low-energy ("E") conformations. Highly efficient algorithms like DEE/A* are available for this problem. (B) Protein design with continuous flexibility must search over a large space of voxels (blue) in a continuous conformational space, but we are usually interested only in the minimum-energy point of each voxel. We thus want a way to search combinatorially over these minimum-energy points. (C) The minimized energy of a voxel in protein conformational space depends on all rotamers in the voxel (arrow 1). But we can expand this minimized energy as a sum of local contributions from low-order tuples (e.g., pairs) of residues (arrows 2, 3). (Minimized conformations shown in red, ideal rotamers in blue). (D) This expansion, known as LUTE, gives us a discrete combinatorial search problem of the same form as protein design with discrete flexibility (arrows 4, 5). But this new discrete problem searches over the minimum-energy points (red) of voxels in continuous conformational space (blue). We can solve this problem very efficiently. Figure shows Leu 29, Leu 51, Phe 55, and Lys 59 of the Atx1 metallochaperone (PDB id 1CC8 [Rosenzweig *et al.*, 1999]).

and thus do not contribute significantly to the energy (see Section 1 and [Hallen *et al.*, 2015]). We can reduce the number of tuples needed substantially further if we only try to represent favorable, non-clashing conformations. By eliminating high-energy conformations, this restriction of conformational space greatly reduces the range of energy values over which $E(\mathbf{r})$ must be accurate. To achieve this, we prune tuples that cannot be part of favorable conformations, and consider only conformations whose tuples are all unpruned. Our expansion is much more efficient to compute after provably unfavorable tuples are pruned. Hence, we are able to represent the energy $E(\mathbf{r})$ as a local unpruned tuple expansion, or LUTE.

Let us consider a conformational space with continuous and discrete degrees of freedom, consisting of RCs , and a mapping $E(\mathbf{r})$ that we can readily calculate. For example, in a typical continuously flexible design, $E(\mathbf{r})$ is defined by Eq. (2), which we assume can be calculated by local minimization. Suppose we have a set T of tuples of RCs at different residue positions. T can contain pairs but also may contain triples, etc. We then define our local unpruned tuple expansion as a mapping $m : T \to \mathbb{R} \cup \{\perp\}$. m defines a real coefficient for each

tuple $t \in T$, except for pruned tuples, for which $m(t) = \perp$. Let $T_{\mathbf{r}}$ denote the set of tuples in T that consist only of RCs in $\mathbf{r}$. For example, if T is the set of all possible RC pairs, then $T_{\mathbf{r}}$ will consists of all pairs of RCs in the list of RCs $\mathbf{r}$. Then LUTE predicts $\mathbf{r}$ to be a pruned conformation if $m(t) = \perp$ for any $t \in T_{\mathbf{r}}$, and otherwise it predicts $E(\mathbf{r}) = \sum_{t \in T_{\mathbf{r}}} m(t)$. We refer to the data structure representing the mapping m as the *LUTE energy matrix*. We call it an energy matrix because it takes a form similar to that of traditional pairwise energy matrices [Desmet *et al.*, 1992, Leach and Lemon, 1998, Gainza *et al.*, 2012, Hallen *et al.*, 2013], although it contains significantly different numerical values when computed for the same design system. In practice, expansions in pairs and sometimes triples have worked well (see Section 3 and Supplementary Information (SI) [**SI**] Section B).

Most algorithms for protein design with discrete rotamers take a matrix of pairwise energies as input. By simply substituting a LUTE energy matrix for this pairwise energy matrix, we can convert any of these algorithms into an equally efficient design algorithm that searches a continuous search space instead of a discrete one, and/or that optimizes a non-pairwise energy function instead of a pairwise one. The LUTE energy matrix is computed once, before the search, which takes only polynomial time in the number of residues. For example, we need quadratic time to compute a LUTE matrix for which T is all pairs of RCs. Details of the computation by least squares of the LUTE matrix, and of the use of this matrix in search algorithms, are provided in the **SI** [**SI**].

