BetaSCP2: A Program for the Optimal Prediction of Side-Chains in Proteins

Joonghyun Ryu[1], Mokwon Lee[1], Jehyun Cha[1], Chanyoung Song[1], and Deok-Soo Kim[2,⋆]

[1] Voronoi Diagram Research Center, Hanyang Univeristy, South Korea
[2] Hanyang Univeristy, South Korea
dskim@hanyang.ac.kr
http://voronoi.hanyang.ac.kr

Abstract. The *side-chain prediction problem* (SCP-problem), is a computational problem to predict the optimal structure of proteins by finding the optimal dihedral angles. The SCP-problem is one of key computational cornerstones for many important problems such as protein design, flexible docking of proteins, homology modeling, etc. The SCP-problem can be formulated as a minimization problem of an integer linear program which is NP-hard thus inevitably invites heuristic approach to find the solution. In this paper, we report a heuristic algorithm, called BetaSCP2, which quickly finds an excellent solution of the SCP-problem. The solution process of the BetaSCP2 is facilitated by the Voronoi diagram and its dual structure called the quasi-triangulation. The BetaSCP2 is entirely implemented using the Molecular Geometry engine called BULL! which has been developed by Voronoi Diagram Research Center (VDRC) in C++ programming language. The benchmark test of the BetaSCP2 with other programs is also provided. The BetaSCP2 program is available as both a stand-alone and a web server program from VDRC.

Keywords: protein structure/function, side-chain prediction, BetaSCP2, Voronoi diagram, quasi-triangulation, beta-complex.

1 Introduction

Bio-molecules such as protein, DNA, and RNA play important biological functions in the living bodies. It is a general consensus that the functions of molecules come from their geometric structures. Hence, there have been tremendous studies which tried to figure out the relationship between the structure and functions either computationally or experimentally.

Protein consists of linearly connected amino acids by a peptide bond where a water molecule leaves during each connection. Residue, the remaining part of each amino acid in a bonded sequence, consists of two parts: backbone and side-chain. While the backbone part is common to each residue, the side-chain

⋆ Corresponding author.

H. Hong and C. Yap (Eds.): ICMS 2014, LNCS 8592, pp. 333–340, 2014.

is different depending on each type of residue. Protein structure is generally determined by the dihedral angles of some rotatable bonds in each residue because other variations except the angles are relatively negligible [11].

The side-chain prediction problem, abbreviated as SCP-problem, is a computational problem to predict the optimal structure of proteins by finding the optimal dihedral angles. Assuming that a backbone structure is fixed (i.e., the coordinates of the atoms in a backbone are given), the SCP-problem finds the optimal side-chain structure by predicting the dihedral angles in side-chains so that the total energy of the structure is minimized.

While each dihedral angle, in theory, can take any value between 0 and 360 degrees, it is well-known that there exists a preferred range of dihedral angle which maps to a representative angle through statistical analysis. A combination of such representatives for each residue is called a *rotamer* which is short for rotational isomer. Example rotamer is shown in Figure 1 . Figure 1(a) shows the chemical formula of aspartic acid whose side-chain has two dihedral angles (χ_1: between CH and CH_2; χ_2: between CH_2 and C) as shown in Fig. 1(b). The backbone atoms are also shown (H_2N, CH, C, OH) at the top of Fig. 1(b). Fig. 1(c) shows the union of nine rotamers for aspartic acid with different values of χ_1 and χ_2 (Hydrogens are usually ignored in the graphical visualization). Different residues have different number of dihedral angles. Hence, there could exist different rotamer set for each type of residue. The collection of rotamer sets for all residue types is called a *rotamer library* [12,23].

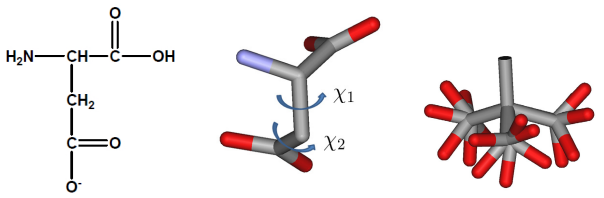


Fig. 1. Example rotamer: (a) the chemical formula of aspartic acid, (b) the two dihedral angles in the side-chain of aspartic acid, (c) the union of nine rotamers for aspartic acid

