CompAMa
www.compama.co.usb.ve

Bull. Comput. Appl. Math., Vol.1, No.1, 2013
ISSN 2244-8659

Global improvements of a protein alignment algorithm and comparison with a global optimization solver

Paulo Gouveia [*], *Roberto Andreani* [†], *Ana Friedlander* [‡], *J. M. Martínez* [§], *Leandro Martínez* [¶]

CompAMa Vol.1, No.1, pp.11-24, 2013 - Accepted November 9, 2012

Abstract

The LovoAlign method for Protein Alignment, based on the Low-Order Value Optimization theory, is recalled. The method is modified in order to improve global convergence properties and compared against other global minimization procedures.

Keywords: Protein Alignment, Global Optimization, Structal score.

1 Introduction

Protein Alignment is the process of discovering similarities between the 3D structures of two proteins, which are generally represented by the sequences

[*]UNORP (Centro Universitário do Norte Paulista), Rio Preto, SP, Brazil, e-mail: paulosg@gmail.com

[†]Department of Applied Mathematics, Institute of Mathematics, Statistics and Scientific Computing, University of Campinas, Campinas SP, Brazil, e-mail: andreani@ime.unicamp.br

[‡]e-mail: friedlan@ime.unicamp.br

[§]e-mail: martinez@ime.unicamp.br

[¶]Department of Physics, University of São Paulo at São Carlos, São Carlos SP and Department of Physical Chemistry, State University of Campinas, Brazil, e-mail: leandro@iqm.unicamp.br

of their atoms. A generic protein $\mathcal{A}$ will be identified, in this paper, with a spatial sequence $A \equiv (A_1, \ldots, A_{na})$ where, for all $i = 1, \ldots, n_a$, $A_i \in I\!R^3$ provides the coordinates of the protein's $i-th$ Cα- atom.

If $T : I\!R^3 \to I\!R^3$ is an arbitrary rigid displacement (i. e. $T(v) = Uv + c$, where U is orthogonal and $\det(U)$=1), the protein $\mathcal{A}$ may be (equivalently) represented by $T(A) \equiv (T(A_1), \ldots, T(A_{na})$. Assuming that $B \equiv (B_1, \ldots, B_{nb})$ provides the spatial structure of a second protein $\mathcal{B}$, the objective of Protein Alignment is to find a rigid displacement T such that $T(A)$ is as close to B as possible. For this purpose, many similarity measures (generally called "scores") have been defined. See, among others, [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11].

Assume that $(a_1, \ldots, a_N)$ is a subsequence of $(T(A_1), \ldots, T(A_{na}))$ and $(b_1, \ldots, b_N)$ is a subsequence of $(B_1, \ldots, B_{nb})$. We consider the sequence of pairs $((a_1, b_1), \ldots, (a_N, b_N))$. If, for some $i = 1, \ldots, N-1$ one has that $(a_i, b_i) = (T(A_r), B_s)$, $(a_{i+1}, b_{i+1}) = (T(A_t, B_u))$, with $t > r+1$ and $u > s+1$, we say that the sequence $((a_1, b_1), \ldots, (a_N, b_N))$ has a "gap" between (a_i, b_i) and (a_{i+1}, b_{i+1}). The Structal Score associated with $((a_1, b_1), \ldots, (a_N, b_N))$ is defined by

$$Structal(a, b) = \sum_{i=1}^{N} \frac{20}{1 + \|a_i - b_i\|_2^2/5} - 10N_{gaps}, \tag{1}$$

where N_{gaps} is the number of gaps [11]. Clearly, the sequence of pairs $(a_1, b_1), \ldots, (a_N, b_N)$ can be viewed as a monotone bijection between subsequences of $T(A)$ and B. Let us denote by $\mathcal{S}(\mathcal{T})$ the set of all monotone bijections (pairs of subsequences) between $T(A)$ and B. Maximizing the expression (1) over $\mathcal{S}(T)$ we obtain the Structal score associated with the rigid transformation T. For simplicity, we denote:

$$Structal(T) = \max_{(a,b)\in\mathcal{S}(T)} \sum_{i=1}^{N} \frac{20}{1 + \|a_i - b_i\|_2^2/5} - 10N_{gaps}. \tag{2}$$

The bijection that maximizes (1) may be obtained by means of a Dynamic Programming procedure that employs $O(\min\{na, nb\}^2)$ computer time units [12].

