BIO DESIGN

pISSN 2288-6982 | eISSN 2288-7105

MINI REVIEW P 26-37

Side-chain Prediction and Computational Protein Design Problems

Deok-Soo Kim[1,2,*] and Joonghyun Ryu[2]

[1]Department of Mechanical Engineering, [2]Voronoi Diagram Research Center, Hanyang University, 17 Haengdang-dong, Seongdong-gu, Seoul 133-791, Korea. *Correspondence: dskim@hanyang.ac.kr

Protein is fundamental for life and thus protein design has been one of the hottest research issues since a couple of decades ago and computational approach has been developed with some significant design cases with experimental verification. One of the fundamental building blocks of computational protein design is to predict side-chain conformation of backbone structure. This paper first reviews prior efforts of computationally predicting sidechains of a given backbone and then reviews computational protein design efforts based on side-chain prediction. The paper views computational protein design problem as three categories: Redesign, De novo design of Type I, and De novo design of Type II. Some well-known computer programs and algorithms related with computational protein design are also reviewed.

INTRODUCTION

Life functions via protein structure where its molecular shape or geometry plays one of the most fundamental roles. Being a linear sequence of amino acids of twenty types, the primary determinant of molecular shape is the type of each amino acid because the interactions between side-chains and between side-chains and backbone determine the conformation.

The prediction of protein structure from an amino acid sequence is known as *protein folding problem* which is one of the most fundamental, attracted, and challenging research issues in computational biology. Folding problem is known NP-hard in computational term which practically means that the problem is extremely hard, infeasible in principle, to computationally find the optimal solution for a protein of moderate size with contemporary mathematical and computational capability. Once folding problem is well-understood, it might be possible to understand protein function from the predicted structure of a given sequence. Consider an inverse problem: Given a target structure, predict the amino acid sequence that shall fold into the target structure. This problem is called *protein design problem* and is not easier, in fact much harder, than folding problem because it contains folding problem. Consider a third problem: Given a fixed backbone structure (i.e., the coordinate of atoms in the backbone is determined) of a fixed amino acid sequence, predict the optimal conformation of the side-chains of all amino acids. This problem is called *side-chain prediction problem*, SCP-problem in short. "P" sometimes translates to "positioning" or "placement" and the problem is also sometimes called the side-chain "modeling," "optimization," "selection," or "packing" problem. The optimality is defined by the minimum potential energy of the structure determined by the conformation of all side-chains where the energy is given as a function of empirical forces such as the van der Waals, electrostatic, hydrogen bonding forces, etc. It is known that the SCP-problem itself is already NP-hard. Folding problem, and thus protein design problem as well, is usually regarded to contain SCP-problem as a subproblem.

From computational theory point of view, the NP-hardness of a problem implies that there does not exist an efficient algorithm, a polynomial time algorithm in practice, for correctly solving the problem. Therefore, an ordinary approach to NP-hard problems is to devise heuristics in that an efficient search of solution space to find a good solution, instead of finding an optimal or correct solution, is of primary concern. Therefore, a heuristic approach is inevitable for both the SCP-problem and protein design problem. Most prior heuristic approaches to protein structure problems begin with the empirical observation that the energetics of protein structure can be explained by decomposing it with the interactions between side-chains and between side-chains and backbone. In other words, two-body interaction is a good approximation of the nature of this problem.

This paper reviews computational methods for solving the SCP-problem and protein design problem. We admit that this review is rather limited from biological point of view because our main research interest and background are mathematical and computational. The predictions produced by such computational methods should be accompanied by the verification and validation through wet lab experiments. Section 2 reviews rotamer library which is the basis for the SCP-problem and computational protein design problem. Section 3 reviews the SCP-problem and

Section 4 the computational protein design problem.

ROTAMER LIBRARY

Protein structure is determined by fixing dihedral angles at rotational bonds. While each dihedral angle can take any value from 0 to 360 degrees in theory, there is a range of dihedral angles that each rotational bond prefers in determined protein structures from statistical point of view. A backbone structure is determined by two types of dihedral angles, φ and ψ, at each alpha-carbon C_α and a side-chain structure is determined by dihedral angles denoted by χ. A side-chain may have zero to four dihedral angles. For example, alanine and glycine have no side-chain dihedral angles and thus their side-chains have no degree-of-freedom contributing to structure; Cystein has one side-chain dihedral angle; Asparagine has two dihedral angles; Glutamine has three dihedral angles; Arginine has four dihedral angles and thus its side-chain has a great contribution to protein structure.

A *rotamer* (rotational isomer) is a single conformation of side-chain and a *rotamer library* is a collection of rotamers for each residue type (Dunbrack Jr., 2002). A rotamer is usually defined as a combination of frequently observed dihedral angles. The concept of rotamer can be considered as an extension of the seminal work of Ramachandran and colleague in 1965 on the backbone conformation in that some regions of the map are usually avoided (Ramakrishnan and Ramachandran, 1965) and Chandrasekran and Ramachandran counted rotamers in the three protein structures available (Chandrasekaran and Ramachandran, 1970). Janin et al. showed that determined protein structures clustered certain χ-angles for many amino acids (Janin et al., 1978). James and Sielecki showed that the distributions of χ_1 and χ_2 angles were more focused than previously observed by analyzing five high resolution enzyme structures (James and Sielecki, 1983). In 1987, Ponder and Richards derived the first rotamer library, say PR87, consisting of 67 rotamers derived from 19 well-solved protein crystal structures (Ponder and Richards, 1987). Each amino acid is associated with a set of rotamers (except alanine and glycine because their side-chains do not have any degrees of freedom) and the set of the 18 sets of rotamers for the 18 types of amino acids is called the rotamer library. A rotamer library is usually derived by statistical analysis of the side-chain conformations observed in the solved protein structures in Protein Data Bank (PDB) (Dunbrack Jr., 2002; Dunbrack Jr. and Karplus, 1993, 1994; Kono, 2009).

Since the first one, PR87, a number of rotamer libraries were reported roughly categorized into three major groups (Dunbrack Jr., 2002; Park et al., 2004): backbone-independent, backbone-dependent, and secondary structure-dependent. A backbone independent library does not take account for backbone conformation in the library definition. PR87 belongs to this category. Dunbrack and Cohen reported one of the most popular backbone-independent one, DCindep97, consisting of 341 rotamers from 518 chains with ≤2.0Å resolution using the Bayesian statistical analysis (Dunbrack Jr. and Cohen, 1997) (later updated in 2002).

Figure 1 shows examples for rotamer instances of DCindep97. Figure 1(a) through (d) are the chemical formulae for cystein, asparagine, glutamine, and arginine, Figure 1(e) through (h) stick models with rotatable bonds, and Figure 1(i) through (l) the collection of rotamer instances, respectively.

A backbone-dependent library takes into account for the backbone structures in the library definition. McGregor et al. observed the possibility of correlation between rotamer preferences and secondary structures (McGregor et al., 1987) and Schrauber et al. observed large deviation (for both energetic and geometric aspects) from rotamer values of PR87 and attempted to extend PR87 based on 20 degree granularity of χ_1 and χ_2 angles (Schrauber et al., 1993). Dunbrack and Karplus

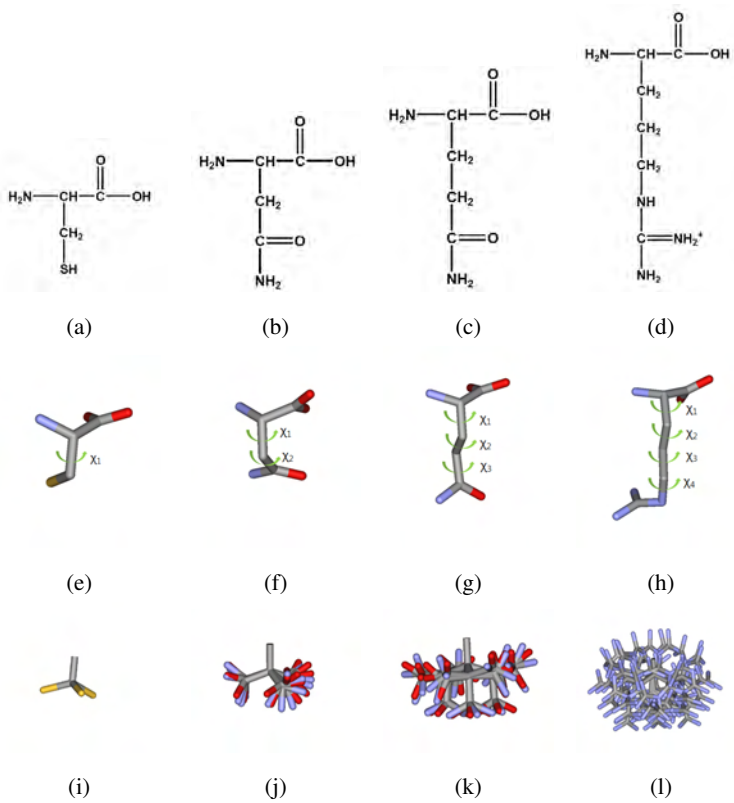

FIGURE 1 | Example of rotamer instances in DCindep97. Hydrogens are removed.: (a), (b), (c), and (d): the chemical formulae for cystein, asparagine, glutamine, and arginine; (e), (f), (g), and (h): the respective stick models with rotatable bonds; (i), (j), (k), and (l): the respective collection of rotamers.

observed a stronger correlation between rotamer preferences and backbone angles and first devised the most popular backbone-dependent library, DK93, from 132 chains in 126 structures with ≤2.0Å resolution (Dunbrack Jr. and Karplus, 1993). Backbone angles φ and ψ are divided into 20 degree by 20 degree blocks. Dunbrack and Cohen (1997) later extended DK93 to DCdep97, consisting of 466,829 rotamers, by the Bayesian analysis of the crystal structures of 850 chains with ≤1.7Å resolution with 10 degree by 10 degree granularity blocks. It seems that a backbone-dependent library can produce a solution of better quality and thus is more popular. Secondary structure dependent library reflects the secondary structures but seems less popular. Good reviews on rotamer library are available (Pupo and Moreno, 2009; Dunbrack Jr., 2002). Pupo and Moreno suggests that the one proposed by Xiang and Honig (Xiang and Honig, 2001) might be the best. Studies for better using rotamers are continuing (Harder et al., 2010; Scouras and Daggett, 2011; Shapovalov and Dunbrack Jr., 2011; Bhuyan and Gao, 2011; Alexander et al., 2013).

