Energy minimization of protein-ligand complexes using GROMACS 4.0.5

Walter Filgueira de Azevedo Jr

Porto Alegre – RS, Brazil.

2009

Walter F. de Azevedo Jr.

http://azevedolab.net

Faculdade de Biociências-PUCRS. Av. Ipiranga, 6681, CEP 90619-900, Porto Alegre, Rio Grande do Sul, Brazil, e-mail: walter@azevedolab.net

Telephone:

+55 51 3353-4529

“Computational scientists solve tomorrow’s problems with yesterday’s computers; computer scientists seem to do it other way around.”

-anonymous

From: Landau, R. H. A first course in Scientific computing. Princeton University Press, 2005. pp-1.

Index

Introduction

Most of the processes related to simulations of biomolecular structures rely on energy minimization (EM) of the system. More specifically, in homology modeling of protein structure a key feature is the optimization of the model, which can be achieved by application of EM methods [1], such as conjugate gradient (CG) [2] and steepest descent (SD) [3]. Furthermore, molecular dynamics (MD) simulations usually start with energy minimization process [4]. The minimized structure is then submitted to MD simulation. From the computational point of view, EM of a protein model is a nonlinear optimization problem. This class of problem involves minimizing an energy function of several variables, which may be subject to restrictions on the values of the variables [5]. Many computational methods have been developed over a number of years for the location of stationary points, some of which are appropriate for simulation of biomolecular systems. In these systems we deal with large molecules, such as proteins, DNA and complexes involving protein and ligands, and as a result the energy functions will depend on thousands of variables [5].

This tutorial describes the applications of the two major EM algorithms to simulation of complexes involving protein and ligand. These methodologies are very useful simulations when one wants to optimize the structure protein-ligand complexes [1]. We begin with an overview of both methodologies, followed by a description of an EM protocol applied to analysis of protein-ligand complexes. We have been using this protocol to successfully optimize structures of several protein targets. All protocols described here are intended to be run in the program GROMACS 4.0 [4]. This program was developed for molecular dynamics and energy minimization simulations of biomolecular systems [4]. It is currently in its 4.0 version (October 2009).

Energy surface

Many problems in simulation of biomolecular systems can be formulated as an optimization of a multi-dimensional function. Optimization is general expression for locating stationary points on a potential energy surface. In most of the cases, the desired stationary point is a minimum, i.e. all second derivatives of this function are positive. In order to establish the general aspects related to optimization methods we need some definitions about surfaces. One dimensional potential energy function shows minima, maxima and points of inflexion. For higher-dimensional surfaces we may have a new attribute referred to as saddle point, as shown in Fig. (**1**). This figure is a plot of the function $f(x,y) = y^2 – x^2$, the vertical axis is shown as a function f(x,y), where we can clearly see the saddle. These points are a maximum in some variables and a minimum in the reminder. Most optimization methods determine the nearest stationary point, but a multi-dimensional functional may contain many different stationary points of the same type.

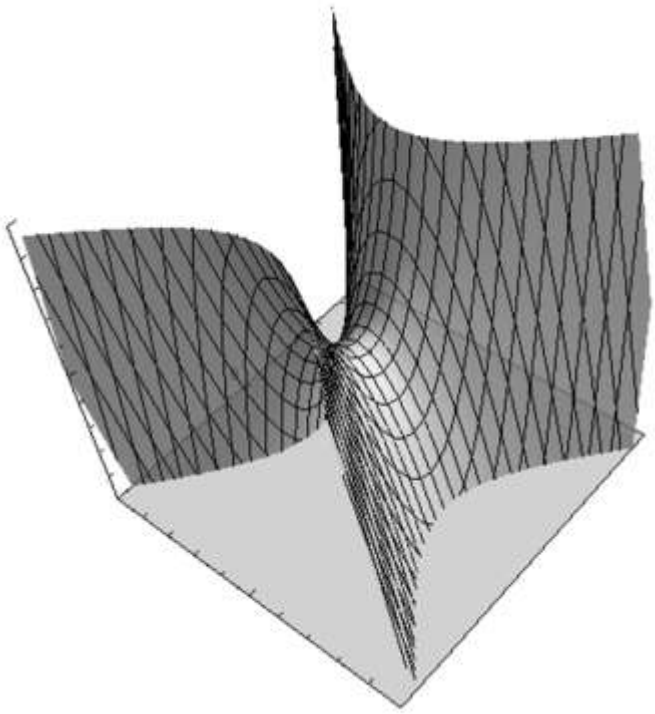

Fig. (1). *Plot of the function f(x,y) = $y^2 - x^2$, the vertical axis is shown as a function f(x,y), where we have the saddle points.*

