

A CLP Approach to the Protein Side-Chain Placement Problem

Martin T. Swain and Graham J.L. Kemp

Department of Computing Science, University of Aberdeen,
King's College, Aberdeen, Scotland, AB24 3UE
{mswain,gjlk}@csd.abdn.ac.uk

Abstract. Selecting conformations for side-chains is an important sub-task in building three-dimensional protein models. Side-chain placement is a difficult problem because of the large search space that has to be explored. We show that the side-chain placement problem can be expressed as a CLP program in which rotamer conformations are used as values for finite domain variables, and bad atomic clashes involving rotamers are represented as constraints. We present a new side-chain placement method that uses a series of automatically generated CLP programs to represent successively tighter side-chain packing constraints. By using these programs iteratively our method produces side-chain conformation predictions whose accuracy is comparable with that of other methods. The resulting system provides a testbed for evaluating the quality of protein models obtained using different domain enumeration heuristics and side-chain rotamer libraries.

1 Introduction

Proteins are large biomolecules that are found in all living organisms where they perform a variety of biochemical functions. For example, proteins are essential for growth, metabolism, the immune and nervous systems, and for catalysing chemical reactions. Knowing the three-dimensional structure of a protein is vital to having a full understanding of its function. Since the experimental determination of a protein's structure can be a slow and difficult process, there is a demand to be able to generate hypothetical 3D protein models.

A significant sub-task in constructing a 3D model of a protein, and the focus of this paper, is *side-chain placement*. In the following section we describe the protein side-chain placement problem, and introduce relevant biochemical concepts and terminology. In section 4 we show how the protein side-chain placement problem can be modelled as a CLP problem. In section 5 we present the results of using our method to model several proteins, and we compare results obtained with several "rotamer libraries" (see section 3) and domain enumeration heuristics. Finally we summarise the main features of our approach, and the main conclusions that can be drawn from our results.

2 Proteins and Side-Chain Modelling

Each protein contains at least one polypeptide chain, which is typically made up of hundreds of amino-acid residues; two such *residues* are shown in Figure 1. All residues have a common configuration of atoms in their *backbone* (or *main-chain*) part with a nitrogen atom (N), a central carbon atom (the alpha-carbon, or $C\alpha$) and a carboxyl group (atoms C and O). Attached to the $C\alpha$ of each residue is a *side-chain*. There are twenty different side-chain types (listed in Table 1) in natural proteins, and these differ from one another in size, charge, and various other properties.

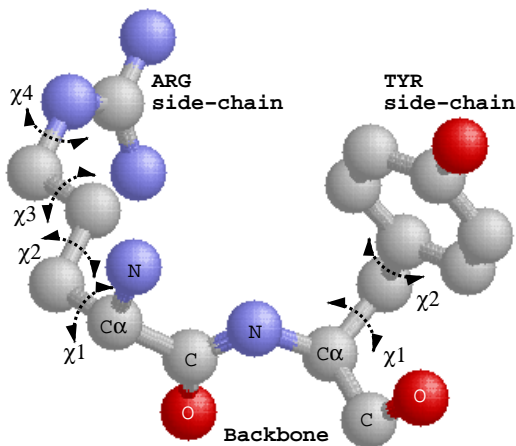


Fig. 1. The backbone and the side-chains of two residues are shown. The amino group consists of the nitrogen atom labelled N, and the carboxyl group consists of atoms C and O. The side-chains are attached to the main-chain at the $C\alpha$ atoms. The χ angles are a measurement of the “twist” around rotatable side-chain bonds (as defined in [14]). These angles define the conformation of each rotamer.

Proteins that have evolved from a common ancestor will have similar (or *homologous*) amino acid sequences. The sequence of amino acid residues in a protein chain, which is always listed from the *N-terminus* to the *C-terminus* of the protein chain, can be determined relatively easily. If two protein chains have a similar sequence of amino acid residues in their chains, then these will generally have very similar 3D structures, with amino acid residues common to both sequences usually expected to occupy the same positions in the model and the known structure. The process of constructing a model of a protein based on predicted similarity to a protein whose structure has been determined previously is called *homology modelling* [5,12,18]. Early protein modelling studies were done using physical wire models (e.g. [3]), however today protein models are usually constructed by computer.