Consider a protein $\Pi = \{\rho_1, \rho_2, \ldots, \rho_n\}$ consisting of n residues. Each residue ρ has the backbone part β and the side-chain σ. Let $\mathcal{B} = \{\beta_1, \beta_2, \ldots, \beta_n\}$ and $\Sigma = \{\sigma_1, \sigma_2, \ldots, \sigma_n\}$ be the backbone parts and side-chains for the residues, respectively. Thus, β_i and σ_i constitute ρ_i and a protein can be represented as $\Pi = \mathcal{B} \cup \Sigma$. $\mathcal{B}$ is called the backbone of the protein Π. Suppose that the structure of Π (i.e. the atom coordinates of $\mathcal{B}$ and Σ) is completely defined. Then, the potential energy E_Π of Π for the SCP-problem is usually defined as follows [10]:

$$E_\Pi = E_{\mathcal{B}\Sigma} + E_{\Sigma\Sigma} \tag{1}$$

where $E_{\mathcal{B}\Sigma}$ is the potential energy between protein backbone and the side-chain of each residue and $E_{\Sigma\Sigma}$ is the potential energy between a side-chain σ_i and another side-chain σ_j, $i \neq j$. The energy function used in this study is the van der Waals interaction between non-bonded atoms modeled by the following Lennard-Jones potential energy function

$$\begin{aligned} E &= E_{\mathcal{B}\Sigma} + E_{\Sigma\Sigma} \\ &= \sum_{i\in\beta}\sum_{j\in\sigma}\{\frac{A_{ij}}{d_{ij}^{12}} - \frac{B_{ij}}{d_{ij}^{6}}\} + \sum_{i\in\sigma_i}\sum_{j\in\sigma_j}\{\frac{A_{ij}}{d_{ij}^{12}} - \frac{B_{ij}}{d_{ij}^{6}}\} \end{aligned} \tag{2}$$

where d_{ij} is the Euclidean distance between the centers of a pair of atoms a_i and a_j. A_{ij} and B_{ij} are constants depending on atom types and the parameters in either AMBER [8] or CHARMM [4] could be used.

Given a rotamer library, the energy function E, and the structure of $\mathcal{B}$, the SCP-problem is to assign the optimal rotamer r^* to each residue ρ for σ so that E_{Π} of Eq. (1) is minimized [10]. Thus, the SCP-problem can be formulated as a minimization problem of an integer linear program[13,21,28] which is NP-hard thus necessarily invites a heuristic approach to find the solution. The NP-hardness of the SCP-problem is proved either by reducing satisfiability (SAT) problem to the decision problem of the SCP-problem [24,7] or by reducing the unconstrained quadratic 0-1 programming problem to the formulation of the SCP-problem [14]. The SCP-problem is one of key computational cornerstones for many important problems such as protein design [9,3], flexible docking of proteins [2,26], homology modeling [27], etc.

2 BetaSCP2 Algorithm

The SCP-problem can be formulated in an integer linear program (ILP) of Formulation 1 [13,6,14,21,28] where the constraints are not shown here due to space constraint. Two types of decision variables corresponding to two types of energies in Eq. (1) are defined as follows: x_{ij} decides whether rotamer j is accepted for residue i or not; x_{ijkl} decides whether the interaction between rotamer k of residue i and rotamer l of residue j is accepted or not. m_i represents the number of rotamers for a residue i in rotamer library.

Formulation 1. (ILP for SCP-problem)

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

$$x_{ij} \in \{0,1\}, x_{ijkl} \in \{0,1\}. \tag{4}$$

While the size $\|x_{ik}\|$ of x_{ik} linearly increases with respect to $M_1 = \sum_{i=1}^{n} m_i$, the size $\|x_{ijkl}\|$ of x_{ijkl} dramatically grows according to $M_2 = \sum_{i=1}^{n-1}\sum_{k=i+1}^{n}(m_i \times m_k)$ where $M_2 \gg M_1$. Therefore, we prefer to cut down $\|x_{ijkl}\|$. BetaSCP1 algorithm, previously reported [25], decomposes the SCP-problem into subproblems and solves the ILP corresponding to each subproblem by using CPLEX

solver [1]. While BetaSCP1 produces a solution very close to the global optimum, it is computationally very inefficient because it should invoke the CPLEX solver repeatedly.

