Algorithm for backrub motions in protein design

Ivelin Georgiev¹, Daniel Keedy², Jane S. Richardson², David C. Richardson² and Bruce R. Donald¹,²,*

¹Department of Computer Science, Duke University and ²Department of Biochemistry, Duke University Medical Center, Durham, NC 27708, USA

ABSTRACT

Motivation: The Backrub is a small but kinematically efficient side-chain-coupled local backbone motion frequently observed in atomic-resolution crystal structures of proteins. A backrub shifts the C<sub>α</sub>-C<sub>β</sub> orientation of a given side-chain by rigid-body dipeptide rotation plus smaller individual rotations of the two peptides, with virtually no change in the rest of the protein. Backrubs can therefore provide a biophysically realistic model of local backbone flexibility for structure-based protein design. Previously, however, backrub motions were applied via manual interactive model-building, so their incorporation into a protein design algorithm (a simultaneous search over mutation and backbone/side-chain conformation space) was infeasible.

Results: We present a combinatorial search algorithm for protein design that incorporates an automated procedure for local backbone flexibility via backrub motions. We further derive a dead-end elimination (DEE)-based pruning stage for pruning candidate rotamers that, in contrast to previous DEE algorithms, is provably accurate with backrub motions. Our backrub-based algorithm successfully predicts alternate side-chain conformations from four ≤ 0.9 Å resolution structures, confirming the suitability of the automated backrub procedure. Finally, the application of our algorithm to redesign two different proteins is shown to identify a large number of lower-energy conformations and mutation sequences that would have been ignored by a rigid-backbone model.

Availability: Contact authors for source code.

Contact: brd+ismb08@cs.duke.edu

1 INTRODUCTION

Protein design algorithms aim at identifying protein mutation sequences with desired improved or novel properties, such as: stability (Kortgegain et al., 2005; Malakas Kas and Mayo, 1998), specificity (Havranek and Harbury, 2003; Kortemme et al., 2004; Lilien et al., 2005; Looger et al., 2003), binding affinity (Lippow et al., 2007), enzymatic function (Jiang et al., 2008; Lasila et al., 2006; Stevens et al., 2006) or even overall fold (Kuhlman et al., 2003). Typically, the input model for a structure-based protein design algorithm includes the following: (1) an initial (usually rigid) backbone structure, used as a template for the redesign; (2) a rotamer library (Dunbrack, 2002; Lovell et al., 2000; Ponder and Richards, 1987) of low-energy side-chain conformations that discretizes the continuous side-chain conformation space, and thus makes the computational search feasible; and (3) a pairwise energy function (Gordon et al., 1999; Kuhlman and Baker, 2000; Viscaarza and Mayo, 2005) for scoring and ranking the algorithm predictions. To further improve the accuracy of the model, extended rotamer libraries (De Maeyer et al., 1997), flexible rotamers (Georgiev et al., 2008; Mendes et al., 1999) and different levels of backbone flexibility (Desjarlais and Handel, 1999; Fung et al., 2007; Georgiev and Donald, 2007; Harbury et al., 1998; Kuhlman et al., 2003; Su and Mayo, 1997; Zanghellini et al., 2006) have also been introduced. Incorporating additional backbone/side-chain flexibility into the computational model allows the identification of lower energy mutations/conformations that would have been ignored by a rigid model (Georgiev and Donald, 2007). The combinatorial problem of considering all possible mutations and conformations for each of the mutable residue positions in a protein poses a significant computational challenge for protein design algorithms. In fact, it has been shown that finding the optimal solution, the Global Minimum Energy Conformation (GMEC), for a given input model with a rigid backbone, a rotamer library and a pairwise energy function, is NP-hard (Chazelle et al., 2004; Pierce and Winfree, 2002). For a protein with a rigid backbone, n mutable residue positions and at most q rotamers per residue position, a brute-force enumeration procedure must consider O(q<sup>n</sup>) possible conformations. Hence, many heuristic techniques, such as Monte Carlo and Self-Consistent Mean Field, have been applied in protein design (Desjarlais and Handel, 1999; Hu and Kuhlman, 2006; Jin et al., 2003; Street and Mayo, 1999; Voigt et al., 2001).

Such heuristics are generally fast since only a small subset of the possible conformations is enumerated, but they cannot guarantee the identification of the GMEC for the given input model and can make significant errors (Voigt et al., 2000).

As an alternative, Dead-End Elimination (DEE; Desmet et al., 1992; Gordon et al., 2003) is a provably accurate deterministic algorithm that efficiently reduces the mutation/conformation search space, while enjoying provable guarantees with respect to the GMEC. DEE uses pairwise upper and lower bounds on the rotameric energy interactions to efficiently prune rotamers that are provably not part of the GMEC. Effectively, the DEE-based pruning stage reduces the base q of the enumeration exponent, typically making the subsequent enumeration of the remaining unpruned conformations computationally feasible. Depending on the types of flexibility allowed, several DEE-based algorithms have been derived, in order to guarantee the identification of the GMEC for the respective model. Traditional DEE (Desmet et al., 1992; Goldstein, 1994; Lasters and Desmet, 1993; Looger and Hellinga, 2001; Pierce et al., 2000; Yanover et al., 2007) is only provably accurate for a model with a rigid backbone and rigid rotamers. The MinDEE pruning criterion (Georgiev et al., 2006, 2008) is provably accurate for a model with a rigid backbone and flexible rotamers over a continuous voxel of side-chain conformation space. The BD DEE-based pruning criterion (Georgiev and Donald, 2007) is provably accurate for a model with rigid rotamers and a continuous family...
of backbone conformations. In general, unlike heuristic approaches, provably accurate algorithms can guarantee the identification of the optimal solution for a given design problem. Furthermore, with provably accurate algorithms, feedback from in vitro experiments can be more reliably incorporated into the model, since discrepancies between experimental results and computational predictions can be attributed solely to deficiencies of the model (as opposed to the algorithm) (Georgiev and Donald, 2007).

1.1 BD: DEE with backbone flexibility

BD (Georgiev and Donald, 2007) is a DEE-based algorithm that, in contrast to traditional DEE, is provably accurate with rigid rotamers and backbone flexibility. BD places restraining boxes around each residue in a protein, in order to define a continuous family of backbone conformations with small ($\phi$, $\psi$) changes that nonetheless can cause global shifts in the backbone coordinates. Upper and lower bounds on the pairwise rotameric energy interactions are then precomputed within the defined restraining boxes and used to determine which rotamers are provably not part of the respective GMEC. The BD algorithm consists of two stages: (1) first, BD is used to prune a large portion of the candidate rotamers; (2) using $A^*$ search (Leach and Lemon, 1998), the remaining unpruned conformations are then enumerated in order of increasing lower bounds on their energies, in order to obtain the GMEC. When tested on two different protein systems, BD was shown to generate conformations and sequences with significantly lower energies than traditional DEE (albeit at slower running times), thus confirming the potential benefit of incorporating backbone flexibility.