In homology modelling it is common to first construct a model of the entire protein backbone, and then to add side-chains to this backbone, adjusting the conformations of the side-chain by rotating their internal chemical bonds (labelled χ_1 , χ_2 , etc. outward along the side-chain, away from the $C\alpha$ atom) so that no side-chain clashes with either the protein backbone or with any other side-chain. To achieve this we consider each atom to be a sphere with a radius (the *van der Waals* radius) determined by the atom's type, and we require that the centres of two atoms that are not covalently bonded to one another should be further apart than the sum of their van der Waals radii. The van der Waals radius of each atom is approximately 2.5 times larger than the radius of the spheres in ball-and-stick representation in Figure 1, and the resulting protein model is usually extremely well packed with atoms entirely filling the internal volume of the protein. Indeed, side-chain placement has been compared to a complex 3D jigsaw puzzle [27].

3 Side-Chain Rotamers

The principal problem encountered when modelling side-chains is the extremely large number of possible combinations of side-chain conformations — infinite if we consider side-chain bonds to be continuously variable. For practical purposes the search space can be discretised by considering a finite set of possible torsion angles for each side-chain. However, this can still result in an enormous search space: if we were to consider a rotational step size of 10° , then a protein with 100 amino acid residues, with 2 rotatable bonds per residue would yield a search space of $(36 \times 36)^{100}$ possible side-chain combinations [19].

The discovery that the distribution of side-chain conformations fell into statistically significant clusters [23], known as *rotamers*, has brought notable advances in side-chain modelling [24,7,10]. Rather than considering regular torsion angle increments for each rotatable bond, one can instead choose from a much smaller set of torsion angles representing the most common side-chain conformations observed in experimentally determined protein structures. Thus the vast combinatorial search space can be greatly reduced.

Rotamer libraries [23,28,8,7,21] typically contain up to 30-50 rotamers for side-chains with four rotatable bonds like Lys and Arg, and about three rotamers for side-chains with only one rotatable bond like Ser, Cys and Thr. An exception is the very extensive rotamer library developed by Dunbrack *et al.* [8,7] which has separate rotamer conformations listed for different backbone configurations. This reflects observations that have shown a correlation between residues' main-chain and side-chain conformations. The backbone dependent library (BBDEP) is based on Bayesian statistics and typically contains three angular conformations per rotatable side-chain bond, resulting in 81 rotamers for Lys and Arg.

Side-chain modelling methods typically consist of a potential energy function to describe the interactions of the side-chains represented by rotamers. This energy function plays an important role in discriminating between possible side-chain combinations since it is generally accepted that as the potential energy

is lowered the protein model becomes more accurate. In a review of side-chain modelling [30], Vasquez states that most side-chain modelling algorithms use comparatively simple energy functions. These functions, based mainly on van der Waals interactions, can give excellent results for residues in hydrophobic cores, however the results for surface and buried polar residues can be poor, and very little difference is made by adding terms for electrostatics and hydrogen bonding [30].

Various algorithms have been used to search through the energy landscape generated by the potential energy function's description of different rotamer combinations [31,11,16,9,15]. Optimisation methods include Monte Carlo algorithms [11,12], genetic algorithms [29] and dead end elimination (DEE) [6,17].

4 Expressing Side-Chain Placement as a CLP Problem

Side-chain modelling is essentially the task of searching through a large combinatorial space of possible side-chain conformations to find a mutually consistent set — a problem for which CLP is well suited. In the modelling method presented here the main determinants of side-chain conformation are:

1. avoiding atomic clashes;
2. using the most common rotamer conformation whenever possible.

Our method starts by placing the complete set of rotamers from a rotamer library onto the protein backbone as shown in Figure 2. We assume that we already have a modelled backbone, and in testing our method we consider the extreme situation in which all side-chains have to be placed.

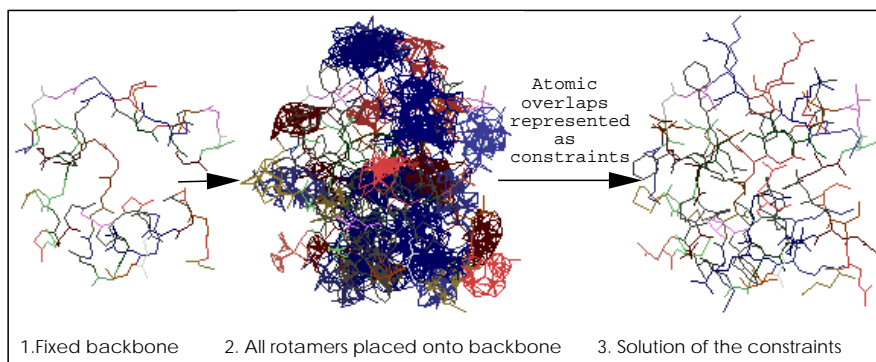


Fig. 2. The CLP method begins with a given backbone. For every residue, all rotamers in the library are placed onto the backbone. CLP selects a single rotamer for each residue.

