ACS Synthetic Biology

pubs.acs.org/synthbio

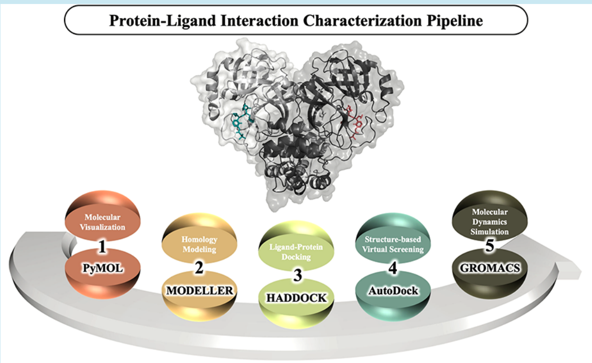
In Silico End-to-End Protein−Ligand Interaction Characterization Pipeline: The Case of SARS-CoV-2

Nícia Rosário-Ferreira,[#] Salete J. Baptista,[#] Carlos A. V. Barreto, Filipe E. P. Rodrigues, Tomás F. D. Silva, Sara G. F. Ferreira, João N. M. Vitorino, Rita Melo, Bruno L. Victor, Miguel Machuqueiro,* and Irina S. Moreira*

Cite This: https://doi.org/10.1021/acssynbio.1c00368 Read Online

ACCESS | Metrics & More | Article Recommendations

ABSTRACT: SARS-CoV-2 triggered a worldwide pandemic disease, COVID-19, for which an effective treatment has not yet been settled. Among the most promising targets to fight this disease is SARS-CoV-2 main protease (M^{pro}), which has been extensively studied in the last few months. There is an urgency for developing effective computational protocols that can help us tackle these key viral proteins. Hence, we have put together a robust and thorough pipeline of *in silico* protein−ligand characterization methods to address one of the biggest biological problems currently plaguing our world. These methodologies were used to characterize the interaction of SARS-CoV-2 M^{pro} with an α-ketoamide inhibitor and include details on how to upload, visualize, and manage the three-dimensional structure of the complex and acquire high-quality figures for scientific publications using PyMOL (Protocol 1); perform homology modeling with MODELLER (Protocol 2); perform protein−ligand docking calculations using HADDOCK (Protocol 3); run a virtual screening protocol of a small compound database of SARS-CoV-2 candidate inhibitors with AutoDock 4 and AutoDock Vina (Protocol 4); and, finally, sample the conformational space at the atomic level between SARS-CoV-2 M^{pro} and the α-ketoamide inhibitor with Molecular Dynamics simulations using GROMACS (Protocol 5). Guidelines for careful data analysis and interpretation are also provided for each Protocol.


Protein-Ligand Interaction Characterization Pipeline
Molecular Visualization
1
PyMOL
Homology Modeling
2
MODELLER
Ligand-Protein Docking
3
HADDOCK
Structure-based Virtual Screening
4
AutoDock
Molecular Dynamics Simulation
5
GROMACS

KEYWORDS: *SARS-CoV-2 main protease, molecular docking, virtual screening, molecular dynamics simulations*

■ INTRODUCTION

Recently, at least three highly pathogenic human coronaviruses, the severe acute respiratory syndrome coronavirus 1 (SARS-CoV-1), the Middle East respiratory syndrome coronavirus (MERS-CoV), and SARS-CoV-2 have emerged worldwide as violent endemic diseases. The last one, named coronavirus disease 2019 (COVID-19) which became a pandemic in 2020, has a human-to-human transmission. COVID-19 is causing a vast clinical, social, and economic burden surpassing the huge barrier of 4.65 M deaths and over 225 M confirmed cases (as of September 15th, 2021; https://coronavirus.jhu.edu/map.html). However, despite recent efforts, there are currently no effective antiviral drugs or vaccines in the market targeting SARS-CoV-2.[1] There is an imperative need for developing effective *in silico* protocols to help us understand and characterize the main interactions established by key viral proteins. One of the best-characterized drug targets among coronaviruses is its main protease (M^{pro}), for which there are several available crystal structures and virtual screening studies ongoing.[2−4,1] The PDB ID code 6Y2G (www.rcsb.org/structure/6Y2G) contains a crystal structure of the complex resulting from the reaction between SARS-CoV-2 (2019-nCoV) M^{pro} and *tert*-butyl (1-((*S*)-1-(((*S*)-4-(benzylamino)-3,4-dioxo-1-((*S*)-2-oxopyrrolidin-3-yl)butan-2-yl)amino)-3-cyclopropyl-1-oxopropan-2-yl)-2-oxo-1,2-dihydropyridin-3-yl)carbamate (α-ketoamide 13b for short), a SARS-CoV-2 M^{pro} reversible covalent-binding inhibitor.[3]

In this tutorial, our aim is to establish and explain a comprehensive pipeline comprising five protocols (Figure 1) that will provide an extensive characterization of protein−ligand complexes using SARS-CoV-2M^{pro} as an example. Although both covalent and noncovalent SARS-CoV-2 M^{pro} inhibitors

Received: August 2, 2021

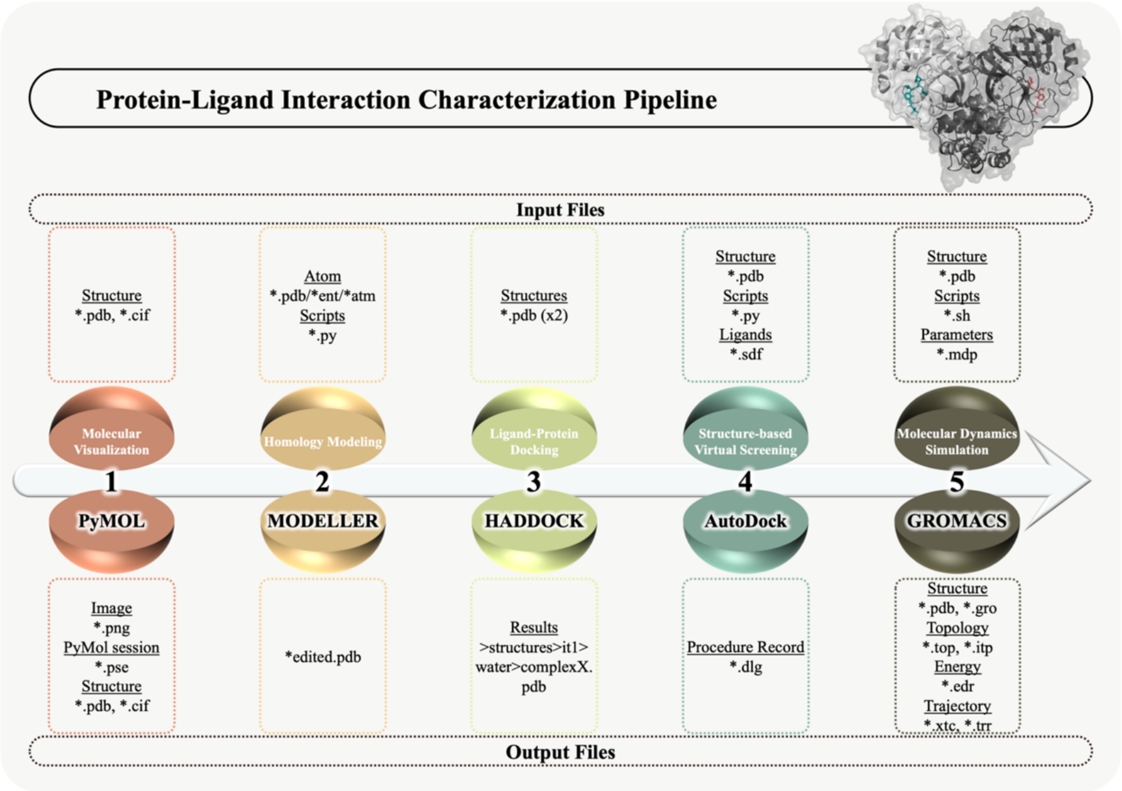


Figure 1. General overview of the pipeline and the color code followed throughout the protocol identifying the Protocols 1 to 5. Mandatory input and output files are also presented. These files along with additional files will be available on the online repository as mentioned in the text.

have already been identified,[5] this tutorial is focused mainly on the binding process. This covers both the noncovalent and the initial steps of the covalent inhibitors, such as α-ketoamide 13b. Additional information about the softwares used in the various Tutorials was compiled in the Availability of Data and Requirements section.

1. PROTOCOL 1: PYMOL—A KEY MOLECULAR VISUALIZATION TOOL

Herein, we introduce molecular visualization with the software PyMOL.[6] PyMOL is open-source and one of the most used molecular visualization tools that allows the three-dimensional (3D) visualization of different biomolecules, as well as surfaces and electron densities. It can also be used to edit molecules and create high-resolution scientific figures for publication. Additional analysis involving protein−ligand modeling, molecular dynamics (MD), and virtual screening can be performed as well using PyMOL. Most of the protocol is dedicated to showcasing general PyMol features (through points 1.1−1.3), specific applications such as the ability to explore the binding site (point 1.4), and production of high-quality scientific figures for publication (point 1.5). See Box 1 for necessary resources for this protocol.

Box 1. Protocol 1 Necessary Resources

Software: PyMOL v2.4+ (https://pymol.org/2/).

Files: No extra files are required. All structure files will be fetched from within the software.

Sample File: http://insilicotutorials.rd.ciencias.ulisboa.pt/pymol.html

1.1. The Graphical User Interface (GUI)—Overview.

The PyMOL[6] GUI consists of multiple panels as shown in Figure 2. The main display panel provides the visualization of the loaded molecules, whereas the object menu panel on the right-hand side settles a menu of options for the loaded molecules, providing various operations by mouse clicking. The same tasks can be achieved using the command line located both above and below the display panel. The upper control window contains a console as well as a set of handy buttons for quick access to various functions. Also, note the mouse and movie controls panel in the lower right corner.

Most tasks in PyMOL are achieved using the object menu on the right-hand side of the display panel. Here, each line (or entry) in the menu corresponds to the objects you loaded into PyMOL (e.g., molecules) as 6Y2G represented in Figure 2. Other menu entries can appear in parentheses in the same menu panel representing atom selections and not objects (see Figure 3 below for example).