1.2 Backrubs

Based on stereochemical intuition, the existence of a subtle backbone motion coupled to rotamer jumps has long been suspected. Such a motion, the ‘backrub’, was recently confirmed by closely examining the electron density for side-chains modeled as alternates in very high-resolution crystal structures and inferring that the backbone must have shifted between the two conformations to maintain reasonably ideal geometry (Davis et al., 2006). It is conservatively estimated that 3% of all residues undergo backrubs, with a large fraction occurring at the protein surface, most likely reacting to bombardment from solvent molecules. In addition to modeling dynamics, we show that backrubs can allow rotamer changes. Hence, by deduction, they can accommodate mutations to amino acid types for which no rotamers fit in the original backbone. Therefore, it is reasonable to assume that backrubs may play an evolutionary role. Such an assumption is of course impossible to demonstrate from single high-resolution structures and, due to coordinate error on the level of backbone shifts, is also difficult to tease out by comparing otherwise identical-in-sequence point mutant structures. However, one way to address the question is by investigating the effects of backrubs in protein design, which is essentially a guided form of evolution that contributes to our knowledge of the determinants of protein packing and folding. If backrubs enable a provable algorithm to design proteins with low energies, we can be confident that they may also contribute on an evolutionary timescale (Davis et al., 2006).

1.3 Contributions of the article

Backbone flexibility in BD is represented by global backbone motions: a change in the backbone conformation of residue position $i$ tends to propagate along the rest of the chain (Georgiev and Donald, 2007). In contrast, in this article, we evaluate the benefits of allowing local backbone flexibility via backrubs. The local backbone motions and the global BD motions represent very different types of flexibility, and should thus be viewed as complementary, rather than competing, approaches for backbone flexibility in protein design.

By using manual interactive model building, the BACKRUB tool (Davis et al., 2006) allows a user to choose the three rotation angles (Section 1.2) and apply the corresponding backrub motion. However, no automated backrub procedure has been previously developed. In this article, we present a straightforward approach that automates the backrub computation (Section 2.2). We further apply this approach as part of a combinatorial search algorithm for protein design (Section 3). The latter captures a theme in computational protein design. Many modeling improvements, such as backrubs, can be suggested for a single protein structure or sequence. A design
algorithm must ‘lift’ each such model to a pairwise bounding and pruning mechanism. Such a mechanism is usually a non-trivial exercise in algorithm design (viz. Sections 2-4), and a prerequisite before the new model (in this article: backrub) can be exploited in a combinatorial search (e.g. DEE) across all allowed protein mutations and conformations.

We show that choosing the primary rotation angle \( \theta_{1,3} \) can define the flanking rotations \( \theta_{1,2} \) and \( \theta_{2,3} \) via kinematics and minimization, and is hence sufficient to parameterize a backrub for a given residue, resulting in a 1 degree-of-freedom per residue design problem. Hence, defining a finite set of possible backbone conformations by sampling the single \( \theta_{1,3} \) parameter for each flexible residue, should not be prone to severe undersampling. A simple approach could then apply traditional DEE separately (1) Backrub DEE (\( \text{Brdee} \)). In particular, we make the following contributions in this article: (2) An automated procedure for the generation and energy-based ranking of backrub motions; (3) A novel algorithm for protein design, incorporating our automated rotamers that are provably not part of the GMEC for a finite set; and (4) We first apply our algorithms to predict alternate conformations from crystal structures. Next, we apply them to redesign two proteins: (a) the adenylation domain of the non-ribosomal peptide synthetase (NRPS) enzyme Gramicidin Synthetase A (GsAA-PheA) and (b) the core of the \( \beta_1 \) domain of protein G (Gj1). Gj1 is a small protein that is a suitable benchmark for protein design algorithms (Georgiev and Donald, 2007). The redesign of GsAA-PheA has potential significant biomedical application to the design of novel antibiotics (Stevens et al., 2006).

2 APPROACH

2.1 Brdee

The BD algorithm (see Section 1.1) is applicable for protein design problems where backbone conformations are represented by a bounded continuous family of solutions. In this section, we first show how analogous ideas can be exploited to derive \( \text{Brdee} \), a provably accurate pruning algorithm for problems where backbone conformations are represented by a finite set of solutions. We then specialize \( \text{Brdee} \) for backrub motions.

2.1.1 DEE for finite backbone sets. First, we make the following definitions. We will define the protein template \( t \) to include the protein backbone, as well as the side-chains of all residues that are fixed and are not subject to rotamer-based modeling; let \( E_t(R_t) \) be the template energy of the system for a given backbone \( R_t \). Let \( i \) denote rotamer identity \( \gamma \) at residue position \( i \). Then, we define \( E(i,R_t) \) and \( E(i,j,R_t) \) to be, respectively, the self-energy of \( i \) (the sum of the intra-rotamer and rotamer-to-template energies for \( i \)) and the pairwise energy between rotamers \( i \) and \( j \) when backbone configuration \( R_t \) is assumed.

Now, let us have a subset \( Q \) of residues that are modeled as flexible. Let \( Y \) be the discrete set of allowed backbone conformations \( R_t \). Let \( Z(i) \) be the set of side-chain dihedral conformations for rotamer \( i \), and let the Cartesian product \( Y \times Z(i) \) be the set of possible conformations of rotamer \( i \) and its associated backbone. Here, we will use the name of rigid rotamers \( (i\{Z(i)\})=1 \), although the following derivation holds for \( Z(i) \) being an integer as well. The following lower and upper bound definitions can now be made:

\[
E_{\min} = \min_{R_t \in Y} E_t(R_t); \quad E_{\max} = \max_{R_t \in Y} E_t(R_t);
\]

(1)

Here, \( E_{\min} \) represents a lower bound on the template energy for the given set of allowed backbone conformations. Similarly, \( E_{\max} \) is an upper bound on the template energy, and \( E_{\max} - E_{\min} \) represents the range of possible template energies.

