BIOINFORMATICS ORIGINAL PAPER

Vol. 21 no. 7 2005, pages 1028–1036
doi:10.1093/bioinformatics/bti144

Structural bioinformatics

Solving and analyzing side-chain positioning problems using linear and integer programming

Carleton L. Kingsford, Bernard Chazelle and Mona Singh*

Department of Computer Science and the Lewis-Sigler Institute for Integrative Genomics,
Princeton University, Princeton, NJ 08544, USA

Received on August 1, 2004; revised on October 10, 2004; accepted on November 8, 2004
Advance Access publication November 16, 2004

ABSTRACT

Motivation: Side-chain positioning is a central component of homology modeling and protein design. In a common formulation of the problem, the backbone is fixed, side-chain conformations come from a rotamer library, and a pairwise energy function is optimized. It is NP-complete to find even a reasonable approximate solution to this problem. We seek to put this hardness result into practical context.

Results: We present an integer linear programming (ILP) formulation of side-chain positioning that allows us to tackle large problem sizes. We relax the integrality constraint to give a polynomial-time linear programming (LP) heuristic. We apply LP to position side chains on native and homologous backbones and to choose side chains for protein design. Surprisingly, when positioning side chains on native and homologous backbones, optimal solutions using a simple, biologically relevant energy function can usually be found using LP. On the other hand, the design problem often cannot be solved using LP directly; however, optimal solutions for large instances can still be found using the computationally more expensive ILP procedure. While different energy functions also affect the difficulty of the problem, the LP/ILP approach is able to find optimal solutions. Our analysis is the first large-scale demonstration that LP-based approaches are highly effective in finding optimal (and successive near-optimal) solutions for the side-chain positioning problem.

Availability: The source code for generating the ILP given a file of pairwise energies between rotamers is available online at http://compbio.cs.princeton.edu/scplp

Contact: msingh@cs.princeton.edu

INTRODUCTION

Side-chain positioning (SCP) is a key step in computational methods for predicting and [designing protein structures (e.g. see Summers and Karplus, 1989; Holm and Sander, 1991; Lee and Subbiah, 1991; Ventura and Serrano, 2004; Park *et al*., 2004). A widely studied formulation of the problem assumes a fixed backbone, a pairwise energy function, and a set of possible rotamer choices (Ponder and Richards, 1987; Dunbrack and Karplus, 1993) for each C_α position on the backbone. The goal is to choose a rotamer for each position so that the total energy of the molecule is minimized. This formulation of SCP has been the basis of some of the more successful methods for homology modeling (e.g. Petrey *et al*., 2003; Xiang and Honig, 2001; Jones and Kleywegt, 1999; Bower *et al*., 1997) and protein design (e.g. Dahiyat and Mayo, 1997; Malakauskas and Mayo, 1998; Looger *et al*., 2003). In homology modeling, the goal is to predict the structure for a protein that is homologous to another of known structure; in that case the rotamers considered at each position correspond to a single amino acid. In protein design, the goal is to find a sequence of amino acids that will fold into a particular backbone, and so the rotamers at each position come from several possible amino acids. Although the goal of SCP is very different in these two cases, the underlying formulation for both problems is identical.

SCP methods are commonly evaluated using two scales. In the predictive scale, one asks how well the side-chain conformations predicted by the method agree with those that are found in the actual structure; or, in the case of protein design, whether the newly designed sequence folds into the desired shape. In the combinatorial scale, one asks how close the total energy resulting from the predicted side-chain conformations is to the lowest possible minimum energy using the given rotamer library and energy function. Of course, the predictive scale measures what we are ultimately interested in (i.e. the quality of the end result). However, the combinatorial scale is useful for improving search algorithms and energy functions, and such improvements are necessary to get higher-quality predictions of side-chain conformations. Theoretical results argue that the SCP problem is difficult on the combinatorial scale: the mathematical problem underlying SCP is not just NP-complete (Pierce and Winfree, 2002), but also inapproximable (Chazelle *et al*., 2004). That is, it is unlikely that there exists a polynomial-time method that can guarantee a good (let alone optimal) solution to SCP for all instances of the problem. However, these are worst-case results: they may not hold for the classes of problems and energy functions that occur in practice. In this paper, we hope to put these theoretical hardness results into practical context.

We present an integer linear programming (ILP) formulation of SCP. Though derived independently, our formulation is similar to previous ILP formulations for SCP (Althaus *et al*., 2000; Eriksson *et al*., 2001) and related problems (Klepeis *et al*., 2003). Our new formulation can tackle larger problem sizes and can obtain successive near-optimal solutions. Multiple near-optimal solutions are especially useful in protein design, where it may be desirable to find several possible sequences for a particular shape. By relaxing the integrality constraint, we get a polynomial-time linear programming (LP) heuristic. Our LP/ILP approach for SCP is as follows.

*To whom correspondence should be addressed.

First, we apply LP to an instance of SCP. If the solution using LP is integral, then that solution is provably the conformation with the global minimum energy, and we have found, in polynomial-time, the optimal solution to the SCP instance. On the other hand, if the solution using LP is fractional, we run the computationally more expensive (i.e. no longer polynomial-time) ILP procedure to find the optimal SCP solution.

Using our LP/ILP approach, we evaluate SCP instances arising on native and homologous backbones, and when choosing side chains for protein design. We show that, LP and ILP are highly effective methods for obtaining optimal solutions for the SCP problem. The LP/ILP approach is shown to tackle problems of size up to 10^{218} easily when packing side chains on native and homologous backbones, and of size up to 10^{201} when redesigning protein cores. As proof of principle, we also obtain multiple (100) near-optimal solutions for a native-backbone SCP problem of size 10^{79}.

We can use our LP formulation to probe the difficulty of SCP instances arising in different applications. We label an instance as 'easy' if LP finds an integral (i.e. optimal) solution. In contrast, if LP finds a fractional solution, we use it as evidence that the instance is more difficult to solve. Our computational experiments on 25 native-backbone problems and 33 homology-modeling problems show that LP can almost always find an integral solution when using an energy function based on van der Waals interactions and a statistical rotamer self-energy term. Similar, even simpler, energy functions have been the basis for successful homology-modeling packages (Bower *et al*., 1997). Since SCP is NP-complete, it is intriguing that integral solutions are found so readily, and in these cases, because a polynomial-time procedure has provably found optimal solutions, it appears that the theoretical hardness results do not apply in practice. On the other hand, when using the same energy function on 25 protein design problems of approximately the same size, the LP does not often find integral solutions. This suggests that the optimization problems underlying protein design may be considerably more difficult to solve than those arising in the native- or homology-modeling settings. We also explore how changing the energy functions affects the problem's hardness. The LP approach sometimes finds optimal solutions under energy function variants; however, different energy functions affect its ability to do so.

