MOLECULAR BIOINFORMATICS

Tutorials

Asclepius, Greek god of medicine, with its distinctive snake-entwined rod, nowadays used as a the symbol of medicine and pharmacy. Archaeological Museum of Epidaurus (photo by Michael F. Mehnert).

João Marques Sousa • Matilde Ribeiro Viegas • Pedro Teixeira Ferreira • Pedro Coelho Paiva • Ana Luísa Oliveira • Pedro Alexandrino Fernandes • Maria João Ramos

Departamento de Química e Bioquímica, Faculdade de Ciências, Universidade do Porto, 2021

DOI: 10.24927/rce2021.075

INDEX

PREFACE

Bioinformatics fuses the two areas of Science that have had the most disconcerting advances in the late 20th and early 21st centuries. On the one hand, our societies are governed by the unprecedented communicative and organizational power that information technology provides. On the other hand, biology, by unveiling many of life's secrets, allows human beings to overcome disease, prolong health, extend life, and manipulate living beings, to a level that is even frightening, typical of "gods" "in the eyes of a man of antiquity.

The fusion power of those two areas is immense, but to be materialized, they must first be merged in the scientist's mind. In other words, the same scientist must simultaneously master biology and information technology in order to reveal the true potential of this merger.

This set of tutorials accompanies the book *Molecular Bioinformatics - basic concepts*, which has been made available also to our students of Bioinformatics. Both of them focus on a specific area of bioinformatics - molecular bioinformatics. In this area, the object of study is molecules, with all their atoms explicitly represented, and simulated by the laws of physics, sometimes more accurately, sometimes more closely. All the "molecules" studied are part of living beings and are fundamental elements of their physiology. They are what we call "biological molecules". We intend to simulate them using powerful computer tools, and thus obtain answers to important biological problems.

The concrete case that illustrates these studies are snake venom molecules, more specifically vipers. The toxin under study is one of its main components, the enzyme phospholipase A2. Throughout the semester, the student will learn a set of bioinformatics techniques that will allow him to study the toxin, understand its mode of action, and develop corresponding antidotes, illustrating the immense potential of molecular bioinformatics to solve complex biological, medical and social problems.

We hope, with these tutorials, to awaken the molecular bioinformatician that is within all of you.

1. HOMOLOGY MODELLING TUTORIAL

This tutorial describes, step by step, how to obtain the 3D structure of a protein using the homology modelling technique. As an example, we will build the basic secreted phospholipase A2 (sPLA2) enzyme of the highly venomous Central/South American pitviper terciopelo (***Bothrops asper***) (Figure 1).

Figure 1. Pitviper Terciopelo(***Bothrops asper***). Photo by Aurélien Salesse.

Before starting the homology model protocol, it is imperative to know some information regarding the protein of interest. In our case, we know that our enzyme of interest, sPLA2, requires, for its catalytic activity, a calcium ion coordinated by a catalytic aspartate, and a water molecule near a catalytic histidine. This information will influence the choice of the template for the construction of the homology model.

I. The first step of any homology modelling protocol involves finding the primary structure of the protein of interest (***i.e.*** sequence of amino acids). For that, we will use the **UniProt** database (https://www.uniprot.org).

   A. On the search engine of the homepage, you can search for the name of your protein of interest on the **UniProtKB** (Uniport Knowledge Base).
   B. On the search results, you should look for your protein of interest and if it belongs to the intended organism. If that is the case, click on the entry code in the second column of the results table.
   C. On the page of the protein that is opened, you will find diverse information about your protein. Explore this page to see the information it contains.

The degree of detail of the information varies depending if the protein is more or less studied. In that page, it is possible to obtain the sequence of the protein on the **FASTA**[1] format that you will need for the next step.

II. In the second step of this protocol, you will search for other proteins whose sequences exhibit homology[2] relatively to your protein of interest and possess three-dimensional structure(s) deposited on the **Protein Data Bank**. To do that, we will use the **SWISS-MODEL** web server (swissmodel.expasy.org). This server allows the search for homologous proteins giving as input solely the amino acid sequence of the target protein. Optionally, the user can specify up to 5 structures to be used as templates. In this work we will only use the first option.

A. On the server's home page click in "**Start Modelling**".
B. Paste the **FASTA** sequence of your protein of interest on the "**Target Sequence**" rectangle. Optionally, it is possible to paste, instead, the **UniProtKB** code of the protein of interest.
C. Give a title to your homology modelling job and, preferentially, an e-mail. The advantage of giving an e-mail address is to obtain and maintain an easy access to the results that you will get.
D. Click on "**Search for Templates**". The server will search in the Swiss Model Library of Templates (**SMTL**) for homologous sequences and, then, align them using **BLAST**[3], which is a sequence similarity search program that can be used to quickly search a sequence database for matches to a query sequence, and **HHblits**[4]. **HHblits** first converts the query sequence (or **MSA**) to a Hidden Markov Model (**HMM**), which is a condensed representation of the **MSA** that specifies for each sequence position the probability of observing each of the 20 amino acids in evolutionarily related proteins. **HHblits** then searches the **HMM** database and adds the sequences from **HMMs** below a defined expected value (**E value**) threshold to the query **MSA**, from which the **HMM** for the next search iteration is built.
E. The server will return a list with the available templates ordered by the predicted quality of the resulting model (Figure 2). The quality is evaluated

[1] Text format used to represent sequences of nucleotides or amino acids, in which these are identified by a single letter.

[2] Two proteins are said to be homologous when they share a common ancestor.

[3] Camacho, C., Coulouris, G., Avagyan, V., Ma, N., Papadopoulos, J., Bealer, K. and Madden, T.L. (2009) BLAST+: architecture and applications. BMC Bioinformatics, 10, 421-430.

[4] Remmert, M., Biegert, A., Hauser, A., and Soding, J. (2012). "HHblits: lightning-fast iterative protein sequence searching by HMM-HMM alignment.", Nat Methods 9, 173-175.

by the quality estimator **GMQE**[5] and, if the target protein is predicted to be an oligomer, by the quality estimator **QSQE**[6]. If, in the results table, any template exhibits 100% (or close to 100%) of shared sequence identity with the target protein, it means that there is a crystallographic structure for the target sequence. If that is the case for your target protein, for learning purposes, do not choose the crystallographic template; choose instead the template with the highest ranking that does not correspond to a tri-dimensional structure of your target.

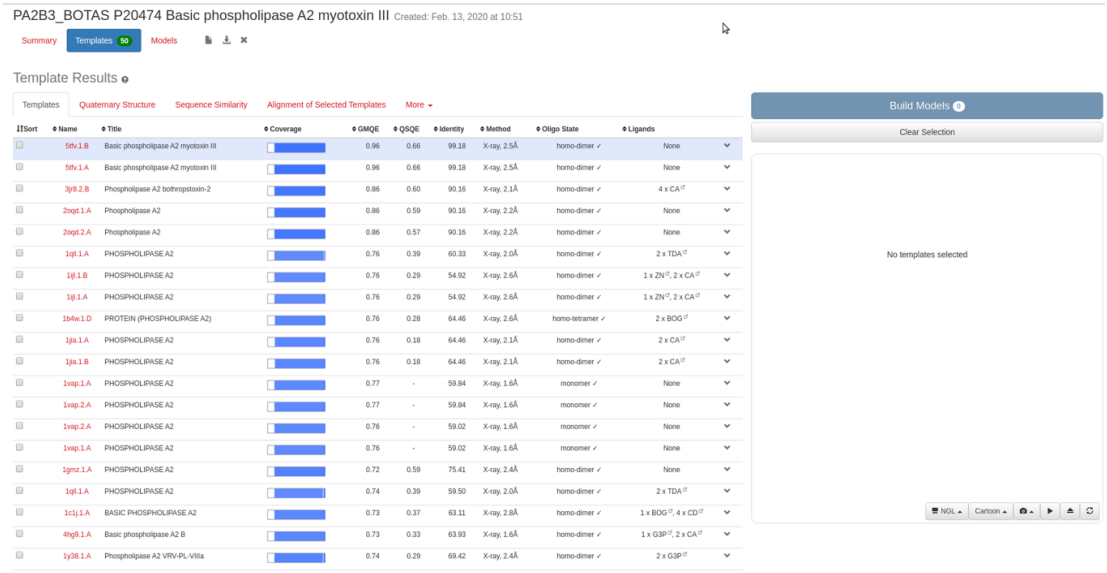


Figure 2. Table showing a list of the templates available for the protein basic sPLA2 Myotoxin I of the viper terciopelo. The first two are crystallographic structures of the target protein of a different isoform of the target protein and that is why the shared identity is not exactly 100%. For each template, the following information is provided:

- A checkbox to select and visualise the template in the 3D panel: “Sort”
- The SMTL ID of the template and a link to the SWISS-MODEL Template Library page associated to that SMTL entry: “Name”
- The protein name of the template “Title”
- The coverage[7] to the target sequence (darker shades of blue refer to higher sequence identity) “Coverage”
- The GMQE (Global Model Quality Estimation)
- The QSQE (Quaternary Structure Quality Estimation)

---

[5] (Global Model Quality Estimation) is a quality estimation which combines properties from the target–template alignment and the template search method. The resulting GMQE score is expressed as a number between 0 and 1, reflecting the expected accuracy of a model built with that alignment and template and the coverage of the target. Higher numbers indicate higher reliability on the predicted tertiary structure.

[6] The QSQE score is a number between 0 and 1, reflecting the expected accuracy of the interchain contacts for a model built based a given alignment and template. In general a higher QSQE is "better", while a value above 0.7 can be considered reliable to follow the predicted quaternary structure in the modelling process. It is calculated as: (is_full_biounit, bin, gmqe + qs_value), where: is_full_biounit is only used for heteromers and is set to 1, if all chains from the template biounit are included for modelling, or 0 otherwise; bin is computed as ceil((gmqe - max_gmqe) / 0.1), where max_gmqe is the best gmqe observed in the templates; gmqe is the GMQE of the template; qs_value is set to QSQE of the template, if the target model is predicted to be an oligomer, or 0 otherwise.

[7] Indicates the percentage of the target sequence that is present on the homology model template.