We define the following rotamer-based terms:

\[
E_{\min}^g(i) = \min_{Z(i)} E(i,Z(i)); \quad E_{\max}^g(i) = \max_{Z(i)} E(i,Z(i))
\]

(3)

Here, \( E_{\min}^g(i) \) represents a lower bound on the self-energy of rotamer \( i \) for the given set of allowed backbone conformations, while \( E_{\max}^g(i,j) \) represents a lower bound on the pairwise energy between rotamers \( i \) and \( j \). The respective upper bounds \( E_{\max}^g(i) \) and \( E_{\max}^g(i,j) \) of ranges of possible energies \( E_{\max}(i) \) and \( E_{\max}(i,j) \) are defined analogously.

The Backrub pruning criterion for a given rotamer \( i \) is then defined to be:

\[
E_{\min}^g(i) + \sum_j \min_j E_{\max}^g(j,i) - E_{\min}^g(i) - \sum_j \max_j E_{\min}^g(j,i) > E_{\max}^g(i) + \sum_j \max_j E_{\max}^g(j,i),
\]

(4)

where \( j \neq i, k \neq j \) and \( i \) is over the rotor sets \( R_i \) for given residues \( j \) and \( k \).

When Equation (5) holds, rotamer \( i \) can be pruned from further consideration, since it provably cannot belong to the GMEC for the allowed set \( Y \) of backbone conformations. The proof of this claim is identical to the proof of Proposition 1 in (Georgiev and Donald, 2007). The inclusion of the \( E_{\max}(\cdot) \) terms in Equation (5) accounts for possible energy changes due to changes in the backbone conformation. Hence, unlike traditional DEE, by appropriately manipulating lower and upper energy bounds, Equation (5) simultaneously takes into account all possible conformations from a given finite set of backbones. Since the \( E_{\max}(\cdot) \) terms can be precomputed, the cost of evaluating Equation (5) is \( \mathcal{O}(n^2) \) for \( n \) residue positions and at most \( q \) rotamers per position, equivalent to the cost of the corresponding traditional DEE. MinDEE and BD conditions.

It should be noted that the form of Equation (5) is identical to the initial BD pruning condition (Georgiev and Donald, 2007). The major difference, however, is that in BD the \( E_{\max}(\cdot) \) and \( E_{\min}(\cdot) \) are defined over a bounded infinite and continuous voxel of backbone
conformation space; in Brdee, these terms are defined over a finite set of backbone conformations. BD and Brdee are thus applicable to significantly different protein design problems.

2.1.2 Specialization for backrubs. Since our interest is in introducing backrubs into protein design, we will now consider the case where the set Y is defined using backrubs. For each residue i ∈ Q, let Ȳi be the set of allowed backbone conformations (resulting from backrub motions) for residues i − 1, i, and i + 1. To avoid combinatorial blowup, we will require that the backrub independence condition (BIC) holds. Let Aij be the set of atoms that can change their position (3D coordinates) upon a backrub at residue i ∈ Q; similarly, we define Aij. BIC then ensures that Aij ∩ Aij = ∅, i.e. that there is no overlap between Aij and Aij, for all pairs i, j ∈ Q. Due to the local nature of backrubs (Section 1.2), BIC only requires that no two residues that are adjacent in the protein sequence will be simultaneously allowed to perform backrub motions. When BIC holds, the set Y of possible backbone conformations can simply be defined as Y = ∏i∈Q Yi, where F is the fixed part of the protein template. Further, when BIC holds, the set of possible conformations for rotamer ir will only depend on Ȳi (and not on Ȳj, for all j ̸= i) ∈ Q), i.e. X(ir) = Yir × Z(i). Finally, when Equation (5) holds, rotamer ir is provably not part of the GMEC when all possible backrubs in Ȳi for all k ∈ Q are considered.

2.2 Automated backrubs

The backrub motion is well defined (Section 1.2). Currently, however, the magnitude of the backrub primary and flanking rotations must be determined manually, through visual inspection, using the Backrub tool (Davis et al., 2006). However, the manual application of backrub motions for each mutation/rotameric conformation in a protein design combinatorial search is infeasible. Automating the computation of backrub motions is therefore essential if these types of motions are to be used as part of a protein design algorithm. Here, we present the following straightforward computational procedure for fully automated backrubs.

(1) Given a residue position i to be backrubbed and a primary rotation angle, determine the magnitude of the two flanking rotations. We use a simple geometric approach to determine the magnitude of the two flanking rotations for a given primary rotation. We must compute the flanking rotations 0_1 and 0_3 (Section 1.2). Let p(O_{i−1}) and p(O_{i+1}) be the positions of the backbone O of residue i − 1 before and after the primary rotation, respectively. Let f_{i−1}(x) be the position of point p after a flanking rotation of α degrees for peptide i − 1. Then, let δ_{i} = argmin_{f_{i−1}(x)}|f_{i−1}(x) − p(O_{i−1})|. This is the flanking rotation angle that moves the backbone O of peptide i − 1 to the point closest to its original position. We can then compute 0_1 = δ_{i} ∈ ε, where 0 ≤ ε ≤ 1 is a scaling factor used to limit the distortion in the respective τ angles (Section 3). The rotation angle 0_3 is computed analogously.

(2) Given a set Q of residues for which backrubs will be applied and a set U_i of primary (together with the corresponding flanking) rotations for each residue i ∈ Q, find the optimal backrub combination j ∈ U_i for each i. Here, a backrub combination (j_1, . . . , j_n) is a particular assignment of backrub rotation angles for each of the n flexible residue positions. Backrub combinations are generated from the Cartesian product of the sets U_i. A steric filter is applied during the backrub enumeration, in order to prune a combinatorial number of backrub combinations from further consideration (see Section 4). Backrub combinations that pass the steric filter are evaluated and ranked using our energy function (Section 4). This step guarantees the identification of the optimal backrub combination, given the input parameters and energy function.

Hence, using as input only: (1) a set of residues for which backrubs will be applied and (2) a set of primary rotation angles, backrub conformations can be generated using the automated procedure described in this section. We therefore now have the necessary tools to use backrubs in protein design.

3 ALGORITHM

We now present our novel protein design algorithm, incorporating the automated backrub procedure described in Section 2.2 and Brdee (Section 2.1). The algorithm consists of four main steps:

(1) Backrub set generation. In this first step, the sets Ȳi of allowed backrubs at each residue position i are generated using step 1 of Section 2.2. The input for this step is the set Q of residue positions that are modeled as flexible using backrubs and rotamers, and a set of allowed primary rotation angles. A steric filter (Section 4) is applied to prune clashing backrub/residue position combinations. Since a backrub at residue i introduces small changes into the τ angles (N-Cα-C') for residues i − 1, i, and i + 1 (Davis et al., 2006), a τ-angle filter (Section 4) further prunes backrubs that introduce large distortions in the τ angles of the affected residue positions.

