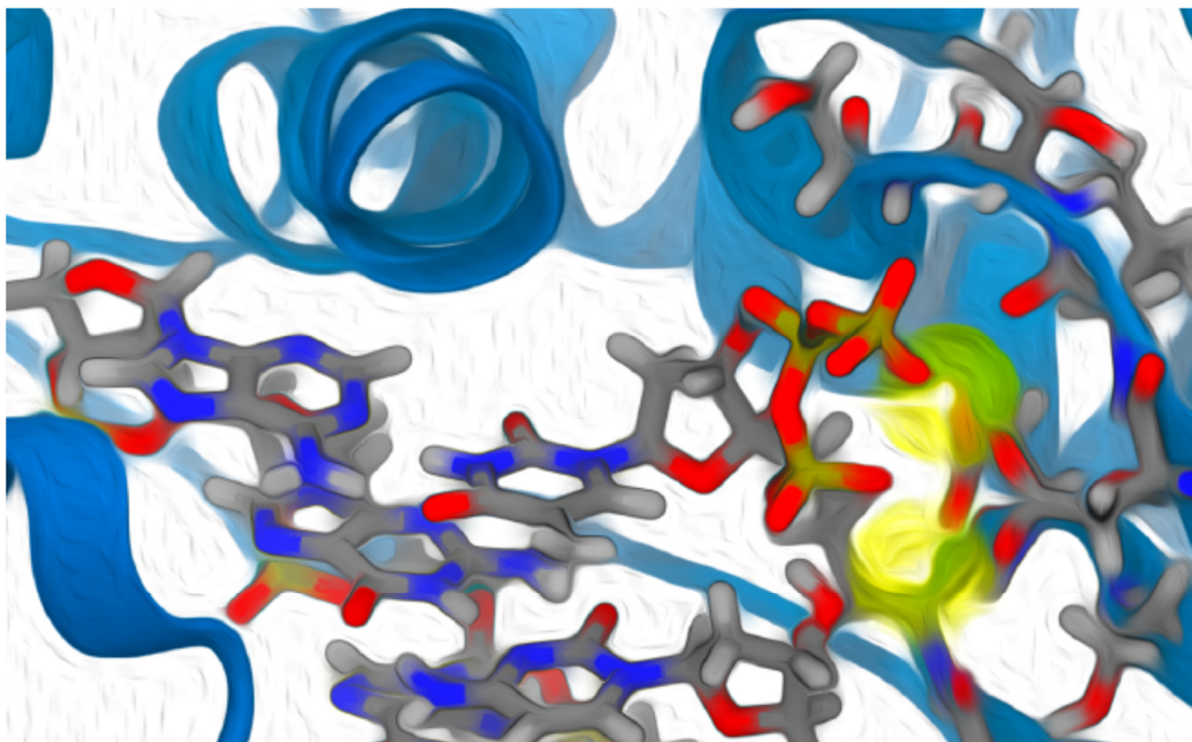




POLITÉCNICA



**IMPLEMENTATION OF A  
MOLECULAR DYNAMICS WORKFLOW  
BASED ON AMBER MD SOFTWARE IN  
SCIPION, A SOFTWARE FRAMEWORK  
TOWARDS INTEGRATION,  
REPRODUCIBILITY, AND VALIDATION OF  
DIFFERENT RESEARCH TOOLS**



COMPUTATIONAL BIOLOGY  
MASTER THESIS

AUTHOR: AIDA PINACHO  
CNB TUTOR: CARLOS ÓSCAR SORZANO  
UPM TUTOR: STEPHAN POLLMANN

CENTRO NACIONAL DE BIOTECNOLOGÍA (CNB-CSIC)

## **Acknowledgements**

**INDEX**

<b>ABSTRACT</b>	<b>5</b>
<b>1. INTRODUCTION</b>	<b>6</b>
<b>1.1. Molecular Dynamics (MD)</b>	<b>6</b>
1.1.1. MD Force Fields	7
1.1.2. MD workflow methodology	7
1.1.3. MD applications	8
1.1.4. MD limitations	9
<b>1.2. Amber software</b>	<b>9</b>
<b>1.3. Scipion framework</b>	<b>11</b>
1.3.1. Scipion-chem	11
1.3.2. Scipion's <i>plugin</i> structure	13
1.3.3. Scipion's GUI	13
<b>2. OBJECTIVES</b>	<b>16</b>
<b>3. MATERIALS AND METHODS: <i>Plugin</i> development</b>	<b>17</b>
<b>3.1. <i>Plugin</i> structure</b>	<b>17</b>
<b>3.2. Amber workflow within Scipion</b>	<b>18</b>
<b>3.3. Integrated protocols</b>	<b>19</b>
3.3.1. Import Atomic Structure	19
3.3.2. Extract Ligands	20
3.3.3. Target/Protein Preparation	20
3.3.4. System preparation	21
3.3.5. MD Simulation	23
<b>4. RESULTS - Practical examples of use</b>	<b>25</b>
<b>5. CONCLUSIONS AND FUTURE WORK</b>	<b>29</b>
<b>6. BIBLIOGRAPHY</b>	<b>30</b>

**INDEX OF FIGURES**

<b>Figure 1.</b> MD simulation workflow adapted from (Patodia, 2014)	8
<b>Figure 2.</b> Virtual drug screening scheme	12
<b>Figure 3.</b> Scipion's main window	13
<b>Figure 4.</b> Scipion's main Project window	14
<b>Figure 5.</b> Scipion's browser window and example of Scipion's protocol form	15
<b>Figure 6.</b> Scipion-chem-amber main structure	17
<b>Figure 7.</b> Workflow scheme integrated in Scipion	19
<b>Figure 8.</b> Protocol form Extract Ligands	20
<b>Figure 9.</b> Protocol form Target preparation ADT from Autodock Software	21
<b>Figure 10.</b> Protocol form System Preparation input tab	22
<b>Figure 11.</b> Protocol form system preparation MD prep tab	22
<b>Figure 12.</b> Protocol form MD simulation	23
<b>Figure 13.</b> Example of the visualization of an Ambersystem trajectory using VMD	24
<b>Figure 14.</b> System Preparation protocol form for MD simulation of 2EWL	25
<b>Figure 15.</b> File 2ewl_check.pdb visualized with MD Software	26
<b>Figure 16.</b> MD Simulation protocol form for MD simulation of 2EWL	27
<b>Figure 17.</b> Viewer window display for MD Simulation protocol	28
<b>Figure 18.</b> Visualization of 2EWL trajectory in VMD	28

**Abbreviations**

MD	Molecular Dynamics
SBDD	Structure-based-drug design
PSP	Protein structure prediction
GPU	Graphics Processing Unit
GUI	Graphical User Interface
VMD	Visual Molecular Dynamics
CC	Cervix Carcinoma
HPV	Human Papilloma Virus

## ABSTRACT

Dynamics simulations have become an established tool in the study of biomolecules, complementary to experimental techniques. The impact of molecular dynamics (MD) simulations in molecular biology and drug discovery has expanded dramatically in recent years. These simulations capture the behavior of proteins and other biomolecules in full atomic detail and at very fine temporal resolution. In addition, MD allows the study of a wide variety of important biomolecular processes, including conformational change, ligand binding, and protein folding. However, specific prior knowledge about the software package of choice and concepts about bioinformatics are mandatory to tackle the MD projects. This is the reason why AMBER MD Software was integrated as a new *plugin* within the Scipion framework, a platform created for the integration of different software packages that allows researchers to combine different tools while facilitating usability since it offers a GUI designed for the correct performing of a simple molecular dynamics simulation. The present work also shows a practical example of use by performing a simple 50 ns MD simulation of the oncoprotein E7 associated with the appearance of cervical cancer.

**Keywords:** Molecular Dynamics, AMBER MD Software, Scipion-chem, *plugin*

# 1. INTRODUCTION

Understanding of the 3D structures of proteins, also known as Structural Biology, has long been acknowledged as having the potential to speed up drug discovery, but recent advances in genome sequencing, robotics, and bioinformatics have dramatically expanded the possibilities (Congreve et al., 2005). Specifically, computational structural biology is an extensive field, and it is gaining importance since it is not a complementary tool anymore but leads research (Nussinov et al., 2019).

For instance, computational biology is useful for filtering and prioritizing when the number of experimental candidates in drug-discovery studies is elevated, which can be very beneficial in saving time and resources. In the same way, computational structural biology is known to be effective for the revealing of how protein functions are performed (Nussinov et al., 2019) such as signaling specificity or mechanisms of mutations

Thus, as biochemical processes originate from structural behavior, structural data can be studied from a dynamic point of view, which has led to the consolidation of Molecular Dynamics (MD) as a standard tool in biomolecule research and has contributed to the detail understanding of biomolecular processes that we have today (Feig et al., 2018; Hansson et al., 2002)

Even though Molecular Dynamics faces many challenges, one of them is the lack of interoperability and consensus among different software packages (Hedges et al., 2019), as well as the difficulty of tackling molecular dynamics projects without prior knowledge of bioinformatics. This phenomenon causes difficulties inside the research community when it comes to reproducibility and methodology sharing. In order to address this problem, the CNB-CSIC Biocomputational Unit created Scipion-chem framework with the objective of integrating and managing the interaction between various packages and programs aimed at virtual drug screening.

## 1.1. Molecular Dynamics (MD)

The science behind how molecules move, deform, and interact over time among themselves, and others is known as Molecular Dynamics (MD). Predicting and understanding these phenomena is critical in fields such as chemistry, physics, biology, or engineering (Leimkuhler & Matthews, 2015).

In addition to experimental techniques, dynamics simulations have become a well-established tool in the study of biomolecules (Hansson et al., 2002). In recent years, the use of molecular dynamics (MD) simulations in molecular biology and drug discovery has grown considerably. The behavior of proteins and other biomolecules is captured in atomic detail and at exceptionally fine temporal resolution in these simulations. These simulations allow researchers to investigate a wide range of critical biomolecular processes, such as conformational change, ligand binding, and protein folding, while also displaying the positions of all atoms at any given time (Hollingsworth & Dror, 2018).