SCP-PROBLEM

Given a rotamer library, the SCP-problem can be regarded as a problem of optimally assigning a rotamer at each residue from a rotamer library on the fixed backbone so that the total energy E of a predicted structure is minimal among all possible combinations of rotamers at all residues. Backbone coordinates are not modified during the solution process. Let B and Σ denote the backbone and the side-chains, respectively. Then, E to be minimized consists of two terms as follows:

$$E = E_{B\Sigma} + E_{\Sigma\Sigma} \quad (1)$$

where $E_{B\Sigma}$ is the energy between B and the side-chain of each residue in Σ and $E_{\Sigma\Sigma}$ is the energy between the side-chain $\sigma_i \in \Sigma$ and another side-chain $\sigma_j \in \Sigma$, $i \neq j$ (Desmet et al., 1992). In theory, E consists of self-energy, the energy of pair-wise combinations, that of triplet-wise combinations, and so on, among backbone and all side-chains. Hence, in principle, the true SCP-problem is to solve a minimization problem of all possible *N*-body interactions (Hopfinger, 1973; Maranas and Floudas, 1994) which is inapproximable (Chazelle et al., 2004b) (i.e., it is unlikely that there exists a polynomial time algorithm that can guarantee a good approximated solution of the problem). Thus, this problem cannot be solved correctly with current computing technology even with an empirical force field (Petrella et al., 1998; Samudrala and Moult, 1998a; Xiang and Honig, 2001).

Fortunately, it turns out that modeling the SCP-problem as a collection of 2-body problems suffices from both practical and empirical point of view (Hopfinger, 1973). Consider the van der Waals interaction between non-bonded atoms is the only force in the system and is modeled by the (12-6) Lennard-Jones form (Holm and Sander, 1992; Eriksson et al., 2001; Kingsford, 2005; Kingsford et al., 2005). Then, the 2-body interaction formulation of Eq. (1) is given as

$$E_{B\Sigma} + E_{\Sigma\Sigma} = \sum_{a_i \in B} \sum_{a_j \in \sigma} \left\{ \frac{A_{ij}}{d_{ij}^{12}} - \frac{B_{ij}}{d_{ij}^{6}} \right\} + \sum_{a_i \in \sigma_i} \sum_{a_j \in \sigma_j} \left\{ \frac{A_{ij}}{d_{ij}^{12}} - \frac{B_{ij}}{d_{ij}^{6}} \right\} \quad (2)$$

where σ is a side-chain of a residues and d_{ij} is the Euclidean distance between the centers of a pair of atoms a_i and a_j . A_{ij} and B_{ij} are parameters of a force field such as AMBER (Assisted Model Building with Energy Refinement) (Cornell et al., 1995) or CHARMM (Chemistry at HARvard Macromolecular Mechanics) (Brooks et al., 1983) that depend on atom types. Once rotamers are assigned at all residues, d_{ij}'s are determined between each pair of atoms, one from a rotamer and the other from either another rotamer or backbone, and so is the corresponding energy. If it is necessary, other forces such as electrostatic force and hydrogen bonding energy can be similarly incorporated. Each of the backbone and side-chains has its own potential energy called self-energy which is regarded as constant and thus can be eliminated from formulation. The NP-hardness of the SCP-problem is proved by reducing the satisfiability problem (Pierce and Winfree, 2002) or the unconstrained quadratic 0-1 programming problem (Fung et al., 2005) to the decision problem of the SCP-problem. Without the use of rotamer library, the SCP-problem should become a nonlinear programming problem in a high-dimensional continuous space and is much harder than its already hard integer linear programming (ILP) counterpart of the problem. The SCP-problem has many important applications: homology modeling (Bower et al., 1997; Petrey et al., 2003; Xiang and Honig, 2001; Dunbrack Jr. and Karplus, 1993), protein folding (Xiang and Honig, 2001; Bower et al., 1997), NMR (Nuclear magnetic resonance) and X-ray structure refinement (Kuszewski et al., 1995), protein design (Dahiyat and Mayo, 1996), etc.

Mathematical programming approach

The view of Eq. (2) allows the SCP-problem to be formulated as a mathematical optimization problem known as an integer linear programming (ILP) problem as follows (Eriksson et al., 2001; Fung et al., 2005; Kingsford et al., 2005; Zhu, 2007).

$$\text{Min.} \sum_{i=1}^{n} \sum_{j=1}^{m_i} E_{B\sigma}(i,j) x_{ij} + \sum_{i=1}^{n-1} \sum_{j=1}^{m_i} \sum_{k=i+1}^{n} \sum_{l=1}^{m_k} E_{\sigma\sigma}(i,j,k,l) x_{ijkl} \quad (3)$$

Subject to

$$\sum_{j=1}^{m_i} x_{ij} = 1, i = 1,2,\dots,n \quad (4)$$

$$\sum_{j=1}^{m_i} x_{ijkl} = x_{kl}, \sum_{l=1}^{m_k} x_{ijkl} = x_{ij} \quad (5)$$

$$x_{ij},\ x_{ijkl} \in \{0,1\} \quad (6)$$

$$i = 1,2,\dots,n-1, j = 1,2,\dots,m_i \quad (7)$$

$$k = i+1, i+2,\dots,n, j = 1,2,\dots,m_k \quad (8)$$

where n represents the number of residues and m_i the number of rotamers at the i-th residue. Eq. (4), together with Eq. (6), asserts that one, and only one, rotamer r_{ik} is assigned to the side-chain σ_i of the i-th residue (because one, and only one, binary variable x_{ik} is unit) and thus the interaction between the rotamer and backbone is accounted for in Eq. (3). Eq. (5), together with Eq. (6), asserts that one, and only one, interaction between r_{ik} and r_{jl} is accepted between σ_i and σ_j, respectively (by having an appropriate binary variable x_{ijkl} unit). It is called an *integer linear* programming problem because of these two types of integer variables and all equations are linear in the variables.

In 2000, Althaus et al. initially formulated the SCP-problem as an ILP-problem and then they first tried to solve LP (Linear programming)-relaxation of some binary decision variables and used the branch-and-bound method if the relaxation does not properly produce an integral solution (Althaus et al., 2000, 2002). Note that LP can be solved much more efficiently than ILP can be, using algorithms such as the simplex method by Danzig (Dantzig, 1949), the ellipsoid method by Khachian (Khachian, 1979), the interior point method by Karmarkar (Karmarkar, 1984), etc. In 2001, Eriksson et al. observed that the LP-relaxation of the ILP-formulation of the SCP-problem always found integral solutions (Eriksson et al., 2001). Similar observations were made by Klepeis et al. (Klepeis et al., 2003, 2004), Kingsford et al (Kingsford et al., 2005) etc. In 2007, Zhu proved that the problem can be MILP (mixed integer linear programming)-relaxed by decomposing the SCP-problem into $n(n-1)/2$ transportation problems where the coefficient matrix of the transformed formulation was totally unimodular (Zhu, 2007). Thus, the SCP-problem can be now formulated as an MILP problem which can be solved much faster than its ILP counterpart. Chazelle et al. formulated the SCP-problem as a quadratic integer programming that could be relaxed into semidefinite programming (SDP) which was solved in polynomial time by an interior-point method (Chazelle et al., 2003, 2004b). However, current SDP solvers are very limited to solving the small-sized problems. While it might have limitation for moderate sized or big proteins, studies on mathematical optimization approach are continuing in pursuit of both to better understand problem nature and to devise computationally efficient algorithms (Xie and Sahinidis, 2006; Canzar et al., 2011).

Search space reduction

A mathematically sound result was reported for the SCP-problem before mathematical programming approach started. In 1992, Desmet and colleagues initially reported the *dead-end elimination* (DEE) algorithm that pruned rotamers that were guaranteed to be incompatible with the global minima thus significantly reducing problem size and solution space (Desmet et al., 1992). This work is important because the MILP formulation remains NP-hard and thus the reduction of problem size is critical even for moderate size proteins. The idea is as follows: For each rotamer r_{ij} at each residue σ_i, to decide if it can be pruned, DEE evaluates its energy E_{ij} (determined by both backbone and the rotamers assigned to all the other residues) to compare with the case of another rotamer $r_{ij'}$ at σ_i. The original Desmet DEE is

$$\{E_{B\sigma}(\tilde{\imath},t)-E_{B\sigma}(\tilde{\imath},a)\}+\left\{\sum_{k\neq\tilde{\imath}}\min_{l}E_{\sigma\sigma}(\tilde{\imath},t,k,l)-\sum_{k\neq\tilde{\imath}}\max_{l}E_{\sigma\sigma}(\tilde{\imath},a,k,l)\right\}>0 \quad (9)$$

where $\tilde{\imath}$ is the index of the target residue (which is being tested) and a is the index of an alternative to the target rotamer t (which is being tested if it can be eliminated from solution). Eq. (9) implies that a rotamer can be removed if its best-case energy is worse than the worst-case energy of its replacement at the same residue. Eq. (9) may iterate as many as the total number of possible rotamers that can be assigned to the protein. The rationale of DEE is that the reduction of the computation to find solution with the reduced rotamer set sufficiently justifies the computation additionally required by the DEE algorithm.

