ISTITUTO DI ANALISI DEI SISTEMI ED INFORMATICA
"Antonio Ruberti"
CONSIGLIO NAZIONALE DELLE RICERCHE

F. Lampariello, G. Liuzzi

GLOBAL OPTIMIZATION OF PROTEIN-PEPTIDE DOCKING BY A FILLING FUNCTION METHOD

R. 25, 2012

Francesco Lampariello – Istituto di Analisi dei Sistemi ed Informatica "A. Ruberti", Consiglio Nazionale delle Ricerche (IASI-CNR), Viale manzoni 30, 00185 Rome, Italy.

Giampaolo Liuzzi – Istituto di Analisi dei Sistemi ed Informatica "A. Ruberti", Consiglio Nazionale delle Ricerche (IASI-CNR), Viale manzoni 30, 00185 Rome, Italy.
giampaolo.liuzzi@iasi.cnr.it.

This work has been partially supported by the FLAGSHIP "InterOmics" project (PB.P05) funded by the Italian MIUR and CNR organizations

ISSN: 1128–3378

Collana dei Rapporti dell'Istituto di Analisi dei Sistemi ed Informatica "Antonio Ruberti", CNR

viale Manzoni 30, 00185 ROMA, Italy

tel. ++39-06-77161
fax ++39-06-7716461
email: iasi@iasi.cnr.it
URL: http://www.iasi.cnr.it

Abstract

Molecular docking programs play a crucial role in drug design and development. In recent years, much attention has been devoted to the protein-peptide docking problem in which docking of a flexible peptide with a given protein is sought. In this work we present a docking algorithm which is based on the use of a filling function method for continuous global optimization. In particular, the protein-peptide docking position is found by minimizing the conformational potential energy function based on an appropriate mathematical model. The resulting global optimization problem presents some difficulties, since it is a large-scale one and the objective function is non-convex, so that it has many local minima. The method consists in modifying the potential function by adding to it a term in order to fill the basin of a local minimum. This allows to escape from that local solution and moreover, by properly exploiting the filling strategy, to explore large regions in the peptide positional and conformational space. Moreover, in order to obtain more accurate results, we search the solution by performing a two-phase optimization process. In particular, in a first step only the carbon C atoms of the protein and peptide are considered thus obtaining an approximate docking solution. Then, the energy function is completed by considering all the peptide and protein atoms so that starting from the solution of the first phase, the new minimization process gives a more accurate result. We present numerical results on a set of benchmark docking pairs and their comparison with those obtained by the well-known software package PacthDock for molecular docking.

Key words: Protein-peptide docking; potential energy model; continuous global optimization.

Global optimization of protein-peptide docking by a filling function method *

F. Lampariello†, G.Liuzzi†

IASI R.25, 2012

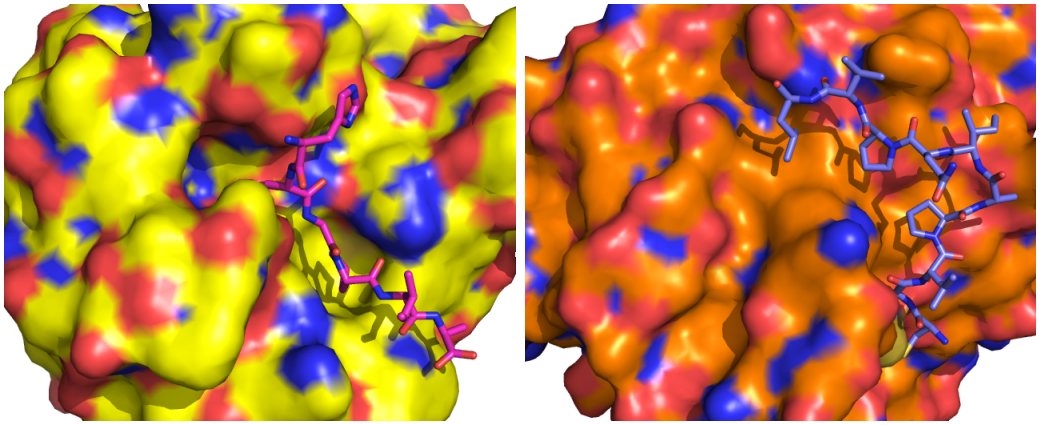
Abstract

Molecular docking programs play a crucial role in drug design and development. In recent years, much attention has been devoted to the protein-peptide docking problem in which docking of a flexible peptide with a given protein is sought. In this work we present a docking algorithm which is based on the use of a filling function method for continuous global optimization. In particular, the protein-peptide docking position is found by minimizing the conformational potential energy function based on an appropriate mathematical model. The resulting global optimization problem presents some difficulties, since it is a large-scale one and the objective function is non-convex, so that it has many local minima. The method consists in modifying the potential function by adding to it a term in order to fill the basin of a local minimum. This allows to escape from that local solution and moreover, by properly exploiting the filling strategy, to explore large regions in the peptide positional and conformational space. Moreover, in order to obtain more accurate results, we search the solution by performing a two-phase optimization process. In particular, in a first step only the carbon C_α atoms of the protein and peptide are considered thus obtaining an approximate docking solution. Then, the energy function is completed by considering all the peptide and protein atoms so that starting from the solution of the first phase, the new minimization process gives a more accurate result. We present numerical results on a set of benchmark docking pairs and their comparison with those obtained by the well-known software package PacthDock for molecular docking.

