Journal of Global Optimization **14:** 299–312, 1999.
© 1999 *Kluwer Academic Publishers. Printed in the Netherlands.*

Optimization of Molecular Similarity Index with Applications to Biomolecules

LUNJIANG LING[1] and GUOLIANG XUE[2]

[1]*Department of Computer Science, University of Vermont, Burlington, VT 05405, USA (e-mail: ling@cs.uvm.edu); Also from Department of Protein Engineering, Institute of Biophysics, Academia Sinica, Beijing 100101, China;* [2]*Department of Computer Science, University of Vermont, Burlington, VT 05405, USA (e-mail: xue@cs.uvm.edu)*

(Received 21 November 1997; accepted in revised form 13 February 1998)

Abstract. Molecular similarity index measures the similarity between two molecules. Computing the optimal similarity index is a hard global optimization problem. Since the objective function value is very hard to compute and its gradient vector is usually not available, previous research has been based on non-gradient algorithms such as random search and the simplex method. In a recent paper, McMahon and King introduced a Gaussian approximation so that both the function value and the gradient vector can be computed analytically. They then proposed a steepest descent algorithm for computing the optimal similarity index of small molecules. In this paper, we consider a similar problem. Instead of computing atom-based derivatives, we directly compute the derivatives with respect to the six free variables describing the relative positions of the two molecules.. We show that both the function value and gradient vector can be computed analytically and apply the more advanced BFGS method in addition to the steepest descent algorithm. The algorithms are applied to compute the similarities among the 20 amino acids and biomolecules like proteins. Our computational results show that our algorithm can achieve more accuracy than previous methods and has a 6-fold speedup over the steepest descent method.

Key words: Computational biology, Global optimization, Molecular similarity

1. Introduction

An important problem in biochemistry and molecular biology is to determine how similar two molecules A and B are. In order to measure the similarity between two molecules, Carbó et al. [2] introduced the concept of *similarity index* which is defined in the following:

$$R_{AB}(\overrightarrow{X}) = \frac{\int \rho_A(r)\rho_B(r)\mathrm{d}r}{[\int \rho_A(r)\rho_A(r)\mathrm{d}r]^{1/2}[\int \rho_B(r)\rho_B(r)\mathrm{d}r]^{1/2}} = \frac{S_{AB}}{S_{AA}^{1/2} \cdot S_{BB}^{1/2}}, \qquad (1)$$

where $\rho_A(r)$ and $\rho_B(r)$ are the electron densities of molecules A and B, respectively, at a point r in the three dimensional space R^3; the vector $\overrightarrow{X} = \{x_c, y_c, z_c, \theta, \phi, \psi\}$ represents the six free variables specifying the relative position (x_c, y_c, z_c) and orientation (θ, ϕ, ψ) of A with respect to B, where $(x_c, y_c, z_c) \in R^3$ (the three dimensional Euclidean space) and $0 \leq \theta \leq \pi$, $0 \leq \phi, \psi \leq 2\pi$; and the integrations go over the whole three dimensional space R^3. S_{AA} and S_{BB} are the normalization integrals which are independent of $\overrightarrow{X}$.

The concept of similarity index makes it possible to quantitatively compute the similarity between two molecules. Although (1) was originally proposed from the viewpoint of quantum chemistry, the idea can be naturally extended to other fields. For example, the function ρ can be substituted by other properties of molecules to get the similarities using different measurements. In particular, the electron density ρ can often be replaced by electrostatic potential or electrostatic field, for they can be easily computed from atom-centered point charges and are closely related to biological activities.

Typically, the molecular similarity index is applied to compare a certain property among a group of molecules. For each pair of molecules A and B in the group, we may compute $R_{AB} = \max_{\overrightarrow{X}} R_{AB}(\overrightarrow{X})$. Then, according to the magnitudes of these indices, we may further order or cluster these molecules. Today, the concept of similarity index has been widely used, especially in drug design, molecular superimpose, and screen drug molecules from databases [1, 4, 10, 6, 15, 17].

Since (1) represents the similarity of the shapes of the electron densities of the two molecules but not of the relative magnitudes (e.g., when $\rho_A = n\rho_B$, $R_{AB} = 1$), Hodgkin and Richards [7] proposed the following improved definition of molecular similarity index:

$$R_{AB}(\overrightarrow{X}) = \frac{2\int \rho_A(r)\rho_B(r)\mathrm{d}r}{\int \rho_A(r)\rho_A(r)\mathrm{d}r + \int \rho_B(r)\rho_B(r)\mathrm{d}r} = \frac{2S_{AB}}{S_{AA} + S_{BB}}. \tag{2}$$

Formula (2) takes into account not only the shapes, but also the magnitudes of the electron densities of the two molecules. Under this definition, $R_{AB} = 2n/(1 + n^2)$ when $\rho_A = n\rho_B$. In this paper, we choose to use this latter definition in our computations.

In order to compute the similarity between two molecules, we need to maximize the similarity index over the six free variables. Therefore an optimization procedure is indispensable. There have been many studies on global optimization methods for the minimization of non-convex energy functions. We refer readers to [3, 11, 14, 16]. In previous research on the optimization of similarity index, the integrals in (1) or (2) were numerically evaluated on a grid, and optimizations were performed usually by a random search procedure. Apparently, the accuracy of the results depends on how the grid was designed (the extent and density). In order to get higher accuracy, we should pay the price of more computing time. On the other hand, random search procedure itself is very time consuming. All these together make it difficult to apply molecular similarity index to big molecular systems.