BetaSCP2 algorithm decomposes the SCP-problem into small-sized subproblems and transforms each subproblem into simple geometric problem which is very efficiently solved via the theory of beta-complex, which is a derived from the Voronoi diagram of spheres (Refer to the Appendix for the brief discussion). The input of the procedure `BetaSCP2` below is a protein backbone $\mathcal{B}$ from a PDB file and a rotamer library $\mathcal{R}$. In STEP 1, `BetaSCP2` initially assigns a rotamer r^0 to each residue by considering the probability of the rotamer instances in $\mathcal{R}$. In STEP 2 and 3, the minimum enclosing sphere (MES) for each residue and the quasi-triangulation of MES set are computed by using BULL! engine [16]. In STEP 4, `BetaSCP2` improves the rotamer r^0 initially assigned to each residue ρ by looking into the only nearby other rotamers which is defined by first-order Voronoi neighbors of Definition 1. Through STEP 4.1 and 4.2, `BetaSCP2` computes the intersection volume of each candidate for ρ and choose the best one with minimum intersection volume. Note that `BetaSCP2` exploits the intersection volume instead of potential energy. It can be easily proved that the rotamer with less intersection volume has the lower potential energy if there exists such an intersection. This important observation will be reported elsewhere in future.

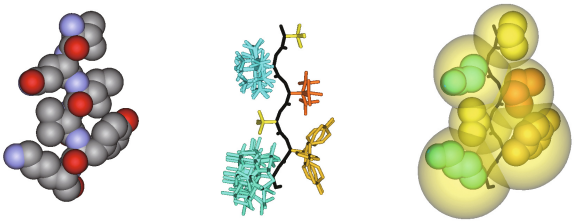
BetaSCP2($\mathcal{B}$, $\mathcal{R}$)

1 1. assign an initial rotamer r^0 to each residue.
2 2. compute the MES of each residue.
3 3. compute the quasi-triangulation for the MES set.
4 4. **for** the first-order Voronoi neighbors $\mathcal{FN}$ of the MES for each residue ρ
5 4.1. **for** each rotamer r of ρ in $\mathcal{R}$
6 4.1.1. compute intersection volume $\mathcal{XV}(r)$ of r with
7 other rotamers in $\mathcal{FN}$.
8 4.2. find out the best rotamer r^* with minimum $\mathcal{XV}(r^*)$.

Figure 2 illustrates the BetaSCP2 algorithm: Figure 2(a) is an input protein structure; Figure 2(b) shows the backbone and the rotamer set corresponding to each residue by stick model; Figure 2(c) shows the rotamers to be initially assigned to each residue and their minimum enclosing spheres (MES) in yellow by space-filling model.

3 Benchmark Test

We have compared the BetaSCP2 algorithm with SCWRL4 [22] and CISRR [5] against 248 data from Protein Data Bank (PDB). The computational environment is as follows: Intel Core 2 Duo CPU E6850 (3.0GHz with 4GB RAM) and Windows 7. For comparing the solution quality, we evaluated Lennard-Jones potential (LJ) energy functions of the structures computed by each program. Figure 3(a) and (b) show LJ energies for three data sets from three programs.



(a) (b) (c)

Fig. 2. Illustration of BetaSCP2 algorithm: (a) input protein structure (PDB code: 3FQP), (b) the backbone and rotamer sets of residues of (a), (c) the initially assigned rotamers to each residue and their minimum enclosing spheres

The X-axis represents the computed structures with respect to their residue sizes. The Y-axis represents LJ energies of the structures. Due to too high energies of SCWRL4, the difference between BetaSCP2 and CISRR are not clearly recognized in Figure 3(a); its zoom-up in Figure 3(b) clearly shows the powerful result of BetaSCP2.

BetaSCP2 produces energetically very stable structures, compared to both SCWRL4 and CISRR as shown in Figure 3(b). It turns out that BetaSCP2 outperforms SCWRL4 for 214 among 248 data. BetaSCP2 outperforms CISRR