Let us consider an energy potential function V(**r**) for a system with N atoms, where we have 3N degrees of freedom. We need n independent variables to describe this system, which we denote here as **r** (x_1, x_2,…x_n) , and it is expressed by the vector **r** represented below,

$$\mathbf{r} = \begin{pmatrix} x_1 \\ x_2 \\ \cdots \\ x_n \end{pmatrix} \qquad \text{(Eq. 1)}$$

The gradient of the potential energy V is given by

$$\mathbf{g} = \begin{pmatrix} \frac{\partial V}{\partial x_1} \\ \frac{\partial V}{\partial x_2} \\ \cdots \\ \frac{\partial V}{\partial x_n} \end{pmatrix} \qquad \text{(Eq. 2)}$$

The hessian of V(**r**) is expressed by the following equation,

$$\mathbf{H} = \begin{pmatrix} \frac{\partial^2 V}{\partial x_1^2} & \cdots & \frac{\partial^2 V}{\partial x_1 \partial x_n} \\ \cdots & \cdots & \cdots \\ \frac{\partial^2 V}{\partial x_n \partial x_1} & \cdots & \frac{\partial^2 V}{\partial x_n^2} \end{pmatrix} \quad \text{(Eq. 3)}$$

At a stationary point we have that the gradient of V(**q**) is zero. In order to further characterize this stationary point we need to find the eigenvalues of the hessian calculated at that point. We a minimum point when all eigenvalues are positives, the maximum point is obtained when all eigenvalues are negative. Otherwise, we have the saddle point.

So, the optimization methods deal with finding the minimum points. There are a plethora of algorithms available for such goal we are interest here in first-order algorithms, for locating the minima on molecular potential energy surfaces, as the one shown in Fig. (**2**), they will be described in the next sections.

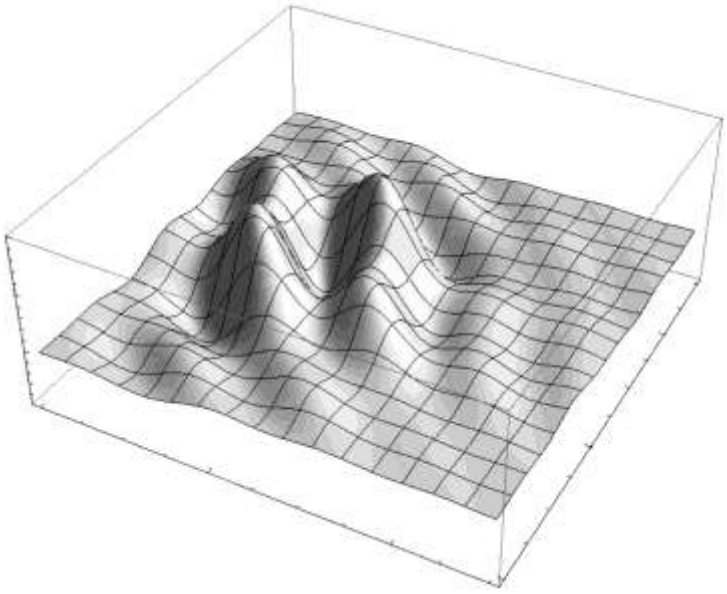

Fig. (2)*. Rugged potential surface.*

Energy minimization methods

Energy minimization is carried out in order to remove or reduce possible geometric problems in the biomolecular systems, such as improbable bond distances, bond angles and torsion angles. The file em.mdp (Table **1**) shows the commands used to inform GROMACS to perform energy minimization. This protocol employed steepest descent method for minimization, but can be easily changed to conjugate gradient.

Table 1. List of GROMACS commands needed to run energy minimization (SD method).

GROMACS Commands	**Options**
title	= prot+lig EM(SD)
define	= -DFLEX_SPC
constraints	= none
integrator	= steep
dt	= 0.002
nsteps	= 2000
nstlist	= 10
ns_type	= grid
rlist	= 1.0
coulombtype	= PME
rcoulomb	= 1.0
rvdw	= 1.4
fourierspacing	= 0.12
fourier_nx	= 0
fourier_ny	= 0
fourier_nz	= 0
pme_order	= 6
ewald_rtol	= 1e-5
optimize_fft	= yes
; Energy minimization parameters	
emtol	= 1000.0
emstep	= 0.01