3 Results

We present here complexity results and computational experiments regarding the performance of LUTE. In Section 3.1, we show that the combination of LUTE with the BWM* [Jou *et al.*, 2015] search algorithm is guaranteed to solve continuously flexible protein designs in polynomial time given a residue interaction graph with branch-width bounded by a constant. In Sections 3.2 and 3.3, we present 30 single-state and 17 multistate protein design calculations using LUTE. We measure the gains in efficiency provided by LUTE and its ability to accurately and efficiently perform calculations that, due to their large amount of continuous flexibility (Section 3.2) or non-pairwise energy function (Section 3.3), are inaccessible to previous algorithms. These results include designs with both continuous sidechain and backbone flexibility. Sidechain dihedrals were allowed 9° of continuous motion in either direction relative to the modal value for each sidechain rotamer [Janin *et al.*, 1978], while backbone flexibility (when present) was modeled as in [Hallen *et al.*, 2013].

3.1 Polynomial-time protein design with continuous flexibility

Protein design in the general case is NP-hard [Pierce and Winfree, 2002, Chazelle *et al.*, 2004]. In practice, however, many designs exhibit special properties that make them more tractable. For example, the residue interaction graph—the graph whose edges encode nonnegligible interactions between pairs of residues—of

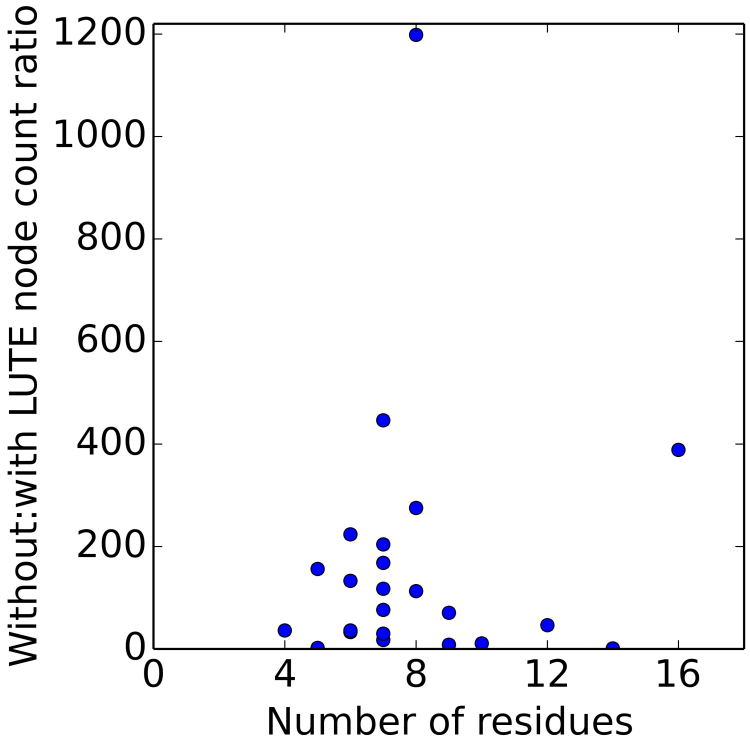

Figure 2: **LUTE markedly reduces the cost of continuously flexible conformational search.** Ratios (without LUTE:with LUTE) of the number of nodes in the A* tree before enumeration of the GMEC (or of the last conformation if several conformations closely spaced in energy were calculated; see [Hallen *et al.*, 2015]), versus number of flexible residues. A 20-residue sidechain placement with node ratio 2×10^5 is not shown because it would break the scale.