Keywords: Protein-peptide docking; potential energy model; continuous global optimization.

*This work has been partially supported by the FLAGSHIP "InterOmics" project (PB.P05) funded by the Italian MIUR and CNR organizations

†Istituto di Analisi dei Sistemi ed Informatica "A. Ruberti", Consiglio nazionale delle Ricerche. Viale Manzoni 30, 00185 Rome, Italy. Emails: francesco.lampariello@iasi.cnr.it, giampaolo.liuzzi@iasi.cnr.it



(a) (b)

Figure 1: Docking position of: (a) peptide HIS-ALA-GLY-PRO-ILE-ALA, also known as CHEMBL356871, with protein 1awq; (b) dipeptide-chymotrypsin with protein 1ab9

1 Introduction

In this paper we address the problem of docking a peptide molecule $\mathcal{Q}$ with small sequence of residues, e.g. a drug candidate, onto the structure of a given "reference" protein $\mathcal{P}$ (with $\mathcal{P}$ and $\mathcal{Q}$ we also refer to the set of their atoms, respectively).
In particular, we search the docking pocket and position of peptide $\mathcal{Q}$ onto protein $\mathcal{P}$. In Fig. 1 are shown two examples (the images are obtained by applying the software PyMol on data taken from [5]).

In this paper we assume that:

(i) the tertiary structure of $\mathcal{P}$ (i.e. all protein atoms coordinates in the space) is known and $\mathcal{P}$ is a rigid body;

(ii) the primary structure of $\mathcal{Q}$ (i.e. its residue sequence) is known and unchangeable and $\mathcal{Q}$ is fully flexible in its tertiary structure.

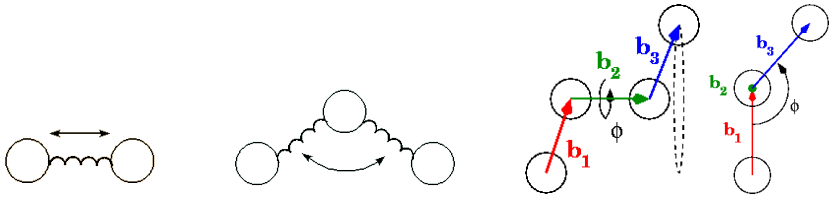
Molecular docking programs play a crucial role in drug design and development. In recent years, much attention has been devoted to the problem of finding the docking position of a flexible peptide $\mathcal{Q}$ onto a known protein $\mathcal{P}$. The automated prediction of protein-peptide interactions is one of the most challenging problems in structural biology, particularly when, having $\mathcal{Q}$ a small sequence of residues, the number of feasible docking pockets is high (see e.g. [3, 15]).
The approaches proposed until now may be classified into two kinds. The first one exploits a geometry-based approach for finding docking transformations yielding good molecular

shape complementarity (e.g., the well-known package PatchDock [15] is of this kind)+. Most of these methods consider also the peptide as a rigid body, which is a rather restrictive assumption.
The second class of methods search for the correct molecular docking by minimizing successive approximations of a given potential energy function, which are constructed by means of some suitable parameters, in order to simplify the single minimization step (see, e.g. [16], [3]). Most of these methods, allow for receptor and ligand flexibility [4, 3, 16] and adopt a minimization process based on a multi-start strategy with the steepest descent or conjugate gradient methods as local minimization tools. We note that this way of performing the optimization process gives rise to docking configurations which may often result far from the searched positions.
All these methods assume a pre-selection by the user of the position in the protein involved in the docking (choice of the receptor or binding site).

We remark that in this work no assumption is made on any prior knowledge about the docking pocket, unlike almost all analogous algorithms (e.g. DynaDock [3], Autodock [14], FDS [16]). The problem is addressed by defining a suitable mathematical model of the total potential energy of the protein-peptide interaction. Then, a minimization process of that energy function is implemented by using a filling function strategy [11], allowing the algorithm to escape from any encountered local solution.
Finally, in order to improve the docking result we adopt a two-phase optimization strategy. In particular, we use a simplified energy function including in the model only the interaction among the C_α atoms of the protein and the peptide. Since the energy contribution of intra-protein and intra-peptide atoms compounds are ignored, we introduce some constraints on the problem variables, in order to preserve the primary sequence of the given peptide.
In the second phase we reconstruct the position of all peptide atoms taking into account the computed C_α positions, and we improve the initial guess by minimizing the more complete interaction energy funtion.

