Molecular Dynamics Simulation of EPSP Synthase

Maria Cano Colino. Universitat Autònoma de Barcelona
DESY, Summer Student Program

September 17th, 2007

Supervised by Hans Bartunik and Grzegorz Domanski
Max-Planck Unit for Structural Molecular Biology, Hamburg, Germany

Contents

1. INTRODUCTION

1.1. Molecular Dynamics

Molecular Dynamics (MD) is a form of computer simulation, wherein atoms and molecules are allowed to interact for a period of time under known laws of physics, giving a view of the motion of the atoms. Because molecular systems generally consist of a vast number of particles, it is impossible to find the properties of such complex systems analytically; MD simulation circumvents this problem by using numerical methods. It represents an interface between laboratory experiments and theory, and can be understood as a "virtual experiment". One of MD's key contributions is creating awareness that molecules like proteins and DNA are machines in motion. MD probes the relationship between molecular structure, movement and function.

The two main families of simulation technique are molecular dynamics and Monte Carlo (MC); additionally, there is a whole range of hybrid techniques which combine features from both. In contrast with the Monte Carlo simulations, molecular dynamics is a deterministic technique if we use a deterministic dynamics: given an initial set of positions and velocities, the subsequent time evolution is completely determined. In more pictorial terms, atoms will “move” into the computer, bumping into each other, wandering around (it the system is fluid), oscillating in waves in concert with their neighbours, perhaps evaporating away the system if there is a free surface, and so on, in a way pretty similar to what atoms in a real substance would do. This is the advantage of MD over MC, MD gives a route to dynamical properties of the system: transport coefficients, time-dependent responses to perturbations, spectra and other properties.

MD is a multidisciplinary field. Its laws and theories stem from mathematics, physics, and chemistry, and it employs algorithms from computer science and information theory. It was originally conceived within theoretical physics in the late 1950's, but is applied today mostly in material science and biomolecules.

1.1.1. Areas of application

Only a briefly mention a few areas of current interest where MD has brought and/or could bring important contributions.

Biomolecules: MD allows studying the dynamics of large macromolecules, including biological systems such as proteins, nucleic acids (DNA, RNA), membranes. Dynamical events may play a key role in processes which affect functional properties of the biomolecule. Drug design is commonly used in the pharmaceutical industry to test properties of a molecule at the computer without the need to synthesize it.

Molecular dynamics as an optimization technique: Molecular dynamics is broadly used as an optimization technique. On the refinement phase of protein crystallography, its contribution is invaluable. Better and faster algorithms might improve in the future the convergence rate of the optimizations.

Liquids: Availability of new realistic interaction models allows studying new systems, elemental and multicomponent. Through non-equilibrium techniques, transport phenomena such as viscosity and heat flow have been investigated. And we must remember that any protein, membrane or DNA that we simulate is in water.

Defects in solids: The defects are crucial for the mechanical properties in solids and therefore of technological interest. The focus shifted perhaps from point defects (vacancies, interstitials) to linear (dislocations) and planar (grain boundaries, stacking faults) defects.

Fracture: Under mechanical action, solids break into two or more pieces. The fracture process can occur in different ways and with different speeds depending of several

parameters. The technological importance is obvious, and simulation is providing useful insights on the fracture process.

Surfaces: Simulation is playing a big role in understanding phenomena such as surface reconstructions, surface melting, faceting, surface diffusion, roughening, etc, often requiring large samples and simulation times. Electronical properties of surfaces also become a key role on science, as newer integration technologies work on the surfaces of Silicon dice. Simulating surfaces on MOS (Metal-Oxide-Semiconductor) transistors and MOS circuits is a key technology to built faster devices.

Friction: Even more recent are investigations of adhesion and friction between two solids, propelled by the development of the atomic force microscope (AFM). The body of macroscopic knowledge is being revised and expanded on microscopic grounds.

Clusters: Clusters (conglomerates of a number of atoms ranging from a few to several thousands) constitute a bridge between molecular systems and solids, and exhibit challenging features. Frequently, an astonishingly large number of different configurations have very similar energies, making it difficult to determine stable structures. Their melting properties can also be significantly different from those of the solid, due to the finite size, the presence of surfaces and the anisotropy. Metal clusters are extremely important from the technological point of view, due to their role as catalysts in important chemical reactions

Electronic properties and dynamics: The development of the Car-Parrinello method, where the forces on atoms are obtained by solving the electronic structure problem instead of by an interatomic potential, allows to study electronic properties of materials fully including their dynamics and, therefore, phase transitions and other temperature-dependent phenomena.

1.1.2. Classical MD. Physical principles

The simplest simulation that we can do is the classical MD (CMD), also called molecular mechanics (MM). CMD using “predefined potentials", either based on empirical data or on independent electronic structure calculations, is well established as a powerful tool to investigate many-body condensed matter systems. The broadness, diversity, and level of sophistication of this technique have been confirmed in several monographs as well as proceedings of conferences and scientific schools.

This simulation consists of the numerical, step-by-step, solution of the classical equations of motion, which for a simple atomic system may be written:

$$m_i \ddot{r}_i = f_i \qquad f_i = -\frac{\partial}{\partial r_i} U \qquad (1)$$

The ingredients for a program are basically threefold:

(i) A model for the interaction between system constituents (atoms, molecules, surfaces etc.) is needed.

(ii) An integrator is needed, which propagates particle positions and velocities from time t to $t + \delta t$.

(iii)A statistical ensemble has to be chosen, where thermodynamic quantities like pressure, temperature or the number of particles are controlled.

These steps essentially define an MD simulation. Having this tool at hand, it is possible to obtain *exact* results within numerical precision. Results are only correct with respect to the model which enters into the simulation and they have to be tested against theoretical predictions and experimental findings.

(i) Models for Particle Interactions

The CMD methods are governed by the system's Hamiltonian and consequently Hamiltonian's equation of motion:

$$\dot{p}_i = -\frac{\partial H}{\partial q_i} \qquad \dot{q}_i = -\frac{\partial H}{\partial p_i} \qquad (2)$$

The Hamiltonian can be written as *intrinsic* part H_0 and *external* part $H_1(t)$:

$$H = H_0 + H_1(t) \qquad (3)$$

H_1 is an external part, which can include time dependent effects or external sources for a force. If the external part of the Hamiltonian is omitted then it is clear from classical mechanics that the system Hamiltonian is a conserved quantity. The intrinsic part of the Hamiltonian can often be written as:

$$H_0 = \sum_{i=1}^{N} \frac{p_i^2}{2m_i} + \sum_{i<j}^{N} u(r_i, r_j) + \sum_{i<j}^{N} u^{(3)}(r_i, r_j, r_k) + \ldots \qquad (4)$$

where **p** is the momentum, **m** the mass of the particles and **u** and $\mathbf{u^{(3)}}$ are pair and three-body interaction potentials.

If not only pair 3-body interactions are to be considered multi-body potentials $u^{(n)}$ can be included into the Hamiltonian. Mainly this is avoided, since it is not easy to model and also it is rather time consuming to evaluate potentials and forces originating from these many-body terms.

All simulated objects are defined within a model description. Often a precise knowledge of the interaction between atoms, molecules or surfaces is not known and the model is constructed in order to describe the main features of some observables. Besides boundary conditions, which are imposed, it is the model which completely determines the system from the physical point of view. In classical simulations the *objects* are most often described by point-like centres which interact through pair or multibody interaction potentials. In that way the highly complex description of electron dynamics is abandoned and an effective picture is adopted where the main features like the hard core of a particle, electric multipoles or internal degrees of freedom of a molecules are modelled by a set of parameters and (most often) analytical functions which depend on the mutual position of particles in the configuration. Since the parameters and functions give complete information of the system's energy as well as the force acting on each particle through $F = -\nabla U$, the combination of parameters and functions is also called a *force field*. Different types of force field were developed during the last ten years; one of them is the AMBER force field which we will use for our simulations.

There may be different terms contributing to the interaction potential between particles, i.e. there is no universal expression, as one can imagine for first principles calculations. In fact, contributions to interactions depend on the model which is used and this is the result of collecting various contributions into different terms, coarse graining interactions or imposing constraints, to name a few. Generally one can distinguish between bonded and non-bonded terms, or intra- and inter-molecular terms:

$$U = \sum_{bonded} U + \sum_{non-bonded} U \qquad (5)$$

<u>Non-bonded Interactions:</u>

This class denotes all contributions originating between particles which are closely related to each other by constraints or potentials which guaranty defined particles as close neighbours. It is traditionally split into 1-body, 2-body, 3-body…terms:

$$U_{non-bonded} = \sum_i u(r_i) + \sum_i \sum_{j>i} v(r_i, r_j) + \ldots \qquad (6)$$

The **u(r)** term represents an externally applied potential field or the effects of the container walls; it is usually dropped for fully periodic simulations of bulk systems. Also, it is usual neglect three-body (and higher order) interactions.

In some simulations of complex fluids, it is sufficient to use the simplest models that faithfully represent the essential physics. We shall concentrate on continuous, differentiable pair-potentials (although discontinuous potentials such as hard spheres and spheroids have also played a role). The Lennard-Jones potential is the most commonly used form:

$$v^{LJ}(r) = 4\varepsilon\left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right] \qquad (7)$$

with two parameters: σ, the diameter, and ε, the well depth.

For applications in which attractive interactions are of less concern than the excluded volume effects which dictate molecular packing, the potential may be truncated at the position of its minimum, and shifted upwards to give what is usually termed the WCA model. If electrostatic charges are present, we add the appropriate Coulomb potentials:

$$v^{Coulomb}(r) = \frac{Q_1 Q_2}{4\pi\varepsilon_0 r} \qquad (8)$$

where Q_1, Q_2 are the charges and ε_0 is the permittivity of free space.

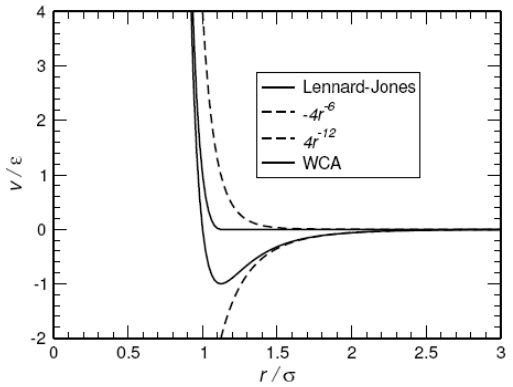

Figure 1: Lennard-Jones pair potential and the WCA termed.

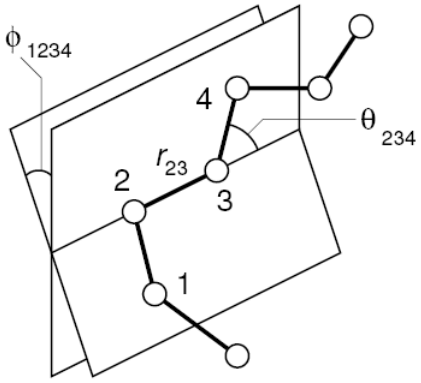

Figure 2: Geometry of a simple chain molecule

Bonded Interactions:

For molecular systems, we simply build the molecules out of site-site potentials. Typically, a single-molecule quantum-chemical calculation may be used to estimate the electron density throughout the molecule, which may then be modelled by a distribution of partial charges or more accurately by a distribution of electrostatic multipoles. For molecules we must also consider the intramolecular bonding interactions. The simplest molecular model will include terms of the following kind:

$$U_{\text{int}\,ramolecular} = \frac{1}{2}\sum_{bonds} k_{ij}^{r}\left(r_{ij} - r_{eq}\right)^2 + \frac{1}{2}\sum_{\substack{bend\\angles}} k_{ijk}^{\theta}\left(\theta_{ijk} - \theta_{eq}\right)^2 + \frac{1}{2}\sum_{\substack{torsion\\angles}}\sum_{m} k_{ijkl}^{\phi,m}\left(1 + \cos\left(m\phi_{ijkl} - \gamma_m\right)\right) \qquad (9)$$

The geometry is illustrated in Fig. 2. The "bonds" will typically involve the separation $r_{ij} = \left|r_i - r_j\right|$ between adjacent pairs of atoms in a molecular framework, and

we assume a harmonic form with specified equilibrium separation, although this is not the only possibility. The "bend angles" θ_{ijk} are between successive bond vectors such as $r_i - r_j$ and $r_j - r_k$, and therefore involve three atom coordinates: $\cos\theta_{ijk} = \hat{r}_{ij} \cdot \hat{r}_{jk}$.

Usually this bending term is taken to be quadratic in the angular displacement from the equilibrium value, although periodic functions are also used. The "torsion angles" Φ_{ijkl} are defined in terms of three connected bonds, hence four atomic coordinates: $\cos\phi_{ijkl} = -\hat{n}_{ijk} \cdot \hat{n}_{jkl}$. Usually the torsional potential involves an expansion in periodic functions of order m = 1, 2,…

A simulation package force-field will specify the precise form of the potential equation, and the various strength parameters k and other constants therein. Actually, the previous potential equation is a considerable oversimplification. Molecular mechanics force-fields, aimed at accurately predicting structures and properties, will include many cross-terms.

(ii) The integrator. MD Algorithms

For a given potential model which characterizes the physical system, it is the integrator which is responsible for the accuracy of the simulation results. If the integrator would work without any error the simulation would provide *exact model results* within the errors occurring due to a finite number representation. However, any finite difference integrator is naturally an approximation for a system developing continuously in time.

For simplicity, a system composed of atoms with coordinates $r^N = (r^1, r^2,\ldots,r^N)$ and potential energy $U(r^N)$, we introduce the atomic momentum $p^N = (p^1, p^2,\ldots,p^N)$, in terms of which the kinetic energy may be written $K(p^N) = \sum_{i=1}^{N} |p_i|^2 / 2m_i$. Then the energy, or hamiltonian, may be written as a sum of kinetic and potential terms H = K + U. Write the classical equations of motion as

$$v = \dot{r} = p/m \quad \text{and} \quad \dot{p} = f = m \cdot a \qquad (10)$$

This is a system of coupled ordinary differential equations. Many methods exist to perform step-by-step numerical integration of them. Characteristics of these equations are:

(a) They are "stiff", i.e. there may be short and long timescales, and the algorithm must cope with both.

(b) Calculating the forces is expensive, typically involving a sum over pairs of atoms, and should be performed as infrequently as possible. Also we must bear in mind that the advancement of the coordinates fulfils two functions: (i) accurate calculation of dynamical properties, especially over times as long as typical correlation times τ_a of properties a of interest (we shall define this later); (ii) accurately staying on the constant-energy hypersurface, for much longer times $\tau_{run} >> \tau_a$, in order to sample the correct ensemble.