practical designs often has low branch-width. It has been previously shown that protein design with discrete rotamers can be performed in asymptotic time exponential only in the branch-width [Jou *et al.*, 2015] w. Furthermore, these branch-widths can be small irrespective of the number of mutable residues [Jou *et al.*, 2015]. Thus, for many protein designs with discrete rotamers the corresponding GMEC can be found in polynomial time. If one substitutes the LUTE matrix for the discrete pairwise energy matrix in this complexity result, then design with continuous flexibility and a constant-bounded branch-width can be solved in polynomial time as well. We can make this rigorous using the following theorem, whose proof is provided in SI [**SI**] Section E. In this theorem, a LUTE energy function is a function $E(\mathbf{r}) = \sum_{t \in T_{\mathbf{r}}} m(t)$, where $m : T \longrightarrow \mathbb{R} \cup \{\perp\}$ maps RC tuples to real coefficients (for this purpose, the coefficient $\perp$ of a pruned tuple is effectively ∞). Let n be the number of mutable residues, q be the maximum number of allowed RCs at any mutable residue position, and β_t and β_s be the time and space costs (respectively) to compute the branch-decomposition. This theorem establishes the complexity both of GMEC calculations and of enumeration of subsequent conformations in gap-free ascending order of energy. The latter is essential for calculation of partition functions, which can be used to account for entropy in predictions of binding [Lilien *et al.*, 2005, Georgiev *et al.*, 2008, Jou *et al.*, 2015].

Theorem 3.1. *For a LUTE energy function whose residue interaction graph has branch-width w, the GMEC can be computed in $\mathcal{O}(nw^2q^{\frac{3}{2}w} + \beta_t)$ time and $\mathcal{O}(nwq^{\frac{3}{2}w})$ space, and each additional conformation can be enumerated in order of LUTE energy in $\mathcal{O}(n \log q)$ time and $\mathcal{O}(n)$ space.*

3.2 Continuous flexibility

LUTE single-state designs were run on 23 protein design systems from [Hallen *et al.*, 2015] with 4-16 mutable residues, as well as five larger systems (17-40 mutable residues), to measure the efficiency of LUTE and to observe the behavior of LUTE on the larger systems. Many of these larger systems are intractable by previous methods (except *post-hoc* minimization methods that do not account for continuous flexibility during search). The results show that the discrete DEE/A* search with LUTE is dramatically more efficient even compared to EPIC, which offers previously state-of-the-art efficiency for continuously flexible design [Hallen *et al.*, 2015] (Fig. 2). They also demonstrate that LUTE can handle very large continuously flexible designs—including a 40-residue sidechain placement, which covers a large fraction of the residues in the Atx1 metallochaperone (Fig. 4, left), and a 20-residue design on the same structure with 5 amino-acid types allowed at every position. Furthermore, the LUTE energy matrix consistently represented the true energy landscape very closely (Fig. 3). Optimal sequences and conformations with LUTE differed significantly from the same designs run without continuous flexibility: the same top conformation was returned in only 2 of the 28 single-state designs. On average, 31% of the RCs in the optimal conformations differed from each other. This is consistent with previous work showing that protein design calculations with and without continuous

