SimuMole

Hi Level Design

Submitted by:

Eyal Faygenboim

Idan Hadad

Ilanit Smul

Michael Shohat

Nitzan Beck

Or Yasso

Supervised by:

Boaz Rosenan

Table of Contents

Abstract

Molecular dynamics (MD) is a computer simulation method for studying the physical movements of atoms and molecules. The atoms and molecules are allowed to interact for a fixed period of time, giving a view of the dynamic evolution of the system. In the most common version, the trajectories of atoms and molecules are determined by numerically solving Newton's equations of motion for a system of interacting particles, where forces between the particles and their potential energies are often calculated using interatomic potentials or molecular mechanics force fields.

The importance of the MD method is that it helps us understand the properties of assemblies of molecules in terms of their structure and the microscopic interactions between them. Computer simulations can be used as a complement to conventional experiments, enabling us to test a theory by conducting a simulation, or create simulations that are difficult or impossible in the laboratory. Therefore, MD enabling us to learn something new about molecular interactions, that sometime cannot be found in other ways.

In this project we will focus on simulating the molecular interactions that occur between two proteins: enzyme and substrate.

1. Introduction

1.1. General Project Description

The goal of the project is to create an interactive web application of MD simulation between two molecules determined by the user.

The user can choose the parameters for the simulation, such as: the participating molecules, the relative position between them, the system temperature, the duration of the simulation, and more.

In addition, the user can store and retrieve his simulations, as well as download a simulation as an animation.

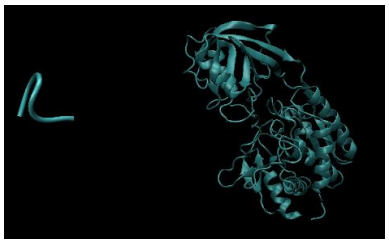


Figure 1a

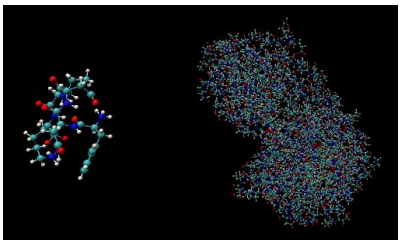


Figure 1b

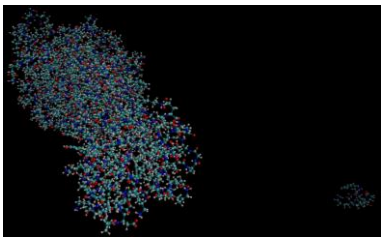


Figure 1c

Figures 1a-1c illustrate several frames from a simulation of two molecules: the large molecule is the enzyme and the smaller is the substrate. The user can choose the graphical representation of the molecules (figure. 1a vs. figures 1b-1c), and rotate the scene (in figures 1a-1b the small molecule is in the front while in figure 1c it is in the back).

1.2. Programming Environment

For creating MD simulations, we use **OpenMM** – a high performance toolkit for molecular simulation. OpenMM can be used as a library (and then the programmer can add MD features to his programs) or as an application (and then the end users who exposed to these features can run simulations). OpenMM is open source, licensed under MIT and LGPL.

For our project, we will use the application layer of OpenMM (written in Python), which provides the following functionality: reading input files (these are files that describe the proteins, such as PDB), editing molecular models (adding or deleting hydrogens, building solvent boxes, and more), defining the simulation parameters (energy minimization, temperature, number of steps, and more), defining the forces acting on the molecular system (either by specifying them explicitly or by loading a force field definition from a file), computing forces and energies, running simulations and outputting results.

For displaying the simulations, we use **PyMOL** – a Python-enhanced molecular graphics tool. It includes 3D visualization of molecules and trajectories, molecular editing (such as combining two molecules), and movies creating.

For creating the web application, we use **Django** – a high-level Python Web framework that encourages rapid development and clean, pragmatic design. Django is free and open source.

2. Theoretical Background

2.1. MD Algorithm

The MD simulation method is based on **Newton's second law**, $F = ma$, where F is the force exerted on the particle, m is its mass and a is its acceleration. From the knowledge of the force on each atom, it's possible to solve the **equations of motion**, and then find the trajectory that describes the positions, velocities and accelerations of the particles as they vary with time. The method is deterministic: once the positions and velocities of each atom are known, the state of the system can be predicted at any time in the future or the past.