- The target–template sequence identity: “Identity”
- The experimental method used to determine the structure (and resolution, if applicable): “Method”
- The oligomeric state of the SMTL biounit: “Oligo-State”
- The ligands present in the experimental structure (if any): “Ligands”
- A clickable arrow to expand the box with the description of the template

F. To choose an adequate template to build your model it is important to know some crucial information regarding your target protein. Regions that may be fundamental for the protein’s function (so they have to be present on the template’s structure) or ligands/cofactors that may be needed for catalysis. In the example given in figure 2, we know that our target protein requires Ca^{2+} coordinated by the catalytic aspartate for catalysis. So, the best template would be that which combines a high sequence similarity with the presence of crystallographic calcium ions on the active site. The best candidate for our example seems to be the 3rd template on the list, corresponding to the Phospholipase A2 bothropstoxin-2 of the (also) highly venomous South American pitviper jararacussu (***Bothrops jararacussu***) that exhibits a very high sequence identity to the target[8] and high GMQE score. Another aspect to take into account is the resolution at which the template was crystallized, which is recommended to be better than 2.2 Å. In addition, it has crystallographic calcium ions present. However, we have to verify if the catalytic aspartate coordinates a calcium ion, as it should in the enzyme active form. The catalytic aspartate is usually either in position 48 or 64 of the catalytically active PLA2, depending whether the sequence contains an initial signal peptide or not. It is always followed by two cysteines so, in the sequence, one can look for a “**DCC**” triad. To visualize the 3D structure of the template in the webpage you have to mark it on the “Sort” column, and unmark any other that may be marked. Then, click on the arrow in the last column of the templates table to expand the line corresponding to your template. You can see additional information regarding the selected template, and the sequence of your target protein and selected template aligned (Figure 3). In, the sequence of the template (the second one) search for the catalytic aspartate, look for a “**DCC**” pattern in position 48 or 64. Then, click on the aspartate (“**D**” letter), and, on the 3D representation on the right, the program will zoom in on the aspartate, and you can see if it is coordinating a calcium ion or not. In this case, in our template, the catalytic aspartate is not coordinating any of the calcium ion (Figure 3), so we will discard this template and find a more suitable one down below the list.

[8] Something that is not surprising as they belong to the same taxonomic genus.

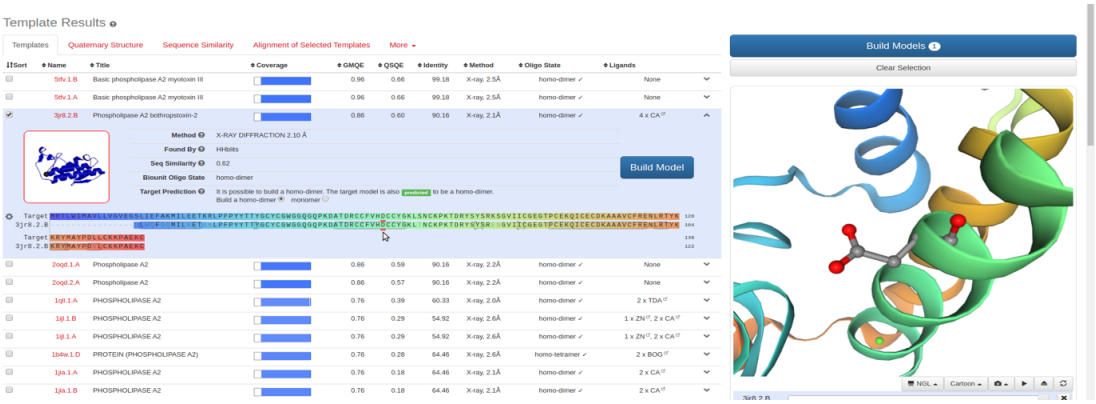


Figure 3. Expanded information on the Phospholipase A2 bothropstoxin-2 of the snake jararacussu (***Bothrops jararacussu***) and 3D representation of its catalytic aspartate. It is clear that the catalytic aspartate is not coordinating a calcium ion as it is required for our enzyme of interest to be catalytically active. The additional information that it is possible to get by expanding the line of the template in the results table is:

- The method by which the 3D structure was determined
- Which algorithm found this template
- The sequence similarity of the template to the target, calculated from a normalised BLOSUM62[9]substitution matrix. The sequence similarity of the alignment is calculated, as the sum of the substitution scores divided by the number of aligned residue pairs. Gaps are not taken into account.
- The oligomeric state of the model that can be built using this template.

G. In our example, the most suitable template would be the 10th in the Figure 2 template list, corresponding to the Phospholipase A2 of the venomous pitviper Halys Viper (***Agkistrodon halys***). This template shares 64.46% of our target protein and exhibits a high **GMQE** value. Moreover, the catalytic aspartate is coordinating the calcium ion in this template (Figure 4).

Figure 4. Extended information and 3D representation of the Phospholipase A2 of the Halys Viper (***Agkistrodon halys***) with a zoom in its catalytic aspartate coordinating a calcium ion. The calcium ion is represented as a green sphere.

[9] Henikoff S. and Henikoff J. G. (1992) Amino acid substitution matrices from protein blocks. Proc Natl Acad Sci USA, 89(22): 10915-10919.

H. When the adequate template is chosen click on the clickable arrow to expand the box with the description of the template, in "**Target Prediction**" select "**Monomer**" to instruct the server to build the monomer only, and click in "**Build Model**". The choice of the adequate oligomeric state of the model varies for each enzyme and objective of each work; in this tutorial we will work with the monomer but that may not be adequate for other works.

I. After approximately a minute, you can click in "**Models**" on the top of the page and your resulting model should appear (Figure 5).

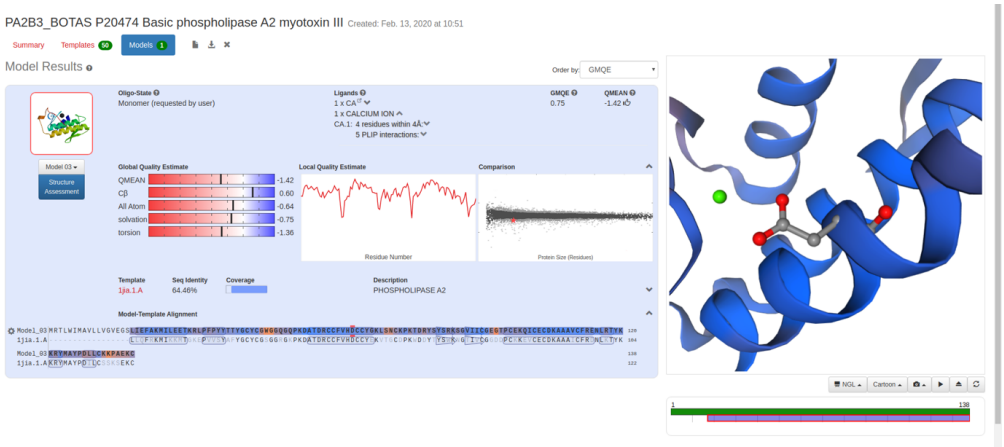


Figure 5. Model Results page example. The following information is provided:

- A file containing the model coordinates along with relevant information on the modelling process
- The oligomeric state of the model
- The modelled ligands
- QMEAN[10] model quality estimation results
- The target–template sequence alignment
- The template name (SMTL ID)
- The sequence identity to the target
- The target sequence coverage

J. Save the model to your workspace, in PDB format, in a folder called 1_Homology_Modelling.

---

[10] QMEAN is an estimator based on different geometrical properties that provides both global (for the entire structure) and local (per residue) absolute quality estimates on the basis of one single model. The QMEAN score is an estimate of the degree of nativeness of the structural features, it indicates if the QMEAN is comparable to what is expected from experimental structures of similar size. Thus, a QMEAN score around 0 indicate a good agreement between model and experimental structures of similar size. Scores of -4.0 or below indicate models of low quality.
The QMEAN is calculated based on four individual items shown in Figure 3 on the Global Quality Estimate Panel. The white regions, on the plots, which are near 0 indicates that the property is similar to what one would expect from experimental structures of similar size. Positive values indicate that the model scores higher than experimental structures on average. Negative numbers indicate that the model scores lower than experimental structures on average. The QMEAN score is shown on top of the panel.
The "Local Quality" plot in Figure 3 shows, for each residue of the model (x-axis), the expected similarity to the native structure (y-axis). Residues exhibiting a score below 0.6 are estimated to be of low quality.
In the Comparison Plot of Figure 3, the model scores are compared to scores of experimental structures of similar size. Each dot represents the QMEAN score (y-axis) of an experimental structure of a specific size (x-axis).

III. In the third step of this protocol, we will visualize our resulting model. In addition, as said at the beginning of this tutorial, our enzyme also requires a water molecule near its catalytic histidine. The catalytic histidine occupies the previous position of the catalytic aspartate in the enzyme sequence. So, in our case, because the catalytic aspartate occupies position 64, the catalytic histidine occupies position 63. The homology model obtained does not contain water molecules, so we will have to model the catalytic water manually from our template.

A. If your target protein has an x-ray structure, download it from the PDB, and put it on the same folder of your model. If not, advance for step III.C

B. Open both your model, and the x-ray structure of your target enzyme in **VMD**. Align both structures using **Extensions > Analysis > RMSd Calculator**. If the x-ray is not a monomer, align them using 'chain A' as reference (Figure 6). If the x-ray is a monomer align them using 'protein' as reference. Click in align and calculate the RMSd between the two structures. You will see that both structures are almost superimposable and that the RMSd value stays around 1Å. Which indicates that both structures are very similar. This is a further indication that the homology model is reliable.

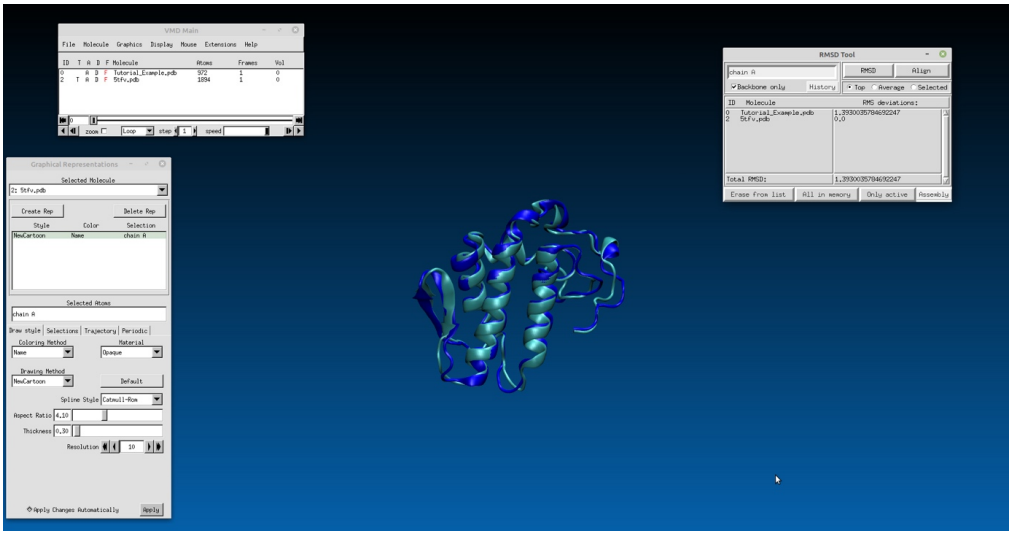


Figure 6. Alignment of the homology model built, and its x-ray structure.