2 The mathematical model

It is known [8] that in nature molecules interact and dock onto each other in such a way that the potential energy of the whole molecule system is as small as possible. Then, by constructing a mathematical representation of the potential energy and by minimizing that function, we can obtain the solution of the protein-peptide docking problem. Since the evaluation of the energy of the two interacting molecules is very expensive from a computational point of view, approximate mathematical models (often called force fields) have been defined, like for instance: AMBER [8] (Assisted Model Building with Energy Refinement); ECEPP [13] (Empirical Conformational Energy Program for Peptides); CHARMM [6] (Chemistry at HARvard Molecular Mechanics). These differ from each other in the values of some key coefficients that are included in the analytical expression of the energy



(a) (b) (c)

Figure 2: Schematic representation of E_{link} (a), E_{angle} (b) and $E_{dihedral}$ (c) energy terms

function.
According to [8], the potential energy of a compound of molecules is given by

$$E_{tot} = E_{bond} + E_{nonbond}$$

where the two terms denote the energy due to covalently bonded and non-bonded set of atoms respectively. In particular, we have

$$\begin{aligned} E_{bond} &= E_{link} + E_{angle} + E_{dihedral} \\ E_{nonbond} &= E_{VdW} + E_{Coulomb}, \end{aligned}$$

where:

$$\begin{aligned} E_{link} &= \sum_{(i,j)\in L} \frac{1}{2} K_i^b (r_{ij} - r_{ij}^0)^2, \\ E_{angle} &= \sum_{a=(i,j,h)\in A} \frac{1}{2} K_a^\theta (\theta_a - \theta_a^0)^2, \\ E_{dihedral} &= \sum_{d=(i,j,h,\ell)\in D} \frac{1}{2} K_d^\phi [1 + \cos(n\phi_d - \gamma)], \\ E_{VdW} &= \sum_{(i,j)\in \mathcal{P}\cup\mathcal{Q}} 4\epsilon_{ij} \left(\frac{\sigma_{ij}^{12}}{r_{ij}^{12}} - \frac{\sigma_{ij}^{6}}{r_{ij}^{6}} \right), \\ E_{Coulomb} &= \sum_{(i,j)\in \mathcal{P}\cup\mathcal{Q}} \frac{q_i q_j}{4\pi\epsilon_0 r_{ij}}, \end{aligned}$$

and L, A, D are sets of pairs, triplets and quadruplets of covalently bonded atoms respectively belonging to $\mathcal{P}$ and/or $\mathcal{Q}$ (see Fig. 2). The values of the constants $K_i^b, r_{ij}^0, K_a^\theta, \theta_a^0, K_d^\phi$, $n, \gamma, \epsilon_{ij}, \sigma_{ij}, q_i, \epsilon_0$ are specific of the chosen model.
Since by assumption (i) the protein is considered a rigid body, the energy contribution due to pairs, triplets and quadruplets of its atoms are negligible, so that terms E_{link}, E_{angle},

and $E_{dihedral}$ are due only to the peptide atoms. Thus, the potential energy function of our model becomes

$$E_{tot} = E_{bond}(\mathcal{Q}) + E_{nonbond}. \tag{1}$$

Moreover, observing that in the protein-peptide interaction substantially only atoms of the protein surface are involved, we further simplify the model including in term $E_{nonbond}$ only the atoms of the $\mathcal{P}$ Connolly surface [9]. Therefore, the cardinality of set $\mathcal{P}$ is considerably reduced, thus obtaining a simplified (but still sufficiently accurate) energy model with advantage in its computation.
Denoting by $x \in \mathbb{R}^{3|\mathcal{Q}|}$ the vector of peptide atoms coordinates, the protein-peptide docking problem can be formulated as:

$$\begin{array}{ll} \min & E_{tot}(x) \\ s.t. & x \in \mathcal{D}, \end{array} \tag{2}$$

where

$$\mathcal{D} = \{x \in \mathbb{R}^{3|\mathcal{Q}|} : l \leq x \leq u\},$$

with $l, u \in \mathbb{R}^{3|\mathcal{Q}|}$ such that all the minimum points of $E_{tot}(x)$ belong to $\mathcal{D}$.

