PRODOCK: Software Package for Protein Modeling and Docking

JEAN-YVES TROSSET, HAROLD A. SCHERAGA

Baker Laboratory of Chemistry and Chemical Biology, Cornell University, Ithaca, New York 14853-1301

Received 2 May 1998; accepted 27 October 1998

ABSTRACT: A new software package, PRODOCK, for protein modeling and flexible docking is presented. The protein system is described in internal coordinates with an arbitrary level of flexiblity for the proteins or ligands. The protein is represented by an all-atom model with the ECEPP/3 or AMBER IV force field, depending on whether the ligand is a peptidic molecule or not. PRODOCK is based on a new residue data dictionary that makes the programming easier and the definition of molecular flexibility more straigthforward. Two versions of the dictionary have been constructed for the ECEPP/3 and AMBER IV geometry, respectively. The global optimization of the energy function is carried out with the scaled collective variable Monte Carlo method plus energy minimization. The incorporation of a local minimization during the conformational sampling has been shown to be very important for distinguishing low-energy nonnative conformations from native structures. To make the Monte Carlo minimization method efficient for docking, a new grid-based energy evaluation technique using Bezier splines has been incorporated. This article includes some techniques and simulation tools that significantly improve the efficiency of flexible docking simulations, in particular forward/backward polypeptide chain generation. A comparative study to illustrate the advantage of using quaternions over Euler angles for the rigid-body rotational variables is presented in this paper. Several applications of the program PRODOCK are also discussed. © 1999 John Wiley & Sons, Inc. J Comput Chem 20: 412–427, 1999

Keywords: docking; Monte Carlo minimization; molecular modeling; quaternions; optimization

Correspondence to: H. A. Scheraga; e-mail: has5@cornell.edu

Contract/grant sponsor: Association Française pour la Recherche Thérapeutique

Contract/grant sponsor: Centre National de la Recherche Scientifique IMABIO

Contract/grant sponsor: National Science Foundation; contract/grant number: MCB95-13167

Contract/grant sponsor: National Institutes of Health; contract/grant numbers: GM-14312 and HL-30616

Contract/grant sponsor: Cornell Biotechnology Center

Contract/grant sponsor: Cornell Theory Center (funded in part by the NSF, New York State, IBM, NIH National Center for Research Resources Grant P41 RR-04293, CTC Corporation Partnership Programs)

CCC 0192-8651 / 99 / 040412-16

Introduction

Determination of the structure of a receptor–ligand complex is a prerequisite for understanding the affinity of an enzyme for various inhibitors. Once the structure is determined, comparative free energy calculations can be carried out on those different inhibitors. One of the main approaches for drug discovery is to screen a large molecular database that is usually generated beforehand by combinatorial chemistry for that particular target receptor. The docking problem thus amounts to determining the structure of the bound ligand correctly, for the binding free energy calculation to be relevant and fast enough for screening a large molecular database in a reasonable amount of time.

This docking problem faces two major challenges common to protein folding and all other projects in protein modeling: The first is to find an energy function for which the global minimum corresponds to the experimentally observed structure of the receptor–ligand complex, or of a single polypeptide chain if one is interested in protein modeling or protein folding. The second challenge is to find the global minimum of this energy function. This is the multiple-minima problem. Finding an energy function having this ''good'' property is in fact a challenge on its own. The reason resides in the difficulty for a given protein model (e.g., lattice model, united residue model, or all-atom model) to have a corresponding potential energy function precise enough so that the global energy minimum of the function corresponds to the structure observed experimentally. For major projects such as docking, free energy calculations, loop modeling, chain packing, etc., an atomic resolution model is required. However, certain simplifications in the molecular model are necessary for the calculations to be realizable in a reasonable amount of time. In loop modeling and docking, the approximation consists of assigning flexibility to a certain part of the system only; for instance, the ligand or the loop, plus eventually some residues of the target receptor, the rest of it being maintained fixed at the X-ray observed atomic coordinates. If one is interested in the dynamics of a single polypeptide chain, approximations can be made by freezing some dihedral angles of the molecules to decrease the total number of variables.

One way to realize this easily, despite substantial programming, is to describe the molecule in internal and external coordinates where the variables are the bond lengths, bond angles, dihedral angles, and translational and rotational degrees of freedom. The great advantage of this representation over Cartesian atomic coordinates is that the variables are separated according to their different time scales. The perturbation step size or time step can be adapted to each of them, making the whole conformational search more efficient in Monte Carlo (MC) or molecular dynamics (MD) simulations. This is especially true for docking, for which the tumbling of the ligand in the active site is independent of its internal conformational changes, making the conformational sampling more efficient. It is also easier for the user to control the different types of conformational moves.

The second advantage of the internal coordinate representation is that it is easy to control the flexibility of the molecular system under study. The number of degrees of freedom can be reduced to an optimum by freezing any type of variables, or only a subset of a given type. The most common approach, which is adopted in the ECEPP/3 force field,[1–4] is the rigid geometry approximation for which bond lengths and bond angles are kept fixed at standard values. This enables one to reduce the number of variables by a factor of about ten compared with a representation in Cartesian coordinates. The main drawback is that, for some cases, the energy barriers between two conformational states can be artificially large because of fixed bond lengths and fixed bond angles.

Although the representation in Cartesian coordinates is useful for refining structures, it becomes less advantageous for sampling the conformational space of a protein. An explicit comparison between the Cartesian and dihedral angle representations has been reported by Abagyan et al.[5] They showed that the radius of convergence of the energy minimization procedure is larger in dihedral angle space than in Cartesian coordinate space. They made the comparison by counting the number of times that a perturbed protein structure returned back to its energy minimum after various degrees of perturbation. For large perturbations, structures do not return to their lowest energy minimum after Cartesian energy minimization, whereas most of them were minimized back to the global minimum using the dihedral angle space representation.

In this article, we present a new software package, PRODOCK, which can be used for docking a ligand of any type (small organic molecules to proteins) onto a target protein receptor. It is also

well adapted for protein modeling such as calculating protein loops, chain packing, and studying protein dynamics and thermodynamics. Two general principles were kept in mind while developing the software: (i) the possibility of defining an arbitrary level of molecular flexiblity easily; and (ii) defining the molecule with enough information for the implementation of new codes (e.g., optimization techniques) to be straightforward.

To meet these two requirements, we created a new residue data dictionary that contains all the basic information concerning atomic connectivity, atom name, atom type, internal geometry coordinates, as well as information concerning the relation between the internal variables and the corresponding atoms moved by those variables. This information is used to build a few fundamental arrays that are used in almost all the major subroutines of the program. All this information is independent of the force field, except for six columns of the dictionary: the ones giving the atom name; the atom type; the type of torsional potential of the bond preceding the given atom; and the three values for bond lengths, bond angles, and dihedral angles.

In PRODOCK, the molecules are described with an all-atom model using the ECEPP/3[4] or AMBER[6] force field depending on the type of the ligand, whether it is a peptide or a protein, or an organic molecule. The molecule is described in internal coordinates for which an arbitrary level of flexibility can be assigned to the ligand and the receptor. For each force field, the rigid approximation was used; that is, the bond lengths and bond angles were always kept fixed at standard values. This is the underlying approximation in ECEPP/3 but not in the AMBER force field for which bond lengths and bond angles are allowed to vary. Implementation of the full internal coordinate model of the molecule is much more demanding in terms of programming and has not yet been implemented in PRODOCK. In this respect, docking structures obtained with the AMBER IV force field may need to be further refined using the original AMBER potential; that is, with the energy terms associated with the bond length elongation and bond angle deformation. With the ECEPP/3 force field, none of the geometry or energy parameters have been modified. The changes concern only the procedure to store the information that makes the programming easier and the definition of flexibility more straighforward.