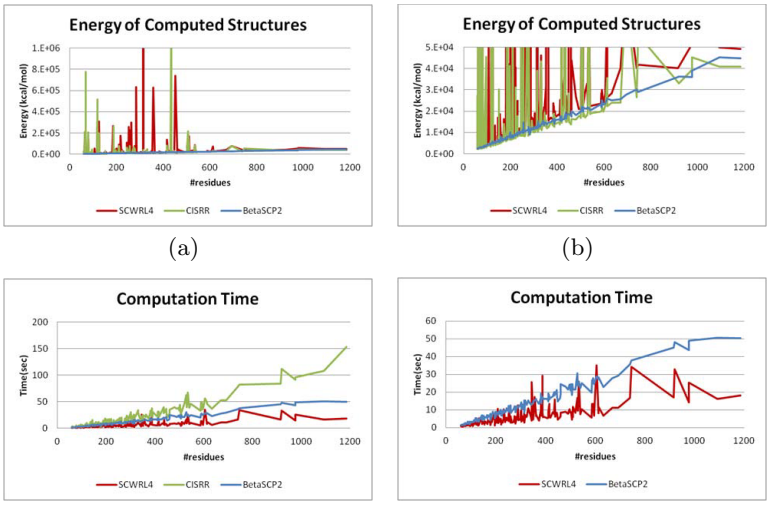


Fig. 3. Benchmark test for BetaSCP2, SCWRL4 and CISRR: (a) energies of the structures computed by BetaSCP2, SCWRL4, and CISRR, (b) the zoom-up of (a), (c) computation times for BetaSCP2, SCWRL4, and CISRR, and (d) computation times for BetaSCP2 and CISRR

for 52 data. However, the energy of BetaSCP2 are extremely lower for those 52 data. The mean and variance of BetaSCP2 are very low compared to those of CISRR. For the other 196 data, their energy differences between BetaSCP2 and CISRR are relatively small.

For comparing the computational efficiency, we counted the computation time of each program. Figure 3(c) shows the computation times of BetaSCP2, SCWRL4, and CISRR. Figure 3(d) shows the computation times of BetaSCP2 and SCWRL4 only. SCWRL4 is fastest among three programs. While BetaSCP2 shows a strongly linear pattern from both graphs, CISRR seems to have a super-linear pattern. BetaSCP2 is approximately three times faster than CISRR. The computation time taken by SCWRL4 relatively fluctuates wildly with respect to protein size.

4 Conclusion

In this paper, we reported the BetaSCP2 algorithm, and its implementation which quickly finds an excellent solution of the SCP-problem. The core idea of the BetaSCP2 algorithm is to transform the SCP-problem into a simple geometric problem whose solution process can be facilitated by the Voronoi diagram and its dual structure called the quasi-triangulation. Due to this idea, BetaSCP2 could improve the computational efficiency compared to BetaSCP1 without degrading the solution quality. The BetaSCP2 algorithm is entirely implemented using the Molecular Geometry engine called BULL! which has been developed by Voronoi Diagram Research Center (VDRC) in C++ programming language. Comparing with other programs, BetaSCP2 produces energetically very stable structures efficiently. Even though there could be other criteria, we compare the programs according to energy of computed structure.

Acknowledgement. This research was supported by the National Research Lab grant funded by the National Research Foundation (NRF) of Korea (No. 2012R1A2A1A05026395).

Appendix: Voronoi Diagram and Its Derivative Structures

Suppose that we are given a set $S = \{s_1, s_2, \ldots, s_n\}$ of spheres $s_i = (p_i, r_i)$ in $\mathbb{R}^3$ where p_i is the center and r_i is the radius. Then *Voronoi diagram* $\mathcal{VD}$ of S consists of n Voronoi cells: $\{\mathrm{VC}(s_1), \mathrm{VC}(s_2), \cdots, \mathrm{VC}(s_n)\}$. A Voronoi cell $\mathrm{VC}(s_i) = \{d(x, p_i) - r_i < d(x, p_j) - r_j, i \neq j\}$ where $d(x, y)$ is the Euclidean distance between two points x and y. Then $\mathcal{VD}$ is represented by the quadruplet $(V^{\mathcal{V}}, E^{\mathcal{V}}, F^{\mathcal{V}}, C^{\mathcal{V}})$: $V^{\mathcal{V}} = \{v_1^{\mathcal{V}}, v_2^{\mathcal{V}}, \ldots\}$, $E^{\mathcal{V}} = \{e_1^{\mathcal{V}}, e_2^{\mathcal{V}}, \ldots\}$, $F^{\mathcal{V}} = \{f_1^{\mathcal{V}}, f_2^{\mathcal{V}}, \ldots\}$, and $C^{\mathcal{V}} = \{c_1^{\mathcal{V}}, c_2^{\mathcal{V}}, \ldots\ c_n^{\mathcal{V}}\}$ are the sets of the Voronoi vertices (V-vertices), Voronoi edges (V-edges), Voronoi faces (V-faces), and Voronoi cells (V-cells) in $\mathcal{VD}$, respectively. Note that $\mathcal{VD}$ is different from the ordinary Voronoi diagram of sphere centers in many respects. One of the important properties for

$\mathcal{VD}$ is that $\mathcal{VD}$ reflects the size differences among spheres in Euclidean distance metric. For the details of $\mathcal{VD}$, refer to [15,20].