To ensure rapid sampling of phase space, we wish to make the timestep as large as possible consistent with these requirements. For these reasons, simulation algorithms have tended to be of *low order* (i.e. they do not involve storing high derivatives of positions, velocities etc.): this allows the time step to be increased as much as possible without jeopardizing energy conservation. It is unrealistic to expect the numerical method to accurately follow the true trajectory for very long times τ_{run}. The `ergodic' and `mixing' properties of classical trajectories, i.e. the fact that nearby trajectories diverge from each other exponentially quickly, make this impossible to achieve. All these observations tend to favour the Verlet:

The Verlet algorithm:

The most commonly used time integration algorithm is probably the Verlet algorithm. The basic idea is write two third-order Taylor expressions for the position r(t), one forward and one backward in time:

$$r(t+dt) = r(t) + v(t)dt + (1/2)a(t)dt^2 + (1/6)b(t)dt^3 + O(dt^4) \quad (11)$$

$$r(t-dt) = r(t) - v(t)dt + (1/2)a(t)dt^2 - (1/6)b(t)dt^3 + O(dt^4) \quad (12)$$

where **b(t)** is the third derivates of **r** with respect to **t**. Adding the two expressions gives:

$$r(t+dt) = 2r(t) - r(t-dt) + a(t)dt^2 + O(dt^4) \quad (13)$$

This is the basic form of the Verlet algorithm. The force divided by the mass is the acceleration **a(t)**, and the force is in turn a function of the position **r(t)**:

$$a(t) = (1/m)\nabla V(r(t)) \quad (14)$$

As one can immediately see, the truncation error of the algorithm when evolving the system by **dt** is of the order of $\mathbf{dt^4}$, even if the third derivates do not appear explicitly. This algorithm is at the same time simple to implement, accurate and stable, explaining its large popularity among molecular dynamics simulations.

A problem with this version of the Verlet algorithm is that velocities are not directly generated. Whiled they are not needed for the time evolution, their knowledge is sometimes necessary. Moreover, they are required to compute the kinetic energy K, whose evaluation is necessary to test the conservation of the total energy E = K + V. This is one of the most important tests to verify that a MD simulation is proceeding correctly. One could compute the velocities from the position using:

$$v(t) = \frac{r(t+dt) - r(t-dt)}{2dt} \quad (15)$$

However, the error associated to this expression is of ordered $\mathbf{dt^2}$ rather than $\mathbf{dt^4}$. To overcome this difficulty, some variants of the Verlet algorithm have been developed. They give rise to exactly the same trajectory, and differ in what variables are stored in memory and at what times. The *leap-frog* algorithm is one of such variants where velocities are handled somewhat better.

An even better implementation of the same basic algorithm is the so-called *velocity Verlet* scheme, where positions, momentous (or velocities) and forces (or accelerations) at time **t + dt** are obtained from the same quantities at time **t** in the following way:

$$r(t+dt) = r(t) + v(t)dt + (1/2)a(t)dt^2 \quad (16)$$

$$v(t+dt/2) = v(t) + (1/2)a(t)dt \quad (17)$$

$$a(t+dt) = -(1/m)\nabla V(r(t+dt)) \quad (18)$$

$$v(t+dt) = v(t+dt/2) + (1/2)a(t+dt)dt \quad (19)$$

Note how we need 9N memory locations to save the 3N positions, velocities and accelerations, but we never need to have simultaneously stored the values at two different times for any one of these equations.

Important features of the Verlet algorithm are: (a) it is *exactly* time reversible; (b) it is symplectic (to be discussed shortly); (c) it is low order in time, hence permitting long timesteps; (d) it requires just one (expensive) force evaluation per step; (e) it is easy to program.

Constraints:

It is quite common practice in classical computer simulations not to attempt to represent intramolecular bonds by terms in the potential energy function, because these bonds have very high vibration frequencies (and arguably should be treated in a quantum

mechanical way rather than in the classical approximation). Instead, the bonds are treated as being constrained to have fixed length. In classical mechanics, constraints are introduced through the Lagrangian or Hamiltonian formalisms. Given an algebraic relation between two atomic coordinates, for example a fixed bond length **b** between atoms 1 and 2, one may write a constraint equation, plus an equation for the time derivative of the constraint

$$\chi(r_1, r_2) = (r_1 - r_2)\cdot(r_1 - r_2) - b^2 = 0 \qquad (20)$$

$$\dot{\chi}(r_1, r_2) = 2(v_1 - v_2)\cdot(r_1 - r_2) = 0 \qquad (21)$$

In the Lagrangian formulation, the constraint forces acting on the atoms will enter thus:

$$m_i \ddot{r}_i = f_i + \Lambda g_i \qquad (22)$$

where **Λ** is the undetermined multiplier and

$$g_1 = -\frac{\partial \chi}{\partial r_1} = -2(r_1 - r_2) \quad g_2 = -\frac{\partial \chi}{\partial r_2} = 2(r_1 - r_2) \qquad (23)$$

It is easy to derive an exact expression for the multiplier **Λ** from the above equations; if several constraints are imposed, a system of equations (one per constraint) is obtained. However, this exact solution is not what we want: in practice, since the equations of motion are only solved approximately, in discrete time steps, the constraints will be increasingly violated as the simulation proceeds. The breakthrough in this area came with the proposal to determine the constraint forces in such a way that the constraints are satisfied exactly at the end of each time step. For the original Verlet algorithm, this scheme is called SHAKE. The appropriate version of this scheme for the velocity Verlet algorithm is called RATTLE.

Periodic Boundary Conditions:

Small sample size means that, unless surface effects are of particular interest, periodic boundary conditions need to be used. Consider 1000 atoms arranged in a 10 x 10 x 10 cube. Nearly half the atoms are on the outer faces, and these will have a large effect on the measured properties. Surrounding the cube with replicas of itself takes care of this problem. Provided the potential range is not too long, we can adopt the minimum image convention that each atom interacts with the nearest atom or image in the periodic array. In the course of the simulation, if an atom leaves the basic simulation box, attention can be switched to the incoming image. This is shown in the figure 3. Of course, it is important to bear in mind the imposed artificial periodicity when considering properties which are influenced by long-range correlations. Special attention must be paid to the case where the potential range is not short.

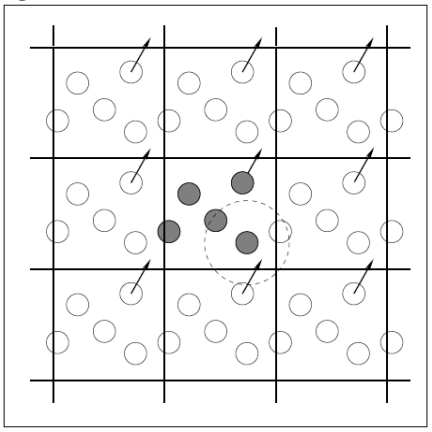

Figure 3: Periodic boundary conditions

(iii) Statistical ensemble

In MD simulations it is possible to realize different types of thermodynamic ensembles which are characterized by the control of certain thermodynamic quantities. If one knows how to calculate a thermodynamic quantity, e.g. the temperature or pressure, it is often possible to formulate an algorithm which fixes this property to a desired value. In the following, different statistical ensembles are presented:

The microcanonical ensemble:

The microcanonical ensemble (NVE) may be considered as the natural ensemble for molecular dynamics simulations. If no time dependent external forces are considered, the system's Hamiltonian is constant, implying that the system's dynamics evolves on a constant energy surface. The corresponding probability density in phase space is therefore given by:

$$\rho(q, p) = \delta(H(q, p) - E) \qquad (24)$$

In a computer simulation this theoretical condition is generally violated, due to limited accuracy in integrating the equations of motion and due to round off errors resulting from a limited precision of number representation.

The canonical ensemble:

The simplest extension to the microcanonical ensemble is the canonical one (N, V, T), where the number of particles, the volume and the temperature are fixed to prescribed values. The temperature T is, in contrast to N and V, an intensive parameter. The extensive counterpart would be the kinetic energy of the system. Different methods were proposed to fix the temperature to a fixed value during a simulation without allowing fluctuations of T:

- In the *Differential Thermostat* the velocities were scaled according to $p_1 \to \sqrt{T_0 / Tp_i}$, where T_0 is the reference temperature and T the actual temperature, calculated from the velocity of the particles. This method leads to discontinuities in the momentum part of the phase space trajectory due to the rescaling procedure. An extension of this method implies a constraint of the equations of motion to keep the temperature fixed.

- The *Proportional Thermostat* tries to correct deviations of the actual temperature T form the prescribed one T_0 by multiplying the velocities by a certain factor λ in order to move the system dynamics towards one corresponding to T_0. The difference with respect to the differential control is that the method allows for fluctuations of the temperature, thereby not fixing it to a constant value. In each integration step it is insured that the T is corrected to a value more close to T_0.

- In the case of a *Stochastic Thermostat*, all or a subset of the degrees of freedom of the system are subject to collisions with "virtual" particles.

- The *Integral Thermostat* is also often called extended system method as it introduces additional degrees of freedom into the system's Hamiltonian for which equations of motion can be derived. They are integrated in line with the equations for the spatial coordinates and momentum. The idea of the method is to reduce the effect of an external system acting as heat reservoir to keep the temperature of the system constant, to one additional degree of freedom.

The Constant-Pressure Constant-Enthalpy Ensemble:

In order to control the pressure in an MD simulation cell, it is necessary to allow for volume variations. A simple picture for a constant pressure system is a box the walls of

which are coupled to a piston which controls the pressure. In contrast to the case where the temperature is controlled, no coupling to the dynamics of the particles (timescales) is performed but the length scales of the system will be modified. There are different algorithms for a constant pressure ensemble:

- *The Proportional Barostat*
- *The Integral Barostat*

1.1.3.Other Potentials in MD Simulations

Like we have already known a molecular dynamics simulation requires the definition of a potential function, or a description of the terms by which the particles in the simulation will interact, the, also called, force field. Potentials may be defined at many levels of physical accuracy; those most commonly used in chemistry are based on molecular mechanics (which we have seen before) and embody a classical treatment of particle-particle interactions that can reproduce structural and conformationals changes but usually cannot reproduce chemical reactions. When finer levels of detail are required, potentials based on quantum mechanics are used; some techniques attempt to create hybrid classical/quantum potentials where the bulk of the system is treated classically but a small region is treated as a quantum system, usually undergoing a chemical transformation.

Empirical potentials (Classical MD)

These empirical potentials that are frequently called force fields are the potentials which used in the classical molecular dynamics. Empirical potentials represent quantum-mechanical effects in a limited way through ad-hoc functional approximations. These potentials contain free parameters such as atomic charge, Van der Waals parameters reflecting estimates of atomic radius, and equilibrium bond length, angle, and dihedral; these are obtained by fitting against detailed electronic calculations (quantum chemical simulations) or experimental physical properties such as elastic constants, lattice parameters and spectroscopic measurements. These are the potentials explained in the previous section.

Semi-empirical potentials (QM MD)

Semi-empirical potentials make use of the matrix representation from quantum mechanics. However, the values of the matrix elements are found through empirical formulae that estimate the degree of overlap of specific atomic orbital. The matrix is then diagonalized to determine the occupancy of the different atomic orbital, and empirical formulae are used once again to determine the energy contributions of the orbital. There are a wide variety of semi-empirical potentials, known as tight-binding potentials, which vary according to the atoms being modelled.

Ab-initio methods

Compared to classical potential function, which is represented by empirical functions, the properties of the system in *ab-intio* calculations are calculated by the wave-functions for electrons moving around the nucleus of atoms. This calculation is usually made "locally", i.e., for nuclei in the close neighbourhood of the reaction coordinate. Although various approximations may be used, these are based on theoretical considerations, not on empirical fitting. *Ab-Initio* produces a large amount of information that is not available from the empirical methods, such as density of states information. Of course, the computational price paid is high. A significant advantage of using *ab-initio* methods is the ability to study reactions that involved breakage or formation of covalent bonds,

this would correspond to multiple electronic states. Classical molecular dynamics is unable to simulate breakage and formation of covalent bonds, however, in recent year's techniques such as thermodynamic integration and ghost particles have been introduced to overcome these limitations. The success however remains limited.

A popular package for *ab-initio* molecular dynamics is the Car-Parinello Molecular Dynamics (CPMD) package based on the density fluctuation theory.

Hybrid QM/MM

QM (quantum-mechanical) methods are very powerful however they are computationally expensive, while the MM (classical or molecular mechanics) methods are fast but suffer from several limitations (require extensive parameterization; energy estimates obtained are not very accurate; cannot be used to simulate reactions where covalent bonds are broken/formed; and are limited in their abilities for providing accurate details regarding the chemical environment). A new class of method has emerged that combines the good points of QM (accuracy) and MM (speed) calculations. These methods are known as mixed or hybrid quantum-mechanical and molecular mechanics methods (hybrid QM/MM).

The most important advantage of hybrid QM/MM methods is the speed. The cost of doing classical molecular dynamics (MM) in the most straight forward case scales $O(n^2)$, where n is the number of atoms in the system. This is mainly due to electrostatic interactions term (every particle interacts with everything else). However, use of cutoff radius, periodic pair-list updates and more recently the variations of the particle-mesh Ewald's (PME) method has reduced this between O(n) to $O(n^2)$. In other words, if a system with twice many atoms is simulated then it would take between twice to four times as much computing power. To overcome the limitation, a small part of the system is treated quantum-mechanically (typically active-site of an enzyme) and the remaining system is treated classically.

1.1.4.Coarse-graining and reduced representations

At the other end of the detail scale are coarse-grained and lattice models. Instead of explicitly representing every atom of the system, one uses "pseudo-atoms" to represent groups of atoms. MD simulations on very large systems may require such large computer resources that they cannot easily be studied by traditional all-atom methods. Similarly, simulations of processes on long timescales (beyond about 1 microsecond) are prohibitively expensive, because they require so many timesteps. In these cases, one can sometimes tackle the problem by using reduced representations, which are also called coarse-grained models.

Very coarse-grained models have been used successfully to examine a wide range of questions in structural biology. Examples of applications of coarse-graining in biophysics:

- Protein folding studies are often carried out using a single (or a few) pseudo-atoms per amino acid.
- DNA supercoiling has been investigated using 1-3 pseudo-atoms per basepair, and at even lower resolution.
- …

1.1.5.Limitations

Molecular dynamics is a very powerful technique but has, of course, limitations. We quickly examine the most important of them:

Use of classical forces:

The most of the abstract and the mathematical concepts that we studied on this work are applicable to all the model that we use; ab initio, semiempirical, based on potentials. . .Anyway, it looks somewhat strange that we can use Newton's law to move atoms, when everybody knows that systems at the atom level obey quantum laws rather than classical laws. Then, Schrödinger's equation is the one to be followed.

We have a simple test of the validity of the classical approximation: the de Broglie thermal wavelength, defined as:

$$\Lambda = \sqrt{\frac{2\pi\hbar^2}{Mk_BT}} \qquad (25)$$

where **M** is the atomic mass and **T** the temperature. The classical approximation is justified if: $\Lambda << a$, where **a** is the mean nearest neighbour separation. The classical approximation is poor for very light systems such as H_2, He or Ne.

Moreover, quantum effects become important in any system when T is sufficiently low. The drops in the specific heat of crystals below the Debye temperature, or the anomalous behaviour of the thermal expansion coefficient, are well known examples of measurable quantum effects in solids.

Molecular dynamics results should be interpreted with caution all these regions and we must thing about using more accurately and heavy models.

Realism of forces:

In molecular dynamics, atoms interact with each other. These interactions originate forces which act upon atoms, and atoms move under the action of these instantaneous forces. As the atoms move, their relative positions change and forces change as well.