3 The global optimization algorithm

In order to solve Problem (2) we use the global optimization algorithm proposed in [12]. Here we briefly describe the method and we refer to [12] for the details.
In particular, the algorithm uses a gaussian-based filling function as a tool to escape from the basin of attraction of a local minimum x_o. Let H_o be the Hessian matrix $H_o = \nabla^2 f(x_o)$ which is positive definite by assuming that any minimum point is isolated. Then, we consider the quadratic model of f at x_o, i.e.

$$q(x; x_o) = \frac{1}{2}(x - x_o)^T H_o (x - x_o), \tag{3}$$

and we consider the following function:

$$\varphi(x; x_o) = \frac{\beta}{1 - \exp(-\alpha q(x; x_o))} - \beta, \tag{4}$$

where α and β are positive scalars.
A modified objective function $\hat{f}$ is constructed by adding $\varphi(x; x_o)$ to the original function $f(x)$:

$$\hat{f}(x; x_o) = f(x) + \varphi(x; x_o). \tag{5}$$

It can be easily seen that function $\hat{f}(x; x_o)$ is substantially the original function modified only locally, i.e., obtained by "filling" f within a neighborhood of x_o, whose shape is determined by the matrix H_o, and hence according to the basin shape, and whose amplitude can be varied by taking different values of the parameter α.

Then, by applying the local optimization routine to $\hat{f}(x; x_o)$ starting from a point near x_o we obtain a stationary point $\hat{x}$ which cannot be x_o, and which (in general) is not a minimum of f.
By applying again the local optimization routine to f starting from $\hat{x}$, we get a minimum point which, either is the same minimum x_o, or is a new minimum point of f. In the first case, the value of α is not sufficiently low to escape the basin of attraction of x_o, and a lower α value is needed.

Using this simple device, we define the following global optimization procedure.
Once a minimum x_o has been found, we perform a prefixed number p of local searches applied to the modified function $\hat{f}(x; x_o)$ by setting decreasing values of the parameter α, and the corresponding local searches applied to $f(x)$. The initial value of α is chosen in such a way that the neighborhood of x_o where $f(x)$ is filled, is sufficiently small in order to locate possible minima near x_o, while the last value is taken low to discover minima far from x_o. Then, a certain number $m \leq p$ of new minimum points of f are determined. Among these, that for which the function value is the lowest, say x_ℓ, is selected as new starting point of the procedure even if $f(x_\ell) > f(x_o)$. Obviously, if $f(x_\ell) < f(x_o)$, the point x_o is replaced by x_ℓ as the current best minimum point. At the subsequent restarts, the minimum point x_ℓ corresponding to the lowest function value is discarded if it coincides with one of the minima from which the process was already restarted, and the new restart is performed from the best minimum point among the remaining. When no new minimum point is found, i.e., $m = 0$ (to avoid as much as possible this last occurrence, the number p should be taken at least of the order of some tens), or when all the new minima were already used as restarting points, the procedure ends. Obviously, it is possible to define an algorithm where a further point is chosen at random and the above procedure is repeated. In this case, to avoid repetition of the same path, this further point is discarded if the new local minimum, determined by the local search routine, is near any one of the minima from which the procedure was already restarted.
As regards the parameters α and β in function (4), in order to establish the amplitude of the neighborhood of a minimum x_o where the function f is filled, let s be a distance from x_o, and, denoting by h_m the mean of the diagonal elements of H_o, $h_m = (1/n)\sum_{j=1}^{n} h_o(j,j)$, we take

$$\alpha = \frac{9}{h_m s^2}\,, \quad \beta = \frac{\sqrt{2\pi}}{3} s\,. \tag{6}$$

Thus, in the direction along which the curvature of f is h_m, the prefixed distance s corresponds to 3σ, where σ is the standard deviation of the Gaussian function in the denominator of (4).
The procedure employs p increasing values of the prefixed distance s, and correspondingly the function f is filled within neighborhoods of x_o larger and larger.
The above procedure is formalized as follows.

Global Optimization ALgorithm (GOAL)

Data. Scalars $s = s_o$, $\varepsilon > 0$, $\Delta s > 0$, an integer $p > 1$, $\Delta x \in \mathbb{R}^n$.

Step 1. Generate at random a point $\tilde{x} \in \mathbb{R}^n$, compute $f(\tilde{x})$, and apply the local search routine for minimizing $f(x)$. Let x_o be the minimum reached.
Set $r = 1$, $x^{(r)} = x_o$, $f_{\min} = f(x_o)$ and $x_{\min} = x_o$.

Step 2. Compute $H_o = \nabla^2 f(x_o)$, and $h_m = (1/n) \sum_{j=1}^{n} h_o(j, j)$. Set $i = 1$ and $f_\ell = f(\tilde{x})$.