A standard residue data dictionary has thus been constructed for the ECEPP/3 and AMBER force fields. For drug molecules, an independent AMBER dictionary is necessary. At the present stage, this is done manually knowing the Cartesian coordinates, atom names, and connectivity matrix of the compound. The type of dictionary used in PRODOCK was very much inspired by the dictionary developed by Robson and Platt,[7] and later improved by Voll and used by Higo et al.[8,9]

The global optimization tool used in PRODOCK is the scaled collective variables Monte Carlo (SCVMCM) method developed by Noguti and Gō[10] with energy minimization after each MC step.[11,12] Energy minimization was shown to be one of the best techniques for distinguishing between native- and nonnative-generated conformations of ligands bound to their receptors.[13] Incorporation of this technique into a Monte Carlo procedure enables one to distinguish the native conformation directly during the conformational search. It avoids the generation of a large number of ligand conformers for which more sophisticated energy evaluation tools would have had to be applied to identify the native-like conformations. The efficiency of the Monte Carlo minimization (SCV-MCM) was greatly improved by incorporating a new grid-based energy evaluation technique using Bezier splines, recently developed in our laboratory.[13,14] The Bezier spline technique enables one to estimate not only the energy at a given point of the continuous 3D space from its surrounding grid points but also the first and all derivatives of the energy, if necessary. Full advantage of the grid technique can thus be taken in the local energy minimization procedure. This improvement can speed up the SCV-MCM method by a factor of ten up to a few hundred, according to the size of the rigid part of the protein receptor.

A general description of this new package is given in this study. The following sections are devoted to a description of the new ECEPP/3 and AMBER residue data dictionaries, the definition of molecular flexibility, and the presentations of the Monte Carlo and simulated annealing protocols used for simulating the dynamics of a protein and for searching for the global minimum of the energy of the system. Some comparison tests to assess the efficiency of different features of the program have also been carried out. These include improvement in the procedure for generating the polypeptide chain, and the advantage of using quaternion parameters over Euler angles for rotational degrees of freedom in docking computations. The differents applications of PRODOCK will be reviewed in the Discussion section. These in-

clude rigid[13] and flexible[15,16] docking simulations using Bezier splines, structure refinement using NMR data,[15] and loop modeling.[17] They incorporate the same features of the program presented here.

Methods

ECEPP/3 AND AMBER IV DICTIONARY

Defining the molecule is the first step of every molecular mechanics program. In a representation in Cartesian coordinates, it is necessary to know only the names of the atoms that constitute the molecule, and the connections between them. In the rigid geometry representation (dihedral angle space), the standard values of bond lengths and bond angles are also needed. This information is coded in a dictionary for all 20 standard amino acids plus some amino and carboxyl endgroups. The main difference between this dictionary and the previous one[1] is that the structure of a given amino acid residue is given here in terms of internal coordinates, bond lengths, bond angles, and dihedral angles, instead of Cartesian coordinates. This has several advantages: first, any updating of the dictionary or creation of nonstandard residues can be done easily, because published structural data are always presented in internal coordinates. It should be noted that, in the previous ECEPP/3 dictionary, structural information was stored as Cartesian coordinates with a limited number of digits. Using them to regenerate the structure of the molecule introduces unnecessary numerical errors in the bond angles compared with the original values published in the literature. These errors are, of course, small ($\pm 0.03°$). and are less than the experimental uncertainties associated with these structural parameters. These small deviations in bond angles and dihedral angles introduce small errors in the ECEPP/3 energies as well. For this reason, slight differences in energy values can be observed between this program and the previous ECEPP/3 package. These differences range between 0.05 and 0.2 kcal/mol for the minimum energy of all 20 naturally occurring amino acid residues with various terminal groups. The second advantage of this new dictionary is that it makes programming much easier, especially for defining the flexibility of the polypeptide chain and for specifying 1–4 and 1–5 nonbonded interactions.

An example of this dictionary is given for the aspartic acid residue (Table I). Columns b–f give the information associated with the atom connectivities. Columns g–j are related to the variable dihedral angles of the residue. The dihedral angles, ω, can be treated as fixed or variable; in columns g and h, dihedral angle ω is denoted as the 0th angle. Column i tells whether the atom belongs to the backbone or the side chain. Column j gives the type of the preceding dihedral angle; that is, ϕ, ψ, ω, or zero if the bond before the atom does not involve a variable dihedral angle. Columns k and l are the torsional potential type and Lennard–Jones atom type, respectively. The following three columns give the geometry of the residue in terms of the internal coordinates r, τ, and γ for bond length, bond angle, and dihedral angle, respectively. In the dictionary, the geometry of the residues is defined with all the variable dihedral angles equal to zero. These internal coordinates are used to calculate the Cartesian coordinates of the atom associated with any given line in the Table I. The last column is the partial charge of the atom. The same information is given in the computer program for all standard residues and different kinds of N- and C-terminal groups. This presentation of the dictionary was proposed several years ago by Voll (personal communication, 1989) based on the Robson/Platt dictionary.[7]

The AMBER dictionary is similar to the ECEPP/3 dictionary. The differences reside in the seven columns (a, k, l, m, n, o, p) of Table I. The atom name and atom types to be used with AMBER are taken from Table I of ref. 6. At the present stage, the dictionary is constructed manually for drug molecules. The basic principles for dictionary construction are as follows: the first atom is defined to be the pivot atom of the molecule; that is, the origin of the local frame to which rigid body rotation and translation is applied. It should correspond to an atom close to the geometric center of the molecule. The atomic numbering scheme is the same as in ECEPP. Partial atomic charges can be taken from the AMBER dictionary or from different *ab initio* calculations consistent with those used to obtain AMBER charges.

MOLECULAR SYSTEM

A polypeptide chain can be defined as completely flexible or can be composed of successive flexible and rigid segments. Each flexible segment is attached at both extremities to its adjacent rigid segments of the chain whose internal conformations do not change during the simulation. The rigid part of the protein is the set of rigid seg-

TABLE I.
Dictionary Information for the Aspartic Acid Residue.

ASP												Aspartic acid			
a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p
N	1	0	3	2	0	−1	0	1	31	0	14	1.325	115.0	0.0	−4.59
HN	2	1	0	0	0	0	0	3	0	0	2	1.000	124.0	−180.0	2.27
CA	3	1	6	5	4	0	1	1	11	0	9	1.453	121.0	0.00	0.82
HA	4	3	0	0	0	1	0	3	0	0	1	1.090	105.2	121.0	0.26
CB	5	3	9	8	7	1	2	2	41	2	6	1.530	111.1	−119.1	−1.29
1HB	7	5	0	0	0	2	0	4	0	0	1	1.090	108.6	122.0	0.78
2HB	8	5	0	0	0	2	0	4	0	0	1	1.090	108.6	−122.0	0.78
CG	9	5	11	10	0	2	3	2	42	1	7	1.530	115.0	0.0	6.44
OD1	10	9	0	0	0	3	0	2	0	0	17	1.240	121.0	0.0	−4.64
OD2	11	9	12	0	0	3	4	2	43	8	18	1.290	115.0	180.0	−4.51
HD2	12	11	0	0	0	4	0	4	0	0	1	1.000	110.0	0.0	2.83
C	6	3	14	13	0	1	5	1	21	0	7	1.530	109.3	0.0	5.80
O	13	6	0	0	0	5	0	1	0	0	17	1.230	120.5	−180.0	−4.95

Columns: a: Name of the current atom *i*. b: Atom number of the current atom *i*. c: Atom number of its father (preceding atom). d: Atom number of the first atom following the current atom *i* (son 1); atom 14 is the N atom of the next residue. e: Atom number of the second atom following the current atom *i* (son 2). f: Atom number of third atom following the current atom *i* (son 3). g: Number of the variable dihedral angle that will move current atom *i*; by definition, the 0th dihedral angle corresponds to the dihedral angle ω, −1 is the dihedral angle ψ of the preceding residue, 1 corresponds to the dihedral angle ϕ of the current residue (here ASP), and 2, 3, 4 . . . correspond to the side-chain dihedral angles χs. The last dihedral angle, in this case 5, corresponds to the dihedral angle ψ of the current residue. h: Number of the variable dihedral angle preceeding atom *i*. A value of zero is assigned for dihedral angles that are not variable or for the dihedral angle ω that can be defined as fixed or variable. 2, 3, 4 . . . are defined in g. i: 1 = backbone heavy atom; 2 = side-chain heavy atom; 3 = backbone hydrogen atom; 4 = side-chain hydrogen atom. j: Type of the dihedral angle preceding atom *i*: $\phi = 11$, $\psi = 21$, $\omega = 31$, χs = 41, 42, 43 . . . . k: Type of torsional potential. In ECEPP / 3, each number corresponds to one of the 11 types of torsional potentials for all standard amino acids. They vary in terms of the value of the energy barrier and the symmetry and the sign of the trigonometric function. The form of these functions and their corresponding codes, from 1 to 11, are the same as in the earlier version of the ECEPP / 3 packages.[2,3,5] l: Atom type for the Lennard–Jones interaction. This is the same code as in the earlier version of ECEPP / 3. m: Bond length *r* (Å) between atom *i* and its father (column c). n: Bond angle τ (degrees) between atom *i*, its father, and its grandfather. o: Dihedral angle γ (degrees) between atom *i*, its father, its grandfather and its great grandfather. p: Partial charge of atom *i* multiplied by $(332 / D)$,[1/2] where *D* is the dielectric constant ($D = 2$); these are ECEPP / 3 charges.[5]