In 1994, Goldstein improved Desmet DEE by looking at the difference of two rotamers at each residue given as the following (Goldstein, 1994)

$$\{E_{B\sigma}(\tilde{\imath},t)-E_{B\sigma}(\tilde{\imath},a)\}+\sum_{k\neq\tilde{\imath}}\min_{l}\{E_{\sigma\sigma}(\tilde{\imath},t,k,l)-E_{\sigma\sigma}(\tilde{\imath},t,k,l)\}>0 \quad (10)$$

Eq. (10) means that a rotamer t can be removed if there exists another rotamer a at the same residue such that the total sum of the energy difference that t and a determine is significant. The Goldstein DEE reduces solution space with a marginal increment of computation more than Desmet DEE does.

Other studies followed in order to improve the power of space reduction but with the cost of computation increment (Fung et al., 2008b; Gordon and Mayo, 1998; Looger and Hellinga, 2001; Pierce et al., 2000; Gordon et al., 2002; Xie and Sahinidis, 2006). Despite of its efficiency for small to moderate-sized proteins, DEE in general quickly deteriorates as protein size increases. This property is particularly important because the SCP-problem remains NP-hard with the remaining rotamers after the DEE-filtration. Piece et al. contains a detailed analytical analysis of the performances of various DEE algorithms (Pierce et al., 2000).

Heuristic methods

As mathematical optimization approach leaves the problem NP-hard and the DEE often still leaves relatively big solution space, most practical and popular approaches are based on heuristic methods of various kinds. There is another, possibly more important, issue for real problems: The SCP-problem involves more than just rotamers; there are other factors that would be better to take into consideration but difficult to formulate in a compact mathematical optimization problem. Various types of heuristic algorithms were reported since the report of the first

rotamer library: Simulated annealing (Lee and Subbiah, 1991; Holm and Sander, 1991; Shenkin et al., 1996; Tuffery et al., 1993), genetic algorithm (Tuffery et al., 1991, 1993), Monte carlo simulation (Holm and Sander, 1991, 1992; Xiang and Honig, 2001), graph theoretic with heuristic search (Canutescu et al., 2003; Samudrala and Moult, 1998b; Xu, 2005).

Along with the report of the PR83 library, Ponder and Richards presented an idea how to use it to predict structure based on two criteria: avoidance of steric overlap and complete filling of available space (Ponder and Richards, 1987). In this very first work, the idea of clash check of both backbone vs. side-chain and side-chain vs. side-chain was already presented where the steric overlap was measured by a table of interatomic contact. For example, two hydrogen atoms were regarded clashing if the distance is less than 2.0Å; between hydrogen and carbon, the distance was 2.4Å; between two carbon atoms, 3.0Å, etc. Dunbrack and Karplus also showed how to use their DK93 library (Dunbrack Jr. and Karplus, 1993) with an idea similar to Ponder and Richards (Ponder and Richards, 1987): After an initial assignment of rotamers at residues, a structure was refined based on the clash measured by the potential energy.

SCWRL (Side-chain with rotamer library) (Canutescu et al., 2003; Krivov et al., 2009) is probably the most popular program used today. SCWRL3 (2003) uses DCdep97 library (Canutescu et al., 2003) and SCWRL4 (2009) uses SD11 (reported later by Shapovalov and Dunbrack in 2011) (Shapovalov and Dunbrack Jr., 2011). SCWRL uses an interaction graph G=(V, E) with vertices in V representing residues and edges in E positive interaction between at least two rotamers, one from each residue. Then, G goes through a decomposition phase consisting of three steps. The first step is *edge-decomposition* where $e \in E$ with an interaction weaker than a threshold is removed after appropriately increasing the self-energy of the vertices of e to account for the interaction. Then, Goldstein DEE is performed to eliminate some rotamers. Sufficient repetition of this step reduces G to a forest G', possibly with several disconnected components, called clusters in SCWRL. The second step is *graph-decomposition* where each cluster is decomposed into a set of biconnected components (A graph is biconnected if the removal of a single vertex still leaves the graph connected). The third step is *tree-decomposition* which transforms G' to a special binary tree. To define $e \in E$ of G(E, V), SCWRL4 uses a priori defined convex polytope, called kDOP, bounding each geometric object such as rotamer, subrotamer, etc. to accelerate clash checking between two rotamers (Klosowski et al., 1998). SCWRL4 uses both repulsive and attractive terms, with CHARMM param19 potential (Brooks et al., 1983), for both van der Waals and hydrobonding forces, but much simplified form compared to Lennard-Jones formula. To optimize several parameters, SCWRL uses statistical learning. The rationale of decomposition of SCWRL is based on the observation of Canutescu et al. (Canutescu et al., 2003): A graph can be decomposed into a number of biconnected components to find the optimal solution for each component by observing that each residue with a single rotamer or a single neighbor can be eliminated from the graph. We mention that SCWRL4 implements the TreePack algorithm, reported by Xu and Berger running five times faster on average and up to 90 times faster than SCWRL3 by constructing and decomposing the geometric neighborhood graph (Xu and Berger, 2006), to improve the performance of earlier SCWRL3.

CIS-RR (Clash-detection guided iterative search with rotamer relaxation) consists of two phases (Cao et al., 2011): CIS and RR. CIS phase is similar to SCWRL in that it assigns initial rotamers and update to avoid clashes among rotamers as much as possible (but the definition of clash is a bit not clear). A simpler potential function is used with the CHARMM param19. RR phase is to modify the dihedral angles of rotamers so that the energy of each rotamer can be minimized in its neighborhood by the local minimization with the conjugate gradient method. RR phase is based on the work (Wang et al., 2005). However, it seems that solution improvement is marginal compared to the increase of computation time after the CIS phase. CIS-RR is further improved to be faster into RASP (Rapid side-chain predictor) using a cocktail of DEE, graph-based search, Monte Carlo search, and backtracking (Miao et al., 2011). CIS-RR and RASP use learning process to optimize parameters.

The nature of the SCP-problem is a combinatorial optimization problem but with properties difficult to precisely translate into combinatorial terms only. In addition, the adjustment of dihedral angles of combinatorially optimal rotamers may improve solution quality. Hence, it seems that heuristic approach to find a reasonably good solution within a reasonable amount of computation time is probably the best strategy. To achieve this goal, good tactics might be to find (T1) an efficient way to represent the potential function easier to evaluate than the current popular Lennard-Jones form, (T2) an effective way to reduce solution space, and (T3) an efficient search strategy to traverse solution space.

Regarding on T1, many prior studies use simpler rational polynomial functions possibly with truncations at both extreme ends in order to evaluate faster but less with a significant discrepancy from the (12, 6) Lennard-Jones form. The trade-off between computation time and solution quality of potential function should be carefully studied and a more carefully traded method might be desirable. DEE algorithms are obviously very effective method for T2 and almost all prior works use one of DEE algorithms at a certain stage regardless a mathematical optimization or a heuristic is used, frequently before an optimization procedure. For T3, a computationally effective and efficient way for detecting clash seems critical. It is notable that, while the detection of clash between rotamers can be conveniently handled as a geometry problem, such an approach has not been explicitly treated in prior studies. We believe that an efficient way of explicitly managing geometry based on the *geometrization* concept might be very much useful for resolving the three tactical points, perhaps simultaneously.

In fact, geometric properties such as solvent accessibility,

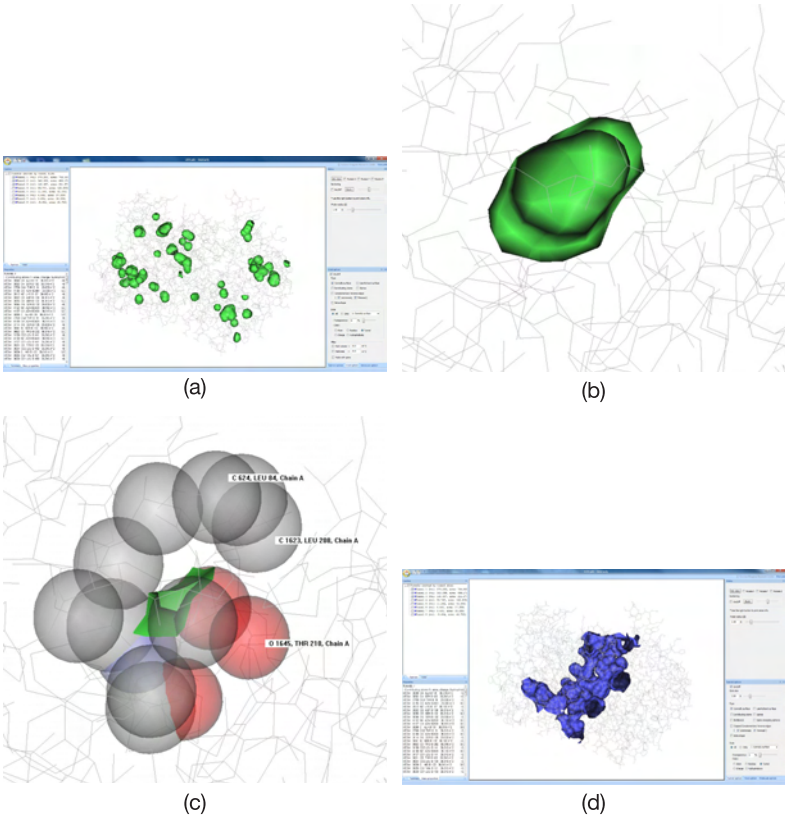

FIGURE 2 | An example of molecular voids and tunnels (PDB accession code: 1JD0). Computed and visualized by the BetaVoid program (Kim et al., 2014): (a) The Connolly voids for water, (b) a zoom-up of one Connolly void, (c) the Lee-Richards void with the atoms contributing to the boundary, and (d) a tunnel.

contact surfaces, excluded volume, etc. have been used as important measures for the quality of protein structure (Eyal et al., 2004; Kussell et al., 2001). In this regard, authors' group has been developing a formal geometric theory under the concept of a new discipline called *Molecular Geometry*. The BetaVoid program (Kim et al., 2014), freely available at the Voronoi Diagram Research Center (VDRC, http://voronoi.hanyang.ac.kr), is one of the computational tools immediately useful for the quality evaluation of designed protein structure and perhaps for many other applications in structural biology in general. Figure 2 shows molecular voids and tunnels computed by the BetaVoid program and another under development (Kim et al., 2013). Figure 2(a) and (b) show the entire voids and the zoom-up of one particular void, respectively. Figure 2(c) shows the same void computed with the atoms contributing to the void boundary. Note that the boundaries of the voids in Figure 2(a) and (b) are Connolly surface and that of Figure 2(c) is Lee-Richards solvent accessible surface, both corresponding to water molecule (i.e. a sphere with the radius 1.4Å) (For the definition of Connolly surface and Lee-Richards surface, see (Kim et al., 2010c)). Figure 2(d) shows a tunnel that water molecule can pass through. Further BetaVoid functions can be easily added upon request of collaboration.