The essential ingredient containing the physics is therefore constituted by the forces. A simulation is realistic (that is, it mimics the behaviour of the real system) only to the extent that interatomic forces are similar to those that real atoms (or, more exactly, nuclei) would experience when arranged in the same configuration.

Forces are usually obtained as the gradient of a potential energy function, depending on the positions of the particles. The realism of the simulation therefore depends on the ability of the potential chosen to reproduce the behaviour of the material under the conditions at which the simulation is run.

The problem of selecting or constructing potentials may be resumed with the basic concepts:

- There is not a set of forces that can be used for all situations.
- More accurately forces uses to use boundary conditions that does less extrapolable its results.
- Finally, the propagation of errors and the theory of perturbation can do high-precision forces less accurately than lesser-ones.

Time and size limitations:

Typical molecular dynamics simulations can be performed on systems containing thousand or, perhaps, millions of atoms and for simulation times ranging from a few picoseconds to hundreds of nanoseconds. While these numbers are certainly respectable, it may happen to run into conditions where time and/or size limitations become important.

A simulation is “safe” from the point of view of its duration when the simulation time is much longer than the relaxation time of the quantities we are interested in. However, different properties have different relaxation times. In particular, systems tend

to become slow in the proximity of phase transitions with variable sized integration step, and it is not uncommon to find cases where the relaxation time of a physical property is orders of magnitude larger than times achievable by simulation.

A limited system size can also constitute a problem. In this case one has to compare the size of the molecular dynamics cell with the correlation lengths of the spatial correlation functions of interest. Again, correlation lengths may increase or even diverge in proximity of phase transitions, and the results are no longer reliable when they become comparable with the box length.

1.2. AMBER

AMBER is the collective name for a suite of programs that allow users to carry out molecular dynamics simulations, particularly on biomolecules. None of the individual programs carries this name, but the various parts work reasonably well together, and provide a powerful framework for many common calculations. The term *amber* is also sometimes used to refer to the empirical force fields that are implemented here.

The AMBER suite of programs was developed by Peter A. Kollman and colleagues at the University of California San Francisco (UCSF). See http://amber.scripps.edu for more information.

AMBER consists of about 50 programs; the major programs to the amber package are as follows:

- Preparatory programs: Leap and Antechamber
- Simulation programs: Sander and others (pmemd and nmode)
- Analysis programs: Ptraj and others (mm-pbsa)

Understanding where to begin in AMBER is primarily a problem of managing the flow of information in this package. You first need to understand what information is needed by the simulation programs (sander, pmemd and nmode). One needs to know where it comes from, and how it gets into the form that the energy programs require.

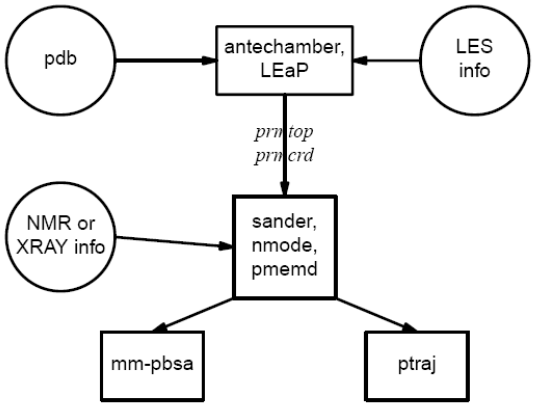

Figure 4: Basic information flow in Amber

Information that all the simulation programs need:

(1) Cartesian coordinates for each atom in the system. These usually come from X-ray crystallography, NMR spectroscopy, or model-building. They should be in Protein Databank (PDB) or Tripos "mol2" format. The program Leap provides a platform for carrying out many of these modelling tasks, but users may wish to consider other programs as well.

(2) "Topology": connectivity, atom names, atom types, residue names, and charges. This information comes from the database, which is found in the

`amber9/dat/leap/prep` directory. It contains topology for the standard amino acids as well as N- and C-terminal charged amino acids, DNA, RNA, and common sugars. The database contains default internal coordinates for these monomer units, but coordinate information is usually obtained from PDB files. Topology information for other molecules (not found in the standard database) is kept in user-generated "residue files", which are generally created using antechamber.

(3) Force field: Parameters for all of the bonds, angles, dihedrals, and atom types in the system. The standard parameters for several force fields are found in the `amber9/dat/leap/parm` directory. These files may be used "as is" for proteins and nucleic acids, or users may prepare their own files that contain modifications to the standard force fields.

(4) Commands: The user specifies the procedural options and state parameters desired. These are specified in the input files (usually called mdin) to the sander, pmemd, or nmode programs.

1.2.1.Antechamber

This program suite automates the process of developing force field descriptors for most organic molecules. It starts with structures (usually in PDB format), and generates files that can be read into Leap for use in molecular modelling. The Antechamber is designed to be used with the "general Amber force field (GAFF)". This force field has been specifically designed to cover most pharmaceutical molecules and is compatible with the traditional AMBER force fields for proteins and nucleic acids in such a way that the two can be mixed during a simulation.

Antechamber: This is the most important program in the package. It can perform many file conversions, and can also assign atomic charges and atom types. As required by the input, antechamber executes the following programs: *divcon*, *atomtype*, *am1bcc*, *bondtype*, *espgen*, *respgen* and *prepgen*. It may also generate a lot of intermediate files. If there is a problem with antechamber, you may want to run other individual programs that the Antechamber suite also contains.

The Antechamber tool set is designed to allow the rapid generation of topology files for use with the amber simulation programs. We will allow antechamber to assign our atom types and parameters automatically and also calculate a set of point charges for us using GAFF. With Antechamber, one may solve the following problems:

- Automatically identify bond and atom types
- Judge atomic equivalence
- Generate residue topology files
- Find missing force field parameters and supply reasonable suggestions

You should note, however, that Antechamber is not a replacement for due diligence. You should always closely examine the atom types that Antechamber assigns and verify to yourself that the choices are reasonable.

Like input files we usually use the **pdb file**. The PDB (Protein Data Bank, from Research Collaboratory for Structural Bioinformatics) format is the standard file format for the XYZ coordinates of atoms in a molecule. Here are a few lines from the PDB file for the enzyme EPSP Synthase structure and shikimate-3-phosphate (S3P) structure (from aroa_pep_s3p.pdb) directly above:

LINE DEFINITION	***ATOM NO.***	***ATOM NAME***	***RESIDUE NAME***	***RESIDUE NO.***	***X***	***Y***	***Y***	***ATOM***
...	...	...	...	...	...	...	...	
ATOM	3280	CB	VAL	423	12.680	15.308	38.877	C
ATOM	3281	CG1	VAL	423	12.019	14.746	40.130	C

```
ATOM      3282   CG2    VAL          423        11.706   16.154   38.082   C
ATOM      3283   N      GLY          424        15.775   14.601   39.144   N
ATOM      3284   CA     GLY          424        16.765   13.653   39.636   C
ATOM      3285   C      GLY          424        17.919   14.350   40.330   C
TER
HETATM    3286   P1     S3P          425        -1.737   31.393   17.748   P
HETATM    3287   C1     S3P          425         3.205   31.372   17.362   C
HETATM    3288   O1     S3P          425        -0.060   31.110   17.595   O
HETATM    3288   C2     S3P          425         2.133   32.143   17.666   C
...       ...    ...    ...          ...        ...      ...      ...
```

Another typical input file is **mol2**. A Tripos Mol2 file (.mol2) is a complete, portable representation of a SYBYL molecule. It is a file which contains all the information needed to reconstruct a SYBYL molecule. An example of this format is the next about the molecule phosphoenolpyruvate (PEP):

```
# MOL2 TOPOLOGY BY PRODRG
# WARNING: THIS FILE IS BUILT FROM A GROMOS TOPOLOGY
# AND MAY NEED FURTHER OPTIMISATION (E.G. AROMATICITY
@<TRIPOS>MOLECULE
                PEP
   12   11    1
                SMALL
         USER_CHARGES

      PRODRG MOLECULE
@<TRIPOS>ATOM
1  O2P          2.231    24.500    15.918   O.3       1 PEP       -1.086
2  P            2.390    25.780    16.710   P.3       1 PEP        1.183
3  O3P          3.345    26.658    15.934   O.3       1 PEP       -1.087
4  O1P          1.120    26.579    16.506   O.2       1 PEP       -0.793
5  O2           2.836    25.711    18.300   O.3       1 PEP       -0.223
6  C2           1.860    26.049    19.519   C.2       1 PEP        0.329
7  C3           0.577    26.355    19.243   C.2       1 PEP       -0.076
8  C1           2.680    25.919    20.809   C.2       1 PEP        0.329
9  O1           2.172    25.795    21.805   O.co2     1 PEP       -0.788
10 O2*          3.831    25.933    20.896   O.co2     1 PEP       -0.788
11 H001         0.261    26.378    18.294   H         1 PEP        0.000
12 H002        -0.060    26.559    19.987   H         1 PEP        0.000
@<TRIPOS>BOND
    1    1    2 1
    2    2    3 1
    3    2    4 2
    4    2    5 1
    5    5    6 1
    6    6    7 2
    7    6    8 1
    8    8    9 2
    9    8   10 2
   10    7   11 1
   11    7   12 1
@<TRIPOS>SUBSTRUCTURE
  1  PEP    1
```

We could easily have used any number of other supported formats including Gaussian Z-Matrix [gzmat], Gaussian Output [gout], MDL [mdl], amber Restart [rst]…

The file that we are really interested in, and the reason we will run Antechamber in the first place, is the **prepi files**. This contains the definition of our organic molecules including all of the charges and atom types that we will load into Leap to when creating our prmtop and inpcrd files. With this we can add new residues to the standard amber residue database, create new databases, or to create new residues as individual LINK-readable files. A residue is the basic molecular unit of the AMBER simulation package. It is typically an amino acid or nucleic acid unit, but could be a prosthetic group, a small molecule, or a single ion.

Let's take a quick look at the prepi file for PEP:

```
0    0    2
This is a remark line
molecule.res
PEP  INT    0
CORRECT  OMIT DU   BEG
0.0
1 DUMM   DU    M    0   -1   -2   0.0000    0.0000    0.0000  0.000
2 DUMM   DU    M    1    0   -1   1.4490    0.0000    0.0000  0.000
3 DUMM   DU    M    2    1    0   1.5220  111.1000    0.0000  0.000
4 O3P    o     M    3    2    1   1.3350  116.6000  180.0000 -0.5200
5 P      p5    M    4    3    2   1.0100  119.8000    0.0000  0.2480
6 O1P    o     E    5    4    3   1.4490  121.9000  180.0000  0.2140
7 O2P    o     E    5    4    3   1.5250  111.1000   60.0000  0.0380
8 O2     os    M    5    4    3   1.5100  115.0000  180.0000  0.0110
9 C2     ce    M    8    5    4   1.4000  120.0000  180.0000 -0.0110
10C3     c2    B    9    8    5   1.4000  120.0000  180.0000  0.0040
...

IMPROPER
C1  C3   C2  O2
C2  H001 C3  H002
C2  O2*  C1  O1

LOOP

DONE
STOP
```

The file contains, in internal coordinates, the 3 dimensional structure of the PEP molecule (final column), the atom number (column 1), its name (column 2) and its atom type (column 3) as well as the charge on each atom. It also specifies loops and improper torsions. This file does not, however, contain any parameters. The GAFF parameters are all defined in `$AMBERHOME/dat/leap/parm/gaff.dat`. The other thing you should notice here is that all of the GAFF atom types are in lower case. This is the mechanism by which the GAFF force field is kept independent of the macromolecular AMBER force fields. All of the traditional AMBER force fields use uppercase atom types. In this way the GAFF and traditional force fields can be mixed in the same calculation.

Parmchk: We can use it to test if all required parameters are available. This program will produce a file called **frcmod** (force field parameter modification file specification). This is a parameter file that can be loaded into Leap in order to add missing parameters. If it can, antechamber will fill in these missing parameters by analogy to a similar parameter. One should checks these parameters carefully before running a simulation. It is hope that as GAFF is developed so the number of missing parameters will decrease. Let's look at frcmod file for PEP:

```
remark goes here
MASS

BOND
os-ce  392.60   1.357       same as c2-os

ANGLE
p5-os-ce   56.436     119.695   Calculated with empirical approach
os-ce-c2   71.200     121.430   same as c2-c2-os
os-ce-c    71.200     121.430   same as os-ce-c2

DIHE
```

```
p5-os-ce-c2   1    1.050      180.000    2.000      same as X -c2-os-X
p5-os-ce-c    1    1.050      180.000    2.000      same as X -c2-os-X

IMPROPER
c -c2-ce-os         1.1       180.0        2.0      Using default value
ce-ha-c2-ha         1.1       180.0        2.0      Using default value
ce-o -c -o          1.1       180.0        2.0      General improper
torsional angle (1 general atom type)

NONBON
```

One can see that there were some missing parameters, like one bond or three angles. Later on one could take a look in xleap and to see what atoms these correspond to. You shall assume that the parameters Antechamber has suggested for are acceptable. Ideally one should test these parameters (by comparing to *ab initio* calculations for example) to ensure they are reasonable.

1.2.2.LEaP

LEaP is the generic name given to the programs teLeap and xaLeap, which are generally run *via* the *tleap* and *xleap* shell scripts. These two programs share a common command language but the xleap program has been enhanced through the addition of an X-windows graphical user interface. Leap is a program that reads in force field, topology and coordinate information and produces files necessary for MD calculations (*i.e.* minimisation, molecular dynamics, analysis ...).

AMBER has a residue topology database to describe all amino acid residues as well as nucleic acid residues. So, for proteins one just needs load in Leap one of the AMBER force field (ff99 or ff03, for example) and then the pdb file, which contains the coordinate information of the system. If the system also contains organic molecules you will need the files generated with antechamber: prepin and frcmod, these will be the Leap input files. You should load them as well as the GAFF force field to include the organic molecules to the AMBER database.

When one have loaded all the information needed, with Leap we can modify our system. For instance we can add ions to neutralize the system, solvate the system, add the hydrogen missing, create double or single bonds….Leap also contains the command "edit" to visualise graphically the structures.

The purpose of Leap is to generate the output files: prmtop and inpcrd, which will be the input files to run a simulation with Sander.

prmtop: There is a parameter/topology file. It defines the connectivity and parameters for a current model such as: number of atoms, distinct atom types, bonds containing hydrogen and not containing hydrogen, angles, dihedrals, residues, constraint bonds and many more. This information is static, or in other words, it doesn't change during the simulation. So in every simulation that will be running it will be needed this information.

inpcrd: This is the file containing all information about coordinates and optionally about box coordinates and velocities. The data are not static and evolve during the simulation run (although the file remains unaltered).

1.2.3.Sander

Sander is the basic energy minimizer and molecular dynamics program. The acronym stands for "Simulated Annealing with NMR-Derived Energy Restraints", but this module is used for a variety of simulations that have nothing to do with NMR refinement. This program relaxes the structure by iteratively moving the atoms down the energy gradient until a sufficiently low average gradient is obtained. The molecular

dynamics portion generates configurations of the system by integrating Newtonian equations of motion. MD samples more conformational space than minimization, and allows the structure to cross over small potential energy barriers. Configurations may be saved at regular intervals during the simulation for later analysis as well; basic free energy calculations using thermodynamic integration may be performed.

