Trust Region Superposition Methods for Protein Alignment [*]

R. Andreani [†] J. M. Martínez [‡] L. Martínez [§]

June 7, 2007

Abstract

Protein Alignment is a challenging applied Optimization problem. Superposition methods are based on the maximization of a score function with respect to rigid-body modifications of relative positions. The problem of score maximization can be modeled as a continuous nonsmooth optimization problem (LOVO). This allows one to define practical and convergent methods that produce monotone increase of the score. In this paper, trust region methods are introduced for solving the problem. Numerical results are presented. Computer software related to the LOVO approach for Protein Alignment is available in `www.ime.unicamp.br/∼martinez/lovoalign`.

1 Introduction

Proteins are large organic compounds formed by chains of α-amino acids bound by peptide bonds. They are essential parts of all living organisms and participate in most cellular processes. Hormone recognition and transport, catalysis, transcription regulation, photosynthesis, cellular respiration, and many other fundamental mechanisms of life are protein-mediated. Proteins can work together to achieve a particular function, and can bind to different chemical structures to be functional [42].

The sequence of amino acids in a protein is defined by a gene. This sequence is known as the *Primary Structure* of a protein. Each amino acid has particular chemical characteristics, but contributes to the main chain of the protein with identical substructures formed by one nitrogen and two carbon atoms. One of these carbon atoms is known as the Cα atom. Roughly speaking, the 3D coordinates of the Cα atoms is known as the *Tertiary Structure* of a protein. Protein structures can be determined by experimental methods, such as X-ray crystallography or Nuclear Magnetic Resonance. A large collection containing atom coordinates for most known

[*]This work was supported by PRONEX-Optimization 76.79.1008-00, FAPESP (Grants 06/53768-0, 05/56773-1 and 02-14203-6) and PRONEX CNPq/ FAPERJ 26/171.164/2003-APQ1. This paper is based on the talk given by J. M. Martínez at the International Conference on Numerical Analysis and Optimization, held in Beijing, and dedicated to the 70th birthday of Prof. M. J. D. Powell in September 2006.

[†]Department of Applied Mathematics, IMECC-UNICAMP, State University of Campinas, CP 6065, 13081-970 Campinas SP, Brazil. E-mail: andreani@ime.unicamp.br

[‡]Department of Applied Mathematics, IMECC-UNICAMP, State University of Campinas, CP 6065, 13081-970 Campinas SP, Brazil. E-mail: martinez@ime.unicamp.br

[§]Institute of Chemistry, State University of Campinas. E-mail: lmartinez@iqm.unicamp.br.

proteins is the Protein Data Bank (PDB) [6], which contains around 35000 structures. This number increases every year.

During evolution, mutations promote changes in the primary structure of a protein by introducing modifications in the genetic code. These mutations may persist in a population if they do not result in impaired protein function. The function of different proteins may be the same in spite of different sequences of amino acids when they share the same overall three dimensional structure. Therefore, the *classification* of 3D structures is useful to determine the function of the proteins and to provide hints on evolutionary mechanisms.

The main ingredient of the classification procedure is the comparison (*alignment*) between two structures. When a new protein structure is obtained, or when a protein structure is conjectured, its comparison with the whole data bank and consequent classification is often used for functional classifications [14].

The degree of similarity between two proteins is usually given by a *score*. From this score, a distance-like function is usually derived and the set of distances is frequently used to produce *structure maps*. A structure map is a 2D or 3D representation of the whole space of proteins [15]. In a structure map, each protein is a point and the distance between two of these points reflects the similarity given by the score. Multidimensional scaling [16, 17, 41] and Kernel methods [28] are useful tools for building the 3D representation that comes from scores. The Structure Space Map developed in [16, 17] provides good predictions of function similarities in many cases.

The primary sequence of amino acids determines the structure of a protein. Protein folding is the molecular mechanism by which a protein achieves its tertiary structure from an unfolded sequence. Some general aspects of protein folding mechanisms are now being elucidated [35], but the prediction of structure from sequence remains one of the greatest challenges of contemporary biochemistry. Methods for structural modeling based on the sequence of amino acids exist, and are frequently based on sequence similarities with proteins with known structure. The evaluation of the quality of the models requires a measure of their potential energy and of their similarity with the structural references used [37]. Therefore, a score must be a reliable measure of similarity not only between known structures but also between potential ones.

We will see that the score that measures the similarity between two proteins may be seen as the maximum of a (continuous-nonsmooth) function in the space of relative positions (displacements). The reliability of the score depends on the accuracy in which we are able to obtain this maximum, therefore robust and fast algorithms are necessary. Algorithms for obtaining the *global* maximum may be defined but are not affordable for the present computer facilities [23].

In this paper we rely on the mathematical characterization of the Protein Alignment problem given in [2] (see, also, [4]). Line-search algorithms that converge to first-order stationary points were defined in [2, 4]. Here we introduce a trust region approach [7, 29, 36] to define second-order convergent algorithms.

This paper is organized as follows. In Section 2, the Protein Alignment problem is formulated as a Low Order Value Optimization problem. In Section 3 we define a trust region method for solving LOVO and we prove convergence. In Section 4 we present numerical results. Conclusions are given in Section 5.

Notation

The symbol $\|\cdot\|$ will denote the Euclidean norm.

If the symmetric matrix A is positive semidefinite, we denote $A \succcurlyeq 0$. Analogously, if A is positive definite, we denote $A \succ 0$.

We denote $I\!N = \{0, 1, 2, \ldots\}$.

The Euclidean ball with center x and radius ε is denoted $\mathcal{B}(x, \varepsilon)$.

2 Formulation

Let $\mathcal{Q} = \{Q_1, \ldots, Q_N\} \subset I\!R^{n_q}$, $\mathcal{P} = \{P_1, \ldots, P_M\} \subset I\!R^{n_p}$. The goal is to find a transformation $D : I\!R^{n_q} \to I\!R^{n_p}$ such that some subset of $\{D(Q_1), \ldots, D(Q_N)\}$ fits some subset of $\mathcal{P}$. In Protein Alignment, D generally represents rigid-body displacements but more general transformations can be considered. For example, assume that $n_q = 3, n_p = 2$ and that $\mathcal{P}$ is the set of possible "shadows" of the points in $\mathcal{Q}$. In that case, one could wish to find the rigid-body displacement of $\mathcal{Q}$ such that a subset of the two-dimensional points represented by the (x, y) coordinates of the displaced $\mathcal{Q}$ fits a subset of $\mathcal{P}$ in the best possible way. In that case, D would be the composition of a rigid-body movement with a projection. A lot of examples of this general problem can be given, from tissue recognition to security systems [3]. We will denote $\mathcal{D}$ the set of *admissible transformations*.

Let $\mathcal{C}$ be the set of *admissible correspondences* between nonempty subsets of $\{1, \ldots, N\}$ and $\{1, \ldots, M\}$. (Sometimes, admissible correspondences must be bijective, sometimes monotonicity will be required.)

Each element $\Phi \in \mathcal{C}$ is a function

$$\Phi : A \to B,$$

where $A \subset \{1, \ldots, N\}$ and $B \subset \{1, \ldots, M\}$. Given $\Phi \in \mathcal{C}$ and a transformation D, an associated score $S(D, \Phi) \geq 0$ is assumed to be defined. This score should reflect the degree of spatial similarity between the sets $\{D(Q_a)\}_{a \in A}$ and $\{P_b\}_{b \in B}$.