C. To model the needed catalytic water, first download the PDB structure of the template that you used to build your homology model. In our example case, it was PDB code: 1jia. Load it into VMD, and create a representation of the active site, where you can see the calcium ion, the catalytic aspartate and histidine and the catalytic water. For example, "**chain A and same residue as all within 8 of resname CA**". This selection will return all residues within a radius of 8 Å from calcium. Identify the catalytic water and write down its number. In our example the water molecule has a "**resid**" number 207 (Figure 7).

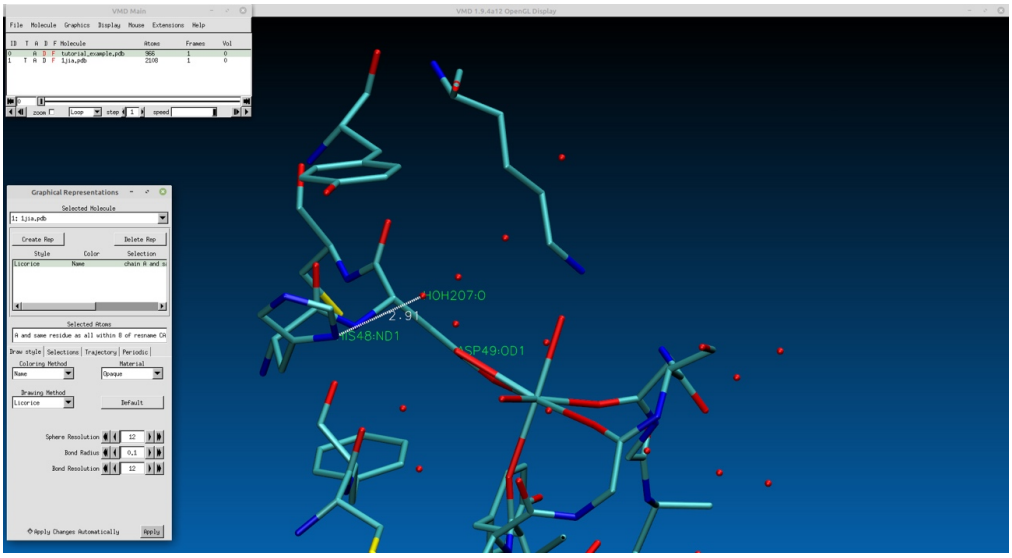


Figure 7. Identification of the catalytic water on the template's crystallographic structure.

D. Open both the template and the homology model pdb files with a text editor, or, if you are comfortable working with it, with vim. Search for the coordinates of the catalytic water and copy the whole line into the bottom of the homology model pdb file, just above the "**END**" mark. Enter a new line before the one you inserted, with the coordinates of the water molecule, and write a "**TER**" mark (Figure 8).

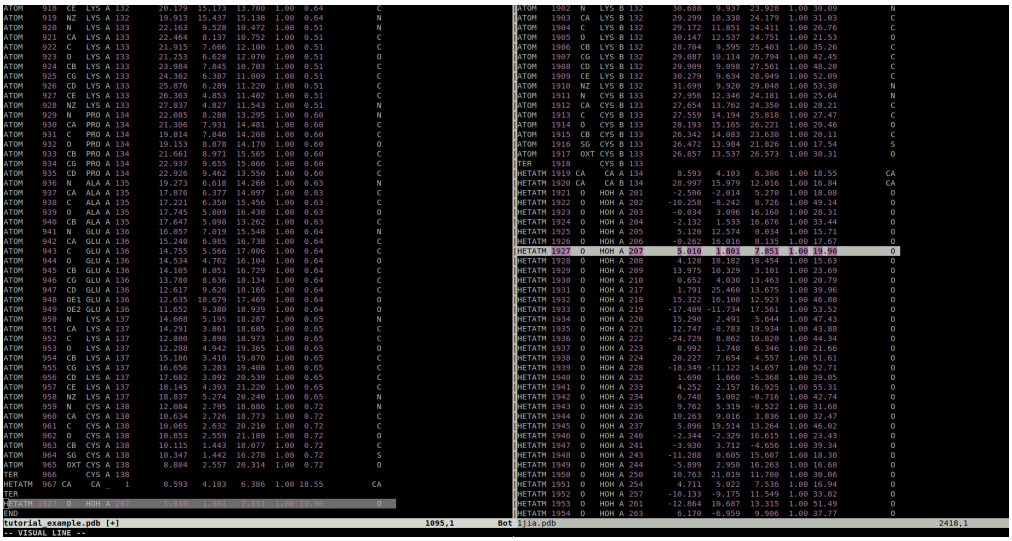


Figure 8. Insertion of the catalytic water coordinates from the template's pdb file into the homology model PDB.

E. Visualize your homology model structure, making sure that the catalytic aspartate is coordinating the calcium ion, and the catalytic water is near the catalytic histidine and the catalytic aspartate.

GEOMETRY OPTIMIZATION OF PROTEINS.

In the final step of this tutorial, we have to optimize the geometry of our pdb model. Therefore, here, we present the steps required to perform an energy minimization (also called *geometry optimization*[11]) of a protein using **Amber**, a suite of biomolecular simulation software. This process approximates the molecular system to the closest local energy minimum, relaxing it from existing tensions and clashes that resulted from excessively close or misdirected atoms. Energy minimization is a necessary step when the structure under study was originated from homology modelling, or from other forms of computational modelling. However, first of all we have to prepare the input pdb file.

IV. Preparation of the input PDB file

In this step, you will use the ***pdb4amber*** script to remove information that is written in the original PDB file emanating from the Protein Databank or from molecular modelling, and that is unnecessary for the calculations, preparing it to be used with the **Amber** software package.

A. Create a folder and name it "**2_Mins**". Copy the PDB file of the homology model/crystallographic structure into the newly created folder
B. Open the folder "**2_Mins**", right-click with the mouse on a blank area and select "**Open in Terminal**"
C. In the Terminal window, write the following command:

```
pdb4amber -i YourPDBname.pdb -o YourPDBname_amber.pdb
```

Note: Change the "**YourPDBname**" by the name of the PDB of your homology model/crystallographic structure

D. The ***pdb4amber*** script will generate 4 files:
    1. **YourPDBname_amber_nonprot.pdb** – a PDB file that contains information about the nonproteic part of your system (if present).
    2. **YourPDBname_amber_renum.txt** – a text file that contains the old (original) and new numbering of each residue.
    3. **YourPDBname_amber_sslink** – a text file that compiles a list of all disulfide bonds of your protein. Each line corresponds to a disulfide bond established between the two cysteine residues whose numbers (given in the new numbering) are specified. Take note of each disulfide bond and the numbers of the cysteine residues, as they will be needed later.

[11] In mathematics, the optimization of a function consists in the search of the points for which the derivatives of the function are zero. These points can be either minima or maxima. When a geometry optimization searches the molecular structures for which the energy is a minimum in relation to all the atomic coordinates, the process is usually called an "energy minimization".

4. **YourPDBname_amber.pdb** – the processed PDB file. Open it using a text editor and remove:
    a) the lines before the "**ATOM 1**" line. They are unnecessary for the calculation.
    b) the lines that start with "**CONECT**" (located at the end of the file). These lines define the bonds between atoms in a PDB file, but these bonds will now be defined by the **Amber** force field.

    Carefully check if the calcium ion is named **CA**, if the catalytic water molecule is named **WAT**, and if the PDB file finishes with an "**END**" word. If the homology modelling tasks were correctly done this should be the case, but it's safer to double check. Save the file after the changes.

V. Generation of the topology and parameter files

In this section, you will use the **XLEaP** module of **Amber** to create a protein system in physiologic conditions: add all hydrogens atoms to the system (they are missing in the Protein Databank and homology modelling files), define the disulfide bonds, add solvent water molecules, and add the counterions that make the overall system neutral. Finally, **XLEaP** will generate the topology and parameter files of the final system. To do so, you will need the **PDB** file generated by ***pdb4amber***, which must be located in the folder "**2_Mins**".

A. In the "**2_Mins**" folder, right-click on a blank space, select "**Create Document**" and then "**Empty File**". Name the new file "**leaprc**". This file will contain the instructions that will be read by the ***XLEaP*** module.

   Most of you have already used ***XLEaP*** in the previous course of **Computational Biochemistry** and are familiar with it. In that occasion you typed all the instructions directly onto the ***XLEaP*** window. Here you will take a step further: you will write the instructions carefully in a file and ask ***XLEaP*** to read the file with all the instructions. This allows you to easily create more complex sets of instructions.

B. Using a text editor, open the "**leaprc**" file and write the following information (the lines that start with a "#" are comments for yourself, to help you organize your work, and will be ignored by ***XLEaP.*** Additionally, note that **0** is zero):

C. You should be already familiar with the commands used in all subsections, except the one in subsection #3, as they repeat tasks that you have already done in the previous course of **Computational Biochemistry**. In the

```
#1 Load parameter libraries
source leaprc.gaff
source leaprc.protein.ff14SB
source leaprc.water.tip3p
loadamberparams frcmod.ionsjc_tip3p

#2 Load the PDB file
prot = loadpdb YourPDBname_amber.pdb

#3 Establish the disulfide bonds.
bond prot.bbb.SG prot.zzz.SG
bond prot.yyy.SG prot.vvv.SG
bond …

#4 Add counterions to neutralize the charge
addions prot Na+ 0
addions prot Cl- 0

#5 Add a 12.0 Å radii solvent box, using the TIP3P parameters for water molecules
solvateBox prot TIP3PBOX 12.0

#6 Generate topology (inpcrd) and parameter (prmtop) files
saveamberparm prot YourPDBname_amber.prmtop YourPDBname_amber.inpcrd
```

subsection #3 of the "**leaprc**" file, you will need to indicate all disulfide bonds that ***XLEaP*** needs to create. Each line that starts with "bond" concerns one specific disulfide bond that will be established. Therefore, you should consider the information written in the **YourPDBname_amber_sslink** file as herein explained:

If a line with "***26 115***" is present in the "**sslink**" file, you will need to add the line "***bond prot.26.SG prot.115.SG***" to the subsection #3 of the "**leaprc**" file. This command orders ***XLEaP*** to establish a bond between the gamma sulfur atom (SG) of the 26th residue and the SG atom of the 115th residue of the PDB file loaded in #2. This needs to be performed for each line in the "**sslink**" file.