Finally, maximizing $Structal(T)$ with respect to all possible rigid displacements T we obtain the Structal Score associated with the proteins $\mathcal{A}$

and $\mathcal{B}$, which will be denoted here by $S(\mathcal{A}, \mathcal{B})$. Namely,

$$S(\mathcal{A}, \mathcal{B}) = \max_{T} \left[\max_{(a,b)\in\mathcal{S}(T)} \sum_{i=1}^{N} \frac{20}{1 + \|a_i - b_i\|_2^2/5} - 10N_{gaps} \right]. \tag{3}$$

In order to maximize the function $Structal(T)$, one needs a suitable parameterization of the space of rigid transformations. In the LovoAlign package[1] [13, 14, 15], rigid displacements are described by 6 parameters $x \equiv (x_1, \ldots, x_6)$. The first three are rotation (Euler) angles whereas x_4, x_5, x_6 represent the spatial translation. So, we may write, for all $V \in I\!R^3$:

$$T(x)(V) = U(x_1, x_2, x_3) \times V + \begin{pmatrix} x_4 \\ x_5 \\ x_6 \end{pmatrix}, \tag{4}$$

where

$$U(x_1, x_2, x_3) = \begin{pmatrix} \cos(x_1) & 0 & \sin(x_1) \\ 0 & 1 & 0 \\ -\sin(x_1) & 0 & \cos(x_1) \end{pmatrix}$$

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(x_2) & -\sin(x_2) \\ 0 & \sin(x_2) & \cos(x_2) \end{pmatrix} \begin{pmatrix} \cos(x_3) & -\sin(x_3) & 0 \\ \sin(x_3) & \cos(x_3) & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

With this parameterization, $Structal : I\!R^6 \to I\!R$ becomes a function of 6 variables, whose evaluation involves finding the maximum of (1) over the set of admissible bijections between subsequences of $T(A)$ and B. The *Structal* function is continuous but non-differentiable. LovoAlign is a freely available computer package that implements a Low Order-Value (LOVO) optimization method [13, 14, 16] for maximizing this function. The LOVO theory guarantees that LovoAlign finds stationary points (very likely, local maximizers) of $Structal(x)$. Practical application of LovoAlign is reported in [17, 18, 19, 20].

In this paper we introduce variations of LovoAlign with enhanced chances of convergence to global maximizers of the *Structal* function, and we compare these variations with a well-established global optimization method.

[1] The LovoAlign package is available in www.ime.unicamp.br/ ∼martinez/lovoalign

2 Reduction of the space of translations

In LovoAlign the translations are represented by a continuum of parameters x_4, x_5, x_6, corresponding to all possible displacements in the three dimensions of the space. Other algorithms consider only those displacements that exactly superimpose atoms $T(A_i)$ with atoms B_j. Given a rotation matrix U, for all $i = 1, \ldots, na$ and $j = 1, \ldots, nb$, we define the translation $d^{ij}(U) \in I\!R^3$ by:

$$UA_i + d^{ij}(U) = B_j.$$

Therefore,

$$d^{ij}(U) = B_j - UA_i. \tag{5}$$

A restricted rigid transformation will be defined by

$$T_R(V) = UV + d^{ij}(U),$$

where U is a 3D rotation and $d^{ij}(U)$ is defined by (5) for all $i = 1, \ldots, na$ and $j = 1, \ldots, nb$.

Consequently, we define the restricted Structal score associated with the proteins $\mathcal{A}$ and $\mathcal{B}$ by

$$S_R(\mathcal{A}, \mathcal{B}) = \max_{T_R} \left[\max_{(a,b)\in\mathcal{S}(T_R)} \sum_{i=1}^{N} \frac{20}{1 + \|a_i - b_i\|_2^2/5} - 10N_{gaps} \right], \tag{6}$$

where the maximum in (6) is taken over the set of restricted rigid transformations T_R.

The most conspicuous method that employs restricted rigid transformations was given by Kolodny and Linial [9] with the aim of proving that approximate solutions of the Protein Alignment problem can be obtained with polynomial complexity. Their method employs a three-dimensional grid in the space of angles and shows that, if the grid is fine enough, the computer time necessary to achieve a given precision depends polynomially on the number of atoms. Unfortunately, their method is not computationally affordable with the currently available computers.