The goal of the general alignment problem is to maximize, both with respect to Φ and with respect to D, the score $S(D, \Phi)$. In other words, we wish to solve the problem:

$$\text{Maximize}_{D \in \mathcal{D}} \ \text{Maximum}_{\Phi \in \mathcal{C}} \ S(D, \Phi). \tag{1}$$

Since $\mathcal{C}$ is a finite set (say, $\mathcal{C} = \{\Phi_1, \ldots, \Phi_m\}$) we may write (1) in the form

$$\text{Maximize}_{D \in \mathcal{D}} \ \text{Maximum} \ \{S(D, \Phi_1), \ldots, S(D, \Phi_m)\}. \tag{2}$$

Protein Alignment is a particular case of the situation explained above. The goal is to find similarities between two proteins $\mathcal{P}$ and $\mathcal{Q}$, represented by the coordinates of their Cα atoms. The similarity is measured by a *score*. Several scores have been proposed in the protein literature. One of them is the *Structal* Score [13, 40]. Assume that the 3D-coordinates of the Cα atoms of protein $\mathcal{P}$ (in angstroms) are $P_1, \ldots, P_M$ and the coordinates of the Cα atoms of protein $\mathcal{Q}$ are $Q_1, \ldots, Q_N$. Under the rigid-body displacement D, the coordinates of the displaced protein $\mathcal{Q}$ are, therefore, $D(Q_1), \ldots, D(Q_N)$. Assume that Φ is a monotone bijection between a subset of $\{1, \ldots, N\}$ and a subset of $\{1, \ldots, M\}$. (We mean that $i < j \Rightarrow \Phi(i) < \Phi(j)$.) The *Structal* score associated to the displacement D and the bijection Φ is:

$$S(D, \Phi) = \sum \frac{20}{1 + \|P_k - D(Q_{\Phi(k)})\|^2/5}, -10 \times gaps, \tag{3}$$

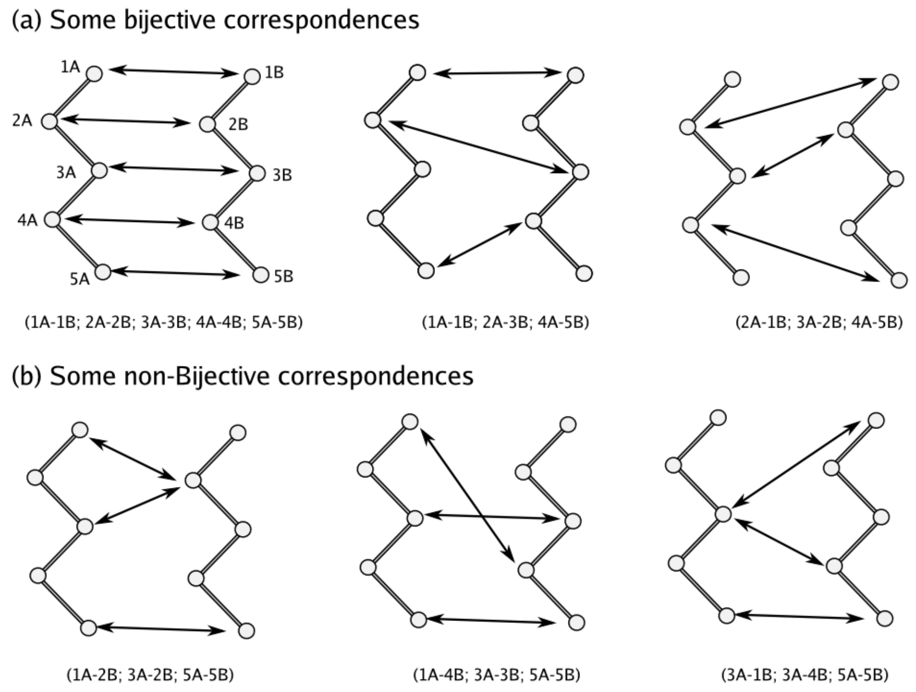

Figure 1: Examples of correspondences that form the Φ domain: (a) Bijective correspondences and (b) Non-bijective correspondences, valid only for NB methods.

where the $\sum$ symbol involves the pairs $(k, \Phi(k))$ defined by the bijection and *gaps* is the number of cases in which at least one of the following situations occur:

- $\Phi(k)$ is defined, there exists $\ell > k$ such that $\Phi(\ell)$ is defined, but $\Phi(\ell + 1)$ is not defined;
- $\Phi^{-1}(k)$ is defined, there exists $\ell > k$ such that $\Phi^{-1}(\ell)$ is defined, but $\Phi^{-1}(\ell + 1)$ is not defined.

In Figure 1 we give examples of bijective and nonbijective correspondences. The concept of *gap* applies only to the bijective case. The bijection on the left has no gaps. The central bijection has two gaps and the bijection on the right has one gap.

The Alignment Problem associated with the *Structal* score consists of finding Φ and D such that $S(D, \Phi)$ is maximal. A global optimization procedure for achieving this objective was given in [23]. However, this method is not computationally affordable and, in practice, an heuristic procedure called `Structal Method` [13, 40] is generally used. In [22], the `Structal Method` was reported as the best available practical algorithm for protein alignment. Each iteration of the `Structal Method` consists of two steps:

1. Update Φ: Given the positions $P_1, \ldots, P_M$ and $D(Q_1), \ldots, D(Q_N)$, the monotone bijection Φ that maximizes the score (fixing D) is computed using Dynamic Programming [33].

2. Update D: Assume that the graph of Φ is $\{(k_1, \Phi(k_1)), \ldots, (k_s, \Phi(k_s))\}$. Then, the rigid-body displacement that minimizes $\sum_{\ell=1}^{s} \|P_{k_\ell} - D(Q_{\Phi(k_\ell)})\|^2$ is computed.

The computation of D at the second step of the `Structal Method` involves the solution of the well known Procrustes problem [18, 19]. The main drawback of the `Structal Method` is that the Update-Φ step aims the optimization of a function (the *Structal* score) with respect to Φ and the Update-D step involves the optimization of a different function (the sum of squared distances) with respect to D. This may lead to oscillation [2].

The `Structal Method` is the most efficient *Superposition* method for Protein Alignment. Superposition methods are iterative algorithms whose main iteration has two phases:

1. Update Φ: Given the positions $P_1, \ldots, P_M$ and $D(Q_1), \ldots, D(Q_N)$, the admissible correspondence Φ that maximizes S (fixing D) is computed.

2. Update D: Assume that the graph of Φ is $\{(k_1, \Phi(k_1)), \ldots, (k_s, \Phi(k_s))\}$. Then, a rigid-body displacement that *presumably improves the score* associated to this correspondence is computed.

In [2] the set of admissible correspondences has been defined in two different ways: in the first case, an admissible correspondence must be a monotone bijection (as in the `Structal Method`) and, in the second case, admissible correspondences are mere functions between subsets of $\mathcal{Q}$ and $\mathcal{P}$ (bijective or not). The *Structal* score is used in both cases, but in the second case the gap-term is not included. In both cases, for a fixed correspondence Φ, the function $S(D, \Phi)$ is a continuous and smooth function of D. As a consequence, Protein Alignment methods that improve a score at every iteration are defined. The typical iteration of these methods has two phases, as in general superposition methods. The first phase is as stated above. In the second phase we find a new displacement D that *improves the score* associated to Φ by means of a continuous optimization method.

3 Low Order Value Optimization Algorithm

Assume that $f_i : I\!R^n \to I\!R$, $i = 1, \ldots, m$. Define, for all $x \in I\!R^n$,

$$f_{min}(x) = \min\{f_1(x), \ldots, f_m(x)\}.$$

We will consider the optimization problem

$$\text{Minimize } f_{min}(x). \tag{4}$$

This is a Low Order Value optimization problem as defined in [3]. Let us identify the transformation D with the set of parameters by means of which D is defined (rotation angles and translation in the case of rigid-body displacements). Writing $x = D$, $f_i(x) = -S(D, \Phi_i)$, we observe that (2) is a particular case of (4). Therefore, the Protein Alignment problems defined in Section 2 are Order Value Optimization problems in the sense of (4). We will assume that the second derivatives of f_i are Lipschitz-continuous on a sufficiently large set, for all $i = 1, \ldots, m$. This requirement is clearly fulfilled when S is the *Structal* score.

For all $x \in I\!\!R^n$, we define:

$$I_{min}(x) = \{i \in \{1, \ldots, m\} \mid \quad f_i(x) = f_{min}(x)\}.$$

Here we will define a Newtonian trust region method for solving (4). This method will be applied to the Protein Alignment problem.

Before defining the main algorithm, let us give a technical lemma, which will be useful both in the well-definiteness proof and in the convergence proof.

Lemma 3.1. *Let $\{x_j\}$ be a sequence that converges to $\widehat{x} \in I\!\!R^n$ and let $f : I\!\!R^n \to I\!\!R$ possess Lipschitz-continuous second derivatives on a open and convex set that contains $\{x_j\}$. We define ψ^j, the second order quadratic approximation of $f(x)$ by:*

$$\psi^j(x) \equiv f(x_j) + \nabla f(x_j)^T(x - x_j) + \frac{1}{2}(x - x_j)^T \nabla^2 f(x_j)(x - x_j).$$

Assume that $\{\Delta_j\}$ is a sequence of positive numbers that tends to zero and define $\overline{x}_j$ as a global minimizer of $\psi^j(x)$ subject to $\|x - x_j\| \leq \Delta_j$. Finally, assume that the condition

$$\nabla f(\widehat{x}) = 0 \textit{ and } \nabla^2 f(\widehat{x}) \succcurlyeq 0 \tag{5}$$

does not hold, and define

$$\rho_j = \frac{f(\overline{x}_j) - f(x_j)}{\psi^j(\overline{x}_j) - \psi^j(x_j)}. \tag{6}$$

Then,

$$\lim_{j \to \infty} \rho_j = 1.$$

Proof. Since $\widehat{x}$ does not satisfy (5), we have that

$$\nabla f(\widehat{x}) \neq 0 \tag{7}$$

or

$$\nabla f(\widehat{x}) = 0 \text{ and } \nabla^2 f(\widehat{x}) \not\succcurlyeq 0\ . \tag{8}$$

By the continuity of ∇f and the convergence of $\{x_j\}$, if (7) takes place, we have that $\nabla f(x_j) \neq 0$ for j large enough. In the case of (8), $\nabla^2 f(\widehat{x})$ has a negative eigenvalue, therefore, $\nabla^2 f(x_j) \not\succcurlyeq 0$ for j large enough. Therefore, for j large enough, x_j *is not* a solution of the problem that defines $\overline{x}_j$. So, there exists $j_1 \in I\!\!N$ such that, for all $j \geq j_1$,

$$\psi^j(\overline{x}_j) - \psi^j(x_j) < 0.$$

This implies that ρ_j is well defined (the denominator in (6) is not null) for $j \geq j_1$.