Step 3. Compute the parameters α and β in (6), where $s = s + \Delta s$, and starting from $x_o + \Delta x$, minimize $\hat{f}(x; x_o)$. Let $\hat{x}_i$ be the minimum reached, minimize $f(x)$ starting from $\hat{x}_i$, and let x_i be the minimum obtained.

Step 4. If $\|x_i - x^{(k)}\| > \varepsilon$, for all $k = 1, \ldots, r$, and $f(x_i) < f_\ell$, set $f_\ell = f(x_i)$, and $x_\ell = x_i$. Set $i = i + 1$, and if $i \leq p$ go to Step 3.

Step 5. If $f_\ell < f_{\min}$, set $f_{\min} = f_\ell$, and $x_{\min} = x_\ell$.
Set $r = r + 1$. If $f_\ell < f(\tilde{x})$, then set $x^{(r)} = x_\ell$, $x_o = x_\ell$, and go to Step 2; otherwise, stop.

The algorithm stops when no new minimum point is found, i.e., $m = 0$.
GOAL is included into an algorithm where the procedure is repeated starting from a suitably generated sequence of initial points. In particular, we refer to the generator of starting points DIRGEN [7], which is a DIRECT-type strategy [10].

GOALr (repeated)

Data. $l, u \in \mathbb{R}^n$. Set $x_0^* = (u - l)/2$, $\mathcal{P}_0 = \emptyset$, and $j = 0$.

Step 1. Set $j = j + 1$. If $\mathcal{P}_{j-1} = \emptyset$, then Apply DIRGEN to generate a set of initial points $\mathcal{P}_j \subset \mathcal{D}$, otherwise set $\mathcal{P}_j = \mathcal{P}_{j-1}$.
Choose a point $\tilde{x}_j \in \mathcal{P}_j$, set $\mathcal{P}_j = \mathcal{P}_j \setminus \{\tilde{x}_j\}$ and let $x_{\min,j}$ be the final minimum point obtained by GOAL sarting from $\tilde{x}_j$.

Step 2. If $f(x_{\min,j}) < f(x_{j-1}^*)$, set $x_j^* = x_{\min,j}$, otherwise set $x_j^* = x_{j-1}^*$, and go to Step 1.

As discussed in [12], Algorithm GOALr enjoys the so-called everywhere dense convergence property.

Table 1: Protein-peptide pairs description

PDB name	N	M
1A30	3	99
1AWQ	6	58
1I31	6	97
1G3F	9	69
1VWG	8	47
1AB9	10	51
1BE9	5	35

PDB name	N	M
1GUX	9	142
2FIB	4	77
1BXL	16	95
1DUZ	9	250
1F95	9	32
1YCQ	11	33
1EG4	13	33

PDB name	N	M
1IO6	10	59
1CKA	9	66
1ELW	8	28
2SEB	12	219
1CE1	8	85
1EVH	5	47
1BC5	5	98

Table 2: Results in terms of RMSD between computed and real docking

PDB name	RMSD(Å)
1A30	13.222720
1AWQ	24.318745
1I31	26.658756
1G3F	6.9376416
1VWG	7.3430915
1AB9	12.465068
1BE9	34.342587

PDB name	RMSD(Å)
1GUX	34.998142
2FIB	7.8162737
1BXL	2.7048140
1DUZ	24.925062
1F95	4.8188996
1YCQ	24.271965
1EG4	20.896873

PDB name	RMSD(Å)
1IO6	7.3286142
1CKA	60.587334
1ELW	25.336105
2SEB	32.416576
1CE1	28.857986
1EVH	15.977412
1BC5	38.954090

4 Preliminary numerical results

We implemented our method in double precision Fortran90 and run the code on an Intel core 2 duo processor with 4GB Ram under Linux operatig system by taking $\Delta s = s = 0.1$, $\epsilon = 10^{-3}$, and $p = 25$. We run GOALr starting from ten initial points (i.e. ten peptide positions) generated by DIRGEN.
We applied GOALr to proteins in complex with specific ligands which are taken from the Protein Data Bank (PDB) [5]. We selected 21 protein-peptide pairs which are described in Table 1, where we report for each pair:

- the name of the PDB entry;

- the number N and the number M of peptide and protein residues, respectively.

The results are summarized in Table 2 where we report, for each protein-peptide pair: (a) the name of the PDB entry; (b) the RMSD (root mean square distance) in Angstrom (Å) between the computed and the known peptide docking position.
These results appear to be in some cases relatively good, but in other cases not sufficiently accurate (RMSD ranging from 2.70Å to 60.59Å, and only in 6 problems over 21, RMSD is lower than 10Å).

5 A two-phase constrained programming approach

We consider here a refined optimization process for improving the preceding results by introducing a two-phase procedure. In the first one we search for the protein pocket by using a simplified potential energy function by introducing some suitable constraints. In the second phase the peptide position obtained in the first one is refined by minimizing a more complete potential energy function.