To avoid these disadvantages, Lee and Smithline [8] first tried to compute the integrals in (1) or (2) analytically. For this purpose, they approximated the electron density ρ by a linear combination of Gaussian functions. Notice that the electrostatic potential has the advantage that it is straightforward to compute classically using atom-centered point charges, as in formula (3),

$$V(r) = \sum_{i=1}^{N} \frac{q_i}{||r - R_i||}, \tag{3}$$

where N is the number of atoms in the molecule, q_i and R_i are the charge and coordinate of the ith atom. McMahon and King [13] uses (3) to replace the electron density $\rho(r)$. Further, a 3-Gaussian expansion was used to approximate the $1/r$ term in formula (3), hence (3) becomes:

$$V(r) = \sum_{i=1}^{N} q_i \sum_{j=1}^{3} \gamma_j e^{-\alpha_j ||r-R_i||^2}, \tag{4}$$

where $\{\alpha_j, \gamma_j | j = 1, 2, 3\}$ are some fitting constants. Then, S_{AB} in (1) or (2) can be analytically computed as:

$$\begin{aligned} S_{AB} &= \sum_{i=1}^{N_A} \sum_{j=1}^{N_B} q_i^A q_j^B \sum_{k=1}^{3} \sum_{l=1}^{3} \left(\frac{\pi}{\alpha_k + \alpha_l} \right)^{3/2} \gamma_k \gamma_l e^{-\frac{\alpha_k \alpha_l}{\alpha_k + \alpha_l} ||R_i^A - R_j^B||^2} &(5) \\ &= \sum_{i=1}^{N_A} \sum_{j=1}^{N_B} q_i^A q_j^B \sum_{k=1}^{6} s_k e^{t_k ||R_i^A - R_j^B||^2}, &(6) \end{aligned}$$

where the values of s_k and t_k are given in Table 1. Since S_{AA} and S_{BB} can be computed in the same way, we can compute R_{AB} analytically for any vector $\overrightarrow{X}$.

In protein engineering and molecular design, it is often needed to compare or superimpose two molecules. As similarity index is straightforward and rigorous, it should be a good candidate tool to solve these problems. However, as we know, the similarity index computations were performed only on small molecules (comprise fewer than 20 heavy atoms) so far, while biomolecules like proteins generally contain thousands of heavy atoms. In this paper, we propose a new similarity index computation and optimization method, and compare it with the steepest descent method and random search methods. Our computational results show that our method is both faster and more accurate. Using our optimization procedure, we further probed the possibility of applying similarity index to proteins, and obtained encouraging results.

2. Method

Suppose we have two molecules A and B, each composed of N_A and N_B atoms respectively. During the computation, we treat both molecules as rigid bodies. The similarity index for these two molecules depends on their relative position and orientation $\overrightarrow{X} = \{x_c, y_c, z_c, \theta, \phi, \psi\}$. Imagine that molecule B is fixed while

Table 1. The constants in the 3-Gaussian expansion: $\alpha_1 = 157.43$; $\alpha_2 = 10.11$; $\alpha_3 = 0.29$; $\gamma_1 = 17.24$; $\gamma_2 = 5.61$; $\gamma_3 = 1.46$.

k	s_k	t_k
1	$\gamma_1^2\left(\frac{\pi}{2\alpha_1}\right)^{3/2}$	$-\left(\frac{\alpha_1}{2}\right)$
2	$2\gamma_1\gamma_2\left(\frac{\pi}{\alpha_1+\alpha_2}\right)^{3/2}$	$-\left(\frac{\alpha_1\alpha_2}{\alpha_1+\alpha_2}\right)$
3	$2\gamma_1\gamma_3\left(\frac{\pi}{\alpha_1+\alpha_3}\right)^{3/2}$	$-\left(\frac{\alpha_1\alpha_3}{\alpha_1+\alpha_3}\right)$
4	$\gamma_2^2\left(\frac{\pi}{2\alpha_2}\right)^{3/2}$	$-\left(\frac{\alpha_2}{2}\right)$
5	$2\gamma_2\gamma_3\left(\frac{\pi}{\alpha_2+\alpha_3}\right)^{3/2}$	$-\left(\frac{\alpha_2\alpha_3}{\alpha_2+\alpha_3}\right)$
6	$\gamma_3^2\left(\frac{\pi}{2\alpha_3}\right)^{3/2}$	$-\left(\frac{\alpha_3}{2}\right)$

molecule A is movable. We use x_c, y_c, z_c to represent the coordinate of the center of molecule A, and use the three Euler angles θ, ϕ, ψ to describe its orientation. Thus, $\overrightarrow{X}$ is now used specially for describing molecule A. For $i = 1, N_A$, let $\{x_i^0, y_i^0, z_i^0\}$ be the coordinates of the ith atom in A, corresponding to $x_c = y_c = z_c = \theta = \phi = \psi = 0$. Also, for $j = 1, N_B$, let $\{x_j^B, y_j^B, z_j^B\}$ be the coordinates of the jth atom in B. Then for any given $\overrightarrow{X} = \{x_c, y_c, z_c, \theta, \phi, \psi\}$, the corresponding coordinates of the atoms in A can be computed as $\{x_i^A, y_i^A, z_i^A\}$, $i = 1, N_A$, using the following formula:

$$\begin{bmatrix} x_i^A \\ y_i^A \\ z_i^A \end{bmatrix} = \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} + \begin{bmatrix} t_{11} & t_{12} & t_{13} \\ t_{21} & t_{22} & t_{23} \\ t_{31} & t_{32} & t_{33} \end{bmatrix} \begin{bmatrix} x_i^0 \\ y_i^0 \\ z_i^0 \end{bmatrix}, \tag{7}$$

where

$i = 1, N_A$,
$t_{11} = \cos\phi\cos\psi - \cos\theta\sin\phi\sin\psi$,
$t_{12} = -\cos\phi\sin\psi - \cos\theta\sin\phi\cos\psi$,
$t_{13} = \sin\theta\sin\phi$,
$t_{21} = \sin\phi\cos\psi + \cos\theta\cos\phi\sin\psi$,
$t_{22} = -\sin\phi\sin\psi + \cos\theta\cos\phi\cos\psi$,
$t_{23} = -\sin\theta\cos\phi$,
$t_{31} = \sin\theta\sin\psi$,
$t_{32} = \sin\theta\cos\psi$,
$t_{33} = \cos\theta$.

Therefore we can compute the corresponding similarity according to formulae (2) and (6) for any given $\overrightarrow{X}$.

Given two molecules A and B, our goal is to find a point $\overrightarrow{X}$ such that $R_{AB}(\overrightarrow{X})$ is globally maximized. If the maximum value of $R_{AB}(\overrightarrow{X})$ is close to 1.0, we say that the two molecules are very similar. If the maximum value of $R_{AB}(\overrightarrow{X})$ is close to 0.0, we say that the two molecules are not similar. Since the function $R_{AB}(\overrightarrow{X})$ is generally non-convex, this is a global optimization problem.

Since many previous works are based on the random search algorithm, we have implemented a version of the random search algorithm to compare with our new method. A general random search algorithm is described in Figure 1.

In our implementation, the criterion of “converged” is defined to be 3000 repeated invalid iterations for random search algorithm. Whenever the random search converges, we obtain a *local minimizer*. The search is then restarted using a different starting point. The best local minimizer is considered as the putative global minimizer.

Following [13], we compute the objective function value analytically using formulae (2) and (6). Rather than computing atom-based derivatives, we directly compute the derivatives with respect to six variables $\overrightarrow{X}$. Since both molecules A and B are rigid, the denominator of (2) is a constant. From (2), (6) and (7), we can compute the partial derivatives of $R_{AB}(\overrightarrow{X})$ with respect to $\overrightarrow{X}(n)$ analytically as

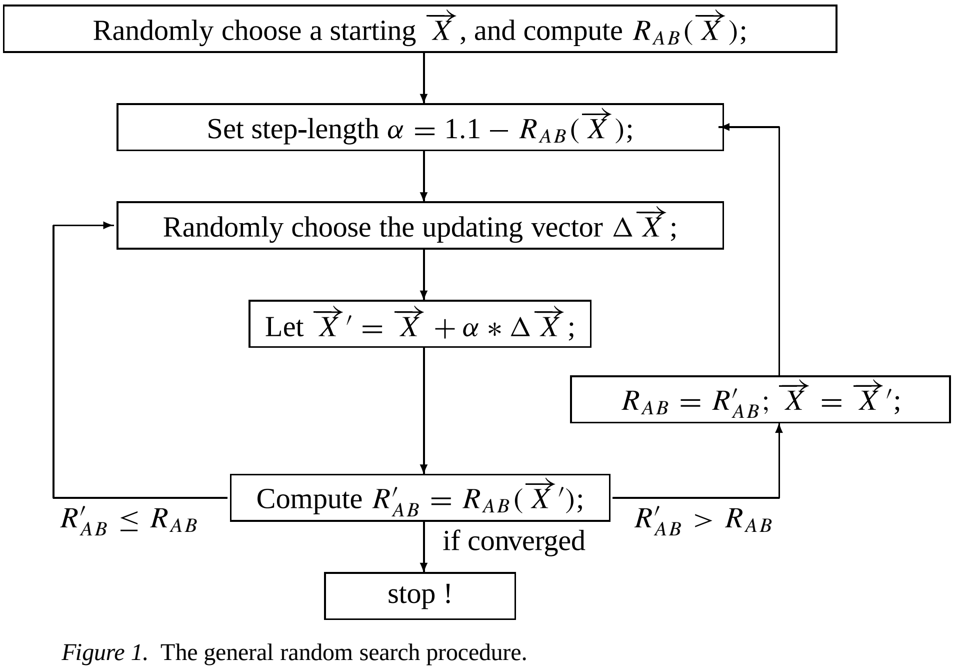
Randomly choose a starting $\overrightarrow{X}$, and compute $R_{AB}(\overrightarrow{X})$;
Set step-length $\alpha = 1.1 - R_{AB}(\overrightarrow{X})$;
Randomly choose the updating vector $\Delta\overrightarrow{X}$;
Let $\overrightarrow{X}' = \overrightarrow{X} + \alpha * \Delta\overrightarrow{X}$;
$R_{AB} = R'_{AB}$; $\overrightarrow{X} = \overrightarrow{X}'$;
Compute $R'_{AB} = R_{AB}(\overrightarrow{X}')$;
$R'_{AB} \le R_{AB}$
$R'_{AB} > R_{AB}$
if converged
stop !