Assume, first, that $\nabla f(\widehat{x}) \neq 0$. Then, there exists $d \in I\!\!R^n$ be such that $\|d\| = 1$ and

$$\nabla f(\widehat{x})^T d < 0. \tag{9}$$

Since $\|\Delta_j d\| = \Delta_j$, for all $j \geq j_1$ we have:

$$\psi^j(\overline{x}_j) \le \psi^j(x_j + \Delta_j d) = \psi^j(x_j) + \Delta_j \nabla f(x_j)^T d + \frac{\Delta_j^2}{2} d^T \nabla^2 f(x_j) d\ .$$

Therefore,

$$\psi^j(\overline{x}_j) - \psi^j(x_j) \le \Delta_j \nabla f(x_j)^T d + \frac{\|\nabla^2 f(x_j)\|\ \Delta_j^2}{2},$$

so,

$$\frac{\psi^j(\overline{x}_j) - \psi^j(x_j)}{\Delta_j} \le \nabla f(x_j)^T d + \frac{\|\nabla^2 f(x_j)\|}{2}\ \Delta_j.$$

Therefore, by (9) and the continuity of ∇f, there exists $j_2 \ge j_1$ such that for all $j \ge j_2$,

$$\frac{\psi^j(\overline{x}_j) - \psi^j(x_j)}{\Delta_j} \le \frac{\nabla f(\widehat{x})^T d}{2} \equiv a < 0. \tag{10}$$

By Taylor's theorem and (10), we have:

$$\begin{aligned}
|\rho_j - 1| &= \left| \frac{f(\overline{x}_j) - f(x_j) - [\psi^j(\overline{x}_j) - \psi^j(x_j)]}{\psi^j(\overline{x}_j) - \psi^j(x_j)} \right| = \left| \frac{f(\overline{x}_j) - \psi^j(\overline{x}_j)}{\psi^j(\overline{x}_j) - \psi^j(x_j)} \right| \\
&= \left| \frac{f(\overline{x}_j) - f(x_j) - \nabla f(x_j)^T(\overline{x}_j - x_j) - \frac{1}{2}(\overline{x}_j - x_j)^T \nabla^2 f(x_j)(\overline{x}_j - x_j)}{\psi^j(\overline{x}_j) - \psi^j(x_j)} \right| \\
&\le o(\Delta_j^2)/(-a\Delta_j) \to 0.
\end{aligned}$$

Therefore, $\lim\limits_{j\to\infty} \rho_j = 1$.

Assume now that (8) holds. Then, there exists $d \in I\!R^n$ such that $\|d\| = 1$ and

$$d^T \nabla^2 f(\widehat{x}) d < 0.$$

For all $j \ge j_1$, define $d_j = d$ if $\nabla f(x_j)^T d \le 0$ and $d_j = -d$ if $\nabla f(x_j)^T d > 0$.

Since $\|\Delta_j d_j\| = \Delta_j$, we have:

$$\psi^j(\overline{x}_j) \le \psi^j(x_j + \Delta_j d_j) \le \psi^j(x_j) + \frac{\Delta_j^2}{2} d_j^T \nabla^2 f(x_j) d_j.$$

Therefore, since $d_j^T \nabla^2 f(x_j) d_j = d^T \nabla^2 f(x_j) d$,

$$\frac{\psi^j(\overline{x}_j) - \psi^j(x_j)}{\Delta_j^2} \le \frac{1}{2}\ d^T \nabla^2 f(x_j) d.$$

Hence, by the continuity of $\nabla^2 f(x)$, there exists $j_3 \in I\!N$ such that for all $j \ge j_3$,

$$\frac{\psi^j(\overline{x}_j) - \psi^j(x_j)}{\Delta_j^2} \le \frac{1}{4}\ d^T \nabla^2 f(\widehat{x}) d \equiv b < 0\ . \tag{11}$$

Therefore,

$$|\rho_j - 1| = \left| \frac{f(\overline{x}_j) - \psi^j(\overline{x}_j)}{\psi^j(\overline{x}_j) - \psi^j(x_j)} \right| \le \frac{o(\|\overline{x}_j - x_j\|^2)}{\Delta_j^2 |b|} \le \frac{o(\Delta_j^2)}{\Delta_j^2} \to 0.$$

Then, $\lim_{j\to\infty} \rho_j = 1$. This completes the proof. □

Algorithm 3.1.

Assume that $\Delta_{\min} > 0$, $\sigma_1, \sigma_2 \in (0,1)$ (with $\sigma_1 < \sigma_2$) and $\alpha \in (0,1)$ are given independently of k. Let $x_0 \in I\!R^n$ be the initial approximation to the solution of (4).

For all $k \in I\!N$, $i \in \{1, \dots, m\}$, $x \in I\!R^n$, we define:

$$\psi_i^k(x) = f_i(x_k) + \nabla f_i(x_k)^T (x - x_k) + \frac{1}{2}(x - x_k)^T \nabla^2 f_i(x_k)(x - x_k).$$

Step 0. Initialize $k \leftarrow 0$.

Step 1. Choose $\nu(k) \in I_{min}(x_k)$. If

$$\|\nabla f_{\nu(k)}(x_k)\| = 0 \quad \text{and} \quad \nabla^2 f_{\nu(k)}(x_k) \succcurlyeq 0, \tag{12}$$

terminate the execution of the algorithm.

Step 2. Newton trust region step.

Step 2.1. Choose $\Delta \ge \Delta_{\min}$.

Step 2.2. Compute $\overline{x}(\Delta)$, a global minimizer of $\psi_{\nu(k)}^k(x)$ subject to $\|x - x_k\| \le \Delta$.

Step 2.3. If

$$f_{min}(\overline{x}(\Delta)) \le f_{min}(x_k) + \alpha[\psi_{\nu(k)}^k(\overline{x}(\Delta)) - \psi_{\nu(k)}^k(x_k)], \tag{13}$$

define $x_{k+1} = \overline{x}(\Delta)$, $\Delta_k = \Delta$, $k \leftarrow k+1$ and go to Step 1.
Else, choose $\Delta_{\text{new}} \in [\sigma_1 \|\overline{x}(\Delta) - x_k\|, \sigma_2 \Delta]$, $\Delta \leftarrow \Delta_{\text{new}}$ and go to Step 2.2.

In Theorem 3.1, we prove that, if (12) does not hold at x_k, then the iteration that computes x_{k+1} is well defined. That is, after a finite number of reductions of Δ, one obtains x_{k+1} such that the sufficient descent criterion (13) holds.

In the rest of the paper we will assume that, for all $i = 1, \dots, m$, $\nabla^2 f_i(x)$ is Lipschitz-continuous in an open and convex set that contains all the iterates generated by Algorithm 3.1.

Theorem 3.1. *If $x_k, \nu(k)$ do not satisfy (12), then, x_{k+1} is well defined and satisfies*

$$f_{min}(x_{k+1}) \le f_{min}(x_k) + \alpha[\psi_{\nu(k)}^k(x_{k+1}) - \psi_{\nu(k)}^k(x_k)] < f_{min}(x_k). \tag{14}$$

Proof. Assume that $x_k, \nu(k)$ do not satisfy (12). Define $i = \nu(k)$. Then,

$$\nabla f_i(x_k) \ne 0 \tag{15}$$

or

$$\nabla f_i(x_k) = 0 \text{ and } \nabla^2 f_i(x_k) \not\succeq 0 \ . \tag{16}$$

Observe that

$$\psi_i^k(x_k) = f_i(x_k) = f_{min}(x_k). \tag{17}$$

For all $\Delta > 0$, we define $\overline{x}(\Delta)$ as a minimizer of $\psi_i^k(x)$ subject to $\|x - x_k\| \leq \Delta$. By (15) and (16), x_k is not a minimizer of this subproblem.