Authors' group has recently reported the first version of the BetaSCP algorithm and program (Ryu and Kim, 2013) based on the geometrization concept using the Voronoi diagram of atoms (Kim et al., 2005; Kim and Kim, 2006), the quasi-triangulation (Kim et al., 2006, 2010a), and the beta-complex (Kim et al., 2010b). This algorithm decomposes an SCP-problem into a number of small SCP-problems in the neighborhood of each residue where the neighborhood is defined by the Voronoi diagram and beta-complex. Then, each small SCP-problem is formulated as an MILP-problem solved by CPLEX library. Experiment with some tested structures shows that the solutions were within 0.01% of optima with time complexity linear with the number of residues.

This version, however, solves small proteins and computational requirement is still high. The second version under development, BetaSCP2, solves problems of arbitrary size very efficiently with excellent results. The basic idea of BetaSCP2 is to use geometrization more explicitly in a more clever way. The solution quality seems to outperform both SCWRL4 and CIS-RR with computation time slightly more than SCWRL4 yet significantly less than CIS-RR.

COMPUTATIONAL PROTEIN DESIGN

Challenges in protein design

A protein can be defined by specifying its amino acid sequence which directly one-to-many maps to RNA sequences where each again one-to-one maps to a DNA sequence of a gene, assuming that translation and transcription are error-free. While a new amino acid sequence can be encoded into a bacterial gene through the symbolic information produced by computer keyboard, such a sequence may or may not be expressed as a meaningful protein. Designing a protein with a desired structure and function is in theory possible but in practice remains a challenge yet in both time and effort.

An amino acid sequence of a fixed length n maps to a set S of 20^n sequence instances, i.e. exponential to n. Even if a tiny subset $\mathscr{S} \in S$ shows a function similar to the desired one, the size of $\mathscr{S}$ is still huge for human to examine. If n is not fixed, such a set $\mathscr{S}$ is even larger. Designing a protein with a target function is to choose an appropriate element of $\mathscr{S}$, hopefully the best one from a set of criteria. An example: For a protein of 100 residues, there are 20^{100} possible combinations of protein sequences and for each of these sequences, there could be about 10^{100} energetically reasonable conformations. Note that 10^{100}, called a googol, is regarded as

the upper bound of the number of atoms in universe. Lest and Chothia actually showed that proteins with sequence similarity ≤ 10% could have remarkably similar structures (Lesk and Chothia, 1982). However, it is also well-known that a random sequence of amino acids does not produce protein just like a set of random alphabets does not make a Shakespear. *Designability* of a protein is a notion to explain if a protein structure is designable or not and in this regard such a random sequence is not designable. Hence, there are measures to access the designablity of a protein structure: The well-packedness and density of protein structure is such one and many early designed proteins did not packed well in their interior.

Protein design is an engineering effort about nano-machinary which has been a target for nanotechnolgy, a term defined by Norio Taniguchi (Taniguchi, 1974). In his seminal paper "Molecular engineering" (Drexler, 1981) and the book "Engines of Creation," (Drexler, 1987), Eric Drexler explicitly pointed out protein design issue inheriting the Richard Feynman's 1959 talk "There's Plenty of Room at the Bottom."

Gutte's early work, in 1975, on the synthesis of an artificial enzyme still showing enzymic activity is probably the beginning of protein engineering, but without any computational aide (Gutte, 1975, 1977). In 1978, Hutchison et al. designed a method for changing a specific nucleotide in a DNA sequence with high efficiency (HutchisonIII et al., 1978). Mas and collaborators produced chimeric phosphoglycerate kinases that contained one domain from human enzyme and the other from yeast enzyme and showed that the 35% difference in amino acid sequences between native enzymes and the chimeras had only a small effect on substrate binding and conformational changes occurring during catalysis (Mas et al., 1986). Thus, it is not surprise that proteins can be extremely tolerant to single mutations (Rennell et al., 1991). The fraction of functional proteins decreases roughly exponentially with the number of substitutions, although the severity of this decline varies among proteins (Bloom et al., 2005). Knowles, using point mutations, described the types of change of an enzyme and discussed the effects on the protein anatomy (i.e. structure and stability) and protein physiology (i.e. enzyme specificity and mechanism) focusing on catalysis at active sites (Knowles, 1987; Ackers and Smith, 1985). In the review paper on *de novo* design, Richardson and Richardson mainly discussed experimental approach (Richardson and Richardson, 1989).

Up to this point, protein engineering was purely experimental, without any computational aide, and is inherited today by directed evolution (Jäckel et al., 2008; Verma et al., 2012; Lutz, 2010; Chen, 2001) to get incrementally improved structure through mutagenesis and screening (In these days, directed evolution is also sometimes guided by computational methods (Wong et al., 2007; Verma et al., 2012)). However, protein design based on only experiment faces challenges of two kinds (Klepeis et al., 2004): i) the space of mutations is too large to thoroughly examine, and ii) single random mutations rarely improve a property of interest and simultaneous multiple mutations may dilute the possibilities for improvement. Computational approach to protein design emerged to supplement experimental approach by overcoming these difficulties. Ponder and Richards first rotamer library in 1987 (Ponder and Richards, 1987) was the basis of the birth of computational protein design which started as an elaboration of Drexler's 1981 suggestion. In fact, Ponder and Richards work was done from folding point of view. It is important to note that the two approaches are not mutually exclusive but are supplementary to each other. Experimental approach should be supported by computations to explore alternatives and computational approach should be verified by experiment.

Definition of computational protein design

Consider a set S of sequences is mapped to a set of points in a high dimensional space, a Euclidean space to make the analogy simple. Let $\mathscr{S} \subset S$ be a densely clustered subset of the points where each has a structure close to the target structure. Assuming that there exists such a sufficiently large set $\mathscr{S}$, the best practical, engineering approach is to find one sequence element $s \in S$, whose structure is not necessarily close to the target structure, and traverse the space to reach another point $s' \in S$ in the vicinity of s with a certain measure f . Hence, the issue is how to find an initial point (i.e. sequence) s, the definition of vicinity v, and the fitness measure f. Obviously, there are multiple ways to each of these three issues and thus it is not surprising to anticipate different solutions to designing a protein which show different specificity. Since each solution needs to be verified and validated, *in vitro* and *in vivo* exploration of the set S does not make sense from both time and cost point of view. Thus, computational protein design should be a prerequisite prior to wet lab studies if a significant solution is to be found. Drexler's early speculation in 1987 (Drexler, 1987) about the popularity of protein design using Computer Aided Design (CAD) system is still being pursued in various research efforts which takes different ways to define the three factors.

Protein design studies can be categorized into three major groups from computational point of view: redesign, *de novo* design of Type I, and *de novo* design of Type II. In the redesign category, given a protein structure in that backbone coordinates and residue types are fixed, we want to modify some residue types with or without perturbing backbone dihedral angles at and around the modified residues. At each case of residue modification and backbone perturbation, the SCP-problem is solved with a certain definition of cutoff radius to trade solution quality with computation time. Thus, in redesign, a given backbone can be considered to give the seed s and residue modification and backbone perturbation the vicinity function v. *De novo* design of Type I assumes backbone coordinates are fixed (thus protein size in terms of the number of residues is also fixed) but residue types are not fixed. The method first determines the type of residues and then solves a redesign problem. *De novo* design of Type II assumes neither backbone coordinate fixed nor residue types fixed (Protein size may be fixed). Thus, this method first

determines protein sequence and predicts its secondary structure and topology using a homology modeling technique. Then, it determines backbone coordinates. Then, it solves a redesign problem. Thus, all methods repeatedly solve the SCP-problem.

Efforts for developing design programs

Computational protein design can only be done through software and an algorithm does not make much sense unless it is implemented into a program that can be easily used by designer who is not necessarily a programming expert. In this regard, a few significant programs are briefly reviewed. Many other good reviews are also available (Lippow and Tidor, 2007; Park et al., 2004; gu Kang and Saven, 2007; Lazar et al., 2003; Butterfoss and Kuhlman, 2006; Damborsky and Brezovsky, 2009; Lutz, 2010; Pokala and Handel, 2001). For the potential functions for protein design, see (Boas and Harbury, 2007; Mendes et al., 2002; Pokala and Handel, 2004, 2005); for electrostatics, in particular, see (Vizcarra and Mayo, 2005).

Dahiyat and Mayo's seminal work (Dahiyat and Mayo, 1997) is considered the first significant work on computational *de novo* protein design which designed a protein of 28 residues using the PR87 rotamer library and DEE algorithm with a fixed backbone (FSD-1, a protein that adopts the zinc finger fold). It designs full sequence of residues for all parts of protein such as the buried core, the solvent-exposed surface, and the boundary between core and surface, beginning with a backbone fold and an SCP-solver based on the DEE algorithm. It was used to find a new design for promoting stability of target protein over the wild type (Malakauskas and Mayo, 1998; Shah et al., 2007), to design calmodulin (Shifman and Mayo, 2003), etc. The effort by the Mayo group has grown up as the ORBIT (optimization of rotamers by iterative techniques) program.