Further previous work. Perhaps foreshadowing the idea that the theoretical hardness results for SCP do not always apply in practice is the considerable progress in the development of both exhaustive and heuristic techniques for this problem. Within the past dozen years, a series of papers on dead-end elimination have given rules for throwing out rotamers that cannot possibly be in the optimal solution (e.g. Desmet *et al*., 1992, 1994; Goldstein, 1994; Lasters *et al*., 1995; Gordon and Mayo, 1998; Looger and Hellinga, 2001; Gordon *et al*., 2002). Special-purpose heuristic search techniques for specific energy functions have been successfully applied, as in the original Scwrl package (Bower *et al*., 1997), and more general search methods such as simulated annealing (e.g. Lee and Subbiah, 1991; Holm and Sander, 1991), A* (Leach and Lemon, 1998), Monte Carlo search (e.g. Xiang and Honig, 2001) and mean-field optimization (Lee, 1994) have also been used. Specialized graph-theoretic approaches have also been developed (Samudrala and Moult, 1998; Canutescu *et al*., 2003; Bahadur *et al*., 2004). Of these previous methods, the exhaustive methods always find the optimal solution but are not efficient (i.e. may require exponential search), whereas the heuristics are efficient but do not guarantee finding the optimal solution. In contrast, our LP formulation is efficient, and when it finds an optimal solution, this is evident through integrality; however, it is not guaranteed to find such a solution (when it does not, ILP is applied). Although different mathematical programming approaches to SCP (Althaus *et al*., 2000; Eriksson *et al*., 2001; Chazelle *et al*., 2004) have been suggested previously, this is the first time such an approach has been extensively tested.

Biological relevance. While our primary goal is to study the combinatorial nature of SCP, in order to verify that the energy functions considered are appropriate for predicting protein structures for native and homologous backbones, we compare side-chain conformations predicted by the LP/ILP approach with those in the native structures. The solutions found for native and homologous backbones give structures that are comparable in quality to those found by other methods using the same rotamer library (Bower *et al*., 1997; Xiang and Honig, 2001).

Practical implications. There are several immediate practical consequences of our analysis. First, our work argues that attempts to improve search methods should be focused on protein design problems, as they seem to be computationally more difficult to solve than homology modeling problems. Second, in our experience, even seemingly small differences in problem instances can have a large impact on the ease with which solutions are obtained. This makes it hard to compare different published benchmarks of SCP algorithms, as these algorithms are often tested with differing energy functions and in different settings (e.g. design versus homology modeling). To facilitate comparisons and to encourage the use of LP/ILP approaches, we are making our software for generating the LP/ILP publicly available. Third, our analysis suggests that the choice of an energy function should depend on two factors: how biologically meaningful it is and how it affects the ease with which optimal or near-optimal solutions are found. For example, a combinatorially 'easy' energy function may be useful in finding a subset of reasonable predictions that can then be evaluated using the desired energy function. Finally, and most importantly, our analysis includes the first large-scale test of an LP/ILP approach, and we demonstrate that such an approach provides an effective and practical technique for solving the SCP problem for both homology modeling and protein design applications. Because there has been decades of research on LP, we can exploit highly developed machinery; the advantage of relying on this off-the-shelf technology is that any subsequent progress in optimizing linear programs will translate into faster running times for our method. While there are many fast heuristics for SCP, in many cases, optimal and successive near-optimal solutions are desired. In these cases, LP-based approaches provide a general, state-of-the-art methodology.

METHODS

Problem formulation

The SCP problem can be stated as follows (Desmet *et al*., 1992): given a fixed backbone of length p, each residue position i is associated with a set of possible candidate rotamers $\{i_{\mathrm{r}}\}$. Once a single rotamer for each residue position has been chosen, the potential energy of a protein system is given by the formula $\mathcal{E} = E_0 + \sum_i E(i_r) + \sum_{i<j} E(i_r j_s)$, where E_0 is the self-energy of the backbone, $E(i_r)$ is the energy resulting from the interaction between

the backbone and the chosen rotamer i_r at position i as well as the intrinsic energy of rotamer i_r, and $E(i_r j_s)$ accounts for the pairwise interaction energy between chosen rotamers i_r and j_s. In this discretized setting, the placement of each side chain is reduced to finding an assignment of rotamers to positions that minimizes the overall energy of the system (the global minimum energy conformation).

It is convenient to reformulate the SCP problem in graph-theoretic terms. Let G be an undirected p-partite graph with node set $V_1 \cup \cdots \cup V_p$, where V_i includes a node u for each rotamer i_r at position i; the V_i's may have varying sizes. Each node u of V_i is assigned a weight $E_{uu} = E(i_r)$; each pair of nodes $u \in V_i$ and $v \in V_j$ $(i \neq j)$, corresponding to rotamers i_r and j_s respectively, is joined by an edge with a weight of $E_{uv} = E(i_r j_s)$. Zero-weight edges can be thought of as equivalent to the absence of an edge. The global minimum energy conformation is achieved by picking one node per V_i to minimize the weight of the induced subgraph.

Integer linear programming formulation

We first formulate the SCP problem as an ILP, so that a solution to the ILP gives an optimal solution to the SCP problem. The ILP is based on the graph formulation of SCP discussed above. The vertex set of this graph is $V = V_1 \cup \cdots \cup V_p$, and its edge set $D = \{(u, v) : u \in V_i, v \in V_j, i \neq j\}$.