4.1 Domain Variables

CLP domain variables are used to represent amino acid residue positions along the folded protein chain. A variable's finite domain is the set of rotamers corresponding to possible conformations of that residue's side-chain.

In the rotamer libraries used in this paper, each rotamer has a stored probability value indicating how common the conformation is for that type of residue. We have ordered the each variable's domain so that the most common rotamer conformations are at the start of the domain i.e. value 1 is the most common rotamer. Thus, the CLP solver will initially try to assign the most common rotamer to a residue.

4.2 Constraints

A C program is used to generate a constraint-based description of atomic packing by calculating clashes between rotamers and the backbone, and between rotamers of different residues. These constraints model the physical forces that prevent any clashes (or *steric overlap*) occurring between atoms i.e. no two atoms can occupy the same space. This need to avoid atomic clashes can be expressed using two types of constraint:

1. rotamers cannot be involved in any steric overlaps with the fixed backbone;
2. rotamers cannot be involved in any steric overlaps with rotamers from other residues.

Rotamers that overlap with the fixed backbone will be eliminated from their residue's finite domain. Two rotamers that overlap with each other cannot both be part of the solution, and so if one of the pair is part of the solution, then the other must be eliminated from its residue's finite domain.

To illustrate how the domain variables and constraints are expressed in SIC-Stubus Prolog syntax [4], Figure 3 shows a mini-example CLP program for a chain containing the amino acid residues Val-Lys-Tyr-Gln-Gly-Ser. When modelling full proteins, the CLP programs generated are much larger than this.

If \mathbf{r}_1 and \mathbf{r}_2 are the position vectors of the centres of two atoms, then the C program uses the following rule to decide whether those atoms overlap:

$$|\mathbf{r}_1 - \mathbf{r}_2| \times \left(1 + \frac{\Delta\text{radius}}{2.5}\right) < \text{ConDist}$$

Here, ConDist is the minimum allowed interatomic distance, and Δradius is the difference in the atomic radii of the atoms being considered: all carbon and sulphur atoms are assumed to have the same radius, nitrogen atoms are 0.2Å smaller, and oxygen atoms are another 0.2Å smaller. Thus if a carbon and an oxygen atom are being considered, $\Delta\text{radius} = (0.0 + 0.4)$.

```

%
% Predicate to solve the constraints using
% the most constrained [ffc] heuristic.
%
solve_constraints :-
    constraints(ResiduePositions),
    labeling([ffc], ResiduePositions).

%
% SICStus Prolog CLP syntax for
% constraints on side-chain placement
%
constraints([VAL1, LYS2, TYR3, GLN4, GLY5, SER6]) :-

    % Finite domains for variables, e.g.
    % VAL1 has 3 rotamers, LYS2 has 81 rotamers
    %
    VAL1 in 1..3,
    LYS2 in 1..81,
    TYR3 in 1..4,
    GLN4 in 1..10,
    GLY5 in 1..1,
    SER6 in 1..3,

    % Clashes with the backbone, e.g.
    % VAL1 cannot be rotamer 2
    %
    VAL1 #\= 2,
    TYR3 #\= 3,
    TYR3 #\= 4,
    GLN4 #\= 1,
    GLN4 #\= 5,

    % Clashes between rotamers, e.g.
    % if VAL1 is rotamer 1 then TYR3 cannot be rotamer 1
    %
    VAL1 #= 1 #=> TYR3 #\= 1,
    VAL1 #= 1 #=> LYS2 #\= 1,
    VAL1 #= 1 #=> LYS2 #\= 2,
    SER6 #= 1 #=> LYS2 #\= 2,
    SER6 #= 1 #=> GLN4 #\= 2,
    SER6 #= 1 #=> GLN4 #\= 8,
    true.

```

Fig. 3. SICStus Prolog syntax for declaring CLP variables, finite domains and constraints. Programs like this are generated automatically by our system; the typical size of such programs is from 3000 to 8000 lines.

4.3 Search Strategy Used by the CLP Solver

The SICStus Prolog `labeling` predicate shown in Figure 3 has search options to control the order in which variables are selected and assigned a rotamer. When the variables are listed in the order in which the residues occur in the protein chain, the biological interpretations of the five alternative `labeling` options are as follows:

- leftmost:** residues are selected in order from the N-terminus towards the C-terminus.
- min:** the residue closest to N-terminus with the smallest lower bound, i.e. with a constraint on its most probable rotamer, is selected first.
- max:** the residue closest to N-terminus with the greatest upper bound, i.e. with a constraint on its most improbable rotamer, is selected first.
- ff:** the residue closest to N-terminus with the least number of rotamers is selected first.
- ffc:** the residue closest to N-terminus with the least number rotamers and with the most constraints suspended on it is selected first.