MD simulations are not a novel concept. The first MD simulations of simple gasses were performed in the late 1950s (Alder & Wainwright, 1959). The first MD simulation of a protein was performed in the late 1970s (McCammon et al., 1977), and the groundwork that

enabled these simulations was among the achievements recognized by the 2013 Nobel Prize in Chemistry (Levitt & Lifson, 1969; Lifson & Waeshel, 1968).

The underlying principle behind an MD simulation is that given the positions of all of the atoms in a biomolecular system, it is possible to calculate the exerted force produced on each atom by all of the other atoms. Then, Newton's laws of motion are used to predict the spatial position of each atom as a function of time and obtain a trajectory that is, in essence, a 3D movie that describes the atomic-level configuration.

The first molecular dynamics simulation was performed 45 years ago by J. Andrew McCammon, Bruce R. Gelin & Martin Karplus (McCammon et al., 1977). These researchers studied the bovine pancreatic trypsin inhibitor dynamics, a folded globular protein, in a vacuum. Since then, biocomputational tools and structural biology knowledge have evolved to what we now know as modern Molecular Dynamics in which solvent and temperature affect the protein structure and dynamics (Karplus & McCammon, 2002).

### 1.1.1. MD Force Fields

The mathematical basis for the calculation of trajectories in molecular dynamics simulations is based on solving the Newton's equations of motion for every particle in the system (Norman & Stegailov, 2013). This is, in a system of many particles, it is possible to predict each particle position after a time interval by calculating which are the forces that are acting on each one after a given time and obtain their new position or *coordinates* (Ponder & Case, 2003). Repeating this procedure various *time-steps*, we end obtaining a sequence of different coordinates, velocities, and forces for each particle in each time interval which is collectively referred to as *trajectory*.

In order to describe the phenomena that allows to predict the special movement of atoms, a combination of mathematical formula and associated parameters is used, and it is referred as "force field" (Guvench & Mackerell, 2008).

In the context of molecular dynamics simulation of proteins, classical approaches are based in the idea of using different functions for each atomic coordinates and describe its potential energy during a defined time period. Usually, these force field parameters are retrieved from quantum mechanical and experimental studies of molecules (Patodia, 2014). In simulations regarding biomolecular systems, de most commonly used force fields involve AMBER, CHARMM and GROMOS.

### 1.1.2. MD workflow methodology

For biosystems simulation, the main objective is to recreate the natural environment of the molecules. For this reason, different parameters such as pressure and temperature must be carefully selected (Patodia, 2014). Normally, protein simulations are performed in canonical ensemble (NVT) or isothermal-isobaric ensemble (NPT) and solvated into an explicit solvent. Additionally, in order to maintain the distance between the different mass points present in a system and ensure constant bond angles, molecular dynamics occasionally use constraint algorithms such as the SHAKE algorithm (Yoneya et al., 1994).



Normally, MD simulation of a protein follows a workflow (*Figure 1*) that starts with the energy minimization or energy optimization of the system. This step is performed with the aim of finding a global minimum energy that represents the minimum energy conformation of the molecule (Roy et al., 2015).



**Figure 1.** MD simulation workflow adapted from (Patodia, 2014)

The next phase regards heating and equilibration. It is performed assigning initial velocities to each atom in the system and then, as time passes, assigning new ones. This process is performed by increasing the temperature until the desired temperature is reached. Then, equilibration step is in charge of balancing potential energies and fix the protein position.

Finally, the production MD phase removes the fixed constrains and allows the movement of the atoms for the selected parameters and conditions generating a trajectory. This retrieved trajectory can be analyzed and might give information about conformational changes and protein function, for example by calculating RMSD differences between two structures.

### 1.1.3. MD applications

MD has been a very useful tool in helping understand biology since its beginning 45 years ago. Since molecular interactions determine biological functions, the study of macromolecular structures allows to explore the phenomena of allostery, molecular docking for drug design and structure prediction (Hospital et al., 2015).

- **Allostery research**

Allostery is a critical element determining the relative populations of accessible conformational states. It is an inherent physical feature of proteins (Nussinov et al., 2014) and it is defined as the shift in the conformational ensemble's distribution caused by a perturbation such as ligand interaction, covalent post-translational modification (PTM) or mutations.

These conformational transitions are responsible for protein regulation in most systems (Hospital et al., 2015) and it is logical to study these structural changes through Molecular Dynamics.

- **Molecular docking and drug design**

Molecular docking, structure-based virtual screening, and molecular dynamics (MD) are the most common strategies performed in structure-based-drug design (SBDD)(Ferreira et al., 2015). Particularly, MD is capable of analyzing the flexibility of the target binding site which is a crucial characteristic in molecular docking.

In modern systems biology, exploration of protein-ligand interaction is gaining importance since the interactions between three-dimensional structures are considered the key point in structured-based drug design (Hospital et al., 2015). In this context, depending on the situation and which is the object of study, MD is performed before docking study to alter the original crystal structure if necessary, or after in order to evaluate the binding predicted (Zhao & Caflisch, 2015).

- **Structure prediction**

Although the cost of obtaining new protein structures is no longer as expensive as it was, solving high-resolution protein structures continues to be time-consuming and requires high costs (Geng et al., 2019). Refining protein structure prediction (PSP) using MD consists of template-based modeling which uses different original structures to predict new ones (Hospital et al., 2015). This process of protein design is based on the application of knowledge about their structure and function and the relationship between these two concepts to create proteins in two different ways, *de novo* or by modifying an existing one (Childers & Daggett, 2017).

#### **1.1.4. MD limitations**

As in any experimental design, Molecular Dynamics has limitations in terms of the accuracy of the proposed models (Childers & Daggett, 2017). It is important to consider the different parameters used in the simulations. First, the force field chosen must be suitable for the system of study, if not, calculations inside MD simulation will not be accurate and the trajectory will not be realistic. Second, parameters such as time of simulation and MD ensemble are crucial to perform and appropriate prediction.

Furthermore, Protein Data Bank formats only provide a partial view of the 3D structure since proteins are flexible and change their dynamics through conformational changes when performing their function. For example, one single protein family structure is variable depending on the different active-site ligands present when the structure was crystallized (Hospital et al., 2015).

## **1.2. Amber software**

AMBER (D.A. Case et al., 2022) is a software package which contains a suite of programs originally developed by Peter Kollman (Weiner & Kollman, 1981) and collectively maintained by different researchers around the world with the aim of allowing users to perform molecular dynamics simulations.

This software package is divided into two different parts depending on the license that governates its programs. Ambertools22 is a set of freely available programs under the GNU General Public License (GPL), and Amber22 which has a more restrictive license but has further applications.

The term Amber also refers to the set of molecular mechanical force fields that are implemented within this software (Ponder & Case, 2003). Depending on the biological characteristics of the system of study, Amber recommends different accurate *force fields* (Table 1).

**Table 1.** Amber recommended force fields depending on the studied system (Extracted from Amber Reference Manual)

Amber recommended force fields	
<b>Proteins</b>	<i>ff14SB</i>
<b>DNA</b>	<i>OL15</i>
<b>RNA</b>	<i>OL3</i>
<b>Carbohydrates</b>	<i>GLYCAM_06j</i>
<b>Lipids</b>	<i>lipid17</i>
<b>Organics molecules</b>	<i>spc/e, tip4pew, and OPC</i>
<b>Water model</b>	<i>gaff2</i>

Amber software may seem complex at the beginning because of the wide range of possibilities offered by its programs. It is therefore important to know which are the most used programs and what is the information they need to work, as well as the information that comes out of them and how to interpret it.

Among the most useful programs within amber, here are their names and main features:

### **Pdb4amber**

Pdb4amber is a preparatory program whose main functionality is to prepare the file formats to be compatible with LEaP. It is important when using Amber that the original pdb downloaded is clean and processed by the user so although this program is useful, the user must work in the pdb file before starting the simulation process.

### **Antechamber**

When analyzing small organic molecules such as ligands, this program allows to create and develop specific force fields that accurately fits the particle.

### **LEaP**

The most important program to prepare a new Amber system or modify an existing one.

This program generates two main files, topology, and coordinates from the system of study. This is performed by specifying different parameters such as the necessary force fields, presence of disulfide bridges, charges of the system... etc.

### **Parmed**

Although is one of the main programs within Amber, it is an auxiliary program which can be used for extracting information from the coordinate and topology files and make sure that the files obtained can be used in complex systems for simulation.

### **Sander** (Simulated Annealing with NMR-Derived Energy Restraints)

Main program for performing Molecular Dynamics distributed freely in Ambertools. This program integrates the Newtonian equations of motions and predict the movement of the system through different time intervals finally retrieving a trajectory.

This program needs the coordinate and topology files extracted from LEaP as input and a .in file that contains the parameters for running the simulation

### **Pmemd**

Optimized version of sander distributed under license in Amber that can run on GPUs

### **MMPBSA.py**

Python script whose propose is automating energy analysis of specific intervals from a MD simulation

### **cpptraj**

cpptraj is considered the main program designed for trajectory analysis within Amber. Among its main features, using this program it is possible to perform superpositions, extractions of coordinates, calculation of bond/angle/dihedral values, etc

## **1.3. Scipion framework**

*Scipion* ( <http://scipion.i2pc.es> ) is a platform that was originally developed for the integration and interoperability in 3D Electron Microscopy (de la Rosa-Trevín et al., 2016).

This bioinformatic framework is designed modularly based on *plugins* of different software packages that add new functionalities to the platform. It is distributed for Linux operating systems, and it is developed and currently maintained by the Biocomputational Unit in the National Biotechnology Center (CNB-CSIC).

The main objective of the Scipion project is to allow the use of different software packages in the same workflow in order to enable the researcher to combine different tools without knowing the full intricated protocols of each software, fixing the lack of interoperability between different programs and file formats. Normally, each protocol within Scipion corresponds to a specific program, although it may happen that several programs are coded inside the same protocol to facilitate the user experience.