We introduce a $\{0, 1\}$ decision variable x_{uu} for each node u in V, and a $\{0, 1\}$ decision variable x_{uv} for each edge in D. Setting x_{uu} to 1 corresponds to choosing rotamer u, and similarly setting x_{uv} to 1 corresponds to choosing to 'pay' the energy between rotamers u and v. We constrain our optimization so that only one rotamer is chosen per residue, and so that we pay the cost for edge $\{u, v\}$ if and only if rotamers u and v are both chosen. The following integer program ensures these conditions:

$$\begin{aligned}
&\text{Minimize} \quad \mathcal{E} = \textstyle\sum_{u \in V} E_{uu} x_{uu} + \sum_{\{u,v\} \in D} E_{uv} x_{uv} \\
&\text{subject to} \\
&\quad \textstyle\sum_{u \in V_j} x_{uu} = 1 \qquad \text{for } j = 1, \ldots, p \\
&\quad \textstyle\sum_{u \in V_j} x_{uv} = x_{vv} \qquad \text{for } j = 1, \ldots, p \text{ and } v \in V \setminus V_j \\
&\quad\quad x_{uu}, x_{uv} \in \{0, 1\}.
\end{aligned} \tag{IP1}$$

The first set of constraints ensures that we choose exactly one rotamer for each residue. The second set of constraints demands that we set the edge variables x_{uv} to 1 for edges that are in the subgraph induced by the choice of rotamers: if $x_{vv} = 0$ then no adjacent edges can be chosen, and if $x_{vv} = 1$ then exactly one adjacent edge is chosen for each vertex set. This formulation is similar to the version of (Althaus *et al*., 2000) (without modifying the energies to be negative) and simpler than that of (Eriksson *et al*., 2001). Additionally, on the experimental side, Klepeis *et al*. (2003) use a similar integer programming formulation to design variants of the peptide Compstatin that are predicted to improved inhibitory activity in complement pathways. However, this is a slightly different model in which side-chain positions are not explicitly represented.

In practice, the ILP given above can have many variables and constraints that do not affect the optimization, and the system can be pruned dramatically. In particular, if all the pairwise energies between rotamers in positions i and j are non-positive, then we can remove all variables x_{uv} with $u \in V_i$ and $v \in V_j$ such that $E_{uv} = 0$, and modify the equality constraints in (IP1) that contain such an x_{uv} by removing those variables and changing '=' to '≤'. Because we are minimizing and all the energies between i and j are zero or less, this change does not affect the optimal solution. A frequent special case has zero energies between all rotamers in two positions; this corresponds to residues that are too far apart in the structure to have any rotamers that interact with each other. The more general case involves residues that are far enough apart that only a subset of their rotamers have interactions with each other.

More formally, for each V_j, let $\mathcal{N}^+(V_j)$ be the set union of the V_i for which there exists some $v \in V_i$ and $u \in V_j$ with $E_{uv} > 0$. Let D' be the set of pairs $\{u, v\}$ with $u \in V_j$ such that either $v \in \mathcal{N}^+(V_j)$, or $v \notin \mathcal{N}^+(V_j)$ but $E_{uv} < 0$. There will be edge variables x_{uv} only for pairs in D'. Our modified ILP is as follows:

$$\begin{aligned}
&\text{Minimize} \quad \mathcal{E}' = \textstyle\sum_{u \in V} E_{uu} x_{uu} + \sum_{\{u,v\} \in D'} E_{uv} x_{uv} \\
&\text{subject to} \\
&\quad \sum_{u \in V_j} x_{uu} = 1 \qquad \text{for } j = 1, \ldots, p \\
&\quad \sum_{u \in V_j} x_{uv} = x_{vv} \qquad \text{for } j = 1, \ldots, p \text{ and } v \in \mathcal{N}^+(V_j) \\
&\quad \sum_{u \in V_j : E_{uv} < 0} x_{uv} \leq x_{vv} \qquad \text{for } j = 1, \ldots, p \text{ and } v \notin \mathcal{N}^+(V_j) \\
&\quad\quad x_{uu}, x_{uv} \in \{0, 1\}
\end{aligned} \tag{IP2}$$

An inequality constraint is not included if the sum on the left-hand side is empty. The simple modification of (IP1) given in (IP2) is crucial in practice, providing in some cases an order of magnitude speed up.

Multiple solutions

Sometimes it is desirable to find several optimal and near-optimal solutions. In the present framework, the LP/ILP can be solved iteratively to find an ensemble of low-energy solutions. At iteration m, all previously discovered solutions are excluded by adding the constraints

$$\sum_{u \in S_k} x_{uu} \leq p - 1 \quad \text{for } k = 1, \ldots, m - 1 \tag{1}$$

to (IP2), where S_k contains the optimal set of rotamers found in iteration k. This requires that the new solution differs from all previous ones in at least one position. As pointed out by an anonymous reviewer, it may be desirable to obtain successive solutions that differ more from each other, and this can be accomplished by replacing $p - 1$ in (1) by $p - q$, where $1 < q \leq p$.

LP/ILP approach

The ILP formulation is as hard to solve as the original SCP problem. If we relax the integrality constraints $x_{uv} \in \{0, 1\}$ by replacing them with constraints $0 \leq x_{uv} \leq 1$ for $u, v \in V$, we obtain a linear program, which can be solved efficiently. If the optimal solution to the relaxed linear program is integral—all variables are set to either 0 or 1—then that solution is also an optimal solution to the ILP and SCP problem. So our LP/ILP approach to find optimal solutions is as follows: solve the problem of interest using the computationally easier LP formulation. If the solution returned is integral, then the problem instance was easy to solve, and we have the optimal solution to the original SCP problem. Otherwise, we run polynomial-time Goldstein dead-end elimination (DEE) (Goldstein, 1994) until no more rotamers can be eliminated and then solve the more difficult ILP.

The CPLEX package (ILOG CPLEX, 2000, http://www.ilog.com/products/cplex/) with AMPL (Fourer *et al*., 2002) was used to solve the linear and integer programs. All computation was done on a single Sparc 1200 MHz processor.

Dataset

The primary protein set (Table 1) consists of 25 proteins taken from Xiang and Honig (2001). The proteins vary in size, ranging from 50 to 221 residues with more than one possible rotamer. As in Xiang and Honig (2001), only the first chain in the Protein Data Bank (PDB) file is used for experiments.