Figure 1. The general random search procedure.

follows:

$$\frac{\partial R_{AB}}{\partial \overrightarrow{X}(n)} = \frac{2}{S_{AA} + S_{BB}} \sum_{i=1}^{N_A} \sum_{j=1}^{N_B} 2q_i q_j W_{ij}(n) \sum_{k=1}^{6} s_k t_k e^{t_k||R_i^A - R_j^B||^2}, \tag{8}$$

where s_k and t_k are as defined in Table 1 and the coefficients $W_{ij}(1), W_{ij}(2), \ldots W_{ij}(6)$ are computed as follows:

$$\begin{aligned}
W_{ij}(1) =& x_i^A - x_j^B;\;\; W_{ij}(2) = y_i^A - y_j^B;\;\; W_{ij}(3) = z_i^A - z_j^B;\\
W_{ij}(4) =& (x_i^A - x_j^B)[(\sin\theta\sin\phi\sin\psi)x_i^0 + (\sin\theta\sin\phi\cos\psi)y_i^0\\
&+ (\cos\theta\sin\phi)z_i^0] - (y_i^A - y_j^B)[(\sin\theta\cos\phi\sin\psi)x_i^0\\
&+ (\sin\theta\cos\phi\cos\psi)y_i^0 + (\cos\theta\cos\phi)z_i^0]\\
&+ (z_i^A - z_j^B)[(\cos\theta\sin\psi)x_i^0 + (\cos\theta\cos\psi)y_i^0 - (\sin\theta)z_i^0];\\
W_{ij}(5) =& (x_i^A - x_j^B)[(-\sin\theta\cos\psi - \cos\theta\cos\phi\sin\psi)x_i^0\\
&+ (\sin\phi\sin\psi - \cos\theta\cos\phi\cos\psi)y_i^0 + (\sin\theta\cos\phi)z_i^0]\\
&+ (y_i^A - y_j^B)[(\cos\phi\cos\psi - \cos\theta\sin\phi\sin\psi)x_i^0\\
&- (\cos\phi\sin\psi + \cos\theta\sin\phi\cos\psi)y_i^0 + (\sin\theta\sin\phi)z_i^0];\\
W_{ij}(6) =& (x_i^A - x_j^B)[(-\cos\phi\sin\psi - \cos\theta\sin\phi\cos\psi)x_i^0\\
&- (\cos\phi\cos\psi - \cos\theta\sin\phi\sin\psi)y_i^0]\\
&+ (y_i^A - y_j^B)[(-\sin\phi\sin\psi + \cos\theta\cos\phi\cos\psi)x_i^0\\
&- (\sin\phi\cos\psi + \cos\theta\cos\phi\sin\psi)y_i^0]\\
&+ (z_i^A - z_j^B)[(\sin\theta\cos\psi)x0_i - (\sin\theta\sin\psi)y0_i].
\end{aligned}$$

In this way, we can compute both the objective function value and the gradient vector analytically. Therefore we can apply more sophisticated gradient algorithms such as quasi-Newton algorithm [5] to maximize the similarity index.

The Limited Memory BFGS (LM-BFGS) algorithm is a modification of the standard BFGS quasi-Newton algorithm for nonlinear minimization. The standard BFGS algorithm is one of the most effective and widely used minimization algorithms where second-order derivative information is not available. For a complete description, see [9]. In our computation, the BFGS algorithms stops when the norm of the gradient is smaller than 0.0001.

3. Computational results

We have used the LM-BFGS code (provided by Nocedal) to maximize the similarity index and compared it with the steepest descent algorithm as well as the random search algorithm. All three algorithms were applied to compute the similarities among amino acids. We also applied our algorithm to compute the similarity

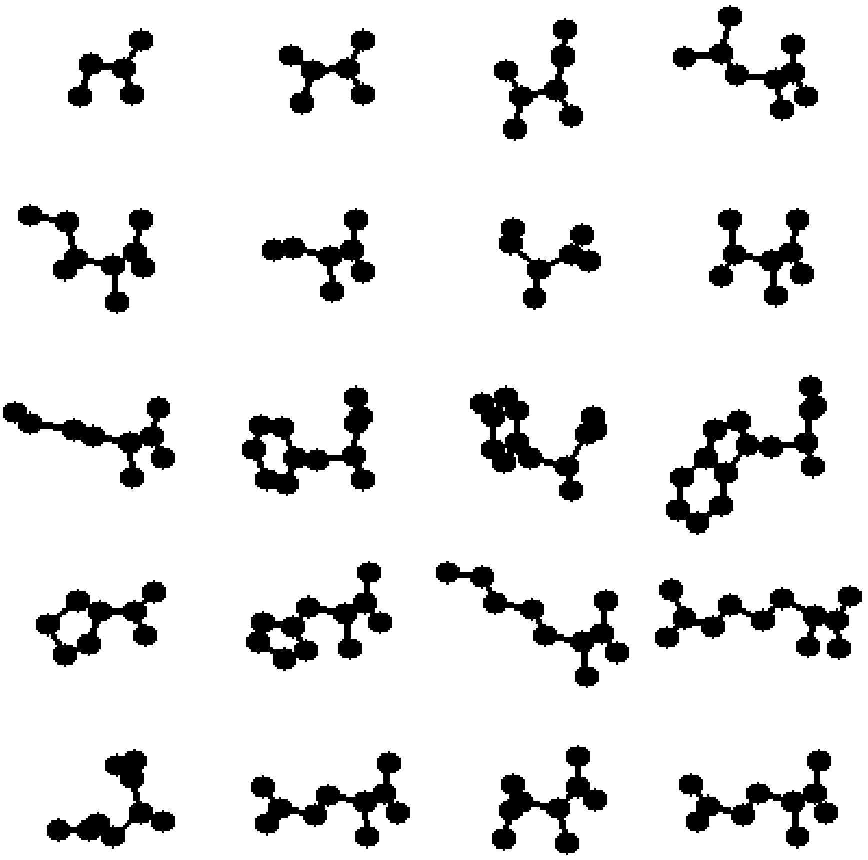

Figure 2. The structures of 20 amino acids.

between proteins (human insulin and pig insulin). All of the computation were performed on a Pentium Pro 200 MHz processor with 128 MB memory.