(2) Pairwise lower and upper energy bounds precomputation. Using the sets Ȳi computed in step 1 above, compute the E_L() and E_U() terms (Section 2.1). Details of the method for computing the lower and upper energy bounds can be found in Section 4.

(3) Brdee pruning. The precomputed E_L() and E_U() terms are applied to evaluate Equation (5). Analogously to the extensions for traditional DEE and BD (Georgiev and Donald, 2007), we have also derived four extensions to Brdee for improved pruning: the simple, general, and pairs Goldstein (1994), and the conformational splitting (Pierce et al., 2000) criteria. These extensions are used in combination with the initial Brdee criterion Equation (5) and the DACS algorithm (Georgiev et al., 2006) in repeated rotamer pruning cycles until no further pruning can be achieved. This pruning step aims at significantly reducing the number of unpruned rotamers that must be considered in the subsequent enumeration stage.

(4) Enumeration and minimization. In the final step of the algorithm, using the E_L() and E_U() terms, a version of A* search enumerates rotamer vectors (an assignment of a particular rotamer identity for each flexible residue position) in order of increasing lower bounds (Section 4) on their energy. For each of the generated rotamer vectors, backrub minimization is then performed by applying the automated procedure from Step 2 of Section 2.2 to find the respective lowest energy backrub combination. A steric filter is applied to prune a combinatorial number of backrub/rotamer combinations (Section 4). The enumeration is halted once the lower bound on the energy of the next rotamer vector generated by A* exceeds the best conformation energy found in the search. At that point, we are guaranteed (cf. Georgiev et al., 2008) to have obtained the GMEC for the given design problem and input model.

4 METHODS

Two different sets of experiments were performed to validate our algorithms: recovery of alternate conformations from atomic-resolution crystal structures
The same set of five primary backrub angles was allowed for each of the five residues: A236, W239, T276, I299, A301, A322, E380. The allowed amino acid types at each of these positions were GAVLYFWM, as well as the wildtype identity. The Penultimate rotamer library modal values (Lovell et al., 2000) were used. The ligand was also modeled using rotamers and was further allowed to rotate/translate. Five primary backrub rotation angles were allowed for each of the flexible residues: −8, −4, 0, 4, and 8 degrees, for a total of five backrub angles per flexible residue. A 2-point mutation search (in a β-point mutation search, any 4 flexible residues are allowed to mutate simultaneously) was performed to switch the GrsA-PheX specificity towards a non-cognate substrate, Leu.

The structural model for Gβ1 is as described in Georgiev and Donald (2007). The same set of five primary backrub angles was allowed for each of the 12 residues (3, 5, 7, 9, 20, 26, 30, 34, 39, 41, 52, 54) in the core of Gβ1 that were modeled as flexible using backrub and rotamers. For the alternate conformation experiments, four atomic-resolution structure databases were used: deamidated bovine pancreatic ribonuclease (PDB id: 1brf; Esposito et al., 2003); Micrococcus lysodeikticus catalase (1gpe; Murshudov et al., 2002), xylanose isomerase (1mow; Finn et al., 2004) and extended-spectrum SHV-2 β-lactamase (1n9b; Nakaga et al., 2003). Hetero atoms and water were not included. The allowed primary rotation angles were from −10° to 10° at steps of 1°, for a total of 21 backrubs per flexible residue. The energy function consists of the Amber electrostatic and vdw terms (Cornell et al., 1995, Weiner et al., 1984) and the Ew pairwise solvent energy term (Lazaridis and Karplus, 1999). The following parameters were used: a distance-dependent dielectric of 6.0, a solvation-energy scaling factor of 0.05 and a vdw radii scaling factor of 0.9. A lower bound on the energy of a conformation (used by the A* enumeration) is computed as a sum of lower bounds on pairwise interactions (Georgiev et al., 2006). The lower bound \( E_{\text{lower}}(\phi, \psi) \) for a given rotamer pair is computed as the minimum energy between \( \phi \) and \( \psi \) of a sterically allowed conformation, over the set of backbone combinations for residues \( i \) and \( j \). Similarly, the upper bound \( E_{\text{upper}}(\phi, \psi) \) is computed as the maximum energy of a sterically allowed conformation for the given set of backrubs. The energies involving the template are computed analogously.

All of traditional DEE, BD and Btree are GMEC-based algorithms: typically, the goal is to identify only the single lowest-energy conformation. These algorithms (e.g. Equation 5) can be modified to guarantee the identification of all sequences/conformations within \( E_c \) from the respective GMEC energy (Georgiev et al., 2008). In our alternate conformation experiments, \( E_c \geq 20 \text{ kcal/mol} \), except for 1brf, where \( E_c = 100 \text{ kcal/mol} \). In all redesign experiments, \( E_c = 5 \text{ kcal/mol} \) for Btree and traditional DEE; BD used a cutoff of 5 kcal/mol relative to the best Btree conformational energy for the given redesign. Hence, the BD and traditional DEE running times are longer than (Georgiev and Donald, 2007), where \( E_c = 0 \).

For the BD experiments, the restraining boxes around each residue in the protein were defined using the following two bounding criteria: (1) a maximum \( C_{\alpha} \) displacement of 1.5 Å from the original PDB coordinates and (2) a maximum change of ±3° from the initial values in the PDB structure for the \( \phi, \psi \) angles of flexible residues (Georgiev and Donald, 2007). A value of 0.7 was used for \( \epsilon \) (Section 2.2), to limit distortion in the \( \tau \) angles. In the Backrub set generation step (Section 3), the \( \tau \)-angle filter prunes backrubs causing large distortions in the \( \tau \) angles. For a backrub at residue \( i \), the \( \tau \) angles at residues \( i-1 \) and \( i+1 \) are checked. If all of these angles are within ±3 degrees from an ideal value \( \lambda \), the backrub is allowed; otherwise, if the post-backrub \( \tau \) angles are closer to \( \lambda \) than the initial \( \tau \) angles, the backrub is allowed; otherwise, the backrub for the given residue position is pruned. In all of the described experiments, \( \lambda = 111.0 \) and \( \delta = 5.5 \).