ments. They usually represent the observed X-ray conformation. The number and the length of these segments are purely arbitrary, and depend on the type of problem under study. An example of the definition of such segments is given for a docking type of problem in Table II. The receptor is human α-thrombin and the ligand is a tripeptide NH_2–D-Phe–Pro–Arg–COOH (FPR). This tripeptide is the analog of NH_2–D-Phe–Pro–Arg–chloromethyl ketone (PPACK), an antagonist of human α-thrombin.[18,19] The residues of the active site and those of the ligand are taken as flexible, whereas residues located further away from the active site are considered as rigid (i.e., they do not move during the simulation). For a rigid docking calculation, the ligand would be rigid but its center of mass (CM) would be movable. In this example, the CMs of the A and B chains of thrombin are fixed, and the CM of the ligand is movable.

For each segment, FLEXIBLE means flexible side chain only (SIDE), flexible backbone only (BACK), or both (ALL). Also, the energy of each segment can be turned ON or turned OFF during the simulation. This feature is particularly important in the docking procedure for two reasons: first, it simplifies the topology of the active site by removing some surrounding loops that prevent the entrance of the ligand. Second, it allows one to remove from the interaction list the atoms that are far from the active site. The access of the substrate to the active site is made possible by turning off the energies of the surrounding loops that lie at the edge of the active site, which usually act as a lid. Thus, it is the set of residues that form the bottom of the active site that remain. Knowledge of the location of the active site is of course necessary, because the residues having energies that are ''turned off'' are determined visually on a graphics screen. With

TABLE II.
Example of Definition of Flexibility for Complex of Thrombin with NH_2 – Phe – Pro – Arg – COOH (FPR).[a]

Segment number	Residue number in the sequence	Chymotrypsin numbering[b]	Flexibility	Type of flexibility	Energy turned off or turned on
FPR:					
1.	1 –3	1 –3	FLEXIBLE	ALL	ON
Thrombin α chain:					
1.	1 –36	1 –15	RIGID		ON
Thrombin β chain:					
1.	1 –19	16 –34	RIGID		OFF
2.	20 –29	35 –43	RIGID		ON
3.	30 –40	44 –54	RIGID		OFF
4.	41 –43	55 –57	FLEXIBLE	SIDE	ON
5.	44 –46	58 –60	RIGID		ON
6.	47 –54	60A –60H	FLEXIBLE	ALL	ON
7.	55 –89	60I –93	RIGID		OFF
8.	90 –99	94 –102	RIGID		ON
9.	100 –140	103 –140	RIGID		OFF
10.	141 –156	141 –151	FLEXIBLE	ALL	ON
11.	157 –175	152 –170	RIGID		OFF
12.	176 –180	171 –175	RIGID		ON
13.	181 –197	176 –187	RIGID		OFF
14.	198 –201	188 –191	RIGID		ON
15.	202 –206	192 –196	FLEXIBLE	SIDE	ON
16.	207 –224	197 –212	RIGID		OFF
17.	225 –230	213 –219	FLEXIBLE	SIDE	ON
18.	231 –241	220 –229	RIGID		ON
19.	242 –259	230 –247	RIGID		OFF

[a] This is just an example for illustration only. It is not used anywhere.
[b] Chymotrypsin numbering rather than sequence numbering is usually used for serine proteases such as thrombin.

this simplification, it is even possible to dock molecules into an active site that is buried inside a deep narrow cleft of the protein.

GENERATION OF STRUCTURE

The generation of a protein structure consists of three steps: the first is the generation of all flexible segments in a reference system using a given set of dihedral angles. At this point, all ''origin atoms'' of each flexible segment are located at the same origin of a reference frame (0, 0, 0). These ''origin atoms'' can be the nitrogen atom of the first residue of the flexible segment, or the pivot atom (i.e., a C^{α}) in the middle of the chain (see definition in what follows) if this flexible segment corresponds to a full molecule (e.g., the ligand). At this stage, the amino acid residues of the flexible segments have been connected and the flexible dihedral angles defined. The second step is the transfer of these newly generated segments from the reference frame to their correct location in the protein. The final step is the alteration of the overall position and orientation of the ligand using rigid body external variables. The procedure to generate the polypeptide chain varies according to the location of the flexible segment in the protein. The first is a ''forward'' generation procedure, which means that the segment is built from the N- to the C-terminal residue. This type of generation is applied for flexible segments located between rigid segments or at the C-terminus of the polypeptide chain. The second type of generation procedure is ''backward generation,'' which means that the chain is generated from the C- to the N-terminal residue. This procedure is applied only for the first segment of the polypeptide chain in cases in which this segment has been defined as flexible. This segment is,

therefore, attached to a rigid segment forward in the sequence. Finally, the last option is the forward/backward generation. This is applied when the whole molecule, protein or ligand, is flexible. The generation is carried out forward from the pivot atom to the C-terminal group and backward from the pivot atom to the N-terminal group. The pivot atom is chosen to be the C^{α} of the residue that is in the middle of the sequence. The side chain of the residue bearing the pivot atom is generated during the forward generation procedure. This forward/backward generation procedure has previously been shown[20] to improve the convergence of the energy minimization procedure significantly when several polypeptide chains are considered. A comparison test for a single chain when generating forward only and backward/forward will be presented in the Results section.

The external variables of the ligand correspond to the translation vector between the pivot atom and the origin of the reference system $(0, 0, 0)$, and the four quaternion parameters (see later) that give the relative orientation of the local frame compared with the reference frame. The local frame centered on the pivot atom at $(0', 0', 0')$ is built by applying the Gramm–Schmidt orthonormalization procedure[21] to produce three mutually perpendicular unit vectors from the two vectors represented by the C^{α}—H^{α} and C^{α}—C' bonds. The first unit vector is colinear with the first bond, the second unit vector is in the plane spanned by the two bonds, and the third unit vector is the crossproduct between the two previously generated unit vectors.

QUATERNION PARAMETERIZATION

Quaternions were discovered by Hamilton in 1843, and independently by Rodrigues.[22] Their motivation was to find some kind of numbers whose product would correspond to rotation in three-dimensional space, just as the product of complex numbers corresponds to the rotation of vectors in a plane.

According to Euler's theorem, any sequence of rotations with one point fixed is equivalent to a single rotation about a given axis. If we designate the unit vector on that axis as $\mathbf{a}(a_1, a_2, a_3)$ and the angle of rotation by ϕ, then the quaternion four-vector $\mathbf{Q}(q_1, q_2, q_3, q_4)$ is defined as:

$$\mathbf{Q}(q_1, q_2, q_3, q_4) \equiv \mathbf{Q}(\mathbf{a}, \varphi) = (\sin[\varphi/2]\mathbf{a}, \cos[\varphi/2]) \quad (1)$$