In addition, Scipion platform also has a traceability function of the executed workflows. It saves the inputs, outputs and parameters obtained in each protocol executed so that the user can examine the results, possible errors produced, and decisions made in the process after it has been finished.

Scipion's structure is based on plugins or complements. These plugins add new functionalities to the Scipion framework and can be downloaded and even modified by the user. It also works with different projects that the user can create, and this approach allows the independency of each project depending on the user's decision.

The original branch *Scipion-em* contains different software for cryo-Electron Microscopy image processing, single Particle Analysis (SPA), Electronic Tomography and Molecular Modeling.

### **1.3.1. Scipion-chem**

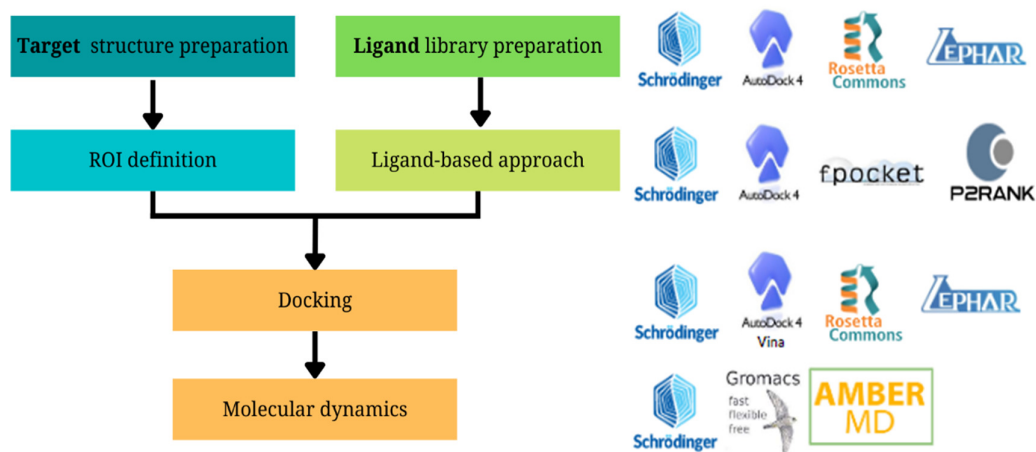
However, Scipion is currently expanding its borders to Biology and Chemistry, and this is the reason why *Scipion-chem* was created. With the same purpose as before, this platform is leading towards the integration of different software packages to allow researchers to

work with different tools in an easier way, this time, directed to virtual-screening and chemoinformatics.

So far, Scipion-chem contains different software packages that perform generic tasks such as downloading structures from web databases (ZINC, PubChem, UniProt), format conversion (OpenBabel, RDKit) or protein structure prediction (I-Tasses, RaptorX).

Furthermore, the aim of this new branch is to integrate software for all the different steps of a virtual drug screening workflow (*Figure 2*).

Scipion-chem already contains different software packages designed to, on the one hand, prepare the target structure and define its regions of interest and on the other hand, prepare the ligands that will be part of a library for a ligand-based approach. In addition, it contains software designed to perform docking studies and finally, software packages to perform molecular dynamics analysis.



**Figure 2.** Virtual drug screening scheme showing the different paths that can be followed inside Scipion-chem. (left) and software packages integrated within Scipion capable of performing each step (right)

Concerning the Molecular Dynamics software already integrated within Scipion-chem, Schrödinger and GROMACS are two software packages designed for the same purpose as Amber: performing MD simulations. However, as Scipion is a platform whose principal objective is to integrate diverse software and let the user decide which programs and protocols are used and how to combine them, including Amber to the list is nothing more than an advantage for the researchers.

In addition, Scipion-chem also acts as the ideal tool to integrate software packages with the same purpose but different input and output file formats. This process is performed through the creation of different Scipion objects depending on the file's nature. Scipion contains different types of objects that define for example ligands (*SetOfSmallMolecules*), protein structure (*AtomStruct*), or it is even possible to create new Scipion objects that contain the different information retrieved by the program's protocol integrated. These objects are used to match the input and output requirements of each program, so that the interaction between them and their actual format are no longer a problem.

### 1.3.2. Scipion's *plugin* structure

Usually, every plugin inside Scipion has the same structure in which the main features are organized:

- **Protocols:** contain the programs that will perform the actions specified
- **Viewers:** Allow the user to use the *Analyze results* button within Scipion's GUI to visualize results with programs such as VMD, Pymol or visualize tables
- **Wizards:** these special programs are designed to help the user in particular tasks that can not be performed in the protocols by themselves.

Moreover, protocols are the backbone of Scipion's framework. The protocols are defined by the creators as a processing task that involves the execution of several steps (*Creating a Protocol — Scipion 3.0.0 Documentation*, n.d.).

Protocols are created following an object-based programming logic, generating functions for a class that is normally derived from Scipion's EMProtocol class. Inside every class in Scipion's protocols, defined functions must be found:

- **Parameter functions:** *\_defineParams* function allows the developer to generate a form to be shown in Scipion's GUI in order to let the user choose between parameters for the protocol to work.
- **Step list:** *\_insertAllSteps* is a function that contains a call for each of the further functions to be executed according to the input parameters
- **Step functions:** Code that will be executed to perform an action

Besides those mentioned functions that are mandatory for the protocol to properly work, in each Scipion's protocol there are **Information and validations functions** that provide parameter validation and useful information in Scipion's GUI such as a summary of what the protocol has performed or information about where to look for more theoretical information.

### 1.3.3. Scipion's GUI

Regarding the Scipion's GUI (*Graphical User Interface*), the main window displayed by the framework present the different projects created by the user (*Figure 3*). When choosing a project, the Scipion's principal GUI is displayed as shown in *Figure 4*.

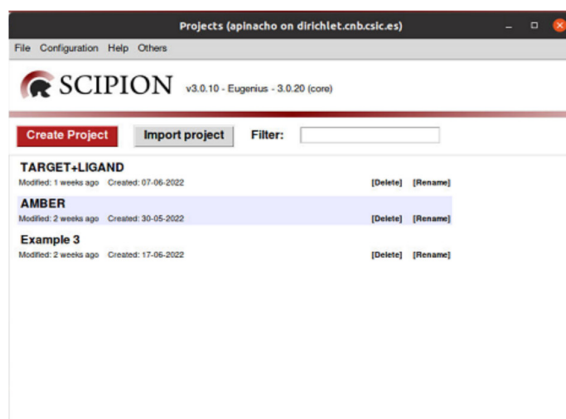
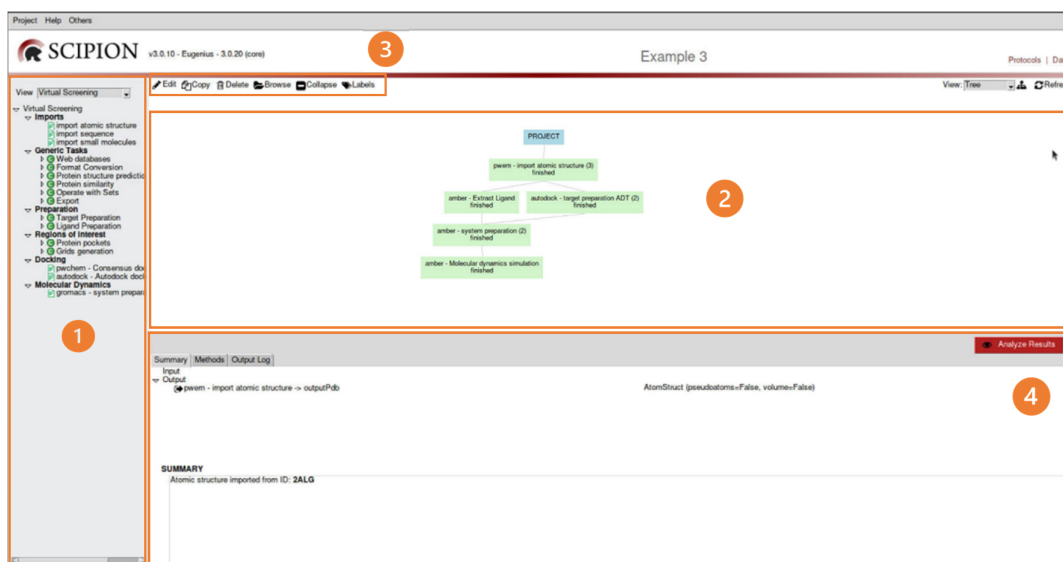


Figure 3. Scipion's main window

The left part of the window (*Figure 4.1*) is presented for the intuitive search of the different programs integrated within Scipion. However, protocols can also be searched with the command ctrl+F. When one of these protocols is chosen, a new window showing its options is displayed and in this same window the user can execute or save the protocol.

The chosen protocols are displayed in the central window of the Scipion's GUI (*Figure 4.2*), linked depending on the input they are using. For example, if the output of the first protocol is used as the input in another one, these two will be linked together in order to track the steps performed. In addition, protocols can be copied to create another branch inside the same project or re-executed changing the form parameters.



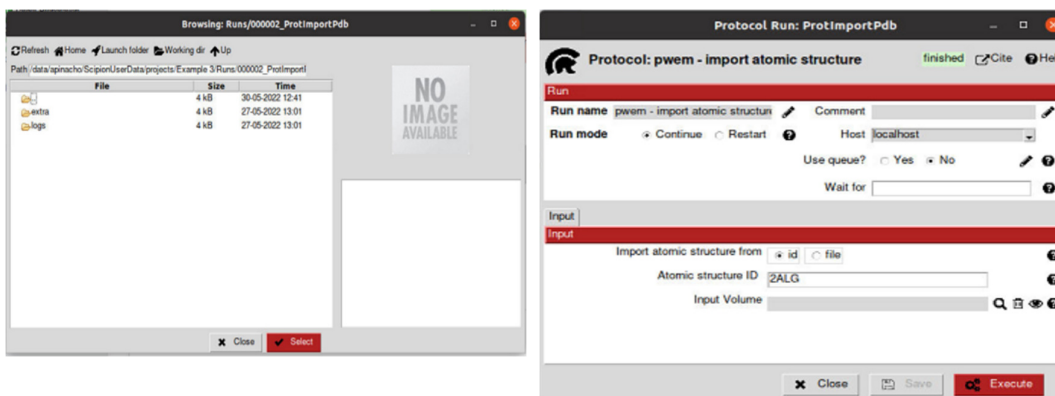
**Figure 4.** Scipion's main Project window

On the top left, different options about each protocol are displayed. Clicking on the browser button it is possible to navigate through the different folders created for each protocol (*Figure 4.3*). Usually, each protocol will have a main folder and an extra and temporal folder (*Figure 5, left*). The latter will be cleaned after the protocol has been completed.