For homology modeling, 33 homologs to the proteins of Table 1 are also used. These protein pairs share between 29 and 87% sequence identity (Table 2). Whereas for some proteins there are other more similar protein sequences present in the PDB, for evaluation purposes, the chosen homologs give a wider range of sequence identity. ClustalW (Thompson *et al*., 1994), with default settings, was used to align the protein pairs. For each pair, the protein in the original dataset was taken as the template

Table 1. The native backbone problem sizes

Prot	Len	Var	Rot	Size	Time (s)
1aac[a]	105	85	1523	79	14
1aho[a]	64	54	981	49	7
1b9o	123	112	2056	112	25
1c5e	95	71	1108	61	8
1c9o	66	53	1130	56	9
1cc7	72	66	1396	66	17
1cex[a]	197	146	2556	136	36
1cku	85	60	1093	58	10
1ctj[a]	89	61	1021	62	6
1cz9	139	111	2332	111	56
1czp	98	83	1170	75	10
1d4t	104	89	1636	84	19
1igd[a]	61	50	926	47	6
1mfm	153	118	2134	112	23
1plc	99	82	1156	73	8
1qj4	256	221	4080	218	100
1qq4	198	143	2045	121	29
1qtn	152	134	2516	132	33
1qu9	126	100	1817	94	20
1rcf	169	142	2396	139	43
1vfy	67	63	939	56	7
2pth	193	151	3077	151	68
3lzt	129	105	2074	102	28
5p21	166	144	2874	146	78
7rsa	124	109	1958	100	26

For each protein, Prot gives its PDB identifier, Len gives its length, Var indicates how many of its side chains have more than one possible rotamer and Rot gives the total number of rotamers considered. Size gives the $\log_{10}$ of the search space size. Time gives the number of seconds for the solve phase of CPLEX.

[a] The proteins were used to determine the weight of the statistical potential in the basic energy function (see text).

Table 2. The homology modeling problems and their sizes

Template/ target	Seq id	Var len	Rot	Size	Time (ILP)
1aac/1id2	62	86	1608	82	14
1aac/2b3i	29	87	1242	73	13
1aho/1dq7	50	53	719	44	4
1b9o/1f6r	75	114	1999	111	24
1c9o/1csp	82	53	1076	56	7
1c9o/1g6p	61	54	1409	60	13
1c9o/1mjc	57	52	862	48	3
1cc7/1fe4	37	62	1222	60	13
1cku/1eyt	87	61	1095	58	10
1cku/3hip	73	65	1079	59	12
1ctj/1c6r	79	64	1030	62	7
1ctj/1cyj	64	66	1291	69	10
1ctj/1f1f	46	64	1219	62	9
1czp/1doy	73	81	990	69	8
1czp/4fxc	79	81	961	70	6
1d4t/1luk	31	93	1877	91	26 (1)
1igd/1fcl	75	51	899	48	7
1igd/1mi0	78	49	723	44	3
1mfm/1b4l	54	117	1978	105	23
1mfm/1cob	80	119	1980	108	19
1mfm/1xso	65	114	1826	104	16
1plc/1byo	71	79	1131	70	7
1plc/1jxf	44	77	1093	64	8
1qj4/1e89	75	220	4154	218	120
1qq4/1hpg	34	139	1514	105	14
1qu9/1j7h	75	101	1885	97	27
1qu9/1qd9	49	104	1749	97	19 (2)
1rcf/1czh	69	140	2151	135	38
1vfy/1hyj	40	57	1060	53	9
3lzt/2mef	59	105	2320	108	52
5p21/1kao	49	147	2977	148	71
7rsa/1bsr	81	110	2242	104	41
7rsa/1rra	67	112	2111	104	42

Template gives the PDB identifier for the protein used as the template backbone, and Target gives the PDB identifer of the protein for which the structure is to be predicted. Seq id gives percentage identity between template and target protein sequences, Var len gives the number of side chains that are varied, and Rot gives the total number of rotamers considered. Size is the $\log_{10}$ of the search space size. Time is the time in seconds that CPLEX takes to solve the LP. For non-integral solutions, the time to solve the ILP is given in parenthesis.

backbone, and its sequence homolog was taken as the target protein to be predicted. If the i-th residue of the target sequence is aligned to the j-th residue of the template sequence, then rotamers corresponding to the i-th residue were considered at the j-th position in the template backbone. Any gaps in the target sequence were handled by modeling the side chains of the native residues of the template. Any gaps in the template sequence caused the corresponding residues in the target sequence to be left out of the model.

Rotamer library and structure manipulation

We used Dunbrack's backbone-dependent rotamer library (Dunbrack and Karplus, 1993). For each 10° range of ϕ, ψ backbone angles, this library has 320 rotamers, with the largest number of rotamers, 81, belonging to arginine and lysine. Backbones were held fixed, and missing backbone hydrogens were added using the BALL C++ library (Kohlbacher and Lenhof, 2000), which was also used to manipulate rotamers. All non-protein atoms were ignored. Each choice of rotamers was converted to a three-dimensional structure using the given backbone atoms and the stock side chains from (Kohlbacher and Lenhof, 2000). For all computations, the backbone, alanines and glycines were held fixed.

Dead-end elimination

In the cases where ILP was necessary, we first processed the problem instances with DEE; no such processing was performed before running the LP. We implemented the Goldstein DEE condition from (Goldstein, 1994), which says that a rotamer $u \in V_i$ can be thrown out if there is some other rotamer $v \in V_i$ such that

$$E_{uu} - E_{vv} + \sum_{j \neq i} \min_{w \in V_j} (E_{uw} - E_{vw}) > 0.$$

The rotamers u are selected in sequence starting with an arbitrary rotamer. Every possible v is tested to see if the above condition hold. This process stops when a pass through the rotamers finds none that can be removed. None of the problems considered here converge when this simple DEE process is applied.

Energy function

All the energy functions considered consist of a rotamer self-energy term and a pairwise rotamer interaction term. For the basic energy function, used for all computations unless otherwise specified, pairwise rotamer energies are computed using van der Waals interactions, and self-energies are computed using both statistical potentials and van der Waals interactions. The basic energy function is similar to that of the Scwrl package (Bower *et al*., 1997), though we use a more realistic van der Waals term.

Van der Waals interactions between rotamers The pairwise van der Waals interaction energy between rotamers u and v is the sum of the van der Waals interactions between the side-chain atoms of u and v. We use the 6–12 Lennard–Jones formulation of the van der Waals force. The parameters used in the van der Waals force are those of AMBER96 except the hydrogen radii are reduced by 50% to account for their uncertain position. As in AMBER96, for atoms separated by three bonds (1–4 pairs), van der Waals interaction parameters are reduced by half, and there is no van der Waals contribution between atoms separated by fewer than three bonds. Each atom–atom interaction is capped at 100 kcal/mol. As an optimization, the van der Waals interactions are taken to be zero at distances longer than 10 Å and residues are assumed not to interact if their C_β atoms are farther apart than 8.0 Å plus the longest possible extensions of their side chains. Any value less than 10^{-6} is considered to be 0. These approximations generally have insignificant effects on the calculated energies.