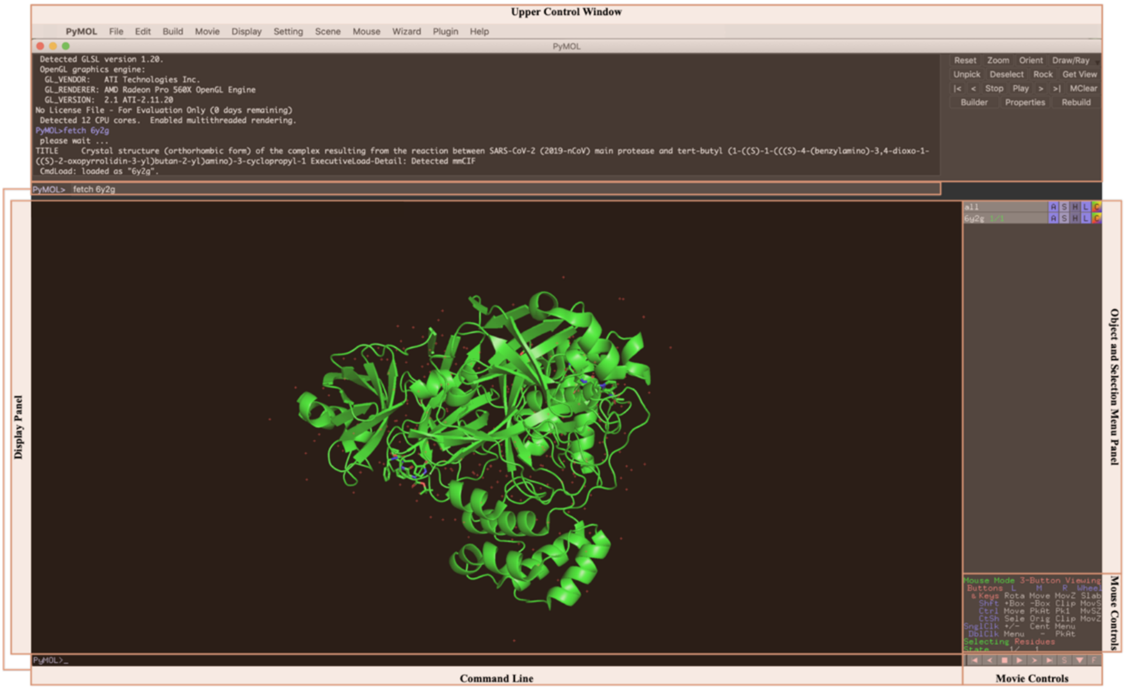
The menu items are located next to the objects and selections (Figure 3). These are labeled **A** (action), **S** (show), **H** (hide), **L** (label), and **C** (color). Upon clicking, the respective menu is opened.

1.2. Molecular Representations.

PyMOL provides multiple molecular representations for visualization. For a quick overview of these use the main menu and click

Wizard → Demo → Representation

This displays the eight different molecular representations available in PyMOL: lines, sticks, spheres, surface, mesh, dots, ribbon, and cartoon (Figure 4). Among them, lines, sticks, and cartoon representations are probably the most used to represent


Upper Control Window
PyMOL File Edit Build Movie Display Setting Scene Mouse Wizard Plugin Help
PyMOL
Reset Zoom Orient Draw/Ray
Unpick Deselect Rock Get View
|< < Stop Play > >| MClear
Builder Properties Rebuild
all A S H L C
6y2g 1/1 A S H L C
Display Panel
Object and Selection Menu Panel
Mouse Controls
Command Line
Movie Controls

```
 Detected GLSL version 1.20.
 OpenGL graphics engine:
  GL_VENDOR:   ATI Technologies Inc.
  GL_RENDERER: AMD Radeon Pro 560X OpenGL Engine
  GL_VERSION:  2.1 ATI-2.11.20
No License File - For Evaluation Only (0 days remaining)
 Detected 12 CPU cores.  Enabled multithreaded rendering.
PyMOL>fetch 6y2g
 please wait ...
TITLE     Crystal structure (orthorhombic form) of the complex resulting from the reaction between SARS-CoV-2 (2019-nCoV) main protease and tert-butyl (1-((S)-1-(((S)-4-(benzylamino)-3,4-dioxo-1-
((S)-2-oxopyrrolidin-3-yl)butan-2-yl)amino)-3-cyclopropyl-1 ExecutiveLoad-Detail: Detected mmCIF
 CmdLoad: loaded as "6y2g".
```

```
PyMOL> fetch 6y2g
```

```
Mouse Mode 3-Button Viewing
 Buttons  L    M    R  Wheel
  & Keys Rota Move MovZ Slab
    Shft +Box -Box Clip MovS
    Ctrl Move PkAt Pk1  MvSZ
    CtSh Sele Orig Clip MovZ
 SnglClk +/-  Cent Menu
  DblClk Menu  -   PkAt
Selecting Residues
State     1/   1
```

```
PyMOL>_
```

Figure 2. PyMOL graphical user interface with PDB ID 6Y2G loaded.

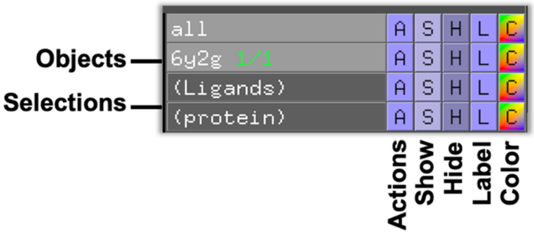


Figure 3. Object and selections names and menu items in GUI of PyMOL.

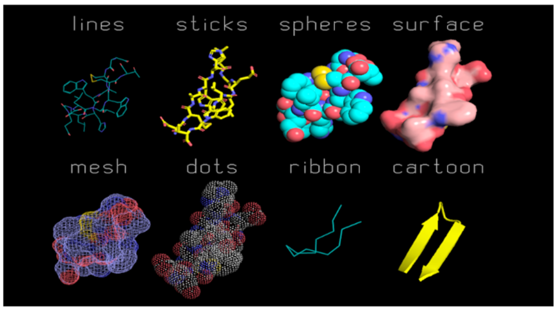


Figure 4. Available molecular representations in PyMOL.

bimolecular structures in an independent or combined approach. The End Demonstration button exits the overview.

1.3. Basic Commands. *1.3.1. Load the Structure.* This command lets you explore the structure of SARS-CoV-2 M^{pro} and investigate the molecular interactions with the α-ketoamide 13b. There are three ways to upload the structure onto the software: (1) via the main menu system, (2) via command line, and (3) by uploading a previously downloaded structure. To load the SARS-CoV-2 M^{pro} PDB structure into PyMOL by the main menu[1] navigate to

File → Get PDB → [enter the 4-character PDB ID 6Y2G]

Alternatively, use the fetch command in the command line[2]

```
fetch 6Y2G
```

The structure upload[3] can be obtained by accessing the free available Protein Data Bank (PDB) https://www.rcsb.org/ and typing "SARS-CoV-2 main protease" to attain the PDB ID of interest (PDB ID: 6Y2G) and downloading the PDB file. Then, to load a PDB file from your own file system use the menu system and navigate to the designated download folder:

File → Open → [select 6Y2G.pdb]

Notes: When you type "SARS-CoV-2 main protease" a list of several SARS-CoV-2 main protease 3D structures were presented. The choice of a reliable PDB structure will be dependent on several parameters, such as the resolution of PDB structure, the binding of a ligand and its nature, and the experimental method used to obtain the 3D structure. The existence of mutant variants should be also an important parameter to take into consideration when you choose your PDB structure.

Usually, the best representative structure from a particular protein (in terms of model and data quality) is displayed at the top of your research results, ranked by Score.

PDB ID: 6Y2G was the structure chosen to proceed with this tutorial, as it represented one of the top results for SARS-CoV-2 main protease complexed with a ligand.

The PyMOL Viewer Window displays the 3D structure of SARS-CoV-2 M^{pro} in the default cartoon representation colored by atom type. The object menu panel now contains an entry named 6y2g representing the newly created object (see Figure 2 above).

1.3.2. Explore Molecule Orientations. The user can experiment with the different mouse controls available:

- Rotate: Press and hold the left mouse button while moving the mouse.

- Zoom: Press and hold the right mouse button while moving the mouse.
- Translate: Press and hold the middle mouse button while moving the mouse.

  Note: In case of using a touch mouse, press "ctrl" (Windows) or "command" (Mac) buttons.
- Clip the view: Scroll.

Click the Reset button in the upper control window any time to come back to the initial view, and the Zoom button to zoom on all loaded molecules. The Orient button orients the view to display the molecule along its longest axis aligned with the *x*-axis.

1.3.3. Use Visualization Presets.

PyMOL comes with multiple built-in visualization preset views, which can be accessed through the action (**A**) button(s) in the object panel and suit different goals, highlighting desired features and regions. There are several options in the present views, which mainly differ in the type of representation chosen for the protein and/or the ligand.

For example, click the **A** button next to the 6y2g object. From the menu that appears click **preset → ligands**. The program will now automatically zoom into the ligand regions of the protein and change the representation of the protein to ribbon, the residues close to the ligand will be shown in lines, and the ligand is shown as sticks. Also, notice the yellow dashed lines (polar contacts) appearing between the protein and ligand that are the new object standardly named "6y2g_pol_conts". As SARS-CoV-2 M^{pro} catalytic activity depends on the protein dimerization, two ligands are displayed, one for each monomer.

1.3.4. Change Molecular Representations.

Alternatively, to the various automatic representations mentioned above (see Figure 4), PyMOL enables several customizations for molecular representations. Click the **S** (show) button from the object menu next to the 6y2g entry and various molecular representations, such as lines, sticks and ribbon are now available to the user. These representations are cumulative, and the software builds them on top of each other sequentially.

S → cartoon

S → ribbon

From the first menu entry (**as**), you can select the representation (a single one) in which you want the protein to be shown as:

S → as → spheres

S → as → surface

S → as → cartoon

S → as → lines

The various representations can also be hidden using the **H** (hide) button from the object menu (see the example below).

H → ribbon

H → lines

1.3.5. Make the Command Line Your Ally.

All commands available in the menu system are also accessible through the PyMOL command line console. Despite appearing complicated at first, understanding and mastering the command line usage will be very useful. Locate the command line (see Figure 2). To show a particular representation, use the show command followed by the representation name:

```
show lines
```

To hide a representation use

```
hide lines
```

You can also show only one representation with the as command:

```
as cartoon
```

The zoom and orient commands are the equivalent to the Zoom and Orient buttons, respectively:

```
zoom
orient
```

Another important usage that can be achieved by using command line is the all-atom RMSD calculation (further applied in Tutorial 3). After loading your molecules in PyMOL, type

```
align object1, object2, cycles = 0,
transform = 0
```

in which *object1* and *object2* correspond to your molecules of interest.

We will come back to additional useful commands through the tutorial. Before continuing to the next section, use the command:

```
as lines
```

1.3.6. Basic Atom Selections Commands via the Main Menu.

Advanced visualization analyses of biomolecular structures often require the functionality to select individual or group of atoms. In PyMOL the simplest approach for atom selection is by simply clicking the atom in the visualization window.

Click any atom in the protein and note the new entry in the object menu on the right-hand side called "*sele*" by default. The selected atoms are shown in the viewer window with small pink squares highlighting the selected residue.

To zoom in on your selection do **A → zoom** on the (sele) entry or "*zoom sele*" in the command line.

Selections can have different molecular representations, for example, with **S → sticks** on the (sele) entry. The selected atoms should now appear as sticks in the view window. The same result is achieved by typing "*show sticks, sele*" in the command line.

Despite having clicked in a single atom, ALL ATOMS in the chosen residue are selected by default. Change it by clicking on the Selecting in the mouse controls panel (lower right corner). Click it once and notice that Selecting now changes from **Residues** to **Chains**. Click now once at the protein and notice that the entire protein is selected (depicted with pink boxes). Click back to Residues in the Selecting field.