Define, for all $\Delta > 0$,

$$\rho(\Delta) = \frac{f_i(\overline{x}(\Delta)) - f_i(x_k)}{\psi_i^k(\overline{x}(\Delta)) - \psi_i^k(x_k)}.$$

By Lemma 3.1, if $\{\Delta_j\}$ is a sequence of positive numbers that tends to zero, we have that

$$\lim_{j\to\infty} \rho(\Delta_j) = 1.$$

Therefore, $\lim\limits_{\Delta\to 0} \rho(\Delta) = 1$. Since $f_{min}(\overline{x}(\Delta)) \leq f_i(\overline{x}(\Delta))$, this implies that for Δ sufficiently small, (14) will be fulfilled. So, the proof is complete. □

Remark. Theorem 3.1 says that, if Algorithm 3.1 terminates at x_k, then there exists $i \in I_{min}(x_k)$ such that x_k is a second-order stationary point of f_i. The reciprocal is not true. For example, define, with $n = 1, m = 2$, $f_1(x) = x$, $f_2(x) = x^2$. Clearly, 0 is a second order stationary point of f_2. However, if one chooses $\nu(k) = 1$, the algorithm will not stop and, in fact, it will find a better point such that $f_{min}(x) < f_{min}(0)$.

Theorem 3.2 *Assume that, for an infinite set of indices $K \subset I\!N$, we have that $\lim_{k\in K} = x_*$, where $\{x_k\}$ is an infinite sequence generated by Algorithm 3.1. Then:*

1. *If $i \in \{1, \ldots, m\}$ is such that $\nu(k) = i$ for infinitely many indices $k \in K$, then*

$$\nabla f_i(x_*) = 0 \text{ and } \nabla^2 f_i(x_*) \succeq 0. \tag{18}$$

2. *There exists $i \in I_{min}(x_*)$ such that $\nabla f_i(x_*) = 0$ and $\nabla^2 f_i(x_*) \succeq 0$.*

Proof. The sequence $\{\Delta_k\}_{k\in K}$ satisfies one of the following possibilities:

$$\liminf_{k\in K} \Delta_k = 0 \tag{19}$$

or

$$\{\Delta_k\}_{k\in K} \text{ is bounded away from } 0. \tag{20}$$

Assume, initially, that (19) holds. Then, there exists an infinite set of indices $K_1 \subset K$ such that

$$\lim_{k\in K_1} \Delta_k = 0 \ . \tag{21}$$

Therefore, there exists $k_1 \in I\!N$ such that $\Delta_k < \Delta_{\min}$ for all $k \in K_2$, where $K_2 \equiv \{k \in K_1 \mid k \geq k_1\}$. Since, at each iteration, the initial trial trust region radius is greater than or equal to Δ_{min}, it turns out that, for all $k \in K_2$, there exist $\overline{\Delta}_k$ and $\overline{x}(\overline{\Delta}_k)$ such that $\overline{x}(\overline{\Delta}_k)$ is global solution of

$$\begin{array}{c} \text{Minimize } \ \psi_i^k(x) \\ \|x - x_k\| \leq \overline{\Delta}_k \end{array} \tag{22}$$

but

$$f_i(\overline{x}(\overline{\Delta}_k)) \geq f_{min}(\overline{x}(\overline{\Delta}_k)) > f_i(x_k) + \alpha[\psi_i^k(\overline{x}(\overline{\Delta}_k)) - \psi_i^k(x_k)] \ . \tag{23}$$

Clearly, (22) implies that $\overline{x}(\overline{\Delta}_k)$ is global solution of:

$$\begin{array}{c} \text{Minimize } \ \psi_i^k(x) \\ \|x - x_k\| \leq \|\overline{x}(\overline{\Delta}_k) - x_k\|. \end{array} \tag{24}$$

By the definition of Δ_k at Step 2 of Algorithm 3.1, we have:

$$\Delta_k > \sigma_1 \ \|\overline{x}(\overline{\Delta}_k) - x_k\| \ . \tag{25}$$

Therefore, by (21) e (25) we have that

$$\lim_{k \in K_3} \|\overline{x}(\overline{\Delta}_k) - x_k\| = 0 \ . \tag{26}$$

Define

$$\rho_k = \frac{f_i(\overline{x}(\overline{\Delta}_k)) - f_i(x_k)}{\psi_i^k(\overline{x}(\overline{\Delta}_k)) - \psi_i^k(x_k)} \ . \tag{27}$$

By Lemma 3.1, if (18) does not hold, we have that $\lim_{k \in K_2} \rho_k = 1$, which contradicts (23). Therefore, (18) is proved in the case (19).

Let us consider the possibility (20). Since $f_{min}(x_{k+1}) \leq f_{min}(x_k)$ for all k, and $\lim_{k \in K} x_k = x_*$, by the continuity of f_{min} we have that $\lim_{k \to \infty}[f_{min}(x_{k+1}) - f_{min}(x_k)] = 0$. Therefore, by (14),

$$\lim_{k \in K} (\psi_i^k(x_{k+1}) - \psi_i^k(x_k)) = 0 \ . \tag{28}$$

Define $\underline{\Delta} = \inf\limits_{k \in K_1} \Delta_k > 0$ and let $\widehat{x}$ be a global solution of

$$\begin{array}{c} \text{Minimize } \ \nabla f_i(x_*)^T(x - x_*) + \frac{1}{2}(x - x_*)^T \nabla^2 f_i(x_*)(x - x_*) \\ \|x - x_*\| \leq \underline{\Delta}/2 \ . \end{array} \tag{29}$$

Let $k_3 \in I\!N$ such that

$$\|x_k - x_*\| \leq \underline{\Delta}/2 \tag{30}$$

for all $k \in K_4 \equiv \{k \in K \mid k \geq k_3\}$.

By (29) and (30), for all $k \in K_4$, we have:

$$\|\widehat{x} - x_k\| \leq \underline{\Delta} \leq \Delta_k. \tag{31}$$

Therefore, since x_{k+1} is a global minimizer of $\psi_i^k(x)$ subject to $\|x - x_k\| \le \Delta_k$, we get:

$$\psi_i^k(x_{k+1}) \le \psi_i^k(\widehat{x}) = \psi_i^k(x_k) + \nabla f_i(x_k)^T(\widehat{x} - x_k) + \frac{1}{2}(\widehat{x} - x_k)^T \nabla^2 f_i(x_k)(\widehat{x} - x_k). \qquad (32)$$

So,

$$\psi_i^k(x_{k+1}) - \psi_i^k(x_k) \le \nabla f_i(x_k)^T(\widehat{x} - x_k) + \frac{1}{2}(\widehat{x} - x_k)^T \nabla^2 f_i(x_k)(\widehat{x} - x_k)\ . \qquad (33)$$

By (28), taking limits in (33) for $k \in K_3$, we have that:

$$0 \le \nabla f_i(x_*)^T(\widehat{x} - x_*) + \frac{1}{2}(\widehat{x} - x_*)^T \nabla^2 f_i(x_*)(\widehat{x} - x_*).$$

Therefore x_* is a global minimizer of (29) for which the constraint $\|x - x_*\| < \underline{\Delta}/2$ is inactive. This implies that $\nabla f_i(x_*) = 0$ and $\nabla^2 f_i(x_*) \succcurlyeq 0$.

So, the first part of the thesis is proved.

Now, let us prove the second part of the thesis. Since $\{1, \dots, m\}$ is finite, there exists $i \in \{1, \dots, m\}$ such that $i = \nu(k)$ for infinitely many indices $k \in K_1 \subset K$. So, for all $k \in K_1$,

$$f_i(x_k) \le f_j(x_k) \ \ \forall \ \ j \in \{1, \dots, m\}.$$

Taking limits in the previous inequality and using the first part of the thesis, we get:

$$f_i(x_*) \le f_j(x_*) \ \ \forall \ \ j \in \{1, \dots, m\}.$$

Therefore, $i \in I_{min}(x_*)$. □

Assumption A1. We say that this assumption holds at x_* if, for all $i \in I_{min}(x_*)$ such that $\nabla f_i(x_*) = 0$, we have that $\nabla^2 f_i(x_*) \succ 0$.

Lemma 3.2. *Assume that x_* is a limit point of a sequence generated by Algorithm 3.1 and that Assumption A1 holds at x_*. Then, there exists $\varepsilon > 0$ such that the reduced ball $\mathcal{B}(x_*, \varepsilon) - \{x_*\}$ does not contain limit points of $\{x_k\}$.*

Proof. If $i \in I_{min}(x_*)$ and $\nabla f_i(x_*) = 0$, we have, by Assumption A1, that $\nabla^2 f_i(x_*)$ is positive definite. Therefore, by the Inverse Function Theorem, $\nabla f_i(x) \ne 0$ for all $x \ne x_*$ in a neighborhood of x_*.