The structures of the 20 amino acids are shown in Figure 2. From left to right and top to bottom, the amino acids in the figure are: *G(Glycine), A(Alanine), V(Valine), L(Leucine), I(Isoleucine), S(Serine), C(Cysteine), T(Threonine), M(Methionine), F(Phenylalanine), Y(Tyrosine), W(Tryptophan), P(Proline), H(Histidine), K(Lysine), R(Arginine), D(Aspartic acid), E(Glutamic acid), N(Asparagine), Q(Glutamine).*

Their geometries were obtained from:

`http://www.chemie.fu-berlin.de/chemistry/bio/amino-acids.html.`

In our computation, if several hydrogen atoms are covalently linked to a heavy atom, we use a united atom to represent these hydrogen atoms and that heavy atom. In the rest of this paper, all real heavy atoms and united atoms are called heavy atoms. The atomic charges for the test problems were obtained from [12].

Table 2. Atomic charges for Phenylalanine and Alanine.

Main chain							
ATOM	N	CA	C	O1	OXT		
F	0.630	0.095	0.380	−0.565	−0.560		
A	0.630	0.095	0.380	−0.565	−0.560		
Side chain							
ATOM	CB	CG	CD1	CD2	CE1	CE2	CZ
F	0.005	0.039	−0.019	−0.019	0.010	0.010	−0.006
A	0.020						

From Figure 2 we can see that all amino acids are composed of two parts: main chains and side chains. All main chains comprise the same α-carbon, amino group and carbonyl group. Their structure and atomic net charges are almost exactly identical. However, the side chains are different from each other in both structure and atomic charge. Generally, the atoms on side chains posses much less net charges than that of the main chains. Therefore most amino acids pairs have very high similarities according to (1) or (2). As an example, let us look at Phenylalanine (F) and Alanine (A), whose atomic charges are shown in Table 2.

Although the side chains are quit different, they contribute very little to the similarity index since the average of the absolute charges of these atoms is just 0.01543, but that of the main chains is 0.446, which is 28.9 times of the contribution of the side chains. As a result, the similarity between F(Phenylalanine) and A(Alanine) according to (2) is as high as 0.999383, which does not reflect their real properties. In biological systems, molecules function through both electrical properties and their space structures. Formulae (1) and (2) contain much more electrical information than structural information. In order to reflect the structural information of molecules in similarity index, we redefined R_{AB} as follows:

$$R_{AB} = \omega R^{q}_{AB} + (1 - \omega) R^{m}_{AB}, \tag{9}$$

where ω ($0.0 \leq \omega \leq 1.0$) is a weight factor adjusting relative weight between R^{q}_{AB} and R^{m}_{AB}. In the present work, R^{q}_{AB} is the same as that computed by (2) and (6). R^{m}_{AB} has the same form, but the atomic charges in (6) are replaced by atomic masses. Since the atomic masses are always positive, R^{m}_{AB} mainly reflects the space structure information of the molecules. Using this modified definition, when $\omega = 0.5$, the similarity between F and A drops down to 0.867 (see Table 3), which is closer to reality.

We applied all three algorithms to compute the similarity indices for all 210 pairs of amino acids (including 20 self comparison pairs). For each such molecular pair, the optimization procedures were performed 10 times (or cycles) starting from 10 different starting point $\overrightarrow{X}$ randomly generated in the following way:

- θ uniformly distributed in the interval $[0, \pi]$;
- ϕ and ψ uniformly distributed in the interval $[0, 2\pi]$;
- x_c, y_c, z_c uniformly distributed in $[-2.5$ Å, 2.5 Å$]$ (for the general tests) or $x_c = y_c = z_c = 0$ (for center-overlapped tests).

Table 3 presents the best similarities among the 20 amino acids obtained by our optimization method when ω in formula (9) is 0.5. We can see that all similarities are between 0.0 and 1.0. Our algorithm has found global optimizers for all 20 self comparison pairs, reflected by the fact that all elements along the diagonal are equal to 1.0.

The effectiveness of an optimization procedure can be judged by (1) *the speed of convergence* and (2) *the probability of getting global maximum* (*or minimum*). Table 4 shows the comparison of the effectiveness of different algorithms. The relative convergence speed is reflected by CPU/CY – the average CPU time it takes to complete a single cycle of optimization (average over all of 2100 optimization cycles for 210 pairs of molecules). The probability of getting global maximum (or minimum), PGMX, is measured as the ratio of the number of self comparison cycles for which the similarity index is $\geq$0.99999 over 200. From the table, we can see that the BFGS method is about six times faster than the steepest descent method and is about 30 times faster than random search. It also has a higher probability of getting the global maximizer.

To investigate the feasibility of applying the similarity index to proteins, we used our optimization method to compute the similarities for some self-compare pairs of molecules containing different number of heavy atoms ranging from 50 to 1000. The corresponding IT/CY, CPU/CY and PGMX are shown in Table 5.