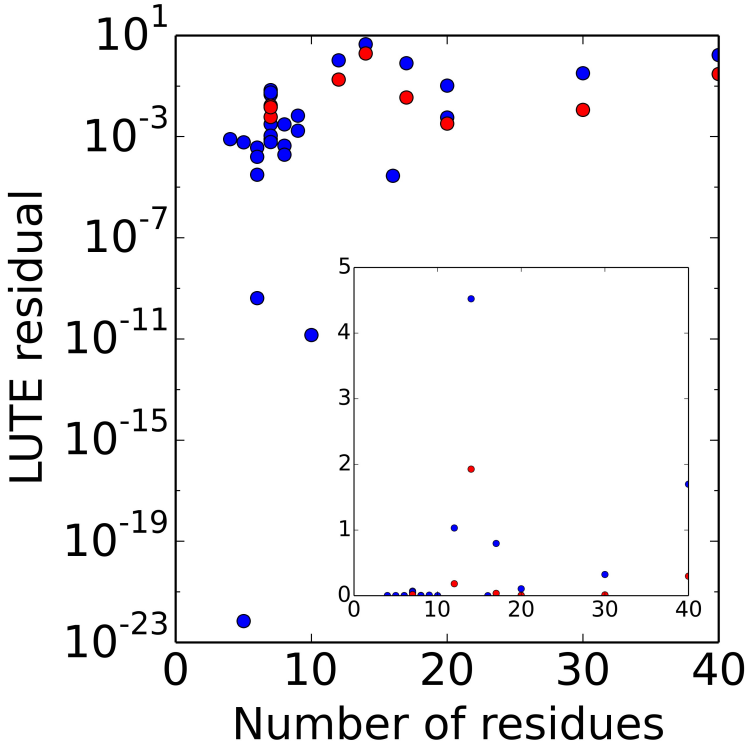

Figure 3: **LUTE accurately represents continuously minimized energies.** Residuals for LUTE ((kcal/mol)^2) on the cross-validation data set, measuring the difference between the EPIC energy and a pairwise expansion (blue) or one with sparse triples (red; computed only if pairwise residual exceeded 0.01). x axis: number of flexible residues. Inset: All the same data plotted on a linear scale.

flexibility differ significantly in their results [Gainza *et al.*, 2012, Hallen *et al.*, 2013].

For many systems, LUTE achieved a fit with residual under 0.01 (kcal/mol)^2 with only a pairwise expansion. In cases when the pairwise expansion's residual was higher, an expansion in sparse triples was performed instead. In all but one case, the triples expansion's residual was less than thermal energy at room temperature (0.59 kcal/mol, i.e., 0.35 (kcal/mol)^2), and thus deemed insignificant.

The one outlier case was a 14-residue design on ponsin (PDB id: 2O9S). It exhibited significant local minimization errors, which caused even the matrix of pairwise lower-bound energies (computed before LUTE precomputation begins) to have errors of at least ∼10 kcal/mol. These errors indicate the failure of either our local minimizer or our assumption that local minimization suffices within RCs. As a result of these errors, the LUTE residual even with triples was 1.9 kcal/mol for this system, seven times worst than the next worst residual (the 40-residue Atx1 design). Our software now detects this problem and warns the user before the LUTE computation begins.

17 multistate protein designs were also performed, using a combination of LUTE with our COMETS [Hallen and Donald, 2015] multistate protein design algorithm (see SI [**SI**] Section C). The systems from

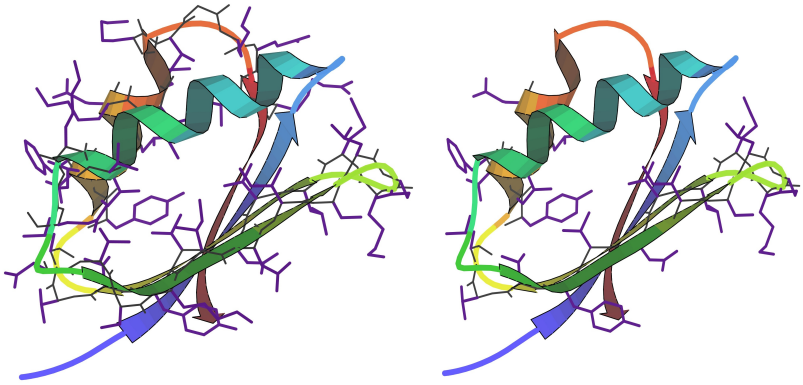

Figure 4: **LUTE enables very large provably accurate protein designs with continuous flexibility (left) and with Poisson-Boltzmann energy functions (right).** Left: previously, protein designs with continuous flexibility only finished when performed with significantly fewer flexible residues, compared to designs with discrete rotamers. Even 20-residue designs were often intractable. But LUTE solved a sidechain placement problem with continuous flexibility in which 40 residues (purple) were made flexible in the Atx1 metallochaperone (PDB id 1CC8 [Rosenzweig *et al.*, 1999]). Right: previous designs using the Poisson-Boltzmann energy function could not optimize this function directly, but only used Poisson-Boltzmann energies to rerank top hits from optimization of a simpler, pairwise energy function. But LUTE can optimize the Poisson-Boltzmann energy function directly—e.g., in a sidechain placement of 20 residues (purple) of Atx1.

these designs were taken from [Hallen and Donald, 2015]; details are provided in SI [**SI**] Section G. The same designs were run with and without continuous flexibility, with LUTE used in the continuous case. As discussed in [Hallen and Donald, 2015], COMETS provably returns the same results as exhaustive search over sequences, but it provides a speedup compared to that exhaustive search by (a) considering only a portion of the sequences in the search space explicitly, and (b) only performing a full conformational optimization for a small portion of the sequences in (a). However, previously [Hallen and Donald, 2015] (a) was only significant in designs without continuous flexibility, and (b) was much more pronounced without continuous flexibility. LUTE brings continuously flexible COMETS designs up to speed with discrete designs on the same system (SI [**SI**] Fig. S3).