The most popular protein design program is Rosetta, developed by David Baker group at University of Washington, which can be considered as the most powerful program of the *De novo* design of Type II. According to (Leaver-Fay et al., 2011), Rosetta was initially written in FORTRAN77 as two separate programs for protein structure prediction by Simons et al. (Simons et al., 1997) and for protein design by Kuhlman and Baker (Kuhlman and Baker, 2000), merged, mechanically ported to C++ and evolved thereafter. The principles and methods implemented in Rosetta algorithms for de novo design is summarized in (Rohl et al., 2004) For the functionality Rosetta program, see (Das and Baker, 2008; Leaver-Fay et al., 2011). Rosetta has also been used for various applications: Predicting RNA structures (Das et al., 2010), protein-DNA interfaces (Morozov et al., 2005; Ashworth and Baker, 2009), domain boundary prediction (Kim, 2005), etc. RosettaDesign was used to design mCrel homing endonuclease (Ulge et al., 2011), Kemp Eliminase KE70 (Khersonsky et al., 2011), designing enzyme more powerful than natural one (Röthlisberger et al., 2008; Jiang et al., 2008), increasing antibody-antigen affinity (Lippow et al., 2007), engineering peptide transcription factors that oligomerize with high specificity (Grigoryan et al., 2009), molecular replacement problem (DiMaio, 2013), an inhibitor of influenza hemagglutinin (Fleishman et al., 2011), design of protein-protein interfaces (Karanicolas et al., 2011), to design enzyme catalysts (Siegel et al., 2010), to predict the transiently formed state of a T4 lysozyme mutant (Bouvignies et al., 2011), design of proteins targeting the conserved stem region of influenza hemagglutinin (Fleishman et al., 2011), etc.

In protein design, one of the very important issues is how to reflect backbone flexibility appropriately (Yin et al., 2007a,b; Friedland et al., 2008; Smith and Kortemme, 2008). Friedl et al. (Friedland et al., 2008) generated an ensemble of ten near-native backbone structures to represent backbone variations using the Monte Carlo simulation of backrub, which is a motion that modifies backbone geometry locally in that the C_α sweeps around a circle perpendicular to an axis passing through two C_α s in the immediate neighbor. By sampling the rotational axis, say by each 10 degree, the perturbed backbone ensemble can be produced. Rosetta also implements backrub algorithm (Lauck et al., 2010). A good review on backbone flexibility is by Mandell and Kortemme (Mandell and Kortemme, 2009).

Floudas' group at Princeton University views protein design problem in a two- stage framework (2003): the selection of sequences and rank the fold specificities of the selected sequences. For the first stage, i.e. sequence selection, assuming a single or an ensemble of backbone structures are given, they formulated a quadratic assignment like model in an integer programming problem formulation (Klepeis et al., 2003, 2004) and its improvement to an ILP formulation for a faster solution process (Fung et al., 2005, 2007) using CPLEX program (ILOG S.A., 2003). The mathematical optimization model incorporates all possible combinations of amino acids at all residues where the cost coefficients are derived by the energy which depended on the distances between alpha carbons on backbone which can be flexible. Rather than being a continuous function, the energy parameters, i.e. the cost coefficients, were stored in a finite set of discrete bins. To reflect flexibility of backbone, the weighted average force field defined by an ensemble of structures from NMR (Fung et al., 2007) or molecular dynamics simulation result (Fung et al., 2008b,a) was used. The second stage actually contains the SCP-problem for a selected sequence on backbone(s) which was solved using an NMR structure refinement program called CYANA (Combined assignment and dynamics algorithm for NMR applications). The potential energy of hundreds of random structures generated from CYANA was evaluated using TINKER program to be ranked. Floudas' design method, later named WISDOM (Smadbeck et al., 2013), has been applied to find entry inhibitors for HIV-1 (Bellows et al., 2010a), to design new compstatin variants (Bellows et al., 2010b; de Victoria et al., 2011; Tamamis et al., 2011), to design complement component 3a receptor (C3aR) agonists and antagonists (Bellows-Peterson et al., 2012), to design lasso peptide antibiotic (Pan et al., 2011).

Maranas from Pennsylvania State University has developed the IPRO (iterative protein redesign and optimization) program

which designs a protein library with targeted ligand specificity by minimizing the binding energy with the desired ligand (Saraf et al., 2006). IPRO belongs to *de novo* design of Type I and relies on identifying mutations in parental sequences which propagates down to the combinatorial library. It is based on the cycling of sequence design, ligand redocking, and backbone movement. Optimization is done by MILP formulation based on the cost coefficients measured by the CHARMM parameters. IPRO has been used to computationally altering the effector binding specificity of a regulatory protein (Fazelinia et al., 2007), cofactor specificity (Khoury et al., 2009), to design the binding portions of antibodies to have high specificity and affinity against any targeted epitope of an antigen (ie. the antibody complementarity determining regions (CDRs) that are most likely to be able to favorably bind the antigen) (Pantazes and Maranas, 2010), etc. Transfer of a binding site onto an existing protein scaffold is also incorporated to IPRO (Fazelinia et al., 2009).

Desjarlais and Handel at UC Berkeley developed a protein redesign program for a fixed backbone called ROC (repacking of cores) based on genetic algorithm for optimization and was used to design the hydrophobic core of proteins (Desjarlais and Handel, 1995). ROC was further developed into the design program called SoftROC that can perturb backbone by randomly adjusting φ and ψ angles using Monte Carlo sampling of dihedral angles and genetic algorithm for optimization in 1999 (Desjarlais and Handel, 1999; Chowdry et al., 2007). Another notable case is Hellinga group at Duke University, mainly applying DEE with fixed backbone for designing receptor and sensor proteins with novel ligand-binding functions (Looger et al., 2003) using the Dezymer program (developed by Hellinga and Richards (Hellinga and Richards, 1991)) even if it has been retracted due to technical problem. Hellinga group reported design to introduce iron and oxygen binding sites in thioredoxin (Benson et al., 1998, 2000), to confer novel enzymatic properties onto ribose-binding protein (Dwyer et al., 2004), etc. Medusa by Dokholyan group at University of North Carolina (Ding and Dokholyan, 2006) is another example using fixed backbones: A specified amino acid is mutated and Monte Carlo simulation is used to find the best rotamer at the residue. In a flexible backbone method, when backbone strain is detected, conjugate gradient minimization is done for the total energy with respect to backbone dihedral angles. Medusa has been also used for RNA fold prediction (Sharma et al., 2008),

CONCLUSION

The relationship between structure and amino acid sequence is important for understanding life. As life functions based on protein structure, the prediction and design of protein structure has long been one of the most intensive research areas. In this paper, we reviewed the side-chain prediction problem and the protein design problem, mainly from computational point of view because algorithms for solving such problems are not much sensible unless they are fully implemented to be used by domain experts.

ACKNOWLEDGEMENTS

This research was supported by NRF (No. 2012R1A2A1A050 26395), Korea.

Original Submission: February 17, 2014
Revised Version Received: March 13, 2014
Accepted: March 14, 2014

REFERENCES

Ackers, G.K., and Smith, F.R. (1985). Effects of site-specific amino acid modification on protein interactions and biological function. *Annu Rev Biochem* **54**, 597–629.

Alexander, N.S., Stein, R.A., Koteiche, H.A., Kaufmann, K.W., Mchaourab, H.S., and Meiler, J. (2013). RosettaEPR: Rotamer library for spin label structure and dynamics. *PLoS ONE* **8**.

Althaus, E., Kohlbacher, O., Lenhof, H.P., and Müller, P., A combinatorial approach to protein docking with flexible side-chains. In *RECOMB '00 Proceedings of the fourth annual international conference on Computational molecular biology*, 15–24.

Althaus, E., Kohlbacher, O., Lenhof, H.P., and Müller, P. (2002). A combinatorial approach to protein docking with flexible side chains. *J Comput Biol* **9**, 597–612.

Ashworth, J., and Baker, D. (2009). Assessment of the optimization of affinity and specificity at protein-DNA interfaces. *Nucleic Acids Res* **37**.

Bellows, M.L., Fung, H.K., Taylor, M.S., Floudas, C.A., de Victoria, A.L., and Morikis, D. (2010a). New compstatin variants through two de novo protein design frameworks. *Biophys j* **98**, 2337–2346.

Bellows, M.L., Taylor, M.S., Cole, P.A., Shen, L., Siliciano, R.F., Fung, H.K., and Floudas, C.A. (2010b). Discovery of entry inhibitors for hiv-1 via a new de novo protein design framework. *Biophys j* **99**, 3445–3453.

Bellows-Peterson, M.L., Fung, H.K., Floudas, C.A., Kieslich, C.A., Zhang, L., Morikis, D., Wareham, K.J., Monk, P.N., Hawksworth, O.A., and Woodruff, T.M. (2012). De novo peptide design with c3a receptor agonist and antagonist activities: Theoretical predictions and experimental validation. *J Med Chem* **55**, 4159–4168.

Benson, D.E., Wisz, M.S., and Hellinga, H.W. (1998). The development of new biotechnologies using metalloprotein design. *Curr Opin biotech* **9**, 370–376.

Benson, D.E., Wisz, M.S., and Hellinga, H.W. (2000). Rational design of nascent metalloenzymes. *P Nati Acad Sci USA* **97**, 6292–6297.

Bhuyan, M.S.I., and Gao, X. (2011). A protein-dependent side-chain rotamer library. *BMC bioinformatics* **12 Suppl 14**.

Bloom, J.D., Meyer, M.M., Meinhold, P., Otey, C.R., MacMillan, D., and Arnold, F.H. (2005). Evolving strategies for enzyme engineering. *Curr Opin Struc Biol* **15**, 447–452.