Another way to select residues is through the sequence viewer that you can toggle through the **S** (for sequence) in the lower right corner of the screen (on the movie controls menu). Upon pressing **S**, you will see the sequence browser on the top of the viewer window. To select a residue from here, simply click (and drag) the relevant residue 1-letter codes.

To disable the selection, click on an empty void in the viewer window, or click the (sele) entry in the object menu.

Selections can be deleted by clicking **A → delete** selection and renamed with **A → rename** selection.

Select both residues named O6K. This will select the atoms of both α-ketoamide 13b ligands bound to the SARS-CoV-2 M^{pro} structure. Rename the selection to (ligands) by clicking:

A → rename selection

and type "ligands" in the dialogue.

Notes:

1. As noted above 6Y2G constitutes a homodimer. Therefore, two O6K ligand molecules are now displayed.
2. When you rename a selection, you will find **Renaming sele to: sele** in the dialogue box, assuming the name "sele" by default. Therefore, you should previously delete "sele" and then type "ligands" for renaming.
3. If you name the selections under different names, please keep that in mind in proceeding steps.

1.3.7. Basic Atom Selections via the Command Line. The command line (using "select") enables more versatile and specific atom selections. This command can be combined with various keywords specifying which atoms, residues, or chains to select. Here are some examples:

select resname selects the residue(s) by name; for example, select residue with name O6K:

```
select resname O6K
```

select chain selects atoms in the specified chain; for example, select chain with ID A:

```
select chain A
```

select name selects atoms with the specified name; for example, select all Cα atoms:

```
select name CA
```

The selection commands above generate a selection with the default name (sele). You can specify another name by adding it to the command. Create a new selection called (ligands):

```
select ligands, resname O6K
```

Create a new selection called (protein):

```
select protein, resid 1–306
```

Similarly, to using the range (−) symbol, you can use the addition (+) syntax. The following command will select the residues in range 30 to 35 as well as residue number 40:

```
select resid 30–35 + 40
```

You can also combine selection statements using the AND/OR operators. AND will select the atoms present in both the first and the second selections (intersection) while OR will select the atoms present in both selections (union). The following example will select five atoms corresponding to the Cα atoms in residues 30−35:

```
select resid 30–35 and name CA
```

1.3.8. Color Atom Selections. Apply different colors to the selection entries (protein, ligands). Coloring can be done using the command line; for example, color the ligand in red and the protein in blue for contrast:

```
color red, ligands
color blue, protein
```

Often, it is useful to color by atom elements, that is, to keep oxygen red, nitrogen blue, sulfur yellow, etc., and only adjust the color of the carbon atoms. This coloring option is available through **C** → **by element** → **CHNOS** (try it on the ligands selection). In the command line, this is less trivial but can be carried out by combining the color command with the selection statements:

```
color green, ligands and elem C
color red, ligands and elem O
color blue, ligands and elem N
```

All carbon atoms are colored green in the ligands selection, while keeping oxygen and nitrogen atoms red and blue, respectively.

1.4. Exploring the Binding Site of SARS-CoV-2 M^{pro} (Protein−Ligand Interactions).

Reverting to the default representation of SARS-CoV-2 M^{pro} PDB ID 6Y2G. Keep only protein and ligands by hiding the rest of the elements in the structure file, the nonbonded:

```
hide nonbonded
```

To simplify even more the visualization, we will only keep one protomer, by deleting chain B:

```
select chB, chain B
remove (chB)
```

With only chain A displayed, start by individually selecting the protein and ligand and show them as cartoon and sticks, respectively. Make sure you focus on the entire protein by clicking on the orient button. Next, change their colors to cyan and yellow, respectively:

```
color cyan, protein and elem C
color yellow, ligand and elem C
```

Explore the 13b ligand binding region in SARS-CoV-2 M^{pro} by representing the protein as surface:

```
as surface, protein
color grey70, protein
```

Zoom to the ligand using your mouse. We are now interested in the "binding site" region around the ligand. To explore this further, select and represent as sticks all residues within 5 Å of the ligand, by typing

```
show sticks, byres all within 5 of
ligand
```

You have now selected all residues within 5 Å, which should cover the most relevant residues interacting with the ligand, since H-bonds, ionic interactions, London dispersion forces, or Van der Waals interactions decrease significantly with the distance. Now, if necessary, hide the surface represented in the previous step:

```
hide surface, protein
```

Display the polar contacts between the ligand and the interacting residues:

```
A –> find –> polar contacts –> to any
atoms
```

PyMOL displays the polar contacts as yellow dotted lines and a new object is created "ligands_polar_conts".

Observe that 13b establishes several interactions with SARS-CoV-2 M^{pro} (Figure 5), including hydrogen bonds with F140, S144, C145, H164, and E166, for which you can also measure their distances (in Å). As an example, select the E166 residue and change its color to green:

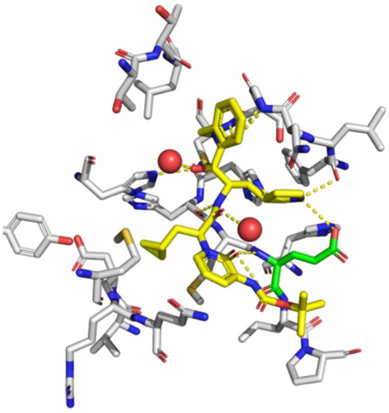


Figure 5. Representation of all polar contacts established by 13b and SARS-CoV-2 M^{pro}, including water molecules. All nonbonded atoms but the two interacting waters were hidden. The coloring by atom elements was applied, in which only the color of carbon atoms was adjusted: SARS-CoV-2 protein residues (gray), but residue 166 (green); the ligand (yellow); and the interactions between the ligand and the protein or water molecules within 5 Å (yellow dashed lines).

```
select E166, resi 166
color green, E166
```

Next, use the GUI window, click **Wizard** → **Measurement** and then in the main window click the two atoms whose distance you want to measure. Note that E166 establishes four possible hydrogen bonds with 13b, indicating that it is an important residue stabilizing the ligand in the binding pocket.

You may have noticed two additional polar contacts without a clear partner from the protein side. These contacts are probably established with H_2O molecules, which were previously hidden (when you typed '**hide nonbonded**'). Water-based interactions can be important in a variety of binding interfaces, therefore, to further analyze them, we need to show these nonbonded elements once more:

```
show nonbonded
```

It is now clear that those two hydrogen bond partners are indeed two water molecules. To better visualize these molecules, manually select them and change their representation to spheres (Figure 5):

```
S –> spheres
```

And the size of these water spheres can also be adjusted using

```
set sphere_scale, 0.5
```

1.5. Publication Quality Figures.

1.5.1. Ray Tracing for Better Resolution. PyMOL allows exporting images of the representations created within the software, but first you need to enhance the quality and resolution by ray tracing, a graphics technique able to smooth jagged edges and improve shadows to render high-quality 3D images. Therefore, type

```
ray
```

in the command line and observe the better quality when the ray tracing has completed. Ray tracing can be done in four different modes and can be set with the set ray_trace_mode command. You can change the ray trace mode to 1 with the command

```
set ray_trace_mode, 1
```

ensuring a normal color ray trace with a black outline, increasing the image contrast.

1.5.2. Adjust the Default Settings for a High-Resolution Image. For publications and to be environmentally friendly, one should use a white background. Set this by typing

```
bg white
```

To create a crisp picture, turn off ray_trace_fog and depth_cue:

```
set ray_trace_fog = 0
set depth_cue = 0
```

The ray_shadows setting can also be set to 0 or 1, depending on what you want. If you want PyMOL ray trace shadows:

```
set ray_shadows, 1
```

if not:

```
set ray_shadows, 0
```

To obtain smoother and better-looking images, antialiasing should be applied, by setting the antialias command. The highest antialiasing settings should be chosen (range: 0–4; 0 is off (default); 4 is the highest value):

```
set antialias, 4
```

If you want a full screen rendering, you need to turn off the internal gui, by typing

```
set internal_gui = 0
```

To set it to back on, type

```
set internal_gui = 1
```

When you have all parameters set, you can simple use ray command, to render the image, as stated in 1.5.1.

Personal preference obviously plays a role, but here are some additional settings that can be used to achieve a nice publication quality figure (see Figure 6 as an example):

```
set cartoon_fancy_helices, on
set cartoon_highlight_color, grey50
```

Also, you can ray trace specifying the width and height of the figure:

```
ray 1500, 1500
```

To save a high-quality figure just select

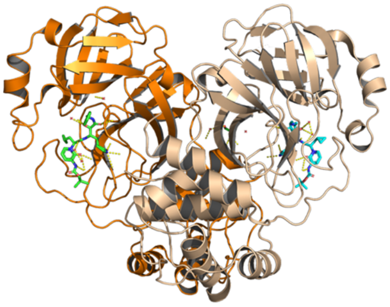


Figure 6. Representation of SARS-CoV-2 M^{pro} dimer using the settings described in the text. Orange represents one monomer complexed with 13b ligand (in green); beige represents another monomer complexed with 13b ligand (in blue). Protein is represented as cartoon, and ligands are represented as sticks. It is noteworthy that this figure is not related to steps 1–4 of Protocol 1.

File → Export Image As → [png]

and to save the session with all the modifications done in the meanwhile do

File → Save Session As → [pse]

Alternatively, you can use command line to save images and sessions:

```
png filename.png, dpi = 300
save filename.pse
```

Notes:

(1) If you want to show important parts of the complex or a particular angle, you should get the view and keep the coordinates of the "camera position" by typing:

```
get_view
```

The output, set_view, gives you the coordinates of the camera, the distance, the zoom and all the kind of information needed to recreate the exact view that we want to show previously, even if you change the angle.

(2) To edit all features mentioned above (in 1.5.2) to create high-quality images, you can use GUI interface (Settings → Edit All; to choose values double click in the second tab shown). Here, you can change other parameters to create a high-quality image, according to your personal preferences (by changing ray_trace_color, spec_reflect, etc).

1.6. Comparison of Multiple Different M^{pro} Structures.

It could become challenging to select the most suitable structure to carry on your research due to the increase in the number of M^{pro} structures recently solved. By searching into the PDB you can find many results regarding M^{pro} from different Coronaviruses or even from SARS-CoV-2. Also, both *apo* and *holo* M^{pro} structures (bound to different ligands) are available. PyMOL allows the comparison of different M^{pro} structures retrieved from the PDB, promoting a more rational choice of the structure to pursue the work. By using the align command, a sequence alignment of M^{pro} structures that we aim to compare is performed, followed by a structural alignment to minimize the RMSD between the alignment residues. Therefore, if you want to align two structures (structure 1 and structure2) loaded into PyMOL type the command

```
align structure2, structure1
```

If you want to specify part of structure 1 you know should match structure 2, you can specify the residues involved:

```
align structure2 and resi 1–150,
structure1 and resi 450–600
```

You can even narrow the alignment to the backbone atoms, typing

```
align structure2 and resi 1–150 and
name n+ca+c+o, structure1 and resi
450–600 and name n+ca+c+o
```

Notes: (1) If you want to align more than to structures (previously loaded into PyMOL) to a target structure use alignto command: *alignto target_structure*. See Box 2 for critical parameters and troubleshooting for Protocol 1.