3.3 Designs that provably optimize Poisson-Boltzmann energies

We also ran LUTE conformational optimization calculations on two proteins using the Poisson-Boltzmann energy function, which is non-pairwise. This energy was evaluated using Delphi [Nicholls and Honig, 1991, Rochia *et al.*, 2002] in place of the pairwise EEF1 [Lazaridis and Karplus, 1999] solvation energy that is used by default in OSPREY. Interestingly, triple energies did not provide significant benefit here, but LUTE was found to describe the Poisson-Boltzmann energy landscape with a high degree of accuracy. Previous work has shown that an accurate pairwise representation can be obtained for Poisson-Boltzmann energies of *discrete, rigid* rotamers [Vizcarra *et al.*, 2008], but our LUTE results show that a very accurate representation of *continuously minimized Poisson-Boltzmann energies* is possible as well. With continuous flexibility, a 6-

residue sidechain placement on the unliganded TIR1/IAA7 complex (PDB code 2P1Q [Tan *et al.*, 2007]) with continuous flexibility achieved a total residual of 6×10^{-4} and took about 4 days. Furthermore, a 20-residue sidechain placement without continuous flexibility on the bacterial metallochaperone protein Atx1 (PDB code 1CC8 [Rosenzweig *et al.*, 1999]; Fig. 4, right) was solved in 2.5 hours, with total residual 0.04 $(\text{kcal/mol})^2$. Unlike previous protein design calculations that use Poisson-Boltzmann energies, our new calculations provably return the minimum of the (LUTE-approximated) Poisson-Boltzmann energy over the entire conformational space, rather than simply over a set of top hits from an initial search that used a cheaper energy function.

4 Conclusions

The protein design problem enjoys a wide array of powerful algorithms for conformational and sequence search. These algorithms take a discrete energy matrix and perform sequence optimizations, both in the single-state and multistate cases. At the same time, previous work in bioinformatics and quantum chemistry has made great progress toward quantitatively accurate modeling of the flexibility and energy landscapes of biomolecular systems. Uniting these fields to perform designs with highly realistic modeling would result in great biomedical impact, both in protein and drug design. However, because state-of-the-art flexibility and energy modeling methods do not produce a discrete matrix, there is a gap between these fields. LUTE offers a strategy to bridge this gap. By representing continuous flexibility and general energy functions in a discrete matrix, it greatly increases the realism of the modeling that discrete combinatorial optimization algorithms like DEE/A* can directly accommodate. We thus believe that LUTE can serve as a foundation for greatly improved biomolecular design protocols.

Acknowledgments We would like to thank Drs. Kyle Roberts and Pablo Gainza for providing PDB files and scripts for testing; all members of the Donald lab for helpful comments; and the PhRMA and Dolores Zohrab Liebmann foundations (MAH) and NIH (grant R01-GM-78031 to BRD) for funding.

Author Disclosure Statement. No competing financial interests exist.

References

[**SI**] Supplementary material available online:
http://www.cs.duke.edu/donaldlab/Supplementary/jcb16/lute/

[Chazelle *et al.*, 2004] Chazelle, B., Kingsford, C., and Singh, M. 2004. A semidefinite programming approach to side chain positioning with new rounding strategies. *INFORMS Journal on Computing, Computational Biology Special Issue* 16, 380–392.

[Chen *et al.*, 2009] Chen, C.-Y., Georgiev, I., Anderson, A. C., and Donald, B. R. 2009. Computational structure-based redesign of enzyme activity. *Proceedings of the National Academy of Sciences of the USA* 106, 3764–3769.