A comparison of these different enumeration heuristics is given in section 5.1.

4.4 Problems with Unsatisfiable CLP Programs

An unfortunate characteristic of a single CLP side-chain placement program is that it either works, or it doesn't — if the value of `ConDist` is too high, variables will be over-constrained, and a model cannot be produced. This failure to find even a poor solution for an over-constrained system is a disadvantage of using CLP to model side-chains, since even a poor solution with known weaknesses can still provide scientists with useful information about a protein's structure, and can serve as a starting point for further structural refinement.

The advantage of using the `ConDist` parameter when calculating steric overlaps is that it can be varied easily, tightening or loosening the constraints, in order to achieve a solution. The largest value of `ConDist` that produces a solution can vary greatly between proteins. Typically, small proteins, with less than 100 residues, can be modelled with a `ConDist` of about 2.4 Å. Larger proteins, with over 200 residues, can be modelled with a `ConDist` of only 1.6 Å — a value that represents some very severe steric overlaps.

The severe steric overlaps present in the models created using CLP highlight some of the problems experienced by approximating continuous side-chain conformations by fixed rotamers. In side-chain modelling methods that use explicit energy functions such close contacts lead to very high van der Waals terms that approach infinity as the distance shrinks to zero. This has led some researchers to fix the van der Waals term to a certain value for small interatomic distances [19,12,15,13].

When `ConDist` is at a value small enough to produce a solution, the constraints on many residues will be so weak that they will be poorly modelled. To

achieve greater modelling accuracy high values of ConDist are needed to place constraints on loosely packed residues, while low values of ConDist are needed for residues that are more tightly packed. A method embracing these apparently conflicting requirements is described in section 4.6.

4.5 Null Rotamers

One method of identifying variables that are likely to be over-constrained is to use *null values* (or *null rotamers*). In doing this, we add an extra value to the end of each variable's finite domain, after the least common rotamer, that corresponds to "no (real) value found". When this value is part of the solution it means that no rotamer can be placed for the corresponding residue. Because the null rotamer has no physical representation, no constraints can be placed upon it. No matter how tight the constraints on a variable may be, there will always be a solution that contains the null rotamer. Thus, under very tightly constrained conditions, the residues in the core of the protein may be over constrained and allocated null rotamers, whereas those under-constrained residues found towards the surface of the protein will be allocated real rotamers.

4.6 An Iterative Implementation of the CLP Method

The simple CLP method for side-chain placement described above has been modified to make use of null rotamers. The basic idea is that ConDist is increased iteratively from zero to around 3.2 Å in steps of 0.4 Å so that at each iteration a CLP program is created with successively tighter packing constraints.

When ConDist is low a solution will be found easily, and this solution is stored for later use. As ConDist is increased to relatively high values, residues will become over-constrained and cause the CLP solver to fail. When this happens the CLP program is rewritten automatically, with null rotamers in the domains of all residue variables. The solution to this program will allocate null rotamers to some of the over-constrained residues. These over-constrained residues are set to the rotamer that was part of the solution solved (and stored) under the previous iteration. They are now considered to be fixed like the backbone, and do not take part in any constraints. Now another CLP program is created without null rotamers. If this program fails then null rotamers are used again, and the process repeated, until a solution is found for a program that does not use null rotamers. When this occurs the rotamers chosen are stored, and ConDist is increased.

The CLP solver will not backtrack when null rotamers are used. This is not a serious problem because our method only uses null rotamers to identify over-constrained residues. Once all the residues causing the CLP solver to fail have been identified, null rotamers are no longer used and the CLP solver is once more able to backtrack.

An outline of the improved CLP side-chain placement algorithm is shown in Figure 4.


```
find close inter-atomic distances between rotamers
ConDist = 0

while ConDist < 3.2 Angstroms

    turn off null rotamers
    automatically write CLP program for current value of ConDist
    try to solve constraints with CLP

    if no solution exists

        turn on null rotamers
        automatically write CLP program for current value of ConDist
        solve constraints with CLP (success guaranteed)
        relace any null rotamers previously recorded solution

    else CLP found a solution

        store the set of rotamers that is the solution
        evaluate model
        ConDist := ConDist + 0.4

    end else

end while
```

Fig. 4. Pseudo-code description of the iterative CLP side-chain placement algorithm.

