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

Francesco Lampariello, Giampaolo Liuzzi
Istituto di Analisi dei Sistemi ed Informatica (IASI)
CNR
Rome, Italy
giampaolo.liuzzi@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 known protein is sought. In this work we present a new docking algorithm which is based on the use of a filling function method for continuos constrained global optimization. Indeed, the protein-peptide docking position is sought by minimizing the conformational potential energy subject to constraints necessary to maintain the primary sequence of the given peptide. The resulting global optimization problem is difficult mainly for two reasons. First, the problem is large scale in constrained global optimization; second, the energy function is multivariate non-convex so that it has many local minima. The method is based on the device of modifying the original objective function once a local minimum has been attained by adding to it a filling term. This allows the overall algorithm to escape from local minima thus, ultimately, giving the algorithm ability to explore large regions in the peptide conformational space. We present numerical results on a set of benchmark docking pairs and comparison with the well-known software package for molecular docking PacthDock.

Keywords-Protein-peptide docking; potential reduction; continuous global optimization

1. Introduction

In this paper we address the problem of docking small peptide molecules, for example, drug candidates, onto a given protein model. Molecular docking programs play a crucial role in drug design and development. In recent years, much attention has been devoted to this problem where docking of a flexible peptide with a known protein is sought. We consider a docking algorithm based on the use of a filling function method for continuos unconstrained global optimization [1].

The correct protein-peptide docking position is obtained by minimizing the function representing the total potential energy according to a specific mathematical model. In order to preserve the primary sequence of the given peptide it is necessary to take into account some constraints on the problem variables, and then we construct the Lagrangian of the original problem. The resulting optimization problem has two main features; it is a large-scale one in constrained global optimization, and the total potential energy function has many local minima. Once a local minimum has been found, the method modifies the original objective function by adding to it a filling term. This allows the algorithm to escape from the local minimum so that it may explore large regions in the search space.

The approaches proposed in the literature are based on the minimization of successive approximations of some potential energy function, obtained by introducing some suitable parameters (see, e.g. [2], [3]) in order to simplify the single minimization step. Moreover, these methods determine the docking between the peptide and a prefixed small part of the protein, namely the prefixed receptor or binding site. Most of these methods allow for receptor flexibility [4], [3], [2]. The minimization process is based on a multi-start strategy that uses the steepest descent or conjugate gradient methods as local minimization tools.

2. The Energy Model

As regards the mathematical model employed, we assume that the protein tertiary structure and the peptide primary residue sequence are known and we denote by N and M the number of peptide and protein residues, respectively.

In order to find the docking pocket and position of the peptide onto the protein, we consider that peptide is flexible in such a way that all its atoms have three degrees of freedom. Moreover, the given peptide primary residue sequence is not modified while calculating the docking position by means of suitable simple nonlinear constraints. Let

$$E_{LJ}^{ij} = 4\epsilon_{ij}\left[\left(\frac{\sigma_{ij}}{r_{ij}}\right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}}\right)^{6}\right], E_{C}^{ij} = \frac{q_i q_j}{4\pi\epsilon_0 r_{ij}}$$

be the Lennard-Jones and Coulomb potentials, respectively. They represent the interaction between protein atom *i* and peptide atom *j* and r_{ij} denotes their distance. Hence, we consider the following total potential energy function [3],

$$E(r) = \sum \left(E_{LJ}^{ij} + E_{C}^{ij}\right) \qquad (1)$$

where the summation is calculated over all pairs ($N\times M$) of atoms.

3. Global Optimization Approach

Here we consider the problem of finding the docking position of a given peptide by taking into account all the given protein atoms, so that the binding site is not prefixed, but directly determined by the algorithm. In order to avoid that the peptide primary sequence is modified, we consider the following constraints between the carbon alpha atoms of the peptide residues:

$$r^{c}_{i,i+1}=3.8,\ \forall i=1,\ldots,N-1, \qquad (2)$$

and

$$r^{c}_{i,k}\geq 2,\ \forall i=1,\ldots,N-2, k=i+2,\ldots,N. \qquad (3)$$

Therefore, our algorithm computes the final docking position by searching for a global minimum of *E(r)* subject to (2) and (3).