Van der Waals interactions in self-energy terms For each rotamer, the van der Waals energy is computed (as described above) between each of its atoms and all the fixed backbone atoms in the system except those corresponding to the current residue and the residues on either side of it. The self-energy also includes the van der Waals interactions with atoms in fixed residues.

Statistical self-energies For each amino acid i in a particular backbone setting, let p_{i_u} be the fraction of times amino acid i is found in rotamer u, and p_{i_0} be the fraction of times amino acid i is in its most common rotamer. These values are obtained from the rotamer library (Dunbrack and Karplus, 1993). As in (Bower *et al*., 1997), the statistical self-energy term for a particular rotamer u is given by $-\ln(p_{i_u}/p_{i_0})$, so that the more common a rotamer, the lower the energy assigned to it.

Combining the statistical self-energies with the van der Waals interactions In summing up the total energy of the system, the statistical self-energy term is weighted by a constant C that is the relative weighting of it in comparison to the physical van der Waals term. The choice of C can have a large effect on the accuracy of the solution and the ease with which it can be found. C can be thought of as the inertia for a residue to remain in a highly favored side-chain conformation. To calibrate C, five proteins of varied structure (1aac, 1aho, 1cex, 1ctj and 1igd) were selected from the test set. The LP/ILP algorithm was applied to each for values of C ranging between 0.5 and 100. Figure 1 shows the average side-chain root mean squared deviation (rmsd) over the five proteins for various values of C. It is best to set C to the smallest value that works well so as to use as much information about the specific fold as possible. $C = 10$ was taken to be a good choice.

Evaluating predicted structures

For each protein, we compare the predicted side-chain conformations with those found in its crystal structure. We use two measures of accuracy. First, we compute the percentage of χ_1 side-chain dihedral angles predicted within 20° of the native structure, and the percentage of both χ_1 and χ_2 side-chain dihedral angles predicted within 20° of native. Second, we compute the rmsd between the predicted structure and the crystal structure. When positioning side chains on native backbones, rmsd is computed between corresponding side-chain atoms only. When positioning side chains of a target protein on a homologous backbone, the native backbone of the target protein and the homologous backbone are first fit together using all the non-hydrogen atoms in both structures (McLachlan, 1982; Martin, 2001, http://www.bioinf.org.uk/software/profit), and then rmsd is computed over the side-chain atoms.

Because performance can vary greatly depending on the location of the residue in the protein, in addition to evaluating predictions over all residues, we report performance over only core residues, defined to be those that have <10% of their possible surface area exposed in the crystal structure. For each residue, exposed surface area is determined as a percentage of the surface area of the residue in isolation using the Surfv package (Nicholls *et al*., 1991).

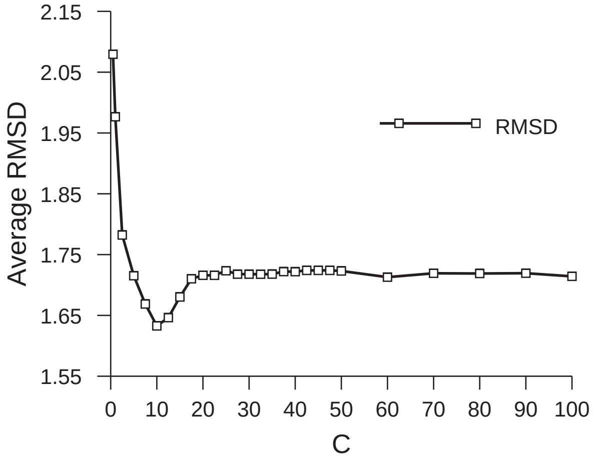

Fig. 1. The average rmsd over five proteins for various values of C.

COMPUTATIONAL RESULTS

We test the hardness of SCP instances and evaluate the LP/ILP approach on problems resulting from three applications: predicting the conformations of a protein's side chains on its native backbone, predicting the structure of a protein using the backbone of a homologous sequence as a template, and designing a protein sequence for a given backbone.

Native backbone tests

For the each of 25 proteins in Table 1, we ran the LP/ILP approach to predict side-chain conformations on native backbones. We used the native protein sequence from the PDB file and allowed each residue to assume all the rotamers listed in the library for the given amino acid and ϕ, ψ backbone angles. This resulted in search spaces with up to 10^{218} possibilities. Using the basic energy function described in the previous section, all problems returned optimal integral solutions using LP, and it was not necessary to use the more computationally expensive ILP formulation. The total CPU time for solving the 25 LPs using formulation (IP2) was under 12 min; this is approximately 13 times faster than when using the formulation (IP1).

To ensure that the energy function produces meaningful structures, we compare the side-chain conformations predicted by the LP with the side-chain conformations in the crystal structure (Table 3). Over all the residues, we find that 80% of χ_1 angles and 51% of the χ_1 and χ_2 angles are predicted within 20° of native. For just core residues, our approach leads to 87% of χ_1 angles and 62% of the χ_1 and χ_2 angles predicted correctly. Additionally, our method obtains an average rmsd per protein of 1.553 Å. This is comparable with values obtained when running the widely used Scwrl package (version 2.9) (Bower *et al*., 1997) (Table 3) and with what is reported in Xiang and Honig (2001) when using the same rotamer library (on a slightly different test set).

Overall, our testing on native backbones shows that when using a simplified energy function, LP can readily obtain optimal solutions

Table 3. Prediction of side-chain conformations on native backbones, with a comparison of the LP/ILP prediction with those of other methods and the crystal structure

	Core residues	All residues
(a) LP/ILP χ_1/χ_{1+2}	87%/62%	80%/51%
(b) Scwrl χ_1/χ_{1+2}	88%/60%	80%/49%
(c) LP/ILP rmsd	1.079 Å	1.553 Å
(d) Scwrl rmsd	1.170 Å	1.649 Å

All values are averaged over the 25 proteins of Table 1. (a) The percentage of residues over all proteins for which LP/ILP predicted conformation has the χ_1 and χ_{1+2} dihedral angles within 20° of the native structure; (b) these values for Scwrl; (c) the rmsd of the predicted side-chain conformations from those of the native side chains using the LP/ILP method; and (d) these are values for Scwrl.

with respect to the energy function, and that these optimal solutions correspond to predicted structures of quality similar to that of other popular approaches.