[Čížek, 2009] Čížek, J. 2009. On the use of the cluster expansion and the technique of diagrams in calculations of correlation effects in atoms and molecules. In *Correlation Effects in Atoms and Molecules*, volume 14 of *Advances in Chemical Physics*, pages 35–90. John Wiley and Sons.

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

[Donald, 2011] Donald, B. R. 2011. *Algorithms in Structural Molecular Biology.* MIT Press, Cambridge, MA.

[Flocke and Bartlett, 2004] Flocke, N. and Bartlett, R. J. 2004. A natural linear-scaling coupled-cluster method. *Journal of Chemical Physics* 121, 10935–10944.

[Floudas *et al.*, 1999] Floudas, C. A., Klepeis, J. L., and Pardalos, P. M. 1999. Global optimization approaches in protein folding and peptide docking. In *Mathematical Support for Molecular Biology*, volume 47 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 141–172. American Mathematical Society.

[Frey *et al.*, 2010] Frey, K. M., Georgiev, I., Donald, B. R., and Anderson, A. C. 2010. Predicting resistance mutations using protein design algorithms. *Proceedings of the National Academy of Sciences of the USA* 107, 13707–13712.

[Gainza *et al.*, 2012] Gainza, P., Roberts, K., and Donald, B. R. 2012. Protein design using continuous rotamers. *PLoS Computational Biology* 8, e1002335.

[Gainza *et al.*, 2013] Gainza, P., Roberts, K. E., Georgiev, I., Lilien, R. H., Keedy, D. A., Chen, C.-Y., Reza, F., Anderson, A. C., Richardson, D. C., Richardson, J. S., and Donald, B. R. 2013. OSPREY: Protein design with ensembles, flexibility, and provable algorithms. *Methods in Enzymology* 523, 87–107.

[Georgiev *et al.*, 2012] Georgiev, I., Acharya, P., Schmidt, S., Li, Y., Wycuff, D., Ofek, G., Doria-Rose, N., Luongo, T., Yang, Y., Zhou, T., Donald, B. R., Mascola, J., and Kwong, P. 2012. Design of epitope-specific probes for sera analysis and antibody isolation. *Retrovirology* 9, P50.

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

[Georgiev *et al.*, 2008] 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. *Journal of Computational Chemistry* 29, 1527–1542.

[Georgiev *et al.*, 2009] Georgiev, I., Roberts, K. E., Gainza, P., Hallen, M. A., and Donald, B. R. 2009. OSPREY (Open Source Protein Redesign for You) user manual. Available online: `www.cs.duke.edu/donaldlab/software.php`. Updated, 2015. 94 pages.

[Georgiev *et al.*, 2014] Georgiev, I. S., Rudicell, R. S., Saunders, K. O., Shi, W., Kirys, T., McKee, K., O'Dell, S., Chuang, G.-Y., Yang, Z.-Y., Ofek, G., Connors, M., Mascola, J. R., Nabel, G. J., and Kwong, P. D. 2014. Antibodies VRC01 and 10E8 neutralize HIV-1 with high breadth and potency even with Ig-framework regions substantially reverted to germline. *The Journal of Immunology* 192, 1100–1106.

[Gorczynski *et al.*, 2007] Gorczynski, M. J., Grembecka, J., Zhou, Y., Kong, Y., Roudaia, L., Douvas, M. G., Newman, M., Bielnicka, I., Baber, G., Corpora, T., Shi, J., Sridharan, M., Lilien, R., Donald, B. R., Speck, N. A., Brown, M. L., and Bushweller, J. H. 2007. Allosteric inhibition of the protein-protein interaction between the leukemia-associated proteins Runx1 and CBFβ. *Chemistry and Biology* 14, 1186–1197.

[Gordon and Mayo, 1998] Gordon, D. B. and Mayo, S. L. 1998. Radical performance enhancements for combinatorial optimization algorithms based on the dead-end elimination theorem. *Journal of Computational Chemistry* 19, 1505–1514.