When the protocol is executing the program, it will remain yellow until it has been completed. If the protocol was successfully performed, it will change to green color, if not, it will change to red color indicating that there was an error during the execution.

On the bottom half of the Scipion's GUI (*Figure 4.4*), information about the protocol execution is displayed. It is divided into three different windows: *Summary*, *Methods*, and *Output Log*. The first one shows a brief summary of the protocol, displaying which are the input and outputs and a quick explanation about what the protocol has performed. The second window shows the register file generated by Scipion and the last one gives the user information about how the protocol was executed and the possible errors occurred.

In addition, on the bottom right there is an *Analyze results* button. It allows the user to visualize the output of a given protocol if a viewer has been integrated for it (e.g Pymol, VMD).



**Figure 5.** Scipion's browser window (left) and example of Scipion's protocol form (right)



## 2. OBJECTIVES

Molecular dynamics studies have become one of the standard tools along with experimental tools in many research fields related to structural biology and drug discovery.

Since the beginning of Molecular Dynamics, generating protein simulation has been a complex task, especially for scientific researchers who do not especially for research scientists who do not have the bioinformatics skills to use programs that do not have a graphical interface and can only be used via command line.

In addition, although there are a variety of software packages for the creation of molecular dynamics simulations, there is not enough interoperability between them since each one uses its own input and output formats, making it difficult to compare results or even exchange data.

It is for this reason that the general objective of this Master Thesis is the **integration of a new *plugin* within the Scipion framework that contains the main features of AMBER Molecular Dynamics Software.**

Thus, thanks to this project, the following specific objectives will be achieved:

**Objective 1:** Uniting the main AMBER MD tools into a simple workflow with defined steps. These defined steps will follow the MD workflow, combining the different programs within AMBER MD that serve the same purpose into a protocol that facilitates the use.

**Objective 2:** Making the use of AMBER MD more accessible through Scipion's GUI by adding user-selectable parameters that automatically integrate the corresponding commands into the internal program.

**Objective 3:** Allowing researchers to keep track of the steps followed as well as the decisions made and the input and output files of each integrated program.

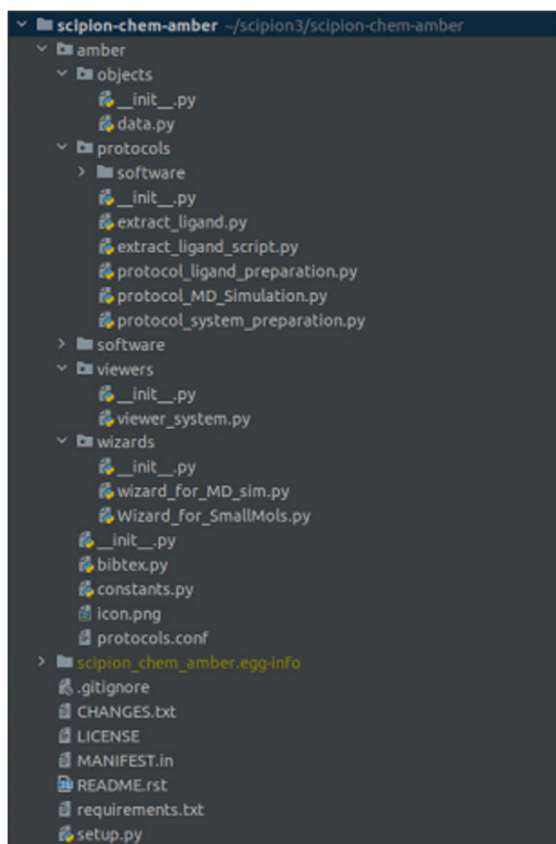
### 3. MATERIALS AND METHODS: *Plugin* development

As mentioned in section 1.3, Scipion was originally created as an image processing framework for the study of 3D models of Macromolecular Complexes. It is a platform that integrates several software packages and unifies them in the form of *plugins*, so the user can use more intuitively and in the same workflow.

The software package integrated is the free distribution of AMBER MD: Ambertoos21. This software is constantly being updated so this plugin must be updated too, as new actualizations are released.

#### 3.1. *Plugin* structure

In this way, the most notable features from AMBER MD Software have been integrated within *Scipion* as a new *plugin*. It has the following main structure shown in *Figure 6* and can be found at Scipion-chem's Github (<https://github.com/scipion-chem/scipion-chem-amber>).



*Figure 6. Scipion-chem-amber main structure*

This new plugin is organized into modules and submodules of Python that integrates all the scripts and files required in order to properly use Scipion, as well as all the protocols needed for an Amber MD simulation performance.

In the first place, files related to description, installation and requirements are found. **README.rst** file contains the instructions for the plugin correct installation and use, **requirements.txt** shows which modules are necessary for the plugin to work correctly and

that will be automatically installed, the **LICENSE** file shows how this plugin is governed by the GNU General Public License and finally the **setup.py** file its necessary for the correct generation and loading of the software package.

Then, other necessary files are found inside the amber submodule:

- **\_\_init\_\_.py**: this script is necessary inside every folder, and it defines the environment, variables and paths to the external programs used in this plugin.
- **data.py**: inside the objects folder, data.py contains the object *AmberSystem* use to record the inputs and outputs generated by the different protocols integrated. It also contains all the other objects such as *AtomStruct* and *SetOfSmallMolecules* that already exist within Scipion because they derive from *MDSystem*, another object already implemented in Scipion-chem, which represents a general molecular dynamics system.
- **Protocols**: explained in more detail in section 3.3, the protocols are the ones that enable the entire MD workflow to be conducted.
- **viewer\_system.py**: the file that allows Scipion to integrate visualization software such as Pymol and VMD for Amber systems. This software allows the user to visualize inputs and outputs during the workflow process.
- **wizard\_for\_MD\_sim.py**: script that allows the addition of different steps in the MD simulation protocol (section 3.3.5)
- **wizard\_for\_SmallMols.py**: script that enables the user to select one ligand from a *SetOfSmallMolecules*

### 3.2. Amber workflow within Scipion

As explained before, within the Scipion-chem-amber module, we find the amber submodule that contains information about the protocols that perform the actions of ligand extraction, system preparation and MD simulation, as well as other files that allow to integrate viewers such as Pymol or VMD and wizards that will allow to extract information from some protocols when necessary.

However, the backbone of a plugin are the protocols. These are where the software package of interest is called and all the commands necessary to perform an action are registered.

In this case, as *Ambertools21* main functionalities consist of system preparation and MD Simulation, both are the principal protocols in this plugin. Moreover, an extra protocol for the extraction of ligands from protein complexes was developed to facilitate this task to the user.

The workflow followed for the use of AMBER MD in Scipion is shown in *Figure 7*. Briefly, the workflow consists of first, obtaining the structure of a protein of interest and processing this file to extract necessary features such as the ligand if it is desired. Second, cleaning the original file and preparing a system compatible with the Amber software. Finally, the molecular dynamics simulation is carried out.

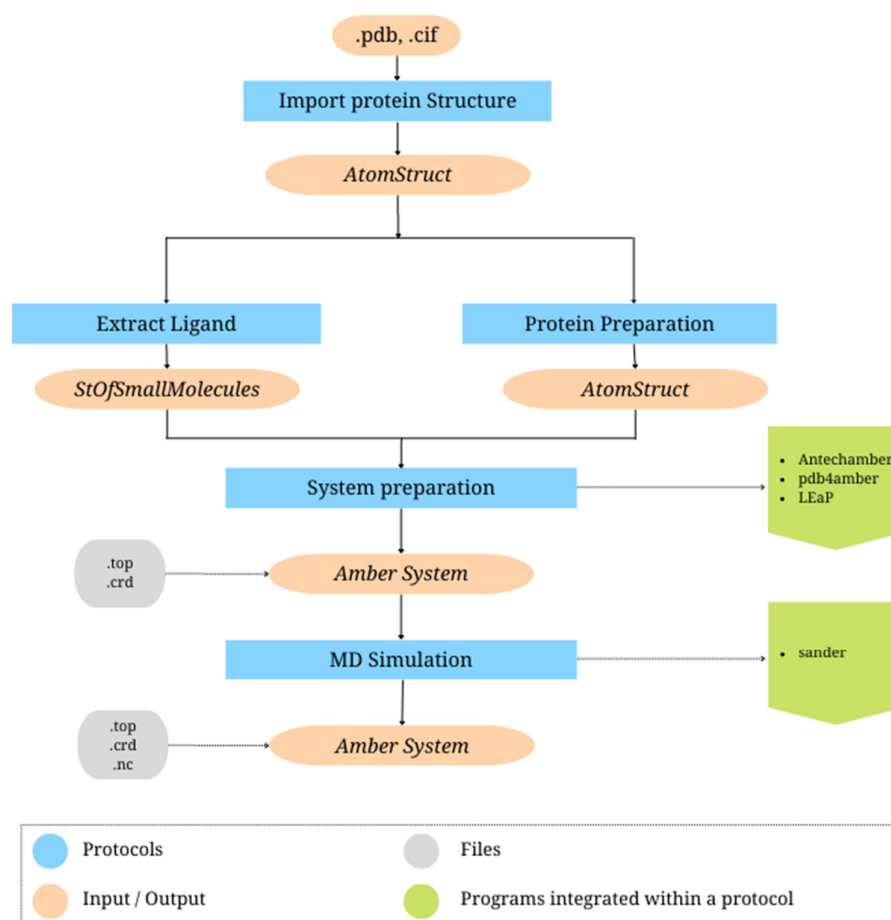


Figure 7. Workflow scheme integrated in Scipion for performing MD simulations with AMBER MD software.