Let us consider a set of independent variables **r** = (x_1,y_1,z_1,x_2,y_2,z_2,……x_n,y_n,z_n) and an objective function V(**r**), the goal here is to find that set of values for the independent variables, for which the objective function has its minimum value V(**r***) = min (V(**r**)). For a binary complex with N atoms, the 3 N components of **r** are the atomic coordinates and V is the potential energy (force field) [6]. Many of the modern molecular mechanics force fields can be interpreted in terms of the relatively simple parameters, such as bond stretching, bond angle and torsion angle. Energetic penalties are related with the deviations from equilibrium values, such as, the equilibrium point of the spring. The most common interaction potential energy (V(**r**)) for a biomolecular system involves atomic positions and can be expressed as follows:

$$V(\mathbf{r}) = \frac{1}{2}\sum_{bond} k_b (b_i - b_{i,o})^2 + \frac{1}{2}\sum_{angle} k_\theta (\theta_i - \theta_{i,o})^2 + \frac{1}{2}\sum_{torsion} V_n [1 + \cos(n\omega - \gamma)]$$
$$+ \sum_{i=1}^{N}\sum_{j=i+1}^{N}\left\{ 4\varepsilon_{ij}\left[\left(\frac{\sigma_{ij}}{r_{ij}}\right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}}\right)^{6}\right] + \frac{q_i q_j}{4\pi\varepsilon_o r_{ij}}\right\} \qquad \text{(Eq. 4)}$$

where the potential energy is function of the positions **r** of N atoms. The first term in the equation 4 provides energy due to the stretching of the bond lengths, the second term to the bending of the bond angles. The third term represents the changes in torsion angles.

The forth term accounts for non-bonded interaction, called Lennard-Jones potential, which is due to van der Waals force. The constants σ_{ij} are empirical parameters and r_{ij} is the interatomic distance between atoms i and j. The fifth term describes electrostatic interaction. This interaction may be modeled using Coulomb potential that varies as the inverse of the interatomic distance (r_{ij}). The variables q_i and q_j are partial atomic charges, these are non-integral charge values attributed to each atom in the system, which are designed to reproduce the electrostatic properties of the molecule. The permittivity of free space is represented by ε_o. In the Lennard-Jones (LJ) potential the term attraction is modeled using $1/r^6$ term. When two atoms are closer there is repulsive interaction, modeled using $1/r^{12}$ term. These two terms together compose the LJ potential.

Modern molecular dynamics programs use force fields built as described, with a small number of differences in terms and parameters used to construct the potential energy (force field). One of the most used force field for biomolecular systems is AMBER (Assisted Model Building and Energy Refined) [7]. AMBER was calibrated against accurate quantum mechanical studies and experimental information derived from techniques such as neutron diffraction and microwave spectroscopy [8]. We are not going to describe in details how this potential energy is calculated (force field) in the program GROMACS, the readers interested in such details may read the original GROMACS reference [4]. GROMACS 4.0 includes quite a few force fields, and additional ones are available on the website, and as described in the GROMACS manual page 89 (www.gromacs.org): "Whenever using this force field, please cite the references [9-14], and do not call it GROMACS force field, instead name it GROMOS-87 [12] with corrections as detailed in [13, 14]."

To understand the first-order minimization methods used in modern molecular simulation packages such as GROMACS [4] let us consider a well-behaved function f(x). This function f(x) needs to be a well-behaved function then we need the gradient to be continuous. The gradient (**g**) is the vector whose ith component is $\partial f/\partial x_i$. It points towards the direction of maximum rate of increase of f(x). On the other hand, $-\nabla f$ points towards the direction of greatest decrease. The vector ∇f is, at any point x_o, normal to the contour of constant functions passing through x_o. For an n-dimensional case the gradient of function f is given by,

$$\mathbf{g} = \begin{pmatrix} \dfrac{\partial f}{\partial x_1} \\ . \\ . \\ . \\ \dfrac{\partial f}{\partial x_n} \end{pmatrix} \qquad \text{(Eq. 5)}$$

The function f(x) can be expanded as Taylor series about the point x_o as follows,

$$\mathrm{f}(x) = \mathrm{f}(x_o) + (x - x_o)\frac{d\mathrm{f}(x_o)}{dx} + \frac{(x - x_o)^2}{2}\frac{d^2\mathrm{f}(x_o)}{dx^2} + \ldots \qquad \text{(Eq. 6)}$$

We may generalize this function to n dimensions substituting x by the vector **r** and introducing matrices for the various derivatives. A given optimization method can be classified by its order, which is the highest order derivative that is employed in the method.