Phase 1.

Only the C_α atoms of the protein-peptide compound are considered. The simplified potential energy function includes only the following energy terms: (a) Van der Waals, (b) Coulomb, and (c) a statistically defined energy-like contribution, which are (between pairs of C_α atoms $(i,j) \in \mathcal{P} \times \mathcal{Q}$):

$$\begin{aligned} E_{VdW}^{ij} &= 4\epsilon_{ij}\left[\left(\frac{\sigma_{ij}}{r_{ij}}\right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}}\right)^{6}\right], \\ E_{C}^{ij} &= \frac{q_i q_j}{4\pi\epsilon_0 r_{ij}}, \\ E_{TD}^{ij} &= a_{ij}^{stat}\left(1 - \frac{1}{1+e^{-\gamma(r_{ij}-r_{eq})}}\right). \end{aligned}$$

In particular, the sigmoid contribution term E_{TD}^{ij} has been introduced in the knowledge-base of Thomas and Dill [17]; for each pair $(i,j) \in \mathcal{P} \times \mathcal{Q}$, the energy-like coefficient a_{ij}^{stat} measures how much the two atoms are statistically likely to contact each other. Thus, our simplified function is

$$E'_{tot} = \sum_{(i,j)\in\mathcal{P}\times\mathcal{Q}} \left(E_{VdW}^{ij} + E_{C}^{ij} + wE_{TD}^{ij}\right), \tag{7}$$

where w is a weighting coefficient used to modulate the relative influence between the third and the first two terms.
In order to preserve $\mathcal{Q}$ primary sequence and to ensure full $\mathcal{Q}$ flexibility, we add some simple constraints:

$$\begin{aligned} r_{i,i+1}^{c} &= 3.8\text{Å}, \qquad \forall\ i \in \mathcal{Q},\ i = 1,\ldots,N-1, && (8)\\ r_{i,k}^{c} &\geq 3.8\text{Å}, \qquad \forall\ i,k \in \mathcal{Q},\ i = 1,\ldots,N-2, k = i+2,\ldots,N, && (9)\\ (r_{1,N}^{c})^2 &\geq 0.4\ell_{max}^2, && (10) \end{aligned}$$

where $r_{i,j}^{c}$ denotes the distance between the i-th and j-th C_α atoms of the peptide and $\ell_{max} = 3.8(N-1)$Å is the maximum length of the peptide C_α atom chain.

Constraints (8) and (9) guarantee that the peptide primary sequence is not modified, while (10) is needed to avoid that the structure results too twisted (that is, the squared distance between the first and the last C_α $\mathcal{Q}$ atoms must be no less than 40% of ℓ^2_{max}).

Then, the first phase consists in solving the following nonlinear constrained problem:

$$\begin{array}{ll} \min\limits_{x\in\mathbb{R}^{3N}} & E'_{tot}(x) \\ s.t. & (8), (9), \text{ and } (10). \end{array} \tag{11}$$

We observe that Algorithm GOALr for unconstrained global optimization problems may also be used to solve this problem, provided that a local constrained optimization routine is available (we employ the augmented Lagrangian algorithm ALGENCAN described in [1, 2]).

Phase 2.

Once the positions of the $\mathcal{Q}$ C_α atoms are calculated, we consider the more complete energy function $E_{tot}(x)$ (1) and we add to it a regularization term in order to penalize the distances d_i, $i = 1, \ldots, N$, between the i-th $\mathcal{Q}$ C_α atom and its corresponding position obtained in the first phase:

$$\begin{array}{ll} \min & E_{tot}(x) + \sum\limits_{i=1}^{N} d_i^2 \\ s.t. & x \in \mathbb{R}^{3|\mathcal{Q}|}. \end{array} \tag{12}$$

6 Final numerical results and conclusion

The final results have been obtained by taking, in term E^{ij}_{TD}, $\gamma = 4$ and $r_{eq} = 7.5$.
In Fig. 3 we show, as an example, the computed docking position for the case 1A30.
In order to evaluate the influence of the weighting parameter w in the modified potential energy (7), we report in Table 3 the results obtained by taking $w = 0, 1, 10, 20$.
We consider as good any result for which RMSD < 10Å, i.e. the solution can be considered sufficiently close to the true position (the number of such cases over 21 is reported at the end of Table 3). We observe that in the column $w = 10$ the mean of RMSD over 21 cases is the lowest. From these results, it can be concluded that the influence of the statistical potential term E_{TD} has a relative importance.

Finally, we report in Table 4 the comparison of the results with $w = 10$ (Table 3) with those obtained by applying the software package PatchDock [15] by using its default settings and starting from the same 10 peptide positions used by GOALr.
We observe that our algorithm gives good results (RMSD < 10Å) on 17 cases over 21, while PatchDock on 6 over 21. Moreover, we note that our worst result is 1CE1 with RMSD=18.44Å while that in the second column is 1CKA with RMSD=59.45Å.