D. Open the ***XLEaP*** module by running the following command on the Terminal window:

Note: The instructions present in the "**leaprc**" file are automatically read by ***XLEaP*** if the file is located in the same folder where the ***XLEaP*** is launched.

```
xleap
```

E. Carefully check the information that is presented in the "**XLEaP: Universe Editor**" window. If all the instructions were correctly followed and the parameter/topology files were successfully created, no errors should be reported by ***XLEaP*** and the following output should be printed:

```
  >
 (no restraints)
>
```

Note: If ***XLEaP*** quits immediately after its launch, it means that the instructions present in the "**leaprc**" file are incorrect. Open the "**leap.log**" file with a text editor and check the final lines of the file, which should inform you of the specific problem that led to the unexpected shutdown of ***XLEaP***.

F. You may quit ***XLEaP*** by entering the following command in the ***"*XLEaP: Universe Editor**" window:

```
quit
```

VI. Visualization of the system

Open the system in VMD and carefully check if the hydrogen atoms were added, if the catalytic water molecule is properly oriented, and if the counterions and solvent are indeed present.

Note: to upload your system onto VMD, you must upload the following two files, one after the other – **YourPDBname_amber.prmtop** (with amber7 parm of VMD) and **YourPDBname_amber.inpcrd** (with amber7 Restart of VMD).

VII. Energy minimization protocol

Before performing a Docking calculation or an MD simulation, it is important to optimize the geometry to relax the structure of the system, eliminating clashes and unfavorable interactions and torsions, that result from imperfections in the modelling. The energy minimization protocol encompasses four consecutive steps in which the constraints exerted on the system are gradually removed. In the first step, only the water molecules are optimized. In the second step, the hydrogen atoms are also allowed to move freely. The third minimization step consists on the relaxation of

the side chains and in the fourth and final step, no constraints are applied to the system.

As you already know, an energy minimization is a process where the software will move all the atoms to try to reduce its total potential energy until it reaches a local energy minimum. All energy minimizations will be performed using the ***Sander*** module of the **Amber** suite. A list and explanation of all keywords used in the input files of the energy minimizations is presented below:

&cntrl	Initiate the variables block to use in Minimizations / MD. The following lines that contain variables should initiate with a space	
imin	Flag to run minimization	= 0 (Default) - Run MD without any minimization
		= 1 – Perform an energy minimization
maxcyc	Maximum number of minimization cycles	
ncyc	Indicates the number of minimization cycles with steepest descent algorithm. If ncyc < maxyc, the remaining cycles will be performed with the conjugated gradient algorithm (which is more accurate near the minima)	
ntb	Controls whether or not periodic boundaries are imposed on the system	= 0 – No periodicity is applied
		= 1 – Periodic system (constant volume)
		= 2 – Periodic system (constant pressure)
cut	Specify the cut radius for nonbonded interactions (in Å)	
ntr	Flag for restraining specified atoms in Cartesian space using a harmonic potential, if ntr > 1	
restraint_wt	The weight (in kcal·mol^{-1}·Å^{-2}) for the positional restraints	
restraintmask	String that specifies the restrained atoms when ntr = 1	
ntxo	Format of the final coordinates, velocities and box size written to the .rst file	= 1 – Formatted (ASCII)
		= 2 (default) – NetCDF file
/	Indicates the end of the variables block	

A. Energy Minimization n. 1 – Water Molecules

1. In the "**2_Mins**" folder, create a new file named "**Min1_water.in**", open it with a text editor and write the following information:

```
Min1_Water_Minimization
 &cntrl
 imin = 1,
 maxcyc = 500,
 ncyc = 250,
 ntb = 1,
 cut = 10,
 ntr = 1,
 ntxo = 1,
 restraint_wt = 50.0,
 restraintmask = ':*&!:WAT'
 /
```

2. Launch the first minimization by running the following command in the Terminal window:

```
mpirun -n 4 sander.MPI -O -i Min1_water.in -o Min1_water.out  -r Min1_water.rst
-p YourPDBname_amber.prmtop -c YourPDBname_amber.inpcrd
-ref YourPDBname_amber.inpcrd
```

B. Energy Minimization n. 2 – Hydrogen Atoms

1. In the "**2_Mins**" folder, create a new file named "**Min2_H.in**", open it with a text editor and write the following information:

```
Min2_Hydrogen_Minimization
 &cntrl
 imin = 1,
 maxcyc = 500,
 ncyc = 250,
 ntb = 1,
 cut = 10,
 ntr = 1,
 ntxo = 1,
 restraint_wt = 50.0,
 restraintmask = ':*&!(:WAT|@H=)'
 /
```

2. Launch the second minimization by running the following command in the Terminal window:

```
mpirun -np 4 sander.MPI -O -i Min2_H.in -o Min2_H.out  -r Min2_H.rst
-p YourPDBname_amber.prmtop -c Min1_water.rst -ref Min1_water.rst
```

C. Energy Minimization n. 3 – Protein side chains

1. In the "**2_Mins**" folder, create a new file named "**Min3_sidechains.in**", open it with a text editor and write the following information:

```
Min3_Side_Chains_Minimization
 &cntrl
 imin = 1,
 maxcyc = 500,
 ncyc = 250,
 ntb = 1,
 cut = 10,
 ntr = 1,
 ntxo = 1,
 restraint_wt = 50.0,
 restraintmask = '@CA,C,N,O&!:WAT'
 /
```

2. Launch the third energy minimization by running the following command in the Terminal window:

```
mpirun -np 4 sander.MPI -O -i Min3_sidechains.in -o Min3_sidechains.out
-r Min3_sidechains.rst -p YourPDBname_amber.prmtop -c Min2_H.rst -ref Min2_H.rst
```

D. Energy minimization n. 4 – All the system

1. In the "**2_Mins**" folder, create a new file named "**Min4_all.in**", open it with a text editor and write the following information:

```
Min4_Whole_System_Minimization
 &cntrl
 imin = 1,
 maxcyc = 500,
 ncyc = 250,
 ntb = 1,
 cut = 10,
 ntxo = 1,
/
```

2. Launch the final minimization by running the following command in the Terminal window:

```
mpirun -np 4 sander.MPI -O -i Min4_all.in -o Min4_all.out -r Min4_all.rst
-p YourPDBname_amber.prmtop -c Min3_sidechains.rst -ref Min3_sidechains.rst
```

E. Visual inspection of the minimized structure

The minimized structure can be visualized using the VMD software (do not forget that you have to upload the two files, one after the other – **YourPDBname_amber.prmtop** (with amber7 parm of VMD) and **Min4_all.rst** (with amber7 Restart of VMD). It is necessary to compare the structural changes between the homology model/crystallographic structure and the minimized one. You should pay special attention to the active site, particularly to the catalytic water molecule, the catalytic histidine and the residues that coordinate the calcium ion.

As you compare the two structures, keep in mind the following questions:

- **i.** Is the overall folding of the minimized structure similar to the original one?
- **ii.** Does the catalytic water remain nearby the catalytic histidine?
- **iii.** Is there any change in the coordination sphere of the calcium ion?

Finally, extract a **pdb** file from VMD by exporting your system with ***File > Save Coordinates*** and name it **sPLA2_H_wat.pdb**

FURTHER READING

Fiser, A. and A. Sali, Modeller: generation and refinement of homology-based protein structure models. Methods Enzymol, 2003. 374: p. 461-91.

2. MOLECULAR DOCKING TUTORIAL

1. Background.

Molecular docking (D) is a computational technique that predicts the preferred position and orientation of one molecule in relation to another, when bound together to form a stable complex, also known as molecular ***pose***. The docking technique, albeit with different characteristics for the different cases, can predict interactions between two proteins, protein and ligand or even protein and DNA. We will only analyse here protein-ligand docking.

There is much molecular docking software that has been successfully used in a myriad of keystone problems, however, as commonly happens with most of the scientific software, the programs are often complex and a deep knowledge is required for the common user to carry out standard steps. This is an obstacle and a cornerstone issue for the research teams in the fields of Chemistry and the Life Sciences, who are interested in conducting this kind of calculations but do not have enough programming skills. To overcome these limitations, we have designed **VsLab** (virtual screening lab), an easy-to-use graphical interface for the well-known molecular docking software **AutoGrid/AutoDock** ((http://autodock.scripps.edu/) that we have included into **VMD** as a **plug-in**. This program allows almost anyone to use AutoDock and AutoGrid for simple docking or for virtual screening campaigns without requiring any deep knowledge on these techniques. The potential associated to this software makes it an attractive choice also for more advanced users that can use VsLab to increase workflow and productivity of everyday tasks.

The scientific work behind **VsLab** has already been discussed and validated by external peer reviewers, and an article containing this software has been published in the International Journal of Quantum Chemistry (Cerqueira NMFSA, Ribeiro J, Fernandes PA, Ramos MJ, ***vsLab-An Implementation for Virtual High-Throughput Screening Using AutoDock and VMD***, Int. J. Quantum Chem., 111, 1208-1212, 2011).

.

We are going to carry out a molecular docking calculation in order to predict the binding pose of a typical ligand in the active site of an enzyme (the receptor). In this particular case, the receptor is the basic secreted phospholipase A2 (sPLA2) enzyme of the Central/South American pitviper terciopelo (***Bothrops asper***) that has been already studied in the previous tutorials of this course and of which you already determined the 3D structure. The ligand is a good inhibitor (IC_{50} = 1.30 nM; to check on the experimental set up used to measure the IC_{50} please have a look at the paper by Hagishita et al., ***Potent inhibitors of secretory phospholipase A2: synthesis and inhibitory activities of indolizine and indene derivatives***. J Med Chem, 39, 3636-58 (1996)) of the human PLA2 enzyme, ***i.e.*** 1-Aminooxalyl-3-(2-benzyl-benzyl)-2-ethyl-indolizin-8-yloxy]-acetic acid.

2. Protein and Ligand input files for VsLab.

2.1. The protein.

In order to use the protein structure for the docking process, we need to prepare the PDB structure of our file **sPLA2_H_wat.pdb**:

a) Remove all the molecules that are alien to the protein structure. In our particular case we do not have such molecules.

b) All the remaining water molecules should also be deleted from the structure as they may occupy some space in the active site that might be required to bind the ligand. However, even if it is not our case, the deletion of the water molecules might sometimes introduce inaccuracies in docking protocols, as sometimes there are water molecules that mediate interactions between the protein and the ligand and are thus needed for the correct binding of the ligand. Please refer to the publication recommended 'Bioinformática Molecular – conceitos fundamentais' to read about such a case.

c) All missing hydrogen atoms must have been added and the structure optimized. We have in fact done that in the previous tutorial.