If $i \in I_{min}(x_*)$ and $\nabla f_i(x_*) \ne 0$, then $\nabla f_i(x) \ne 0$ in a neighborhood of x_*.

Finally, if $i \notin I_{min}(x_*)$, we have that $f_i(x_*) > f_{min}(x_*)$. So, $f_i(x) > f_{min}(x)$ and $i \notin I_{min}(x)$ for all x in a neighborhood of x_*.

Therefore, there exists $\varepsilon > 0$ such that $\nabla f_i(x) \ne 0$ whenever $i \in I_{min}(x)$ and $x \in \mathcal{B}(x_*, \varepsilon) - \{x_*\}$. Therefore, by Theorem 3.2, x cannot be a limit point of a sequence generated by Algorithm 3.1, for all $x \in \mathcal{B}(x_*, \varepsilon) - \{x_*\}$. □

Lemma 3.3. *Suppose that x_* satisfies Assumption A1 and $\lim_{k \in K} x_k = x_*$, where $\{x_k\}$ is a sequence generated by Algorithm 3.1 and K is an infinite subset of indices. Then,*

$$\lim_{k \in K} \|x_{k+1} - x_k\| = 0.$$

Proof. Let I be the set of integers in $\{1, \ldots, m\}$ such that $i = \nu(k)$ for infinitely many indices $k \in K$. Since $f_i(x_k) = f_{min}(x_k)$ infinitely many times, we obtain, taking limits, that $i \in I_{min}(x_*)$ for all $i \in I$. Therefore, by Theorem 3.2, $\nabla f_i(x_*) = 0$ and, consequently, $\lim_{k \in K} \nabla f_i(x_k) = 0$ for all $k \in I$. This implies that

$$\lim_{k \in K} \nabla f_{\nu(k)}(x_k) = 0. \tag{34}$$

Moreover, by Assumption A1, $\nabla^2 f_i(x_k) \succ 0$ for all $i \in I$. So, by the continuity of the Hessians, there exists $c > 0$ such that, for all $\lambda \geq 0$ and $k \in K$ large enough:

$$\|[\nabla^2 f_{\nu(k)}(x) + \lambda I]^{-1}\| \leq \|\nabla^2 f_{\nu(k)}(x)^{-1}\| \leq 2 \max_{i \in I} \|\nabla^2 f_i(x_*)^{-1}\| = c. \tag{35}$$

Now, x_{k+1} is a solution of

$$\text{Minimize } \psi^k_{\nu(k)}(x) \text{ subject to } \|x - x_k\| \leq \Delta_k \tag{36}$$

Therefore, by the KKT conditions of (36),

$$\begin{gathered} (\nabla^2 f_{\nu(k)}(x_k) + \lambda_k I)(x_{k+1} - x_k) + \nabla f_{\nu(k)}(x_k) = 0 \\ \lambda_k \|x_{k+1} - x_k\| = 0 \\ \lambda_k \geq 0, \ \ \|x_{k+1} - x_k\| \leq \Delta_k. \end{gathered} \tag{37}$$

Therefore, by (35), $\|x_{k+1} - x_k\| \leq c\|\nabla f_{\nu(k)}(x_k)\|$ for $k \in K$ large enough, and, by (34), $\lim_{k \in K} \|x_{k+1} - x_k\| = 0$, as we wanted to prove. □

Theorem 3.3. *Assume that x_* is a limit point of a sequence $\{x_k\}$ generated by Algorithm 3.1. Suppose that Assumption A1 holds at x_*. Then, the whole sequence x_k converges quadratically to x_*.*

Proof. Let K be an infinite sequence of indices such that $\lim_{k \in K} x_k = x_*$. Let us prove first that

$$\lim_{k \to \infty} x_k = x_*.$$

By Lemma 3.2, there exists $\varepsilon > 0$ such that x_* is the unique limit point in the ball with center x_* and radius ε. Define

$$K_1 = \{k \in I\!N \mid \|x_k - x_*\| \leq \varepsilon/2\}.$$

The subsequence $\{x_k\}_{k \in K_1}$ converges to x_*, since x_* is its unique possible limit point. Therefore, by Lemma 3.3,

$$\lim_{k \in K_1} \|x_{k+1} - x_k\| = 0. \tag{38}$$

Let k_1 be such that, for all $k \in K_1, k \geq k_1$,

$$\|x_{k+1} - x_k\| \leq \varepsilon/2.$$

The set $\mathcal{B}(x_*, \varepsilon) - \mathcal{B}(x_*, \varepsilon/2)$ does not contain limit points of $\{x_k\}$. Therefore, there exists $k_2 \in I\!N$ such that, for all $k \geq k_2$,

$$\|x_k - x_*\| \leq \varepsilon/2 \text{ or } \|x_k - x_*\| > \varepsilon.$$

Let $k \in K_1$ such that $k \geq \max\{k_1, k_2\}$. Then,

$$\|x_{k+1} - x_*\| \leq \|x_k - x_*\| + \|x_{k+1} - x_k\| \leq \varepsilon/2 + \varepsilon/2 = \varepsilon.$$

Since x_{k+1} cannot belong to $\mathcal{B}(x_*, \varepsilon) - \mathcal{B}(x_*, \varepsilon/2)$, it turns out that $\|x_{k+1} - x_*\| \leq \varepsilon/2$. Therefore, $k+1 \in K_1$. So, we may prove by induction that $x_\ell \in K_1$ for all $\ell \geq k$. By (38), this implies that

$$\lim_{k\to\infty} x_k = x_*. \tag{39}$$

Let us prove now the quadratic convergence.

Let $I_\infty \subset \{1, \ldots, m\}$ be the set of indices i such that $i = \nu(k)$ infinitely many times. Then, there exists k_2 such that for all $k \geq k_2$, $\nu(k) \in I_\infty$.

Now, if $i \in I_\infty$ it turns out that $f_i(x_k) \leq f_{min}(x_k)$ infinitely many times. Therefore, by (39), and the continuity of f_i and f_{min}, we have that $i \in I_{min}(x_*)$. By Theorem 3.2, $\nabla f_i(x_*) = 0$. Therefore, by Assumption A1, $\nabla^2 f_i(x_*) \succ 0$. Thus, by the continuity of the Hessians, there exists $c_i > 0$ such that $\nabla^2 f_i(x)$ is positive definite and $\|\nabla^2 f_i(x)^{-1}\| \leq c_i$ for all x in a neighborhood of x_*. This implies that there exists $k_3 \geq k_2$ such that $\nabla^2 f_{\nu(k)}(x_k)$ is positive definite and $\|\nabla^2 f_{\nu(k)}(x)^{-1}\| \leq \beta \equiv \max\{c_i, i \in I_\infty\}$ for all $k \geq k_3$.

Therefore, for all $k \geq k_3$, we may define

$$\overline{x}_k = x_k - \nabla^2 f_{\nu(k)}(x_k)^{-1} \nabla f_{\nu(k)}(x_k). \tag{40}$$

Since $\nabla f_i(x_*) = 0$ for all $i \in I_\infty$, we have that $\lim_{k\to\infty} \nabla f_{\nu(k)}(x_k) = 0$. Then, by the boundedness of $\|\nabla^2 f_{\nu(k)}(x)^{-1}\|$, we have that $\|\overline{x}_k - x_k\| \to 0$, therefore, for k large enough, $\|\overline{x}_k - x_k\| \leq \Delta_{min}$. But $\overline{x}_k$ is, for k large enough, the unconstrained minimizer of $\psi^k_{\nu(k)}$. Therefore, since the first trust region radius at each iteration is greater than or equal to Δ_{min}, it turns out that, for k large enough, $\overline{x}_k$ is the first trial point at each iteration of Algorithm 3.1.