Box 2. Protocol 1 Critical Parameters and Troubleshooting

The major problems using PyMOL are related to the use of the Object and Selection Menu Panel. One of the most common problems arises from the selection option. When a selection is shown (**S**), it is not removed unless you select hidden (**H**). This happens because selections are additive, allowing the generation of images representing a mixed graphical representation. Another important note to take into account is the need of saving PyMOL sessions periodically, since PyMOL lacks undo command. It is recommended to save different versions of the session during the work progress.

Since PyMOL is an open-source software widely used among the scientific community, it is probable that most common problems encountered using this tool have already been faced by someone else. To solve issues related with PyMOL use, you can subscribe PyMOL Users Mailing List (https://sourceforge.net/projects/pymol/lists/pymol-users) to exchange ideas, tips, and information with other knowledgeable users, as well as stay up-to-date on the most recent PyMOL news. PyMOLWiki (https://pymolwiki.org/index.php/Main_Page), which is a user knowledge database, can also guide you in tutorials, plugins, or answers for FAQs.

2. PROTOCOL 2: MODELLER—A TOOL FOR HOMOLOGY MODELING

The number of 3D structures available has been growing rapidly in the last two decades. Nevertheless, despite the recent technological improvements not all structures are available, and some are still quite scarce (e.g., membrane proteins). For molecules of which the structures are not yet resolved or are incompletely determined, *in silico* models can be devised. There are several approaches available in the structure modeling field, but they mainly fall into three categories:

- comparative modeling methods, which use evolutionarily related proteins with similar sequences
- *de novo* methods, which model all energetics in the folding process until a lowest free energy structure is obtained
- threading methods, which compare a target sequence against a structure library

From these approaches, comparative modeling, mostly known as homology modeling, remains the most accurate and widely used. The main objective of this method is to build a 3D model for the protein with an unknown structure (target) by sequence similarities to that of a known structure (template). This technique relies on the principle that the 3D structure of proteins has been conserved through evolution.

Any homology modeling protocol follows the general steps presented in Figure 7:

- Template identification: Good template selection is crucial to obtain an accurate model and several factors need to be considered such as sequence identity between template and target sequences (should always be > 25%), template resolution, phylogenetic similarity, and other environmental factors (presence of a ligand, mutations, pH).
- Sequence alignment: Alignment between target and template sequences and further corrections. Alignment may be difficult when a low percentage of sequence identity is observed. Clustal Omega, MUSCLE, and T-Coffee are the most widely used alignment methods.
- Model building: The information of the template structure and alignment is used to generate a 3D model of the target:
    - Backbone is the easiest part of the model and mostly trivial.
    - Loops are regions with higher structural variability and often not present in crystal structures. Nevertheless, the accuracy of loop modeling can determine the value of the whole model. One can use either knowledge-based methods or energy-based methods to predict these regions of the structure.
    - Multiple side-chain conformations are possible and the choice of a certain rotamer affects the neighbor residues and so on. Libraries of common rotamers in X-ray structures are used for successful side-chain placement.
- Model evaluation: Stereochemical analysis (bond length, torsion, and rotational angles) of the models is a necessary step of homology protocol. Ramachandran plots are powerful tools to assess structure quality.

There are several tools and web servers available to perform homology modeling, each with their algorithms, user-friendly, graphics-based or with command line-based interfaces. Here, we apply a simple homology modeling protocol based on the MODELLER software package.[7]

Homology modeling can be used to predict an entire biomolecule structure or to complete an experimental structure that might be missing some parts due to experimental difficulties. Since the 6Y2G structure is only missing a few residues at the C-terminal with no functional importance, we will be using an altered version of this structure with a deleted helix between residues 227 and 237 to illustrate the potential of this protocol. Readers interested in a more detailed and

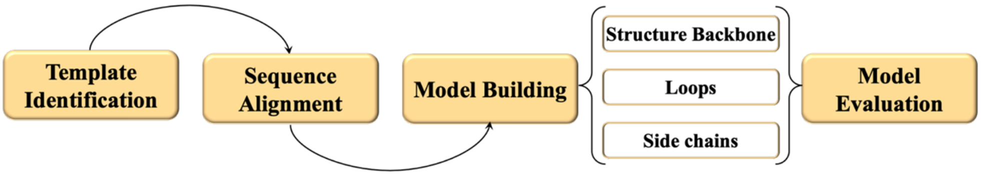


Figure 7. Workflow of Protocol 2.

dedicated tutorial about homology modeling using MODELLER software should refer to the original paper.[7] See Box 3 for necessary resources for Protocol 2.

Box 3. Protocol 2 Necessary Resources

Software: MODELER 9.24, matplotlib package, PyMOL.
Files: *.pdb, *.ali, *.profile, and *.py files.
Sample File: http://insilicotutorials.rd.ciencias.ulisboa.pt/modeller.zip, http://insilicotutorials.rd.ciencias.ulisboa.pt/model.html

2.1. Extract Files. The user needs to download the modeller.zip file provided with this tutorial. Several files can be extracted from the zip file:

- python scripts for the protocol (align.py; build_model.py; evaluate_model.py)
- mpro.ali file (an example of the input file)
- mpro-6y2g.ali: alignment file between M^{pro} full sequence and the sequence representing the altered structure 6Y2G (an example of a result file for the align.py step)
- 6y2g_edited.profile
- 6y2g_edited.pdb: the edited (incomplete) structure of M^{pro}
- 6y2g_edited.ali: the sequence of the incomplete structure in the PIR format (readable by MODELLER):

The first line contains the variable name for the sequence in the format ">P1;variable". The second line contains 10 colon-separated fields:

Field 1. Indication of the availability or not of a 3D structure and the type of method used to obtain. (sequence, sequence; structureX, X-ray; structureN, NMR).
Field 2. The PDB code or file.
Fields 3. Residue index for the first residue of the sequence.
Field 4. Chain identifier for the first residue of the sequence.
Field 5. Residue index for the last residue of the sequence.
Field 6. Chain identifier for the last residue of the sequence.
Field 7. Protein name.
Field 8. Organism of the protein.
Field 9. Resolution of the crystallographic analysis.
Field 10. R-factor of the crystallographic analysis.

Note: Fields 7−10 are optional. In case of sequences only the first and second field are necessary. "*" marks the end of the sequence. "/" marks a chain break. The user can also indicate in the sequence the presence of HETATM (ligand, DNA, RNA) or water residues using a '.' and 'w', respectively.

2.2. Target Sequence Preparation. The user needs to download the sequence of the M^{pro} (3C-like proteinase), which can be retrieved from UNIPROTid P0DTD1 (https://www.uniprot.org/uniprot/P0DTD1). However, this sequence covers all 7066 amino acid proteins, and we are only interested in a smaller part of the system. So, go to https://www.rcsb.org/fasta/entry/6Y2G/ and download the fasta file. Convert it into the ali format (mpro.ali) by using the predefined structure as in the below example. Be careful with the formation of the first two lines (the name used here is "mpro" as to be consistent throughout the protocol), add "." to identify the ligand in the 6Y2G structure, and add "*" to identify the ending of the protein file. Make sure that you have the repeated sequence as in the example below as we are dealing with a protein dimer:

2.3. Sequence Alignment. Use the script align.py to align the sequence retrieved from UNIPROT with the sequence representing the structure

```
python align.py
```

This file align.py could be used for a variety of modeling tasks. You just have to alter at the model *mdl* line, the name of the template file as well as their chains; and at the *aln.append* line the name of the file and code of the target sequence.

Notice that herein we consistently use 6y2g_edited for the template protein sequence and pdb file and mpro for the target sequence. Open the file created (alignment.ali; you could also change this name at the last line of the align.py file) and check for any inconsistencies in the alignment. Add "." at the end of each chain to consider the two ligand molecules in the model building. A correct alignment file is also provided if the user wants to skip this step (mpro-6y2g.ali).

Note: The sequence alignment will be used to arrange the backbone of the target according to the template structure. Hence, the quality of the models depends heavily on the quality of the alignment. Carefully check any misplaced indels in the alignment (this will happen more often in the loop regions, due to less conservation of the residues). Sequence viewers, such as Seaview,[8] can help in this step.

2.4. Model Building. The user now has all the necessary files for the building step: a template structure, a target sequence, and an alignment file. To build the model run:

```
python build_model.py > build_model.log
```

If you open *build_model.py* file you will see a section called *MyModel* in which specific instructions were given for this case, in particular the existence of symmetry between chains A and B of a dimer and the missing residues to be in an α helix format. The *env.io.atom_files_directory* line could be changed if your structure and alignment files are in different directories. Herein we presume that all files are in the same one. You can change the name of the alignment file, target, and sequence if you use other ones at lines 28−31. At line 36 you can change the number of

```
>P1;6y2g_edited
structureX:6y2g_edited.pdb:
1 : A:+586: B: main protease:Severe acute respiratory syndrome coronavirus
2:2.20:0.19
SGFRKMAFPSGKVEGCMVQVTCGTTTLNGLWLDDVVYCPRHVICTSEDMLNPNYEDLLIRKSNHNFLQAGNVQL
RVIGHSMQNCVLKLKVDTANPKTPKYKFVRIQPGQTFSVLACYNGSPSGVYQCAMRPNFTIKGSFLNGSCGSVGF
NIDYDCVSFCYMHHMELPTGVHAGTDLEGNFYGPFVDRQTAQAAGTDTTITVNVLAWLYAAVINGDRWFLNRFTT
TLNYEPLTQDHVDILGPLSAQTGIAVLDMCASLKELLQNGMNGRTILGSALLEDEFTPFDVVRQCS./SGFRKMA
FPSGKVEGCMVQVTCGTTTLNGLWLDDVVYCPRHVICTSEDMLNPNYEDLLIRKSNHNFLVQAGNVQLRVIGHSM
QNCVLKLKVDTANPKTPKYKFVRIQPGQTFSVLACYNGSPSGVYQCAMRPNFTIKGSFLNGSCGSVGFNIDYDCV
SFCYMHHMELPTGVHAGTDLEGNFYGPFVDRQTAQAAGTDTTITVNVLAWLYAAVINGDRWFLNRFTTTLNYEPL
TQDHVDILGPLSAQTGIAVLDMCASLKELLQNGMNGRTILGSALLEDEFTPFDVVRQCSGV.*
```

```
>P1;mpro
sequence:mpro:1: A : 614  :B :undefined:undefined::
SGFRKMAFPSGKVEGCMVQVTCGTTTLNGLWLDDVVYCPRHVICTSEDMLNPNYEDLLIRKSNHNFLVQAGNVQL
RVIGHSMQNCVLKLKVDTANPKTPKYKFVRIQPGQTFSVLACYNGSPSGVYQCAMRPNFTIKGSFLNGSCGSVGF
NIDYDCVSFCYMHHMELPTGVHAGTDLEGNFYGPFVDRQTAQAAGTDTTITVNVLAWLYAAVINGDRWFLNRFTT
TLNDFNLVAMKYNYEPLTQDHVDILGPLSAQTGIAVLDMCASLKELLQNGMNGRTILGSALLEDEFTPFDVVRQC
SGVTFQ./SGFRKMAFPSGKVEGCMVQVTCGTTTLNGLWLDDVVYCPRHVICTSEDMLNPNYEDLLIRKSNHNFL
VQAGNVQLRVIGHSMQNCVLKLKVDTANPKTPKYKFVRIQPGQTFSVLACYNGSPSGVYQCAMRPNFTIKGSFLN
GSCGSVGFNIDYDCVSFCYMHHMELPTGVHAGTDLEGNFYGPFVDRQTAQAAGTDTTITVNVLAWLYAAVINGDR
WFLNRFTTTLNDFNLVAMKYNYEPLTQDHVDILGPLSAQTGIAVLDMCASLKELLQNGMNGRTILGSALLEDEFT
PFDVVRQCSGVTFQ.*
```

```
# Read structure:
aln = alignment(env)
mdl = model(env, file='6y2g_edited', model_segment=('FIRST:A','LAST:B'))
aln.append_model(mdl,                          align_codes='6y2g_edited',
atom_files='6y2g_edited.pdb')

#Read sequence:
aln.append(file='mpro.ali', align_codes='mpro')
```