Please name the resulting pdb file, **sPLA2.pdb**

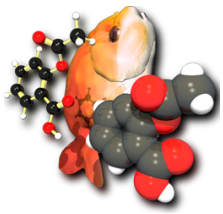
2.2. The ligand.

The 3D-structure of the ligand (IUPAC name: **2-[2-Ethyl-1-oxamoyl-3-[[2-(phenylmethyl)phenyl]methyl]indolizin-8-yl]oxyacetic acid**), can be obtained from the **PubChem** database:

https://pubchem.ncbi.nlm.nih.gov/compound/10389981

The resulting file can be named: **ligand1.mol2** to be coherent with the virtual screening calculations that we will carry out in the next tutorial as it corresponds to the first ligand of the ligands database that we will be working with then. However, before you can use the file that you will save from PubChem (a file in the format "sdf"), you have to transform it into a file in the format "mol2", which is the file format accepted by Autodock.

The 3D structure of all the individual compounds present in the file with sdf extension can be obtained using the open source software called "openbabel", which can be found at:



www.openbabel.org

This software allows us to collect all the molecules available in the file with the sdf format and save it in individual files with mol2 format, needed for VsLab.

The software openbabel can be used from a linux terminal. The **options** of this application can be obtained writing in the terminal window the following command:

```
obabel -H
```

In order to convert the sdf file into a mol2 files the following command can be used:

```
obabel -i sdf ligand.sdf  -o mol2 -O ligand.mol2
```

where ligand.sdf is the name of the file with the sdf extension that you obtained from the PubChem database, -o is a flag that allows the user to choose the format of the output files (pdb, mol2, xyz, etc., depending in what the user writes next to the flag -o) and -m a flag that that indicates the division of the structures of the sdf file in individual files.

Other options are available that might be important to prepare the files. Some are given below. Each option is introduced after the corresponding flag:

openbabel options	**Description**
- p <pH>	Add hydrogens appropriate for this pH Example: obabel database.sdf -o mol2 -m output.mol2 **-p 7**
-f <#>	Start import at molecule # specified.
-l <#>	End import at molecule # specified. Example: obabel database.sdf -o mol2 -m output.mol2 -p 7 **-f 50 -l 60**

Note: In the sdf file, each compound has a given name. Unfortunately openbabel does not allow us to save the output files with the name of the compound that is present in the sdf file. This means that the name of the compounds have to be included manually by the users, if they so wish; obviously, it is not necessary. The index/order of the files is the same that is present in the sdf file.

3. Generating the input Files for AutoDock and AutoGrid.

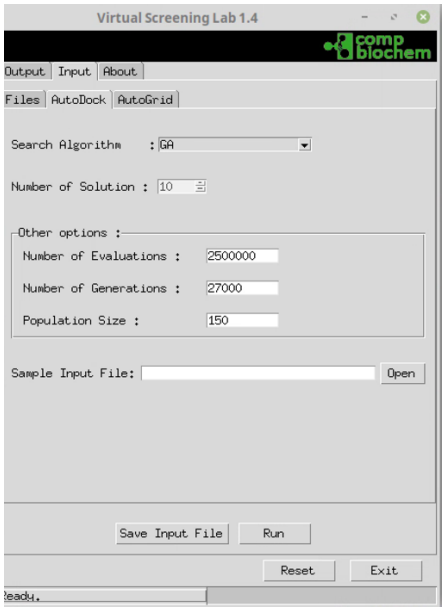
We now have to prepare all the input files that are required to run the AutoDock and AutoGrid programs using the VsLab plug-in. For this, we have to open VMD and load the file containing the protein. This can be done using the VMD menu (File » New Molecule).

We must now select the VsLab plug-in in the VMD main window. Click on:

Extensions » **PortoBioComp** » **VsLab**

The VsLab plug-in is composed of two main sections: one that is used to analyze the results of the Docking Process (tab with the name **Output**) and another that is used to create the input file (tab with the name **Input**). In order to create the input files that are required to

run AutoDock and AutoGrid we will focus on the tab with the name Input.



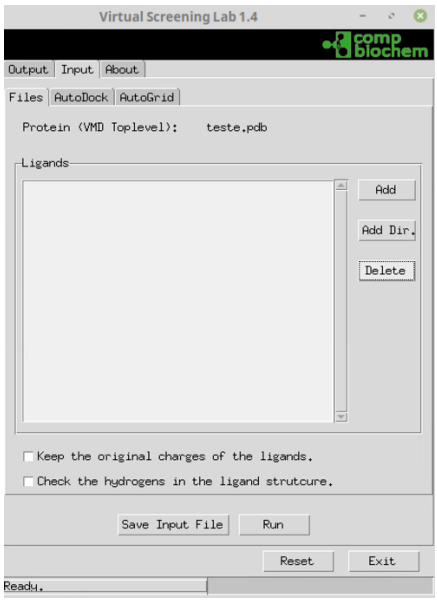
Main window of the VsLab plug-in.

The Input tab has three subsections that are divided in three different tabs, called **Files**, **AutoDock** and **AutoGrid**.

A – Ligand tab

The Ligand tab allows the user to add one or more ligands to the process. In our case we are going to use the file ligand.mol2 that you have prepared. The user can easily add it clicking on the button **Add**.

Note: The target protein is automatically selected by VsLab as the top level structure of the VMD program.

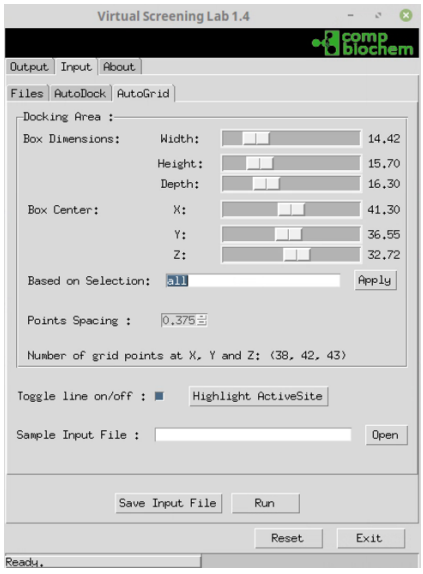


The Ligand section of the Input tab of VsLab.

B – AutoGrid tab

The **AutoGrid** tab allows the user to select the region of the protein where the ligand is going to be fitted in. The selected region is marked by a yellow-edged box in the VMD display window. The size and centre of the box can be modified in this tab. The AutoGrid algorithm generates a grid in the selected area in order to perform the docking procedure. This grid will later on be used to generate a set of grid points that determine the binding of the ligand. More grid points will give more reliable results, but the computational time can also increase dramatically. The number of grid points in each face of the box is displayed in the interface (number of grid points at x,y and z), and it is calculated using the following formula (for example in the x axis):

*Number of grid points (x axis) = width of the square * grid spacing*

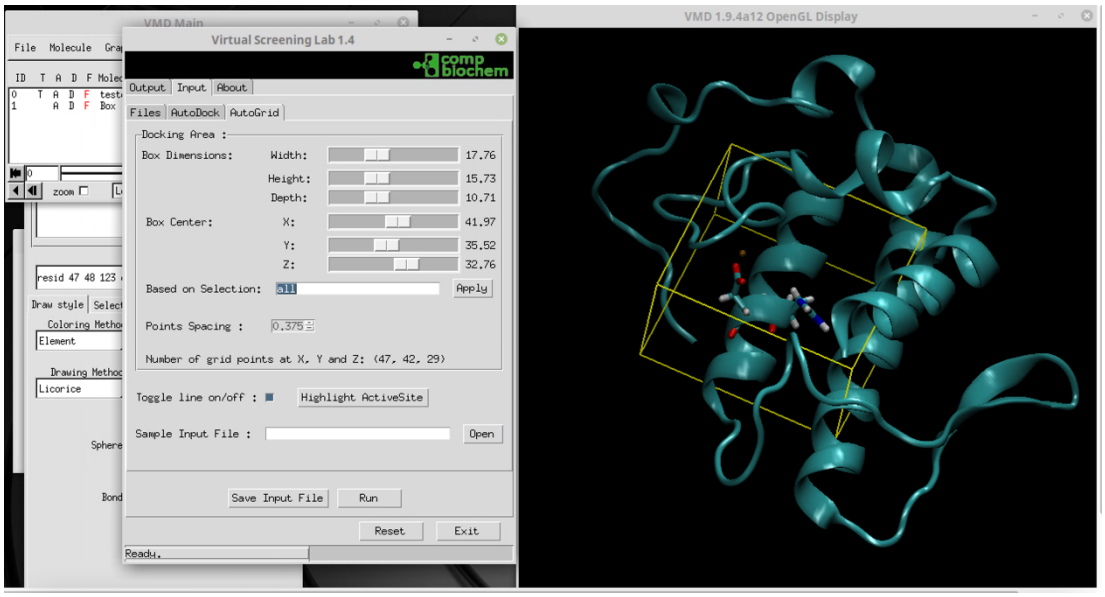


The AutoGrid section of the Input tabs.

The width of the square can be handled by the user using the box dimensions bars (in this case the width bar). The user can also modify the grid spacing (the standard value is 0.375 Å). The user should take into account that the maximum number of grid points in each side of the box is 126, and therefore a compromise between the dimension of the box and the grid spacing must be done in order not to overcome this value.

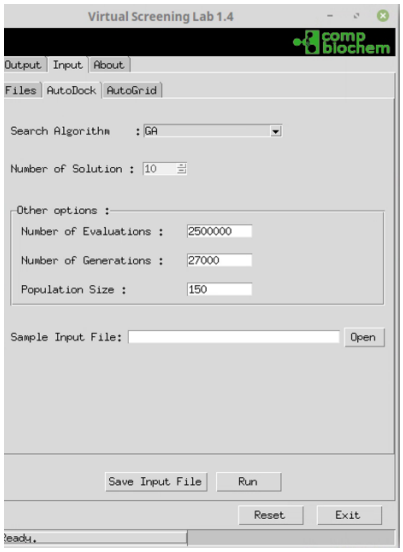
How do we select the binding site?

The best way to see if the box contains all the residues with which the ligand should interact with, is to highlight some residues of the active site using the VMD Representations window (click in Graphics » Representations in the VMD Main window). To select the residues that make the binding site type: **resid 47 48 123 or resname CA**



The main residues from the active site that interact with the substrate, ***i.e.*** His47, Asp48, Ca^{2+}. We chose not to include the catalytic water for fear of interference with the docking procedure.

C– Autodock tab



The AutoDock tab.