As a final example, we used our method to compute the similarity between the human insulin (one kind of protein functionally related with diabetes) and pig insulin. The structures of the two molecules were obtained from PDB database (with PDB codes: 1HIU and 1WAV). The atomic charge were obtained from [12]. Both molecules contain two peptide chains (chain A contains 21 residues, and chain B 30 residues). However, the human insulin has 405 heavy atoms while the pig insulin has 403 heavy atoms. The backbone structures of the two molecules were shown in Figure 3 (left and right, in original orientation). For each of the three molecular pairs (HUM-HUM, HUM-PIG, PIG-PIG), 20 cycles of optimization were performed. The results are shown in Table 6.

4. Discussions

From Table 3 we can see that the most similar amino acid to D is E (0.869), to K is R (0.928), to F is Y (0.984), to G is A (0.985), and the pair with lowest similarity is

Table 3. The best similarities found among 20 amino acid zwitterions.

	G	A	V	L	I	S	C	T	M	F	Y	W	P	H	K	R	D	E	N	Q
G	1.000	0.985	0.920	0.894	0.897	0.949	0.907	0.920	0.835	0.829	0.797	0.765	0.913	0.827	0.704	0.669	0.764	0.719	0.871	0.836
A	0.985	1.000	0.960	0.939	0.941	0.981	0.946	0.959	0.878	0.867	0.829	0.805	0.935	0.870	0.745	0.706	0.777	0.760	0.914	0.878
V	0.920	0.960	1.000	0.960	0.982	0.977	0.972	0.986	0.922	0.913	0.874	0.866	0.948	0.923	0.654	0.746	0.813	0.805	0.955	0.924
L	0.894	0.939	0.960	1.000	0.963	0.964	0.928	0.952	0.952	0.900	0.875	0.863	0.895	0.911	0.807	0.748	0.776	0.804	0.915	0.916
I	0.897	0.941	0.982	0.963	1.000	0.954	0.972	0.987	0.921	0.924	0.887	0.872	0.927	0.938	0.783	0.757	0.823	0.821	0.954	0.937
S	0.949	0.981	0.977	0.964	0.954	1.000	0.952	0.965	0.920	0.878	0.839	0.828	0.936	0.886	0.765	0.720	0.781	0.776	0.929	0.894
C	0.907	0.946	0.972	0.928	0.972	0.952	1.000	0.980	0.893	0.952	0.915	0.894	0.945	0.958	0.742	0.764	0.850	0.828	0.969	0.938
T	0.920	0.959	0.986	0.952	0.987	0.965	0.980	1.000	0.901	0.925	0.888	0.866	0.941	0.930	0.761	0.754	0.825	0.808	0.956	0.925
M	0.835	0.878	0.922	0.952	0.921	0.920	0.893	0.901	1.000	0.893	0.874	0.894	0.857	0.910	0.803	0.776	0.759	0.822	0.883	0.915
F	0.829	0.867	0.913	0.900	0.924	0.878	0.952	0.925	0.893	1.000	0.984	0.949	0.899	0.976	0.741	0.792	0.815	0.823	0.943	0.936
Y	0.797	0.829	0.874	0.875	0.887	0.839	0.915	0.888	0.874	0.984	1.000	0.937	0.871	0.950	0.738	0.798	0.772	0.795	0.905	0.917
W	0.765	0.805	0.866	0.863	0.872	0.828	0.894	0.866	0.894	0.949	0.937	1.000	0.841	0.939	0.728	0.789	0.742	0.777	0.885	0.904
P	0.913	0.935	0.948	0.895	0.927	0.936	0.945	0.941	0.857	0.899	0.871	0.841	1.000	0.895	0.713	0.719	0.781	0.757	0.919	0.882
H	0.827	0.870	0.923	0.911	0.938	0.886	0.958	0.930	0.910	0.976	0.950	0.939	0.895	1.000	0.768	0.813	0.826	0.828	0.949	0.960
K	0.704	0.745	0.654	0.807	0.783	0.765	0.742	0.761	0.803	0.741	0.738	0.728	0.713	0.768	1.000	0.928	0.631	0.609	0.744	0.781
R	0.669	0.706	0.746	0.748	0.757	0.720	0.764	0.754	0.776	0.792	0.798	0.789	0.719	0.813	0.928	1.000	0.613	0.594	0.757	0.790
D	0.764	0.777	0.813	0.776	0.823	0.781	0.850	0.825	0.759	0.815	0.772	0.742	0.781	0.826	0.631	0.613	1.000	0.869	0.829	0.790
E	0.719	0.760	0.805	0.804	0.821	0.776	0.828	0.808	0.822	0.823	0.795	0.777	0.757	0.828	0.609	0.594	0.869	1.000	0.817	0.858
N	0.871	0.914	0.955	0.915	0.954	0.929	0.969	0.956	0.883	0.943	0.905	0.885	0.919	0.949	0.744	0.757	0.829	0.817	1.000	0.933
Q	0.836	0.878	0.924	0.916	0.937	0.894	0.938	0.925	0.915	0.936	0.917	0.904	0.882	0.960	0.781	0.790	0.790	0.858	0.933	1.000

Table 4. The comparison among different optimization procedures.