The quaternion has a norm of 1; that is:

$$\sum_i q_i^2 = 1.$$

The q_is are related to the Euler angles by the following equations:

$$\begin{aligned} q_1 &= \sin\frac{\theta}{2}\cos\frac{\phi-\psi}{2} & q_3 &= \cos\frac{\theta}{2}\sin\frac{\phi+\psi}{2} \\ q_2 &= \sin\frac{\theta}{2}\sin\frac{\phi-\psi}{2} & q_4 &= \cos\frac{\theta}{2}\cos\frac{\phi+\psi}{2} \end{aligned} \quad (2)$$

where the definition of the three Euler angles (ϕ, θ, ψ) is that of Goldstein.[23] The rotation matrix, D, is given in terms of the four parameters (q_1, q_2, q_3, q_4) by[24, 25]:

$$D = \frac{1}{p}\begin{bmatrix} q_1^2 - q_2^2 - q_3^2 + q_4^2 & 2(q_1q_2 - q_3q_4) & 2(q_1q_3 + q_2q_4) \\ 2(q_1q_2 + q_3q_4) & -q_1^2 + q_2^2 - q_3^2 + q_4^2 & 2(q_2q_3 - q_1q_4) \\ 2(q_1q_3 - q_2q_4) & q(q_2q_3 + q_1q_4) & -q_1^2 - q_2^2 + q_3^2 + q_4^2 \end{bmatrix} \quad (3)$$

where:

$$p = q_1^2 + q_2^2 + q_3^2 + q_4^2 \quad (4)$$

is the norm of $\mathbf{Q}$ that is equal to 1. There are two ways to minimize the energy with quaternions as variables. The first is to let the four quaternion parameters, q_1 to q_4, vary in such a way that they always stay on the three-dimensional sphere manifold, S_3; that is, by determining q_4 from the constraint eq. (4). The disadvantage of this protocol is that it requires a constrained minimizer. The second and simpler approach is to consider the four quaternion parameters as independent variables evolving in the four-dimensional real field R^4. The rotation matrix D is scaled by the norm of the quaternion vector $\mathbf{Q}$ to keep Det $D = 1$. This scaling would be sufficient if random rotation matrices were needed, with no correlation between successive matrices. On the contrary, if we wanted to simulate the dynamic trajectory of the molecule

in space, as in molecular dynamics for example, it would be necessary to keep the quaternions as close as possible to the manifold S^3. This is achieved by adding a harmonic constraint term in the energy function, which keeps the norm close to 1 [see eq. (7)].

$D\mathbf{r_o}$ defines an active rotation of the vector $\mathbf{r_o}$ centered at the origin of the local frame. ''Active'' means that the vector $\mathbf{r_o}$ is rotated and the axes of the local frame are fixed. Goldstein[23] defined a ''passive'' rotation as one in which the vector $\mathbf{r_o}$ is fixed and the axes of the local frame are rotated. Goldstein's corresponding rotation matrix is equal to D^{-1}.

In a standard MC procedure, the generation of random unit-quaternion vectors is realized using the three following steps[26,27]:

1. Generate one pair of random numbers r_1, r_2 independently and uniformly distributed in the interval $(-1, 1)$ until:

$$S_1 = r_1^2 + r_2^2 < 1$$

2. Do the same for pairs r_3, r_4 until:

$$S_2 = r_3^2 + r_4^2 < 1$$

3. Form the random unit four-vector:

$$Q = \left\{ r_1, r_2, r_3\sqrt{(1-S_1)/S_2}\,, r_4\sqrt{(1-S_1)/S_2} \right\} \quad (5)$$

and use it to prepare a new orientation.

Beside the elegance of quaternion algebra, there are several advantages to working with quaternions instead of the more familiar Euler angle parameters: The most important advantage is that the orientation of a rigid body relative to a reference system is determined uniquely by the quaternion $\mathbf{Q}(q_1, q_2, q_3, q_4)$ except for a change of sign. In the Euler angle representation, there is an infinite number of triplet angles (ϕ, θ, ψ) that define the same orientation of the rigid body, when the second Euler angle, θ, is equal to zero or π. This means that different sets of Euler angles produce the same energy. In the special case in which the orientation of the molecule corresponds to $\theta = 0$ or π and to a local energy minimum or a saddle point of the energy hypersurface, a local energy minimizer encounters weak or degenerate minima and, in practice, would take an extremely long time to converge, as we will show in the Results section. Another advantage of quaternions over Euler angle parameters is that computers work faster with algebraic functions than with trigonometric functions.

ENERGY FUNCTION

A general description of the molecular system is presented in Figure 1. The different energy terms associated with this model are as follows:

$$\begin{aligned} E = {} & \lambda_{\text{intra}} E_{\text{intra}} + \lambda_{\text{inter}} E_{\text{inter}} + \lambda_{\text{tors}} E_{\text{tors}} \\ & + \lambda_{\text{bridge}} E_{\text{bridge}} + \lambda_{\text{closure}} E_{\text{closure}} \\ & + \lambda_{\text{quat}} E_{\text{quat}} + \lambda_{\text{dist}} E_{\text{dist}} + \lambda_{\text{x-ray}} E_{\text{x-ray}} \\ & + \lambda_{\text{NOE}} E_{\text{NOE}} \end{aligned} \quad (6)$$

The λs are the weights of the different terms of the potential energy. They offer much variability for simulated annealing calculations. In such calculations, changing the temperature consists of changing the λs with the condition that they are all equal. If the weights were to differ, this would be

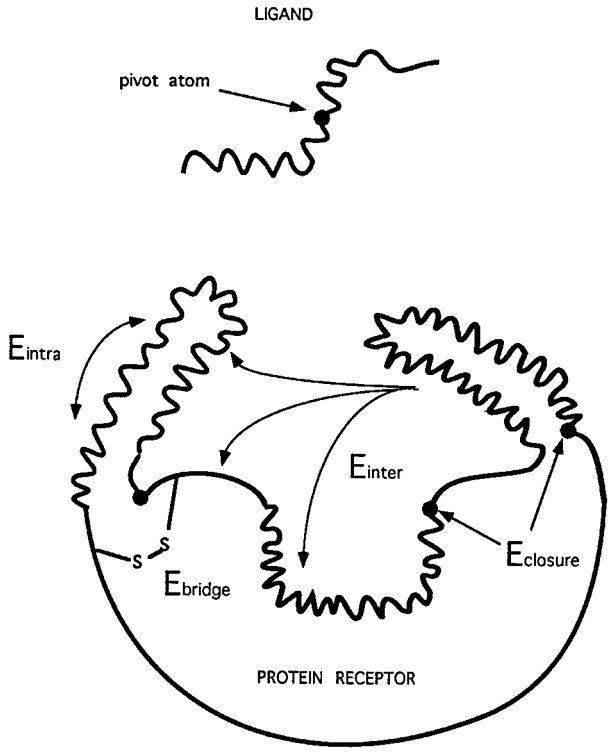

FIGURE 1. Schematic diagram of a molecular system. The uniform lines correspond to rigid segments, whereas the wavy lines represent flexible segments. In this example, the two molecules represent a ligand and its receptor.

equivalent to having a different temperature for each energy term. The standard ECEPP/3 energy[1–4] that is, nonbonded, electrostatic, hydrogen bonded, plus the hydration free energy,[28] are incorporated into the first two terms. The ''intra'' energy, E_{intra}, stands for the interaction energy between atoms of a given segment (only if flexible) and ''inter'' energy, E_{inter}, refers to the interaction energy between different segments. This energy is taken into account only if the relative atomic positions of these segments change during the simulation (if at least one of them is flexible or belongs to a protein for which the CM is movable). Two other terms are part of ECEPP/3: E_{tors}, the intrinsic torsional potential associated with certain side-chain dihedral angles and E_{bridge}, the constraint energy associated with the disulfide bridges.[1–4] The harmonic constraint potential E_{closure} plays the role of loop closure for joining a flexible segment onto a rigid segment. It is applied to the last three backbone atoms C^{α}, C′, and O of the flexible segment. E_{quat} is the harmonic penalty function to constrain the norm of the quaternions to unity[24,25]:

$$E_{\text{quat}} = \frac{1}{2}(1 - p)^2 \tag{7}$$

The last three terms are distance penalty functions. E_{dist} is used in our docking experiments to maintain the CM of the ligand within a given region around the active site. In this study, we use a quadratic function of the form:

$$\begin{aligned} E_{\text{dist}} &= \sum_i \left[d_i - d_i^{\text{low}} \right]^2 && \text{if } d_i < d_i^{\text{low}} \\ E_{\text{dist}} &= 0 && \text{if } d_i^{\text{low}} < d_i < d_i^{\text{up}} \\ E_{\text{dist}} &= \sum_i \left[d_i - d_i^{\text{up}} \right]^2 && \text{if } d_i > d_i^{\text{up}} \end{aligned} \tag{8}$$

where d_i^{low} and d_i^{up} are the lower and upper limits of the distance constraint. The term $E_{\text{x-ray}}$ is a harmonic distance constraint of similar form:

$$E_{\text{x-ray}} = \sum_i [\, r_i - r_i^{\text{x-ray}} \,]^2 \tag{9}$$

This term is used mainly to regularize the PDB structure (see Regularization section) or during the first stage of a minimization procedure to prevent the protein atoms from moving too far from the positions given by the X-ray experiments. The last term, E_{NOE}, is used for incorporating NOE distance constraints during the simulation. Different types of functions can be adopted here. Usually, the function is chosen in such a way that E_{NOE} is zero when the distance between two atoms, i and j, are within the distance interval provided by the NMR experiments. When the distance is smaller than the lower limit, there is a quadratic constraint. When the distance is greater than the upper limit, there is a quadratic constraint until a certain limit δ: when the violation is larger than δ, the quadratic parabola is replaced by a linear asymptotic branch. This term in the potential prevents the system from being trapped in high-energy minima due to large NOE violations, especially at the beginnning of the simulations. See ref. 15 for details.