The AutoDock tab allows for the modification of some variables that control the AutoDock executable. Even though you have learned that a docking protocol is composed by a search algorithm that generates trial poses and a scoring function that ranks them, the Autodock tab only allow you to tune the search algorithm. Autodock has a single scoring function that does not allow for further specifications or modifications.

The options present in the AutoDock tab are:

Search Algorithm: The algorithm used to generate trial binding poses of the ligand onto the target protein. In this case you will use a genetic algorithm (GA). Please refer to the publication recommended 'Bioinformática Molecular – conceitos fundamentais' to read more about genetic algorithms,

Number of Solutions: Number of poses that will be included in the output by the execution of AutoDock. These are the best scored poses among the many poses generated by the genetic algorithm. Note that here a "solution" is a synonym for "pose", as the poses are the solutions of the searching algorithm procedure.

Other Options: These options control the execution of the genetic algorithm. The options that can be modified are: Number of Evaluations, Number of Generations and Population Size (the modification of the default values should only be done by advanced users).

However, the user can leave this part untouched, since it has got already the default values of the AutoDock software. Generally, only the Number of Solutions option might be modified.

4. **Saving the Input file and running the calculations.**

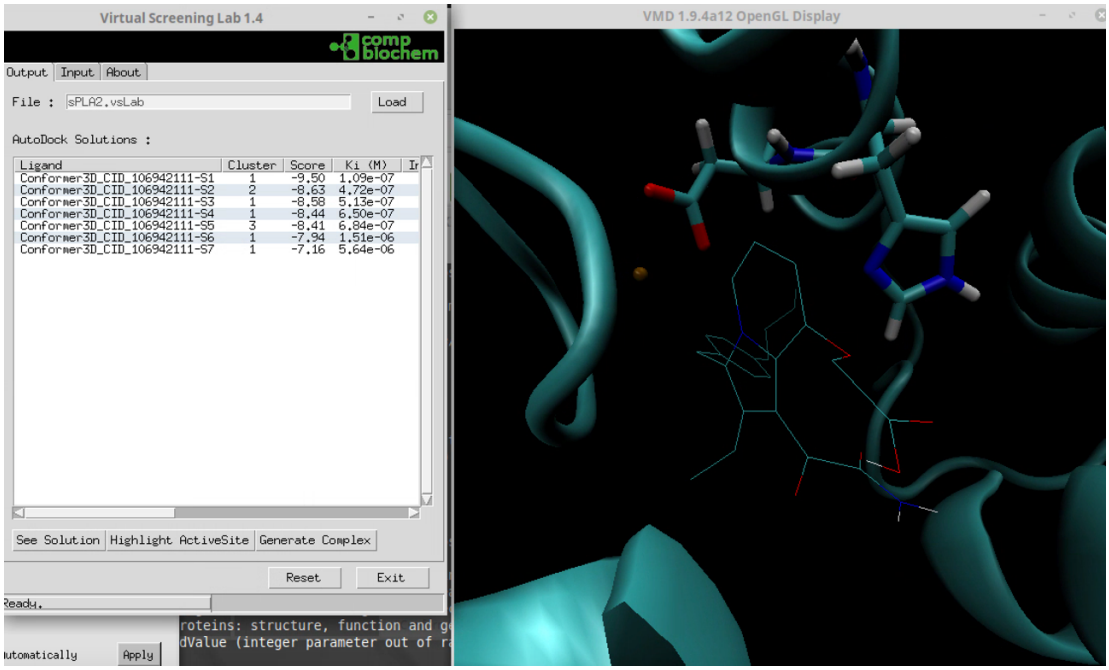
Once all the variables have been introduced, the input file can be created by clicking the button **Save Input File** - you may call it **sPLA2**. Subsequently, the user just has to click **Run** to start running the job. Depending on the values attributed to the variables that were introduced and the number of ligands that are analyzed, the computational time can become very long. During this period the VMD interface will remain blocked. The VMD interface should not be turned off during this period.

5. **Analyzing the Results.**

Once the job is finished, the users can upload the resulting file (**sPLA2.vslab**) in the **Output Tab**. If the job ran with no errors the table will display the solutions.

In our case we have to:

1. Load sPLA2.vsLab
2. Click on each solution (pose)
3. Analyze visually each solution (pose)



First solution from docking of inhibitor onto sPLA2.

Each line contains a solution (pose) for each generated complex, its score and its inhibition constant (K_I). The score corresponds to the calculated binding free energy (ΔG_{bind}) between the receptor and the ligand in aqueous phase, which is the fundamental quantity that measures the strength of the binding of the ligand to the protein. The inhibition constant is related to the binding free energy through the equation $K_I=\exp(\Delta G_{bind})/RT$. This information can then be used to find out the complex for which the interaction between the protein and the ligand is the best, ***i.e.*** which complex got the best score. Thus, make sure you find out the binding poses in which the complementarity between the ligand and receptor is best, in terms of protein-ligand interactions. See if this best complementarity correlates with a better score. Scoring functions are not infallible and we should not blindly accept their results. Instead, we should use chemical knowledge to check and, eventually, choose the binding pose it considers as the best.

6. Checking the result against a corresponding x-ray structure.

In order to evaluate the robustness of the procedure, the higher-scored solution can be compared to an x-ray structure of sPLA2 bound to the inhibitor you have just docked. See whether you can find one such file in the Protein Data Base.

Note: Remember that two proteins can be aligned using the RMSD calculator plug-in available at: Extensions » Analysis » RMSD Calculator in the VMD Main window.

FURTHER READING

Sousa, S.F., P.A. Fernandes, and M.J. Ramos, Protein-ligand docking: current status and future challenges. Proteins, 2006. 65(1): p. 15-26.

Sousa, S.F., et al., Protein-ligand docking in the new millennium--a retrospective of 10

years in the field. Curr Med Chem, 2013. 20(18): p. 2296-314.

3. VIRTUAL SCREENING TUTORIAL

1. Background

Virtual screening (VS) campaigns have emerged with the objective of increasing the probability of finding novel hit and lead compounds, decreasing in this way the cost associated to bringing a new drug to the market. Virtual screening is more than just a multiple molecular docking exercise and is dependent upon the simplifications that must be adopted, to make a good balance between accuracy and speed. The main issues that continue to preclude the progress of the virtual screening concern i) the quality of the scoring functions, ii) the poor exploration of the conformational space for the ligand and mainly the receptor, iii) the correct representation of the protonation states of the ligands' ionizable groups and iv) the inclusion of the water molecules that mediate links between receptor and ligand.

For the purpose of virtual screening, a scoring function does not necessarily have to rank hits correctly (remember that a hit is a molecule that binds to the receptor with good affinity, having a K_I in the micromolar (μM) range), but it should be able to discriminate hits from non-binders. The exploration of the conformational space is also an important issue because it can potentiate the search of new interesting scaffolds, and in virtual screening low hit rates of interesting scaffolds are clearly preferable over high hit rates of already known scaffolds.

The VS method is the one that has been gathering more adepts and consensus, and aims at reaching a state in which novel methodologies can decrease the time required for the calculations to be performed, increase the accuracy and reproducibility of the results (particularly when 'cheaper' computational methods are used), improve the statistical methods used to analyze the resulting volume of data, etc.

From the enunciated variables, the limited scope of the properties that can be calculated by computational resources at the moment is perhaps one of the weakest spots in virtual screening methodologies. In fact, if one could obtain an appropriate treatment of all molecular interactions, ionization and tautomerization states of both ligand and receptor, be capable of dealing with target/ligand flexibility and multiple binding modes, and precisely calculate the solvation free energy, then a correct and precise binding free energy for each compound could be calculated, from which the compounds that bind better could then be identified (remember that the binding free energy is the property that truly measures how tightly each compound binds its receptor). However, at the current stage, most of these properties cannot be fully described and simplifications must be employed to enable the analysis of large virtual collections. This means that the computed properties may differ from their true values and therefore a small inaccuracy can alter the correct judgement. Consequently, the output of virtual screening includes a significant number of compounds that are identified as binders but that are not truly binders (false positives), and compounds that are binders (the hits) but identified as non-binders (false negatives). Nevertheless, most of the compounds identified as non-binders are compounds that in reality do not bind the target (true negatives) and most hits are identified as binders (true positives).

A good virtual screening method should therefore be capable of minimizing the number of both false positives and false negatives. The major objective is to build up a list of compounds to test experimentally that includes the largest number of true positives and the minimum number of false positives. Even though, it is not uncommon that the list of compounds chosen to be tested contains 80%-90% of false positive, and the experimentalist needs to test 5-10 compounds to get a single hit. However, as the rigorous experimental testing of hundreds of compounds is easily feasible nowadays, it is easy to understand the enormous power of virtual screening to effectively discover new hit molecules for the desired therapeutic targets.

In fact, virtual screening is still the best option available nowadays to explore a large chemical space in terms of cost effectiveness and commitment in time and material, as it allows access to a large number of possible ligands, most of them easily available for purchase and subsequent testing. Although the number of therapeutic targets that have been fully characterized by crystallography is currently limited, this situation is set to change significantly in the immediate future as structural genomics and proteomics initiatives begin to yield fruit. With the development of new docking methodologies capable of predicting better hit rates and better predictions of geometries, virtual screening methodologies have already gathered a preponderant role in drug discovery.

2. Protein and Ligand for the VsLab input files

2.1. The protein

Our protein, **sPLA2.pdb**, has been already prepared during the Molecular Docking tutorial.

2.2. The ligands

To get our compound library, we will work with the database **BindingDB**.

BindingDB is a public, web-accessible database of measured binding affinities (K_I, $\Delta G_{bind,}$ or the less relevant IC_{50}), focusing mainly on the interactions of proteins considered to be candidate drug-targets with ligands that are small, drug-like molecules. This database can be found at:

https://www.bindingdb.org/bind/index.jsp

The data available at the BindingDB derives from a variety of measurement techniques, including enzyme inhibition and kinetics, isothermal titration calorimetry, NMR, and radioligand and competition assays. BindingDB also includes data extracted from the

literature by the BindingDB project, selected PubChem confirmatory BioAssays, and ChEMBL entries for which a well-defined protein target ("TARGET_TYPE='PROTEIN'") is provided.

The BindingDB's web-interface provides a range of browsing, query and data download tools. These include browsing by the name of a protein Target or by journal citation, query by chemical similarity and substructure, and downloads by target or query result.

Currently, the BindingDB contains over 2 million binding data for ***circa*** 8,200 protein targets and nearly 1 million drug-like molecules.