Table 3: Influence of the statistical potential term E_{TD} with $w = 0, 1, 10, 20$

PDB name	$w = 0$ RMSD (Å)	$w = 1$ RMSD (Å)	$w = 10$ RMSD (Å)	$w = 20$ RMSD (Å)
1A30	8.405454	4.1021442	4.624203	5.017705
1AWQ	9.324394	8.6659565	7.666582	7.204328
1I31	9.145371	8.9724503	9.277303	9.010107
1G3F	7.2817044	7.0847816	10.022137	10.56941
1VWG	8.511778	9.9112635	7.932511	8.555477
1AB9	7.2132416	8.4754601	7.9945745	7.586982
1BE9	6.7077765	5.4053221	5.9691744	6.318897
1GUX	18.004704	16.225876	13.140095	13.14232
2FIB	7.701962	6.9578347	8.358985	8.296103
1BXL	6.5164466	3.744463	3.7833543	3.997882
1DUZ	19.590076	9.2800999	7.613755	19.65474
1F95	7.091435	6.9069061	7.9120264	8.162895
1YCQ	6.3804975	5.5156422	6.6956816	6.545094
1EG4	6.467807	7.1725159	6.5862813	6.062716
1IO6	4.4245906	4.8167729	4.2974496	4.168911
1CKA	3.5217075	5.2367024	3.4743752	4.817621
1ELW	6.946433	5.5941148	5.3917737	5.373043
2SEB	6.9365654	11.290374	7.027582	6.813974
1CE1	20.054298	19.159685	18.438683	20.04466
1EVH	10.398325	10.793805	9.914731	9.801800
1BC5	21.662464	19.027071	16.828922	19.51020
RMSD < 10Å	16	16	**17**	**17**
mean of RMSD	9.63272	8.778059	**8.235723**	9.078803

It is likely that, by allowing aklso for receptor flexibility, the results could be even more accurate.

The results obtained show that our approach may be considered a valuable alternative to other methods proposed in the literature.

References

[1] R. Andreani, E.G. Birgin, J.M. Martinez, and M.L. Schuverdt. On augmented lagrangian methods with general lower-level constraints. *SIAM Journal on Optimization*, 18:1286–1309, 2007.

Table 4: Distances between computed and known positions

PDB name	GOAL RMSD (Å)	PatchDock RMSD (Å)
1A30	4.624203	15.969474
1AWQ	7.666582	24.954113
1I31	9.277303	22.708910
1G3F	10.022137	6.327445
1VWG	7.932511	7.602935
1AB9	7.9945745	11.081827
1BE9	5.9691744	32.219646
1GUX	13.140095	32.202145
2FIB	8.358985	9.243341
1BXL	3.7833543	3.931751
1DUZ	7.613755	25.406809
1F95	7.9120264	3.985806
1YCQ	6.6956816	24.322502
1EG4	6.5862813	18.821020
1IO6	4.2974496	5.870293
1CKA	3.4743752	59.454960
1ELW	5.3917737	23.193996
2SEB	7.027582	31.363981
1CE1	18.438683	28.343372
1EVH	9.914731	15.931898
1BC5	16.828922	37.939034
RMSD < 10Å	17	6

[2] R. Andreani, E.G. Birgin, J.M. Martinez, and M.L. Schuverdt. Augmented lagrangian methods under the constant positive linear dependence constraint qualification. *Mathematical Programming*, 111:5–32, 2008.

[3] I. Antes. Dynadock: A new molecular dynamics-based algorithm for proteinpeptide docking including receptor flexibility. *Proteins: Structure, Function and Bioinformatics*, 78:1084–1104, 2010.

[4] J. Apostolakis, A. Plückthun, and A. Caflisch. Docking small ligands in flexible binding sites. *Journal of Computational Chemistry*, 19:21–37, 1998.

[5] H.M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T.N. Bhat, H. Weissig, I.N. Shindyalov, and P.E. Bourne. The protein data bank. *Nucleic Acids Research*, 28:235–242, 2000.