For the sake of clarity, let us consider the general constrained nonlinear global optimization problem

$$\begin{array}{l} globminf(x) \\ s.t.\, x\in F, \end{array} \qquad (4)$$

where $x\in\Re^{n}$,

$$F=\{x\in\Re^{n} : g(x)\leq 0\},$$

$f:\Re^{n}\rightarrow\Re\ ,g:\Re^{n}\rightarrow\Re^{m}.$

Let us assume, as usual, that *f* and *g* are continuously differentiable functions and that *f (x)* is radially unbounded on the feasible set. Now, let $L(x,\lambda)=f(x)+\lambda^{T}g(x),$ with $\lambda\in\Re^{m},$ be the Lagrangian function for Problem (4). A pair $(x^{¿},\lambda^{¿})$ is a Karush-Kuhn-Tucker (KKT) pair if [5]

$$g(x^{¿})\leq 0,\ \ g(x^{¿})^{T}\lambda^{¿}=0,\ \ \lambda^{¿}\geq 0,\ \nabla L(x^{¿},\lambda^{¿})=0. \qquad (5)$$

Under some mild regularity assumptions on the constraint functions, if $x^{¿}$ is a local minimum point of Problem (4), we have [5]:

Proposition 1: Let $x^{¿}$ be a local minimum of Problem (4). Then, a vector $\lambda^{¿}\in\Re^{m}$ of KKT multipliers exists such that $(x^{¿},\lambda^{¿})$ is a KKT pair. Furthermore, the hessian of the Lagrangian function $\nabla^{2}_{x}L(x^{*},\lambda^{*})$ is positive definite.

Finally, we assume that a local optimization routine is available, namely, a routine that, given an arbitrary starting point x, provides a local minimum of Problem (4). More in particular, as local optimization engine we use the augmented Lagrangian algorithm ALGENCAN [6], [7].

A. The filling function technique

Starting from a point randomly chosen in $\Re^{n}$, let x_0 be the local minimum reached by the local optimization routine and $\lambda_0\in\Re^{m}$ be the vector of corresponding KKT multipliers.

In order to search over the various minima for finding that corresponding to the lowest f value, we have to escape the basin of attraction of the current minimum x_0. To do this, we consider the quadratic model of Problem (4) in x_0, that is,

$$q(x;x_0)=\frac{1}{2}(x-x_0)^{T}H_0(x-x_0) \qquad (6)$$

where H_0 is the Hessian of the Lagrangian function for Problem (4) computed at x_0 Then, we construct the following gaussian-based function

$$\varphi(x;x_0)=\frac{\beta}{1-\exp(-\alpha q(x;x_0))}-\beta \qquad (7)$$

where α and β are positive scalars, and we consider the new function obtained by adding to the original function $f(x)$

$$\tilde{f}(x;x_0)=f(x)+\varphi(x;x_0). \qquad (8)$$

Since

$$q(x_0;x_0)=0$$

we have

$$\varphi(x_0;x_0)=+\infty$$

so that

$$\tilde{f}(x_0;x_0)=+\infty.$$

Moreover, since $\varphi(x;x_0)>0,\ \ for\ allx,$

$$\tilde{f}(x;x_0)>f(x),\ \ \forall x.$$

Then,

$$\varphi(x;x_0)\rightarrow 0,\ \text{and}\ \tilde{f}(x;x_0)\rightarrow f(x)$$

as $\|x-x_0\|\rightarrow+\infty$, that is far from x_0 in any direction.

Therefore, function (8) is substantially the original function modified only locally, i.e., obtained by “filling” *f* within a neighborhood of x_0, whose aplitude can be varied by taking different values of the parameter α.

Thus, by applying the local search routine to starting from a point near x_0, we reach a minimum point $\hat{x}$ which can not be x_0 and which is not a minimum of *f.*

Now, by reapplying the local search to the original function *f (x)* starting from $\hat{x}$, either the same point x_0 is reached, or a new minimum point is found.