The basic form of MD algorithm:

1. Divide time into discrete time steps.
2. At each time step:
   2.1. Compute the forces acting on each atom, using a molecular mechanics force field.
   2.2. Update the position and the velocity of each atom using Newton's laws of motion.

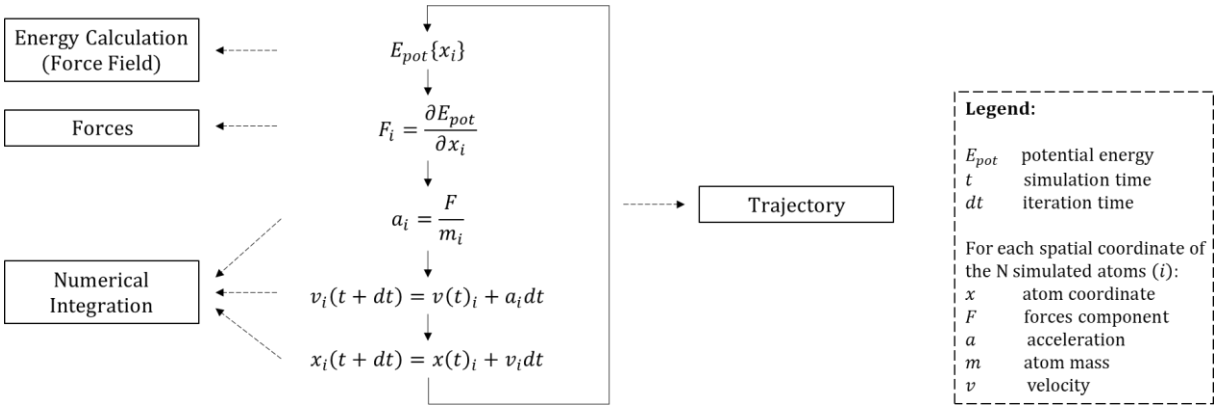


Figure 2

Figure 2 describes the MD algorithm scheme.

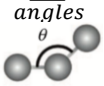
The difference between MD algorithms is in the **force fields** – each force field is based on experimental data, therefore most of them are different from each other. For example, the force fields of CHARMM and AMBER treat water model in the different pattern.

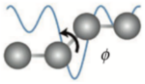
OpenMM supports CHARMM and AMBER force fields.

For example, the **CHARMM potential energy function**: [source of images: reference No. 6]

$$V(R) = V(R)_{bonded} + V(R)_{non-bonded}$$

$$V(R)_{bonded} = \sum_{bonded} K_b(b - b_0)^2 + \sum_{angles} K_\theta(\theta - \theta_0)^2 + \sum_{torsions} \sum_{n} K_\varphi(1 - \cos(n\varphi))$$





$$V(R)_{bonded} = \sum_{van\ der\ waals} \left(\frac{A}{r_{ij}^{12}} - \frac{B}{r_{ij}^{6}}\right) + \sum_{electrostatic} \left(\frac{q_i q_j}{\varepsilon r_{ij}}\right)$$

2.2. File Formats

Our application mainly uses the following file formats in order to create a MD simulation:

2.2.1. PDB File

- PDB (Protein Data Bank) file stores atomic coordinates of the molecule (in its initial form, and without further editing, it does not include the hydrogen atoms).
- The most significant part PDB file is the ATOM records that are defined for each atom in the molecule and contain the following fields: the record type (ATOM), atom ID, atom name, residue name, residue ID, x, y, and z coordinates, occupancy, temperature factor, segment name, and line number.
- PDB files may be generated by hand, but they are also available via the Internet (RCSB PDB, Worldwide PDB).
- PyMOL allows to edits PDB files (adding and removing atoms, combining several PDB files together, and more), and visualize the molecule represented by a PDB file (i.e., single frame).

2.2.2. DCD File

- DCD trajectory file contains for every timestep of the simulation the positions of the atoms, their velocities and forces.
- The DCD files are single precision binary FORTRAN files, so they can be transferred between computer architectures.
- PyMOL can visualize the simulation represented by the DCD file (i.e., multiple frame). In addition, there are tools for analyzing DCD files that can, for example, access the atomic coordinates at each timestep of the simulation.