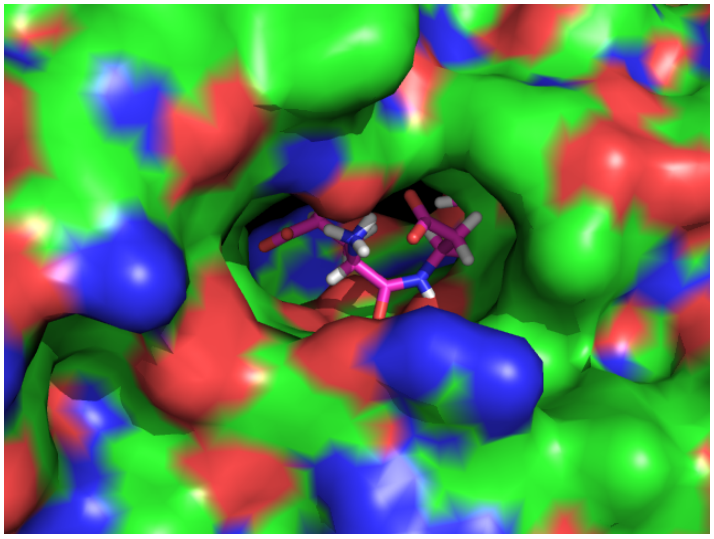


Figure 3: Computed docking position for the 1A30 protein-peptide pair

[6] B. R. Brooks, C. L. Brooks III, A. D. Mackerell, L. Nilsson, R. J. Petrella, B. Roux, Y. Won, G. Archontis, C. Bartels, S. Boresch, A. Caflisch, L. Caves, Q. Cui, A. R. Dinner, M. Feig ane S. Fischer, J. Gao, M. Hodoscek, W. Im, K. Kuczera, T. Lazaridis, J. Ma, V. Ovchinnikov, E. Paci, R. W. Pastor, C. B. Post, J. Z. Pu, M. Schaefer, B. Tidor, R. M. Venable, H. L. Woodcock, X. Wu, W. Yang, D. M. York, and M. Karplus. Charmm: The biomolecular simulation program. *Journal of Computational Chemistry*, 30:1545–1615, 2009.

[7] E.F. Campana, G. Liuzzi, S. Lucidi, D. Peri, V. Piccialli, and A. Pinto. New global optimization methods for ship design problems. *Optimization and Engineering*, 10:533–555, 2009.

[8] D.A. Case, T.A. Darden, T.E. Cheatham III, C.L. Simmerling, J. Wang, R.E. Duke, R. Luo, R.C. Walker, W. Zhang, K.M. Merz, B. Roberts, S. Hayik, A. Roitberg, G. Seabra, J. Swails, A.W. Goetz, I. Kolossváry, K.F. Wong, F. Paesani, J. Vanicek, R.M. Wolf, J. Liu, X. Wu, S.R. Brozell, T. Steinbrecher, H. Gohlke, Q. Cai, X. Ye, J. Wang, M.-J. Hsieh, G. Cui, D.R. Roe, D.H. Mathews, R. Salomon-Ferrer M.G. Seetin and, C. Sagui, V. Babin, T. Luchko, S. Gusarov, A. Kovalenko, and P.A. Kollman. Amber 12. Technical report, University of California, San Francisco, 2012.

[9] M.L. Connolly. Analytical molecular surface calculation. *Journal of Applied Crystallography*, 16:548–558, 1983.

[10] D.R. Jones, C.D. Perttunen, and B.E. Stuckman. Lipschitzian optimization without the lipschitz constant. *Journal of Optimization Theory and Applications*, 79(1):157–181, 1993.

[11] F. Lampariello. A filling function method for continuos unconstrained global optimization: application to morse clusters. Technical Report R.615, IASI-CNR, 2004.

[12] F. Lampariello and G. Liuzzi. A filling function method for unconstrained global optimization. 2012. Submitted to Computational Optimization and Applications.

[13] F.A. Momany, R.F. McGuire, A.W. Burgess, and H.A. Scheraga. Energy parameters in polypeptides. vii. geometric parameters, partial atomic charges, nonbonded interactions, hydrogen bond interactions, and intrinsic torsional potentials for the naturally occurring amino acids. *Journal of Physical Chemistry*, 79:2361–2381, 1975.

[14] G.M. Morris, D.S. Goodsell, R.S. Halliday, R. Huey, W.E. Hart, R.K. Belew, and A.J. Olson. Automated docking using a lamarckian genetic algorithm and an empirical binding free energy function. *Journal of Computational Chemistry*, 19:1639–1662, 1998.

[15] D. Schneidman-Duhovny, Y. Inbar, R. Nussinov, and H.J. Wolfson. Patchdock and symmdock: servers for rigid and symmetric docking. *Nucleic Acids Research*, 33:363–367, 2005.

[16] R. D. Taylor, P. J. Jewsbury, and J. W. Essex. FDS: Flexible ligand and receptor docking with a continuum solvent model and soft-core energy function. *Journal of Computational Chemistry*, 24:1637–1656, 2003.

[17] P.D. Thomas and K.A. Dill. An iterative method for extracting energy-like quantities from protein structures. *Proceedings of the National Academy of Science, USA*, 93:11628–11633, 1996.