### 3.3. Integrated protocols

#### 3.3.1. Import Atomic Structure

First, the structure of a protein, that may or may not contain a ligand, can be downloaded by using a protocol already integrated into Scipion called **Import Atomic Structure**. This protocol allows to obtain a protein structure from a local file or from the Protein Data Bank in .pdb or .cif format. Moreover, the protein structure obtained can be visualize by clicking on the *viewer* button in Scipion's GUI.

This protocol retrieves a Scipion object *AtomsStruct* that will be used as an input in the Extract Ligands and Target Preparation protocols (*Figures 8 and 9*)

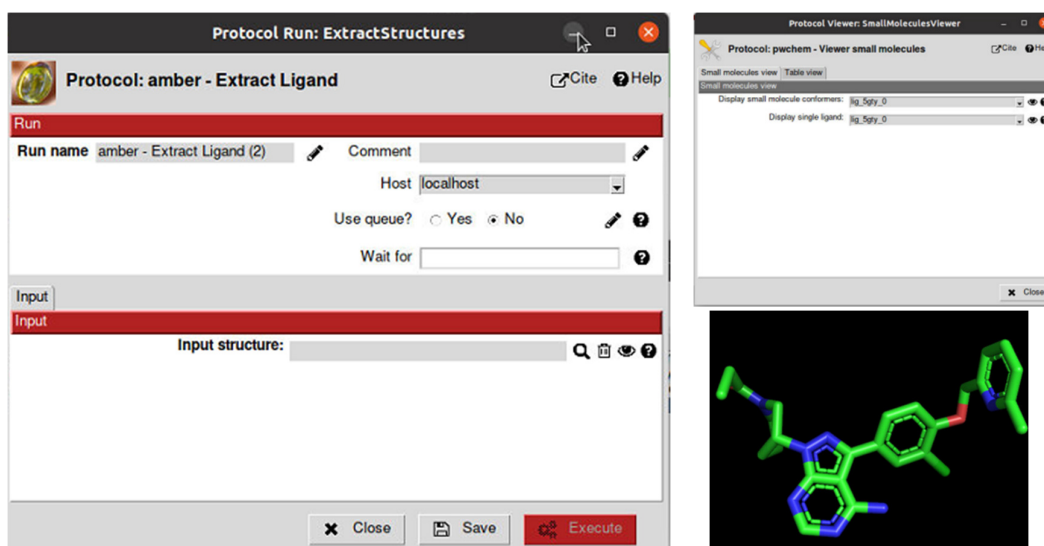
It is important to mention that for the correct use of Amber, the original pdb formats should be carefully examined before experiment. This is because some of the features integrated within Amber are not compatible with the pdb nomenclature. For example, all CYS atoms involved in disulfide bridges must be renamed by CYX, it is also mandatory to ensure that there is an END or TER record at the end of the file.

### 3.3.2. Extract Ligands

Then, if the user needs to use a ligand in the simulation, a protocol created for this plugin called **Extract Ligands** (Figure 8) is used in order to obtain the ligand(s) as *SetOfSmallMolecules*.

This protocol traverses the original pdb (if given in a .cif format, a function imported from the Scipion pwem module, *cifToPdb*, transforms it) and extracts only those HETATM lines that correspond to the ligands present in the file.

The *SetOfSmallMolecules* retrieved contains all the small molecules present in the original file. By clicking on the *viewer* button present in Scipion's GUI, the user can visualize the molecule chosen in Pymol Software.

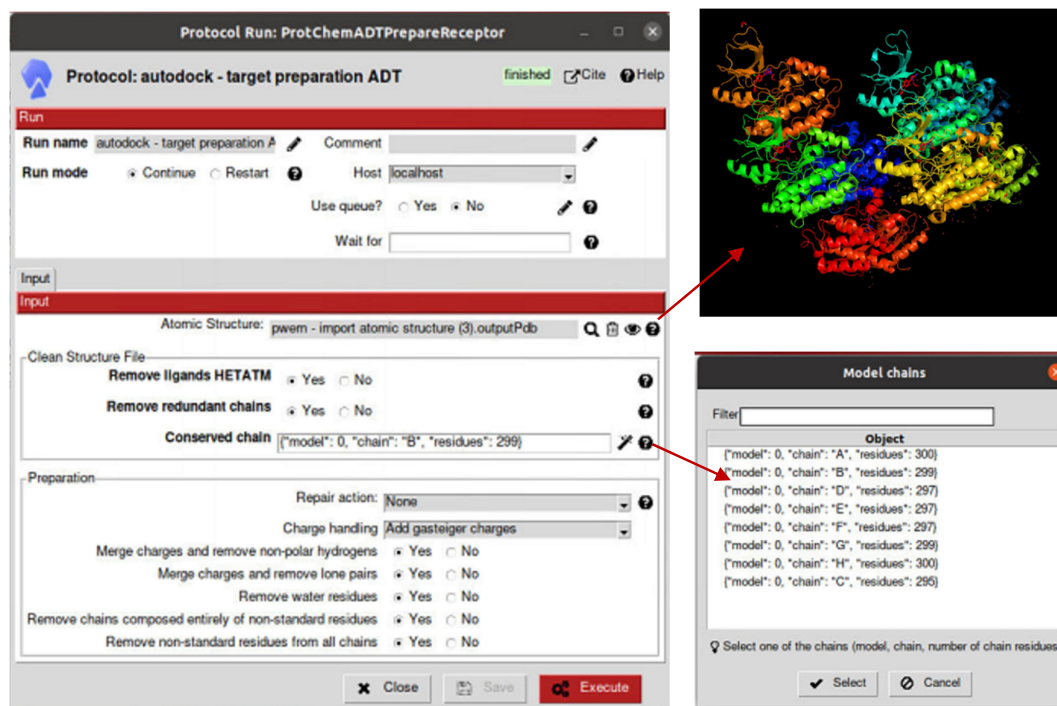


**Figure 8.** Protocol form Extract Ligands shown in Scipion's GUI (left) and an example of how it is presented the viewer system in this protocol (right)

### 3.3.3. Target/Protein Preparation

Furthermore, it is possible to combine these new protocols with an already implemented protocol that integrates the Autodock software to prepare the structure of the protein obtained. Among other functionalities, the user can eliminate water molecules, HETATM atoms or select a chain for use (Figure 9). In this protocol the user must choose an Atomic Structure already imported with the Import Atomic Structure protocol and choose between the different possible parameters.

This protocol also contains a *wizard* that allows the user to choose between different chains in the original file as shown below.



**Figure 9.** Protocol form Target preparation ADT from Autodock Software (right), example of viewer display of the entire input molecule (upper-right) and example of wizard display for different chain choosing (bottom-right).

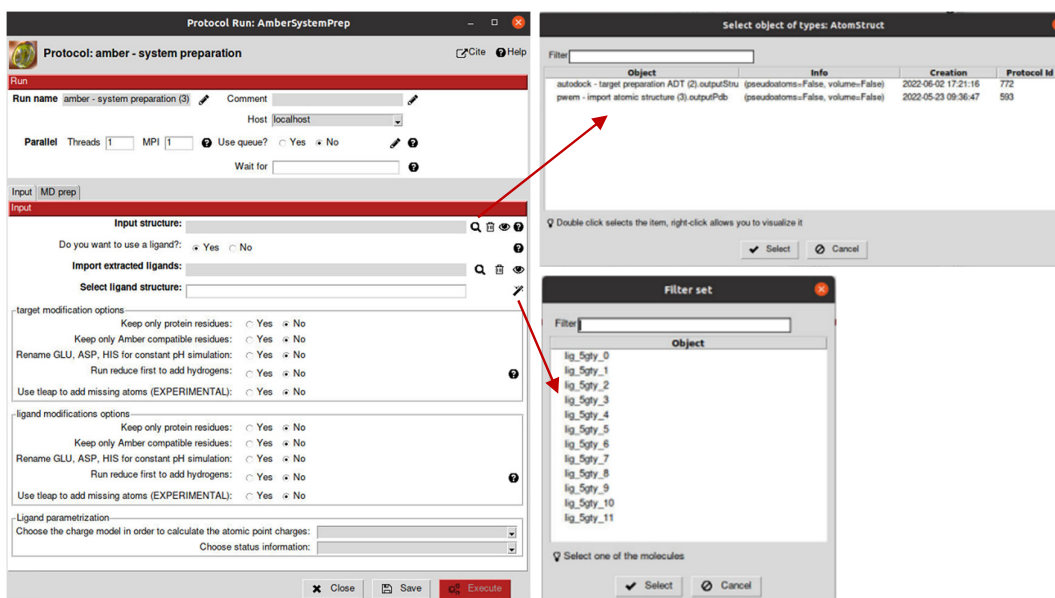
### 3.3.4. System preparation

This protocol integrates three of the main AMBER MD programs (**pdb4amber**, **LEaP** and **Antechamber**) and returns a Scipion object called *AmberSystem* that will contain the coordinates and topology of the system, as well as a file in .pdb format that will allow the user to visually check the characteristics of the system using Pymol software.

First, on the “input” tab the user must choose an *AtomStruct* object for the target protein, and if a ligand needs to be added, the user can also choose between the molecules present in a *SetOfSmallMolecules* (Figure 10, bottom left) previously obtained thanks to a wizard called *Wizard\_for\_SmallMols.py*.

Then, it is possible to modify the initial files using the **pdb4amber** program that allows, among other functionalities, to choose only the protein residues or add hydrogen molecules.