Homology modeling

We next explore the combinatorial problems associated with homology modeling. The 33 pairs of homologous proteins considered, their percentage sequence identity, and the rmsd between their backbones are shown in Table 2.

We solved the resulting LP formulations for all 33 problems; this took under 12 min of CPU time. The LP found optimal solutions for 31 of the 33 pairs. For only two template/target pairs, 1d4t/1luk and 1qu9/1qd9, the optimal LP solutions were not integral. For these two problems, the optimal integral solution was found using DEE and the integer programming algorithm of CPLEX. A good measure of how close the LP relaxation objective is to the optimal solution is the relative gap, defined as:

$$100\,\frac{|\mathrm{OPT} - lp|}{|\mathrm{OPT}|} \qquad (2)$$

where OPT is the energy value of the optimal integral solution and *lp* is the optimal objective for the LP relaxation. The relative gaps for both 1d4t/1luk and 1qu9/1qd9 were fairly small (0.207 and 15.260, respectively), and the total time for solving these two integer linear programs was <1 min.

In order to show that the basic energy function is useful in the homology modeling scenario, we report the accuracies of our predicted structures. We computed the side-chain rmsd between the target and predicted structures, as well as the side-chain rmsd obtained by the Scwrl rotamer choices. The average side-chain rmsd obtained by the LP/ILP approach with the basic energy function is 3.230 Å, which is competetive with Scwrl's performance of 3.260 Å when run on the same test set (Table 4).

For these tests, we did not optimize many important aspects of homology modeling, such as choosing the homolog with the most similar sequence or correcting alignments, hence the results should not be taken to be the best possible for any of the methods. However, the use of a simplified energy function results in predicted structures that are biologically reasonable. Additionally, optimal solutions with respect to this energy function are easily found using the LP/ILP approach.

Table 4. Prediction of side-chain conformations using homology modeling, with a comparison of the LP/ILP prediction with those of other methods and the crystal structure

	Core residues (Å)	All residues (Å)
(a) LP/ILP rmsd	2.177	3.230
(b) Scwrl rmsd	2.137	3.260
(c) Backbone rmsd	1.385	1.978

All values are averaged over the 33 problems of Table 2. (a) The rmsd between just side-chain atoms when comparing the LP/ILP predicted structure with the crystal structure; (b) this value when comparing the Scwrl predictions with the native structure; and (c) the rmsd between template and target structures when only considering backbone atoms.

Protein design

We considered the problem of designing novel sequences that fold into known backbones. We partitioned the amino acids into the following classes: AVILMF / HKR / DE / TQNS / WY / P / C / G. For each of the 25 proteins in our native test set (Table 1), we fixed the surface residues and the native backbone and allowed the core residues to assume any rotamer of any amino acid in the same class as the native residue. We focused on core residues since the basic energy function optimizes primarily van der Waals interactions. The sizes of the resulting problems are shown in Table 5.

When applying LP to the resulting problems with the basic energy function, only 6 out of 25 solutions had integral solutions. Thus, from the perspective of this LP, the design problem is more difficult than fitting side chains on native and homologous backbones. CPU time for solving the the 25 LP problems was approximately 20 h, with one protein (1qj4) taking ~10.5 h.

To obtain optimal solutions for the 19 proteins with non-integral solutions, we apply DEE and then run the ILP solver of CPLEX. When solving the ILP, CPLEX, in addition to using many other heuristics, solves several linear programs that are subproblems of the ILP (these subproblems are referred to as the branch-and-bound nodes). The number of such subproblems is a very rough indication of the computational effort expended by CPLEX. The number used for the design problems is shown in the '*N*' column of Table 5. For several of the problems, many branch-and-bound nodes were needed. CPLEX was able to find the optimal integral solutions to all the problems in ~138 h. Nearly all of that time (125 h) was spent on the largest problem, 1qj4; the other 18 problems took only 13 h of computation.

The best way to test a designed sequence is to make the protein and confirm its structure and/or biological properties (e.g. Dahiyat and Mayo, 1997; Harbury *et al*., 1998; Malakauskas and Mayo, 1998; Looger *et al*., 2003; Klepeis *et al*., 2003; Lilien *et al*., 2004); this is beyond the scope of this paper. However, the basic energy function is reasonable for designing protein cores as it focuses on van der Waal interactions, and the use of other energy functions is not likely to make the problem easier (see below). Thus, while the LP/ILP approach found optimal solutions for these protein design problems, our analysis shows that protein design problems are likely to be considerably more difficult to solve than homology modeling problems.

Other energy functions

We also investigated how changing the energy function affects the ability of LP to find optimal solutions. For five proteins from Table 1

Table 5. Proteins for which the core was redesigned

Prot.	Var len	Rot	Size	Time (ILP)	Rel gap	N
1aac	38	2153	62	3.3e2 (1.3e2)	2.630	4
1aho	18	668	22	4.4	Integral	
1b9o	48	1842	69	2.4e2 (9.4)	1.099	0
1c5e	25	1369	42	5.8e1	Integral	
1c9o	14	757	24	9.1e1 (4.6e1)	3.936	34
1cc7	18	866	29	9.5e1 (2.4)	0.272	0
1cex	78	3926	126	2.6e3 (7.0e2)	0.913	30
1cku	22	897	31	8.8	Integral	
1ctj	24	1262	40	2.8e1	Integral	
1cz9	53	2664	87	1.2e3 (3.2e2)	0.702	27
1czp	30	1475	47	4.4e2 (1.4e2)	1.202	39
1d4t	32	1691	52	1.8e2 (8.9e1)	1.039	33
1igd	11	552	18	3.4	Integral	
1mfm	46	3215	80	6.5e3 (5.4e3)	3.234	233
1plc	33	1691	54	4.7e2 (1.3e2)	3.991	8
1qj4	124	6655	201	3.8e4 (4.5e5)	2.677	7293
1qq4	72	3500	115	1.5e3 (6.9e2)	4.272	38
1qtn	49	2181	74	2.6e2 (7.0e1)	0.558	8
1qu9	43	2057	70	2.3e2 (6.4)	0.162	2
1rcf	65	3189	105	2.7e3 (9.6e1)	0.053	0
1vfy	15	665	20	4.0	Integral	
2pth	76	4395	127	1.1e4 (2.4e4)	2.115	1623
3lzt	48	1940	71	4.2e2 (3.9e2)	3.445	45
5p21	70	3624	114	4.1e3 (1.3e4)	2.259	1453
7rsa	46	1993	66	5.7e2 (1.4e1)	0.120	0

Var len gives the number of core positions that were allow to vary, and Rot gives the total number of rotamers considered. Size is the $\log_{10}$ of the search space size. Time is the number of seconds CPLEX spent solving the LP, and given in parentheses, the time for solving the ILP. Rel gap gives the relative gap, as defined in Equation (2), and is a measure of how far the energy of the solution of the LP is from that of the optimal rotamer choice. N gives the number of subproblems CPLEX considered in finding the optimal choice of rotamers.