First order methods cut the Taylor expansion after the second term, making use of information associated with local slope of the potential energy surface (the first derivative). Physically, this first derivative of the potential energy is the force (**F**), as follows,

$$\mathbf{F} = -\nabla V \qquad \text{(Eq. 7)}$$

Here we need to make a small note to explain how all algorithms will be described in the this tutorial. All algorithms described here are iterative; that is, we begin from some start point on a surface, and then proceed further in cycles, according to an algorithm hopefully toward a stationary point. Each cycle is known as iteration, and it will be represented by a symbol k to indicate the iteration count. We use this symbol as subscript to variables to indicate this iteration count. For instance, the biomolecular configuration prior to the kth iteration is specified by the 3N dimensional vector $\mathbf{r}_{k-1}$. This is the initial structure, which can be a structure obtained from X-ray crystallography. The SD method is classified as a first-order method and was put forward by Wiberg [3]. The method of SD for finding a local minimum of f(x) consists of three steps.

1) A descendent direction is taken, which is represented by a 3N dimensional vector of unit length, $\mathbf{u}_k$.
2) A descent step size, indicated by the scalar λ_k, is determined.
3) Finally, the descent step is taken according to the following equation,

$$\mathbf{r}_k = \mathbf{r}_{k-1} + \lambda_k \mathbf{u}_k \qquad \text{(Eq. 8)}$$

where k indicates the iteration count. The unit vector ($\mathbf{u}_k$) is given by the following equation,

$$\mathbf{u}_k = \frac{\mathbf{F}}{|\mathbf{F}|} \qquad \text{(Eq. 9)}$$

In most of the applications of the SD method in energy minimization the step size firstly has a predetermined default values. If the initial iteration leads to a decrease in energy, the step size is augmented by a multiplicative factor (e.g. 1.2) for the second iteration. This process is repeated as long it reduces the energy. When a step produces a raise in energy function, it is understood that the algorithm has leapt across the valley which contains the minimum and up the slope on the opposite face [15]. The step size is then reduced by a multiplicative factor (e.g. 0.2). The step size depends on the characteristics of the energy function; for an energy function represented by a flat surface, large step size would be more appropriate. On the other hand, rugged energy surfaces as shown in Fig. (**2**) are difficult to be evaluated with SD algorithm, since a large step size would take far from the minimum. The rugged potential energy surfaces are characteristics of multidimensional biomolecular systems, such as protein structures.

Specifically for implementation of SD method in GROMACS, a maximum initial displacement (λ) should be given (e.g. 0.01 nm). Then force and potential energy are calculated, and new positions $\mathbf{r}_k$ are calculated by a modified equation (8), given by,

$$\mathbf{r}_k = \mathbf{r}_{k-1} + \lambda_k \mathbf{t}_k \qquad \text{(Eq. 10)}$$

In the GROMACS the unit vector $\mathbf{u}_k$ is replaced by $\mathbf{t}_k$ given by the following equation,

$$\mathbf{t}_k = \frac{\mathbf{F}_k}{\max\left(|\mathbf{F}_k|\right)} \qquad \text{(Eq. 11)}$$

where $\mathbf{F}_k$ is the force. The notation max($|\mathbf{F}_k|$) means the largest of the absolute values of the force components. The forces and potential energy are again calculated for the new positions.

If ($V_k < V_{k-1}$) the new positions are accepted and $\lambda_{k+1} = 1.2\ \lambda_k$.

If ($V_k \geq V_{k-1}$) the new positions are rejected and $\lambda_{k+1} = 0.2\ \lambda_k$.

The algorithm stops when either a user specified number of force evaluations (nstep in Table **1**) has been performed (e.g. 100), or when the maximum of the absolute values of the force (gradient) components is smaller than a specified value (emtol in Table **1**). Since force truncation generates some noise in the potential energy evaluation, the stopping criterion should not be made too tight to avoid endless iterations.

When running the implementation of SD algorithm for energy minimization in the GROMACS the command integrator is set to steep (integrator = steep, in Table **1**), the maximum step size is indicated by emstep [nm, 10^{-9} m], and the tolerance is emtol [kJ mol^{-1} nm^{-1}]. For the protocol described here emstep = 0.01 nm and emtol = 1000 kJ mol^{-1} nm^{-1}, as shown in Table **1**.