output models from the current 5 to other value. As a rule of thumb for more complicated cases use 100. The script has many comments inside to clarify the process, step-by-step. This will create five 3D models of the complete M^{pro} structure (mpro.B99990001.pdb to mpro.B99990005.pdb).

2.5. Model Evaluation. Generally, a set of models are generated and therefore it is necessary to select the best one. There are several scoring functions available to characterize and rank these models. MODELLER offers its objective function (molpdf) that is calculated by default and reported inside the model.pdb file. Within MODELLER the user can also use DOPE and GA341 scores that are reported at the end of the log file. The molpdf and DOPE score values should only be compared within the same alignment. The user can extract the score values from the end of the log file and rank the models produced by lowest molpdf and DOPE score. Alternatively, other web-based scores, such as ProSA[9] and ProQ,[10] can also be used.

The python script evaluate_model.py is available to further evaluate the DOPE potential per residue of each model. In line 11 you can change the name of the output model from a previous step that you intent to model. This script will create the mpro.profile which can be used as input for plots:

```
python evaluate_model.py
```

The line plot in Figure 8 was built using plot_profiles.py script (matplotlib package necessary) to compare model 5 and template profiles as an example. Lines 31, 34, and 37 use the alignment, template, and target ids, so change it for other modeling tasks or if you used different names from the ones given in this protocol:

```
python plot_profiles.py
```

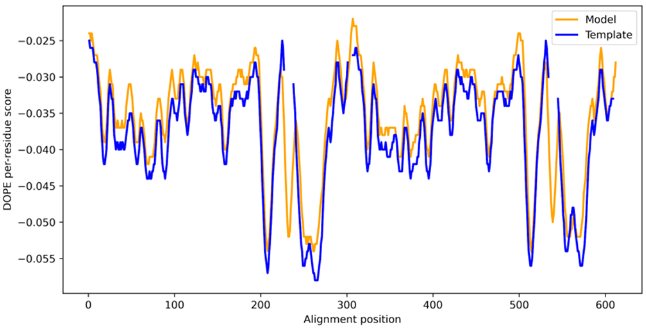


Figure 8. DOPE per residue score for model 5 (orange) and Template (blue). The gaps in the plot correspond to the gaps in the alignment between the two sequences.

The user is encouraged to use this script to analyze the DOPE profile of other models. A direct comparison between the values of models and template should be avoided, but the user can assess the overall quality of the alignment used.

2.6. Model Visualization. Although scores give a general picture of the model quality, visual inspection is always important to evaluate models at the residue level. The user should apply the knowledge from the previous protocol and align the "best" model, the template used (6y2g_edited.pdb) and the original 6y2g in a PyMOL session to compare the modeled helix with the one experimentally determined in the crystallographic structure (Figure 9). Please note that in most cases, the real solution (the original 6y2g, in our case) is not available and we will rely only on the alignment scores and visual inspection to assess the model quality. See Box 4 for critical parameters and troubleshooting for Protocol 2.

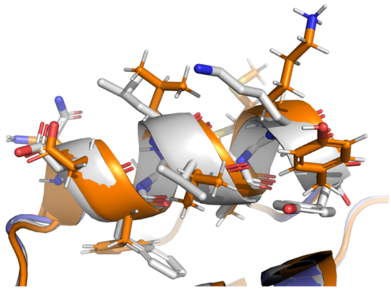


Figure 9. Comparison of the helix between residues 227 and 237 in the 6Y2G structure (white cartoon and sticks) and in model 5 (orange cartoon and sticks).

3. PROTOCOL 3: HADDOCK—A DOCKING TOOL FOR LIGAND–PROTEIN INTERACTION

Molecular docking is one of the widely used structurally based approaches, which allows the study of molecular recognition and prediction of the binding mode and binding affinity of a complex formed by two or more constituent molecules with known structures. The aim of this protocol is to provide basic knowledge on running the freely available web server HADDOCK (https://bianca.science.uu.nl/haddock2.4/).

Box 4. Protocol 2 Critical Parameters and Troubleshooting

In Protocol 2, MODELLER was used to prepare the alignment of the sequences, and build and evaluate the models. The most common mistakes by beginners include inconsistencies in the residue numbering in the .ali files and suboptimal alignments. Always compare the .ali with the pdb files of the template and check the alignment file for major errors (minor adjustments on the alignment are often necessary after visual inspection of the models).

Other minor problems may occur during protocol execution that are not necessarily linked to the homology modeling technique but to user inexperience to programming, in particular, to python programming, the language which MODELER is built in. For this reason, we encourage the user to open and read the log files. Additional help can be found in the official manual of MODELER available online.

HADDOCK is a docking method driven by experimental knowledge searching for information about the interface region and/or their orientations relative to mass spectrometry and NMR databases, among others. When experimental information is not available, bioinformatic interface predictors can also be used. HADDOCK stands out among other docking programs for allowing conformational changes of the molecules during complex formation, not only of the side chains but also of the backbone. Additionally, HADDOCK directly supports the docking of NMR structures and other PDB structures containing multiple models. Therefore, this web server constitutes a valuable tool using a data-driven docking approach.[11] In HADDOCK, experimental data are given in the form of active and passive residues. These are converted by HADDOCK into Ambiguous Interaction Restraints (AIRs), or the residues at the interface for each molecule used to drive the docking. The result is an energy function accounting for all interactions and from which we want to achieve its minima. After identifying the key residues at the interface, HADDOCK calculates and evaluates all distances between the interfacial groups between molecules. This molecular docking procedure brings the molecules together with random interface orientations, for which the final binding score is calculated. The docking process relies on three steps (it0, it1, and itw) of gradient driven energy minimization and molecular dynamics (MD) simulations. In the it0 stage, molecules are rigid, and the energy minimization relies strongly on AIRs data. The second stage, it1, introduces flexibility and increases temperature while maintaining restraints on the interface, allowing its packing. The final stage, itw, adds in solvation and short MD simulations. In the end, a clustering analysis is performed to identify the best binding modes of all the docking solutions from the itw stage.[11] See Box 5 for necessary resources for Protocol 3.

Box 5. Protocol 3 Necessary Resources

Software: An up-to-date web browser, PyMOL, text editor software, guru access level account to HADDOCK's web server.

Files: *.pdb files to submit to HADDOCK in a single folder.

Sample File: http://insilicotutorials.rd.ciencias.ulisboa.pt/haddock.zip, http://insilicotutorials.rd.ciencias.ulisboa.pt/haddock.html

Note: When not mentioned otherwise, the default parameters were used since they were tuned for the most common and basic usage.

In this tutorial, we demonstrate three different molecular docking methods within the HADDOCK software as shown in Figure 10. This diagram helps understand the workflow and connection between its different stages.

3.1. Extract Files. The user needs to download the haddock.zip file provided with this tutorial. After decompressing the zip file, the user will have

- initial structures for docking: 6y2g_mpro.pdb and 6y2g_ligand.pdb
- docking results: top10_resi, top10_blind, top10_rp, 433_resi, 433_blind, 433_rp
- PyMOL session: 6y2g.pse

3.2. File Submission. *3.2.1. Open and Log in to the HADDOCK 2.4 Web Server (https://wenmr.science.uu.nl/haddock2.4/).* If you have not yet registered and applied to guru access, please do so.

3.2.2. Go to Submit a New Job. Once in the submission page, fill out the information in the "Input data" tab (refer to Figure 11).

- **Job name**: Insert the job name, please consider that upon multiple queries submission, a good description will help you identify each job later.[1]
- **Number of molecules**: In this case, select 2 since the docking will be performed between a ligand and an apo protein. Nevertheless, keep in mind that HADDOCK can dock up to 20 molecules simultaneously.[2]
- **Where is the structure provided?**: Frequently, a modified file is submitted as is the case, but there is an option to upload the desired PDB ID.[3]
- **Which chain of the structure must be used?**: In this case, both molecules only have 1 chain, so select "All", but this option should be adjusted if the .pdb files have multiple chains.[4]
- **PBD structure to submit:** Select the file with the apo form of SARS_CoV_2 protease.[5]
- **What kind of molecule are you docking?:** Choose from the dropdown menu which kind/type of molecule is to be docked. In this case, choose "protein/peptide/ligand".

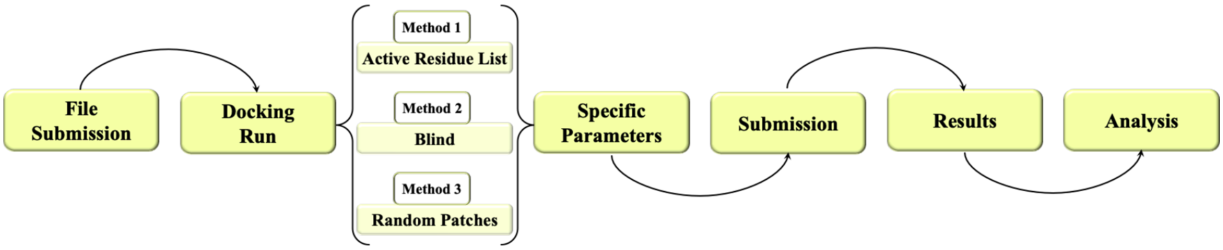
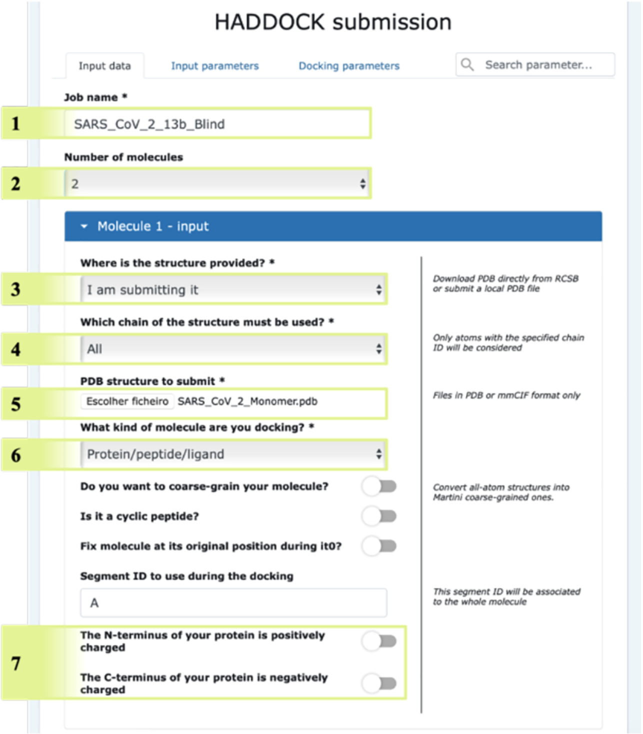


Figure 10. Workflow of the Protocol 3.