More elaborate conformational searching and modelling MD studies can also be carried out using the SANDER module. This allows a variety of constraints to be added to the basic force field, and has been designed especially for the types of calculations involved in NMR structure refinement.

The basic usage for *sander* is as follows:

```
sander [-O] -i mdin -o mdout -p prmtop -c inpcrd -r restrt
 [-ref refc] [-x mdcrd] [-v mdvel] [-e mden] [-inf mdinfo]
```

- Arguments in []'s are optional
- If an argument is not specified, the default name will be used.
- **-O** overwrite all output files (the default behaviour is to quit if any output files already exist)
- **-i** the name of the input file (which describes the simulation options), mdin by default.
- **-o** the name of the output file, mdout by default.
- **-p** the parameter/topology file, prmtop by default.
- **-c** the set of initial coordinates for this run, inpcrd by default. Also can be the restrt file from the previous step.
- **-r** the final set of coordinates from this MD or minimisation run, restrt by default.
- **-ref** reference coordinates for positional restraints, if this option is specified in the input file, refc by default.
- **-x** the molecular dynamics trajectory file (if running MD), mdcrd by default.
- **-v** the molecular dynamics velocities file (if running MD), mdvel by default.
- **-e** a summary file of the energies (if running MD), mden by default.
- **-inf** a summary file written every time energy information is printed in the output file for the current step of the minimisation of MD, useful for checking on the progress of a simulation, mdinfo by default.

When a simulation is running by sander the information is saved in an **output** file. This generally looks like the next:

```
 NSTEP =      600   TIME(PS) =     100.600  TEMP(K) =   303.15  PRESS =   157.0
 Etot   =    -37714.6399  EKtot   =     14367.4259  EPtot      =    -52082.0658
 BOND   =     10030.6168  ANGLE   =        35.8883  DIHED      =        17.2064
 1-4 NB =         7.5556  1-4 EEL =       239.5419  VDWAALS    =     10971.9383
 EELEC  =    -73384.8132  EHBOND  =         0.0000  RESTRAINT  =         0.0000
 EKCMT  =      4815.6322  VIRIAL  =      4290.1703  VOLUME     =    154970.3717
                                                   Density    =         1.0264
 Ewald error estimate:   0.1858E-03
 ------------------------------------------------------------------------------

 NSTEP =      800   TIME(PS) =     100.800  TEMP(K) =   303.66  PRESS =  -155.6
 Etot   =    -37894.3085  EKtot   =     14392.0020  EPtot      =    -52286.3106
 BOND   =      9973.4877  ANGLE   =        39.6500  DIHED      =        17.3151
 1-4 NB =         6.5171  1-4 EEL =       239.3089  VDWAALS    =     10841.2700
 EELEC  =    -73403.8594  EHBOND  =         0.0000  RESTRAINT  =         0.0000
 EKCMT  =      4789.1139  VIRIAL  =      5310.0846  VOLUME     =    155098.8248
                                                   Density    =         1.0255
 Ewald error estimate:   0.4175E-04
 ------------------------------------------------------------------------------
```

The output above contains the information about: current number of the step, time, temperature, pressure, volume, density, total, potential and kinetics energies, energies of

the bonds, angles, dihedrals, electrostatics, Van der Waals, the Ewald error value and so on. All of the information could be extracted for further analyses.

All the simulations that are running by sander for one determined system will need the prmtop file created by Leap. In the first simulation run will be needed also the inpcrd file. This first simulation will generate another coordinate file, **restrt** file, with the new information about the coordinates and it could be used in the next simulation. This file is very similar to the inpcrd, there is just a recalculation of the coordinate information.

During every MD run (not for the minimisations) sander will generate a trajectory file, **mdcrd**. It will be utilized later to analyse the results of the simulation, because in it is written all the information about the dynamics. There are also the possibilities to write the information about the velocities (mdvel) or the energies (mden).

Besides the serial version of sander, the parallel version of the program exists and it allows to utilize more than one processor shortening the time needed for calculations

1.2.4.Ptraj

It is a general purpose utility for analysing and processing trajectory or coordinates files created from MD simulations, including superposition, extractions of coordinates, calculation of bond/angle/dihedral values, calculation of RMSd, atomic positional fluctuations, time-correlation functions, analysis of hydrogen bonds, and many more. The same executable, when named rdparm (from which the program evolved), can examine and modify prmtop files. To use the program it is necessary to:

(1) Read in a parameter/topology file

The information in these files is used to setup the global *state* which gives information about the number of atoms, residues, atom names, residue names, residue boundaries, *etc*. It is very important that the input coordinates must match exactly the order specified by the state. In other words, when reading, for instance, a pdb file, the atom order must correspond exactly to that of the parameter/topology information; in the pdb the names/residues are ignored and only the coordinates are read.

(2) Set up a list of input coordinate files

This is done with the `trajin` command which specifies the name of a file for reading the coordinates. This could be a trajectory file as well as a file containing just a single snapshot frame of the system.

(3) Optionally specify an output trajectory file

This is done with the `trajout` command. Trajectories can currently be written in Amber trajectory files (mdcrd), Amber restrt files, pdb file, etc

(4) Specify a list of actions

There are a variety of coordinate analysis/manipulation actions provided. The results can be saved in an output file to plot it later.

1.3. EPSP Synthase

The enzyme 5-enolpyruvylshikimate 3-phosphate (EPSP) synthase (EC 2.5.1.19) is the sixth enzyme on the shikimate pathway, which is essential for the synthesis of aromatic amino acids and of almost all other aromatic compounds in algae, higher plants, bacteria, and fungi, as well as in apicomplexan parasites. Because the shikimate pathway is absent from mammals, EPSP synthase is an attractive target for the development of new antimicrobial agents effective against bacterial, parasitical, and fungal pathogens. A valuable lead compound in the search for new drugs and herbicides is glyphosate, which has proven as potent and specific inhibitor of EPSP synthase.

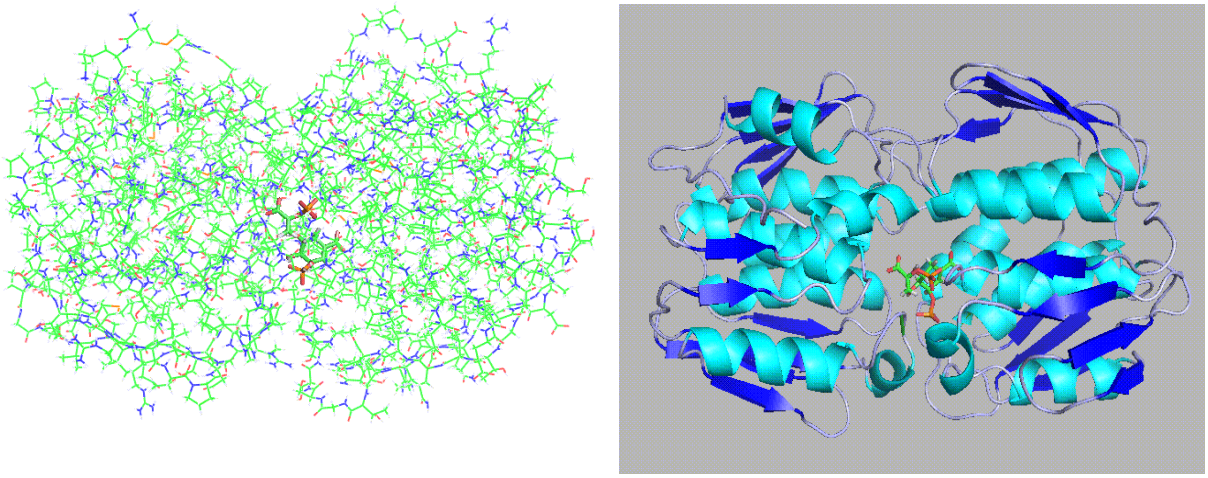

Figure 5: EPSP Synthase with S3P and PEP line (left) and the secondary structure representation (right)

EPSP synthase is a transferase which catalyzes the transfer of the enolpyruvyl moiety from phosphoenol pyruvate (PEP) to shikimate-3-phosphate (S3P) forming the products EPSP and inorganic phosphate (Figure 6). The reaction is chemically unusual because it inhibits EPSP synthase in a slowly reversible reaction, which is competitive versus PEP and uncompetitive versus S3P.

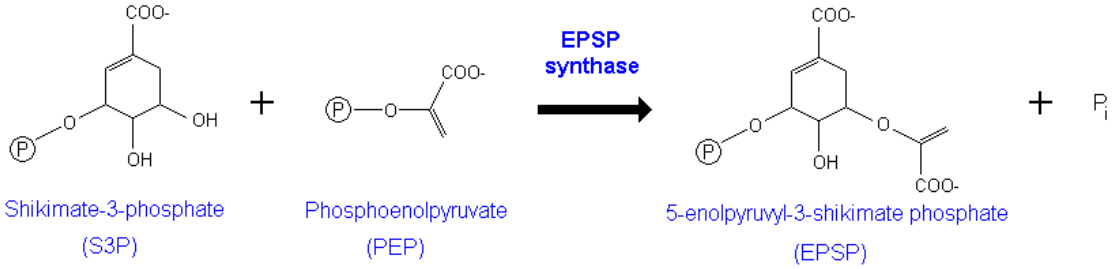

Figure 6: Reaction that EPSP Synthase catalyzed

The monomeric enzyme EPSP synthase (*Mr* 46,000) folds into two similar domains (Figure 5 and 7), each comprising three copies of a βαβαββ-folding unit. In between are two crossover chain segments that hinge the nearly topologically symmetrical domains together and allow conformational changes necessary for substrate conversion. Each domain structure has been described as a "mushroom button" where the inner helices form the stem and the outer beta strands act as the cap. Each stem base faces the other, and the parallel helices extend with the positive end of their macro dipoles situated at this interface. It has been postulated that part of the binding mechanism is facilitated by a helical macro dipole effect. Since the substrates are heavily anionic, it is reasonable to assume that the localized positive core of the active site would assist in anion binding.

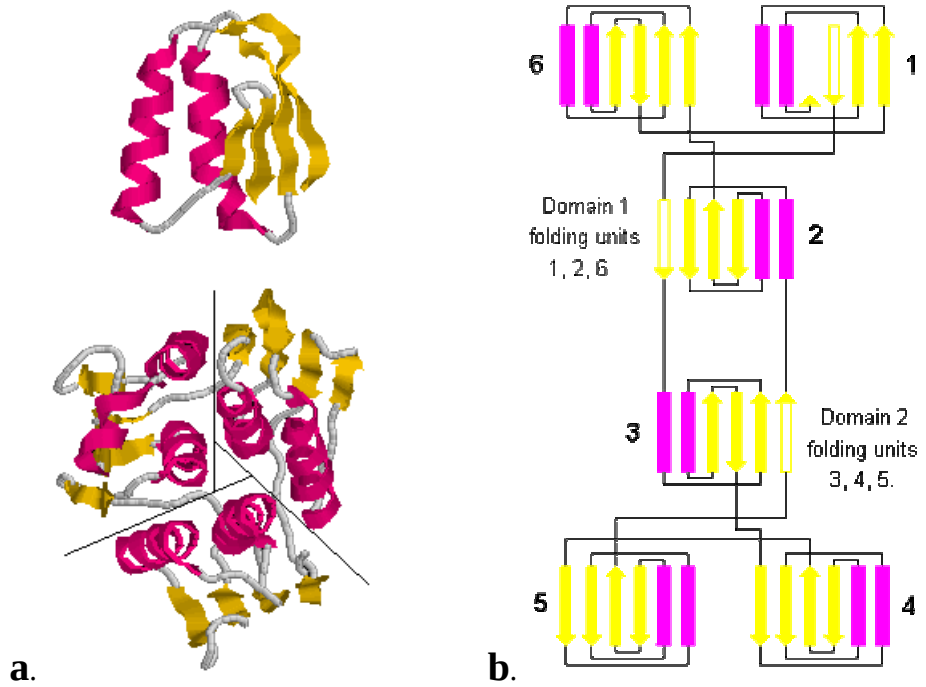

Figure 7: a. *Top:* One ßaßaßß-folding unit. *Bottom:* N-terminal domain from bottom with lines separating the three folding units. **b.** Stallings' nomenclature for subdomain tertiary structures. Yellow arrows represent beta-strands, with open arrows indicating a strand that is not directly connected to other secondary structures within its own folding unit, and magenta rectangles represent alpha-helices. Domain 1 is composed of folding units 1, 2, and 6, and Domain 2 is composed of folding units 3, 4, and 5. Both carboxy and amino termini are contained within folding unit 1. Letter assignments for each secondary structure within a folding unit begin at the n-terminal end of the folding unit and are lettered in alphabetical order consecutively to the c-terminal end. (From Stallings et al. (9)).

EPSPS has two possible conformations: open conformation, when the enzyme is an "apo" form, and closed conformation, when the enzyme is liganded with S3P and PEP:

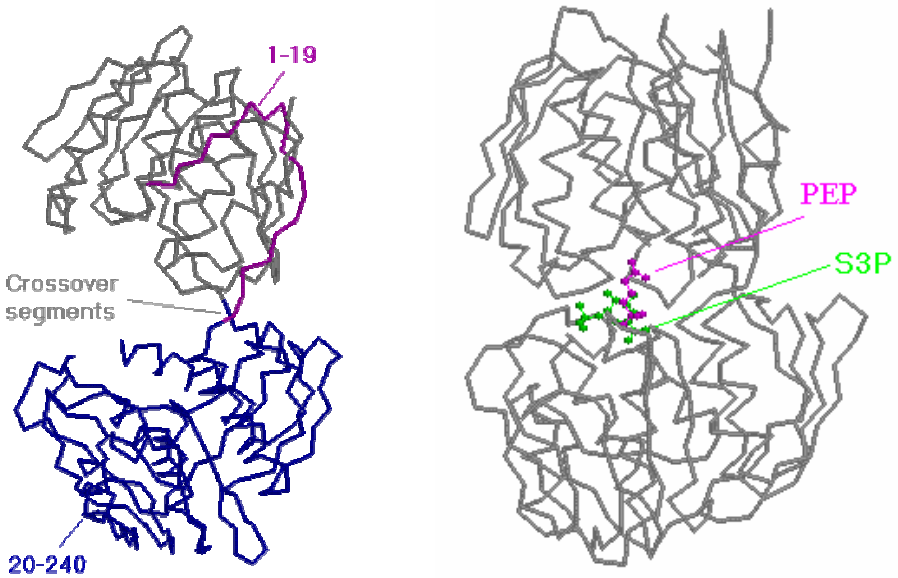

Figure 8: *Left:* Open conformation with C-terminal domain, grey and magenta, and N-terminal domain, blue. *Right:* Closed conformation with bound substrate, S3P and PEP.

The only other enzyme known to exhibit this architecture is the mechanistic homologue UDP-*N*-acetylglucosamine enolpyruvyl transferase (MurA, EC 2.5.1.7), which catalyzes the transfer of the intact enolpyruvyl moiety of PEP to a sugar nucleotide. MurA is essential for the synthesis of the bacterial cell wall and is the target of the broad spectrum antibiotic fosfomycin. On sugar nucleotide binding, MurA undergoes large conformational changes leading to the formation of the active site.