[Grigoryan *et al.*, 2009] Grigoryan, G., Reinke, A. W., and Keating, A. E. 2009. Design of protein-interaction specificity affords selective bZIP-binding peptides. *Nature* 458, 859–864.

[Grigoryan *et al.*, 2006] Grigoryan, G., Zhou, F., Lustig, S. R., Ceder, G., Morgan, D., and Keating, A. E. 2006. Ultra-fast evaluation of protein energies directly from sequence. *PLoS Computational Biology* 2, e63.

[Hallen and Donald, 2015] Hallen, M. A. and Donald, B. R. 2015. COMETS (Constrained Optimization of Multistate Energies by Tree Search): A provable and efficient algorithm to optimize binding affinity and specificity with respect to sequence. In *Research in Computational Molecular Biology*, volume 9029 of *Lecture Notes in Computer Science*, pages 122–135. Springer International Publishing. ISBN 978-3-319-16706-0.

[Hallen *et al.*, 2015] Hallen, M. A., Gainza, P., and Donald, B. R. 2015. A compact representation of continuous energy surfaces for more efficient protein design. *Journal of Chemical Theory and Computation* 11, 2292–2306.

[Hallen *et al.*, 2013] 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: Structure, Function and Bioinformatics* 81, 18–39.

[Hart *et al.*, 1968] Hart, P. E., Nilsson, N. J., and Raphael, B. 1968. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics* 4, 100–107.

[Janin *et al.*, 1978] Janin, J., Wodak, S., Levitt, M., and Maigret, B. 1978. Conformation of amino acid side-chains in proteins. *Journal of Molecular Biology* 125, 357–386.

[Jou *et al.*, 2015] Jou, J. D., Jain, S., Georgiev, I., and Donald, B. R. 2015. BWM*: A novel, provable, ensemble-based dynamic programming algorithm for sparse approximations of computational protein design. *Journal of Computational Biology* In press.

[Karanicolas and Kuhlman, 2009] Karanicolas, J. and Kuhlman, B. 2009. Computational design of affinity and specificity at protein-protein interfaces. *Current Opinion in Structural Biology* 19, 458–463.

[Kingsford *et al.*, 2005] 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 and Baker, 2000] Kuhlman, B. and Baker, D. 2000. Native protein sequences are close to optimal for their structures. *Proceedings of the National Academy of Sciences of the USA* 97, 10383–10388.

[Lazaridis and Karplus, 1999] Lazaridis, T. and Karplus, M. 1999. Effective energy function for proteins in solution. *Proteins: Structure, Function, and Bioinformatics* 35, 133–152.

[Leach and Lemon, 1998] 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: Structure, Function, and Bioinformatics* 33, 227–239.

[Leaver-Fay *et al.*, 2011] Leaver-Fay, A., Jacak, R., Stranges, P. B., and Kuhlman, B. 2011. A generic program for multistate protein design. *PLoS One* 6, e20937.

[Lee and Levitt, 1991] 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.

[Lilien *et al.*, 2005] Lilien, R. H., Stevens, B. W., Anderson, A. C., and 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.

[LuCore *et al.*, 2015] LuCore, S. D., Litman, J. M., Powers, K. T., Gao, S., Lynn, A. M., Tollefson, W. T. A., Fenn, T. D., Washington, M. T., and Schnieders, M. J. 2015. Dead-end elimination with a polarizable force field repacks PCNA structures. *Biophysical Journal* 109, 816–826.

[Nicholls and Honig, 1991] Nicholls, A. and Honig, B. 1991. A rapid finite difference algorithm, utilizing successive over-relaxation to solve the Poisson-Boltzmann equation. *Journal of Computational Chemistry* 12, 435–445.

[Nocedal and Wright, 2006] Nocedal, J. and Wright, S. J. 2006. *Numerical Optimization*. Springer-Verlag, Berlin, 2nd edition.

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

[Roberts *et al.*, 2012] Roberts, K. E., Cushing, P. R., Boisguerin, P., Madden, D. R., and Donald, B. R. 2012. Computational design of a PDZ domain peptide inhibitor that rescues CFTR activity. *PLoS Computational Biology* 8, e1002477.