By (40) we have that

$$|\psi^k_{\nu(k)}(\overline{x}_k) - \psi^k_{\nu(k)}(x_k)| = \frac{1}{2}(\overline{x}_k - x_k)^T \nabla^2 f_{\nu(k)}(x_k)(\overline{x}_k - x_k).$$

Therefore, by Assumption A1, there exists $c > 0$ such that for k large enough,

$$|\psi^k_{\nu(k)}(\overline{x}_k) - \psi^k_{\nu(k)}(x_k)| \geq c\|\overline{x}_k - x_k\|^2. \tag{41}$$

Define

$$\rho_k = \frac{f_{\nu(k)}(\overline{x}_k) - f_{\nu(k)}(x_k)}{\psi^k_{\nu(k)}(\overline{x}_k) - \psi^k_i(x_k)}$$

By (41), and Taylor's formula, we have that

$$|\rho_k - 1| = \left| \frac{f_i(\overline{x}_k) - f_i(x_k) - [\psi^k_i(\overline{x}_k) - \psi^k_i(x_k)]}{\psi^k_i(\overline{x}_k) - \psi^k_i(x_k)} \right| \leq \frac{o(\|\overline{x}_k - x_k\|^2)}{c\|\overline{x}_k - x_k\|^2} \tag{42}$$

Since $\|\overline{x}_k - x_k\| \to 0$, we have that $\rho_k \to 1$. Therefore, for k large enough, the sufficient descent condition (14) is satisfied at the first trial point $\overline{x}_k$. Therefore, $x_{k+1} = \overline{x}_k$ for k large enough. This means that, for $k \geq k_3$ large enough,

$$x_{k+1} = x_k - \nabla^2 f_{\nu(k)}(x_k)^{-1} \nabla f_{\nu(k)}(x_k).$$

Then, by the elementary local convergence theory of Newton's method (see, for example [8], pp. 90-91) we have that, for k large enough:

$$\|x_{k+1} - x_*\| \leq \beta\gamma \|x_k - x_*\|^2,$$

where γ is a Lipschitz constant for all the Hessians $\nabla^2 f_i(x)$. □

4 Numerical results

Algorithm 3.1 was implemented with the following specifications:

1. We choose $\alpha = 0.1, \sigma_1 = 0.001, \sigma_2 = 5/9$.

2. The initial Δ at each iteration was chosen as the average norm of the Cα atoms of the protein $\mathcal{Q}$ at their original positions multiplied by 10. This is a "big Δ" decision, the consequence of which is that, in most iterations, the initial trial point comes from an unconstrained Newton step. Clearly, in Algorithm 3.1 we may consider that Δ_{min} is equal to the adopted big Δ.

3. When (13) does not hold, Δ_{new} is computed in the following way: We define

$$Ared = f_{min}(x_k) - f_{min}(\bar{x}),$$

$$Pred = \psi^k_{\nu(k)}(x_k) - \psi^k_{\nu(k)}(\bar{x}),$$

   and

$$\Delta_{\text{new}} = \max\left\{0.001, \frac{Pred}{2(Pred - Ared)}\right\} \times \|\bar{x} - x_k\|. \tag{43}$$

   The fact that $\Delta_{\text{new}} \leq \sigma_2 \|\bar{x} - x_k\|$ is guaranteed for $\sigma_2 = 5/9$ because $Ared > \alpha Pred$ in the case that Δ_{new} needs to be computed and $\alpha = 0.1$ in our implementation.

We arrived to the formula (43) after careful experimentation with other possibilities, including the classical ones associated to smooth trust region methods (see, for example [10], pp. 95-96).

In order to optimize the behavior of Algorithm 3.1, it was crucial to define a "big" trust region radius at the beginning of each iteration. The trust region radius that defines the first trial point was chosen to be independent of the last trust region radius employed at the previous iteration. This decision allowed the algorithm to use, very frequently, pure Newton steps and avoided artificial short steps far from the solution. Since the function f_i that defines f_{min} at a trial point may be different than the one that defines f_{min} at the current point, the quadratic model of f_{min} tends to underestimate the true value in many cases. This fact seems to stimulate the use of initial optimistic large steps.

We implemented Algorithm 3.1 under the framework of the standard trust region method `Betra` for box-constrained optimization [5] (see `www.ime.usp.br/∼egbirgin/tango`). Since our problem is unconstrained, we set artificial bounds $-10^{20}, 10^{20}$ for each variable. The code

`Betra` needed to be adapted to the algorithmic decisions described at the beginning of this section. In the unconstrained case, `Betra` is a standard trust region method that uses the Moré-Sorensen algorithm [30]. Many line-search and trust region methods for smooth unconstrained minimization may be used for solving (4) if one simply ignores the nonsmoothness of the objective function "defining" $\nabla f_{min}(x) = \nabla f_i(x)$ and $\nabla^2 f_{min}(x) = \nabla^2 f_i(x)$ for some $i \in I_{min}(x)$. Of course, the theoretical properties may change for each algorithmic choice. For example, we cannot expect that the thesis of Theorem 3.2 hold if one uses the algorithms TRON [26] or BOX-QUACAN [11] [26], since this theorem does not hold for those algorithms in the ordinary smooth case ($m = 1$).

Numerical experiments were run on an AMD Opteron 242 with 1Gb of RAM running Linux. The software was compiled with the GNU fortran compiler version 3.3 with the "-O3 -ffast-math" options.

We employed two different versions of the trust region method for solving Protein Alignment problems:

1. Dynamic Programming Newton Trust (DP-Trust) method: Admissible correspondences are monotone bijections between subsets of $\mathcal{Q}$ and $\mathcal{P}$. This is the trust region counterpart of the line-search method DP-LS (Dynamic Programming Line-Search) introduced in [2].

2. Non-Bijective Newton Trust (NB-Trust) method: Admissible correspondences are, merely, functions (not necessarily bijective) between subsets of $\mathcal{Q}$ and $\mathcal{P}$. This is the trust region counterpart of the line-search method NB-LS (Non-Bijective Line-Search) introduced in [2].

Given a rigid-body displacement D, the correspondence that maximizes the *Structal* Score (with gap penalization) is obtained using Dynamic Programming in DP-Trust, and using a cheap procedure for computing non-bijective correspondences (described in [2, 4]) in NB-Trust (without gap penalization). The second phase at each iteration of DP-Trust and NB-Trust is a Newtonian trust region iteration as described in Section 3. The computer time of both methods is dominated by the computation of the objective function. For DP-Trust, 98% of the time is spent on performing the dynamic programming steps and other 1.4% computing the gradient and the hessian, which are steps also performed by the line-search procedure. For NB-Trust, the DP steps required for obtaining the initial point and for computing the final *Structal* score takes 61% of the time, followed by the identification of the shortest distances (16%) and by the computation of the gradient and hessian of the objective function (15%). The cost of solving trust region subproblems represents less than 0.1% in both cases, thus being negligible. Therefore, it is not worthwhile to use approximate solutions of subproblems, instead of exact ones, as many well established trust region methods for large-scale optimization do (see [7, 11, 26] and [34](Chapter 4), among others).

We used the same initial approximations for the application of all the methods under consideration to Protein Alignment problems [2].

Method	Average time per alignment (s)	One-to-all in PDB	All-to-all in PDB
DP-LS	0.127	75 min	2.5 years
DP-Trust	0.141	82 min	2.8 years
NB-LS	0.033	19 min	7.7 months
NB-Trust	0.033	19 min	7.7 months
`Structal`	0.224	130 min	4.4 years

Table 1: Computer time required for an average alignment by each method.

4.1 Numerical comparison

4.2 Average computer times

We compared the performances of DP-Trust, DP-LS, NB-Trust, NB-LS and `Structal` by performing 79800 alignments containing both related and unrelated proteins. The set of 400 proteins for which this all-on-all comparison was performed was chosen from a DALI [14, 15] classification: 20 proteins were selected randomly within the DALI alignment database and the 20 best matches for each of these 20 proteins were also included in the set. Therefore, the 400 proteins include both different and similar structures, approximately grouped in sets of 20. The list of proteins used in the comparison is available at the lovoalign site. Each alignment was stopped when the difference between scores at two consecutive iterations was less than or equal to 10^{-6}. The average time per alignment required by each method is shown in Table 4.2.

Table 1 reports average computer times, disregarding the effective scores obtained by the different methods. The methods that use nonbijective correspondences (NB-LS and NB-Trust) are faster because computing the best nonbijective correspondence (given the displacement) is much easier than computing the best bijective and monotone correspondence using Dynamic Programming. In fact, the whole application of NB-methods involves two Dynamic Programming calls, one at the beginning, to compute the initial approximation, and the other at the end, to compute the final *Structal* Score. As shown before, more than 60% of the computer time used by these methods is spent in these two Dynamic Programming calculations.

Line Search and trust region methods turned out to be more efficient than `Structal` in terms of overall computer time. The introduction of the trust region strategy in place of the line-search procedure did not improve the speed of the LS methods. However, we will show in the following section that the slight speed decrease of DP-Trust with respect to DP-LS is compensated by the best quality of the scores obtained.

We observe that the alignment of one protein to the whole Protein Data Bank (presently with about 35 thousand structures) takes less than two hours. The alignment of all the proteins in the PDB would require several months, using a single processor. However, this job may be obviously done in parallel, since different alignments are entirely independent, so that the whole task may be completed in quite affordable computer time for practical purposes.