All conformations for which at least one pair of atoms has a steric overlap of more than 0.4 Å are pruned. For the Backrub set generation step (Section 3), the steric filter reduces the set of backrubs allowed at each flexible residue position; here, steric checks are performed only against the fixed part of the molecule. For the Pairwise lower and upper energy bounds precomputation step (Section 3), the steric filter prunes backrub combinations for the given rotamers; here, steric checks also include the side-chains of the given rotamers. For the Enumeration and Minimization step (Section 3), the steric filter prunes entire subtrees of the conformation search tree (cf. Lilien et al., 2005). The alternate conformation experiments used \( \eta = 0.5 \). For the protein redesign experiments, hydrogens were not used in steric checks; hence, a stricter cutoff of \( \eta = 0.4 \) was used. Finally, for the Pairwise lower and upper energy bounds precomputation, initial rotamer conformations (before backrub application) with a steric overlap of more than 1.75 Å (alternate conformation experiments) and 1.5 Å (redesign) were pruned. Rotamers with a self-energy lower bound and rotamer pairs with a pairwise energy lower bound of more than 30 kcal/mol were also pruned.

5 RESULTS AND DISCUSSION

5.1 Alternate conformation recovery

A first-level test of \( \text{Btree} \) was whether it could reproduce well-characterized backrubs in high-resolution (≤0.9 Å) crystal structures (Table 1). The residues chosen (Table 1) are at least partially buried, represent both hydrophobic and polar amino acids, and have two alternate conformations related by a backrub (Davis et al., 2006), so they serve as excellent starting points for this analysis. Here, we define side-chain plus backbone conformations that are not present in either of the two alternate conformations as decoys. The initial backbone conformation and \( C_{\beta} \) position was used as a starting point for each alternate; ideal geometry rotamers were subsequently introduced by Btree. If the side-chain rotamers and backbone conformations represented by the alternates A and B scored better (i.e. had lower energy) than other decoy conformations, we could be confident that our automated backrub procedure produces physically reasonable conformations. The results were as follows.

5.1.1 1muw

In the deposited model, the alternate side-chain and backbone conformations for Val168 have a relatively large \( C_{\beta} \) displacement (0.66 Å). This residue is found on the buried side of a helix with minimal exposure to solvent. The swap between \( m \) and \( t \) rotamers (Lovell et al., 2000) is enabled by a backrub in a manner that is commonly observed for valines (Fig. 2). When starting from A, Btree finds the lowest conformational energy to be the A-like rotamer (that is, the rotamer whose side-chain dihedrals are on average closest to those of the deposited alternate A) with a backrub in the A direction and the second lowest energy (about 9 kcal/mol worse) to be the B-like rotamer with a backrub in the B direction. When starting from B, the order of returned conformations is reversed, but the calculated energy difference (about 0.3 kcal/mol) is negligible, suggesting that the initial \( C_{\beta} \) position biases the comparison between these putatively equivalent alternates but that Btree approximates the correct relationship either way. From either starting conformation, the third possible rotamer (\( p \), definitively not observed in the experimental density, is sterically allowed, but it scores 5–13 kcal/mol worse than its closest competitor.

5.1.2 1gpe

Asp163 is a helix N-cap (Richardson and Richardson, 1988) that alternates between the two common hydrogen bonds at such a position, satisfying either the NH of residue \( i+2 \) (A) or \( i+3 \) (B). Although the structure was deposited with a single backbone and the \( C_{\beta} \) displacement between alternate side-chains is relatively low (0.2 Å), close examination reveals...
Ala59 ends a three-residue alternate conformation and is deposited in Table 1.

Table 1. Alternate conformation results

| PDB  | Res  | Starting from A’ | Starting from B’ |
|------|------|----------------|-----------------|
| 1muw | V168 | A-like 0 | B-like +1 | E’ |
|      |      | -188.6 | 186.0 | 1.3 |
|      |      | +10  | -179.4 | A-like -5 | -185.8 |
|      |      | -9  | -174.5 | decoy 0 | -172.9 |
| 1gwe | D163 | A-like 0 | B-like -1 | E’ |
|      |      | -280.1 | 280.1 | 1.3 |
|      |      | +4  | -270.8 | B-like +3 | -268.8 |
| 1nf6 | I47  | A-like 0 | B-like 0 | E’ |
|      |      | -254.1 | 226.8 | 1.3 |
|      |      | +10  | -251.4 | B-like -10 | -222.6 |
| 1dy5 | M629 | B-like +1 | B-like +3 | E’ |
|      |      | -256.4 | 254.6 | 1.3 |
|      |      | -2  | -250.4 | A-like +1 | -253.4 |
|      |      | -2  | -253.1 | decoy -1 | -252.8 |
|      |      | -3  | -253.1 | decoy -1 | -252.4 |
|      |      | +10  | -240.0 | |

The PDB id for each structure is shown along with the residue for which alternate conformation recovery was performed. A and B refer to the alternate conformations labeled as A and B in the PDB files. The conformation predicted by the algorithm is similar to A (A-like), to B (B-like), or to neither (a decoy). E’ is the primary backrub angle (in degrees); 1.3 the computed energy (in kcal/mol).

that the electron density can be equally well satisfied when this displacement is modeled by a backrub; this assertion is supported by observations in the Richardson lab that backrubs may play an important role at N-caps (data not shown). Both alternates are very close to modal rotamers in dihedral space, so Bdee is able to recapitulate them very accurately (when starting from either A or B) with approximately the same 10 kcal/mol energy difference between A-like and B-like conformations. No decoys were identified by Bdee.

5.1.3 1n9b. Ile47 is in an antiparallel β-sheet. The adjacent Ala59 ends a three-residue alternate conformation and is deposited with separate backbones, and although Ile47 is not, it should be: conformation A is non-ideal with a 0.38 Å Cα deviation (Lovel et al., 2003) and 0.59 Å between alternate Cα’s. It has been shown that a backrub better relates the two alternate rotamers, while allowing near-ideal geometry (Davis et al., 2006). To demonstrate this in the context of our new algorithm, we first used the original, distorted backbone and Cα conformations. When starting from the A form on both strands, the only conformation returned reproduces the A-like rotamer and backrub of Ile47. When starting from both B forms, however, the A-like rotamer and backrub at Ile47 is returned first, followed by a decoy conformation and then the B-like rotamer and backrub. Notably, all three of these B-derived conformations scored at least 40 kcal/mol worse than the A-derived one, indicating distortion in the original B alternate backbone and Cα. To improve the geometry of the model, we split Ile47’s backrub with a manual backrub and idealized the Cα’s of both its alternates as a pre-processing step. As a result, the only conformations returned from either starting point (A or B on both strands) represent the original A- and B-like rotamers and backrubs, in that order. Moreover, the energy differences between A-like and B-like conformations at Ile47 are only about 4 kcal/mol or less, indicating the success of our backrub modeling.