Although EPSP synthase has been extensively studied over more than three decades, conclusions on the enzyme mechanism remained controversial.

2. OBJECTIVES

The main objectives of the carried project were as follows:

2.1.Hybrid QM/MM MD Simulation of the system containing EPSPS, PEP and S3P

With this setup we would like to run a hybrid QM/MM simulation. The QMMD region included S3P, PEP moieties and the side chains of the amino acids in the active site of EPSPS, and MM/MD was applied for the rest of the system. Hybrid QM/MM MD simulation allows to find a more realistic dynamic of the QM region as well as it can give the insight into direction of the chemical reactions in the enzyme active site. During the simulation we ought to analyse the conformations of and the positions of S3P, PEP and the enzyme active site before and during the catalysed reaction.

2.2.Comparison of Hybrid QM/MM MD with CMD of S3P and PEP in a water box

Because a hybrid QM/MM MD of the solved protein-ligand system is computationally very expensive, a small system consisted of S3P and PEP moieties solved in water box was prepared to test the parameters for these two molecules as calculated with Antechamber. The ligands (S3P and PEP) were placed in a water box and were set up as the QM region while water was treated with MM force field parameters. Such a simulation can be completed within a few days and it can testify the stability of the studied structures. So, we will use this simulation to check the charges, geometry and other characteristics of our molecules in a QM simulation.

Parallely, the same system was studied under conditions of the classical MD simulation to be able to compare both methods and to check which of the method used is more accurate.

Each of the simulations (Classical MD and QM/MM MD) was run with two different production runs (the last step of the simulation): with and without SHAKE algorithm to restrain the hydrogen bonds. We could observe the differences between them and determined which is more adequate.

2.3.Classical MD of EPSPS in unliganded and liganded states

EPSPS occurs in close and open states and the structural changes are induced by attachment of two small molecules. A classical MD simulation was carried on close structure of EPSPS but with the ligands removed. It will allow to check if the unliganded system goes to the open conformation, which is the natural conformation the enzyme apo-form. The classical MD simulation of liganded form EPSPS was conducted as a test for stability of the system and to asses the correctness of the parameters calculated for S3P and PEP with Antechamber. It was assumed that system stable during classical MD run will remain stable during QM/MM simulation.

3. METHODS

Software:

- Molecular dynamics simulations

The version 9 of the **AMBER** software suite, released in March, 2006, was used to conduct all of the computational work. Input files for simulation were prepared with Antechamber and LeaP along with GAFF and AMBER 1999 force fields. Serial and parallel versions of sander were employed to run the simulations and the data post-processing, like RMSd calculations or trajectory analyses, were done with ptraj program. More information about AMBER is on its website: http://amber.scripps.edu/. For new uses there are a number of tutorials (of varying level) available on the AMBER website: http://amber.scripps.edu/tutorial/index.html. Other source of information is the Amber manual of the current version: http://amber.scripps.edu/doc9/amber9.pdf.

- Structure and trajectory visualisation

To visualise the structures and the trajectories calculated during the simulations the program **PyMOL** version v0.99 was used. The website: http://pymol.sourceforge.net/ provides all the information that one can need, like tutorials or the manual, as well as download the current version.

- Plotting the results.

Results from trajectory analyses were plotted using **Grace** plotting program. Grace is an interactive graphic and command-line 2D plotting software, xmgrace is the current development of the program formally known as xmgr. Grace is a descendant of ACE/gr, also known as Xmgr, originally written by Paul Turner.

We utilized the version 5.1.19. To get more information visiting the Grace home page: http://plasma-gate.weizmann.ac.il/Grace/ is the best choice to start. There are available the Grace user's guide for the current version (version 5.1.21) (http://plasma-gate.weizmann.ac.il/Grace/doc/UsersGuide.html), an introductory tutorial (http://plasma-gate.weizmann.ac.il/Grace/doc/Tutorial.html), etc.

3.1.Classical MD of EPSPS in unliganded and liganded states

Two different simulations were run at the same time on almost similar structures. One of them involved the polypeptide chain alone, called "*aroa unliganded*", and the another one with the three molecules: the enzyme, S3P and PEP, named "*aroa liganded*". All simulations have been run in three steps. First, starting coordinates and the topology of each systemwere created with LEaP. These files provided the input information to run the simulation with sander in the next step followed by trajectory analyzes and visualization with ptraj, pymol and xmgrace.

Step 1. Preparing starting structures and generating missing parameters for ligands:

To start a simulation have to be generated the initial structure and set up the molecular topology/parameter and coordinate files necessary for performing minimisation or dynamics with *sander*. It can be done with Leap but before will be needed the files that it can read (usually a pdb file). So the first step is the building of residues and creates the starting coordinates. The input files of the biomolecules can be found in the Protein Data Bank. These files usually need some editing before it can be used by Amber. But in this case it was begun with the pdb file "aroa_pep_s3p.pdb" which contains the three biomolecules (AroA, S3P and PEP). This file has been hand over by Galina Kachalova, who determinated the structure by X-Ray diffraction.

The pdb file will be enough for the proteins, like our enzyme, because AMBER can read the standard amino acids. But this file will not be enough for the other

biomolecules, S3P and PEP. Many coenzymes or other organic molecules are not pre-defined in the Amber database and so are considered to be non-standard residues. It is necessary to provide structural information and force field parameters for all of the non-standard residues that will be present in the simulation before can be created the Sander input files. For them you will need Antechamber.

For the aroa unliganded you need just the enzyme structure, so you have to remove the other molecules (S3P, PEP and water molecules) from the pdb file. LEaP will be able to read this file. This simulation will be running without the waters molecules of the crystallization (the water molecules that there are in the pdb file) and without explicit water (no salvation the system). In explicit water the system would be too large and the simulation time would be several weeks. So, you can run the simulation using implicit water, which is less realistic but shorter.

For the aroa liganded it will be more complicated. From the initial pdb file you should just remove the water molecules. You also should ensure that there is a TER card between the protein and the ligand (if thare are ligands). Otherwise xleap will assume that these are part of the same chain.

But LEaP can't read the S3P and PEP from this file, so you can make use of the Antechamber tools, in order to create an input file that can be read by LEaP so that you can create prmtop and inpcrd files for simulations of organic molecules. You shall use *antechamber* to assign atom types to this molecule and also calculate a set of point charges for us. First you should create two different files from S3P and PEP, because sometimes *antechamber* can't read pdb files. You can copy pdb coordinates from the original pdb file and with PRODRG we will create a S3P and PEP mol2 files. These can be read for *antechamber* and it can be generate prepin files from the mol2 files. These are the files that you really need, because they contain the definition of the molecules (S3P and PEP) including all of the charges and atom types that it will be loaded into Leap when creating the prmtop and inpcrd files. These files do not, however, contain any parameters. You also need the frcmod files which can be created with *parmchk* (another program from Antechamber) to add all the missing parameters to the molecules structures when you load it into Leap.

You now have everything you need to load S3P and PEP as a unit in Leap.

Step 2: Running LEaP to generate the parameter and topology files

First, decision must be done about the force field to be used. There is a lot of information about the different force field in the AMBER users' manual. The force field *ff99* were applied to describe the polypeptide chain and the force field *gaff* for the ligands. The latter is compatible with the traditional AMBER force fields in such a way that the two can be mixed during a simulation.

Second, coordinate files containing either apo or liganded form of EPSPS in PDB format together with parameters files for PEP and S3P were loaded. Since the net charge of simulated systems must be neutral counterions have to be added. Method implemented in *xleap* works by constructing a Coulombic potential on a 1.0 angstrom grid and then placing counterions (Na+ ions) one at a time at the points of lowest/highest electrostatic potential. It is necessery to check the structures created with LEaP, using command *edit*.

After all, command *saveamberparm* creates topology and coordinate files for the apo and liganded forms of EPSPS.

Whole preparation process can be done automatically by calling from the shell:

```
$AMBERHOME/exe/xleap –f xleap_aroa_lig.in &
  $AMBERHOME/exe/xleap –f xleap_aroa.in &
```

where the file “xleap_aroa_lig.in” is:

```
source leaprc.ff99
source leaprc.gaff
loadamberprep s3p.prepi
loadamberparams s3p.frcmod
loadamberprep pep.prepi
loadamberparams pep.frcmod
aroa_liganded = loadpdb aroa_pep_s3p.pdb
addions aroa_liganded Na+ 0
saveamberparm aroa_liganded aroa_liganded.top
aroa_liganded.crd
quit
```

and the file “xleap_aroa.in” is:

```
source leaprc.ff99
aroa = loadpdb aroa.pdb
addions aroa Na+ 0
saveamberparm aroa aroa.top aroa.crd
quit
```

Step 3: Run MD simulation

A typical MD simulation consists of the certain steps. This usually is realized in three stages: minimization, equilibration and long production run. Depends on the accuracy wanted and on the characteristics of the investigated system it can be needed to do more than one minimization or equilibration. One also should choose the adequate parameters in every stage in order to run the correct simulation.

- Minimization

Each system must be minimized prior to the MD run. The positions of the protein atoms were restrained (fix up) in order to remove any bad contacts that may lead to unstable molecular dynamics and a short (500 steps) minimization was run. Minimization algorithms (steepest descend and conjugate gradient move the structures towards the closest local minimum. It cannot cross transition states to reach lower minima but is enough to remove the largest strains in the system. Here is an input file for the minimization:

```
min.in
Minimisation of our complex
&cntrl
imin=1, maxcyc=500, ncyc=350,
cut=16, ntb=0, igb=1,
/
```

In total, 500 steps of minimization (maxcyc) with the first 350 being steepest descent (ncyc), the remainder the conjugate gradient (maxcyc-ncyc).A reasonably large cut off of 16 Å was used since this is not going to be a periodic simulation and we want to deal with our electrostatics accurately (ntb=0,cut=16). The GB (Generalized Born) model of Hawkins, Cramer and Truhlar (igb=1) was used as an implicit solvent model.

For instance, to run the minimization of aroa_liganded we simply execute the following:

```
$AMBERHOME/exe/sander -O -i min.in -o aroa_lig_min.out
 -c aroa_liganded.inpcrd -p aroa_liganded.prmtop –r
                  aroa_lig_min.rst
```

where *aroa_liganded.inpcrd/prmtop* are the coordinate and topology files created by LEaP. *aroa_lig_min.out* is the output file of the minimization and *aroa_lig_min.rst* is the new coordinate file, which we will use like input file in the next stage.

The progress can be monitored by tailing the output file:

```
tail -f aroa_lig_min.out
```

The result of the minimization can be viewed after converting the *aroa_lig_min.rst* file into PDB format and analysing it with PyMol:

```
ambpdb -p aroa_liganded.prmtop <aroa_lig_min.rst >
    aroa_lig_min.pdb
```

- Equilibration

In the next step both systems are to be heated and equilibrated. For the next step we will use this coordinate file as the starting structure for our MD simulation. This is the variables of the equilibration run:

```
md1.in
MD equilibration
&cntrl
imin=0, irest=0,nstlim=100000,dt=0.001, ntc=1,
ntpr=100, ntwx=100, cut=16, ntb=0, igb=1,
ntt=3, gamma_ln=1.0, tempi=0.0 , temp0=300.0,
/
```

We will run MD (imin=0) and this is not a restart (irest=0). Here I will use a time step of 1 fs and run for 100000 steps [100 ps] (dt = 0.001, nstlim=100000, ntc=1). We disable periodicity (ntb=0) and again set `igb=1` to use the Born implicit solvent model. A snapshot of the system was written to an output file every 100 steps and to a trajectory [mdcrd] file every 100 steps (ntpr=100,ntwx=100). The temperature control was based on Langevin dynamics approach with a collision frequency of 1 ps^{-1}. The systems were heated from 0Kto a target temperature of 300K (ntt=3, gamma_ln=1.0, tempi=0.0, temp0=300.0):

```
$AMBERHOME/exe/sander -O -i md1.in -o aroa_lig_md1.out
  -p aroa_liganded.prmtop -c aroa_lig_min.rst -r
     aroa_lig_md1.rst -x aroa_lig_md1.mdcrd &
```

- Production run

The final stage of the MD simulation was a production run. It was identical to the equilibration state described above.

Here is the input file:

```
md2.in
Production MD
&cntrl
imin=0, irest=1, ntx=5, nstlim=500000, dt=0.001,
ntc=1, ntpr=500, ntwx=500, cut=16, ntb=0, igb=1,
ntt=3, gamma_ln=1.0, tempi=300.0, temp0=300.0,
/
```

All settings were the same as before, excepting *ntx*=5, because it was started from an MD resrt file created during the equilibration stage. For the same reason *irest* was setup to 1. Also the number of steps were larger (nstlim=500000) giving simulation lengths of 500 ps. Now the information were written to the output files every 500 steps (ntpr=500, ntwx=500). The initial and final temperatures were 300 K (tempi=300.0, temp0=300.0) which means that the temperature of both systems should remain around 300 K.

Results analysis

When the simulation is finished results can be analyzed. The information contained in the output files can be processed:

```
./process_mdout.perl aroa_lig_md1.out aroa_lig_md2.out
```

The Perl script `process_mdout.perl` processed the output files of the MD simulation and generates files with the values obtained during the simulation about temperature, density, pressure, energy…vs. time.

To calculate other parameters you can use ptraj, another tool from AMBER package. After the trajectory files (mdcrd) from the simulation are loaded, the actions (calculations) have to be specified. Following is the file executed to analyse the results of aroa_liganded:

```
aroa_lig.ptraj
trajin aroa_lig_md1.mdcrd
trajin aroa_lig_md2.mdcrd
rms first :1-426 out whole_aroa_lig.rms
rms first out backbone_aroa_lig.rms @CA,N,O,C
rms first :S3P out s3p.rms
rms first :PEP out pep.rms
distance s3p-pep :S3P@O3 :PEP@O2 out s3p-pep.dat
atomicfluct out pep.apf :425
atomicfluct out s3p.apf :426
```

In this case it will be executed:

```
ptraj aroa_liganded.prmtop < aroa_lig.ptraj
```

Ptraj read the trajectory information from mdcrd files and calculated the RMSd values for the whole system, the backbone of the enzyme, S3P and PEP, the distance between two specified atoms in S3P and PEP, and the atomic fluctuations of the two ligands. Every action was saved in another file (like pep.rms or s3p.apf) which can be plotted with *xmgrace*.

3.2.Hybrid QM/MM MD & Classical MD S3P and PEP in a water box

Two simulations were run at the same time, one with the classical MD method and the other with the hybrid QM/MM MD method using the semi-empirical PM3 Hamiltonian. The system consists of the two ligand molecules, S3P and PEP, in a periodic box of TIP3P water containing 5286 water molecules (Figure 9). The simulation was performed in explicit water in order to make more realistic the movement.