HYDRATION FREE ENERGY

A new hydration volume model has recently been developed in our laboratory.[28] The purpose of this model was to include the solvation free energy in a pairwise atomic potential in a form that can be treated by the diffusion equation method (DEM), a technique to search for the global energy minimum of peptides and proteins.[29] This solvation model is based on the approximation that the hydration free energy of a given atom is proportional to the solvent-exposed volume of the hydration shell. In the original work,[28] the volume of intersecting spheres was approximated by Gaussian functions because this form is easily transformed analytically by the Fourier–Poisson integral, a key element of the DEM. Because we are not smoothing the potential function by the DEM, the Gaussian approximation of the volume of intersecting spheres is not necessary. A saving in CPU time can thus be achieved by using the exact formula for intersecting spheres. This formula involves only rational functions of the interatomic distances and does not make use of exponential functions, which are computationally expensive. We found that the CPU time to estimate an exponential in double precision corresponds to 18 multiplication operations on an IBM-SP2 RX6000 computer. By transforming the four exponential functions for each atomic pairwise interaction back into a power function of the interatomic distance r_{ij}, our energy minimization procedure with total energy (ECEPP/3 + hydration) becomes only 1–5% more expensive than with the ECEPP/3 energy alone. Another reason that this additional expense is so small is that a cutoff distance ($r_2 < R_1 + R_2$) can be applied for all the atomic interactions to estimate the exact volume.

The formula used for the volume of intersection of two spheres with radii R_1 and R_2 separated by a distance r_{12} is a spline of three functions[28]:

$$V(r_{12}, R_1, R_2) = \frac{2\pi}{3}\left(R_1^3 + R_2^3 + \frac{r_{12}^3}{8}\right) - \frac{\pi r_{12}}{2}(R_1^2 + R_2^2) - \frac{\pi}{4r_{12}}(R_1^2 - R_2^2)^2 \qquad (10)$$

where $R_1 - R_2 < r_{12} < R_1 + R_2$, $V(r_{12}, R_1, R_2) = 0$ when r_{12} is greater than the separation distance $R_1 + R_2$ and $V(r_{12}, R_1, R_2) = (4\pi/3)R_2^3$ when $r_{12} < R_1 - R_2$. This spline function is continuous and differentiable at $r_{12} = R_1 - R_2$ and $r_{12} = R_1 + R_2$. In this formulation, we assume that $R_1 > R_2$. The hydration free energy parameters for each atom type have been determined by least-squares fitting of some experimental free energy of transfer values with the theoretical free energy values calculated with the Gaussian approximation for the intersecting volume.[28] When the exact formula as expressed in eq. (10) is used to calculate the hydration free energy, it would be necessary to recalibrate the hydration free energy parameters. However, the very good fit (correlation coefficient $\sim$ 0.9) obtained in the original work between the exact and Gaussian-approximated volume (Fig. 4 of ref. 28) suggests that the error made by keeping the same set of hydration free energy parameters for calculating the hydration free energy of a molecule or peptide would be similar to the average deviation between the experimental and fitted hydration free energies obtained in the previous work. This average deviation, obtained for the 140 compounds used for the calibration, was 0.35 kcal/mol, with a standard deviation of 0.53 kcal/mol (J. D. Ausgpurger, personal communication). A test on the complex FPR–human α-thrombin showed that the differences ranged from 0.3 to 1.0 kcal/mol between the hydration energies calculated with the exact volume formula and the Gaussian approximation. We thus make the assumption that the general feature of the hydration potential is conserved between these two expressions for the volume of intersecting spheres.

GRADIENT AND HESSIAN CALCULATION

The analytical gradient of the energy with respect to the dihedral angles was computed with the method proposed by Levitt.[30] The second derivative matrix (Hessian) of the energy with respect to the internal and external variables has been obtained from the numerical derivatives of the analytical gradient.

REGULARIZATION

The bond lengths and bond angles observed in the X-ray structure of the protein are generally slightly different from the standard values defined by ECEPP/3 geometry. Regularization consists of finding the set of dihedral angles that produces the best fit between the ECEPP/3 and X-ray structures. The regularization procedure is composed of three steps: (i) calculating the dihedral angles from the X-ray structure; (ii) generating an ECEPP/3 conformation using this set of dihedral angles; and (iii) minimizing the cost function [eq. (9)] applied to all the heavy atoms of the molecule. The minimization is carried out with the minimizer SUMSL (secant unconstrained minimization solver) developed by Gay.[31]

MONTE CARLO SIMULATION

Scaled collective variables (*SCV*) *Monte Carlo*. The Monte Carlo (MC) algorithm used in this program serves several different purposes. First, it is a method of global optimization; that is, finding the lowest value of the energy function [eq. (6)]. Second, it can also be used as a sampling method to calculate free energy. Because of the large anisotropy of the energy surface due to the high density of protein atoms, unbiased Monte Carlo methods, especially in Cartesian coordinate space, are known to be rather inefficient for conformational sampling (see discussion in refs. 9, 32, and 33). To simulate the concerted motion of the atoms inside proteins, we use the MC algorithm developed by Noguti and Gō.[10] The trial conformations of the protein are generated by using information about the topology of the energy hypersurface. At a particular point in the conformational space, we approximate the energy surface by a multidimensional parabola and determine the principal axes components of this hyperparabola. This is obtained by calculating the eigenvectors of the second derivative matrix of the energy with respect to the internal and external variables. By scaling each variable increment by a quantity proportional to the inverse of the corresponding eigenvalue, the perturbation vector will be larger in the directions of small curvature (''soft'' directions) and smaller

in the directions where the energy increases rapidly (''hard'' directions). The perturbation vector, $\Delta\Theta$, with elements called ''scaled collective variables,'' is defined as:

$$\Delta\Theta = s \sum_{k=1}^{N} r_k \frac{1}{\sqrt{|\sigma_k|}} \mu_k \tag{11}$$

It is obtained by a linear combination of the eigenvectors μ_k, where the coefficients are the corresponding eigenvalues σ_k. The second derivative matrix can be calculated at an arbitrary point on the potential energy surface, different from the global or local minimum leading to negative eigenvalues. Hence, the absolute values are taken instead.[9] The length of the vector is scaled by a step size, s. In the eigenvector space, the sampling is isotropic. Therefore, random numbers, r_k, are generated with a uniform distribution in the interval $[-0.5; 0.5]$. The summation is carried out over N, the total number of variables (internal plus external). The perturbation vector, $\Delta\Theta$, is added to the vector of variables, Θ, defining the current conformation of the molecular system. The energy of the trial conformation is accepted according to the Metropolis criterion[34]; that is, with the probability $\min[1, \exp(-\Delta E/k_B T)]$ where ΔE is the energy difference between the trial and the current conformation, k_B is the Boltzmann constant, and T is the absolute temperature.