3 Methods

In this section we describe the methods used in our comparative study. In one extreme we have LovoAlign, which, as we mentioned above, is a "local"

method in the sense that convergence is guaranteed by rigorous mathematical theory towards points that satisfy first-order optimality conditions [13, 14, 15]. These limit points are not necessarily global optimizers of the problem

$$\text{Maximize } \; \mathit{Structal}(T), \tag{7}$$

whose global solution is $S(\mathcal{A}, \mathcal{B})$, as defined in (3). In the other extreme we will consider the method MCS (Multilevel Coordinate Search) [21], which is a well established publicly available method for optimize continuous functions. Several additional methods, consisting in "global variations" of LovoAlign, will be tested[2].

3.1 LovoAlign

LovoAlign is an iterative method for solving (7), in which rotations are represented by the three Euler angles and translations are continuous variables. Therefore, with some abuse of notation, problem (7) can be written as follows:

$$\text{Maximize } \mathit{Structal}(x), \; x \in I\!R^6. \tag{8}$$

Given an iterate $x^k \in I\!R^6$ and the corresponding rigid movement T_k, defined by (4), we compute the displaced protein $A(T_k)$ and, using Dynamic Programming, we compute the best bijection between subsequences of $A(T_k)$ and B. Assume that the subsequence of paired atoms found by Dynamic Programming is $((a_1^k(x^k), b_1^k), \ldots, (a_{N_k}^k(x^k), b_{N_k}^k))$. Then, by (1), we have that

$$\mathit{Structal}(x^k) = \sum_{i=1}^{N_k} \frac{20}{1 + \|a_i^k(x^k) - b_i^k\|_2^2/5} - 10N_{gaps,k}. \tag{9}$$

Since $a_i^k(x^k)$ depends explicitly on x^k we may compute, formally, the derivatives of $\mathit{Structal}(x)$ at the iterate x^k. Rigorously speaking, these derivatives may not exist, because the same value $\mathit{Structal}(x^k)$ could be obtained for different sequences of paired atoms. In spite of this lack of derivative unicity, we denote by $\nabla \mathit{Structal}(x^k) \in I\!R^6$ and $\nabla^2 \mathit{Structal}(x^k) \in I\!R^{6\times 6}$ the gradient and the Hessian so far computed, respectively. If the Hessian is not negative-definite we replace it by a negative definite matrix that comes from adding a suitable negative term to the diagonal. (An alternative trust-region strategy is described in [22].) The (perhaps modified)

[2] The MCS method is available in www.mat.univie.ac.at/~neum/software/mcs

Hessian matrix will be denoted by H_k. It turns out that the Newtonian direction $-H_k^{-1}\nabla Structal(x^k)$ is an ascent direction for the function $Structal(x)$ [14, 15] and, so, a sufficiently bigger functional value can be obtained along this direction. Therefore, we may obtain a better iterate x^{k+1} by means of a simple one-dimensional search.

A rigorous theoretical analysis allows one to prove that, given an arbitrarily small $\varepsilon > 0$, there exists k_0 such that, for all $k \geq k_0$, all the iterates satisfy

$$\|\nabla Structal(x^k)\| \leq \varepsilon.$$

Roughly speaking, this indicates that the method converges to local maximizers of the objective function.

The theoretical convergence behavior of LovoAlign independs of the initial iterate x^0. Nevertheless, in order to enhance the probability of convergence to global maximizers, the LovoAlign package always employs a particular initial point which is computed by means of a single alignment on the distance structures of the two involved proteins. In our experiments we will employ the default algorithmic parameters [15] of the standard implementation in `www.ime.unicamp.br/~martinez/lovoalign`.

3.2 Multilevel Coordinate Search (MCS)

MCS is a global optimization method for continuous functions defined on an n-dimensional box, developed by Huyer and Neumaier [21]. The method is based in successive partitioning of a set of boxes into smaller ones and local searches based on interpolations.

A convergence theorem guarantees that, given an arbitrarily small $\varepsilon > 0$, the method eventually finds a point whose functional value differs from the optimal one in less than ε. The proof of this theorem is based on the fact that, ultimately, the algorithm generates a dense set of points in the original box [21].

MCS can be applied to the Protein Alignment problem both in its original form (3) or employing the restricted set of translations. In the first case the problem has 6 variables while, in the second one, the number of variables is 3. However, in the 3-variables case the objective function is much more expensive. The reason is that, in the 6 variable-case we evaluate the score for a given rotation and translation, both of them being variables. In the 3-variable case, only rotations are variables but we need to minimize over the

finite set of selected translations. The consequences of this fact are impressive. For example, aligning the proteins 1MJC and 1SHF we obtained the score 465.1 in 3 minutes of computer time using the 6-variable formulation, whereas the obtained score was 463.3 and 110 minutes were necessary for the MCS computations in the restricted-translation formulation. The same behavior was observed in many other cases. Consequently, we always used the formulation with 6 variables.