5 Results and Discussion

We have used the iterative CLP method described in section 4.6 to investigate the accuracy of models constructed using different enumeration options and rotamer libraries, and we have compared the accuracy of our CLP method with other side-chain placement algorithms. In this study we have modelled a set of forty-three proteins¹ collated from those modelled by Keohl and Delarue [15], Shenkin *et al.* [25], and Holm and Sander [12]. All of these structures are high quality, with a resolution value less than or equal to 2.0 Å. Comparing side-chain modelling methods is complicated by the different criteria used by authors to assess the accuracy of their predictions [25,26]. Predicted side-chain conformations are commonly compared to the X-ray structures obtained from the Protein Data

¹ The Protein Data Bank [1] codes of the proteins are: 1BP2, 1CA2, 1CCR, 1CRN, 1CTF, 1HOE, 1LZ1, 1MBA, 1PAZ, 1PPD, 1PPT, 1R69, 1RDG, 1UBQ, 256B 2CAB, 2CDV, 2CGA, 2CI2, 2CTS, 2IIB, 2LYZ, 2LZT, 2MLT, 2OVO, 2RHE, 2UTG, 3APP, 3GRS, 3LZM, 4HHB, 4LYZ, 4PEP, 4PTI, 4TNC, 5CYT, 5PCY, 5PTI, 5RXN, 6LDH, 6LYZ, 7RSA, 8DFR. The number of amino acid residues in these proteins ranges from 36 to 574, with an average of 165.

Bank [1] by calculating the root mean square distance (RMSD) of the side-chain atoms (excluding hydrogens), or by comparing side-chain dihedral (χ) angles (as defined in [14]).

5.1 Enumeration Option Comparison

Figure 5 shows the accuracy of the modelling method when using the different CLP variable enumeration heuristics. Both the `ffc` and `ff` search options perform very similarly. The `leftmost` and `min` options also give similar predictions to each other, and the `max` is the least successful. We have used the `ffc` heuristic to obtain the results given in this paper.

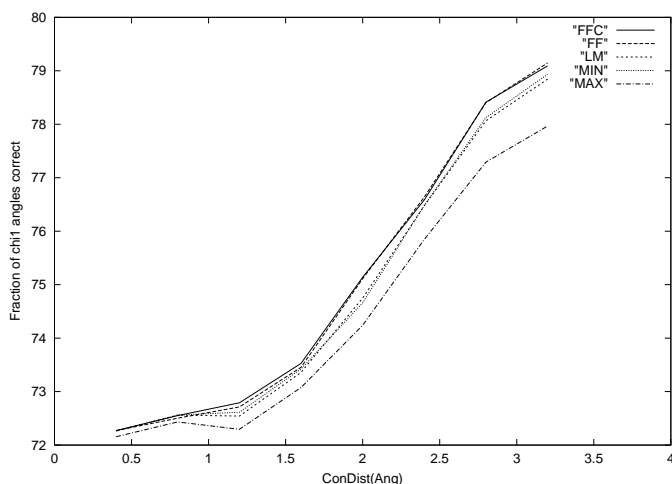


Fig. 5. The average side-chain atom RMSD, including $C\beta$ atoms, of forty-three models built using the CLP method, with five different enumeration heuristics, and ConDist parameter increasing from 0.4 Å to 3.2 Å.

We believe that the first fail options place the smaller side-chains first and, having determined those conformations, they propagate constraints onto the larger, more flexible side-chains. Thus the larger side-chains are packed around the smaller ones. This is the opposite to what happens when the `max` option is used: more flexible side-chains are placed first, and propagated constraints eliminate the most common conformations for small side chains with fewer rotamers.

The results presented in Figure 5 were obtained using the CULL2 library, described in Section 5.2. Tests with other rotamer libraries give similar results.

5.2 Reducing the Variables' Domain Sizes

Since each variable's domain has been ordered with the most common rotamers first, the CLP solver will try to find a solution with these rotamers before it tries the less common rotamers. In the BBDEP library rotamers are included for every region of backbone torsion space [7]. Some of these rotamers are very uncommon, have large internal clashes, and are unlikely to be genuinely observed [21]. When the rotamers of two residues are involved in a clash, the CLP solver will backtrack through the least probable rotamers of the first residue before trying a different, more common conformation of the second residue. By culling very improbable rotamers from the rotamer library we build models with only the most common rotamers. Although the maximum theoretical accuracy of the rotamer libraries has decreased because they contain fewer side-chain conformations (the complete BBDEP library covers 97% of χ_1 conformations, whereas the CULL1 version covers 93%), the accuracy of the models created tends to increase, as is shown in Figures 6 and 7.