Given $\mathcal{VD}$, its dual structure *quasi-triangulation* $\mathcal{QT}$ is defined as follows: Each V-vertex maps to a tetrahedral cell simplex (q-cell); Each V-edge maps to a triangular face simplex (q-face); Each V-face maps to an edge simplex (q-edge); And each V-cell maps to a vertex simplex (q-vertex). Then $\mathcal{QT}$ is represented by the quadruplet $(V^{\mathcal{Q}}, E^{\mathcal{Q}}, F^{\mathcal{Q}}, C^{\mathcal{Q}})$: $V^{\mathcal{Q}} = \{v_1^{\mathcal{Q}}, v_2^{\mathcal{Q}}, \ldots\ v_n^{\mathcal{Q}}\}$, $E^{\mathcal{Q}} = \{e_1^{\mathcal{Q}}, e_2^{\mathcal{Q}}, \ldots\}$, $F^{\mathcal{Q}} = \{f_1^{\mathcal{Q}}, f_2^{\mathcal{Q}}, \ldots\}$, and $C^{\mathcal{Q}} = \{c_1^{\mathcal{Q}}, c_2^{\mathcal{Q}}, \ldots\}$ are the sets of the q-vertices, q-edges, q-faces, and q-cells in $\mathcal{QT}$, respectively.

$\mathcal{VD}$ and $\mathcal{QT}$ are equivalent to each other in mathematical and computational point of view. Given $\mathcal{VD}$, $\mathcal{QT}$ of S is computed in $O(m)$ time in the worst case where m is the number of the q-simplexes in $\mathcal{QT}$. The reverse conversion from $\mathcal{QT}$ to $\mathcal{VD}$ takes linear time in the worst case with respect to the number of the topological entities in $\mathcal{VD}$. For the details of $\mathcal{QT}$, see [19,17,18].

Definition 1 (First-order Voronoi Neighbors)
Suppose that we have $\mathcal{VD}$ of a set $S = \{s_1, s_2, \ldots, s_n\}$ of spherical balls. Let $F_i^{\mathcal{V}}$ be the V-faces bounding $VC(s_i)$ of a ball s_i. $F_j^{\mathcal{V}}$ is similarly defined. Given a spherical ball $s_i \in S$, a set $\mathcal{FN}_i = \{s_j \in S \mid F_i^{\mathcal{V}} \cap F_j^{\mathcal{V}} \neq \emptyset,\ i \neq j\}$ is called the first-order neighbors of s_i.

References

1. IBM ILOG CPLEX Optimizer (2013), `http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/`
2. Althaus, E., Kohlbacher, O., Lenhof, H.P., Müller, P.: A combinatorial approach to protein docking with flexible side chains. Journal of Computational Biology 9(4), 597–612 (2002)
3. Ashworth, J., Havranek, J.J., Duarte, C.M., Sussman, D., Monnat Jr., R.J., Stoddard, B.L., Baker, D.: Computational redesign of endonuclease DNA binding and cleavage specificity. Nature 441(7093), 656–659 (2006)
4. Brooks, B.R., Bruccoleri, R.E., Olafson, B.D., States, D.J., Swaminathan, S., Karplus, M.: CHARMM: A program for macromolecular energy, minimization, and dynamics calculations. Journal of Computational Chemistry 4(2), 187–217 (1983)
5. Cao, Y., Song, L., Miao, Z., Hu, Y., Tian, L., Jiang, T.: Improved side-chain modeling by coupling clash-detection guided iterative search with rotamer relaxation. Bioinformatics 27(6), 785–790 (2011)
6. Chazelle, B., Kingsford, C., Singh, M.: The side-chain positioning problem: A semidefinite programming formulation with new rounding schemes. In: Goldin, D.Q., Shvartsman, A.A., Smolka, S.A., Vitter, J.S., Zdonik, S.B. (eds.) Proceedings of the ACM International Conference Proceeding Series;Proceedings of the Paris C. Kanellakis Memorial Workshop on Pr., vol. 41, pp. 86–94 (2003)
7. Chazelle, B., Kingsford, C., Singh, M.: The inapproximability of side-chain positioning. Tech. rep., Princeton University (2004)
8. Cornell, W.D., Cieplak, P., Bayly, C.I., Gould, I.R., Merz Jr., K.M., Ferguson, D.M., Spellmeyer, D.C., Fox, T., Caldwell, J.W., Kollman, P.A.: A second generation force field for the simulation of proteins, nucleic acids, and organic molecules. Journal of the American Chemical Society 117, 5179–5197 (1995)