Job name *
SARS_CoV_2_13b_Blind

Number of molecules
2

Molecule 1 - input

Where is the structure provided? *
I am submitting it
Download PDB directly from RCSB or submit a local PDB file

Which chain of the structure must be used? *
All
Only atoms with the specified chain ID will be considered

PDB structure to submit *
Escolher ficheiro SARS_CoV_2_Monomer.pdb
Files in PDB or mmCIF format only

What kind of molecule are you docking? *
Protein/peptide/ligand

Do you want to coarse-grain your molecule? ○
Convert all-atom structures into Martini coarse-grained ones.

Is it a cyclic peptide? ○

Fix molecule at its original position during it0? ○

Segment ID to use during the docking
A
This segment ID will be associated to the whole molecule

The N-terminus of your protein is positively charged ○

The C-terminus of your protein is negatively charged ○

Figure 11. First step of web server submission. A description of the presented green boxes is available in the text.

HADDOCK can deal with different types of macromolecules such as protein, peptide, or even heteroatom ligands.[6]

- The ionization states for the N- and C-terminus can also be selected.[7] If using a complete protein structure, ionize/charge their termini, since this will be the most abundant species in solution at neutral pH. However, if the protein structure is somehow truncated at the termini, which is quite common for crystallographic structures, then leave them in their default neutral forms.

Repeat the above steps adequately for molecule 2 (ligand) and click NEXT. The next tab, "Input parameters" will open.

3.3. Setup the Docking Run.

On the parameter's setup stage, decide which are the best conditions for the docking run. In this protocol, we will perform and compare three different docking approaches: Active and/or passive residues list, blind docking, and random patches docking.

Note: Several authors[12−14] have described in-detail the different approaches tested herein, so read and explore these papers to understand the parameters and settings adopted in this noncomprehensive protocol.

3.3.1. Method 1: Active and/or Passive Residues List.

The user can make use of a true integrative approach and provide the key residues (active) located at the interface. In this case, there is experimental data available from which active residues are identified. Active residues are defined as the ones that establish hydrogen bonds between the SARS-CoV-2 protease and the 13b ligand.[15] Usually, active residues can display higher values for chemical shifts on NMR experiments, high exposition to solvent essential for binding, etc. If docking solutions do not include all active residues, penalties are applied, resulting in a lower probability of having a good model. Complementarily, passive residues are also important to define the interactions and the complex stability, but not pivotal.

To perform (active) residue-oriented docking, in the **Input parameters** page, provide active and passive residues. After providing a list of active residues, automatically define passive residues around the active residues or provide a list of passive residues when available. Show options for **Molecule 1** (the M^{pro} protein) and fill them according to the instructions below:

- Active residues (directly involved in the interaction): 1, 41, 49, 140, 143, 144, 145, 163, 164, 168, 189.
- Automatically define passive residues around the active residues: switch the toggle on.

Notes: (1) The active residues list should also include the residue number that corresponds to the ligand molecule.

(2) Keep in mind that there are different sources to provide active and/or passive residues apart from experimental information. There are tools like cPort[16] (https://alcazar.science.uu.nl/services/CPORT/) or SpotONE[17] (http://www.moreiralab.com/resources/spotone/) that can compute these parameters.

HADDOCK supports two ways to input active and/or passive residues: manually add the list of the residues or select them from the FASTA sequence displayed under **Active/Passive residues**. Additionally, check the selected residues in a 3D view by accessing the **Visualize residues** button. Once all the prior steps are complete, move forward to the last tab, **Docking parameters**, where further parameters must be adjusted for protein−ligand docking.

Show options in **Clustering parameters** and fill them according to the instructions below:

- Clustering method (RMSD or Fraction of Common Contacts (FCC)): RMSD.
- RMSD Cutoff for clustering (recommended: 7.5A for RMSD, 0.60 for FCC): 2.0.

Continue with adjustments to **HADDOCK score settings** and **Sampling parameters**. Since an active residue list was provided without any additional information, we will consider the rest of the parameter fields as default. Show options under **Scoring Options** and fill them according to the instructions below:

- Evdw > Evdw 1: 1.0.
- Eelec > Eelec 3: 0.1.

Then, proceed to **Advanced sampling parameters**, show options under **it1 parameters,** and fill them according to the instructions below:

- Initial temperature for second TAD cooling step with flexible side chain at the interface: 500.
- Initial temperature for third TAD cooling step with fully flexible interface: 300.
- Number of MD steps for rigid body high temperature TAD: 0.
- Number of MD steps during first rigid body cooling stage: 0.

At this stage, everything is ready to submit the files on the web server and wait for the results.

HADDOCK's Web site displays a page that you can refresh and see the progress, but, once complete, the results will also be e-mailed to your account.

3.3.2. Method 2: Blind Docking.

Blind docking is used when there is no information about the binding sites. This can be helpful to identify brand new active sites at the protein's surface. With improved scoring functions and using Artificial Intelligence (AI) methods, docking results are becoming more reliable even when no information is provided. Nonetheless, it is always better to gather knowledge prior to docking or guide the docking process by using different methods for distance restraints.

After going through the file submission, click next in the **Input parameters** tab and proceed to the next tab: **Docking parameters**. Here, in **Docking parameters** activate the **Surface contact restrains** toggle.

Since no further details on the binding location are provided, the sampling and scoring parameters must be suited to improve the statistical analysis performed by the HADDOCK algorithms, similarly to the **Clustering parameters** of Method 1. Also, proceed to the changes mentioned before for **Advanced sampling parameters** in Method 1, while making sure that the list of active residues at the Web site is completely empty. Then, adjust the **Sampling parameters** according to the following instructions:

- Number of structures for rigid body docking: 10000.
- Number of structures for semiflexible refinement: 400.
- Number of structures for water refinement: 400.

An increased sampling is used to improve our probabilities of attaining an accurate docking decoy. At this stage, everything is ready to submit the files on the web server and wait for the results.

3.3.3. Method 3: Random Patches Docking. Random patches are one of the 3 types of "Distances restraints" encompassed by HADDOCK. In this method, the software will randomly choose regions (patches) of all the surfaces to be considered as active residues in both partners. In this section, we will perform a docking experiment with random patches distance restraints and compare it to other methods.

After going through the file submission, click next in the **Input parameters** tab and proceed to the next tab: **Docking parameters**. Here, activate the **Random patches** toggle. Additionally, perform the necessary changes mentioned in Method 2. At this stage, everything is ready to submit the files on the web server and wait for the results.

3.4. Understanding Results. A compressed file with all data and graphical representations can be downloaded from the results page sent via e-mail. Typically, HADDOCK's Web site only keeps output files for a week, but it provides the hyperlink where the user can download the folders and/or files regarding the docking run. A summary section is also provided, where you can find an overview of how your docking results can be clustered/grouped. In the same page from HADDOCK's Web site, you will find top structures and graphs. All data and graphical representations are included in the already mentioned compressed file and accessible using the "index.html" file.

Note: As previously mentioned, after you create an account and submit a job, HADDOCK will send an email to the address registered with the link to access and download the compressed file, which remains available for one week after completion.

In this protocol, two different approaches were considered in the results analysis. First, the 4 3 3 approach where the user retrieves the clustering results provided by HADDOCK. The 4 3 3 method considers the 10 best complexes selected by combining the top 4 complexes of the best cluster and the top 3 complexes from the second and third clusters. The other approach considers the Top 10 complexes regarding only the HADDOCK score as criterion.

3.4.1. Haddock Score. The top 10 results are sorted by lowest HADDOCK score considered as the most likely solutions according to HADDOCK standards. HADDOCK score is a weighted sum of van der Waals, electrostatic, desolvation, and restraint violation energies together with buried surface area. The accuracy of the energies and scores used in HADDOCK is highly dependent on the starting structure quality. The results will not be meaningful if a poor approximation of the molecule's conformation in the complex form occurs. Thus, other evaluation parameters should be applied to evaluate the best structures.

3.4.2. Root-Mean-Square-Deviation. RMSD measures the deviation from the molecules in the complex structure to the reference. PyMOL is used to align each structure to the reference and report the RMSD obtained.

3.4.2.1. Top 10 HADDOCK Score Approach. The user should download and extract the compressed file. Inside the results folder, we can open with a text editor the *file.list* file as

structures → *it1* → *water* → *file.list*

In the file.list the user can check the Top10 structures in this run as classified by the program. Table 1 provides an overview of the results obtained for the docking for the three methods already with RMSD calculation included along with cluster identification.

Table 1. Results Obtained in the Three Docking Protocols

complex	cluster	HADDOCK score[b]	RMSD
		Method 1: Active and/or Passive Residues List	
197w	1	−51.64421	0.498
198w	**1**	**-51.39445**	**0.483**
2w	1	−51.34778	0.503
6w	1	−50.76358	0.484
14w	1	−49.92913	0.489
1w	1	−49.73249	0.477
21w	1	−49.48172	0.484
5w	1	−49.02384	0.488
177w	1	−48.95887	0.632
25w	1	−48.50288	0.537
		Method 2: Blind Docking	
128w	13	−48.06723	4.283
106w	13	−46.20571	4.294
177w	14	−45.17535	4.348
131w	13	−45.05686	4.261
148w	13	−43.95589	4.328
2w	NA[a]	−43.76324	4.260
314w	NA	−43.64583	1.049
1w	**NA**	**-43.54786**	**0.502**
28w	2	−43.37321	4.604
27w	2	−43.16816	4.587
		Method 3: Random Patches	
130w	NA	−40.34081	1.534
267w	2	−39.87502	6.657
133w	NA	−39.29341	4.317
177w	1	−38.49039	3.936
264w	2	−38.12264	6.639
231w	NA	−37.75637	3.828
165w	1	−37.45315	3.977
92w	2	−37.45132	6.707
153w	**NA**	**-37.31763**	**1.174**
60w	3	−37.22954	4.257

[a]NA: not part of a cluster. [b]The data were obtained for the HADDOCK score and RMSD based on the Top10 structures from the HADDOCK web server.