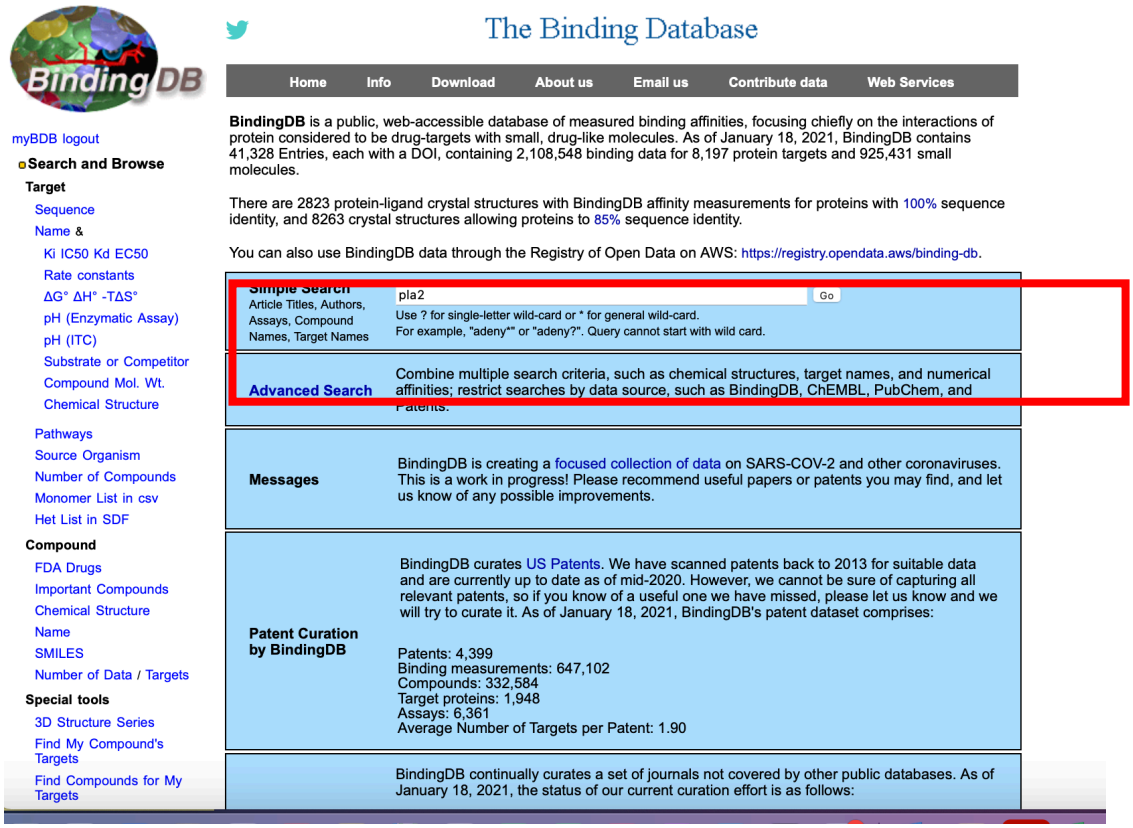
2.2.1. Downloading the compound library.

One of the most important tasks in virtual screening campaigns is to choose the library of compounds. Common virtual screening libraries are made of combinatorial libraries and libraries of available molecules from in-house compound repositories or vendor offerings. In these studies, the performance of a virtual screening technique is prognostically measured by its ability to retrieve a small set of previously known hits from a library containing a much higher proportion of assumed non-binders, that might be very different from the hits or hit decoys. This allows to test the virtual screening process and evaluate its performance in the correct identification of hits compounds (true positives) and non-binders (true negatives), as well as the incorrect identification of false positives and false negatives. If the results are satisfactory, the methodology can then be used in a more prospective virtual screening way. In this process more compounds are evaluated or the same set of compounds is re-analyzed and the best scored ligands are subjected to experimental confirmation (***e.g.*** simple IC_{50} measurements first, and more laborious K_I or ΔG_{bind} determination afterwards, for the best candidates). The process can then be repeated with an increasing number of compounds.

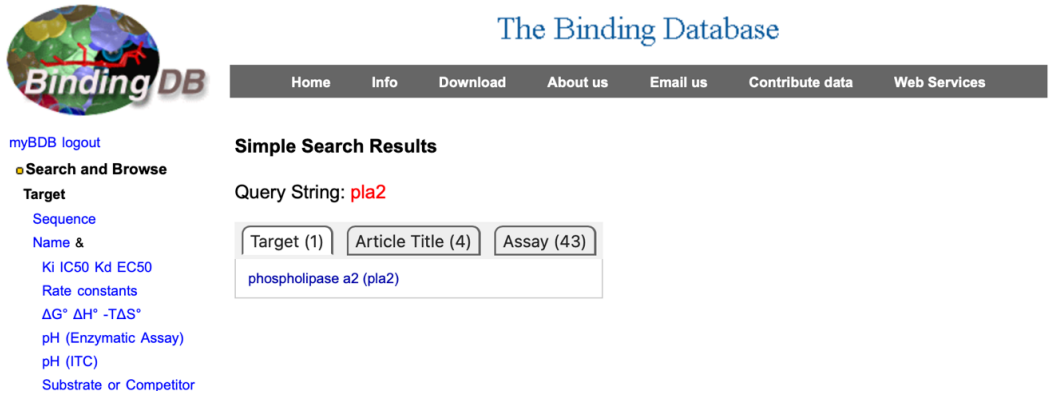
In this tutorial we are going to screen a small set of compounds that have affinity or are related with the target of interest. The compounds will be retrieved from the BindingDB. This will allow us to validate the virtual screening process and simultaneously collect important information regarding the type of characteristics (scaffold, chemical groups, etc.) that the compounds addressed to the basic secreted phospholipase A2 (sPLA2) enzyme must have in order to present some significant biological activity.

The list of compounds related with the basic secreted phospholipase A2 (sPLA2) enzyme can be retrieved from the BindingDB following the next steps:

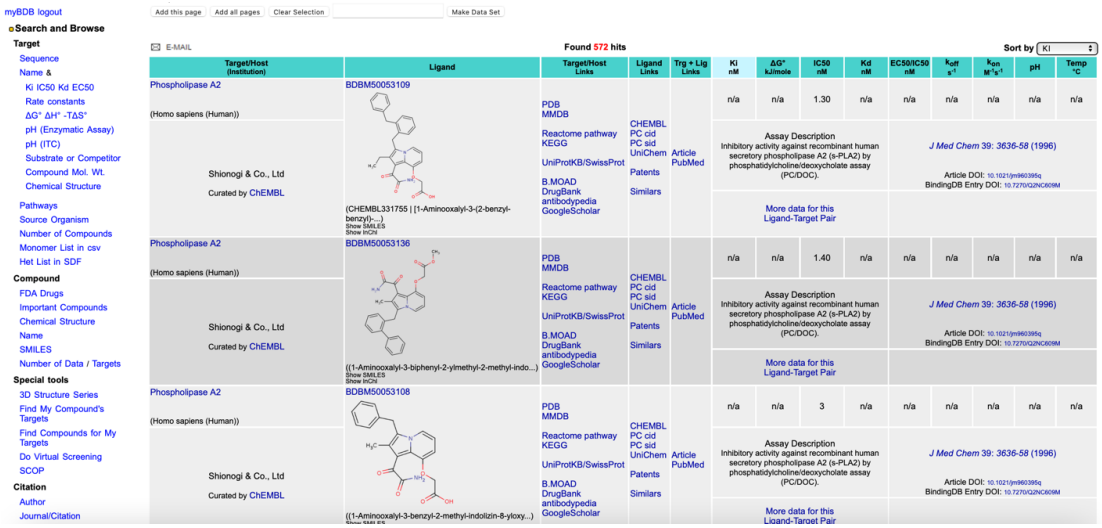
3. In the **https://www.bindingdb.org/bind/index.jsp** search for the PLA2 target.



4. In the next web page select the target of interest.



3. The next page contains all the compounds (572) that are related with the chosen target. The chemical structure, the biological data and other information can be selected by the user and analyzed independently.



4. To compile the dataset and download the information the users have to select "Add All Pages" at the top of the web page and select "Make Data Set".

Note that the molecules that you are going to download are potential inhibitors for PLA2 of a number of organisms, including Homo sapiens.

5. In order to retrieve the 3D structure of all the compounds, press GO. This will generate an file in sdf format that you will be able to download. First you have to register. However, make sure that beforehand you order them using their IC_{50} (experimental conditions should be observed) as this will tell us whether the compound is a hit (IC_{50} ~<1 µM) or a non-binder. Please save the file in a directory in your Desktop, named **Virtual Screening**.

6. The file in the SDF format can be viewed as a database of structures. This file contains the 3D structure of the molecules and additional information regarding these structures. An example of such file can be viewed at:

https://www.bindingdb.org/bind/chemsearch/marvin/BindingDB-SDfile-Specification.pdf.

2. Obtaining the 3D structure of the compounds from the downloaded database.

The 3D structure of all the individual compounds present in the sdf file, can be obtained using the open source software called openbabel, just as you have done beforehand in the Molecular Docking tutorial.

It will be too much time consuming for us to run a Virtual Screening for the 572 molecules that we obtained from **BindingDB** in the classroom, because the computational time taken would be too long for our class. Therefore, we will run only the last 5 molecules that were obtained in **BindingDB** (remember that all the 572 molecules were ordered by their IC_{50} values). This means that we will run the 5 ligands that are taken as being probably non-binders. During the docking procedure, we did in fact dock the best inhibitor and we will therefore be able to compare it with those from the VS campaign that we will run here.

3. Running the virtual screening campaign.

Create a directory **VS** under the Virtual Screening directory and place there the 5 worst ligands. Once all the data has been prepared the virtual screening campaign can be set up, the input file can be saved as **sPLA2_VS** and the campaign can be run.

4. Analyzing the results.

Visualize all the structures and identify the major interactions that there are between the ligands and the active centre of the protein. Can you justify the ranking established by the VS?

Compare the docked ligands and their binding to the active site of the enzyme with the result we got from the previous molecular docking tutorial.

FURTHER READING

Sousa, S.F., et al., Virtual screening in drug design and development. Comb Chem High Throughput Screen, 2010. 13(5): p. 442-53.

Lionta, E., et al., Structure-based virtual screening for drug discovery: principles, applications and recent advances. Curr Top Med Chem, 2014. 14(16): p. 1923-38.