The slightly bigger computer time required by DP-Trust relative to DP-LS is not a serious limitation, since score improvement is more important in massive alignments.

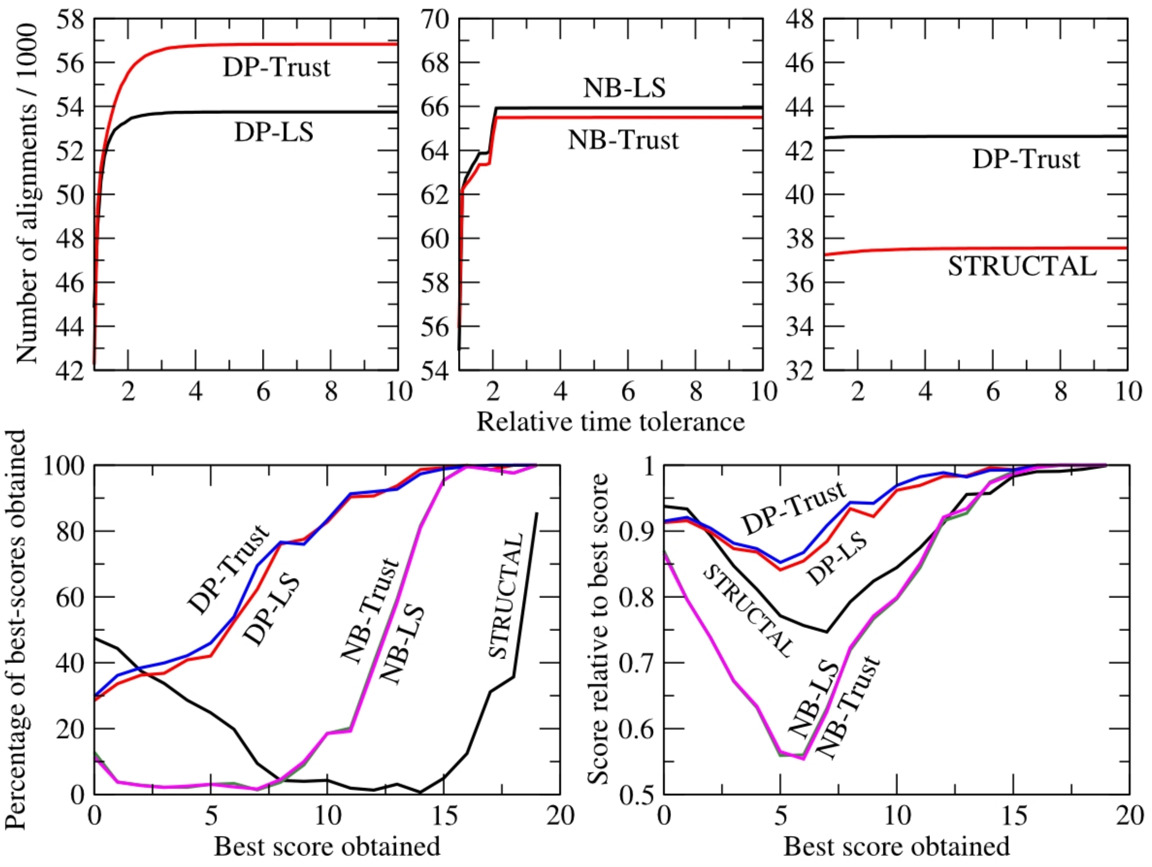

Figure 2: Performance profiles. (a) DP-Trust against DP-LS. (b) NB-Trust againts NB-LS. (c) DP-Trust against `Structal`. Analysis of the robustness of the methods for different alignment qualities are shown in (d) and (e).

4.3 Performance Profiles

Performance profiles [9] concerning the comparison of the 5 methods are presented in this section. We consider that a method "solves" a problem when it obtains the best score up to a tolerance 0.1%. If a method "does not solve" a problem, we consider, as usually, that the computer time used is ∞. Given the abscissa $x > 1$, the profile curve of a method takes the value y if there are y problems in which the computer time employed by this method is less than x times the computer time used by the best of the methods for the problem.

In Figure 2(a) the performance profile of DP-Trust against DP-LS is shown. DP-Trust obtains the best scores in 71% of the alignments, while DP-LS obtains the best scores in 67%. Therefore, the substitution of the line-search procedure by the trust region one improves the robustness of the alignment algorithm. Furthermore, for a relative time tolerance greater than 2, DP-Trust appears to be more efficient than DP-LS.

There is no meaningful differences when we compare NB-Trust and NB-LS strategies, as shown in Figure 2(b). In this set of alignments, NB-Trust obtains the best score (relatively *only* to these two methods) in 81% of the problems while NB-LS obtains best scores in 82% of the alignments. The differences in score values are not meaningful.

Finally, in order to contextualize the present results in terms of previously reported algorithms, we compute the performance profile of DP-Trust against `Structal`. DP-Trust is, as expected, both faster and more robust than `Structal`, as shown in Figure 2(c).

4.4 Robustness and score relevance

Good protein alignment methods should be able to obtain the best possible scores. However, if two proteins are completely different, obtaining the best score is not so relevant. In practice, one is interested in giving accurate distances and identifying similarities only when the proteins are similar.

With this in mind, we compared the scores obtained by each algorithm as a function of the best score obtained. In other words, we want to compare the scores of different algorithms as a function of the presumed protein similarity.

Figure 2(d) shows the percentage of cases in which each method was able to obtain the best score (up to a relative precision of 10^{-3}) as a function of the best score obtained for each problem. The best-*Structal* scores in the $x-$axis are normalized dividing by the number of atoms of the smallest protein being compared, so that normalized scores are between 0 and 20.

We observe that, for bad alignments, the `Structal` strategy is able to obtain the best scores in the greatest number of cases, followed by DP methods and NB methods. However, for best-scores greater than only 2.5, DP-LS and DP-Trust methods obtain the best alignments more frequently. For best-scores greater than 10 the best alignments are obtained by DP-Newton methods in more than 80% of the problems and, for best-scores greater than 13 this percentage goes to more than 98%. Non-bijective algorithms fail to obtain the best scores for medium to bad alignments. However, for best-scores greater than 14, these methods also obtain the best alignments, because the bijectivity of the best correspondence is automatically satisfied.

Figure 2(e) shows how close to the best score each method gets. The figure shows the average score obtained by each method relative to the best score obtained, as a function of the overall alignment quality. We can see, now, that the `Structal Method` obtains scores which are, on average, better than the ones obtained by NB methods. However, as shown in (Figure 2(d)), NB-methods obtain better scores than `Structal` for good alignments.

5 Final remarks

Protein Alignment is an challenging area for rigorous continuous Optimization. There is a lot of space for the development of algorithms with well established convergent theories that, presumably, converge to local optimizers, and many times to global ones. We feel that line-search and Trust-Region methods for (1) are rather satisfactory, but different alternatives should be mentioned. In [1], problems like (1) were reformulated as smooth nonlinear programming problems with complementarity constraints. This reformulation should be exploited in future works.

In this paper we showed that the trust region approach has some advantages over the line-search algorithm in terms of robustness, at least when one deals with DP-methods. We conjecture that the advantages of trust region methods over line-search methods may be more impressive in other structural alignment problems. In particular, preliminary results for alignments in which we allow internal rotations of the objects (see [4]) suggest that, in those cases, pure Newton directions are not so effective and restricted trust region steps could help. Further research is expected with respect to flexible alignments [25, 38, 43] in the near future.

In the experiments reported here, we always used the *Structal* score and we mentioned the fact that the `Structal Method` iteration maximizes this score at its first phase and minimizes the sum of squared distances (RMSD) for the selected bijection at its second phase. This second phase minimization admits an analytical solution [19]. Our approach here has been to maintain the first phase (and the *Structal* Score) changing the second phase to preserve coherence. The opposite choice is possible. We may preserve the Procrustes second phase, employing, at the first phase, a different score. An entirely compatible score with the Procrustes second phase may be defined by:

$$S(D,\Phi) = 20 \sum \max \left[0, 1 - \left(\frac{\|P_k - D(Q_{\Phi(k)})\|}{d_0} \right)^2 \right] - 10 \times gaps,$$

where $D, \Phi, \sum$ and *gaps* are as in (3) and d_0 is a threshold distance. If the distance between two Cα atoms (associated by Φ) exceeds d_0, its contribution to this score is zero. Maximizing this score is equivalent to minimize the RMSD for the atoms for which d_i is less than d_0. Three methods based on different d_0 values are also available in the LovoAlign software package.