3. Basic System Functionalities

- **Choose how to load the PDB input files:** whether by loading a locally stored PDB file, or by fetching a protein by its ID (when the wanted molecule indeed has its own ID that can be retrieved from Internet databases such as RCSB or wwPDB).

- **Select the type of the simulation:** simulation of only one molecule or simulation of two molecules. In addition, the user chooses whether the wanted output is a simulation of the molecules or just their display (single frame).

  In each case, the user should be allowed to view the simulation in "**rotate mode**": the user should be able to rotate and scale the viewpoint of the molecules.

  In addition, when saving the simulation (whether it's a single frame or an animation), the user should be allowed to determine where to place the "**camera**" in the output simulation.

- **Determine the initial relative position between molecules:** in case the user is interested in simulating two molecules, he should provide parameters relating to their initial relative position, such as distance between the surface of the molecules (this functionality will provide by PyMOL).

- **Select the parameters related to the building of the simulation:** such as temperature, timestep duration, simulation duration, and more.

- **Display an existing simulation** that was previously created by our application (DCD file).

4. Software Implementation

4.1. Main Menu

The user can perform one of the following:

- **Create simulation:** given PDB file(s), simulation type, and simulation parameters, the application will produce an appropriate simulation, display it to the user, and finally allow to download it (as a DCD file, or as an animation / image of a single frame).

- **Display an existing simulation**: given a DCD file representing a simulation (whether created directly by our application or not), the application will display the simulation to the user.

4.2. Create Simulation

In order to create a simulation, the user should perform the following steps:

4.2.1. Select Simulation Type

The user can choose whether to simulate only one molecule or two molecules.

Then, the user determines whether to:

- Create a simulation (number of frames).
- Only display molecule(s) (single frame) – in this case the user can skip steps 4.2.4-4.2.5.

4.2.2. Upload the Input Files

For each of the wanted molecules, the user chooses how to load the relevant PDB file:

- By loading the corresponding PDB file directly.
- By fetching the molecule by its ID – this option requires Internet connectivity in order to fetch files over the Web.

4.2.3. Determinate the initial position of the molecules

In case its required to simulate two molecules, the user must determine the initial relative position between the molecules. This step is done using PyMOL which combines the two input PDB files into a single PDB file.

4.2.4. Set the Simulation Parameters

The user sets the parameters required for the simulation by OpenMM, such as:

- System temperature.
- Timestep duration.
- Simulation duration or total number of timesteps.

4.2.5. Build the Simulation

At this stage the application builds the simulation using OpenMM, and returns as output a DCD file which describes the simulation.

4.2.6. Displays the Simulation

At this stage the application visualizes the molecule(s) using PyMOL: if a simulation is required – the application loads the CDC file, whereas if a single frame is required – it loads the PDB file.

4.2.7. Download the Simulation

The user can save the output files (PDB, DCD) and the simulation (in the form of an image or animation). For the second option, the user decides where to place the camera.

4.3. Display an Existing Simulation

The user can upload a simulation previously created by the application (DCD file), and then repeat sections 4.2.6-4.2.7 (view and download the simulation).

5. References

Molecular dynamics (MD):

1. "Molecular dynamics", *Wikipedia: The Free Encyclopedia*.
   en.wikipedia.org/wiki/Molecular_dynamics
2. "Introduction to Molecular Dynamics Simulation", *John von Neumann Institute for Computing*.
   udel.edu/~arthij/MD.pdf
3. "Theory of Molecular Dynamics Simulations", *The European Molecular Biology network (EMBnet).*
   embnet.vital-it.ch/MD_tutorial/pages/MD.Part1.html
4. "Potential Energy Function", *The European Molecular Biology network (EMBnet).*
   embnet.vital-it.ch/MD_tutorial/pages/MD.Part2.html
5. "Molecular dynamics simulation", *"Computational Biology: Structure and Organization of Biomolecules and Cells" course, Stanford University.*
   web.stanford.edu/class/cs279/lectures/lecture4.pdf
6. "3D RNA simulations", *Paris Descartes University*.
   algosb2019.sciencesconf.org/data/Algo2019_3Dmodeling.pdf

OpenMM:

7. Official website.
   openmm.org
8. "OpenMM 7: Rapid development of high-performance algorithms for molecular dynamics", *"PLOS Computational Biology" journal*.
   journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1005659

PyMOL:

9. Official website.
   pymol.org

Django:

10. Official website.
    djangoproject.com