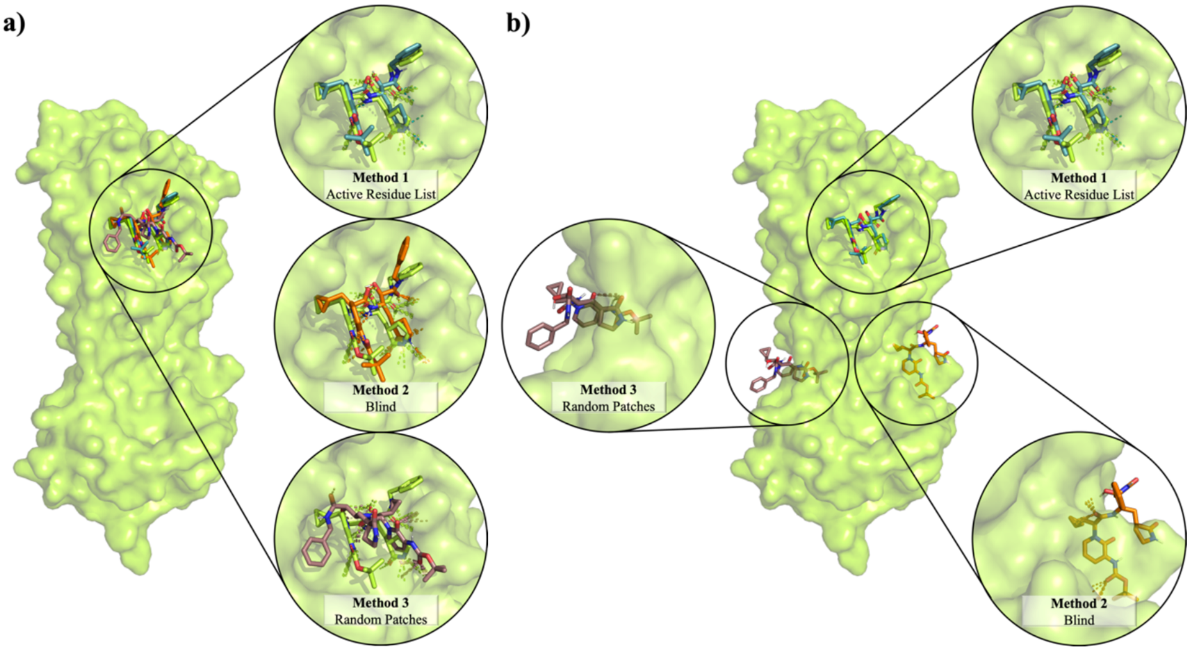


Figure 12. Top docking results according to HADDOCK score and RMSD values. (a) Top10 results: general overview of the docking result on the left with zoom-in for each method performed on the right. (b) 4 3 3 results: general overview of the docking result on the left with zoom-in for each method performed on the right. (Green ligand, crystallographic structure; blue, method 1 top pose; orange, method 2 top pose; violet, method 3 top pose).

The complex with the best HADDOCK performance (most negative HADDOCK score) is obtained through Method 1, for which the user provides the list of active residues. The choice of the best complex will have to be based on the compromise between the one with the lowest RMSD (closer to 0) and lowest HADDOCK scoring. Thus, complex_198w, complex_1w, and complex_153w seems to be the best structures for Methods 1, 2 and 3, respectively. The three top results are depicted in Figure 12a.

3.4.2.2. 4 3 3 Approach. The clustering results can be verified in the results web page for each method. Apart from the results' clustering from Method 1 where an active residues list was supplied and all models were very similar (197 (out of 200) models representing 98% of all available data), no docking decoy pose was attained in methods 2 and 3 as the most representative.

Inside the results folder, the user can find "file.list_clustX" file and check the best structures in each run as classified by the program:

structures → it1 → water → file.list_clustX

Table 2 provides an overview of the results obtained taking the 4 3 3 approaches into account. Looking at file.list_clustX file for each method shows that in Method 1 only two clusters were considered. Thus, we considered the six best structures from the top performing cluster (Cluster 1) and the four best structures from the second-best performing cluster (Cluster 2). In Methods 2 and 3 the 4 3 3 was fully considered. For each complex, HADDOCK score and RMSD are displayed.

The complex with the best HADDOCK performance (most negative HADDOCK score) is obtained through Method 1, for which the user provides the list of active residues. Thus, complex_1w, complex_13w and complex_196w seems to be the best structures for Methods 1, 2 and 3, respectively. The three top results are depicted in Figure 12b.

HADDOCK uses, by default, the Fraction of Common Contacts (FCC) clustering algorithm since it is a quick and easy method to cluster conformations according to their conformational similarity based on atomic contacts.[18] This clustering method is less time-consuming and fit for biomacromolecular complexes where it yields better results than the standard interface backbone RMSD. However, in the case of the ligand-protein system presented here, the RMSD is a great tool to evaluate the complexes since the difference in RMSD values is mostly attributed to the changes in ligand's position and orientation. This can be assessed in the complexes output as we have the crystal structure of the ligand−protein complex. As shown in Figure 12 (top panels), the best docking solutions of 13b inhibitor obtained through HADDOCK are very similar to its cocrystalized conformation. See Box 6 for critical parameters and troubleshooting for Protocol 3.

4. PROTOCOL 4

4.1. AutoDock 4 and AutoDock Vina—Two Tools for Structure-Based Virtual Screening. The identification and optimization of lead compounds are inherent components in drug design and discovery pipelines. Structure-based virtual screening (SBVS) is a computational approach used in early stages of drug discovery campaigns to search within compound libraries for new bioactive molecules targeting a specific protein drug target.

This approach takes advantage of the available 3D structural information of the biological target (obtained from X-ray, NMR, homology modeling, etc.), to dock a series of different chemical compounds at its binding site and select a small subset of chemical entities based on predicted binding scores for further biological evaluation.[19] During the past decade, molecular docking algorithms have arisen as a valuable tool in such approaches. However, the conformational search for the docking poses, together with the scoring problem, has pushed the development of new algorithms and computational protocols to overcome the well-known problems associated with this type of methodology. Currently, AutoDock 4

Table 2. Results Obtained in the Docking Protocols

complex	cluster	HADDOCK score[a]	RMSD
Method 1: Active and/or Passive Residues List			
197w	1	51.64421	0.498
198w	1	51.39445	0.483
2w	1	51.34778	0.503
6w	1	50.76358	0.484
14w	1	49.92913	0.489
1w	**1**	**49.73249**	**0.477**
194w	2	31.99595	1.513
99w	2	18.86900	1.495
152w	2	15.52363	1.643
160w	2	15.47099	1.429
Method 2: Blind Docking			
128w	13	48.06723	4.283
106w	13	46.20571	4.294
131w	13	45.05686	4.261
148w	13	43.95559	4.328
28w	2	43.37321	4.604
27w	2	43.16816	4.507
45w	2	42.39155	4.612
16w	1	43.13368	4.280
7w	1	42.20694	4.256
13w	**1**	**41.98818**	**4.250**
Method 3: Random Patches			
267w	2	39.87505	6.657
264w	2	38.12264	6.639
92w	2	37.45132	6.707
49w	2	36.89445	6.685
300w	5	40.34081	4.264
60w	5	37.22954	4.257
55w	5	35.56638	4.269
177w	1	38.49039	3.936
165w	1	37.45315	3.977
196w	**1**	**35.33295**	**3.920**

[a]The data was obtained for the HADDOCK score and RMSD based on the Top10 structures from HADDOCK web server.

Box 6. Protocol 3 Critical Parameters and Troubleshooting

Most errors in the docking process are linked to the initial files submitted and as such use as high-quality structures as possible. HADDOCK has some clear rules about the formatting of the PDB files to be submitted. If submitting ensembles, make sure that all share the same organization in terms of atoms, residue numbering, and nomenclature. Regarding residue numbering, there are several aspects to be considered such as no overlapping numbering, only numeric characters in residue numbering, providing charges for elemental ions, cofactors must be designated as HETAM lines, and some modified atoms can be submitted according to HADDOCK parameters (http://haddock.science.uu.nl/services/HADDOCK2.2/library.html).

To address these PDB files problems, you can use either LEaP (https://ambermd.org/tutorials/pengfei/index.htm) or PDB-Tools (http://www.bonvinlab.org/pdb-tools/).

Docking methods yield better results as the information given is of higher quality as well. As mentioned during the Protocol 3, you should preferentially direct HADDOCK's docking protocol by providing lists of active and/or passive residues. If you do not possess experimental data, use other platforms to attain these residues such as cPort or SpotONE. If you choose not to calculate the interfacial residues via other platforms, then, increase sampling as demonstrated in Methods 2 and 3, to increase the probability of finding the correct pose.

Once again, be mindful that the results are kept in HADDOCK's web server for just one week before being deleted, so download your results timely. Also, the file "job_params.json" has all the parameters for the run you submitted. Use it to replicate the same run or to change small details while keeping the major parameters.

(ADT4)[20] and AutoDock Vina (VINA),[21] are two of the most widely used free computer software in SBVS campaigns. In this protocol, both softwares are used to perform such demanding tasks, and the advantages and disadvantages of using their native algorithms to address the docking problem are evaluated. Although these softwares were developed in the same laboratory, they use different approaches and algorithms to perform the 3D conformational search of the protein−ligand complex and its binding scoring. As reported in their work, Forli *et al.*[22] performed a direct comparison between the two docking programs and concluded that overall the selection of one software over the other one depends on the target system. As highlighted by these authors, "The AutoDock Vina scoring function is highly approximate, with spherically symmetric hydrogen bond potentials, implicit hydrogens, and no electrostatic contribution. It has been shown to perform well with ligands with typical biological size and composition. The AutoDock force field includes physically based contributions, including a directional hydrogen-bonding term with explicit polar hydrogens, and electrostatics. If these contributions are important in a particular system, AutoDock would be the appropriate tool.". In this protocol, we intend to provide the user with an overview of the necessary workflow to enroll in a SBVS campaign of a small database of compounds into the binding site of the M^{pro} protein using both VINA and ADT4. Both programs have already been successful in identifying many new inhibitors

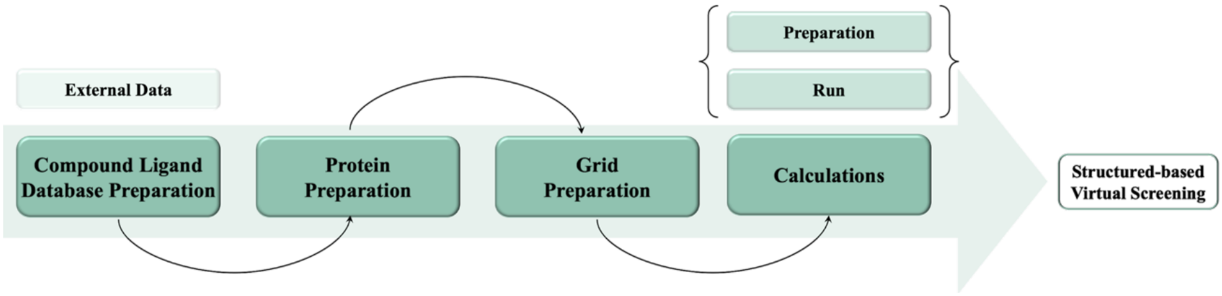


Figure 13. Workflow of Protocol 4.

for several protein targets.[23] Note however that the objective of this protocol is not to focus on the methodological aspects and detailed options of ADT4 and VINA, but on the SBVS protocol workflow (Figure 13), which will allow the user to identify new compounds that bind/inhibit M^{pro}, hence, halting the replication rate of the SARS-COV-2 virus. Readers interested in a more detailed and dedicated tutorial should read the paper published by Forli *et al.*[22] Nevertheless, despite being a tutorial for beginners, it assumes that the student is acquainted with a linux command line and knows the basic principles of file manipulations. See Box 7 for necessary resources for Protocol 4.