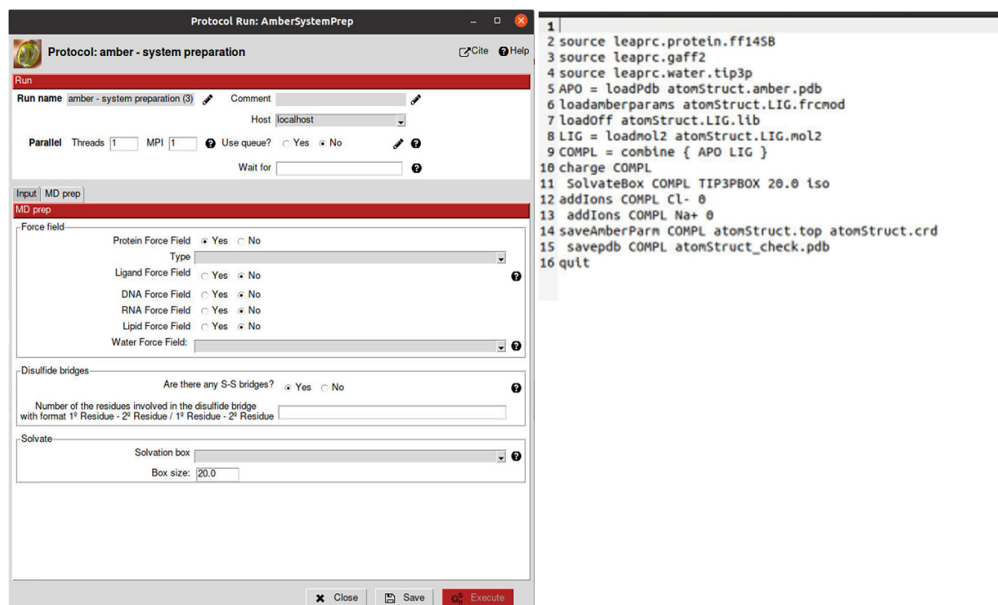
In this step, if a ligand was added to the study, the user must choose between different parameters for the ligand parametrization performed by the Amber program **Antechamber**. This program is designed to generate ligand topology and coordinate files that have to be used later by further preparation programs to identify bonds and atom types, judging atomic equivalence and find missing force field parameters if necessary.



**Figure 10.** Protocol form System Preparation input tab (left). Example of windows displayed for input choosing (right)

When the input parameters are filled, the user must click on the “MD prep” tab (Figure 11) in which they will choose between different parameters that will be written into a file called leap\_commands.txt. This file will work within the Amber program **LEaP**, used to generate topology and coordinate files that will be used in the simulation step.

The written file (Figure 11, right) runs the chosen commands on LEaP, loading information about the force fields present in the system, loading the input target and information about the parametrization of the ligand previously obtained from Antechamber. It also loads the information of the presence of disulfide bridges and combines both molecules in order to create a system that will be solvated and neutralized as the user has chosen to.



**Figure 11.** Protocol form system preparation MD prep tab (left). Example of an input command file for LEaP (right)

Finally, this preparation program generates and saves a topology (.top) and a coordinate (.crd) file and stores them into a Scipion's object called *AmberSystem*. Additionally, this program also creates a file named “\_check.pdb” that can be visualize by the Pymol viewer in this protocol.

### 3.3.5. MD Simulation

The final protocol created for this Amber-plugin within Scipion is the MD Simulation protocol (*Figure 12*). Here, the user can choose among different simulation parameters and finally visualize a trajectory in VMD Software.

This protocol is designed in such a way that simulation steps can be added. That is, the user can choose a series of parameters and add steps that will be run one after the other.

First, the user must choose whether the trajectory generated in this step is saved or not and if so, after how many picoseconds the information is recorded in the trajectory file. Additionally, the user can choose how long simulation should be and how long each step of the simulation will last.

Then, the user must choose which Ensemble to use: non-periodicity, NVT or NPT. These ensembles control whether periodic boundaries are imposed on the system during calculation of non-bonded interactions. Given these ensembles, the user can choose temperature and pressure, the thermostat and barostat that controls them and if the SHAKE algorithm will be used or not.

All these parameters chosen by the user are recorded into input files in different folders within Scipion framework and then processed by the Amber program **Sander**.

The image shows a screenshot of the Scipion protocol form for MD simulation. The window title is "Protocol Run: Amber/MDSimulation". The main form is titled "Protocol: amber - Molecular dynamics simulation". It contains several sections:

- Run name:** "amber - Molecular dynamics simulation".
- Host:** "localhost".
- Use queue?** "Yes" (selected).
- Wait for:** (empty field).
- Minimization:**
  - Input Amber System: (empty field)
  - Trajectory:
    - Save trajectory: "Yes" (selected)
    - Interval time (ps): "1.0"
  - Simulation time:
    - Simulation time (ps): "100.0"
    - Simulation time steps (ps): "0.002"
  - Ensemble:
    - Energy Minimization: "Yes" (selected)
    - Simulation type: "no periodicity"
    - SHAKE algorithm: "Shake not performed"
  - Summary:
    - Insert relaxation step number: (empty field)
    - Summary of steps: (empty field)
    - Delete relaxation step number: (empty field)
    - Watch relaxation step number: (empty field)
- Buttons:** "Close", "Save", "Execute".

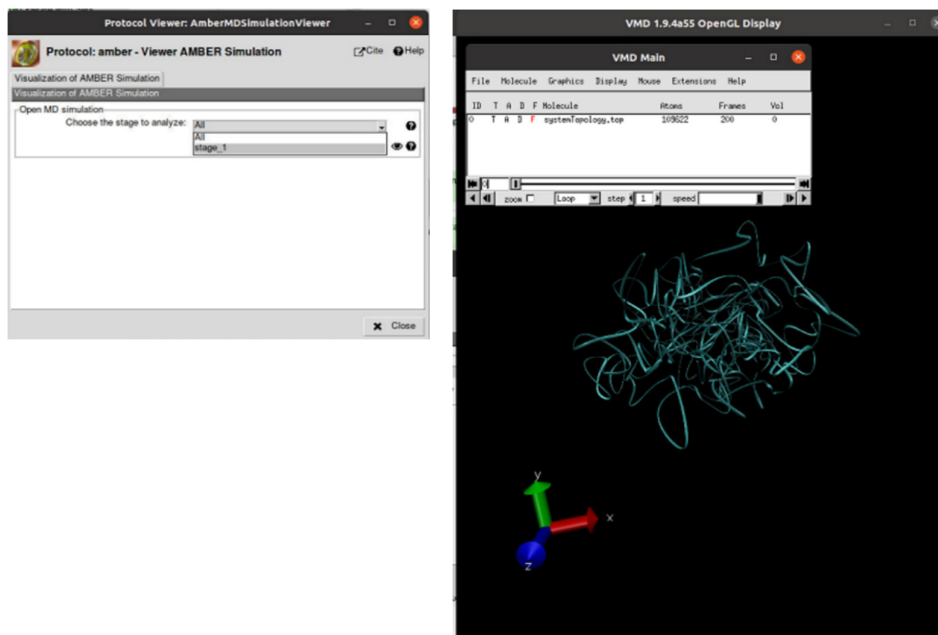
Below the main form, there are two detailed views of the Ensemble settings:

- Ensemble 1:**
  - Energy Minimization: "Yes" (selected)
  - Simulation type: "NPT"
  - Temperature settings: Temperature: "300.0", Thermostat: "Berendsen"
  - Pressure settings: Pressure (bar): "1.0", Pressure dynamics type: "No pressure scaling", Barostat type: "Berendsen"
  - SHAKE algorithm: "Shake not performed"
- Ensemble 2:**
  - Energy Minimization: "Yes" (selected)
  - Simulation type: "NVT"
  - Temperature settings: Temperature: "300.0", Thermostat: "Berendsen"
  - SHAKE algorithm: "Shake not performed"

**Figure 12.** Protocol form MD simulation showing the different parameters that the user must choose between



This protocol runs the simulation steps and finally returns a new Scipion's object *AmberSystem* that now contains a trajectory file. This file can be visualized in VMD by clicking on the viewer button in Scipion's GUI (*Figure 13*).



**Figure 13.** Example of the visualization of an *Ambersystem* trajectory using VMD Software

## 4. RESULTS – Practical examples of use

Cervical cancer (CC), a largely preventable disease which is diagnosed to more than 500.000 women worldwide each year (Cohen et al., 2019), is in most cases caused by high-risk subtypes of the human papilloma virus (HPV).

HPV types 16, 18 and 45 are found the most predominant in cervix carcinomas and its oncogenic potential is due to the presence of E6 and E7 oncoproteins. This oncogenic production leads to the triggering of alterations associated with the activation of p53 and pRB (Ohlenschläger et al., 2006). Specifically, it has been described that the E7 oncoprotein plays an especially important role in the generation of tumors (Hellner et al., 2009) due to conformational changes in its structure caused by different mutations. It is for this reason that studies on the structure of the different variants of this oncoprotein can elucidate how these mutations affect its morphology and how they affect its interaction with the environment to favor the appearance of tumors.

Taking as a reference the protocol followed in a research study in 2021 (Bello-Rios et al., 2021) for the molecular dynamics simulation of the oncoprotein E7 and its variants, this example shows a simulation obtained by using the protocols integrated within Scipion containing the AMBER MD software, with the aim of testing the usability of this program and its correct integration within the Scipion framework.

First, the structure of the E7 oncoprotein of the HPV45 variant (Ohlenschläger et al., 2006) was obtained by downloading a pdb file from the Protein Data Bank (PDB code: 2EWL). Prior to using this file within Scipion, it was manually edited to remove the information regarding repeated protein chains. Then, the steps of the workflow created for the use of the AMBER MD Software in Scipion were followed.