In Figures 6 and 7 we compare the accuracy of the CLP method when used with different rotamer libraries, and show that reducing the size of the rotamer library can lead to more accurate side-chain placement. These libraries have been created by removing all rotamers with probabilities less than a certain minimum value.² In addition we added some extra rotamers to CULL1 and CULL2. For these libraries, CULL1X and CULL2X, rotamers with χ_2 angles differing by $\pm 10^\circ$ were added. These extra rotamers are intended to alleviate the slight steric overlaps that the most common rotamers may be involved in although, as can be seen in Figures 6 and 7, the gain in accuracy is relatively small.

5.3 Comparisons with Other Side-Chain Prediction Methods

In Table 1 we show the modelling predictions for all residues with one or more rotatable side-chain bond in the set of 43 proteins, and compare our results with those obtained using SCWRL [2] and confmat [15]. Implementations of SCWRL and confmat were obtained via the web, and were tested using the same set of proteins and evaluation methods as our CLP method. Our method compares favourably with the other methods; with the CULL2 library it predicts just over 79% of χ_1 angles correctly, an improvement of about 1% over SCWRL, and 6% over confmat. The method presented here, which has not been optimised for speed, took about 60 minutes to model the set of 43 proteins — 30 minutes longer than SCWRL.

² For the smallest library, CULL1, the amino acids Cys, Pro, Ser, Thr, Val had minimum probabilities of 0.1, Asp, Asn, Ile and Leu had minimum probabilities of 0.075, Arg, Gln, Glu, Lys, Met had minimum probabilities of 0.05, and Phe, Tyr, Trp, His had minimum probabilities of 0.025. For CULL2 the minimum probabilities were 0.075, 0.05, 0.025 and 0.125 for each of the groups, and for CULL3 the minimum probabilities were 0.05, 0.025, 0.0125 and 0. The rotamers in the BBDEP library cover 97% of the χ_1 angles in our set of 43 proteins; this decreases to 96%, 95% and 93% for the CULL3, CULL2 and CULL1 libraries.

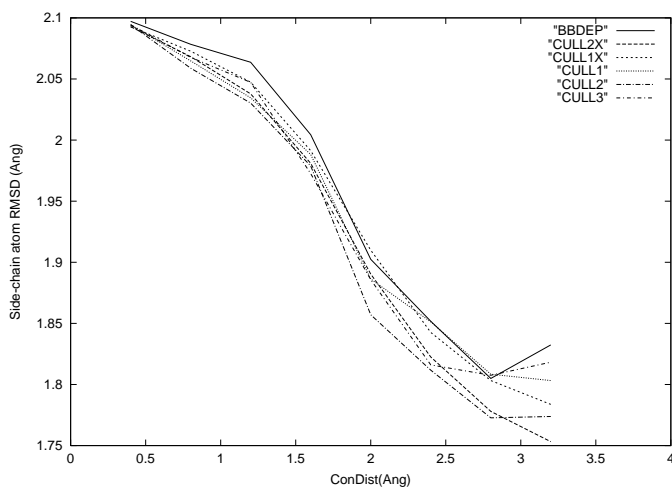


Fig. 6. The average side-chain atom RMSD, including $C\beta$ atoms, of forty-three models built using the CLP method, with ConDist parameter increasing from 0.4 Å to 3.2 Å. Each curve represents a modification, described in the main text, to the BBDEP library [7].

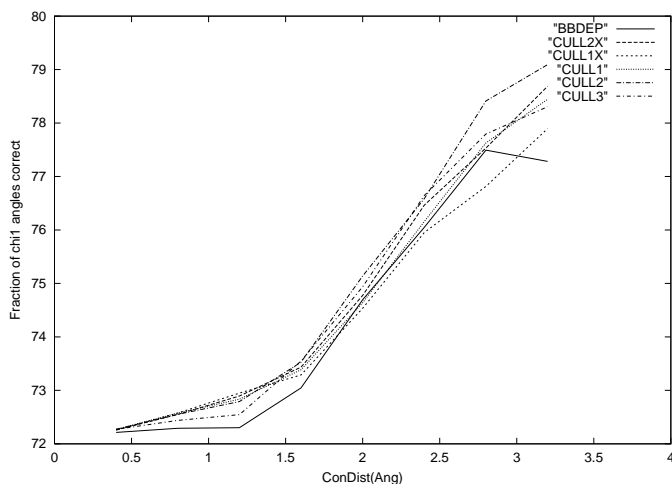


Fig. 7. The average percentage of modelled side-chains with χ_1 angles within 40° of those in the forty-three X-ray structures. The models were built using the CLP method, with ConDist parameter increasing from 0.4 Å to 3.2 Å. Each curve represents a modification, described in the main text, to the BBDEP library [7].