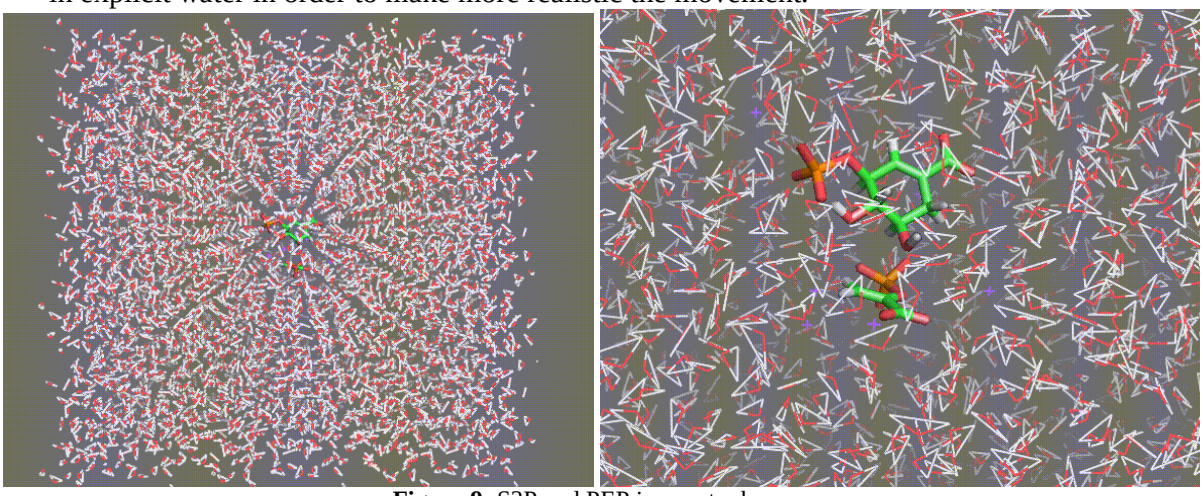

Figure 9: S3P and PEP in a water box

The first two steps, generating input coordinates and creating the topology and coordinate files with LEaP, were the same for both simulations, the differences started with the setup of the simulation run. In fact, the first step was also similar to that in the other simulation (see chapter 3.1). Additionally, S3P and PEP prepi and frcmod files created by Antechamber in the first simulation can be applied, too. The structures of the ligands were extracted from the PDB file of the EPSPS structure.

- Generating the parameter and topology files with LEaP

The prepi and frcmod files were loaded as well as the pdb file of the molecules into LEaP along with the force fields, in this case again ff.99 and gaff. The LEaP script file was as follows:

```
source leaprc.gaff
source leaprc.ff99
oadamberprep s3p.prepi
loadamberparams s3p.frcmod
loadamberprep pep.prepi
loadamberparams pep.frcmod
lig = loadpdb pep_s3p.pdb
solvatebox lig TIP3PBOX 30 1.8
addions lig Na+ 0
saveamberparm lig lig.prmtop lig.inpcrd
```

The command *solvatebox* created a rectangular parallelepiped solvent box around the solute UNIT (lig here). The water model TIP3PBOX is a pre-equilibrated box of TIP3P water. The number "30" means that the distance between the S3P and PEP atoms and the edge of the water box was at least 30 Å. The optional closeness parameter can be used to control how close, in Å, solvent atoms can come to solute atoms, 1.8 Å in this simulation. Then the Na+ ions were added in order to neutralize the system. And at last, the prmtop and frcmod files of our system were created.

-MD simulation runs:

From the same topology and coordinate files (lig.prmtop and lig.inpcrd) two simulations with very similar characteristics were run. The first one was a classical MD simulation and involved two minimizations, two equilibration steps and a production run at the end. The second simulation was setup using the same steps and parameter settings, but it was setup as a hybrid QM/MM simulation where S3P and PEP were treated quantum mechanically and the water molecules made a MM region.

Classical MD simulation:

First the system minimization in two stages was conducted. In the first stage the S3P and PEP molecules were kept fixed to remove the contacts between the water molecules. Then, in the second stage, the restrains on the S3P and PEP molecules were slightly release and the entire system underwent minimization:

```
minimization 1 (min1.in)
&cntrl
imin=1,maxcyc=1000,ncyc=350,cut=12,ntb=1,igb=0,ntr=1,
restraint_wt=100.0, restraintmask=':1-2'
/
```

It run for 1000 steps with the first 350 being steepest descent (ncyc) and the remainder conjugate gradient (maxcyc-ncyc). A smaller nonbonded cutoff (cut=12)periodic boundary simulations based on the particle mesh Ewald (PME) method, in this case with constant volume (ntb=1), were used. This is explicit water in

the system, so it is not necessary use the generalized Born implicit solvent model (igb=0). During first minimization big restraints (restraint_wt=100.0) onto S3P and PEP (restraintmask=':1-2') were imposed to minimize just the water molecules. With the command *ntr=1* one specifies that there are restrained atoms.

The second minimization took the same parameters like the first, except the restraints, which were smaller (restraint_wt=10.0).

There were two stages to the equilibration. In the first one the system was heated from 0 to 300K over 50 ps keeping all other parameters like in second minimization step. At this stage the equilibration was done with constant pressure (ntb=1). The second equilibration step took additional 50 ps but equilibration was at constant pressure (ntb=2) to get to a proper density. For a periodic system, constant pressure is the best way to equilibrate density, but first one has to equilibrate at constant volume (ntb=1 in the first equilibration) to something close to the final temperature, before turning on constant pressure. The options irest=1 and ntx=5 indicated that it started from coordinates and velocities calculated during the first run. The input file was:

```
equilibration 2 (md2.in)
&cntrl
imin=0,irest=1,ntb=2,ntp=1,igb=0,ntr=1,ntx=5,ntpr=200,
ntwx=200,ntwr=200,ntwe=200,ntt=3,gamma_ln=1.0,
tempi=300,temp0=300,nstlim=50000,dt=0.001,
cut=12.0,restraint_wt=10.0,restraintmask=':1-2'
/
```

Now the last MD run followed with parameters:

```
production run (md3.in)
&cntrl
imin=0,irest=1,ntb=2,ntp=1,igb=0,ntx=5,
ntpr=200,ntwx=200,ntwr=200,ntwe=200,ntt=3,
gamma_ln=1.0,tempi=300.0,temp0=300.0,
nstlim=200000,dt=0.001,cut=12.0
/
```

It was longer (200 ps), at constant pressure (ntb=2) and without any restraints (ntr=0, default value). The other parameters were like in the previous step.

Another possibility in the production run is use SHAKE algorithm to perform bond length constraints. SHAKE removes the bond stretching freedom, which is the fastest motion, and consequently allows a larger timestep to be used. Another simulation using SHAKE was run, where the bonds involving hydrogens were constrained (ntc=2, ntf=2). This allowed to decrease the number of calculation steps keeping the simulation time unchange, nstlim=100000 and dt=0.002, the simulation time 200 ps.

When the simulations were finished the output files were processed with the script "`process_mdout.perl`" as well as with AMBER analysis program *ptraj* in the similar way as for former simulations.

QM/MM MD simulation:

During simulation with coupled QM/MM potential two ligands were treated using the semi-empirical PM3 Hamiltonian, while all the water was modeled classically. There were no bonds that crossed the boundary of the QM and MM regions and no hydrogen link atoms must be placed, which usually is one of the biggest problems in this simulations.

The input files for QM/MM MD looked very similar to the MM MD files. just a few extra variables defining QM region were added:

```
QM minimization 1 (qm_min1.in)
&cntrl
imin=1,maxcyc=1000,ncyc=350,cut=12,ntb=1,igb=0,ntr=1,
restraint_wt=10.0, restraintmask=':1-2', ifqnt=1
/
&qmmm
qmmask=':1-2', qmcharge=-6,
qmtheory=1, qm_ewald=1, qm_pme=1
/
```

It is practically the same like the classical simulation. The differences were:

- ifqnt=1, that tells sander that you want a QM/MM run;
- and the &qmmm commands:qmmask=':1-2', to specify the QM region. In our case it will be the residues 1 and 2, which correspond with the molecules S3P and PEPqmcharge=-6. This is the charge on the QM regionqmtheory=1, to use the PM3 semi-empirical Hamiltonian
- qm_ewald=1, to use PME or an Ewald sum to calculate long range QM-QM and QM-MM electrostatic interactions
- qm_pme=1, to use a QM compatible PME approach to calculate the long range QM-MM electrostatic energies and forces and the long range QM-QM forces. The long range QM-QM energies are calculated using a regular Ewald approach.

In all other input files to run the simulations in sander the parameters were the same like in the classical MD, except those necessary to perform the QM/MM simulations. Also the steps, i.e. minimization, equilibration and production run, were exactly the same. Then the results were processed like in the classical simulations.

4. RESULTS

4.1.Classical MD of EPSPS in unliganded and liganded states

Analyses of the thermodynamics quantities, i.e. temperature, did not show any differences between simulations of liganded and unliganded states of EPSP synthase (Figure 10). As expected the temperature increased during the heating period of the simulation and fluctuated around 300K for the remaining time.

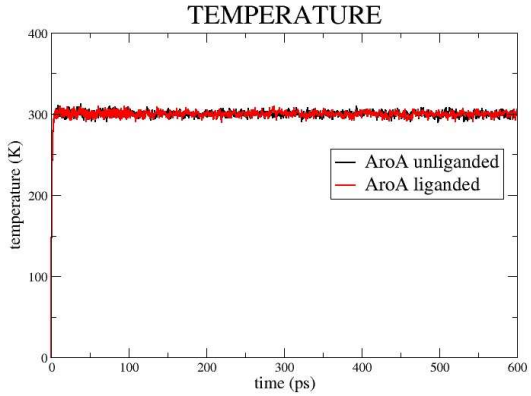

Figure 10: Temperature vs. time for liganded and unliganded states of EPSPS

In respect to energies, there was not observable difference in the kinetics energy between simulations of two forms of the EPSP synthase. In contrast, the two systems differed in the potential energy and, consequently, in the total energy (Figure 11):

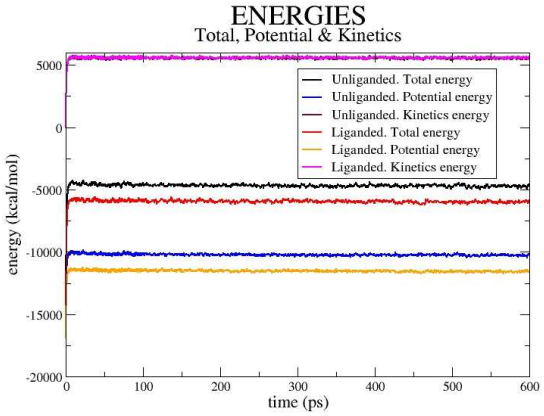

Figure 11: Energy (total, potencial and kinetics) vs. time for liganded and unliganded states of EPSPS

The potential energy of the system with the ligands is lower than its unliganded counterparts. In other words, the liganded system is more stable than the enzyme without ligands. It looks contradictory because the system with protein and ligands has more atoms and thus its potential energy should be higher. On the other hand, the side chains in the active site have reduced state of freedom, because of can be set up the interactions with the ligands, and this, of course will stabilize the system. It should be remembered that the enzyme structure is the close conformation, i.e. the structure when the enzyme is with S3P and PEP. We have used, like a starting file, the pdb file with the whole system and have just removed the ligands, so this is an unnatural structure. The enzyme with is unliganded is in the open conformation, where the two domains are separate. It should be a more stable structure than the close conformation without ligands.

It takes us to another question in the results. One can expect that the unliganded form go to open conformation during the simulation, because this is the natural conformation when EPSPS is alone. But when we visualize the results by PyMOL of the trajectory files can be observed that this doesn't happen. The movement of both systems is practically the same, and the two domains remain together during also in the unliganded system. Can be calculated the RMSd values for the two systems to compare them. In the next plot can be observed the RMSd value for both system and each one for the whole system and for the backbone:

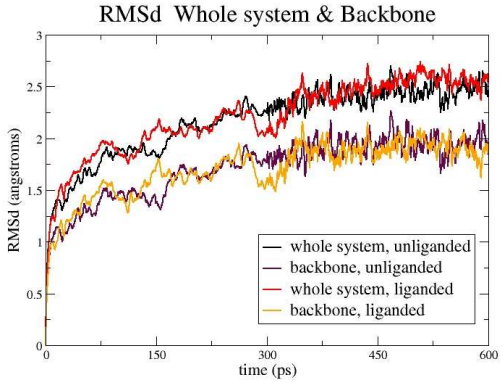

Figure 12: RMSd vs. time for whole system and the backbone for liganded and unliganded states of EPSPS

Obviously the RMSd for the backbone is smaller in both than the value for the whole system, due to the backbone atoms are quitter than the side chain atoms. In the plot there aren't big differences between both systems, like we have already explained. In the lasts 50 ps the values for the unliganded system increase a bit more than the others, but one can also see that in the lasts ps both values are practically the same. This small difference is not enough to say than the unliganded form go to the open conformation, perhaps is just for the random movement. One ought to run another simulation larger to determinate if it's a really difference and to check if with more time the unliganded system is opened.

With PyMOL can be visualized the movement of the system during the simulation. It must be done to check if everything is correct (sometimes it's easier check graphically than with the plots). PyMOL can do some calculations too, like distances, angles, dihedrals, rms values and so on, it will be really useful to start to know which calculations are necessary to do with ptraj. Another possibility is to visualize the polar interactions. We will show between the ligands (S3P and PEP) and the enzyme atoms:

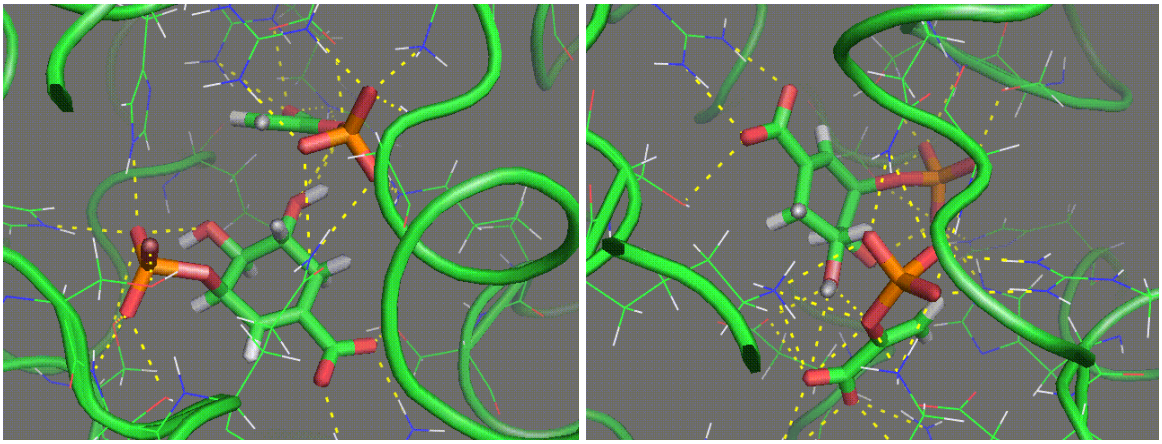

Figure 13: EPSPS active site whit S3P ans PEP. In yellow are the polar interactions between the two molecules with the rest od the system

One can observe which amino acids interact with S3P and PEP. When the movement is visualized it will show which interactions remained more time, these are stronger. The phosphate groups in both molecules interact with several side chains and it's quit stable. For instance, there are three serines (Ser169, Ser170 and Ser197) interacting with the S3P phosphate group during all the simulation (Figure 13, left) and others amino acids with the PEP phosphate group (Arg 124, Lys22, Gly96 and Gln171) (Figure 13, left and right). Also the acid group in S3P interacts which another argenine, Arg27, and one serine, Ser23, as can be seen very well in the right picture of the figure 13. And the PEP acid group with two argenines: 344 and 386. With PyMOL we can observe all this interactions in movement and from all the perspective. Another interaction that remains is between S3P and PEP. Interacts two oxygens and is quite stable during the simulation.