Occupancies of the A and B conformations are nearly equal for both Ile47 and Ala59, and therefore it seems possible that they were paired incorrectly in the deposited coordinates. To investigate this question, we tried starting from the A conformation at 47 and the B conformation at 57–59 and vice versa, to determine if a better strand-strand coupling might be obtained by Bdee. In all cases, only the same A-like and B-like Ile47 conformations are returned, but the A conformation for 57–59 is preferred by about 30 kcal/mol, regardless of the returned conformation for Ile47. This indicates that from a purely energetic perspective, there is no strand-strand coupling and the deposited alternate conformation designations are satisfactory.

To ensure that a B-like conformation was returned in the tests just described, it was necessary to augment the rotamer library with the deposited B side-chain’s dihedrals. Before splitting the backbone and adding this side-chain to the library, Bdee scored the B-like model as much worse than the A-like model when starting from B and forwent it entirely when starting from A. This can likely be attributed to local packing constraints—the deposited B side-chain has a Cα 20° away from the model rotamer, which therefore does not match well. More generally, this indicates that denser coverage of side-chain and backbone conformational space can be combined to achieve more realistic modeling.

5.1.4 1dy5. Met b29 (from chain b) was chosen because its alternate side-chains are better rotamers, have more closely matched occupancies, and fit the experimental electron density more clearly than those of Met a29 (from chain a). This residue lies on a helix approximately half-exposed to solvent. It is also deposited with a single backbone, but the displacements of the side-chain atoms as seen in the electron density and that density’s anisotropy perpendicular to the chain’s direction reveal that a backrub better models this residue. Starting from either A or B, a B-like rotamer and backrub are ranked highest by Bdee, followed by an A-like rotamer and backrub, and two decoys differing from alternate A only at Cα. Starting from A, a third, substantially worse (13 kcal/mol greater than any other conformation) decoy is also returned. This decoy...
is therefore clearly energetically discriminated from more realistic conformations, but it was identified by the algorithm, since the backrub allowed the side-chain to displace itself laterally to mostly escape a steric clash on one side. The first two decoys most likely would have been pruned had the nearby acetate from the PDB structure been included in the structural model. The reason the B-like conformation scores slightly better than the A-like is likely due to the fact that the deposited alternate A has a modest steric clash with Tyr B25 of the preceding helical turn. This explanation is further supported by the observation that Ser B32 on the following turn of the helix from Met B29 is modeled with alternate side-chains, indicating the presence of helical turn-to-turn interactions.

Interestingly, all generated conformations from each starting point (except for the third decay from A) were within 2.2 kcal/mol of each other, implying that despite dealing with small energetic differences, BRede correctly discriminated physically realistic conformations from decoys.

After proper remodeling of 1nqb’s backbone, A- and B-like conformations in terms of rotamer and backbone direction and approximate recovery were recovered in every case, whether starting from the A or B backbone and C<sub>β</sub>. Moreover, if generated at all, decoys always scored worse than the crystallographically observed conformations. This brings up an interesting point that arose from energy minimization of the side-chain dihedrals for 1nqw Val66’s BRede conformations starting from A. After energy minimization, the difference between the real A-like and B-like conformations generated by BRede decreased by over 5 kcal/mol whereas the difference between the best scoring conformation (A-like) and the decoy changed by <1 kcal/mol. If generally applicable, this simple test reveals the potential improvement in accuracy/decoy discrimination that BRede might gain by further exploring conformational space around rotamers, e.g. with an extended rotamer library or continuous side-chains (Georgiev et al., 2008).

5.2 Protein redesign

5.2.1 Redesign of GrsA-PheA for Leu. For comparison, each of traditional DEE, BRede, and BD was applied in separate redesigns of GrsA-PheA (Table 2). As Table 2 shows, the lowest conformation energy identified by BRede is >1 kcal/mol lower than the lowest conformation energy identified by traditional DEE. Moreover, traditional DEE identified only 88 rotameric conformations (each conformation represents a unique rotamer vector), representing 7 unique sequences, with energies within 5 kcal/mol from the lowest BRede conformation energy. In contrast, BRede identified more than 600 conformations, representing 39 unique sequences, with energies within 5 kcal/mol from the lowest BD conformation energy. This confirms that BRede is capable of generating a significantly larger number of low-energy sequences and conformations, as compared to traditional DEE. It is interesting to note that the set of low-energy conformations/sequences identified by BD (Table 2) is much larger than BRede. This finding can be explained by the fact that backbone flexibility in BD is represented by global motions, whereas backrubs represent much smaller scale local motions. Moreover, BD is defined over a continuous family of solutions, whereas BRede is defined over a discrete backbone set. Hence, since BD and BRede represent very different (and complementary) types of backbone motions, interesting future work would involve the derivation of an algorithm for protein design that simultaneously allows both types of backbone flexibility.

Furthermore, as Figure 3 shows, the distribution and frequencies of the Leu-binding mutations for each of the seven GrsA-PheA (Table 2). As Table 2 shows, the lowest conformation identified at all flexible positions, BRede identified 3 (for 239) and 6 (for 301) amino acid types. Thus, as expected, the incorporation of backrubs into the protein design algorithm leads to the identification of lower energy sequences and conformations that would have been ignored by a rigid-backbone model.

As an adjunct to comparisons of sequence diversity, it can also be insightful to examine the conformational diversity. One example that illustrates the complementary capabilities of the two algorithms with backbone flexibility is the sequence predicted most often by BRede, 236L/239L (represented by 74 low-energy conformations in BRede and 68 in BD). The rotamers identified at all flexible positions are similar, overall between BD and BRede, but BRede allows an additional rotamer for Ile299. A closer examination of the structures generated by both algorithms reveals why this is the case. Ile299 is situated on a β-strand, and the new rotamer (t) has a different χ<sub>1</sub> from the other rotamers (mm, mp, mt) (Lovell et al., 2000) identified by the algorithms. This would normally cause the C<sub>α</sub> atom to clash with Thr278 on an adjacent β-strand. However, BRede enables these two strands to move away from each other locally, whereas in the BD conformations, the strands were observed to move together in the same direction, as part of a coordinated shift of the entire active site’s backbone. This implies that BRede can facilitate small-scale, anti-correlated backbone motions, whereas the strength of BD is large-scale, correlated backbone motions.