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

Bouvignies, G., Vallurupalli, P., Hansen, D.F., Correia, B.E., Lange, O., Bah, A., Vernon, R.M., Dahlquist, F.W., Baker, D., and Kay, L.E. (2011). Solution structure of a minor and transiently formed state of a t4 lysozyme mutant. *Nature* **477**, 111–117.

Bower, M.J., Cohen, F.E., and Dunbrack Jr., R.L. (1997). Prediction of protein sidechain rotamers from a backbone-dependent rotamer library: A new homology modeling tool. *J Mol Biol* **267**, 1268–1282.

Brooks, B.R., Bruccoleri, R.E., Olafson, B.D., States, D.J., Swaminathan, S., and Karplus, M. (1983). CHARMM: A program for macromolecular energy, minimization, and dynamics calculations. *J Comput Chem* **4**, 187–217.

Butterfoss, G.L., and Kuhlman, B. (2006). Computer-based design of novel protein structures. *Annu Rev Bioph Biom* **35**, 49-65.

Canutescu, A.A., Shelenkov, A.A., and Dunbrack Jr., R.L. (2003). A graph-theory algorithm for rapid protein side-chain prediction. *Protein Sci* **12**, 2001–2014.

Canzar, S., Toussaint, N.C., and Klau, G.W. (2011). An exact algorithm for side-chain placement in protein design. *Optim Lett* **5**, 393–406.

Cao, Y., Song, L., Miao, Z., Hu, Y., Tian, L., and Jiang, T. (2011). Improved side-chain modeling by coupling clash-detection guided iterative search

with rotamer relaxation. *Bioinformatics* **27**, 785–790.

Chandrasekaran, R., and Ramachandran, G.N. (1970). Studies on the conformation of amino acids. xi. analysis of the observed side group conformation in proteins. *Int J Prot Res research* **2**, 223–233.

Chazelle, B., Kingsford, C., and Singh, M. (2003). The side-chain positioning problem: A semidefinite programming formulation with new rounding schemes. In D.Q. Goldin, A.A. Shvartsman, S.A. Smolka, J.S. Vitter, and S.B. Zdonik, editors, *Proceedings of the ACM International Conference Proceeding Series;Proceedings of the Paris C. Kanellakis memorial workshop on Pr*, **41**, 86–94.

Chazelle, B., Kingsford, C., and Singh, M. (2004a). The inapproximability of side-chain positioning. Technical report, Princeton University.

Chazelle, B., Kingsford, C., and Singh, M. (2004b). A semidefinite programming approach to side chain positioning with new rounding strategies. *Informs J Comput* **16**, 380–392.

Chen, R. (2001). Enzyme engineering: Rational redesign versus directed evolution. *Trends Biotechnol* **19**, 13–14.

Chowdry, A.B., Reynolds, K.A., Hanes, M.S., Voorhies, M., Pokala, N., and Handel, T.M. (2007). Software news and update an object-oriented library for computational protein design. *J Comput Chem* **28**, 2378–2388.

Cornell, W.D., Cieplak, P., Bayly, C.I., Gould, I.R., Merz, J., M., K., Ferguson, D.M., Spellmeyer, D.C., Fox, T., Caldwell, J.W., and Kollman, P.A. (1995). A second generation force field for the simulation of proteins, nucleic acids, and organic molecules. *J Am Chem Soc* **117**, 5179–5197.

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

Dahiyat, B.I., and Mayo, S.L. (1997). De novo protein design: Fully automated sequence selection. *Science* **278**, 82–87.

Damborsky, J., and Brezovsky, J. (2009). Computational tools for designing and engineering biocatalysts. *Curr Opin Chem Biol* **13**, 26–34.

Dantzig, G.B. (1949). Programming of interdependent activities, 11: Mathematical model. *Econometrica* **17**, 200–211.

Das, R., and Baker, D., Macromolecular modeling with Rosetta. *Annu Rev Biochem* **77**, 363-382.

Das, R., Karanicolas, J., and Baker, D. (2010). Atomic accuracy in predicting and designing noncanonical RNA structure. *Nat Methods* **7**, 291–294.

Desjarlais, J.R., and Handel, T.M. (1995). De novo design of the hydrophobic cores of proteins. *Protein Sci* **4**, 2006–2018.

Desjarlais, J.R., and Handel, T.M. (1999). Side-chain and backbone flexibility in protein core design. *J Mol Biol* **290**, 305–318.

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

DiMaio, F. (2013). Advances in rosetta structure prediction for difficult molecular-replacement problems. *Acta Crystallogr D* **69**, 2202–2208.

Ding, F., and Dokholyan, N.V. (2006). Emergence of protein fold families through rational design. *Plos Comput Biol* **2**, 0725–0733.

Drexler, E., *Engines of Creation: The Coming Era of Nanotechnology* (Anchor, 1987).

Drexler, K.E. (1981). Molecular engineering: An approach to the development of general capabilities for molecular manipulation. *P Natl Acad Sci U S A* **78**, 5275–5278.

Dunbrack Jr., R.L. (2002). Rotamer libraries in the 21st century. *Curr Opin Struc Biol* **12**, 431–440.

Dunbrack Jr., R.L., and Cohen, F.E. (1997). Bayesian statistical analysis of protein side-chain rotamer preferences. *Protein Sci* **6**, 1661–1681.

Dunbrack Jr., R.L., and Karplus, M. (1993). Backbone-dependent rotamer library for proteins. *J Mol Biol* **230**, 543–574.

Dunbrack Jr., R.L., and Karplus, M. (1994). Conformational analysis of the backbone-dependent rotamer preferences of protein sidechains. *J Mol Biol* **1**, 334–340.

Dwyer, M.A., Looger, L.L., and Hellinga, H.W. (2004). Computational design of a biologically active enzyme. *Science* **304**, 1967–1971.

Eriksson, O., Zhou, Y., and Elofsson, A. (2001). Side chain-positioning as an integer programming problem. *Lect Notes Comput SC* **2149**, 128–141.

Eyal, E., Najmanovich, R., Mcconkey, B.J., Edelman, M., and Sobolev, V. (2004). Importance of solvent accessibility and contact surfaces in modeling side-chain conformations in proteins. *J Comput Chem* **25**, 712–724.

Fazelinia, H., Cirino, P.C., and Maranas, C.D. (2007). Extending iterative protein redesign and optimization (ipro) in protein library design for ligand specificity. *Biophys J* **92**, 2120–2130.

Fazelinia, H., Cirino, P.C., and Maranas, C.D. (2009). Optgraft: A computational procedure for transferring a binding site onto an existing protein scaffold. *Protein Sci* **18**, 180–195.

Fleishman, S.J., Whitehead, T.A., Ekiert, D.C., Dreyfus, C., Corn, J.E., Strauch, E.M., Wilson, I.A., and Baker, D. (2011). Computational design of proteins targeting the conserved stem region of influenza hemagglutinin. *Science* **332**, 816–821.

Friedland, G.D., Linares, A.J., Smith, C.A., and Kortemme, T. (2008). A simple model of backbone flexibility improves modeling of side-chain conformational variability. *J Mol Biol* **380**, 757–774.

Fung, H.K., Taylor, M.S., and Floudas, C.A. (2007). Novel formulations for the sequence selection problem in de novo protein design with flexible templates. *Optim Method Softw* **22**, 51–71.

Fung, H., Rao, S., Floudas, C., Prokopyev, O., Pardalos, P., and Rendl, F. (2005). Computational comparison studies of quadratic assignment like formulations for the In silico sequence selection problem in De Novo protein design. *J Comb Optim* **10**, 41–60.

Fung, H.K., Floudas, C.A., Taylor, M.S., Zhang, L., and Morikis, D. (2008a). Toward full-sequence de novo protein design with flexible templates for human beta-defensin-2. *Biophys J* **94**, 584–599.

Fung, H.K., Welsh, W.J., and Floudas, C.A. (2008b). Computational de novo peptide and protein design: Rigid templates versus flexible templates. *Ind Eng Chem Res* **47**, 993–1001.

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

Gordon, D.B., and Mayo, S.L. (1998). Radical performance enhancements for combinatorial optimization algorithms based on the dead-end elimination theo- rem. *J Comput Chem* **19**, 1505–1514.

Gordon, D., Hom, G.K., Mayo, S.L., and Pierce, N.A. (2002). Exact rotamer optimization for protein design. *J Comput Chem* **24**, 232–243.

Grigoryan, G., Reinke, A.W., and Keating, A.E. (2009). Design of protein-interaction specificity gives selective bZIP-binding peptides. *Nature* **458**, 859–864.

Gutte, B. (1975). A synthetic 70-amino acid residue analog of ribonuclease S-protein with enzymic activity. *J Biol Chem* **250**, 889–904.

Gutte, B. (1977). Study of RNase a mechanism and folding by means of synthetic 63-residue analogs. *J Biol Chem* **252**, 663–670.

Harder, T., Boomsma, W., Paluszewski, M., Frellsen, J., Johansson, K.E., and Hamelryck, T. (2010). Beyond rotamers: A generative, probabilistic model of side chains in proteins. *BMC Bioinformatics* **11**.

Hellinga, H.W., and Richards, F.M. (1991). Construction of new ligand binding sites in proteins of known structure: I. computer-aided modeling of sites with pre-defined geometry. *J Mol Biol* **222**, 763–785.

Holm, L., and Sander, C. (1991). Database algorithm for generating protein back-bone and side-chain co-ordinates from a c^{α} trace: Application to model building and detection of co-ordinate errors. *J Mol Biol* **218**, 183–194.

Holm, L., and Sander, C. (1992). Fast and simple monte carlo algorithm for side chain optimization in proteins: Application to model building by homology. *Proteins* **14**, 213–223.

Hopfinger, A.J., *Conformational Properties of Macromolecules* (Academic Press, 1973).