GROMACS presents yet another energy minimization method, known as conjugate gradient algorithm. This method has been introduced by Fletcher and Reeves [2]. Conjugate gradient algorithm produces a set of directions which does not show the oscillatory behavior of the SD algorithm in rugged surfaces (Fig. (**2**)). In the SD algorithm, both gradient and the direction of successive steps are normal to the contour of constant energy functions. On the other hand, in the conjugate gradient method, the gradients at each point are orthogonal but the directions are not. These directions are conjugate, indeed, as said by Leach in page 265 [8]: “the method is more properly called the conjugate directions method”.

Starting from an initial structure (iteration 1), where the gradient is $\mathbf{g}_1$, the first search direction is given by,

$$\mathbf{s}_1 = -\mathbf{g}_1 \qquad \text{(Eq. 12)}$$

For n-th iteration, the conjugate gradient algorithm moves in a direction $\mathbf{s}_k$ from point $\mathbf{x}_k$ where $\mathbf{s}_k$ is determined from the gradient at the point and the preceding direction vector $\mathbf{s}_{k-1}$, as follows,

$$\mathbf{s}_k = -\mathbf{g}_k + b_k \mathbf{s}_{k-1} \qquad \text{(Eq. 13)}$$

where the parameter b_k is a weighting factor given by

$$b_k = \frac{\mathbf{g}_k \cdot \mathbf{g}_k}{\mathbf{g}_{k-1} \cdot \mathbf{g}_{k-1}} \qquad \text{(Eq. 14)}$$

CG method is considered slower than SD in the first steps of the energy minimization, but becomes faster closer to the energy minimum, and the overall performance may be faster for CG method as we will see in the next sections. In the program GROMACS, the parameters and stop criterion are the same as for SD.

In addition to CG and SD, GROMACS has option to perform energy minimization using L-BFGS (limited-memory Broyden-Fletcher-Goldfarb-Shanno quasi-Newtonian minimizer). This algorithm approximates the inverse hessian by a predetermined number of corrections from previous steps [16, 17]. For minimization of protein-ligand complexes discussed in this tutorial we employed both methodologies, in order to compare the overall performance of them.

Commands needed to run GROMACS 4.0.5 for EM

In this tutorial all commands are shown in *italics* in the following lines. Lines started with ">" are command lines. We employed the double precision option to run GROMACS, in order to have more accurate results. A small comment here, you may generate lig.gro directly the PRODRG [18], we keep this step for didactic reasons. To generate GROMACS structure files (__.gro) we do as follows,

>editconf_d -f lig.pdb -o lig.gro
>pdb2gmx_d -ignh -ff gmx -f prot.pdb -o prot.gro -p prot.top -water spce

Edit file *lig.gro* and copy atomic coordinates and paste in the file *prot.gro*. Add number of atoms to the second line of file *prot.gro*. Edit the file *prot.top* and add the following line "#include lig.itp", after the "*forcefield*" include information about the ligand in the last line. The information to be included in the last line is the following:

LIG 1

And then type the following commands:

>editconf_d -bt cubic -f prot.gro -o prot.gro -d 1.0
>genbox_d -cp prot.gro -cs spc216.gro -o prot_b4ion.gro -p prot.top
>grompp_d -f em.mdp -c prot_b4ion.gro -p prot.top -o prot_b4ion.tpr

Here we may get the message "Note: system has non-zero charge", and also indication of charge. This information will be employed in the following step "genion". For instance, if your system presents a charge of +4 you need to neutralize this positive charge adding counter-ions, specifically we neutralize the positive charge adding Chlorine atoms, as follows:

>genion_d -s prot_b4ion.tpr -o prot_b4em.gro -nname Cl -nn 4 -g prot_ion.log

Choose option 13. Edit the file *prot.top* and subtract the number of charges from the number of solvent molecules. For instance, the line with information about the solvent was:

SOL 18725

and it will be,

SOL 18721

Now we add a line with ion, for instance Cl, as follows:

Cl 4

For systems with negative charges we add positive charges represented by Na ions. The commands for energy minimization using SD are described in Table 1. To change the EM

to CG you should edit em.mdp file modify the commando integrator CG. We now run energy minimization step, we type the following commands:
>grompp_d -f em.mdp -c prot_b4em.gro -p prot.top -o em.tpr
>mdrun_d -v -s em.tpr -e em -o em -c after_em -g emlog >& em.job &

Now we have finished our EM for a complex involving protein and ligand. We will describe the analysis of potential energy. Use "g_energy" to analyze the potential energy output (the md.edr file), as follows,
>g_energy_d –f em.edr –o em.xvg