When the system is large, the calculation of the Hessian can be computationally expensive. Therefore, it is updated every ten or hundred MC steps according to the size of the system. It should be noted that, within a given interval, microreversibility of the transition probabilities is maintained, but not when the Hessian is recalculated.[9] Therefore, the convergence toward the Boltzmann distribution is no longer certified. However, because we are using the Monte Carlo technique as a global optimization method, we are not interested in calculating statistical average quantities. Hence, this problem is not relevant in our study. A way to circumvent this problem partially for the calculation of thermodynamic quantities has been discussed by Gibrat et al.[9] and more recently by Leontidis and Suter.[35] The SCV Monte Carlo procedure was used in refs. 8–10, 13, and 15–17.

Quaternion generation in standard Monte Carlo algorithm. In the standard MC algorithm,[34] all variables are perturbed randomly without any bias. For dihedral angles and translation variables, the perturbation vector is added to the current variables. For rotation variables, the new orientation $(\mathbf{Q}', Q_4')$ of the molecule is determined by multiplying the current quaternion $(\mathbf{Q}, Q_4)$ by the quaternion increment $(\mathbf{q}, q_4)$ according to the following equation[36]:

$$\begin{aligned}(\mathbf{Q}', Q_4') &= (\mathbf{Q}, Q_4) \times (q, q_4) \\ &= (Q_4 q_4 - \mathbf{Q} \cdot Q, Q_4 q + q_4 Q + Q \times q)\end{aligned} \tag{12}$$

where $\mathbf{Q}$ and $\mathbf{q}$ are the vector parts of the quaternions and Q_4 and q_4 are the scalar parts.

For standard MC simulations, a step size is assigned to each type of variable (dihedral angles, translations, and rotations). For the dihedral angles and translation vectors, the corresponding variables are multiplied by the step sizes, s_{tors} and s_{trans}. For the rotation variables, we cannot simply multiply the quaternion parameters q_i by a scalar s, because the norm of the quaternion, q, must be preserved. However, the definition of the step size, s_{quat}, for quaternions should be such that a small step size should correspond to a small perturbation of the current orientation of the molecule and a large step size should produce a completely random orientation of the molecule uncorrelated with the previous orientation. The procedure to generate the quaternion increment $(\mathbf{q}, q_4)$ is thus the following:

- Step 1. Generate three random numbers, r_1, r_2, r_3, in the interval $[-1, 1]$ using a Gaussian distribution in which the width of the distribution is controlled by the step size. We repeat that process until:

  $$S = r_1^2 + r_2^2 + r_3^2 < 1$$

  The Gaussian probability density function is:

  $$\Psi(x, \lambda) = \frac{1}{\sqrt{2\pi\lambda^2}} \exp\left(-\frac{x^2}{2\lambda^2}\right) \tag{13}$$

  where $\lambda = s_{quat}$ is the step size for quaternions and represents the standard deviation of the distribution, and x stands for r_1, r_2, r_3 successively.
- Step 2. Form the quaternion $q = \{r_1, r_2, r_3, r_4\}$ with:

  $$r_4 = \sqrt{1 - S}$$

- Step 3. The new orientation $(\mathbf{Q}', Q_4')$ is obtained by multiplying the quaternion incre-

ment $(\mathbf{q}, q_4)$ with the previous orientation $(\mathbf{Q}, Q_4)$ using eq. (13).

The larger the step size s_{quat}, the broader the Gaussian function and more uniform the probability distribution. On the contrary, a small step size would correspond to a narrow Gaussian distribution, which will produce three numbers, r_1, r_2, r_3 very close to zero, producing an r_4 close to 1. The corresponding rotation matrix will therefore be close to the unit matrix.

Monte Carlo with minimization (*MCM*). The MCM algorithm used in our study is the one presented several years ago by Li and Scheraga.[11,12] The algorithm consists of carrying out a random MC move followed by a local energy minimization. In their procedure, the MC moves were unbiased. In our case, they are generated according to the Noguti–Gō algorithm. In the following, MCM really stands for SCV-MC with energy minimization (SCV-MCM). The minimization aims at removing the clashes between the atoms of the ligand and the protein. Hence, the minimization might not need to be carried out until convergence. In practice, the minimization is stopped after a certain number of iterations predetermined by the user.

SIMULATED ANNEALING

In this package, simulated annealing (SA)[37] consists of a series of homogeneous Markov chains computed at a given temperature.[8,9] The temperature is controlled by changing the weights, λ, associated with each energy term. In the case in which all the λs are the same, the effective temperature of the simulation is $T_{eff} = T/\lambda$, where $T =$ 1000 K. Decreasing the temperature is thus similar to increasing the weight factors, λ, and keeping parameter T constant during the MC run. The possibility of choosing different weights for the components of the energy function [eq. (6)] provides additional alternatives for the user than simply changing temperature parameter T. This approach can be pursued even further when applied to flexible docking, by assigning a small λ_{intra} (i.e., high T) for the ligand and $\lambda_{intra} = 1$ for the rest of the flexible segments; that is, proteins loops. Similarly, λ_{inter} can be set to less than 1.0 (i.e., high T) when interactions between the ligand and the receptor are involved, and $\lambda_{inter} = 1.0$ when considering the interactions between protein loops. This allows the ligand to escape easily from local minima while simulating the protein loops at normal temperature.

Results

FORWARD/BACKWARD CHAIN GENERATION

The forward/backward chain generation procedure has been shown to provide a major improvement in multichain simulations.[20] We found that this type of generation procedure is also of prime importance for docking a flexible ligand into its active site. To test the gain in efficiency, we made some comparisons with the standard generation procedure (forward from the *N*-terminal residue), used in the earlier version of the ECEPP/3 program. The test consists of superimposing FPR onto the X-ray structure. The superposition is done by minimizing the cost function given by eq. (9). At the beginning of the minimization, the tripeptide is in the extended conformation in a random orientation and located at a distance about 10 Å away from the target structure. Both the initial and the target structure have ECEPP/3 geometry. The minimum of the cost function is thus unique and equal to 0, within six digits of accuracy.

To test the efficiency of the minimization, we compared the convergence for both generation types, forward and forward/backward. The two curves of Figure 2 show the evolution of the logarithm of the rms gradient of the energy with the number of iterations during the energy minimization. A clear difference appears in terms of convergence. The forward/backward procedure reached the minimum after 131 iterations compared with 2121 for the forward simulation. The ratio of efficiency between the two minimizations is about 15, similar to the one reported previously.[20] This forward/backward type of generation is thus adopted in this package whenever a molecule is defined as completely flexible as is usually the case for the ligand in docking simulations.

QUATERNIONS VERSUS EULER ANGLES

The degeneracy of the energy when the second Euler angle, θ, is equal to zero or π creates singularities for certain quantities that are calculated during simulations. In molecular dynamics (MD), singularities appear in the time derivative of the angular momentum. Evans and Murad[38] showed that these singularities induce large numerical fluctuations in the total energy in simulations of polyatomic fluids where the energy is considered as constant (microcanonical ensemble). In molecular mechanics, singularities of the Hessian occur

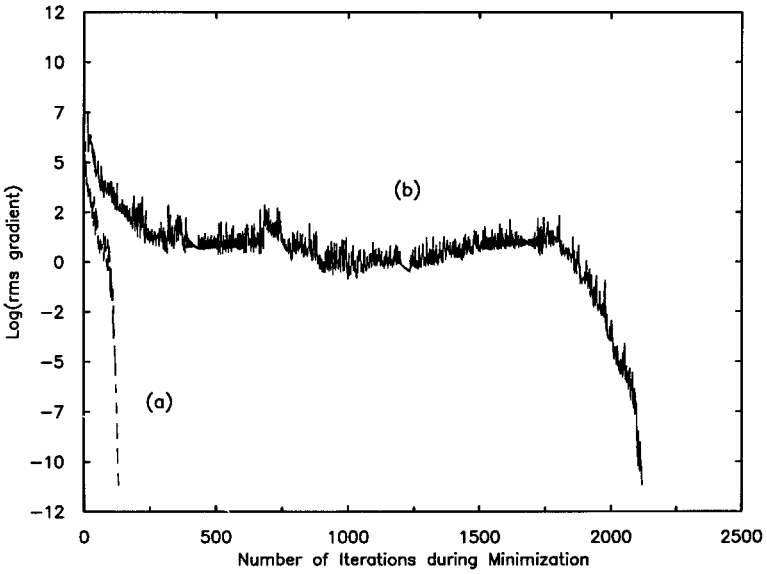

FIGURE 2. Comparison of the convergence of the rms gradient during energy minimization when the chain is generated forward / backward from the C^{α} of the middle residue (a) or forward from the first residue (b).