9. Dahiyat, B.I., Mayo, S.L.: De novo protein design: Fully automated sequence selection. Science 278(3), 82–87 (1997)
10. Desmet, J., Maeyer, M.D., Hazes, B., Lasters, I.: The dead-end elimination theorem and its use in protein side-chain positioning. Nature 356, 539–542 (1992)
11. Dunbrack Jr., R.L.: Rotamer libraries in the 21st century. Current Opinion in Structural Biology 12(4), 431–440 (2002)
12. Dunbrack Jr., R.L., Cohen, F.E.: Bayesian statistical analysis of protein side-chain rotamer preferences. Protein Science 6(8), 1661–1681 (1997)
13. Eriksson, O., Zhou, Y., Elofsson, A.: Side chain-positioning as an integer programming problem. In: Gascuel, O., Moret, B.M.E. (eds.) WABI 2001. LNCS, vol. 2149, pp. 128–141. Springer, Heidelberg (2001)
14. Fung, H., Rao, S., Floudas, C., Prokopyev, O., Pardalos, P., Rendl, F.: Computational comparison studies of quadratic assignment like formulations for the In silico sequence selection problem in De Novo protein design. Journal of Combinatorial Optimization 10(1), 41–60 (2005)
15. Kim, D.S., Cho, Y., Kim, D.: Euclidean Voronoi diagram of 3D balls and its computation via tracing edges. Computer-Aided Design 37(13), 1412–1424 (2005)
16. Kim, D.S., Cho, Y., Kim, J.K., Ryu, J., Lee, M., Cha, J., Song, C.: Bull! - the molecular geometry engine based on voronoi diagram, quasi-triangulation, and beta-complex. In: The 4th International Congress on Mathematical Software (2014)
17. Kim, D.S., Cho, Y., Sugihara, K.: Quasi-worlds and quasi-operators on quasi-triangulations. Computer-Aided Design 42(10), 874–888 (2010)
18. Kim, D.S., Cho, Y., Sugihara, K., Ryu, J., Kim, D.: Three-dimensional beta-shapes and beta-complexes via quasi-triangulation. Computer-Aided Design 42(10), 911–929 (2010)
19. Kim, D.S., Kim, D., Cho, Y., Sugihara, K.: Quasi-triangulation and interworld data structure in three dimensions. Computer-Aided Design 38(7), 808–819 (2006)
20. Kim, D., Kim, D.S.: Region-expansion for the Voronoi diagram of 3D spheres. Computer-Aided Design 38(5), 417–430 (2006)
21. Kingsford, C.L., Chazelle, B., Singh, M.: Solving and analyzing side-chain positioning problems using linear and integer programming. Bioinformatics 21(7), 1028–1036 (2005)
22. Krivov, G.G., Shapovalov, M.V., Dunbrack Jr., R.L.: Improved prediction of protein side-chain conformations with SCWRL4. PROTEINS: Structure, Function, and Bioinformatics 77(4), 778–795 (2009)
23. Lovell, S.C., Word, J.M., Richardson, J.S., Richardson, D.C.: The penultimate rotamer library. Proteins: Structure, Function, and Genetics 40(3), 389–408 (2000)
24. Pierce, N.A., Winfree, E.: Protein design is NP-hard. Protein Engineering 15(10), 779–782 (2002)
25. Ryu, J., Kim, D.S.: Protein structure optimization by side-chain positioning via beta-complex. Journal of Global Optimization 57(2), 217–250 (2013)
26. Schumann, M., Armen, R.S.: Systematic and efficient side chain optimization for molecular docking using a cheapest-path procedure. Journal of Computational Chemistry 34, 1258–1269 (2013)
27. Xiang, Z., Honig, B.: Extending the accuracy limits of prediction for side-chain conformations. Journal of Molecular Biology 311(2), 421–430 (2001)
28. Zhu, Y.: Mixed-integer linear programming algorithm for a computational protein design problem. Industrial and Engineering Chemistry Research 46, 839–845 (2007)