Expert Systems with Applications 40 (2013) 5210–5218

Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

A cluster-DEE-based strategy to empower protein design

Rafael K. de Andrades, Márcio Dorn *, Daniel S. Farenzena, Luis C. Lamb

Federal University of Rio Grande do Sul, Institute of Informatics, Av. Bento Gonçalves 9500, 91501-970 Porto Alegre, RS, Brazil

ARTICLE INFO

Keywords:
Integrated intelligent methods
Protein design
Structural bioinformatics
Clustering algorithms
Boolean Satisfiability problem
Dead-End-Elimination

ABSTRACT

The Medical and Pharmaceutical industries have shown high interest in the precise engineering of protein hormones and enzymes that perform existing functions under a wide range of conditions. Proteins are responsible for the execution of different functions in the cell: catalysis in chemical reactions, transport and storage, regulation and recognition control. Computational Protein Design (CPD) investigates the relationship between 3-D structures of proteins and amino acid sequences and looks for all sequences that will fold into such 3-D structure. Many computational methods and algorithms have been proposed over the last years, but the problem still remains a challenge for Mathematicians, Computer Scientists, Bioinformaticians and Structural Biologists. In this article we present a new method for the protein design problem. Clustering techniques and a Dead-End-Elimination algorithm are combined with a SAT problem representation of the CPD problem in order to design the amino acid sequences. The obtained results illustrate the accuracy of the proposed method, suggesting that integrated Artificial Intelligence techniques are useful tools to solve such an intricate problem.

1. Introduction

Computational Protein Design (CPD) is one of the most important research problems in Computational Molecular Biology (Lippow and Tidor, 1997; Tian, 2010). The Medical and Pharmaceutical industries are widely interested in precisely understanding how to engineer protein hormones and enzymes that perform existing functions under a wide range of conditions. Proteins are long sequences of 20 different amino acid residues that in physiological conditions adopt a unique 3-D structure (Lehninger et al., 2005). This structure is important because it determines the function of the protein in the cell, for example, catalysis in chemical reactions, transport and storage, regulation, recognition and control (Lesk, 2002). Protein design has become a powerful approach for understanding the relationship between amino acid sequence and 3-D structure and consequently to study the functional aspects of the protein (Sander et al., 1992). The ability to design sequences compatible with a fold may also be useful in structural and functional Genomics, with the identification of functionally important domains in sequences of proteins.

The general goal of CPD is to identify an amino acid sequence that folds into a particular protein 3-D structure with a desired function (Fig. 2). Protein design can be considered as the inverse of the protein folding (PF) problem (Osguthorpe, 2000) because it starts with the structure rather than the sequence and looks for all sequences that will fold into such 3-D structure. Considering that there are 20 naturally occurring amino acids for each position, the combinatorial complexity of the problem amounts to 20^{110} or 10^{130} (Floudas et al., 2006).

Over the last decade, a tremendous advance in protein design was witnessed. The maturation of a number of component technologies, including optimization algorithms and combinatorial discrete search, contributed for advances in CPD research. Techniques such as Dead-End-Elimination (Georgiev et al., 2008; Desmet et al., 1992), Integer Programing (Xie and Sahinidis, 2006) and Monte Carlo (Yang and Saven, 2005; Hom and Mayo, 2006; Allen and Mayo, 2006) have been applied to protein design, but the problem still remains very challenging. In this article, we present a new method based on clustering strategy, Dead-End-Elimination techniques and SAT-based methods to determine the amino acid sequence of protein 3-D structures.

The remainder of the paper is structured as follows. Section 2 contextualizes the protein design problem and basic concepts used in this article. Section 3 introduces the computational and AI techniques used in the proposed method. Section 4 introduces the new hybrid method for protein design. Section 5 reports several results illustrating the effectiveness of our method. Section 6 concludes and points out directions for further research.

2. Preliminaries

2.1. Protein structure and protein design

A peptide is a molecule composed of two or more amino acid residues chained by a bond called the peptide bond. There are 20

* Corresponding author.
E-mail address: marcio.dorn@acm.org (M. Dorn).

0957-4174/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.eswa.2013.03.011

different amino acid residues in nature (Lodish et al., 1990) and each amino acid residue is a molecule containing both amine and carboxyl functional groups. The various amino acids differ in which side chain (R group) is attached to their alpha carbon (Lodish et al., 1990; Lehninger et al., 2005). Larger peptides are generally referred to as polypeptides or proteins (Creighton, 1990). A peptide has three main chain torsion angles, namely phi (ϕ), psi (ψ) and omega (ω).

In the model peptide (Fig. 1) the bonds between N and C_α, and between C_α and C are free to rotate. These rotations are described by the ϕ and ψ torsion angles, respectively. The rotational freedom about the ϕ (C_α-N) and ψ (C_α-C) angles is limited by steric hindrance between the side chain of the residue and the peptide backbone. Consequently, the possible conformation of a given polypeptide chain is quite limited (Ramachandran and Sasisekharan, 1968; Branden and Tooze, 1998). Side-chains also present dihedral angles and the number of χ angles depends on the residue type (Table 1).

Protein design starts with the structure (set of torsion angles, for example) and looks for all sequences that will fold into such 3-D structure. In rational protein design, the scientist uses detailed knowledge of the structure and function of the protein to make desired changes. The design of proteins that fold to a specified target backbone structure is of great interest of the Medical and the Pharmaceutical Industries. Additional material related to PD can be found in Dahiyat and Mayo (1997), Park et al. (2011), Guerois and de La Paz (2006), Samish et al. (2011), Hom and Mayo (2006), Yang and Saven (2005), Lippow and Tidor (1997), Tian (2010), Sander et al. (1992), Voigt et al. (2000), Pokala and Handel (2000) and Floudas et al. (2006). A good review about protein structure can be found in Tramontano (2006), Lesk (2002), Branden and Tooze (1998) and Altman and Dugan (2005).