We entered "Potential" followed by <enter> to end selection. All energies are expressed in kJ/mol. Use the g_energy command to obtain other components (e.g. potential energy, etc.). To visualize we use the xmgrace, which is WYSIWYG tool to generate two-dimensional plots of numerical data. It runs under various (if not all) flavors of Unix/Linux. It also runs under VMS, OS/2, and Windows (95/98/NT/2000/XP). It can be downloaded at http://plasma-gate.weizmann.ac.il/Grace/ . To generate a plot for potential energy do as follows,
>xmgrace em.xvg

In the next sections three applications of both EM methods will be discussed, all involving protein targets employed in SBVS studies. The binary complexes involving CDK2-roscovitine, PNP-Immucillin-H and SK-shikimate were minimized using SD and CG methodologies. All simulations were performed in a HP mobile workstation Processor Intel Core 2 Duo T9400 / 2.53 GHz, 4GB of RAM, which shows a performance of 2594.810 GFlops.

Human cyclin-dependent kinase 2

We have applied the above described EM protocols to the structure of CDK2 (EC 2.7.11.22) in complex with roscovitine (PDB access code: 2a4l) [19]. Plots for potential EM simulations using SD and CG methodologies are shown in Fig. (**3**) and Fig. (**4**), respectively. For SD we have that the structure converged 2317 steps, to a potential energy of -10.3 $.10^5$ kJ/mol in a CPU time of 1926 s. CG minimization converged in 106 energy evaluations, to a potential energy of -9.40 $.10^5$ in a CPU time of 90 s. The CG is 21.6 times faster than SD nevertheless SD generated a structure with lower potential energy.

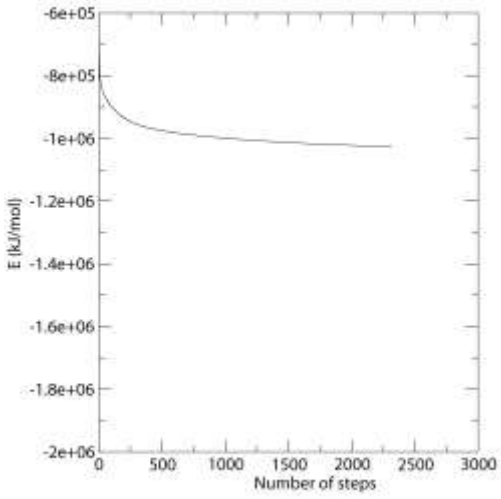

***Fig. (3)**. EM of the complex CDK2-roscovitine using SD methodology.*

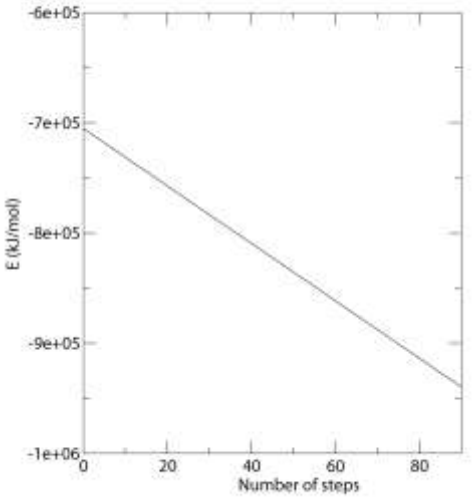

***Fig. (4)**. EM of the complex CDK2-roscovitine using CG methodology.*

Human purine nucleoside phosphorylase (HsPNP)

EM protocols were applied to the structure of PNP (EC 2.4.2.1) in complex with immucilin-H (PDB access code: 1pf7) [20]. Plots for potential EM simulations using SD and CG methodologies are shown in Fig. (**5**) and Fig. (**6**), respectively. For SD we have that the structure converged 2346 steps, to a potential energy of -9.40 $.10^5$ kJ/mol in a CPU time of 1804 s. CG minimization converged in 106 energy evaluations, to a potential energy of -8.87 $.10^5$ in a CPU time of 138 s. The CG is more 17 times faster than SD

nevertheless SD generated a structure with lower potential energy, as observed for the complex involving CDK2 and roscovitine.