(1c9o, 1czp, 1d4t, 1qtn and 1vfy), we fit side chains on their native backbones using two additional energy function variants.

In the first variant, the self-energies include the van der Waals interactions with the backbone (as before), but the statistical term is replaced by a torsion term as well as intra-side-chain van der Waals interactions. These self-energy terms are meant to measure the local favorability of a side-chain conformation. The pairwise interaction energies between rotamers consist of only van der Waals interactions.

The second variant is the same as the first, except that the self-energies include electrostatic interactions with the backbone and the pairwise energies include electrostatic interactions between side-chains. In all cases, the electrostatic interactions were modeled using the distance-dependent electrostatic component ($\epsilon = r$) of the AMBER96 force field.

In contrast to the basic energy function, for which 100% of the solutions were integral, the LP finds optimal solutions for only 60% (three out of five) of the proteins using either variant of the energy function. Thus, small changes in the energy function can influence the ease with which solutions are found. We note that ILP can still find optimal solutions for these problems, and additionally that the basic energy function gives the best accuracy over these proteins (1.634 Å average rmsd versus 2.069 and 2.409 Å for variants 1 and 2, respectively).

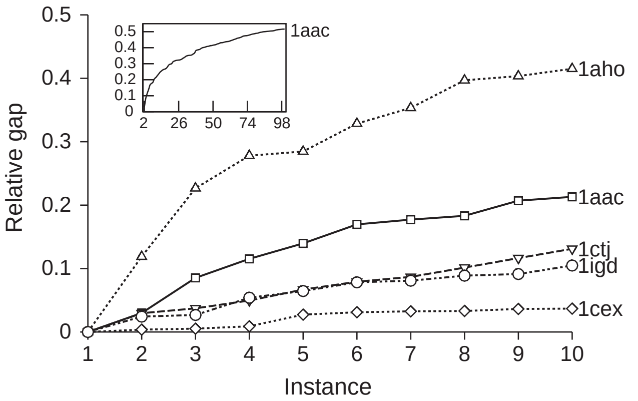

Fig. 2. Relative gap between the optimal solution (with value OPT) and the nine next lowest-energy solutions (where the i-th solution has value x_i). Inset shows relative gaps for the 100 lowest-energy solutions for 1aac. Relative gap at each iteration i is defined as $100(|\mathrm{OPT} - x_i|/|\mathrm{OPT}|)$.

Obtaining multiple solutions

By adding constraints (1) to the integer program, we can look at an ensemble of provably near-optimal solutions. Near-optimal solutions can be used to generate several candidates for protein design, as well as to analyze the energy landscape and gauge the difficulty of the global optimization problem. We found the 10 lowest-energy solutions for four proteins (1aho, 1cex, 1ctj and 1igd) and the 100 lowest-energy solutions for 1aac, using the basic energy function to fit each sequence onto its native backbone. Since at each step we are excluding all previously found solutions, each successive solution takes longer to find. The relative gap [Equation (2)] between each successive solution and the global optimum is plotted in Figure 2. These gaps are very small, and from the point of view of this energy function, any of several solutions perform similarly. This indicates that even though LP has no difficulty finding optimal solutions, no one choice of rotamers clearly stands out as the right one.

DISCUSSION

Our experiments suggest that mathematical programming should become a widely used technique for attacking SCP in the context of both homology modeling and protein design. The described approach exploits general, highly developed optimization machinery, and it is likely that problems much larger than those studied here can be solved by employing faster hardware and more effectively exploiting the CPLEX package (e.g. using parallelized versions of the software, or specifying alternate strategies for branching and node selection). The addition of valid inequalities in a branch and cut framework as in Althaus *et al.* (2000) might further speed up solution of the problems.

For even larger problems, further specialized optimizations may be necessary. As a first step, we have shown how to reduce the size of the ILP dramatically, without compromising optimality, by exploiting the fact that in protein structures amino acids do not interact with other amino acids that are far away in 3D. Furthermore, in practice, to solve large instances optimally, we would suggest first running basic DEE, and then following with either LP or ILP. We also note that some of the techniques developed for DEE can be incorporated directly into ILP if necessary. For example, we can disallow choosing a certain pair of rotamers (between positions that have some positive pairwise rotamer energy between them) by removing the corresponding edge variable from the objective function and constraints. Alternatively, the LP/ILP approach can be applied in cases where the DEE procedure does not converge to a single solution. Finally, as compared with other methods, the LP/ILP approach is simple to model and flexible enough to extend easily. For example, we have already shown how to use ILP to obtain successive near-optimal solutions.

Our analysis suggests that protein design problems are considerably more difficult to solve than homology modeling problems. For native-backbone and homology modeling, optimal, biologically realistic solutions can usually be found quickly using a simple LP relaxation. For protein design, fewer solutions of the LP relaxation are integral, even with the same energy function. As suggested by Gordon *et al*. (2002), similar-sized side-chain repacking and protein design problems have different characteristics. For repacking side chains on backbones, there are many positions with few rotamer choices, whereas for protein design, there are few positions with many rotamer choices. From a computational viewpoint, our results suggest that efforts to improve the optimization scheme should be focused on design problems.

We also find that the choice of energy function affects the ease with which optimal solutions are found using LP. For positioning side chains on native and homologous backbones, optimal solutions using the basic energy function are found quickly (typically in polynomial time), and this energy function yields good solutions (better than the other energy function variants considered in our tests). This suggests that even if alternate energy functions are required, it may be beneficial to use an energy function such as the one considered here for which optimal solutions are readily found. These solutions can then be used as starting points for an iterative procedure such as that given by Xiang and Honig (2001) or for heuristic search algorithms [e.g. as in the original Scwrl program (Bower *et al*., 1997)].