In next figure you can see a scheme of the interactions in the EPSPS active site between S3P, glyphosphate and the enzyme side chains[8]. Comparing with our results we can see a good correspondence.

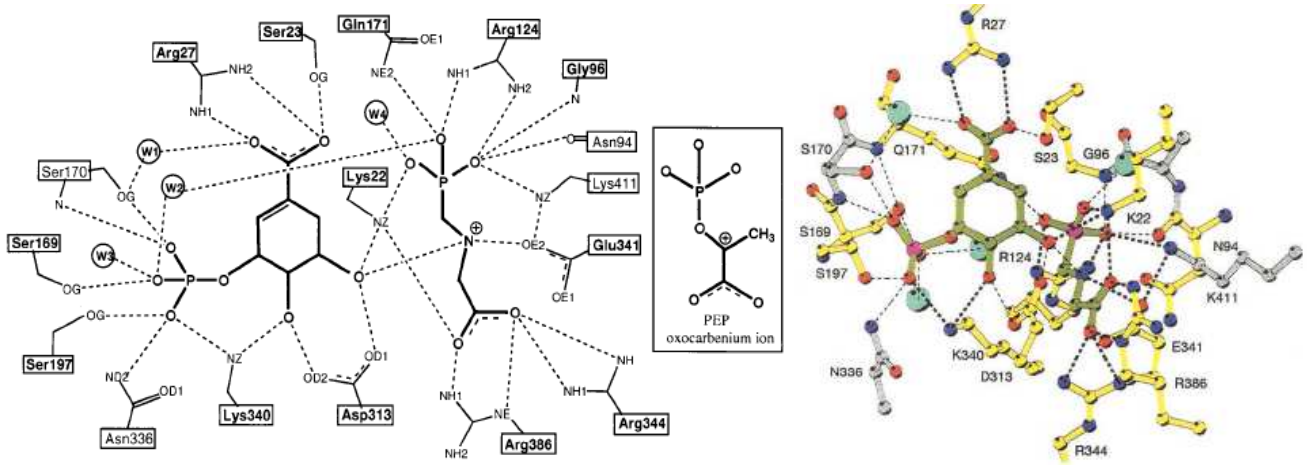

Figure 14: Scheme representation of ligand binding in the EPSPS-S3P-glyphosphate complex (from Schönbrum et al.)

One also can compare the RMSd values for the amino acids in the active site between both simulations: AroA alone and with the ligands:

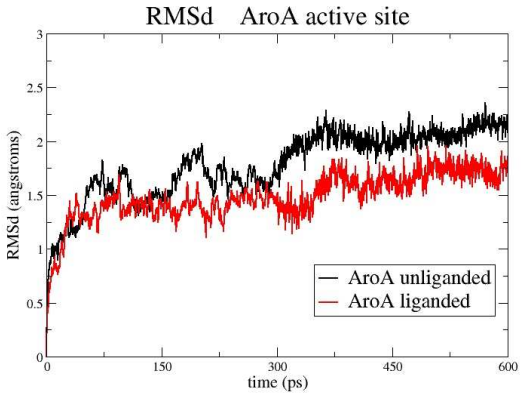

Figure 15: RMSd of the active site vs. time for liganded and unliganded states.

It was explained before that there are strong interactions in the active site between S3P and PEP with the amino acid side chains. It will produce that the movement are smaller, like can be checked in the previous RMSd plot. In the system which the enzyme is alone, the amino acids side chains of the active site have more movement freedom, as can be noticed in the plot, because there are several interactions missing.

The aroa_liganded simulation can be useful, because is present the whole system (EPSPS, S3P and PEP). It can be used like the first step to run later the QM/MM MD simulation. Can be checked if there are problems with the structures, charges, angles, distances and so on, because if here there are problems they will be also in the other simulation. Obviously this isn't so real and with the results we can't conclude anything. First because it was be used implicit water to the simulation to do it shorter. One can repeat this simulation with explicit water before to run with the QM/MM method. It will approximate more to the real system and could observe if some problems come up. Moreover in our systems there aren't the crystallize water molecules; we have removed from the pdb file. These are really important for the interactions in the active site, where there are 4 or 5 molecules which stabilized the ligands. Also is less realistic because we

have utilized classical MD to run the simulation, but, like I said before, it will be suitable like a first step.

4.2.Hybrid QM/MM MD & Classical MD S3P and PEP in a water box

The second simulation that was run is with the ligands in explicit water. It was done by two methods: Classical MD and hybrid QM/MM MD. You can compare the results to see the different accuracy in the results between them.

First one can see that the outputs files create by Sander are a bit different. This is the first step in the production run:

Classical MD, production run

```
 NSTEP =      200   TIME(PS) =     100.400  TEMP(K) =   291.40  PRESS =   311.0
 Etot   =    -43604.0416  EKtot   =      9216.4480  EPtot      =    -52820.4896
 BOND   =         9.4223  ANGLE   =        41.9022  DIHED      =        17.2118
 1-4 NB =         5.8829  1-4 EEL =       248.0994  VDWAALS    =      8025.4633
 EELEC  =    -61168.4716  EHBOND  =         0.0000  RESTRAINT  =         0.0000
 EKCMT  =      4586.5924  VIRIAL  =      3530.0172  VOLUME     =    157358.8192
                                                   Density    =         1.0108
 Ewald error estimate:   0.2346E-03
 ------------------------------------------------------------------------------
```

QM/MM MD, production run

```
 NSTEP =      200   TIME(PS) =     100.400  TEMP(K) =   284.97  PRESS =   192.3
 Etot   =    -44457.3080  EKtot   =      9013.1422  EPtot      =    -53470.4502
 BOND   =         0.0000  ANGLE   =         0.0000  DIHED      =         0.0000
 1-4 NB =         0.0000  1-4 EEL =         0.0000  VDWAALS    =      7881.8958
 EELEC  =    -58951.3977  EHBOND  =         0.0000  RESTRAINT  =         0.0000
 PM3ESCF=     -2400.9483
 EKCMT  =      4536.4364  VIRIAL  =      3875.6279  VOLUME     =    159188.0652
                                                   Density    =         0.9992
 Ewald error estimate:   0.4141E-02
 ------------------------------------------------------------------------------
```

The first thing one should notice is that the QM/MM result has an extra field (PM3ESCF). This is the energy due to the quantum part of the calculation (S3P and PEP) within the presence of the charge field of the MM part (the water). It should be also noticed here that the QM/MM result lacks ANGLE, DIHEDRAL, 1-4 NB and 1-4 EEL energies. This is exactly as we expect since the S3P and PEP bonds, angles, dihedrals and VDW and electrostatic terms are now dealt with inside the QM calculation. The only atoms remaining in the MM section are TIP3P water molecules. These are actually triangulated water and so only have bond terms (TIP3P water does not have an angle component). The water molecules are also only 3 atoms in size and so do not have any 1-4 NB or 1-4 EEL interactions.

Come now to compare the results. Respect the thermodynamic quantities can be said that are very similar. The temperatures are reasonably stable from the first ps and practically the same in both simulations. They will never be exactly the same since we are tracking different trajectories. The pressure will be similar and the volume and density we have had some problems which comment later. One also can observe in the next plot that the energies (total, potential and kinetics) are behaved like the temperature. They increase during the first ps of the first equilibration (when we have heated the system) and then remain quite stable during the rest simulation. If the plot is looked exhaustively can be notice that the energies in the hybrid method are a bit slower, which indicates that in this case the system is more stable.

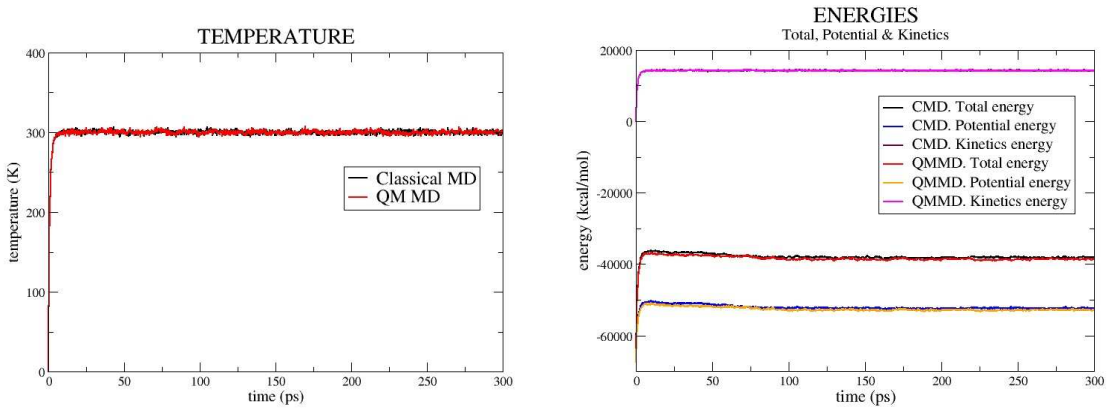

Figure 16: Temperature and energies vs. time for both methods

With the density it had some problems. Two different simulations were run for each method after the second equilibration, one using SHAKE and the other without it. Here are the results obtained:

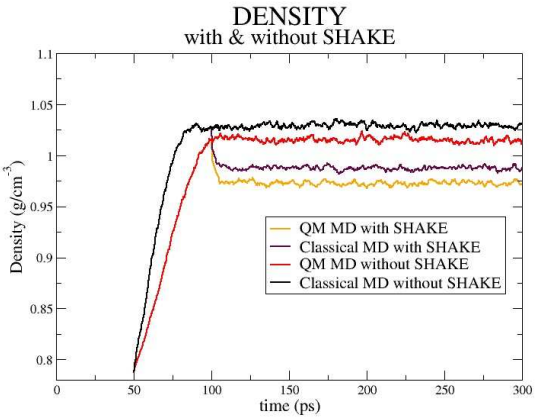

Figure 17: Density vs time for both methods with and without SHAKE

When the equilibrations finish (50 ps), in the simulations with SHAKE (purple and yellow lines) the density jump to and it decrease 0.05 g/cm^3. This decrease is similar in both and could be due to the equilibrations were running without SHAKE. Maybe if one will run all the steps in the MD with SHAKE (not the minimizations) could be obtained better results. Without SHAKE it hadn't this problem. Moreover in none of the simulations the value is totally correct. Remember that our systems are two small molecules enveloped by a big water box at 300 K, so the density ought to be 1 g/cm^{-3}, like the water alone. The disturbance that could cause the molecules isn't so big to produce this change. In the QM without SHAKE is quite correct, but the others are or bigger (CMD without SHAKE) or smaller (both with SHAKE) than the optimal value.

Analyzing the results by ptraj can be compared other parameters. Can be calculated, for instance the atomic position fluctuation:

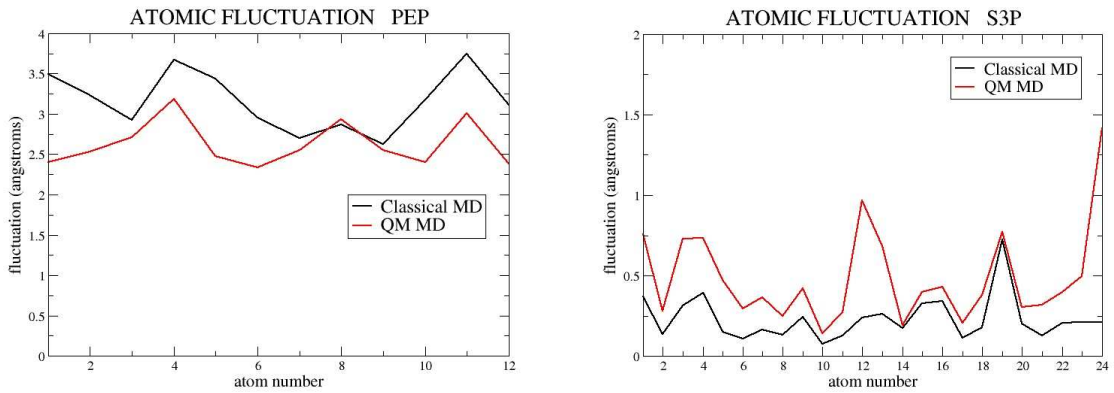

Figure 18: Atomic position fluctuations for PEP and S3P in both methods.

Comparing both methods is difficult say something, for the PEP is higher the fluctuation in the classical method and the opposite to S3P, smaller the classical. Can be also observed that the fluctuation is bigger in PEP, although isn't really big in any of them, i.e. the molecules remain quit stable. In our system somebody could think that during the simulation the molecules will be separate, due to only they are envelop by water molecules. How this don't happen it can be conclude that the interactions between them are strong, at least enough to keep it together. It will be observed better in the next plot which represents the distance between S3P and PEP, concretely between the oxygen O3 in S3P and the oxygen O2 in PEP. This is the strongest interaction between both molecules, like you can look in the animations in PyMOL (it can be also seen it in the simulation of the enzyme with the ligands).

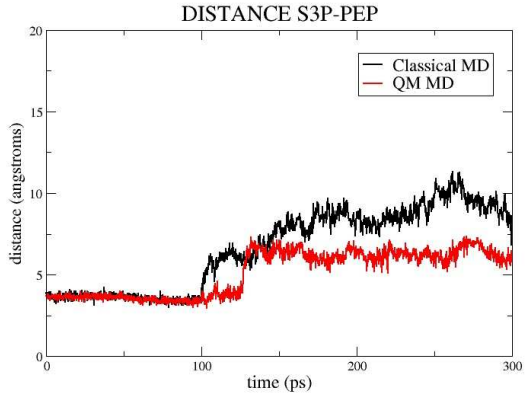

Figure 19: Distance between PEP and S3P vs. time in both methods.

There are clear changes after the equilibration in the distance. Remember that the equilibrations were run with restrains, and they were removed in the production run. This last step starts at 100 ps, before can be noticed that the distance is stable. At 100 ps the distance start to increase in the classical simulation and it continue increase during the simulation, except in the lasts ps. In the QM simulation the value is stable until about 120 ps, when it increase 5 Å and then remain stable again. It could be because when the system is free (without restrains) after a few ps the conformation changes to other more stable. When this is adopted the simulation goes on without big changes.

The explanation about the results in the classical simulations is difficult to determinate. But this is less accurate so you can suppose that the QM results are more realistic.