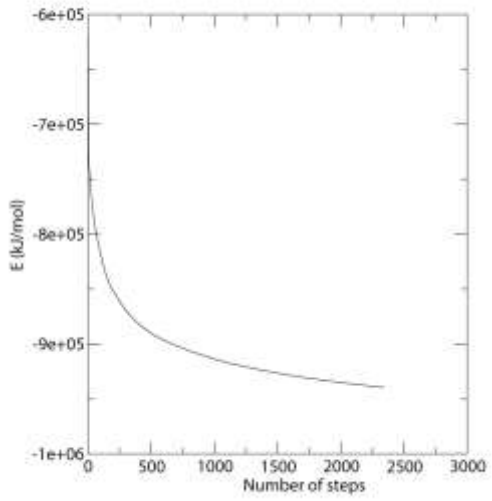

***Fig. (5)**. EM of the complex PNP-ImmH using SD methodology.*

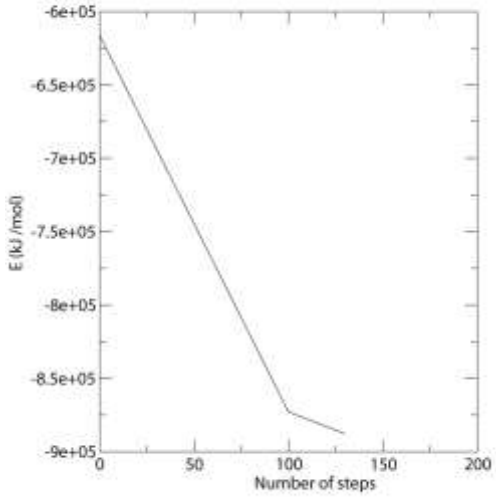

***Fig. (6)**. EM of the complex PNP-ImmH using CG methodology.*

Shikimate kinase

We have applied the above described EM protocols to the structure of SK (EC 2.7.1.71) in complex with shikimate (PDB access code: 1we2) [21]. Plots for potential EM

simulations using SD and CG methodologies are shown in Fig. (**7**) and Fig. (**8**), respectively. For SD we have that the structure converged 553 steps, to a potential energy of -5.60 $.10^5$ kJ/mol in a CPU time of 250 s. CG minimization converged in 33 energy evaluations, to a potential energy of - 5.22 $.10^5$ in a CPU time of 16 s. The CG is more 15.6 times faster than SD nevertheless SD generated a structure with lower potential energy, as observed for previously analyzed complexes.

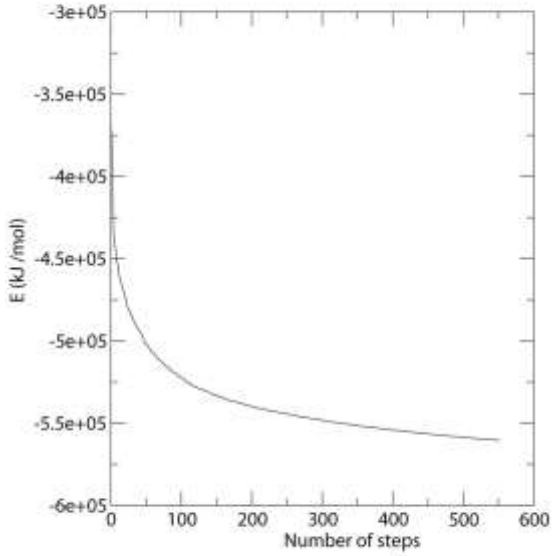

***Fig. (7)**. EM of the complex SK-shikimate using SD methodology.*

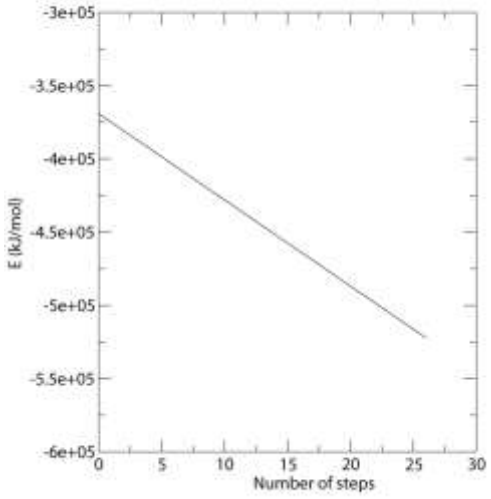

***Fig. (8)**. EM of the complex SK-shikimate using CG methodology.*

Final remarks

In this tutorial we discussed the main features of SD and CG methodologies that can be employed to evaluate protein-ligand interaction. All examples were based on protein targets that we have been working on the last years [19-21]. From the three examples discussed here, we summarize the key structural characteristics inferred from MD simulations focused on protein-ligand complexes.