The theoretical limit of side-chain prediction accuracy is set by the differences in X-ray structures crystallised by different laboratories [20]. More recent side-chain prediction algorithms approach this theoretical limit, modelling up to 85%

of χ_1 conformations correctly [22,32], improving on the results of the SCWRL algorithm by up to 4%. However, these recent approaches are time-consuming, taking hours rather than minutes to model a single protein.

Table 1. The percentage of χ_1 angles correct, are shown for each residue when modelled by the CLP side-chain method using the `ffc` enumeration heuristic and different rotamer libraries. These values were taken when ConDist was equal to 2.8Å for the BBDEP library, and 3.2Å for CULL2, and CULL2X.

Residue	No. χ Angles	Number	CLP	CLP	CLP	SCWRL	confmat
			BBDEP	CULL2	CULL2X	[2]	[15]
Ala	0	-	-	-	-	-	-
Arg	4	226	64	68	67	69	67
Asn	2	333	76	75	75	74	71
Asp	2	379	80	77	77	75	66
Cys	1	170	80	84	85	76	58
Gln	3	228	70	73	74	71	73
Glu	3	306	62	60	61	62	63
Gly	0	-	-	-	-	-	-
His	2	127	78	80	81	83	84
Ile	2	303	86	91	90	88	85
Leu	2	490	81	85	85	84	84
Lys	4	432	67	67	68	67	67
Met	3	126	76	80	80	78	74
Phe	2	222	88	93	91	91	92
Pro	2	275	89	89	81	90	83
Ser	1	483	60	62	62	60	38
Thr	1	362	82	84	84	83	74
Trp	2	103	90	91	90	91	90
Tyr	2	219	92	94	94	89	93
Val	1	437	86	86	86	89	84
Overall			77.5	79.1	78.7	78.3	73.3

6 Conclusions

The side-chain placement problem can be expressed as a CLP program in which rotamer conformations are used as values for finite domain variables, and bad steric contacts involving rotamers are represented as constraints. We have described an initial CLP method of side-chain placement that is fast and accurate. Our method uses a series of automatically generated CLP programs to represent successively tighter side-chain packing constraints. By using these programs iteratively our method predicts 79% of χ_1 angles correctly.

We have presented results obtained using several different domain enumeration heuristics, and have found those based on “first fail” to be the most successful for this application. We have constructed several rotamer libraries based on

the backbone independent library of Dunbrack *et al.* [7] and our results indicate that discarding the least common rotamers from this library both improves the accuracy of the predicted side-chain conformations, and reduces the size of the combinatorial search space.

Acknowledgements. M.T.S. is supported by a BBSRC CASE award with Biovation Ltd.

References

1. F. C. Bernstein, T. F. Koetzle, G. J. B. Williams, E. F. Mayer, M. D. Bruce, J. R. Rodgers, O. Kennard, T. Shimanouchi, and M. Tasumi. The Protein Data Bank: a Computer-Based Archival File for Macromolecular Structures. *J. Mol. Biol.*, 112:535–542, 1977.
2. M. J. Bower, F. E. Cohen, and R. L. Dunbrack. Prediction of protein side-chain rotamers from a backbone-dependent rotamer library: A new homology modeling tool. *J. Mol. Biol.*, 267:1268–1282, 1997.
3. W.J. Browne, A.C.T. North, D.C. Phillips, K. Brew, T.C. Vanman, and R.L. Hill. A Possible Three-dimensional Structure of Bovine α -Lactalbumin based on that of Hen's Egg-White Lysozyme. *J. Mol. Biol.*, 42:65–86, 1969.
4. M. Carlsson, G. Ottosson, and B. Carlsson. An open-ended finite domain constraint solver. *Proc. Programming Languages: Implementations, Logics, and Programs*, 1997.
5. G. Chinaea, G. Padron, R. W. W. Hooft, C. Sander, and G. Vriend. The use of position specific rotamers in model building by homology. *Prot. Struct. Funct. Genet.*, 23(415–421), 1995.
6. J. Desmet, M. De Maeyer, B. Hazes, and I. Lasters. The dead-end elimination theorem and its use in protein side-chain positioning. *Nature*, 356:539–542, 1992.
7. R. L. Dunbrack and F. E. Cohen. Bayesian statistical analysis of side-chain rotamer preferences. *Protein Science*, 6:1661–1681, 1997.
8. R. L. Dunbrack and M. Karplus. Backbone-dependent rotamer library for proteins: application to side-chain prediction. *J. Mol. Biol.*, 230:543–574, 1993.
9. D. B. Gordon and S. L. Mayo. Branch-and-terminate: a combinatorial optimization algorithm for protein design. *Structure*, 7:1089–1098, 1999.
10. J. Heringa and P. Argos. Strain in protein structures as viewed through nonrotameric side-chain: I. their positions and interaction. *Prot. Struct. Funct. Genet.*, 37:30–43, 1999.
11. L. Holm and C. Sander. Database Algorithm for Generating Protein backbone and Side-chain Co-ordinates from a C α Trace. *J. Mol. Biol.*, 218:183–194, 1991.
12. L. Holm and C. Sander. Fast and Simple Monte Carlo Algorithm for Side Chain Optimization in Proteins: Application to Model building by Homology. *Prot. Struct. Funct. Genet.*, 14:213–233, 1992.
13. J. K. Hwang and W. F. Liao. Side-chain prediction by neural networks and simulated annealing optimization. *Prot. Eng.*, 8:363–370, 1995.
14. IUPAC-IUB Commission on Biochemical Nomenclature. Abbreviations and Symbols for the Description of the Conformation of Polypeptide Chains. *Eur. J. Biochem.*, 17:193–201, 1970.