when $\theta = 0$ or π and all the components of the gradient are equal to zero. These singularities thus occur at any local minimum or saddle point, where $\theta = 0$ or π. At those points, the second partial derivative of the energy with respect to the second Euler angle θ and any other type of variable X is equal to zero. All the elements of the Hessian, $\delta^2 E/\delta\theta\delta X$, are equal to zero, because both terms, $\delta E/\delta\theta$ and $\delta E/\delta X$, are equal to zero. The first term is zero because $\theta = 0$ or π (in that case, a small perturbation, $\Delta\theta$, would not change the orientation of the molecule; therefore, the energy does not change and $\delta E/\delta\theta = 0$). The second term is zero because the molecule is at a local energy minimum or a saddle point. The Hessian is thus singular. If the minimizer does not involve the calculation of the Hessian matrix, there is still a convergence problem when the molecule is at a local energy minimum and when the second Euler angle $\theta = 0$ or π. It is important to note that there are no singularities at $\theta = 0$ when the energy is not a minimum. Therefore, starting the simulation at a point that is not a minimum, with a unit rotation matrix ($\theta = 0$), does not create any convergence problem contrary to what is sometimes stated in the literature; e.g., ref. 24).

The inconvenience of these weak minima becomes more apparent as the number of independent molecules in the system increases (e.g., simulations with explicit water molecules). Indeed, the larger the number of molecules, the higher the probability that one of the molecules will have a θ close to 0 or π when the system is at an energy minimum (global or local). In MD simulations of liquids, the presence of such singularities is thus an important issue. In docking simulations, this problem might remain unnoticed because the number of minima with θ close to 0 is probably a small fraction of all the local minima and saddle points of the energy hypersurface. For docking, the simulation will fail mainly if the optimized structure of the ligand (the one of lowest energy) has a ''bad'' orientation (θ close to 0 or π). In this section, we assess the bad convergence properties of the energy minimization due to these singularities. This illustrates the advantage of quaternions over the Euler-angle representation.

The tests consist of docking the tripeptide FPR onto α-thrombin by minimizing an energy function which is the sum of the X-ray constraint plus the ECEPP/3 energy. We use the X-ray constraint to force the simulation to finish very close to the X-ray structure, which is very close to the global minimum, because the harmonic constraint energy term makes the major contribution. Therefore, we can compare the convergence properties for different trajectories that all finish at the same point. At the beginning of the docking simulation, the ligand is located about 10 Å outside the active site in the extended conformation in a random orientation. The ligand is thus flexible and the atoms of the receptor are fixed at their ECEPP/3 energy-minimized X-ray coordinates. The weights λ_{intra}, λ_{inter}, and λ_{tors} are equal to 1, whereas $\lambda_{\text{x-ray}}$ is

equal to 1000. For simulations with quaternions, λ_{quat} is equal to 5000. A more detailed description of the system and docking simulation can be found in ref. 13.

Two types of docking experiments have been carried out: in the first, the target structure of the ligand is the X-ray structure with a θ value for the second Euler angle (117°) different from zero. In the second experiment, the X-ray structure of the ligand has been rotated in such a way that the θ value is equal to zero. Ten docking energy minimizations have been carried out in both cases for which a comparison between quaternions and Euler angles has been made.

For the first series of docking simulations, the Euler angles and quaternions perform similarly because the final lowest energy conformation corresponds to a $\theta \neq 0$. The energy minimum is reached after an average number of iterations of 341 ± 22 with quaternions and 341 ± 41 with Euler angles. For these minimizations, the angle θ crosses the zero value without affecting the convergence of the minimization. For example, by choosing the initial conformation as one for which $\theta = 0$ does not create any problem of convergence, and the global minima is achieved after 352 iterations. Problems arise when $\theta = 0$ coincides with the global minimum as seen in what follows.

In the second series of simulations in which the target structure corresponds to $\theta = 0$, the two representations differ significantly: with Euler angles, none of the ten docking minimizations led to a minimum close to the global minimum of the energy. The minimization was stopped because the relative convergence had been achieved; that is, the energy decrease was less than 10^{-8} kcal/mol between two successive iterations. The energy of the final structures ranged between $0.4 \times 10^{+4}$ and $0.2 \times 10^{+5}$ kcal/mol (with $\lambda_{x\text{-}ray} = 1000$), and the root-mean-square-deviation (rmsd), between the calculated and the X-ray structure, for all heavy atoms ranged between 0.3 and 0.8 Å. Using quaternions, all the simulations except one led to a structure which was close to the global minimum. The energies ranged between -67.6 and -71.2 kcal/mol and the rmsd values were less than 0.001 Å. For the one simulation that led to a higher energy, relative convergence was achieved. This corresponds to a local minimum of the potential energy surface. The minimizer did not stop because of the presence of weak minima as was the case with the Euler-angle representation.

In conclusion, the quaternion representation avoids the convergence problem encountered with Euler angles. The use of quaternions is highly recommended when the number of molecules in the system becomes large, for example when explicit water molecules are present.

Discussion

The Results section shows how technical details, such as the way to describe the variables of the molecule for example, determine the efficiency of a molecular simulation. The applications of the program PRODOCK that have already been made concern the docking of a small ligand onto a target receptor,[13,16] a study of protein–protein interactions (work in progress), the determination of the structure of a fibrinogen-like peptide bound to human α-thrombin using NMR data,[15] and an analysis of the global motion of a large single polypeptide chain (work in progress). These applications are discussed in what follows.

SIMULATION OF LARGE POLYPEPTIDE CHAIN

Molecular simulation of a very large protein is very demanding computationally because of the large number of atomic interactions to be considered. There are different approaches to decrease the computational cost. One of the most prominent is the multipole cell expansion method.[39] For large distances, it considers only the interactions between multipole moments that have been estimated for each cell into which the protein system is divided. Using a cell-hierarchical representation, the calculation of the electrostatic energy scales linearly with the number of atoms instead of as N^2. The second approach, that was used here for a large protein, is to freeze the dihedral angles of certain domains or structural elements of the protein. For example, helices might be frozen, and variable dihedral angles assigned only to the turns and loops that connect these structural elements. This is done simply by changing appropriate elements of the arrays associated with columns g and h; that is, ''switching on'' these dihedral angles. The list of interactions would be modified accordingly in such a way that no interactions need to be calculated within the fixed structural elements. This approximation decreases the number of variables, making standard MC or MD simulations of large polypeptide chains possible.

DOCKING OF SMALL LIGANDS OR PROTEINS ONTO TARGET PROTEIN RECEPTORS

Docking of a small ligand or protein was made efficient by combining different techniques, such as the grid-based energy calculation using Bezier splines,[14] the SCV Monte Carlo method, the growing MC procedure[15] and the multiscale annealing approach that uses an independent annealing schedule for each weight of the potential energy function [eq. (6)].

With the Bezier splines, it is possible to carry out local energy minimization during the docking simulation. This has been shown to be crucial for distinguishing between native and nonnative low-energy structures. Any conformational search procedures, such as MC or genetics algorithm (GA) for example, might generate structures of the ligand that are very close to the native X-ray structure but with very high energy because of some atomic clashes. Without energy minimization, those conformations would be rejected from the list of acceptable candidates, whereas the ones that are accepted might be only remotely related to the X-ray structure. The Bezier spline technique enables local energy minimization of the molecules during the conformational sampling to be carried out in a reasonable amount of time. This technique speeds up the MCM simulation by one or two orders of magnitude depending of the size of the rigid part of the receptor. The docking of two proteins, both being rigid except perhaps at the interface, might be carried out by MCM within a few hours of CPU time using the grid-based energy evaluation.

Most of the advantages of the SCV-MCM method are realized when the protein is fully flexible or when protein loops are simulated. The global motion of a large or even a small protein can be characterized by concerted motions of all the atoms. Standard MC in such a system would be very inefficient because it is very unlikely to generate a low-energy conformation of the protein by randomly changing all the dihedral angles, unless the step size is very small. The ellipsoid associated with the hessian at a given point of the energy landscape might differ significantly from the long-range shape of the energy landscape at that particular region, but at least the eigenvectors of the Hessian give some information about which directions of the energy landscape should be avoided. For protein loop calculations, the SCV-MCM method always ensures that each new set of dihedral angles for the loop will more or less satisfy the loop closure penalty term of eq. (6).