1) CG algorithm is from 15.6 to 21.6 times faster than SD, which indicates the superiority of this algorithm for time performance. Analysis of potential energy plots (Figures 3 to 8) clearly shows the dramatically drop of energy in few steps of minimization.
2) Analysis of potential energy at the end of minimization process indicated that the SD presents better performance in all three cases analyzed here.
3) Since reaching global minima is the ‘ultimate goal’ of EM algorithms it is tempting to speculate that we should employ SD for simulations of protein-ligand complexes to guarantee such objective.

Abbreviations

CG = Conjugate gradient
CDK = Cyclin-dependent kinase
HsPNP = Human purine nucleoside phosphorylase
ImmH = Immucillin-H
MtSK = Shikimate kinase from *Mycobacterium tuberculosis*
PDB = Protein data bank
PNP = Purine pucleoside phosphorylase
SBSV = Structure-based virtual screening
SD = Steepest descent
SK = Shikimate kinase

References

[1] Uchoa, H. B.; Jorge, G. E.; da Silveira, N. J.; Camera, J. C.; Canduri, F. and De Azevedo Jr., W.F. (2004) *Biochem. Biophys. Res. Commun.* **325**(4), 1481-1486.

[2] Fletcher, R. and Reeves, C. M. (1964) *Comp. J.*, **7**, 149-154.

[3] Wiberg K. B. (1965) *J. Am. Chem. Soc.,* **87**, 1070-1078.

[4] van der Spoel, D.; Lindahl, E.; Hess, B.; Groenhof, G.; Mark, A. E. and Berendsen, H. J. C. (2005) *J. Comp. Chem.* **26**, 1701–1718.

[5] Hinchliffe, A. Molecular modelling for beginners. John Wiley & Sons Ltd.: West Sussex, 2003.

[6] McCammon, J. A. and Harvey, S. C. Dynamics of protein and nucleic acids, Cambridge University Press.: Cambridge, 1987.

[7] 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. and Kollman, P.A. (1995) *J. Am. Chem. Soc.*, **117**, 5179-5197.

[8] Leach, A.R. Molecular Modeling. Principles and Applications, 2nd Ed. Pearson Education Limited: Essex, 2001.

[9] van Buuren, A. R.; Marrink, S. J. and Berendsen, H. J. C. (1993) *J. Phys. Chem*. **97**, 9206–9212.

[10] Mark, A. E.; van Helden, S. P.; Smith, P. E.; Janssen, L. H. M. and van Gunsteren, W. F. (1994) *J. Am. Chem. Soc.* **116**, 6293–6302.

[11] Jorgensen,W. L.; Chandrasekhar, J.; Madura, J. D.; Impey, R.W. and Klein, M. L. (1983) *J. Chem. Phys.* **79**, 926–935.

[12] van Gunsteren, W. F., Berendsen, H. J. C. Gromos-87 manual. Biomos BV Nijenborgh 4, 9747 AG Groningen, The Netherlands 1987.

[13] van Buuren, A. R. and Berendsen, H. J. C. (1993) *Biopolymers,* **33**, 1159–1166.

[14] Liu, H.; Muller-Plathe, F. and van Gunsteren, W. F. (1995) *J. Am. Chem. Soc.* 117, 4363–4366.

[15] Radaport, D. C. The art of molecular dynamics simulations, 2nd Ed. Cambridge University Press: New York, 2004.

[16] Byrd, R. H.; Lu, P. and Nocedal, J. (1995) *SIAM J. Scientif. Statistic. Comput*. **16**, 1190–1208.

[17] Zhu, C.; Byrd, R. H. and Nocedal, J. (1997) *ACM Trans. Math. Softw.* **23**, 550–560.

[18] Schüttelkopf A. W. and van Aalten D. M. (2004) Acta *Crystallogr. D Biol. Crystallogr.* **60**(Pt 8), 1355-63.

[19] De Azevedo Jr., W.F.; Leclerc, S.; Meijer, L.; Havlicek, L.; Strnad, M. and Kim, S. H. (1997) *Eur. J. Biochem.*, **243**(1-2), 518-526.

[20] De Azevedo Jr., W.F.; Canduri, F.; Dos Santos, D.M.; Pereira, J.H.; Dias, M.V.B.; Silva, R.G.; Mendes, M.A.; Basso, L.A.; Palma, M.S. and Santos, D.S. (2003) *Biochem. Biophys. Res. Commun.*, **309**(4), 917-922.

[21] Pereira, J.H.; de Oliveira, J.S.; Canduri, F.; Dias, M.V.; Palma, M.S.; Basso, L.A.; Santos, D.S. and De Azevedo Jr., W.F. (2004) *Acta Crystallogr D Biol Crystallogr*, **60**(Pt 12 Pt 2), 2310-2319.