### I. Simulation system preparation

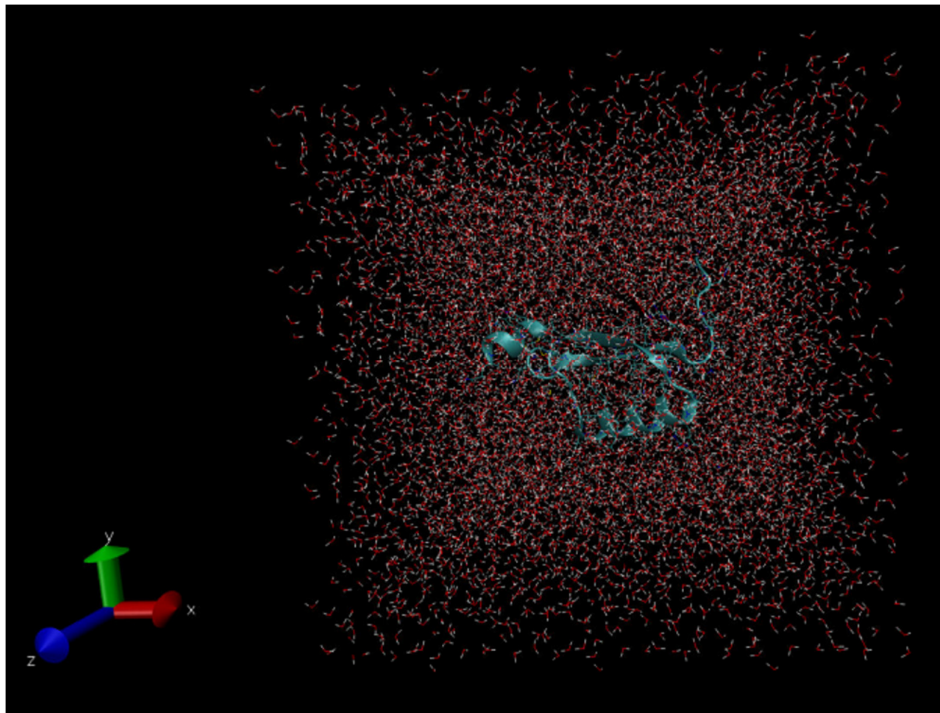
In this first step, the protocol *System Preparation* was used, and the different parameters shown in *Figure 14* were selected.

Figure 14. *System Preparation* protocol form for MD simulation of 2EWL

This protocol first uses **pdb4amber** program to clean the PDB format and remove any formation concerning where the protein comes from and information about the study that elucidated its structure to finally retrieve a file (2ewl.amber.pdb) compatible with the next amber program.

Next, in the form it is specified which type of protein and water force fields must be used and how large the solvation box will be. With this information, **LEaP** program is capable of generating the necessary parameters to return topology and coordinate files (2ewl.top and 2ewl.crd) which will be used in the next step of MD Simulation.

In addition, LEaP also generates a pdb format file which can be used to visualize how the system was built (*Figure 15*)



**Figure 15.** File 2ewl\_check.pdb visualized with MD Software

## II. MD Simulation

An *Amber system* object containing 2ewl.top, 2ewl.crd and 2ewl\_check.pdb files will be generated by the previous protocol and will be used as the input for the **MD Simulation** protocol.

During this protocol, different steps following the MD workflow (*Figure 1*) will be performed with the aim of first minimize or optimize the systems energy, equilibrate by gradually increasing the temperature and finally performing the simulation.

As shown in *Figure 16*, a total of 7 steps were performed:

- **Step 1 – Energy minimization:** This step is crucial for determining the proper molecular arrangement of the system. 1 ns was defined as simulation time, with a time step duration of 0.002 ps. Energy minimization was selected with constant pressure (1 bar) and temperature (300 K), and all bonds constrained using the SHAKE algorithm.
- **Steps 2-6 – Equilibrium:** the next steps are intended to move from when atoms of the macromolecule of study and of the surrounding solvent undergo a relaxation until the system reaches a stationary state. For this purpose, five steps that gradually increase the temperature of the system are performed. In each step, 1 ns of simulation time was defined, with a time step duration of 0.002 ps. In addition, NPT ensemble was selected with isotropic position scaling and 1 bar of pressure governed by Berendsen dynamics. The temperature was gradually increased in each step from 100 K to 300K employing Langevin thermostat and with all bonds constrained using the SHAKE algorithm.
- **Step 7 – MD Simulation:** This final step was performed without position restraints (SHAKE not performed) but following the same parameters as in the previous one. The simulation time was set for 50 ns with a time step duration of 0.001 ps.

**Figure 16.** MD Simulation protocol form for MD simulation of 2EWL

Finally, thanks to the viewer also integrated within Scipion, it is possible to visualize all stages of the total simulation. The final one represents the actual MD Simulation

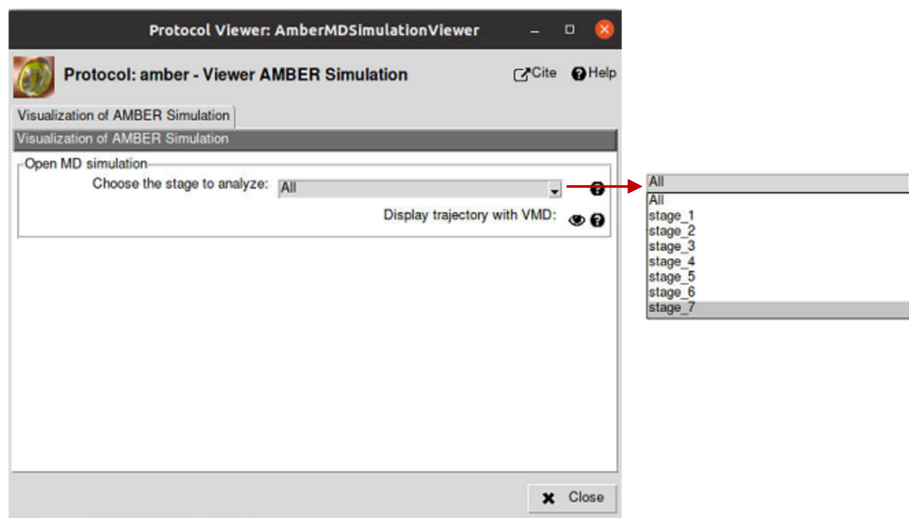


Figure 17. Viewer window display for MD Simulation protocol

Using Visual Molecular Dynamics Software, it is possible to visualize the obtained trajectory and perform a simple dynamic analysis via labels or study the dynamic RMSD in order to compare structural details. A video showing the simulation can be found at [this link](#).

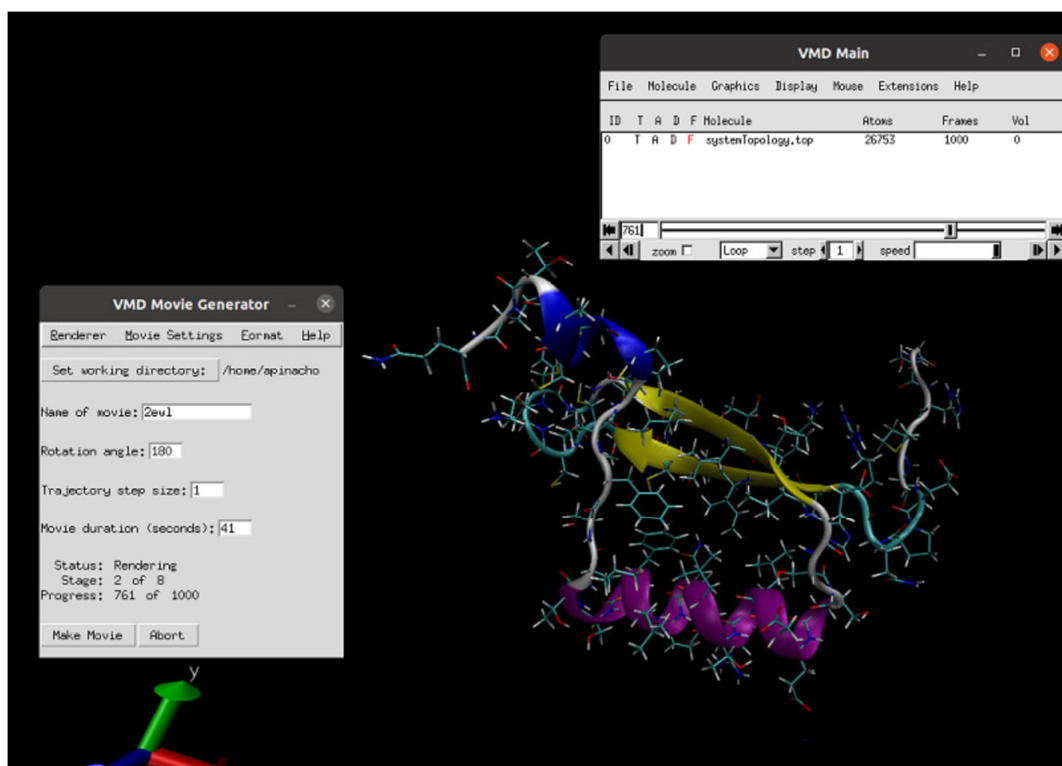


Figure 18. Visualization of 2EWL trajectory in VMD

## 5. CONCLUSIONS AND FUTURE WORK

Several conclusions can be drawn from the integration of AMBER MD Software into Scipion as a new plugin for performing MD simulations:

1. The integration of this new plugin allows to perform simple molecular dynamics simulations
2. Thanks to the integration of the Software within the Scipion-chem framework, it is now possible for new MD researchers to have easier access to a tool such as AMBER MD.
3. Although amber has a wide range of programs and within these, a variety of parameters to choose from, within Scipion the most important features have been integrated allowing a good understanding of the mandatory choices.
4. The creation of a simple workflow allows the user to save the chosen parameters as well as the files obtained in each protocol to be able to review each step if an error occurs.

Despite the fact that the integration of AMBER MD into Scipion has been successful, this software package has more functionalities that would be beneficial to integrate in the same way. For example, ccpraj is considered the main program for performing analysis of the simulations within AMBER. It would be beneficial to create a new protocol that allows different analyses such as RMSD or PCA calculation for the study and comparison of different structures.