HutchisonIII, C.A., Phillips, S., EdgeII, M.H., Gillam, S., Jahnke, P., and Smith, M. (1978). Mutagenesis at a specific position in a DNA sequence. *J Biol Chem* **253**, 6551–6560.

ILOG S.A. (2003). *ILOG CPLEX 9.0 User's Manual*.

Jäckel, C., Kast, P., and Hilvert, D. (2008). Protein design by directed evolution. *Ann Rev Biophys* **37**, 153–173.

James, M.N., and Sielecki, A.R. (1983). Structure and refinement of penicillopepsin at 1.8 Å resolution. *J Mol Biol* **163**, 299–361.

Janin, J., Wodak, S., Levitt, M., and Maigret, B. (1978). Conformation of

amino acid side-chains in proteins. *J Mol Biol* **125**, 357–386.

Jiang, L., Althoff, E.A., Clemente, F.R., Doyle, L., Röthlisberger, D., Zanghellini, A., Gallaher, J.L., Betker, J.L., Tanaka, F., Carlos F. Barbas, I., Hilvert, D., Houk, K.N., Stoddard, B.L., and Baker, D. (2008). De novo computational design of retro-aldol enzymes. *Science* **319**, 1387–1391.

Kang, S.G., and Saven, J.G. (2007). Computational protein design: structure, function and combinatorial diversity. *Curr Opin Chem Biol* **11**, 329–334.

Karanicolas, J., Corn, J.E., Chen, I., Joachimiak, L.A., Dym, O., Peck, S.H., Al- beck, S., Unger, T., Hu, W., Liu, G., Delbecq, S., Montelione, G.T., Spiegel, C.P., Liu, D.R., and Baker, D. (2011). A de novo protein binding pair by computational design and directed evolution. *Mol Cell* **42**, 250–260.

Karmarkar, N., A new polynomial-time algorithm for linear programming. In *Proceedings of the Sixteenth Annual ACM Symposium on Theory of Computing*, 302-311.

Khachian, L. (1979). A polynomial algorithm in linear programming. *Soviet Mathematics Doklady* **20**, 191–194.

Khersonsky, O., Röthlisberger, D., Wollacott, A.M., Murphy, P., Dym, O., Albeck, S., Kiss, G., Houk, K., Baker, D., and Tawfik, D.S. (2011). Optimization of the in-silico-designed kemp eliminase KE70 by computational design and directed evolution. *J Mol Biol* **407**, 391–412.

Khoury, G.A., Fazelinia, H., Chin, J.W., Pantazes, R.J., Cirino, P.C., and Maranas, C.D. (2009). Computational design of Candida boidinii xylose reductase for altered cofactor specificity. *Protein Sci* **18**, 2125–2138.

Kim, D.S., Cho, Y., and Kim, D. (2005). Euclidean Voronoi diagram of 3D balls and its computation via tracing edges. *Comput Aided Design* **37**, 1412–1424.

Kim, D.S., Cho, Y., Kim, J.K., and Sugihara, K. (2013). Tunnels and voids in molecules via voronoi diagrams and beta-complexes. *Transactions on Computational Science* **LNCS 8110**, 92–111.

Kim, D.S., Cho, Y., and Sugihara, K. (2010a). Quasi-worlds and quasi-operators on quasi-triangulations. *Comput Aided Design* **42**, 874–888.

Kim, D.S., Cho, Y., Sugihara, K., Ryu, J., and Kim, D. (2010b). Three-dimensional beta-shapes and beta-complexes via quasi-triangulation. *Comput Aided Design* **42**, 911–929.

Kim, D.S., Kim, D., Cho, Y., and Sugihara, K. (2006). Quasi-triangulation and interworld data structure in three dimensions. *Comput Aided Design* **38**, 808–819.

Kim, D.S., Won, C.I., and Bhak, J. (2010c). A proposal for the revision of molecular boundary typology. *J Biomol Struct Dyn* **28**, 277–287.

Kim, D., and Kim, D.S. (2006). Region-expansion for the Voronoi diagram of 3D spheres. *Comput Aided Design* **38**, 417–430.

Kim, J.K., Cho, Y., Laskowski, R.A., Ryu, S.E., Sugihara, K., and Kim, D.S. (2014). BetaVoid: molecular voids via beta-complexes and Voronoi diagrams. *Proteins* **DOI: 10.1002/prot.24537**.

Kim, V.N. (2005). MicroRNA biogenesis: Coordinated cropping and dicing. *Nat Rev Mol Cell Bio* **6**, 376–385.

Kingsford, C.L., *Computational Approaches to Problems in Protein Structure and Function* (Princeton University, 2005).

Kingsford, C.L., Chazelle, B., and Singh, M. (2005). Solving and analyzing side- chain positioning problems using linear and integer programming. *Bioinformatics* **21**, 1028–1036.

Klepeis, J.L., Floudas, C.A., Morikis, D., Tsokos, C.G., and Lambris, J.D. (2004). Design of peptide analogues with improved activity using a novel de novo protein design approach. *Ind Eng Chem Res* **43**, 3817–3826.

Klepeis, J.L., Floudas, C.A., Morikis, D., Tsokos, C.G., Argyropoulos, E., Spruce, L., and Lambris, J.D. (2003). Integrated computational and experimental approach for lead optimization and design of compstatin variants with improved activity. *Society J Am Chem Soc Society* **125**, 8422–8423.

Klosowski, J.T., Held, M., Mitchell, J.S., Sowizral, H., and Zikan, K. (1998). Efficient collision detection using bounding volume hierarchies of k-dops. *IEEE T Vis Comput GR* **4**, 21–36.

Knowles, J.R. (1987). Tinkering with enzymes: What are we learning? *Science* **236**, 1252–1262.

Kono, H., Rotamer libraries for molecular modeling and design of proteins. In S.J.Park, and J.R. Cochran, editors, *in Protein Eng and Design* (2009).

Krivov, G.G., Shapovalov, M.V., and Dunbrack Jr., R.L. (2009). Improved prediction of protein side-chain conformations with SCWRL4. *Proteins* **77**, 778–795.

Kuhlman, B., and Baker, D. (2000). Native protein sequences are close to optimal for their structures. *P Natl Acad Sci USA* **97**, 10383–10388.

Kussell, E., Shimada, J., and Shakhnovich, E.I. (2001). Excluded volume in protein side-chain packing. *J Mol Biol* **311**, 183–193.

Kuszewski, J., Qin, J., Gronenborn, A.M., and Clore, G.M. (1995). The impact of direct refinement against $^{13}C^{\alpha}$ and $^{13}C^{\beta}$ chemical shifts on protein structure determination by nmr. *J Magn Reson Ser B* **106**, 92–96.

Lauck, F., Smith, C.A., Friedland, G.F., Humphris, E.L., and Kortemme, T. (2010). Rosettabackrub-a web server for flexible backbone protein structure modeling and design. *Nucleic Acids Res* **38**, W569–W575.

Lazar, G.A., Marshall, S.A., Plecs, J.J., Mayo, S.L., and Desjarlais, J.R. (2003).

Designing proteins for therapeutic applications. *Curr Opin Struc Biol* **13**, 513–518.

Leaver-Fay, A., Tyka, M., Lewis, S.M., Lange, O.F., Thompson, J., Jacak, R., Kaufman, K., Renfrew, P.D., Smith, C.A., Sheffler, W., Davis, I., Cooper, S., Treuille, A., Mandell, D.J., Richter, F., et al. (2011). Rosetta3: An object-oriented software suite for the simulation and design of macromolecules. *Methods Enzymol* **487**, 545–74.

Lee, C., and Subbiah, S. (1991). Prediction of protein side-chain conformation by packing optimization. *J Mol Biol* **217**, 373–388.

Lesk, A.M., and Chothia, C. (1982). Evolution of proteins formed by *beta*-sheets: Ii. the core of the immunoglobulin domains. *J Mol Biol* **160**, 325–342.

Lippow, S.M., and Tidor, B. (2007). Progress in computational protein design. *Curr Opin Biotech* **18**, 305–311.

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

Looger, L.L., Dwyer, M.A., Smith, J.J., and Hellinga, H.W. (2003). Computational design of receptor and sensor proteins with novel functions. *Nature* **423**, 185–190.

Looger, L.L., and Hellinga, H.W. (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.

Lutz, S. (2010). Beyond directed evolution-semi-rational Protein Eng and design. *Curr Opin Biotech* **21**, 734–743.

Malakauskas, S.M., and Mayo, S.L. (1998). Design, structure and stability of a hyperthermophilic protein variant. *Nat Struct Biol* **5**, 470–475.

Mandell, D.J., and Kortemme, T. (2009). Backbone flexibility in computational protein design. *Curr Opin Biotech* **20**, 420–428.

Maranas, C.D., and Floudas, C.A. (1994). Global minimum potential energy conformations of small molecules. *J Global Optim* **4**, 135–170.

Mas, M.T., Chen, C.Y., Hrrzeman, R.A., and Riggs, A.D. (1986). Active human-yeast chimeric phosphoglycerate kinases engineered by domain interchange. *Science* **233**, 788–790.

McGregor, M.J., Islam, S.A., and Sternberg, M.J. (1987). Analysis of the relationship between side-chain conformation and secondary structure in globular proteins. *J Mol Biol* **198**, 295–310.

Mendes, J., Guerois, R., and Serrano, L. (2002). Energy estimation in protein design. *Curr Opin Struc Biol* **12**, 441–446.

Miao, Z., Cao, Y., and Jiang, T. (2011). RASP: rapid modeling of protein side chain conformations. *Bioinformatics* **27**, 3117–3122.

Morozov, A.V., Havranek, J.J., Baker, D., and Siggia, E.D. (2005). Protein-dna binding specificity predictions with structural models. *Nucleic Acids Res* **33**, 5781–5798.