The separation of the energy into different terms allows the user to experiment with different strategies for modeling protein structure. Also, it offers a much better control of the protein system during simulated annealing. Each energy term E_i has its own annealing schedule; that is, a starting and a final temperature T (where $T = 1/\lambda$), and a rate that tells how fast the λ_i is upgraded during the MC procedure.

Finally, some techniques have been incorporated into PRODOCK to calculate the structure of large ligands weakly bound to their receptor, using NMR data. The method consists of optimizing simultaneously the NOE distance constraints of the ligand and its energy of interaction with the receptor. To make this possible, a growing MC procedure has been implemented. The ligand is generated in an extended conformation inside the active site. The intramolecular energy and the NOE distance constraints are optimized during the SCV-MCM procedure using a certain schedule for these different energy terms. To avoid atomic clashes with the receptor, the size of the ligand is reduced by a certain factor (around 0.3) before calculating the intermolecular energy with the receptor. As the simulation proceeds, the ligand grows in size, and the influence of the receptor becomes more and more important. This method can be applied for the docking of a large peptide onto a well-defined groove of the receptor. For such a system, the random tumbling of the ligand into the active site (used in refs. 13 and 16) is not adequate.

Conclusion

The PRODOCK package incorporates many simulation techniques that are relevant for docking. The advantage of this package for docking simulations resides in the possibility of defining arbitrary flexibility for the ligand and the receptor molecule and the incorporation of the Bezier spline energy grid to speed up the global optimization procedure. This Bezier spline interpolation scheme provides the possibility to estimate gradients from grid points, making the grid technique fully adaptable for energy minimization. Technical issues such as forward/backward generation and quaternion parameters were shown to be important ingredients for efficient multichain simulations.

From experience, each protein system under study requires special modeling techniques to make the conformational sampling or the search for the global minimum efficient. Such techniques might consist, for example, of defining various simulated annealing temperatures for different terms of the energy, and various chain perturbation protocols depending on whether the dihedral angles to be considered belong to the ligand, to a protein loop, or to a residue side chain inside the active site. Also, the possibility of shrinking the size of the ligand or part of it has been shown to be very efficient to generate the structure of a decapeptide inside the active site cavity.[15]

Our goal was to achieve an optimal choice between the precision of the model, the degree of flexibility of each molecule, and the efficiency of the energy optimization procedure. At present, the program presents all the important features necessary to carry out protein simulations efficiently, especially docking. Further major improvements concern the updating of the energy function, e.g., incorporation of the solvation electrostatic energy using a Poisson–Boltzmann approach and the implementation of new global optimization methods. One of the latter is conformational space annealing[40] using GA. It has proven to be very efficient for calculating protein structures starting only from the amino acid sequence. Finally, by enabling PRODOCK to incorporate the 2-D and 3D structures of a given compound automatically, this will enable it to be used for database screening and drug design.

Acknowledgment

We thank J. Kostrowicki for helpful discussions.

References

1. Momany, F. A.; McGuire, R. F.; Burgess, A. W.; Scheraga, H. A. J Phys Chem 1975, 79, 2361–2381.
2. Némethy, G.; Pottle, M. S.; Scheraga, H. A. J Phys Chem 1983, 87, 1883–1887.
3. Sippl, M. J.; Némethy, G.; Scheraga, H. A. J Phys Chem 1984, 88, 6231–6233.
4. Némethy, G.; Gibson, K. D.; Palmer, K. A.; Yoon, C. N.; Paterlini, G.; Zagari, A.; Rumsey, S.; Scheraga, H. A. J Phys Chem 1992, 96, 6472–6484.
5. Abagyan, R.; Totrov, M.; Kuznetsov, D. J Comput Chem 1994, 15, 488–506.
6. Cornell, W. D.; Cieplak, P.; Bayly, C. I.; Gould, I. R.; Merz, K. M., Jr.; Ferguson, D. M.; Spellmeyer, D. C.; Fox, T.; Caldwell, J. W.; Kollman, P. A. J Am Chem Soc 1995, 117, 5179–5197.
7. Robson, B.; Platt, E. J Mol Biol 1986, 188, 259–281.
8. Higo, J.; Collura, V.; Garnier, J. Biopolymers 1992, 32, 33–43.
9. Gibrat, J. F.; Higo, J.; Collura, V.; Garnier, J. ImmunoMethods; Padlan, E. D., Ed.; Academic: New York; 1992, pp 107–125.
10. Noguti, T.; Gō, N. Biopolymers 1985, 24, 527–546.
11. Li, Z.; Scheraga, H. A. Proc Natl Acad Sci USA 1987, 84, 6611–6615.
12. Li, Z.; Scheraga, H. A. J Molec Struct (Theochem) 1988, 179, 333–352.
13. Trosset, J.-Y.; Scheraga, H. A. Proc Natl Acad Sci USA 1998, 95, 8011–8015.
14. Oberlin, D., Jr.; Scheraga, H. A. J Comput Chem 1998, 19, 71–85.
15. Maurer, M. C.; Trosset, J.-Y.; Lester, C. C.; DiBella, E. E.; Scheraga, H. A. Prot Struct Function Genet, in press.
16. Trosset, J.-Y.; Scheraga, H. A. J Comput Chem, in press.
17. DiBella, E. E.; Scheraga, H. A. Biochemistry 1996, 35, 4427–4433.
18. Bode, W.; Mayr, I.; Baumann, U.; Huber, R.; Stone, S. R.; Hofsteenge, J. *EMBO J* 1989, 8, 3467–3475.
19. Bode, W.; Turk, D.; Karshikov, A. Prot Sci 1992, 1, 426–471.
20. Gibson, K. D.; Scheraga, H. A. J Comput Chem 1994, 15, 1403–1413.
21. Strang, G. Linear Algebra and Its Applications, 3rd ed.; Harcourt, Brace, Jovanovich: New York; 1988, p 172.
22. Lambek, J. Math Intell 1995, 17, 7–15.
23. Goldstein, H. Classical Mechanics, 2nd ed.; Addison-Wesley: Reading, MA; 1980, p 146.
24. Griewank, A. O.; Markey, B. R.; Evans, D. J. J Chem Phys 1979, 71, 3449–3454.
25. Kearsley, S. K. J Comput Chem 1990, 11, 1187–1192.
26. Marsaglia, G. Ann Math Stat 1972, 43, 645–646.
27. Vesely, F. J Comput Phys 1982, 47, 291–296.
28. Augspurger, J. D.; Scheraga, H. A. J Comput Chem 1996, 17, 1549–1558.
29. Kostrowicki, J.; Scheraga, H. A. DIMACS Series in Discrete Mathematics and Theoretical Computer Science, 1996, 23, 123–132.
30. Levitt, M.; Annu Rev Biophys Bioeng 1982, 11, 251–271.
31. Gay, D. M. ACM Trans Math Software 1983, 9, 503–524.
32. McCammon, J. A.; Harvey, S. C. Dynamics of Proteins and Nucleic Acids; Cambridge University Press: New York; 1987, p 38.
33. Jorgensen, W. L.; Tirado-Rives, J. J Phys Chem 1996, 100, 14508–14513.
34. Metropolis, N.; Rosenbluth, A. W.; Rosenbluth, M. N.; Teller, A. H.; Teller, E. J Chem Phys 1953, 21, 1087–1092.
35. Leontidis, E.; Suter, U. W. Mol Phys 1994, 83, 489–518.
36. Lynden-Bell, R. M.; Stone, A. J. Mol Simul 1989, 3, 271–281.
37. Kirkpatrick, S.; Gelatt, C. D., Jr.; Vecchi, V. P. Science 1983, 220, 671–680.
38. Evans, D. J.; Murad, S. Mol Phys 1977, 34, 327–331.
39. Ding, H.-D.; Karasawa, N.; Goddard, W. A., III. J Phys Chem 1992, 97, 4309–4312.
40. Lee, J.; Scheraga, H. A.; Rackovsky, S. J Comput Chem 1997, 18, 1222–1232.