To quantify the effect of incorporating backrubs on the conformational energies, we further analyzed the set of 605 low-energy conformations (Table 2) returned by BRede. For this analysis only, the ligand was removed from the complex, since its rotation/translation could also impact the conformational energies. The computed energy decrease resulting from backrub minimization (Step 4 from Section 3) varied significantly between conformations, ranging from no effect to an improvement of almost 80 kcal/mol (data not shown). However, in 462 of the 605 conformations (76%),

| Table 2. DEE comparison for GrsA-PheA and Gj1 redesign |
|---------------------------------------------------------|
| Redesign       | DEE<sup>a</sup> | Best energy<sup>b</sup> | Sequences<sup>c</sup> | Conf<sup>d</sup> |
|----------------|-----------------|--------------------------|----------------------|-----------------|
| GrsA-PheA      |                 |                          |                      |                 |
| traditional DEE| −241.41         | 7                        | 88                   |                 |
| BRede          | −242.58         | 39                       | 605                  |                 |
| BD             | −251.19         | 422                      | 6805                 |                 |
| Gj1            |                 |                          |                      |                 |
| traditional DEE| −371.19         | 67                       | 169                  |                 |
| BRede          | −372.86         | 164                      | 599                  |                 |
| BD             | −375.13         | >950                     | >3500                |                 |

The lowest conformation energy (in kcal/mol) identified by each of the three DEE algorithms (traditional DEE, BRede, and BD), which is shown along with the number of sequences and number of conformations with energy lower than −237.58 kcal/mol (GrsA-PheA) and −367.86 kcal/mol (Gj1) that all sequences and conformations within 5 kcal/mol from the corresponding lowest BRede conformation energy are included. The BD experiments for Gj1 were halted after not completing in more than 4 weeks.
DEE separately for backbone. However, as a simple comparison, performing traditional Brdee guarantees significantly reduce the computational efficiency of incorporating backbone flexibility and the provable algorithmic backbone) completed in $\sim 2\text{ weeks}$ on a cluster of 16 processors, as compared to $\sim 3\text{ h}$ for traditional DEE for a single rigid backbone. However, without the pruning filters of our backrub algorithm, a mutation search over backbone/rotamer space would be computationally infeasible.

To determine whether Brdee introduces unreasonable bond strain, we compared the $\tau$-angle distributions for the residues affected by backrubs in the top 605 (GrsA-PheA) and 599 (G$\beta$1) Brdee-generated structures against all residues in the respective crystal structures (data not shown). In addition, all $\tau$-values are very close to the ideal value of 111°. Therefore, we can conclude that Brdee introduces only small $\tau$ changes that do not cause significant strain.

5.2.2 Redesign of G$\beta$1. The results from the G$\beta$1 redesign (Table 2) show a similar trend to the GrsA-PheA results. Brdee identified a significantly larger number of low-energy sequences and conformations than traditional DEE; similarly, BD identified a larger number of low-energy sequences and conformations than Brdee. The application of the steric and $\tau$ filters (Step 1 of Section 3) reduced the possible backbone combinations by a factor of more than 3000. The Brdee pruning stage reduced the number of possible rotameric conformations by a factor of more than $10^7$. Including the pairwise energy precomputation, the Brdee redesign required 2 weeks on a cluster of 16 processors, as compared to $\sim 3\text{ h}$ for traditional DEE for a single rigid backbone. However, without the pruning filters of our backrub algorithm, a mutation search over backbone/rotamer space would be computationally infeasible.

6 CONCLUSION

In this article, we presented an algorithm for protein design that incorporates local backbone flexibility via backrub motions. As confirmed by the redesigns of the GrsA-PheA active site and the core of G$\beta$1, the additional flexibility provided by the backrub algorithm allows the identification of a large number of low-energy sequences and conformations that would have otherwise been ignored by a rigid-backbone model. Such an expansion in the predicted sequence/conformation space for rational protein design likely implies that backrubs may play an important evolutionary role as well; further computational and experimental validation will be necessary to confirm this hypothesis.

Brdee only requires that the three most expensive steps of the design algorithm be performed once, simultaneously for all backbones (Section 1.3, paragraph 3). Preliminary benchmark tests (Section 5.2) indicate that Brdee obtains significant computational benefits, as compared to applying traditional DEE separately for each backbone. However, further experiments on more proteins will be necessary to determine the precise benefits of Brdee.

Backrubs represent local motions, whereas in the BD algorithm, backbone flexibility is represented by global motions. Brdee and BD are thus complementary in nature, so combining these two backbone flexibility approaches within a single protein design algorithm presents interesting future work. The main challenge for such an algorithm will be to overcome the combinatorial explosion resulting from the consideration of all possible backbone conformations. Preliminary evidence from our alternate conformation recovery

---

**Figure 3. Distribution of Mutations: GrsA-PheA redesign for the non-cognate substrate Leu.** The distribution of mutations for all conformations with an energy within 5 kcal/mol from the lowest Brdee energy (Table 2) is shown for traditional DEE (top), Brgee (middle), and BD (bottom). Allowing backrubs resulted in a decrease of the conformational energy, and in 261 of these conformations, the decrease in energy was $>1\text{ kcal/mol}$, thus confirming the potential benefit of including backrubs in protein design.