For running MCS we imposed the bounds $x_i \in [0, 2\pi]$ for $i = 1, 2, 3$ and we used a box that guaranteedly contains the global maximizer of the problem. Without loss of generality, we assumed that the origin of $\mathbb{R}^3$ is the barycener of of both proteins A and B.

In our experiments we employed the standard implementation of MCS, available in the NAG library (`www.nag.co.uk`). We used the following algorithmic parameters:

Maximum number of division of each box: 60.

Maximum number of function evaluations: 10^6.

Stopping criterion based on lack of objective function improvement *stclim*: 5000.

Maximum number of local searches: 50.

3.3 Multistart LovoAlign (ML)

The Multistart LovoAlign method (ML) consists of running LovoAlign using a (large) set of initial points, which are defined as follows: We define a grid in the space of angles (x_1, x_2, x_3), using n values for each x_i, $i = 1, 2, 3$. For each rotation matrix U we employ m initial points defined by different translations. Given a rotation, the $j - th$ translation will be defined by the condition

$$T(C) = B_{\nu[(j-1)/(nb-1)/(m-1)+1]}, \tag{10}$$

where C is the barycenter of A and $\nu(z)$ denotes the closest integer to z and $j = 1, 2, \ldots, m$.

In our experiments we used $m = n = 100$.

3.4 Weighted Multistart LovoAlign (WML)

This method is a modification of ML in which we try to reduce the number of rotations used for the generation of initial points.

Recall that the set of rotations may be represented by a box (cube) in the space of angles. We divide this initial box into 8 smaller cubes C_i. We select a random point (rotation) within each of these cubes and we generate q associated initial points for LovoAlign according to the rule (10). We run LovoAlign with these initial points and we associate each cube with the best score s_i so far obtained. Without loss of generality, assume that

$$s_1 \leq \ldots \leq s_8.$$

Then, we generate p new random points in C_1, $2p$ new random points in C_2 and so on, and we repeat the LovoAlign process. Again, we associate each cube with its best score. This time, the cube with smallest score is discarded, new random points are generated in the remaining cubes (according to their internal scores) and the process is repeated. Therefore, at each stage at least one cube is eliminated and, at the end, the number of admissible cubes is empty and the algorithm stops.

In our experiments we used $p = 10, q = 50$.

3.5 Bayesian Multistart LovoAlign (BML)

In the ML method we employ a given number of rotations corresponding to a grid of equally spaced angles and m different translations per rotation to define initial points for LovoAlign executions. This approach does not take into account several factors that may contribute to the efficiency of the method. The comparison between the number of different local maximizers already obtained and the number of initial points already tested is an important issue that could be used to determine whether it is worthwhile to continue LovoAlign executions or not. Different authors [23, 24, 25] considered this issue and defined methods that take stopping decisions supported by Bayesian statistics. They suggest to minimize the "expecteda posterior loss" taking into account the "termination loss", which is defined as the loss of stopping before finding all the local maximizers and the "execution loss", which corresponds to the cost of running the method for a new initial point. The BML method is the ML method described before, with Bayesian stopping rules that are fully described in [26].

3.6 Normalized Bayesian Multistart LovoAlign (NBML)

By (1), (2) and (3), we have that for every pair of proteins $\mathcal{A}, \mathcal{B}$, one has:

$$S(\mathcal{A}, \mathcal{B}) \leq 20 \ \min\{na, nb\}.$$

This inequality motivates one to define the Normalized Structal Score S_N by

$$S_N(\mathcal{A}, \mathcal{B}) = \frac{1}{\min\{na, nb\}} \times S(\mathcal{A}, \mathcal{B}).$$

The NBML method coincides with BML, with the exception that the normalized score S_N is used instead of the usual Structal score in order to decide that two local maximizers are the same or not. Therefore, the application of NBML involves the employment of an additional parameter $\delta > 0$: If the modulus of the different between normalized scores obtained at two different runs of LovoAlign is smaller than δ we declare that the corresponding local maximizers are equivalent.

In our experiments we used $\delta = 0.1$.