	With centers overlapped at beginning			Without centers overlapped at beginning		
	BFGS method	Steepest decent	Random search	BFGS method	Steepest decent	Random search
ITRS	96372	675192	24346760	108234	717423	23329431
EFIT	66419	534552	1071811	73260	567964	1143516
ITSR	68.9%	79.2%	4.4%	67.7%	79.2%	4.9%
IT/CY	45.9	321.5	11593.7	51.5	341.6	11109.6
CPU	625.3	4008.4	18296.0	702.6	4291.0	17618.0
CPU/CY	0.298	1.909	8.712	0.335	2.043	8.390
RLSP	6.4	1.0	0.22	6.1	1.0	0.24
PGMX	53.0%	53.5%	52.5%	52.5%	46.5%	45.0%

where:
ITRS is the number of total iterations for all of 2100 cycles;
EFIT is the effective it (made progress) out of ITRS;
ITSR is the success rate of iterations;
IT/CY is the average number of iterations for a single cycle;
CPU is the total CPU time it takes to complete all of 2100 cycles;
CPU/CY is the average CPU time it takes for a single cycle (in seconds);
RLSP is the relative converge speed;
PGMX is the probability of getting global maximum.

Table 5. IT/CY, CPU/CY, OPTSR values of the systems with different size.

	The number of atoms						
	50	80	120	200	300	500	1000
IT/CY	58.8	66.4	79.0	78.1	69.2	95.6	82.8
CPU/CY	10.7	32.4	88.1	356.0	490.8	1871.7	6586.7
PGMX	20.0%	20.0%	20.0%	30.0%	20.0%	10.0%	20.0%

R and E (0.594). These are basically consistent with their chemical and structural properties. Of course, with the different ω value in (9), we may get somewhat different results. We didn't explore what ω value is the most reasonable because that is not the main goal of the present work. From Table 4, we can see that our method and steepest descent method have similar values for PGMX. However, the

Table 6. Comparison results between human and pig insulins.

	HUM–HUM	HUM–PIG	PIG–PIG
ITRS	1345	1493	1506
EFIT	923	1094	1085
ITSR	68.6%	73.3%	72.0%
IT/CY	67.3	74.7	75.3
CPU	17589.0	19271.9	19511.8
CPU/CY	879.5	963.6	975.6
PGMX	30.0%	MXS: 0.46037	25.0%

MXS, the maximum of similarity found (4 out of 20 cycles)

BFGS method is about 6 folds faster than the steepest descent method and is about 25 folds faster than random search algorithm.

As our experience, if the centers of two molecules are overlapped before optimization, the converge speed would generally faster, especially for those with high similarities. This is not always the case. For molecules with large differences in size and structure, some times the converge speed could be faster if the two centers were separated in an appropriate distance before optimization.

To consider the possibility of applying the similarity index to proteins, we should first consider what will happen as molecular sizes increase. Generally speaking, although the number of free variables is always 6, the larger the system is, the more complex of $R_{AB}(\overrightarrow{X})$ will be, i.e. the more rough of the landscape of $R_{AB}(\overrightarrow{X})$; therefor, the less probability to find the global maximum (or minimum). Also more iteration steps are needed for convergence. The PGMX and IT/CY values in table 4 and 5 reflect these dependencies, but not as strong as a linear function. The converge speed depends on two factors: the CPU time need for a single step of iteration and the number of iteration steps needed for convergence (IT/CY). The former factor, according to (6) and (8), will mainly depends on $N_A{*}N_B$. The second factor (IT/CY) is somewhat more complex. It depends on optimization method, but might also be affected by many other situations. For example, if the initial state is near by a shallow and smooth minimum, the IT/CY might be smaller, otherwise, bigger. According to the tendency of Table 5 and suppose PGMX=0.1, using our optimization method, we could expect to get the best superimpose of two proteins each with 2000 heavy atoms in about 80 h on a Pentium Pro 200 MHz processor. Although this seems time consuming, it is at least possible.

Our computation on human insulin and pig insulin shows that similarity index can be used to superimpose two proteins through optimizing their similarity indices. Out of 20 optimizations cycles, similarity of human-human reached 1.0 for 6 times; of pig-pig, 5 times. For human-pig, we do not know the best similarity index.