2.2. Energy functions

An *energy function* describes the internal energy of the protein and its interactions with the environment in which it is inserted. In Protein design the goal is to find a sequence which generates a 3-D protein structure with the global minimum of free energy that corresponds to the native or functional state of the protein (Osguthorpe, 2000; Tramontano, 2006). The energy function used for protein design still contain elements accounting for van der Waals force, electrostatics, solvation, and hydrogen bonding.

A potential energy function incorporates two main types of terms: bonded and non-bonded. The bonded terms (bonds, angles and torsions) are covalently linked. The bonded terms constrain bond lengths and angles near their equilibrium values. The bonded terms also include a torsional potential (torsion) that models the periodic energy barriers encountered during bond rotation. The non-bonded potential includes: ionic bonds, hydrophobic interactions, hydrogen bonds, van der Waals forces, and dipole–dipole bonds. van der Waals force is usually described by the equation for Lennard–Jones 6–12 potential (Boas and Harbury (2007)). There is a variety of potential energy functions used in protein design. In this article we use the CHARMM potential energy function (Brooks et al., 1983; Field et al., 1998) (Eq. (1)).

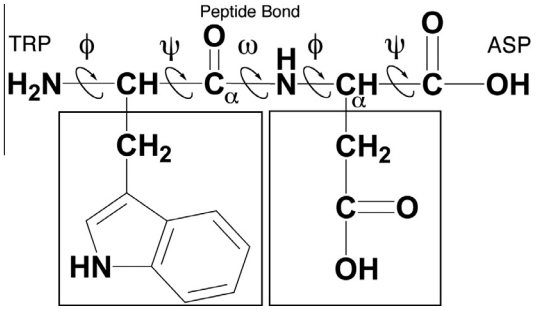

Fig. 1. Chemical representation of two amino acid residues after the condensation reaction. The carboxyl group of one amino acid (amino acid 1) reacts with the amino group of the amino acid 2. A molecule of water is removed from two amino acids to form a peptide bond and the ω angle is formed (peptide bond). N is nitrogen, C and C_α are carbons.

$$E_{\text{total}} = \sum_{\text{bonds}} K_b(b-b_0)^2 + \sum_{\text{UB}} K_{\text{UB}}(S-S_0)^2 + \sum_{\text{angle}} K_\theta(\theta-\theta_0)^2 + \sum_{\text{dihedrals}} K_\chi(1+\cos(\eta_\chi-\delta)) + \sum_{\text{impropers}} K_{imp}(\varphi-\varphi_0)^2) + \sum_{\text{nonbond}} \epsilon\left[\left(\frac{R_{minij}}{r_{ij}}\right)^{12} - \left(\frac{R_{minij}}{r_{ij}}\right)^{6}\right] + \frac{q_i q_j}{\epsilon_1 r_{ij}} \quad (1)$$

where K_b, K_{UB}, K_θ, K_χ and K_{imp} are the bond, Urey Bradley angle (Hagler et al., 1979; Lifson and Warshel, 1968), dihedral angle and improper dihedral angle force constants, respectively; b, S, θ, χ and φ are the bond length, Urey-Bradley 1.3 distance, bond angle, dihedral angle and improper torsion angle, respectively. The subscript zero represents the equilibrium value for the individual terms. Coulomb and Lennard–Jones 6–12 terms contribute to the external or non-bonded interactions; ϵ is the Lennard–Jones (the depth of the potential well) and R_{min} is the distance at the Lennard–Jones minimum, q_i is the partial atomic charge, ϵ_1 is the effective dielectric constant, and r_{ij} is the distance between atoms i and j. Reviews of protein energy functions and their application can be found in Lazaridis and Karplus (2000), Jorgensen and Tirado-Rives (2005), Gordon et al. (1999) and Hao and Scheraga (1999).

3. Computational techniques applied to protein design

3.1. The Boolean Satisfiability problem

Boolean Satisfiability (SAT) is the problem of determining if a given propositional logic formula can be satisfied given suitable value assignments (Selman et al., 1992). SAT is also a well-known NP-complete decision problem (Cook, 1971) and require worst-case exponential time. However, state-of-the-art SAT algorithms are effective at coping with large search spaces by exploiting the problem structure when it exists (Marques-Silva, 2008). Recently, SAT solvers have been applied in combinatorial problems such as protein folding and protein design (Gomes and Selman, 2005).

The notation for SAT instances usually follow the Conjunctive Nominal Form (CNF). The CNF of a Boolean function is a function formula with the following structure:

$$\begin{aligned} \phi &= \vee_i c_i \\ c_i &= \wedge_j l_j \end{aligned} \quad (2)$$

where c_i are referred as clauses and literal l_j represents the variable x_j or its negation $\neg x_j$ (Ollikainen et al., 2009). Thus, adding clauses is equivalent to adding constraints to variables x_j, reducing the search space. Moreover, to describe a problem as a SAT instance, one add constraints encoded as clauses to ϕ until the SAT instance corresponds to the problem structure that is being solved. For instance, in protein design we state a rule R that rotamers r_u and r_v cannot exist simultaneously in a protein. If r_i denotes the existence of rotamer i in a protein, then we can encode rule R as $\neg(r_u \wedge r_v)$.

Since literals in SAT problems can be either a boolean variable or a boolean variable complement, constraint problems using SAT can only encode discrete search spaces. Although there are SAT extensions to allow for continuous search spaces, we want to keep the original definition so we can rely on modern, faster