3.7 Normalized Bayesian Multistart LovoAlign with Initialization (NBMLI)

In ML, BML and NBML we discard the initial point provided by LovoAlign. This initial point is computed by means of the alignment of two pseudo-proteins A' and B' whose "pseudo-atoms" are the differences between consecutive atoms of A and B. The alignment of the pseudo-proteins is computed by means of a single DP process.

The NBMLI method coincides with NBML with the exception that the alignment based on pseudo-proteins is used to define the first initial point for ordinary LovoAlign execution. In other words, NBMLI uses a standard LovoAlign execution in addition to the LovoAlign runs of NBML.

4 Experiments

We chose, randomly, 5 proteins from the Protein Data Bank [27]: 1BK2A, 1AG7A, 1CBNA, 2VY5A, and 3E7RA. For each of these proteins, we select the 10 more similar proteins according to DALI [4, 5, 6]

Method	Time	Solved Problems	Critical Score
LovoAlign	2 seconds	330	12.4
MCS	7 hours and 35 minutes	1140	8.3
ML	7 hours and 51 minutes	1221	6.6
WML	4 hours and 37 minutes	1114	6.6 (*)
BML	1 hour and 9 minutes	1138	8.4
NBML	22 minutes and 39 seconds	972	8.7
NBMLI	18 minutes and 40 seconds	976	8.7

Tab. 1: Comparison of Alignment methods. (*): WML did not solve a problem in which BNS(P) was greater than 6.6.

(`http://ekidna.biocenter.helsinki.fi/dali-server`). Therefore, we have a set of 50 proteins, which define 1225 alignment problems. This is the set of problems used in the present study. All the tests were run using a computer with Core 2 Quad Q9550 2.83 GHz with 8 Gb, employing Linux Ubuntu 9.04 and the gfortran compiler.

For each alignment problem P, we define BNS(P) as the best normalized score obtained by any of the methods considered in this report. On the other hand, the Structal score obtained by a method M for solving a problem P will be denoted by St(M, P). We say that a method M *solves* a problem P if St(M, P) differs form BNS(P) in less than a given tolerance $\varepsilon > 0$.

It was observed that, for each method M, there exists a Critical Score CS(M) such that M solves all the problems for which BNS(P) is bigger than CS(M). (There exists one exception to this observation, denote by (*) in Table 1.) In Table 1 we display, for each method, the total CPU time employed for handling all the problems, the percent of solved problems with $\varepsilon = 0.5$, and the critical score.

Optimization methods may be compared in terms of robustness and efficiency. Robustness corresponds to the ability of finding solutions and Efficiency is usually associated with execution time. Strictly speaking, a method may be considered better than a competitor if solves more problems and uses less computer time simultaneously. A single ideal graphic could be drawn, where each method could be represented by a point in the xy plane. The abscissa corresponds to the number of problems in which the method failed to find the global solution and the ordinate represents the computer time

employed. An imaginary curve connecting the points that correspond to methods that are not overcome in both criteria may be viewed as the Pareto (indiference) curve for this situation. From this bi-objective point of view, we see that only WML and NBML can be discarded for being overcome by some competitor. BML is better than WML and NBLI is bettern than NBML. The remaining 5 methods are in the Pareto indiference curve.

5 Conclusions

LovoAlign is a method for Protein Alignment that employs the Structal Score and, very likely, finds the true global score if the proteins are similar enough. In addition, the Newtonian features of the method make it very fast. These features are shared by other Protein Alignment methods using different scores.

As a consequence, it is natural to develop global-optimization variations of LovoAlign. It is interesting to try well established global optimization methods with this purpose, as well as global optimization ideas that exploit the structure of the problem. In the present contribution we used the global software MCS and five global variations of LovoAlign.

In the numerical experiments we observed that all the methods find accurate results in the case that the proteins are similar, the degree of similarity required being different for different methods. A single bi-objective representation could be used to visualize robustness and time-efficiency simultaneously showing that only two of the algorithms tested can be discarded from this bi-objective point of view.

Acknowledgement

We are indebted to an anonymous referee for his/her insightful reading and helpful comments.

This work was supported by CEPID on Industrial Mathematics (FAPESP), PRONEX-CNPq/FAPERJ Grant E-26/171.164/2003 - APQ1, FAPESP Grants 05/02163-8 and 06/53768-0, and CNPq.

AMS Subject Classification: 90C30, 49K99, 65K05.

References

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

[2] Gerstein M. and Levitt M. Comprehensive assessment of automatic structural alignment against a manual standard, the scop classification of proteins. *Protein Science*, 7:445–456, 1998.