[Roberts *et al.*, 2015] Roberts, K. E., Gainza, P., Hallen, M. A., and Donald, B. R. 2015. Fast gap-free enumeration of conformations and sequences for protein design. *Proteins: Structure, Function, and Bioinformatics* 83, 1859–1877.

[Rochia *et al.*, 2002] Rochia, W., Sridharan, S., Nicholls, A., Alexov, E., Chiabrera, A., and Honig, B. 2002. Rapid grid-based construction of the molecular surface and the use of induced surface charge to calculate reaction field energies: Applications to the molecular systems and geometric objects. *Journal of Computational Chemistry* 23, 128–137.

[Rosenzweig *et al.*, 1999] Rosenzweig, A. C., Huffman, D. L., Hou, M. Y., Wernimont, A. K., Pufahl, R. A., and O'Halloran, T. V. 1999. Crystal structure of the Atx1 metallochaperone protein at 1.02 A resolution. *Structure* 7, 605–617.

[Rudicell *et al.*, 2014] Rudicell, R. S., Kwon, Y. D., Ko, S.-Y., Pegu, A., Louder, M. K., Georgiev, I. S., Wu, X., Zhu, J., Boyington, J. C., Chen, X., Shi, W., Yang, Z.-Y., Doria-Rose, N. A., McKee, K., O'Dell, S.,

Schmidt, S. D., Chuang, G.-Y., Druz, A., Soto, C., Yang, Y., Zhang, B., Zhou, T., Todd, J.-P., Lloyd, K. E., Eudailey, J., Roberts, K. E., Donald, B. R., Bailer, R. T., Ledgerwood, J., Program, N. C. S., Mullikin, J. C., Shapiro, L., Koup, R. A., Graham, B. S., Nason, M. C., Connors, M., Haynes, B. F., Rao, S. S., Roederer, M., Kwong, P. D., Mascola, J. R., and Nabel, G. J. 2014. Enhanced potency of a broadly neutralizing HIV-1 antibody *in vitro* improves protection against lentiviral infection *in vivo*. *Journal of Virology* 88, 12669–12682.

[Sitkoff *et al.*, 1994] Sitkoff, D., Sharp, K. A., and Honig, B. 1994. Accurate calculation of hydration free energies using macroscopic solvent models. *Journal of Physical Chemistry* 98, 1978–1988.

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

[Tan *et al.*, 2007] Tan, X., Calderón-Villalobos, L. I. A., Sharon, M., Zheng, C., Robinson, C. V., Estelle, M., and Zheng, N. 2007. Mechanism of auxin perception by the TIR1 ubiquitin ligase. *Nature* 446, 640–645.

[Traoré *et al.*, 2013] Traoré, S., Allouche, D., André, I., de Givry, S., Katsirelos, G., Schiex, T., and Barbe, S. 2013. A new framework for computational protein design through cost function network optimization. *Bioinformatics* 29, 2129–2136.

[Traoré *et al.*, 2016] Traoré, S., Roberts, K. E., Allouche, D., Donald, B. R., André, I., Schiex, T., and Barbe, S. 2016. Fast search algorithms for computational protein design. *Journal of Computational Chemistry* .

[Vizcarra *et al.*, 2008] Vizcarra, C. L., Zhang, N., Marshall, S. A., Wingreen, N. S., Zeng, C., and Mayo, S. L. 2008. An improved pairwise decomposable finite-difference Poisson-Boltzmann method for computational protein design. *Journal of Computational Chemistry* 29, 1153–1162.

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

[Yanover *et al.*, 2007] Yanover, C., Fromer, M., and Shifman, J. M. 2007. Dead-end elimination for multistate protein design. *Journal of Computational Chemistry* 28, 2122–2129.

[Zhang and Zhang, 2003] Zhang, D. W. and Zhang, J. Z. H. 2003. Molecular fractionation with conjugate caps for full quantum mechanical calculation of protein-molecule interaction energy. *Journal of Chemical Physics* 119, 3599–3605.