The changes after the equilibration can be observed also in the RMSd. It can be also calculated by ptraj and it was realized for the two molecules separate:

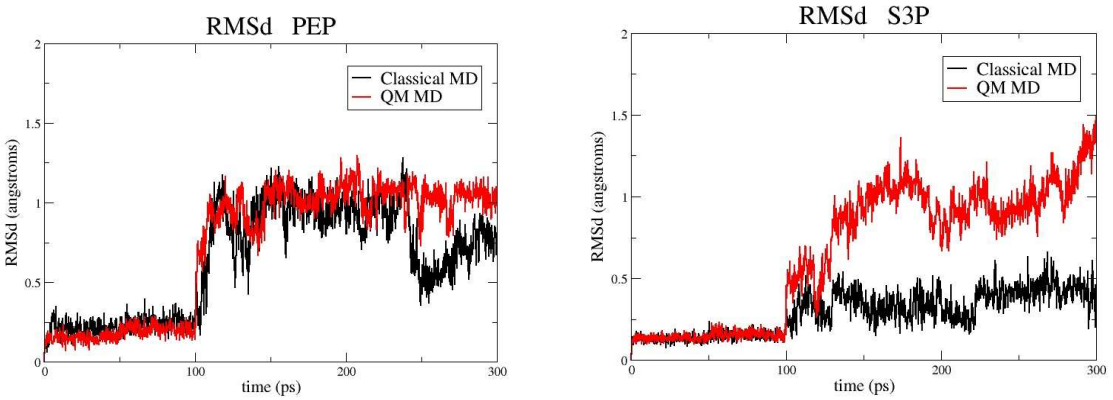

Figure 20: RASD value vs. time for PEP and S3P for both methods

A big change is produced in both methods for PEP when the restrains are removed. After few ps the system is stabilized, it is around 150 ps, when the distance between S3P and PEP is also stabilized. In the QM simulation the RMSd value remains quite constant during all the running. In contrast, the value in the classical simulation decreases a lot about 250 ps and then increase again. It can be supposed that is due to a change in the conformation, but isn't important, the QM is more realistic. The problem is that for S3P happen the opposite. During the first 100 ps both are constant, like it is expected, but then there are big changes. In the classical MD the value increase a bit and remain without big differences around 0.5 Å. In the other hand, the QM MD RMSd value is very different, it first increases about one angstrom, then decrease and continues variation during all the simulation. Nevertheless one can notice that all the values are very small, not more than 1.5 Å, so in spite of the differences are all the values quite correct or at least normal.

Can be compared these RMSd values with the values calculated from the system with the enzyme. Obviously if the molecules S3P and PEP are enveloped by the enzyme their movement will be smaller than when they are in a water box. One can check it in the next plots:

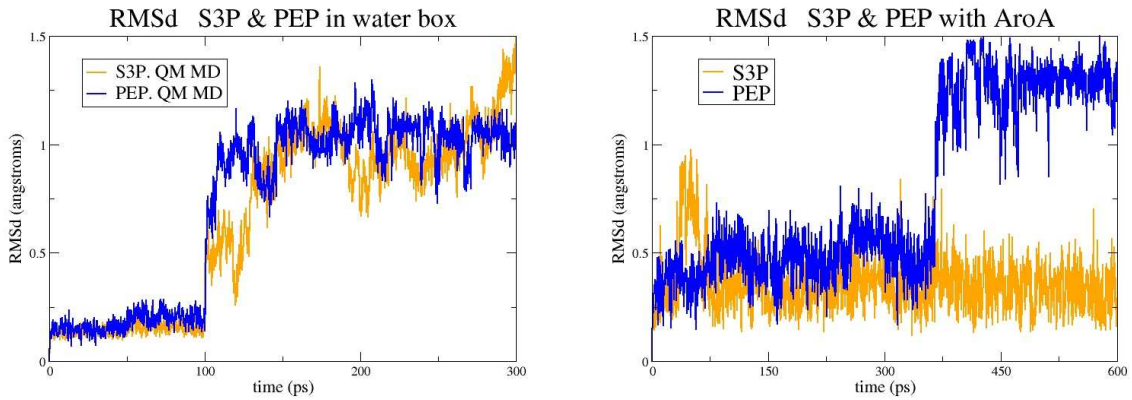

Figure 21: RMSd values vs time for PEP and S3P in a water box (left) and with EPSPS (right)

This is completely correct for S3P, its RMSd value is smaller with the molecules is enveloped by the enzyme. The movement for PEP is also smaller in the system with the enzyme. The RMSd value is a bit bigger in the lasts 250 ps, probably because the PEP molecule changes its conformation or position with a big movement and then it is stabilized again. The RMSd value is calculated comparing by the first position, for this reason, although it could seem that PEP moves more in the system with the enzyme in fact is the opposite. In spite of this have to be remembered that in the system where the molecules are in a water box there are also strong interactions and the movement of S3P and PEP it's not too big.

All this results obtained in the QM MD simulation seem correct, but when the structures are visualized it is observed important problems. Respect PEP can be observed that during the simulation the angles in the phosphate group are distorted. Before start the simulation they are around 109º, when one can expect. Then it begins to open, like a flower, and after the minimizations are less than 100º. When the simulation finish the values are around 94 in both of them. First we thought that it was due to use SHAKE, but it happens in both simulations. In fact is worse without SHAKE, two angles are around 94 at the end, like in the simulation with SHAKE, but now the other is nearly 120º, which is also a bad result. In the following figure can be visualized these values:

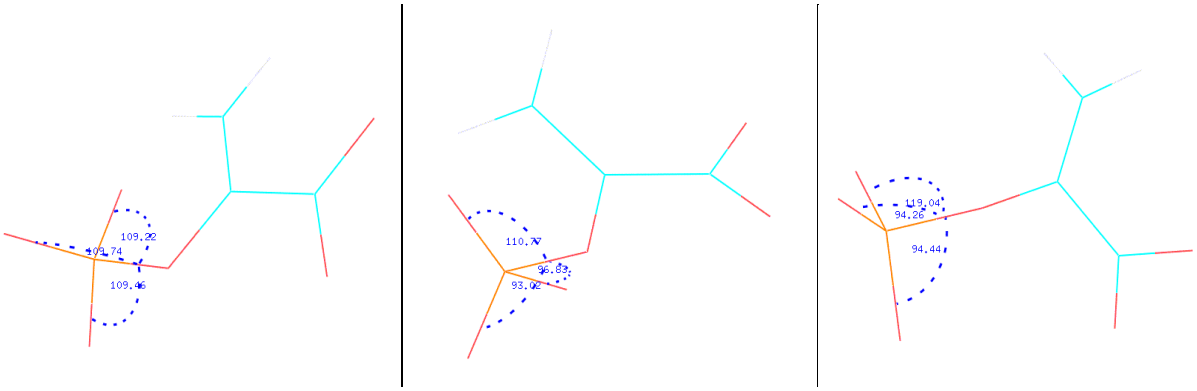

Figure 22: PEP and the angle values of the phosphate group. *Left*: before minimization. *Center*: after production run with SHAKE. *Right*: after production run without SHAKE.

This distortion can be consequence of the incorrect charges values in the phosphate. There is a big negative charge and if it isn't distributed correctly the atoms can repel ones with the others and it can cause this deformation. But comparing our initial charges with others in similar molecules with phosphate group seems likeness; equally it's obvious that there are some problems, so it should be recalculating to start again. The rest of PEP structure seems correct during the both simulations.

Other problem arises in the QM MD simulation. In this case is the S3P structure which also is distorted during the simulation. In the next picture obtained by PyMOL can be watched how the bond length between oxygen and hydrogen in one of the hydroxyl groups is increasing. Specifically is the oxygen O2 and the hydrogen HAB:

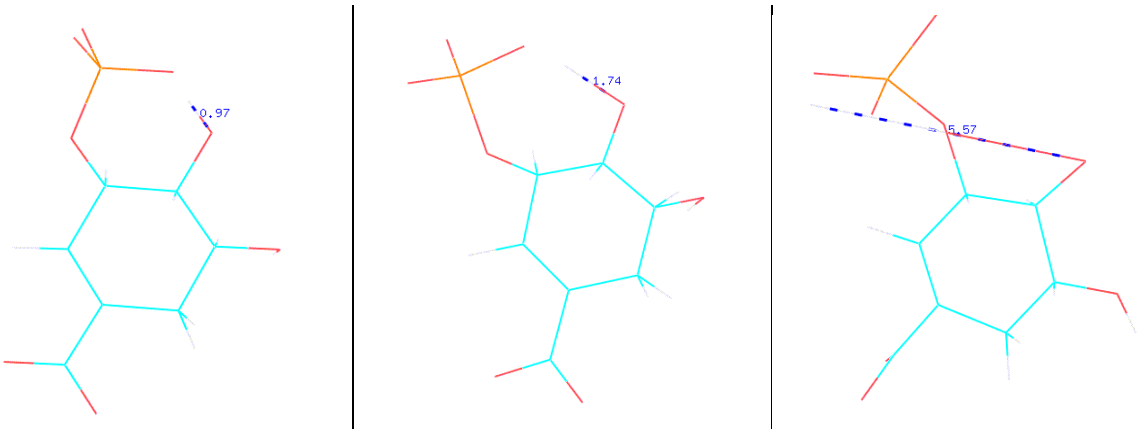

Figure 23: S3P and the bond length between O2 and HAB. *Left*: before minimization. *Center*: after production run with SHAKE. *Right*: after production run without SHAKE.

The distance before start the simulation is correct: 0.97 Å, like one can expect in a bond with this characteristics. The bond length starts to be higher during the minimization and continua increasing in the equilibration. When SHAKE is used in the production run the last value is 1.74 which is too big. But in the simulation without SHAKE, this bond is completely distorted, 5.57 Å. It's obvious that something is wrong. It can seem that use SHAKE is worse, because it has been obtained bad results. But normally with SHAKE is more correct. One should start using it from the equilibrations like we have already said.

Maybe the problem could be that the coordinate file which was used wasn't correct. If the starting coordinate file from the pdb file have some errors it can disturb all the simulation. Perhaps the problem was it, because seem that in the first structure the phosphate group is too near to this hydroxyl group. There are a hydrogen bond between the hydrogen HAB and one of the oxygen from the phosphate group. If this distance begin being to small it could go worse during the simulation. The phosphate group can pull on the hydrogen due to the strong hydrogen length, and it will be done the distance every time larger. For this reason it has to be checked again to insure that the starting files are correct.

5. CONCLUSION

The results presented here usefulness of the numerical solutions to study biochemical and thermodynamic properties of the biochemical systems. They allow to study and to compare, in relatively short time, the influence of different properties, i.e. protonation state of histidines or absence of co-factors, on protein thermodynamic states. As example system, AroA (EPSP synthase) together with S3P and PEP have been chosen. Molecular dynamics simulations showed that parameters applied for calculations were reliable as proofed by system stability. On other hand, some problems arose that should improve during the next simulations. The most important one is proving that charges assigned to atoms involved into phosphate groups in both S3P and PEP. This could be done if not Antechamber but Gaussian or other specialized software is used for *ab-initio* charge calculations. Another task would be improvement of the molecular dynamics of EPSPS. One possibility could be to run a simulation with the system solved in explicit water, but using still the classical MM/MD method. From X-ray structure, it is known that water molecules could play an important role in building the correct structure of the active side. Therefore, it should give more accurate results than in implicit water although the

simulation would take much more time and would extend over the available time period. It could be also more complete if we use SHAKE over the whole simulation time and compare with simulation without bonds restraining, or simply run the simulations over longer time to find the stable values. Another possible solution is realized by conducting several short simulations. It is also a good idea because the system does random movements and one can compare the results obtained in all of them to check if the variances are due to this random movement or are really variances.

One should also check again the charges and the starting coordinates with the optimal structure which won't be distorted during the simulation. With this system we ought to found the best prepi and frcmod files for PEP and S3P, which could be used to run other simulations, also the QM/MM.

Like the final step, when all would have been optimized, we should run the simulation with the whole system using the hybrid method. It could be treating S3P, PEP and the side chain of the active site amino acids with QM and the rest of the system classically. This isn't simple; it would have to determine correctly the QM region and the adequate boundary. Furthermore, this simulation would be really large, could spend weeks or months to run, depend on the other variables. For this reason we should start checking all the possibilities with these shorter simulations before run the most accurate and realistic simulation.

Bibliography

[1]. G. Sutmann, *Classical Molecular Dynamics*, published in *Quantum Simulations of Complex Many-Body Systems: From Theory to Algorithms*, Lecture Notes, J. Grotendorst, D. Marx, A. Muramatsu (Eds.), John von Neumann Institute for Computing, Jülich, NIC Series, Vol. 10, ISBN 3-00-009057-6, pp. 211-254, 2002.

[2]. M. P. Allen, *Introduction to Molecular Dynamics Simulation*, published in *Computational Soft Matter: From Synthetic Polymers to Proteins, Lecture Notes*, Norbert Attig, Kurt Binder, Helmut Grubmüller, Kurt Kremer (Eds.), John von Neumann Institute for Computing, Jülich, NIC Series, Vol. **23**, ISBN 3-00-012641-4, pp. 1-28, 2004.

[3]. D. S. Orcero, *Topics on Protein Dynamics*, http://www.orcero.org/irbis , 2004

[4]. G. Sutmann, *Molecular Dynamics – Vision and Reality*, published in *Computational Nanoscience: Do It Yourself!*, J. Grotendorst, S. Blügel, D. Marx (Eds.), John von Neumann Institute for Computing, Jülich, NIC Series, Vol. **31**, ISBN 3-00-017350-1, pp. 159-194, 2006.

[5]. F. Ercolessi, *A molecular dynamics primer*, http://www.sissa.it/furio, Spring Collage in Computacional Physics, ICTP, Trieste, 1997

[6]. D. Marx and J. Hutter, *Ab initio molecular dynamics: Theory and Implementation*, published in *Modern Methods and Algorithms of Quantum Chemistry*, J. Grotendorst (Ed.), John von Neumann Institute for Computing, Jülich, NIC Series, Vol. 1, ISBN 3-00-005618-1, pp. 301-449, 2000.

[7]. S. Eschenburg et al., *A New View of the Mechanisms of UDP-N-Acetylglucosamine Enolpyruvyl Transferase (MurA) and 5-Enolpyruvylshikimate-3-phosphate Synthase (AroA) Derived from X-ray Structures of Their Tetrahedral Reaction Intermediate States.* Published, JBC Papers in Press, September 16, 2003.

[8]. E. Schönbrunn et al., *Interaction of the herbicide glyphosate with its target enzyme 5-enolpyruvylshikimate 3-phosphate synthase in atomic detail.* Edited by Gregory A. Petsko, Brandeis University, Waltham, MA, PNAS, vol. 98, no. 4, 1376–1380, 2001

[9]. Stallings WC, Abdel-Meguid SS, Lim LW, Shieh HS, Dayringer HE, Leimgruber NK, Stegeman RA, Anderson KS, Sikorski JA, Padgette SR, Kishore GM. (1991) Biochemistry. 88:5046-50.

[10]. Amber 9. User's Manual.

[11]. AMBER Home page: http://amber.scripps.edu

[12]. Amber tutorials: http://amber.scripps.edu/tutorial/index.html

[13]. PyMOL home page: http://pymol.sourceforge.net/

[14]. Grace home page: http://plasma-gate.weizmann.ac.il/Grace/

[15]. http://www.wikipedia.org