15. P. Koehl and M. Delarue. Application of a self-consistent mean field theory to predict protein side-chains conformation and estimate their conformational entropy. *J. Mol. Biol.*, 239:249–275, 1994.
16. H. Kono and J. Doi. A new method for side-chain conformation prediction using a hopfield network and reproduced rotamers. *J. Comp. Chem.*, 17:1667–1683, 1996.
17. I. Lasters, M. De Maeyer, and J. Desmet. Enhanced dead-end elimination in the search for the global minimum energy conformation of a collection of protein side chains. *Prot. Eng.*, 8:815–822, 1995.
18. C. A. Laughton. Prediction of Protein Side-chain Conformations from Local Three-dimensional Homology Relationships. *J. Mol. Biol.*, 235:1088–1097, 1994.
19. C. Lee and S. Subbiah. Prediction of protein side-chain conformation by packing optimization. *J. Mol. Biol.*, 217:373–388, 1991.
20. M. Levitt, M. Gerstein, E. Huang, S. Subbiah, and J. Tsai. Protein folding: The endgame. *Annu. Rev. Biochem.*, 1997.
21. S. C. Lovell, M. Word, J. S. Richardson, and D. C. Richardson. The Penultimate Rotamer Library. *Prot. Struct. Funct. Genet.*, 40:389–408, 2000.
22. J. Mendes, A.M. Baptista, M.A. Carrondo, and C.M. Soares. Improved modeling of side-chains in proteins with rotamer-based methods: A flexible rotamer model. *Prot. Struct. Funct. Genet.*, 37:530–543, 1999.
23. J. W. Ponder and F. M. Richards. Tertiary templates for proteins. *J. Mol. Biol.*, 193:775–791, 1987.
24. H. Schrauber. Rotamers: to be or not to be? *J. Mol. Biol.*, 230:592–612, 1993.
25. P.S. Shenkin, H. Farid, and J.S. Fetrow. Prediction and evaluation of side-chain conformations for protein backbone structures. *Prot. Struct. Funct. Genet.*, 26:323–352, 1996.
26. M.T. Swain and G.J.L. Kemp. Modelling protein side-chain conformations using constraint logic programming. *Computers Chem.*, in press.
27. W. Taylor. New paths from dead ends. *Nature*, 356:748–479, 1992.
28. P. Tuffery, C. Etchebest, and S. Hazout. Prediction of protein side chain conformations: a study on the influence of backbone accuracy on conformation stability in the rotamer space. *Prot. Eng.*, 10:361–372, 1997.
29. P. Tuffery, C. Etchebest, S. Hazout, and R. Lavery. A new approach to the rapid determination of protein side-chain conformations. *J. Biomol. Struct. Dynam.*, 8:1267–1289, 1991.
30. M. Vasquez. Modeling side-chain conformations. *Curr. Opin. Struct. Biol.*, 6:217–221, 1996.
31. C. Wilson, L. M. Gregoret, and D. A. Agard. Modeling side-chain conformation for homologous proteins using an energy-based rotamer search. *J. Mol. Biol.*, 229:996–1006, 1993.
32. Z. Xiang and B. Honig. Extending the accuracy limits of prediction for side-chain conformations. *J. Mol. Biol.*, 2001.