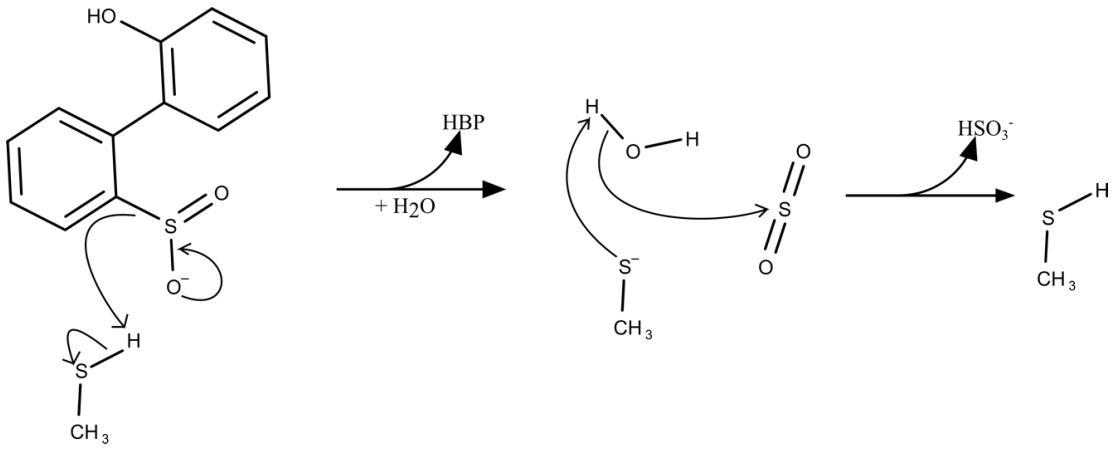
4. QM/MM TUTORIAL

1. Background

In this tutorial the students will learn the basic procedures to establish the reaction mechanism of an enzyme. Enzymes constitute ideal systems to apply QM/MM (Quantum Mechanics/Molecular Mechanics) methods because they have a small region (the active site and substrate/cofactors) where electronic rearrangements (in this case chemical reactions) take place, and a very large enzymatic molecular scaffold, with thousands of atoms, plus the solvent, all making electrostatic interactions with the active site but having no significant electronic rearrangements. In such scenario, the active site, substrate and cofactors may be described with quantum mechanics and the remaining system with classical mechanics.

DszB is a very interesting enzyme, with 365 residues, that participates in the sulfur metabolism of several bacteria and that has great biotechnological potential due to its ability to remove sulfur from organic compounds at room temperature and pressure. In order to use this enzyme in industrial applications its catalytic mechanism has to be known, so that mutations can be rationally predicted and introduced to tune its activity and reactivity to the desired levels.

The catalytic mechanism is proposed to follow two possible pathways: either a nucleophilic addition or an electrophilic substitution pathway. Here we will investigate the feasibility of the latter. For that we will calculate the activation free energy of the rate-determining step and see if it matches the experimental value. If this happens the simulated mechanism is probably the correct one. According to this mechanism, the reaction starts with the protonation of a carbon atom of the substrate, with elimination of SO_2, and in a second step SO_2 is attacked by a water molecule giving origin to HSO_3^- (experimentally observed in solution) and restoring the original active site protonation state (**Scheme 1**).



Scheme 1. Electrophilic aromatic substitution mechanism proposal for the DszB reaction mechanism.

The study consists in i) the division of the enzyme into two layers - a Quantum Mechanics layer and a Molecular Mechanics layer, ii) a geometry optimization of the initial reactant state, previously modeled from the pdb structure 2DE3 and iii) the determination of an approximate transition state structure for the first reaction step of the mechanism. All the files necessary to carry out this tutorial can be found in the folder **Files**.

2. Defining the QM and MM layers.

Open a terminal window and launch the molecular visualization software VMD.

Load the initial reactant structure:

➢ vmd **Reactant.pdb**

After loading the reactant structure, identify the substrate and the residues that are most important for the first, rate-determining step of this reaction. The name of the substrate in the PDB file is OBP and the important residues are Ser6, Cys8, His41, Arg51 and Gly54.

Question: why these residues are expected to be the most important for the reaction?

To prepare our system for a QM/MM calculation, we start by defining which atoms will be described by quantum mechanics (the QM layer) and by classical mechanics (the MM layer). For that we run the software VMD molUp and load the **QMMM_prep.com** file. (Note: *.com files are the input files for the software Gaussian16. These files contain information on the molecule structure and instructions that define the calculation to be done). In this file we have worked the x-ray structure, in order to add the missing hydrogen atoms and add a layer of solvent.

➢ vmd > Extensions > MolUP

On the tab *"Model"*, select the atoms for the QM layer. To do so, you can go to ***Model > Layer*** tab, use the molUP selection tools ***(Atom selection (Change ONIOM layer)"***) and pick the atoms by typing the selection "resid 6 8 51 54 or resname OBP" for the QM layer (initially, all atoms are defined as Low Layer/ MM atoms).

Challenge: Based on what you have learned in the lectures, try to choose the appropriate QM layer.

Then go to the *"Freeze"* tab and select all the protein and substrate atoms to "unfreeze" them (frozen atoms are defined by number *"-1"* and unfrozen atoms are defined

by number *"0"*), to do so type "protein or resname OBP" on the atom selection box, select number 0 and *"Apply"*.

Check visually the QM and MM layers on the molecular representations (check *"High layer"* box at the bottom of the molUP interface).

On the tab "*Input"* introduce the theoretical level to be employed in the QM layer. B3LYP is a good choice for geometry optimizations and the basis set 6-31G(d) offers a good compromise between the accuracy of the QM layer geometry and the time needed for the calculation. The MM layer will be treated with the AMBER force field. To insert these specifications in the input file, go to the *"Calculations"* box and type:

oniom(B3LYP/6-31G(d):AMBER=softfirst) geom=connectivity opt

To save the input file go to *"File"* and select *"Save",* save the file as **QMMM.com**.

3. Optimizing the geometry of the reactant.

Launch the QM/MM calculation on the terminal window:

➢ g16 **QMMM.com** &

This calculation can take many hours, or days. To avoid spending too much time waiting, you can stop the calculation and copy the file **initial_opt.log** from the folder **outputs** to the folder you are working in. This file contains the result of the calculation you have just stopped.

On the molUP interface, load the file **initial_opt.log**. Load all structures to see each step of the geometry optimization. Check the box *"High Layer"* and uncheck the box *"All"* on the molUP interface. Press "*Play*  " on the VMD Main Window and visually inspect how the structure relaxes throughout the optimization.

Question: The most important interactions for the chemical reaction are preserved through the geometry optimization?

Save the last structure of the optimization calculation (i.e. the optimized structure), as an input file for Gaussian16, with the name of **PEP.com**, so we can proceed with the study of the first step of the reaction mechanism.

4. Calculating a potential energy profile and identifying an approximate transition state structure.

To study a reaction step, you need to generate a good guess of the transition state structure. The usual method is to calculate a Potential Energy Surface (PES). A PES is a function that has as arguments some, or all, of the atomic coordinates of the system, and as a

value the energy of the system. The specific coordinates of each PES are chosen depending on the studied phenomenon. For a chemical reaction, the PES usually involves the distances between reacting atoms. If a PES has as argument a single coordinate, then it is usually called a Potential Energy Profile (PEP).

In the case we are studying, it is reasonable to assume that the distance between the SO_2 bound C atom and the electrophilic proton from the Cys27 thiol group may be a reasonable representation of the reaction coordinate. As such, we will calculate a PEP along this distance and see if the PEP increases up to a maximum in energy and comes down again in value, defining a transition state structure and a product along the PEP. The energy maximum should be close to the true transition state structure along the true reaction coordinate. For that we will constrain the SH—C distance to a set of specific points and relax the remaining system, to calculate the energy along the SH—C distance. Such calculation is usually named a ***linear scan***.

On the molUP window, go to the "***Input"*** section. Change the calculation keywords to perform the linear scan. In this case, it is important to introduce the "ModRed" option associated with the "opt" keyword in order to instruct Gaussian16 to perform a linear scan:

Click on "***Other information (Parameters, ModRedundant…)"*** menu and select "***ModRedundant Editor"***.

On the pop-up window, click on "***Pick"*** to select the pair of atoms whose distance you want to scan.

Subsequently, you need to define the points along the SH—C distance that will be calculated. You should start measuring the initial and "expected" final SH—C distances, to see the distance range you need to span. Afterwards you need to assume a reasonable step size (the distance between points), not too fine so that the calculation does not takes too long, and not too large, so that the PEP has good definition. Generally, for a first linear scan exploration, points separated by 0.1 Å is a good choice.

Introduce the number of points that will be used in the PES. Provide the size of the step in Å. Positive numbers will increase the distance between both atoms whereas negative numbers will short their distance. Click in "***Apply"***. On the "***Other information (Parameters, ModRedundant…)"*** field, insert a blank line before the ModRedundant information (i.e., the line B 116 5157 S 15 -0.1) and two blank lines after the ModRedundant information. Save as a new Gaussian16 input file, with the name **PEP.com**. Launch QM/MM calculation on the terminal window:

➢ g16 **PEP.com &**

Again, this calculation will take ages. Stop the calculation and copy the file **PEP.log** from the **Files** folder. This would be the result you would get if you wait for a few days.

Open the output file (**PEP.log)** on molUP, select "All optimized structures" option.

Identify the approximate transition state structure and activation energy through the plot available in the "***Output"*** tab of molUP. Measure the key interatomic distances of the reaction all along the potential energy profile. Pay particular attention to their values at the reactant and transition state.

Identify the approximate product structure and reaction energy through the plot available in the "***Output"*** tab of molUP.

Compare the calculated activation energy with the experimental activation free energy (19.0 kcal/mol). As these reactions are dominated by enthalpy, the energy barrier that you calculate should not be far from the experimental free energy barrier.

In a real research case, more steps would be needed. Namely:

i) To use the transition state structure you have identified as a starting guess to perform a full, unconstrained transition state optimization; ii) to determine the connected reactants and products through internal reaction coordinate calculations; iii) to calculate the zero point energies of all species; iv) to calculate rigid rotor/harmonic oscillator entropies to obtain free energies for all species; v) to recalculate the final energies with higher theoretical levels, or at least, with larger basis sets.

5. FINAL REMARKS.

This tutorial has given you a basic idea on how to run QM/MM calculations. As any other area of Science, you would need years to fully dominate the techniques. Our purpose is to give you a bit of the flavor of this field, to help you to understand better how to establish the mechanism of an enzymatic reaction. We hope you have enjoyed it as well as having enjoyed the previous tutorials.

FURTHER READING

Sousa, S. F.; Ribeiro, A. J. M.; Neves, R. P. P.; Bras, N. F.; Cerqueira, N. M. F. S. A.; Fernandes, P. A.; Ramos, M. J., Application of quantum mechanics/molecular mechanics methods in the study of enzymatic reaction mechanisms. Wires Comput Mol Sci 2017, 7 (2).