We conjecture that the Low Order Value Optimization methodology may be employed in connection to alignment and protein classification in a number of different related problems: conservation of residues in columns of a multiple sequence alignment [27], percentage identity [32], SVM detection of distant structural relationships [31], protein-protein interfacial residual identification [24], prediction of subcellular localization [21], three-dimensional enzyme modeling [39], determination of score coefficients [20], hierarchical clustering [12] and many others.

Acknowledgements

The authors are indebted to two anonymous referees, whose comments were very useful to improve the presentation of the paper.

References

[1] R. Andreani, C. Dunder, and J. M. Martínez, Nonlinear-Programming Reformulation of the Order-Value Optimization problem, *Mathematical Methods of Operations Research* 61, pp. 365-384 (2005).

[2] R. Andreani, J. M. Martínez, and L. Martínez, Convergent algorithms for protein structural alignment, Technical Report, Department of Applied Mathematics, Unicamp, 2006. Currently available in the site `www.ime.unicamp.br/~martinez/lovoalign`.

[3] R. Andreani, J. M. Martínez, L. Martínez, and F. Yano, Low Order Value Optimization and Applications. Technical Report, Department of Applied Mathematics, Unicamp, 2006. Currently available at `www.ime.unicamp.br/∼martinez/lovoalign`.

[4] R. Andreani, J. M. Martínez, L. Martínez, and F. Yano, Continuous Optimization Methods for Structural Alignment, *To appear in Mathematical Programming*; 2007. Currently available at `www.ime.unicamp.br/∼martinez/lovoalign`.

[5] M. Andretta, E. G. Birgin and J. M. Martínez, Practical active set Euclidian trust-region method with spectral projected gradients for bound-constrained optimization, *Optimization* 54, pp. 305-325 (2005).

[6] H. M. Berman, *et al.*, The Protein Data Bank, *Nuclei Acids Research* 17, pp. 235-242 (2000).

[7] A. R. Conn, N. I. M. Gould, and Ph. L. Toint, *Trust-Region Methods*, MPS/SIAM Series on Optimization, SIAM-MPS, Philadelphia, 2000.

[8] J. E. Dennis and R. B. Schnabel, *Numerical Methods for Unconsconstrained Optimization and Nonlinear Equations*, Prentice-Hall, Englewood Cliffs, NJ, 1983. Reprinted by SIAM Publications, 1993.

[9] E. Dolan and J. J. Moré, Benchmarking optimization software with performance profiles, *Mathematical Programming* 91, pp. 201-213 (2002).

[10] R. Fletcher, *Practical Methods of Optimization*, John Wiley & Sons, New York, second ed., 1987.

[11] A. Friedlander, J. M. Martínez and S. A. Santos, A new trust region algorithm for bound constrained minimization, *Applied Mathematics and Optimization* 30, pp. 235-266 (1994).

[12] A. Gambin and P. R. Slonimski, Hierarchical clustering based upon contextual alignment of proteins: a different way to approach phylogeny, *Comptes Rendus Biologies* 328, pp. 11-22 (2005).

[13] M. Gerstein and M. Levitt, Comprehensive assessment of automatic structural alignment against a manual standard, the Scop classification of proteins, *Protein Sci.* 7, pp. 445-456 (1998).

[14] Holm, L., Sander C. Protein structure comparison by alignment of distance matrices. J. Mol. Biol. 1993;233:123-138.

[15] Holm, L, Sander, C. Mapping the Protein Universe. Science 1996;273:595–602.

[16] J. Hou, S.R. Jun, C. Zhang, and S. H. Kim, Global mapping of the protein structure space and application in structure-based inference of protein function, *P. Natl. Acad. Sci. USA* 2006; 103: Early Edition.

[17] J. Hou, G. E. Sims, C. Zhang, and S. H. Kim, A global representation of the protein fold space, *P. Natl. Acad. Sci. USA* 100, pp. 2386-2390 (2003).

[18] W. Kabsch, A discussion of the solution for the best rotation to relate two sets of vectors, *Acta Crystallog. A* 34, pp. 827-829 (1978).

[19] S. K. Kearsley, On the orthogonal transformation used for structural comparisons, *Acta Crystallog. A* 45, pp. 208-210 (1989).

[20] J. Kececioglu and E. Kim, Simple and fast inverse alignment, *Research in Computational Molecular Biology*, Proceedings Lecture Notes in Computer Science 3909, pp. 441-455 (2006).

[21] J.K. Kim, G. P. S. Raghava, B. Y. Bang, and S. J. Choi, Prediction of subcellular localization of proteins using pairwise sequence alignment and support vector machine, *Pattern Recognition Letters* 27, pp. 996-1001 (2006).

[22] R. Kolodny, P. Koehl, and M. Levitt, Comprehensive evaluation of protein structure alignment methods: scoring by geometric measures, *Journal of Molecular Biology* 346, pp. 1173-1188 (2005).

[23] R. Kolodny and N. Linial, Approximate protein structural alignment in polynomial time. *P. Natl. Acad. Sci. USA* 101, pp. 12201-12206 (2004).

[24] J-J Li, D-S Huang, B. Wang, and P. Chen, Identifying protein-protein interfacial residues in heterocomplexes using residue conservation scores, *International Journal of Biological Macromolecules* 38, pp. 241-247 (2006).

[25] Z-W Li, Y-Z Ye, and A. Godzik, Flexible structural neighborhood - a database of protein structural similarities and alignments, *Nucleic Acids Research* 34, D277-D280 (2006).

[26] C. Lin and J. J. Moré, Newton's method for large bound-constrained optimization problems, *SIAM Journal on Optimization* 9, pp. 1100-1127 (1999).

[27] X-S Liu, J. Li, W-L Guo, and W. Wang, A new method for quantifying residue conservation and its applications to the protein folding nucleus, *Biochemical and Biophysical Research Communications* 351, pp. 1031-1036 (2006).

[28] F. Lu, S. Keles, S. J. Wright, and G. Wahba, Framework for kernel regularization with application to protein clustering, *P. Natl. Acad. Sci. USA* 102, pp. 12332-12337 (2005).

[29] J. J. Moré, Recent developments in algorithms and software for trust region methods. In A. Bachem, M. Grötschel and B. Korte, eds., *Mathematical Programming: The State of the Art* pp. 258-287, Springer-Verlag, 1983.

[30] J. J. Moré and D. C. Sorensen, Computing a trust region step, *SIAM Journal on Scientific and Statistical Computing* 4, pp. 553-572 (1983).

[31] H. Ogul and E. U. Mumcuoglu, SVM-based detection of distant protein structural relationships using pairwise probabilistic suffix trees, *Computational Biology and Chemistry* 30, pp. 292-299 (2006).

[32] G. P. S. Raghava and G. J. Barton, Quantification of the variation in percentage identity for protein sequence alignments, *BMC Bioinformatics* 7, Art 415 (2006).

[33] B. Needleman and C. D. Wunsch, A general method applicable to the search for similarities in the amino acid sequence of two proteins, *Journal of Molecular Biology* 48, pp. 443-453 (1970).

[34] Nocedal J, Wright, SJ. *Numerical Optimization*, Springer 1999, New York.

[35] J. N. Onuchic and P. G. Wolynes, Theory of protein folding, Curr. Opin. Struct. Biol. 2004;14:70-75.

[36] M. J. D. Powell, A new algorithm for unconstrained optimization. In J. B. Rosen, O. L. Mangasarian and K. Ritter, eds., *Nonlinear Programming*, pp. 31-65, Academic Press, 1970.

[37] A. Sali, T. L. Blundell, Comparative protein modeling by satisfaction of spatial restraints. *J. Mol. Biol.* 234, 779-815, 1993.

[38] M. Shatsky, R. Nussinov, and H. J. Wolfson, Flexible protein alignment and hinge detection, *Proteins* 48 (2002) 242-256.

[39] N. Singh, G. Cheve, M. A. Avery, and C. R. McCurdy, Comparative protein modeling of 1-deoxy-D-xylulose-5-phospate reductoisomerase enzyme from Plasmodium falciparum: A potential target for antimalarial drug discovery, *Journal of Chemical Information and Modeling* 46, pp. 1360-1370 (2006)

[40] S. Subbiah, D. V. Laurents, and M. Levitt, Structural similarity of DNA-binding domains of bacteriophage repressors and the globin core, *Curr. Biol.* 3, pp. 141-148 (1993).

[41] M. Vendruscolo and C. M. Dobson, A glimpse at the organization of the protein universe, *P. Natl. Acad. Sci. USA* 102, pp. 5641-5642 (2005).

[42] D. Voet and J. Voet, *Biochemistry*, John Wiley & Sons, 3d ed. 2004.

[43] Y-Z Ye and A. Godzik, Flexible structure alignment by chaining aligned fragment pairs allowing twists, *Bioinformatics* 19 (Suppl. 2) (2003) ii246-ii255.