In the first case, the value of α is not sufficiently low to escape the basin of attraction of x_0, and a lower α value is needed.

B. The global optimization procedure

The global optimization algorithm can be summarized in the following scheme.

Global Optimization Algorithm (GOAL)

Data. $s_0, \Delta s, \epsilon$, integers $p \geq 10$, and $R \geq 1$.

Step 0. Set $r=1$ and $\lambda=1$.

Step 1. Generate at random a point $\tilde{x} \in \Re^n$, compute $f(\tilde{x})$ and apply the local search routine for minimizing *f(x)*. Let x_0 be the minimum reached; set $x^{(r)}=x_0$.

Step 2. If $r=1$, set $f_{min}=f(x_0)$, $x_{min}=x_0$, else, if $\|x_0 - x^{(k)}\| < \epsilon$, for an index $k=1,\dots,r-1$, set $\lambda=\lambda+1$, and if $\lambda \leq r$ go to Step 1; otherwise STOP.

Step 3. Set $i=1$ and $f_l=f(\tilde{x})$.

Step 4. Compute the parameters (see (9) below), where $s=s_0+(i-1)\Delta s$.

Step 5. Starting from $x_0+\Delta x$, minimize $\hat{f}(x;x_0)$ 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 6. If $\|x_i - x^{(k)}\| > \epsilon$, for all $k=1,\dots,r$, and $f_l<f(x_i)$, set $f_l=f(x_i)$, and $x_l=x_i$. Set $i=i+1$, and if $i \leq p$ go to Step 4.

Step 7. If $f_l<f_{min}$, set $f_{min}=f_l$, and $x_{min}=x_l$. If $f_l<f(\tilde{x})$, then set $x^{(r)}=x_l, x_0=x_l$, and go to Step 3; otherwise, set $\lambda=1$and go to Step 1. If $r<R$, set $r=r+1$; otherwise STOP.

The value of ϵ which is the lowest distance between two minimum points over which they are considered distinct, should be chosen taking into account the stopping criterion of the local search routine.

Note that the procedure, differently from the classical multistart algorithm, terminates after the prefixed maximum number *R* of restarts have been performed, regardless the number of initial points chosen at random. Thus, it is even possible that a new restarting minimum point is always found, i.e., from Step 6 the algorithm never returns to Step 1, so that only one initial random point is employed.

As regards the parameters α and β in function (7), in order to establish the amplitude of the neighborhood of a minimum xo where the function f is filled, we take

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

where *s* is the distance from x_0. Thus, the prefixed distance *s* corresponds to 3σ, where σ is the standard deviation of the Gaussian function in (6). The procedure employs *p* increasing values of the prefixed distance *s*, and correspondingly the function *f* is filled within neighborhoods of x_0 larger and larger.

4. Preliminary Results

In this Section we present some preliminary numerical results and comparison with the well-known software package for molecular docking PatchDock [8], [9] in terms of computed protein-peptide dockings for 23 protein-peptide pairs. 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 using $\Delta s=s_0=0.1, \epsilon=10^{-3}, p=25$and $R=50$.

We applied our method to proteins in complex with specific ligands which are taken from the PDB [10]. The obtained results are summarized in Table I where we report, for each protein-peptide pair: (a) the name of the PDB entry; (b) the number of residues, NPE and MPR, composing, respectively, the peptide and the protein; (c) the avarage RMSD (root mean square distance) in Angstrom (A) between the computed and the known peptide docking position obtained by running our code (GOAL) and PatchDock from ten randomly chosen starting peptide positions.

In the table we use boldface to highlight a success. As it can be seen, GOAL outperforms PatchDock on 19 out of 23 pairs.