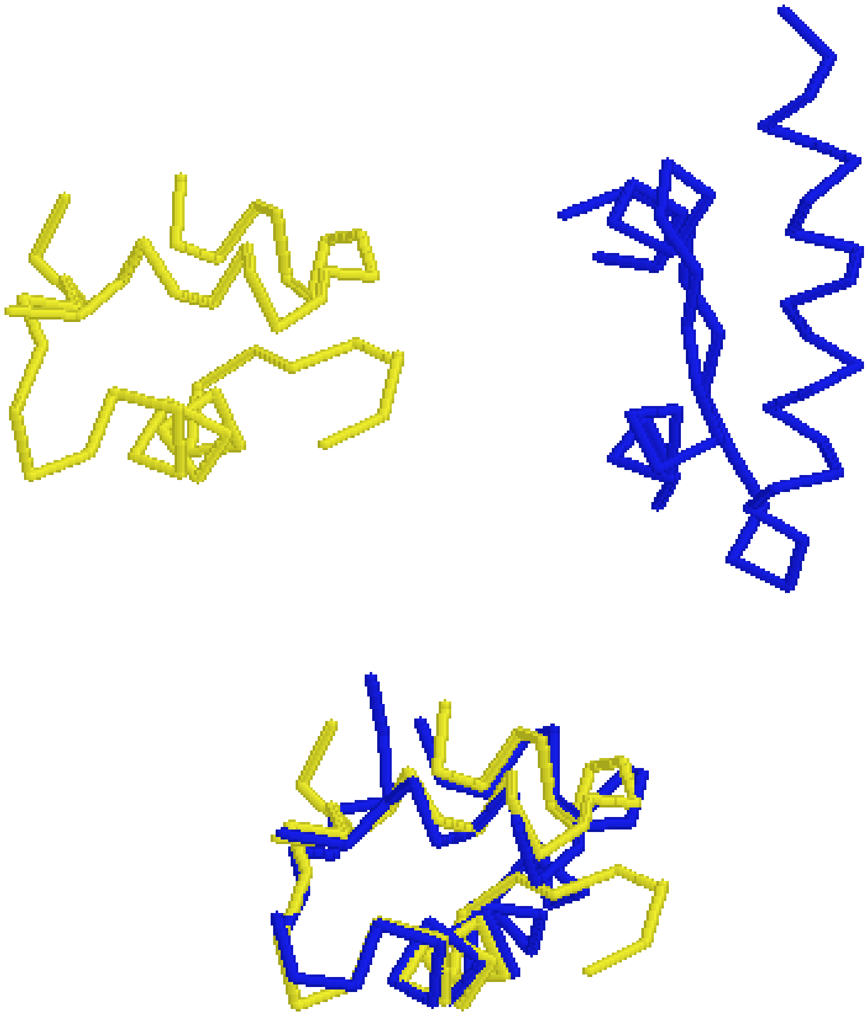

Figure 3. The structures of human insulin (left) and pig insulin (right) and their superimpose (bottom).

The best result we got is 0.46037. It was found 4 times in 20 tries. According this similarity, the two insulins were superimposed as in Figure 3 (bottom). There we see the orientation of human insulin not changed, but that of pig insulin changed.

Rather than computing atom-based derivatives as [13] did, we directly compute the similarity index derivatives with respect to six free variables. This saves the computation of resultant of forces and torques for atom-based derivatives. Hence it increases the accuracy of computation. According to our implementation, for those self comparison pairs, if start from a “good” initial states, the similarity can reach 0.999999 within 70 iterations when using derivatives with respect to six variables, but it was very difficult to reach 0.9999 by using atom-based derivatives.

Acknowledgments

L. Ling's research was supported in part by NSF grant ASC-9409285. Research of G. Xue was supported in part by the National Science Foundation grants ASC-9409285 and OSR-9350540, and by the US Army Research Office grant DAAH04-9610233.

References

1. Burt, C., Richards, W.G. and Huxley, P. (1990), The Application of Molecular Similarity Calculations, *Journal of Computational Chemistry* 11, 1139–1146.
2. Carbo, R., Leyda, L. and Arnau, M. (1980), How Similar is a Molecule to Another?, *International Journal of Quantum Chemistry* 17, 1185–1189.
3. Coleman, T., Shalloway, D. and Wu, Z. (1993), Isotropic Effective Energy Simulated Annealing Searches for Low Energy Molecular Cluster States, *Computational Optimization and Applications* 2, 145–170.
4. Dean, P.M. (ed.) (1995), *Molecular Similarity in Drug Design*, Chapman Hall, London.
5. Fletcher, R. (1987), *Practical Methods of Optimization*, 2nd ed., John Wiley, Chichester.
6. Johnson, M.A. and Maggiora, G.M. (1990), *Concepts and Application of Molecular Similarity*, Wiley, New York.
7. Hodgkin, E.E. and Richards, W.G. (1987), Molecular Similarity Based on Electrostatic Potential and Electric Field, *International Journal of Quantum Chemistry, Quantum Biology Symposium* 14, 105–110.
8. Lee, C. and Smithline, S. (1994), An Approach to Molecular Similarity Using Density Functional Theory, *Journal of Physical Chemistry* 98, 1135–1138.
9. Liu, D.C. and Nocedal, J. (1989), On the Limited Memory BFGS Method for Large Scale Optimization, *Mathematical Programming* 45, 503–528.
10. Mark, A.E., van Gunstern, W.F., King, P.M. (1994), Fundamentals of Drug Design from a Biophysical Viewpoint, *Quarterly Reviews of Biophysics* 27, 435.
11. Maranas, C.D. and Floudas, C.A. (1994), Global Minimum Potential Energy Conformations of Small Molecules, *Journal of Global Optimization* 4, 135–170.
12. McCammon, J.A., Wolynes, P.G. and Karplus, M. (1979), Picosecond Dynamics of Tyrosine Side Chains in Proteins, *Biochemistry* 18, 927–942.
13. McMahon, A.J. and King, P.M. (1997), Optimization of Carbo Molecular Similarity Index Using Gradient Methods, *Journal of Computational Chemistry* 18, 151–158.
14. Pardalos, P.M., Shalloway, D. and Xue, G.L. (1994), Optimization Methods for Computing Global Minima of Non-convex Potential Energy Functions, *Journal of Global Optimization* 4, 117–133.
15. Richard, A.M. and Rabinowitz, J.R. (1987), Modified Molecular Charge Similarity Indices for Choosing Molecular Analogues, *International Journal of Quantum Chemistry* 31, 309–323.
16. Shalloway, D. (1992), Application of the Renormalization Group to Deterministic Global Minimization of Molecular Conformation Energy Functions, *Journal of Global Optimization* 2, 281–311.
17. Strnad, M. and Ponec, R. (1994), Novel Approach to Molecular Similarity: Second-order Similarity Indices from Geminal Expansion of Pair Densities, *International Journal of Quantum Chemistry* 49, 35–43.