## 6. BIBLIOGRAPHY

- Alder, B. J., & Wainwright, T. E. (1959). Studies in molecular dynamics. I. General method. *The Journal of Chemical Physics*, 31(2), 459–466. <https://doi.org/10.1063/1.1730376>
- Bello-Rios, C., Montaña, S., Garibay-Cerdenares, O. L., Araujo-Arcos, L. E., Leyva-Vázquez, M. A., & Illades-Aguiar, B. (2021). Modeling and Molecular Dynamics of the 3D Structure of the HPV16 E7 Protein and Its Variants. *International Journal of Molecular Sciences* 2021, Vol. 22, Page 1400, 22(3), 1400. <https://doi.org/10.3390/IJMS22031400>
- Childers, M. C., & Daggett, V. (2017). Insights from molecular dynamics simulations for computational protein design. *Molecular Systems Design and Engineering*, 2(1), 9–33. <https://doi.org/10.1039/C6ME00083E>
- Cohen, P. A., Jhingran, A., Oaknin, A., & Denny, L. (2019). Cervical cancer. *The Lancet*, 393(10167), 169–182. [https://doi.org/10.1016/S0140-6736\(18\)32470-X](https://doi.org/10.1016/S0140-6736(18)32470-X)
- Congreve, M., Murray, C. W., & Blundell, T. L. (2005). Keynote review: Structural biology and drug discovery. *Drug Discovery Today*, 10(13), 895–907. [https://doi.org/10.1016/S1359-6446\(05\)03484-7](https://doi.org/10.1016/S1359-6446(05)03484-7)
- Creating a protocol — Scipion 3.0.0 documentation*. (n.d.). Retrieved June 20, 2022, from <https://scipion-em.github.io/docs/docs/developer/creating-a-protocol>
- D.A. Case, H.M. Aktulga, K. Belfon, I.Y. Ben-Shalom, J.T. Berryman, S.R. Brozell, & D.S. Cerutti. (2022). *Amber 2022 Reference Manual Principal contributors to the current codes*. <https://ambermd.org/contributors.html>
- de la Rosa-Trevín, J. M., Quintana, A., del Cano, L., Zaldívar, A., Foche, I., Gutiérrez, J., Gómez-Blanco, J., Burguet-Castell, J., Cuenca-Alba, J., Abrishami, V., Vargas, J., Otón, J., Sharov, G., Vilas, J. L., Navas, J., Conesa, P., Kazemi, M., Marabini, R., Sorzano, C. O. S., & Carazo, J. M. (2016). Scipion: A software framework toward integration, reproducibility and validation in 3D electron microscopy. *Journal of Structural Biology*, 195(1), 93–99. <https://doi.org/10.1016/J.JSB.2016.04.010>
- Feig, M., Nawrocki, G., Yu, I., Wang, P. H., & Sugita, Y. (2018). Challenges and opportunities in connecting simulations with experiments via molecular dynamics of cellular environments. *Journal of Physics. Conference Series*, 1036(1). <https://doi.org/10.1088/1742-6596/1036/1/012010>
- Ferreira, L. G., dos Santos, R. N., Oliva, G., & Andricopulo, A. D. (2015). Molecular Docking and Structure-Based Drug Design Strategies. *Molecules* 2015, Vol. 20, Pages 13384-13421, 20(7), 13384–13421. <https://doi.org/10.3390/MOLECULES200713384>
- Geng, H., Chen, F., Ye, J., & Jiang, F. (2019). Applications of Molecular Dynamics Simulation in Structure Prediction of Peptides and Proteins. *Computational and Structural Biotechnology Journal*, 17, 1162–1170. <https://doi.org/10.1016/J.CSBJ.2019.07.010>
- Guvench, O., & MacKerell, A. D. (2008). Comparison of protein force fields for molecular dynamics simulations. *Methods in Molecular Biology*, 443, 63–88. [https://doi.org/10.1007/978-1-59745-177-2\\_4/COVER/](https://doi.org/10.1007/978-1-59745-177-2_4/COVER/)

- Hansson, T., Oostenbrink, C., & van Gunsteren, W. F. (2002a). Molecular dynamics simulations. *Current Opinion in Structural Biology*, *12*(2), 190–196. [https://doi.org/10.1016/S0959-440X\(02\)00308-1](https://doi.org/10.1016/S0959-440X(02)00308-1)
- Hansson, T., Oostenbrink, C., & van Gunsteren, W. F. (2002b). Molecular dynamics simulations. *Current Opinion in Structural Biology*, *12*(2), 190–196. [https://doi.org/10.1016/S0959-440X\(02\)00308-1](https://doi.org/10.1016/S0959-440X(02)00308-1)
- Hedges, L. O., Mey, A. S. J. S., Laughton, C. A., Gervasio, F. L., Mulholland, A. J., Woods, C. J., & Michel, J. (2019). *BioSimSpace: An interoperable Python framework for biomolecular simulation*. <https://doi.org/10.21105/joss.01831>
- Hellner, K., Mar, J., Fang, F., Quackenbush, J., & Münger, K. (2009). HPV16 E7 oncogene expression in normal human epithelial cells causes molecular changes indicative of an epithelial to mesenchymal transition. *Virology*, *391*(1), 57–63. <https://doi.org/10.1016/J.VIROL.2009.05.036>
- Hollingsworth, S. A., & Dror, R. O. (2018). Molecular Dynamics Simulation for All. *Neuron*, *99*(6), 1129–1143. <https://doi.org/10.1016/J.NEURON.2018.08.011>
- Hospital, A., Goñi, J. R., Orozco, M., & Gelpí, J. L. (2015). Molecular dynamics simulations: advances and applications. *Advances and Applications in Bioinformatics and Chemistry*, *8*(1), 37–47. <https://doi.org/10.2147/AABC.S70333>
- Karplus, M., & McCammon, J. A. (2002). Molecular dynamics simulations of biomolecules. *Nature Structural Biology*, *9*(9), 646–652. <https://doi.org/10.1038/NSB0902-646>
- Leimkuhler, B., & Matthews, C. (2015). *Molecular Dynamics*. *39*. <https://doi.org/10.1007/978-3-319-16375-8>
- Levitt, M., & Lifson, S. (1969). Refinement of protein conformations using a macromolecular energy minimization procedure. *Journal of Molecular Biology*, *46*(2), 269–279. [https://doi.org/10.1016/0022-2836\(69\)90421-5](https://doi.org/10.1016/0022-2836(69)90421-5)
- Lifson, S., & Waeshel, A. (1968). Consistent force field for calculations of conformations, vibrational spectra, and enthalpies of cycloalkane and n-alkane molecules. *The Journal of Chemical Physics*, *49*(11), 5104–5107. <https://doi.org/10.1063/1.1670007>
- McCammon, J. A., Gelin, B. R., & Karplus, M. (1977). Dynamics of folded proteins. *Nature*, *267*(5612), 585–590. <https://doi.org/10.1038/267585A0>
- Norman, G. E., & Stegailov, V. v. (2013). Stochastic theory of the classical molecular dynamics method. *Mathematical Models and Computer Simulations*, *5*(4), 305–333. <https://doi.org/10.1134/S2070048213040108>
- Nussinov, R., Tsai, C. J., & Liu, J. (2014). Principles of allosteric interactions in cell signaling. *Journal of the American Chemical Society*, *136*(51), 17692–17701. [https://doi.org/10.1021/JA510028C/ASSET/IMAGES/LARGE/JA-2014-10028C\\_0005.JPEG](https://doi.org/10.1021/JA510028C/ASSET/IMAGES/LARGE/JA-2014-10028C_0005.JPEG)
- Nussinov, R., Tsai, C.-J., Shehu, A., Jang, H., Jagodzinski, F., & Chen, B. Y. (n.d.). *molecules Computational Structural Biology: Successes, Future Directions, and Challenges*. <https://doi.org/10.3390/molecules24030637>



- Ohlenschläger, O., Seiboth, T., Zengerling, H., Briese, L., Marchanka, A., Ramachandran, R., Baum, M., Korbass, M., Meyer-Klaucke, W., Dürst, M., & Görlach, M. (2006). Solution structure of the partially folded high-risk human papilloma virus 45 oncoprotein E7. *Oncogene* 2006 25:44, 25(44), 5953–5959. <https://doi.org/10.1038/sj.onc.1209584>
- Patodia, S. (2014). Molecular dynamics simulation of protein Physical Chemistry & Biophysics. *J Phys Chem Biophys*, 4, 6. <https://doi.org/10.4172/2161-0398.1000166>
- Ponder, J. W., & Case, D. A. (2003a). Force Fields for Protein Simulations. *Advances in Protein Chemistry*, 66, 27–85. [https://doi.org/10.1016/S0065-3233\(03\)66002-X](https://doi.org/10.1016/S0065-3233(03)66002-X)
- Ponder, J. W., & Case, D. A. (2003b). Force Fields for Protein Simulations. In *Advances in Protein Chemistry* (Vol. 66, pp. 27–85). Academic Press. [https://doi.org/10.1016/S0065-3233\(03\)66002-X](https://doi.org/10.1016/S0065-3233(03)66002-X)
- Roy, K., Kar, S., & Das, R. N. (2015). Computational Chemistry. *Understanding the Basics of QSAR for Applications in Pharmaceutical Sciences and Risk Assessment*, 151–189. <https://doi.org/10.1016/B978-0-12-801505-6.00005-3>
- Weiner, P. K., & Kollman, P. A. (1981). AMBER: Assisted model building with energy refinement. A general program for modeling molecules and their interactions. *Journal of Computational Chemistry*, 2(3), 287–303. <https://doi.org/10.1002/JCC.540020311>
- Yoneya, M., Hirasawa, K., & Berendsen, H. J. C. (1994). A non-iterative matrix method for constraint molecular dynamics simulations. *Molecular Simulation*, 13(6), 395–405. <https://doi.org/10.1080/08927029408022001>
- Zhao, H., & Caflisch, A. (2015). Molecular dynamics in drug design. *European Journal of Medicinal Chemistry*, 91, 4–14. <https://doi.org/10.1016/J.EJMECH.2014.08.004>