Pan, S.J., Cheung, W.L., Fung, H.K., A.Floudas, C., and Link, A.J. (2011). Computational design of the lasso peptide antibiotic microcin J25. *Protein Eng Des Sel* **24**, 275–282.

Pantazes, R., and Maranas, C. (2010). OptCDR: A general computational method for the design of antibody complementarity determining regions for targeted epitope binding. *Protein Eng Des Sel* **23**, 849–858.

Park, S., Stowell, X.F., Wang, W., Yang, X., and Saven, J.G. (2004). Computational protein design and discovery. *Annual Reports Section C (Physical Chemistry)* **100**, 195–236.

Petrella, R.J., Lazaridis, T., and Karplus, M. (1998). Protein sidechain conformer prediction: a test of the energy function. *Fold Des* **3**, 353–377.

Petrey, D., Xiang, Z., Tang, C.L., Xie, L., Gimpelev, M., Mitros, T., Soto, C.S., Goldsmith-Fischman, S., Kernytsky, A., Schlessinger, A., Koh, I.Y., Alexov, E., and Honig, B. (2003). Using multiple structure alignments, fast model building, and energetic analysis in fold recognition and homology modeling. *Proteins* **53**, 430–435.

Pierce, N.A., Spriet, J.A., Desmet, J., and Mayo, S.L. (2000). Conformational splitting: A more powerful criterion for dead-end elimination. *J Comput Chem* **21**, 999–1009.

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

Pokala, N., and Handel, T.M. (2001). Review: Protein design - where we were, where we are, where we're going. *J Struct Biol* **134**, 269–281.

Pokala, N., and Handel, T.M. (2004). Energy functions for protein design i: Efficient and accurate continuum electrostatics and solvation. *Protein Sci* **13**, 925–936.

Pokala, N., and Handel, T.M. (2005). Energy functions for protein design: Adjustment with protein-protein complex affinities, models for the unfolded state, and negative design of solubility and specificity. *J Mol Biol* **347**, 203–227.

Ponder, J.W., and Richards, F.M. (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.

Pupo, A., and Moreno, E. (2009). Do rotamer libraries reproduce the side-chain conformations of peptidic ligands from the PDB? *J Mol Graph Model* **27**, 611–619.

Ramakrishnan, C., and Ramachandran, G.N. (1965). Stereochemical criteria for polypeptide and protein chain conformations. ii. allowed conformations for a pair of peptide units. *Biophys J* **5**(6), 909–933.

Rennell, D., Bouvier, S.E., Hardy, L.W., and Poteete, A.R. (1991). Systematic mutation of bacteriophage T4 lysozyme. *J Mol Biol* **222**, 67–86.

Richardson, J.S., and Richardson, D.C. (1989). The de novo design of protein structures. *Trends Biochem Sci* **14**, 304–309.

Rohl, C.A., Strauss, C.E.M., Misura, K.M.S., and Baker, D. (2004). Protein struc- ture prediction using rosetta. *Method Enzymol* **383**, 66–93.

Röthlisberger, D., Khersonsky, O., Wollacott, A.M., Jiang, L., DeChancie, J., Betker, J., Gallaher, J.L., Althoff, E.A., Zanghellini, A., Dym, O., Albeck, S., Houk, K.N., Tawfik, D.S., and Baker, D. (2008). Kemp elimination catalysts by computational enzyme design. *Nature* **453**, 190–197.

Ryu, J., and Kim, D.S. (2013). Protein structure optimization by side-chain posi- tioning via beta-complex. *J Global Optim* **57**, 217–250.

Samudrala, R., and Moult, J. (1998a). Determinants of side chain conformational preferences in protein structures. *Protein Eng* **11**, 991–997.

Samudrala, R., and Moult, J. (1998b). A graph-theoretic algorithm for comparative modeling of protein structure. *J Mol Biol* **279**, 287–302.

Saraf, M.C., Moore, G.L., Goodey, N.M., Cao, V.Y., Benkovic, S.J., and Maranas, C.D. (2006). IPRO: An iterative computational protein library redesign and optimization procedure. *Biophys J* **90**, 4167–4180.

Schrauber, H., Eisenhaber, F., and Argos, P. (1993). Rotamers: To be or not to be? an analysis of amino acid side-chain conformations in globular proteins. *J Mol Biol* **230**, 592–612.

Scouras, A.D., and Daggett, V. (2011). The dynameomics rotamer library: Amino acid side chain conformations and dynamics from comprehensive molecular dynamics simulations in water. *Protein Sci* **20**, 341–352.

Shah, P.S., Hom, G.K., Ross, S.A., Lassila, J.K., Crowhurst, K.A., and Mayo, S.L. (2007). Full-sequence computational design and solution structure of a thermostable protein variant. *J Mol Biol* **372**, 1–6.

Shapovalov, M.V., and Dunbrack Jr., R.L. (2011). A smoothed backbone-dependent rotamer library for proteins derived from adaptive kernel density estimates and regressions. *Structure* **19**, 844–858.

Sharma, P., Mitra, A., Sharma, S., Singh, H., and Bhattacharyya, D. (2008). Quantum chemical studies of structures and binding in noncanonical RNA base pairs: The trans Watson-Crick:Watson-Crick family. *J Biomol Struct Dyn* **25**, 709–732.

Shenkin, P.S., Farid, H., and Fetrod, J.S. (1996). Prediction and evaluation of side-chain conformations for protein backbone structures. *Proteins* **26**, 323–352.

Shifman, J.M., and Mayo, S.L. (2003). Exploring the origins of binding specificity through the computational redesign of calmodulin. *P Natl Acad Sci USA* **100**, 13274–13279.

Siegel, J.B., Zanghellini, A., Lovick, H.M., Kiss, G., Lambert, A.R., St.Clair, J.L., Gallaher, J.L., Hilvert, D., Gelb, M.H., Stoddard, B.L., Houk, K.N., Michael, F.E., and Baker, D. (2010). Computational design of an enzyme catalyst for a stereoselective bimolecular diels-alder reaction. *Science* **329**, 309–313.

Simons, K.T., Kooperberg, C., Huang, E., and Baker, D. (1997). Assembly of protein tertiary structures from fragments with similar local sequences using simulated annealing and bayesian scoring functions. *J Mol Biol* **268**, 209–225.

Smadbeck, J., Peterson, M.B., Khoury, G.A., Taylor, M.S., and Floudas, C.A. (2013). Protein WISDOM : A workbench for in silico de novo design of biomolecules. *Journal of Visualized Experiments* **77**.

Smith, C.A., and Kortemme, T. (2008). Backrub-like backbone simulation recapitulates natural protein conformational variability and improves mutant side-chain prediction. *J Mol Biol* **380**, 742–756.

Tamamis, P., Pierou, P., Mytidou, C., Floudas, C.A., Morikis, D., and Archontis, G. (2011). Design of a modified mouse protein with ligand binding properties of its human analog by molecular dynamics simulations: The case of C3 inhibition by compstatin. *Proteins* **79**, 3166–3179.

Taniguchi, N., On the basic concept of nano-technology. In *Proc. International Conference Production Engineering Tokyo, Part II, Japan Society of Precision Engineering* (1974).

Tuffery, P., Etchebest, C., Hazout, S., and Lavery, R. (1991). A new approach to the rapid determination of protein side chain conformations. *J Biomol Struct Dyn* **8**, 1267–1289.

Tuffery, P., Etchebest, C., Hazout, S., and Lavery, R. (1993). A critical comparison of search algorithms applied to the optimization of protein side-chain conformations. *J Comput Chem* **14**, 790–798.

Ulge, U.Y., Baker, D.A., and Monnat Jr., R.J. (2011). Comprehensive computational design of mCrel homing endonuclease cleavage specificity for genome engineering. *Nucleic Acids Res* **39**, 4330–4339.

Verma, R., Schwaneberg, U., and Roccatano, D. (2012). Computer-aided protein directed evolution: a review of web servers, databases and other computational tools for Protein Eng. *Computational And Structural Biotechnology Journal* **2**.

de Victoria, A.L., Gorham Jr., R.D., Bellows-Peterson, M.L., Ling, J., Lo, D.D., Floudas, C.A., and Morikis, D. (2011). A new generation of potent complement inhibitors of the compstatin family. *Chem Biol Drug Des* **77**, 431–440.

Vizcarra, C.L., and Mayo, S.L. (2005). Electrostatics in computational protein design. *Curr Opin Chem Biol* **9**, 622–626.

Wang, C., Schueler-furman, O., and Baker, D. (2005). Improved side-chain modeling for protein-protein docking. *Protein Sci* **14**, 1328–1339.

Wong, T.S., Roccatano, D., and Schwaneberg, U. (2007). Steering directed protein evolution: Strategies to manage combinatorial complexity of mutant libraries. *Environ Microbiol* **9**, 2645–2659.

Xiang, Z., and Honig, B. (2001). Extending the accuracy limits of prediction for side-chain conformations. *J Mol Biol* **311**, 421–430.

Xie, W., and Sahinidis, N.V. (2006). Residue-rotamer-reduction algorithm for the protein side-chain conformation problem. *Bioinformatics* **22**, 188–194.

Xu, J. (2005). Rapid protein side-chain packing via tree decomposition. *Research in Computational Molecular Biology, Lect Notes Comput SC* **LNBI 3500**, 423–439.

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

Yin, H., Slusky, J.S., Berger, B.W., Walters, R.S., Vilaire, G., Litvinov, R.I., Lear, J.D., Caputo, G.A., Bennett, J.S., and DeGrado, W.F. (2007a). Computational design of peptides that target transmembrane helices. *Science* **315**, 1817–1822.

Yin, S., Ding, F., and Dokholyan, N.V. (2007b). Modeling backbone flexibility improves protein stability estimation. *Structure* **15**, 1567–1576.

Zhu, Y. (2007). Mixed-integer linear programming algorithm for a computational protein design problem. *Ind Eng Chem Res* **46**, 839–845.