[3] Hasegawa H. and Holm L. Advances and pitfalls of protein structural alignment. *Current Opinion in Structural Biology*, 19:341–348, 2009.

[4] Holm L. and Park J. Dalilite workbench for protein structure comparison. *Bioinformatics*, 16:566–567, 2000.

[5] Holm L. and Sander C. Protein structure comparison by alignment of distance matrices. *Journal of Molecular Biology*, 233:123–138, 1993.

[6] Holm L. and Sander C. Mapping the protein universe. *Science*, 273:595–602, 1996.

[7] Hou J., Jun S.R., Zhang C., and Kim S.H. Global mapping of the protein structure space and application in structure-based inference of protein function. *Proceedings of the National Academy of Sciences USA*, 102:3651–3656, 2005.

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

[9] Kolodny R. and Linial N. Approximate protein structural alignment in polynomial time. *Proceedings of the National Academy of Sciences USA*, 101:12201–12206, 2004.

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

[11] Subbiah S., Laurents D.V., and Levitt M. Structural similarity of dna-binding domains of bacteriophage repressors and the globin core. *Current Biology*, 3:141–148, 1993.

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

[13] Andreani R., Martínez J.M., Martínez L., and Yano F. Continuous optimization methods for structural alignments. *Mathematical Programming*, 112:93–124, 2008.

[14] Andreani R., Martínez J.M., Martínez L., and Yano F. Low order value optimization and applications. *Journal of Global Optimization*, 43:1–10, 2009.

[15] Martínez L., Andreani R., and Martínez J.M. Convergent algorithms for protein structural alignment. *BMC Bioinformatics*, 8:306, 2007.

[16] Martínez J.M. Generalized order-value optimization. *TOP*, 20:75–98, 2012.

[17] Forli S. and J. Olson A. A force field with discrete displaceable waters and desolvation entropy for hydrated ligand docking. *Journal of Medicinal Chemistry*, 55:623–638, 2012.

[18] Jordan D.M., Kiezun A., Baxter S.M., Agarwala V., Murray M.F. Green R.C., Pugh T., Lebo M.S., Rehm H.L., Funke B.H., and Sunyaev S.R. Development and validation of a computational method for assessment of missense variants in hypertrophic cardiomyopathy. *Americal Journal of Human Genetics*, 88:183–192, 2011.

[19] Liberato M.V., Nascimento A.S., Ayers S.D., Lin J. Z., Cvoro A., Silveira R.L., Martínez L., Souza P.C.T., Saidemberg D., Deng T., Amato A.A., Togashi M., Hsueh W.A., Phillips K., Palma M.S., Neves F.A.R., Skaf M.S., Webb P., and Polikarpov I. Medium chain fatty acids are selective peroxisome proliferator activated receptor (ppar) gamma activators and pan-ppar partial agonists. *PLOS ONE*, 7(5):241–247, 2012. Article Number: e36297 DOI: 10.1371/journal.pone.0036297.

[20] Puhl A.C., Bernardes A., Silveira R.L., Yuan J., Campos J.L.O., Saidemberg D.M., Palma M.S., Cvoro A., Ayers S.D., Webb P., Reinach P.S., Skaf M.S., and Polikarpov I. Mode of peroxisome proliferator-activated receptor gamma activation by luteolin. *Molecular Pharmacology*, 81:788–799, 2012.

[21] Huyer W. and Neumaier A. Global optimization by multilevel coordinate search. *Journal of Global Optimization*, 14:331–355, 1999.

[22] Andreani R., Martínez J.M., and Martínez L. Trust-region superposition methods for protein alignment. *IMA Journal on Numerical Analysis*, 28:690–710, 2008.

[23] Boender C. and Rinooy Kan A. A bayesian stopping rule for multistart global optimization methods. *Mathematical Programming*, 37:59–80, 1987.

[24] Hart W. Sequential stopping rules for random optimization methods with applications to multistart local search. *SIAM Journal on Optimization*, 9:270–290, 1998.

[25] Orsenigo C. and Vercellis C. Bayesian stopping rules for greedy randomized procedure. *Journal of Global Optimization*, 36:365–377, 2006.

[26] Gouveia P. Métodos de otimização global para alinhamento de proteínas. *Tese de Doutorado, Instituto de Matemática, Estatística e Computação Científica, Universidade Estadual de Campinas, Brazil*, 2011.

[27] Berman H. M. The protein data bank. *Nuclei Acids Research*, 17:235–242, 2000.