Several other authors have considered the combinatorial difficulty of SCP in the context of packing side chains onto native backbones. An excellent, exhaustive study on side-chain positioning has used very different reasoning to argue that the associated combinatorial problem appears not to be that difficult (Xiang and Honig, 2001). This study considers packing side chains on native backbones, and shows empirically that predicting the conformation of a single side chain while fixing all others in their native conformations is only slightly more accurate than the simultaneous prediction of all side chains. Unlike when integral solutions are found using our approach, their computational approach cannot guarantee that they have found a minimum energy solution according to their energy function. Eriksson *et al*. (2001) also use an ILP formulation to suggest that the SCP problem is easy; they apply the method to a single protein (lambda repressor protein) and find that the solution of the relaxed linear program always seems to be integral, even for artificial ‘nonsense’ energy functions. The hardness result (Pierce and Winfree, 2002; Chazelle *et al*., 2004) suggests this is unlikely to be true for all energy functions and proteins, and indeed the LP approach does find non-integral solutions for two of the homology modeling cases in our dataset. On the other hand, others (Gordon *et al*., 2002) have argued that it is important to consider the precise energy function being optimized; our results are consistent with this view.

In light of the hardness results (Pierce and Winfree, 2002; Chazelle *et al*., 2004), it is clear that the frequent integrality of the LP formulation in our experiments is not a result of the general structure of the problem but instead is a feature of the properties of the proteins and energy functions studied. It is well known that if the constraint matrix for an LP is totally unimodular (e.g. as in formulations for shortest path or max-flow problems), the LP has integer optimal solutions. This is not the case here, however, as changing only the energy function can change whether integral solutions are found. Nevertheless, the constraint matrices are sparse, and perhaps the LP is exploiting some other type of underlying structure. An intriguing open question is to uncover what features of side-chain positioning allow LP and ILP to find optimal solutions quickly.

ACKNOWLEDGEMENTS

The authors would like to thank Amy Keating, Jessica Fong, Gevorg Grigoryan, Elena Nabieva, Robert Osada, Elena Zaslavsky and the reviewers for insightful comments on this manuscript. M.S. thanks the NSF for PECASE grant MCB-0093399 and DARPA for grant MDA972-00-1-0031. B.C. thanks the NSF for grant CCR-998817, DARPA for ARO grant DAAH04-96-1-0181 and the NEC Research Institute.

REFERENCES

Althaus,E., Kohlbacher,O., Lenhof,H.-P. and Müller,P. (2000) A combinatorial approach to protein docking with flexible side-chains. In *Proceedings of the 4th Annual International Conference on Computational Molecular Biology*. ACM Press, New York, NY, pp. 15–24.

Bahadur,K.C.D., Akutsu,T., Tomita,E. and Seki,T. (2004) Protein side-chain packing problem: a maximum edge-weight clique algorithmic approach. In *Proceedings of the Second Conference on Asia-Pacific Bioinformatics*, Australian Computer Society Inc., Darlinghurst, Australia, pp. 191–200.

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

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

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

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

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

Desmet,J., De Maeyer,M. and Lasters,I. (1994) The ‘dead-end elimination’ theorem as a new approach to the side-chain packing problem. In Merz,K. and LeGrand,S. (eds), *The Protein Folding Problem and Tertiary Structure Prediction*, Birkhäuser, Boston, MA, USA, pp. 307–337.

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

Eriksson,O., Zhou,Y. and Elofsson,A. (2001) Side chain-positioning as an integer programming problem. In *Proceedings of 1st Workshop on Algorithms in BioInformatics*, BRICS, University of Aarhus, Denmark, pp. 129–141.

Fourer,R., Gay,D.M. and Kernighan,B.W. (2002) *AMPL A Modeling Language for Mathematical Programming*. Brooks/Cole Publishing Company, Pacific Grove, CA.

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

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

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

Harbury,P.B., Plecs,J.J., Tidor,B., Alber,T. and Kim,P.S. (1998) High-resolution protein design with backbone freedom. *Science*, **282**, 1462–1467.

Holm,L.S. and Sander,C. (1991) Database algorithm for generating protein backbone and side-chain coordinates from a Calpha trace: application to model building and detection of coordinate errors. *J. Mol. Biol.*, **218**, 183–194.

ILOG CPLEX (2000). ILOG CPLEX 7.1.

Jones,T.A. and Kleywegt,G.J. (1999) CASP3 comparative modeling evaluation. *Proteins*, **37**, 30–46.

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. *J. Am. Chem. Soc.*, **125**, 8422–8423.

Kohlbacher,O. and Lenhof,H.-P. (2000) BALL—rapid software prototyping in computational molecular biology. *Bioinformatics*, **16**, 815–824.

Lasters,I., De Maeyer,M. and Desmet,J. (1995) Enhanced dead-end elimination in the search for the global minimum energy conformation of a collection of protein side chains. *Protein Eng.*, **8**, 815–822.

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

Lee,C. (1994) Predicting protein mutant energetics by self-consistent ensemble optimization. *J. Mol. Biol.*, **236**, 918–939.

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

Lilien,R.H., Stevens,B.W., Anderson,A.C. and Donald,B.R. (2004) A novel ensemble-based scoring and search algorithm for protein redesign, and its application to modify the substrate specificity of the gramicidin synthetase a phenylalanine adenylation enzyme. In *Proceedings of the 8th Annual International Conference on Computational Molecular Biology*, ACM Press, New York, NY, pp. 46–57.

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.

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

Martin,A.C.R. (2001) Profit program version 2.2.

McLachlan,A.D. (1982) Rapid comparison of protein structures. *Acta Crystallogr.*, **A38**, 871–873.

Nicholls,A., Sharp,K.A. and Honig,B. (1991) Protein folding and association: insights from the interfacial and thermodynamic properties of hydrocarbons. *Proteins*, **11**, 281–296.

Park,S., Yang,X. and Saven,J.G. (2004) Advances in computational protein design. *Curr. Opin. Struct. Biol.*, **14**, 487–494.

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

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

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.

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

Summers,N. and Karplus,M. (1989) Construction of side-chains in homology modeling: application to the C-terminal lobe of rhizopuspepsin. *J. Mol. Biol.*, **210**, 785–811.

Thompson,J.D., Higgins,D.G. and Gibson,T.J. (1994) Clustal W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, positions-specific gap penalties and weight matrix choice. *Nucleic Acids Res.*, **22**, 4673–4680.

Ventura,S. and Serrano,L. (2004) Designing proteins from the inside out. *Proteins*, **56**, 1–10.

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