For the GrsA-PheA redesign, the application of the steric and $\tau$ filters from Step 1 of Section 3 reduced the possible backrub combinations by a factor of 20, to a total of 3888. The Brdee pruning stage reduced the number of possible rotameric conformations from $1.99 \times 10^{12}$ to $4.78 \times 10^7$. Including the pairwise energy precomputation, the entire redesign took almost a day on a cluster of 10 processors. In contrast, traditional DEE (for a single rigid backbone) completed in $\sim 45\text{ min}$ on 10 processors. Thus, the incorporation of backbone flexibility and the provable algorithmic guarantees significantly reduce the computational efficiency of Brdee when compared to traditional DEE for a single rigid backbone. However, as a simple comparison, performing traditional DEE separately for each of the 3888 backbones (assuming similar CPU times for each backbone), would require $\sim 120\text{ days}$ on the same number of processors. This implies Brdee is approximately two orders of magnitude faster, although further benchmarks would be necessary to determine its precise computational benefits.
Georgiev, I. et al. (2007) Dead-end elimination with backbone flexibility. Fung, H.K. et al. (2007) Novel formulations for the sequence selection problem in de novo protein design with flexible templates. Optim Method Softw. 22, 51–71. Georgiev, I. and Donald, B.R. (2007) Dead-end elimination with backbone flexibility. Bioinformatics, 23, 1855–1854. Georgiev, I. et al. (2006) Improved pruning algorithms and divide-and-conquer strategies for dead-end elimination, with application to protein design. Bioinformatics, 22, e174–e183. In Proceedings of the International Conference on Intelligent Systems for Molecular Biology (ISMB), Fortaleza, Brazil. Georgiev, I. et al. (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., Feb 21 [Epub ahead of print]. PMID: 18292924. Goldstein, R. (1994) Efficient rotamer elimination applied to protein side-chains and related spin glasses. Biophys. J., 66, 1335–1340. Gordon, D.B. et al. (2003) Exact rotamer optimization for protein design. J. Comput. Chem., 24, 232–243. Gordon, D.B. et al. (1999) Energy functions for protein design. Curr. Opin. Struct. Biol., 9, 505–513. Harbury, P.B. et al. (1998) High-resolution protein design with backbone freedom. Science, 282, 1462–1467. Havranek, J.J. and Harbury, P.B. (2003) Automated design of specificity in molecular recognition. Nat. Struct. Biol., 10, 45–52. Hu, X. and Kuhlman, B. (2006) Protein design simulations suggest that side-chain conformational entropy is not a strong determinant of amino acid environmental preferences. Proteins, 62, 739–748. Jiang, L. et al. (2006) De Novo computational design of retro-aldol enzymes. Science, 319, 1387–1391. Jin, W. et al. (2003) De novo design of foldable proteins with smooth folding funnel. Automated negative design and experimental verification. Structure, 11, 581–591. Korkiean, A. et al. (2005) Computational thermostabilisation of an enzyme. Science, 308, 857–860. Kortemme, T. et al. (2004) Computational redesign of protein-protein interaction specificity. Nat. Struct. Mol. Biol., 11, 371–379. Kuhlman, B. and Baker, D. (2000) Native protein sequences are close to optimal for their structures. Proc. Natl Acad. Sci., 97, 10383–10388. Kuhlman, B. et al. (2003) Design of a novel globular protein fold with atomic-level accuracy. Science, 302, 1364–1368. Lassila, J.K. et al. (2006) Combinatorial methods for small-molecule placement in computational enzyme design. Proc. Natl Acad. Sci. USA, 103, 16710–16715. Laster,J. and Doonan,J. (1993) The fancy-end elimination theorem: correctly implementing the side chain placement algorithm based on the dead-end elimination theorem. Proteins, 7, 717–722. Lazaridis, T. and Kapushesko, M. (1999) Effective energy function for proteins in solution. Proc. Struct. Funct. Genet., 38, 133–152. Leach, A. and Lemon, A. (1998) Exploring the conformational space of protein side chains using dead-end elimination and the A* algorithm. Proteins. 33, 227–239. Lilien, B. et al. (2005) A novel ensemble-based scoring and search algorithm for protein redesign, and its application to modifying the substrate specificity of the Gramicidin Synthetase A pyrophosphate adenylylation enzyme. J. Comput. Biol., 12, 740–761. Lippow, S.M. et al. (2007) Computational design of antibody-affinity improvement beyond in vivo maturation. Nat. Biotechnol., 25, 1371–1376. Looger, L. and Hellinga, H. (2001) Generalized dead-end elimination algorithms make large-scale protein side-chain structure prediction tractable: implications for protein design and structural genomics. J. Mol. Biol., 307, 429–445. Looger, L. et al. (2003) Computational design of receptor and sensor proteins with novel functions. Nature, 423, 185–190. Lovell, S.C. et al. (2000) The penultimate rotamer library. Proteins, 40, 389–408. Lovell, S.C. et al. (2005) Structure validation by Cu geometry: ϕ, ψ and Cζ deviation. Proteins, 59, 437–440. Malakauskas, S.M. and Mayo, S.L. (1998) Design, structure and stability of a hyperthermophilic protein variant. Nat. Struct. Biol., 5, 470–475. Mendes, J. et al. (1999) Improved modeling of side-chains in proteins with rotamer-based methods: a flexible rotamer model. Proteins, 37, 530–543. Marshman, G. et al. (2002) The structures of Micrococcus lysodeikticus catalase, its ferryl intermediate (compound II) and NADPH complex. Acta Crystallogr. D. Biol. Crystallogr., 68 Pt 12, 1972–1982. Nakagami, M. et al. (2003) Ultra-high resolution structure of a class A beta-lactamase: on the mechanism and specificity of the extended-spectrum SHV-2 enzyme. J. Mol. Biol., 329, 289–301. Pierce N. and Winzurz, E. (2002) Protein design is NP-hard. Prot. Eng., 15, 779–782. Pierce, N. et al. (2000) Conformational splitting: a more powerful criterion for dead-end elimination. J. Comput. Chem., 21, 999–1009. Ponder, J. and Richards, F. (1987) Tertiary templates for proteins: use of packing criteria in the enumeration of allowed sequences for different structural classes. J. Mol. Biol., 193, 775–791. Richardson, J.S. and Richardson, D.C. (1988) Amino acid preferences for specific locations at the ends of alpha helices. Science, 240, 1648–1652. Stevens, B. et al. (2006) Redesigning the Phe/Dom domain of Gramicidin Synthetase lead to a new understanding of the enzyme’s mechanism and selectivity. Biochemistry, 45, 15495–15504. Street, A. and Mayo, S. (1999) Computational protein design. Structure, 7, R105–R109. Sua. A. and Mayo, S. L. (1997) Coupling backbone flexibility and amino acid sequence selection in protein design. Prot. Sci., 6, 1701–1707. Viscardi, C. et al. and Mayo, S.L. (2005) Electrostatics in computational protein design. Curr. Opin. Chem. Biol., 9, 622–626. Vogt, C.A. et al. (2000) Trading accuracy for speed: a quantitative comparison of search algorithms in protein sequence design. J. Mol. Biol., 299, 789–803. Vogt, C.A. et al. (2001) Computational method to reduce the search space for directed protein evolution. Proc. Natl Acad. Sci. USA, 98, 3778–3783. Weiner, S. et al. (1994) A new force field for molecular mechanical simulation of nucleic acids and proteins. J. Am. Chem. Soc., 106, 765–768. Yanover, C. et al. (2007) Dead-end elimination for multi-site protein design. J. Comput. Chem., 28, 2122–2129. Zanghellini, E. et al. (2006) New algorithms and an in silico benchmark for computational enzyme design. Prot. Sci., 15, 2785–2848.