TABLEI DISTANCES BETWEEN COMPUTED AND KNOWN POSITIONS

			GOAL	**PatchDock**
PDB name	**NPE**	**MPR**	**RMSD**	**RMSD**
1A30(A)	3	99	**9.7270784**	15.969474
1A30(B)	3	99	**10.265117**	15.969474
1AWQ	6	58	**8.6928129**	24.954113
1I31	6	97	**7.7272010**	22.708910
1G3F	9	69	6.9038916	**6.327445**
1VWG	8	47	8.6765423	**7.602935**
1AB9	10	51	**7.9983487**	11.081827
1BE9	5	35	**6.5084529**	32.219646
1GUX	9	142	**9.7264299**	32.202145

2FIB	4	77	**6.9589953**	9.243341
1BXL	16	95	5.6981535	**3.931751**
1DUZ	9	250	**4.5037251**	25.406809
1F95	9	32	7.3566127	**3.985806**
1YCQ	11	33	**5.9804459**	24.322502
1EG4	13	33	**7.1259413**	18.821020
1IO6	10	59	**2.7675009**	5.870293
1CKA	9	66	**5.8080659**	59.454960
1ELW	8	28	**4.2639108**	23.193996
2SEB	12	219	**10.671972**	31.363981
1CE1	8	85	**19.134851**	28.343372
1PAU	4	52	**17.125254**	52.701431
1EVH	5	47	**11.544429**	15.931898
1BC5	5	98	**18.764696**	37.939034

As we can see, the method is able to guess the peptide docking position with a maximum r.m.s.d of 10.7A (for 2SEB), meaning that the region of correct binding has been located by the method. Then, it is reasonable to think that the computed position can be further refined to the exact position, by allowing for receptor flexibility, applying, for instance, a proviously proposed tool (e.g. DynaDOCK [3], AutoDOCK [11], FDS [2]) which is the subject of ongoing work.

5. Conclusions

In the paper we present a global optimization method based on a gaussian filling function applied to the protein-peptide docking problem. Indeed, the problem is defined as that of minimizing the potential energy function (1) subject to some constraints necessary to preserve the peptide primary sequence. The preliminary numerical results obtained, in terms of r.m.s.d. values, show viability of the proposed approach.

We are currently trying to improve the approach by considering more accurate potential energy functions. Moreover, we are attempting to exploit to a greater extent the chemical-physical properties of the atoms composing the peptide and the given protein, and to use some refinement tools for improving the computed final docking position.

REFERENCES

[1] F. Lampariello, “A filling function method for continuos unconstrained global optimization: application to morse clusters,” IASI-CNR, Tech. Rep. R.615, 2004.

[2] 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, vol. 24, pp. 1637–1656, 2003.

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

[4] J. Apostolakis, A. Pluckthun, and A. Caflisch, “Docking small ligands in flexible binding sites,” Journal of Computational Chemistry, vol. 19, pp. 21–37, 1998.

[5] D. Bertsekas, Nonlinear Programming. Massachusetts, USA: Athena Scientific, 1999.

[6] R. Andreani, E. Birgin, J. Martinez, and M. Schuverdt, “On augmented lagrangian methods with general lower-level constraints,” SIAM Journal on Optimization, vol. 18, pp. 1286–1309, 2007.

[7] ——, “Augmented lagrangian methods under the constant positive linear dependence constraint qualification,” Mathematical Programming, vol. 111, pp. 5–32, 2008.

[8] D. Duhovny, R. Nussinov, and H. Wolfson, “Efficient unbound docking of rigid molecules,” in Proceedings of the 2’nd Workshop on Algorithms in Bioinformatics(WABI) Rome, Italy, ser. Lecture Notes in Computer Science 2452, Gusfield et al., Ed. Springer Verlag, 2002, pp. 185–200.

[9] D. Schneidman-Duhovny, Y. Inbar, R. Nussinov, and H. Wolfson, “Patchdock and symmdock: servers for rigid and symmetric docking,” Nucleic Acids Research, vol. 33, pp. 363–367, 2005.

[10] H. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. Bhat, H. Weissig, I. Shindyalov, and P. Bourne, “The protein data bank,” Nucleic Acids Research, vol. 28, pp. 235–242, 2000.

[11] G. Morris, D. Goodsell, R. Halliday, R. Huey, W. Hart, R. Belew, and A. Olson, “Automated docking using a lamarckian genetic algorithm and an empirical binding free energy function,” Journal of Computational Chemistry, vol. 19, pp. 1639–1662, 1998.