Box 7. Protocol 4 Necessary Resources

Software: Terminal, OpenBabel, Autodock Tools, AutoDock 4.2, AutoDock VINA and Datawarrior

Files: All the structures, input files and scripts to perform the calculations.

Sample File: http://insilicotutorials.rd.ciencias.ulisboa.pt/sbvs.zip, http://insilicotutorials.rd.ciencias.ulisboa.pt/adt.html

4.1.1. Extract Files. The user needs to download the sbvb.zip file where the required files to run and prepare all the calculations in this protocol are provided. After decompressing the zip file, the user will have the following folder/file organization:

- **scripts**. all the fully annotated scripts to prepare, run and analyze the calculations
- **Protein**. original .pdb file of the protein structure (6Y2G), and all treated files used in the calculations
- **DTB_compounds**. all the database compounds in different file formats
- **DTB_screening**. all the files obtained from the database screening using ADT4 and VINA
- **Validation**. all the files obtained in the validation protocol

4.1.2. Validation of the ADT4 and VINA Protocols—Using the GUI. We can take advantage of the currently available crystallographic structure of the complex to develop ADT4 and VINA protocols capable of mimicking the interaction pose of 13b at M^{pro}'s binding site. This will be accomplished using the menu-driven interface of ADTools– GUI approach (Figure 14).

4.1.2.1. Ligand and Protein Preparation. The initial step of setting up ADT4 and VINA calculations is to prepare the required protein and ligand files in the .pdbqt format. For simplicity, start by splitting the original 6Y2G.pdb file into two separate files: one containing chain A of the M^{pro} protein (protein.pdb) and another one containing solely the cocrystallized 13b ligand (ligand.pdb). Processing these files is easily done using the grep command available in any Linux command line (egrep "A" 6Y2G.pdb | egrep ATOM > protein.pdb ; egrep "A" 6Y2G.pdb | egrep O6K > ligand.pdb).

Start ADTools GUI and several action tabs will be available to set up, run, and analyze the ADT4 and VINA docking calculations (Ligand, Flexible Residues, Grid, Docking, Run, and Analyze).

Prepare the ligand by selecting the tab **Ligand** → **Input** → **Open**, and then navigate into the folder where the ligand.pdb file was previously saved. After completing these steps, ADTools will parametrize the ligand by adding hydrogens, computing the Gasteiger charges, assigning an AutoDock atom-type to each atom and, finally, defining the number of degrees of freedom detected in the ligand (torsions). The identification of these torsions is of utmost importance, since they will define the flexibility of the ligand which will be used in the calculations of both ADT4 and VINA. Typically, ligands with a higher number of torsions will have more complex conformational and interaction spaces at the binding site, which makes them more difficult to converge in the calculations. In these situations, we advise the reader to follow one of two alternatives: (1) look to the bonds between the atoms of the ligand and manually reduce the number of torsions/degrees of freedom to the minimum

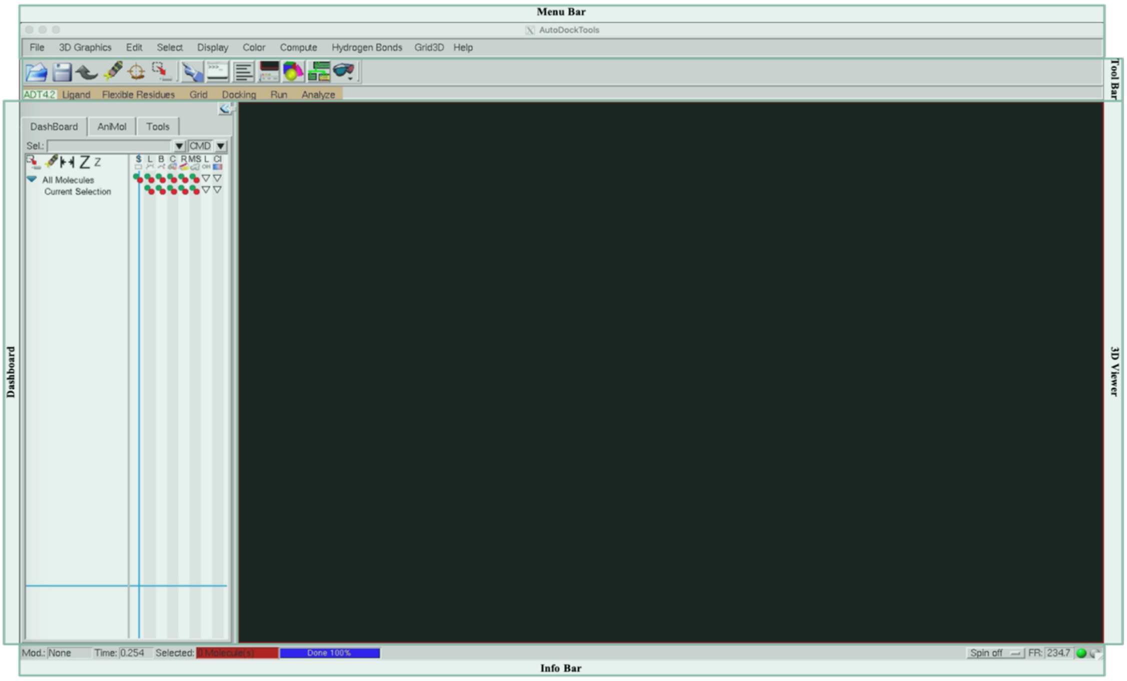


Figure 14. ADTools GUI representation.

required (depends a lot on the chemistry of the compound) by going to **Ligand → Torsion Tree → Choose Torsions** and then by clicking **SHIFT+LEFT MOUSE BUTTON** (the bonds will change their color from green, fully rotatable, to magenta, rigid); (2) change the settings of the docking protocol, to perform a more exhaustive conformational search of the docking poses (for this task, it is advised to look up the many available parameters in the AutoDock manual). This last option will compromise the speed of the ADT4 docking calculations (making it sometimes unfeasible to apply in the screening of databases with thousands of compounds. In VINA, this is not so much the case, since the number of tunable available search parameters is reduced to one—exhaustiveness (look into the manual of the VINA software for additional information regarding this matter).

For 13b, the number of rotatable bonds was reduced to 8 (all the bonds of the ligand connecting to different branches were rigidified) instead of the default 13 torsions identified. This number of torsions/degrees of freedom is too high (as a rule of thumb, the limit should be around 8 to 10 rotatable bonds), and if used, could decrease the convergence and accuracy of the predictions. However, be aware that applying this manual approach to a large compound database is unfeasible.

Set up the protein file similarly to what was applied to the ligand above. No flexibility will be defined to the side chains of the residues found in the M^{pro} binding site since the binding site is shallow and open with few amino acids with highly flexible sidechains. Therefore, skip the **Flexible Residues** ADTools tab, and prepare the protein and the grid used in the calculations.

To prepare the protein pdbqt file, go to the tab **Grid → Macromolecule → Open** and select the pdb file previously created (protein.pdb). Click the **OK** button, to check if hydrogens and charges have already been added to the structure. If the system has already some attributed charges, the program will ask if it should conserve them or not. In the end of this step, ADTools merges the nonpolar hydrogens, adds the previously mentioned charges, performs an atom-type assigning, and asks the user to indicate the folder where the protein.pdbqt file should be saved.

4.1.2.2. Preparation of the Grids for ADT4 Calculations. After preparing the protein.pdbqt file, define and create the grid region at the protein's surface on top of the active site, where the conformational search for ligand poses will be performed.

Navigate to **Grid → Set Map Types** (one map per atom type found in the ligand will be calculated) → **Choose Ligand** in the ADTools GUI. By selecting the **Ligand** option, ADTools automatically identifies the atom types of the ligand and consequently the grid maps that must be created. These steps are only required in the ADT4 calculations since VINA does not use them.

Define the position and size of the 3D space used in the conformational search during the docking calculations (grid dimensions). Focus on the binding site of the M^{pro} protein and choose the center of the grid box to be the center of the ligand position found in the crystallographic structure (this option is easily obtained from the **Grid Options menu**, in the **Center** tab), and setup the grid spacing to 0.375 Å and the size to enclose the entire binding site region of the protein. In this specific situation, use a 60-points grid size, in all x, y, and z coordinates.

Save the grid file (with the extension .gpf) by choosing **Grid → Output → Save GPF** in the desired folder. This file will be one of the input files necessary to run the autogrid program, which is the preparation step run prior to an ADT4 docking calculation.

4.1.2.3. Preparation of ADT4 and VINA Calculations. Parameterize the ADT4 docking calculation by selecting in the ADTools GUI **Docking → Macromolecule → Set Rigid Filename.**

Select the protein which has already been parametrized and proceed to **Docking → Ligand → Choose → Ligand** and choose the previously setup ligand.

Define the Docking and Search parameters to be used in **Docking → Search Parameters → Genetic Algorithm**. This option pops up a window with all default Genetic Algorithm search parameters already filled, which should be used as is.

To finish and save these settings, select **Docking → Output → Lamarckian Genetic Algorithm (GA)**, saving the docking file (with extension .dpf) in the folder where the docking calculations will run.

For the VINA calculations, select the option **Docking → Output → Vina config**. This option will fill the configurations to run the VINA calculation, based on the information provided in the previous inputs of the ADTools GUI. After confirming all the available data, save the config.txt file required as input for the VINA docking calculations.

4.1.2.4. Running ADT4 and VINA Calculations. The previous steps allowed the preparation of all the necessary files to run the ADT4 and VINA docking calculations. To fulfill this step, use the tab menu **Run** in ADTools.

For the ADT4 simulation, we will start by running the autogrid program (**Run → Run AutoGrid**) that generates the required grid maps. A new menu will pop-up and the user should indicate the paths to the "autogrid" program and to the previously created .gpf file. Press the "Launch" button to start autogrid and the menu will disappear. If desired, follow the calculations by selecting the menu **Run → Job Status**.

At completion, the grid maps will be available in the running folder, which allows the execution of AutoDock by selecting **Run → Run AutoDock** in ADTools. A new menu will appear, and again the paths to the AutoDock binary and the previously created .dpf file should be given. After all the required information is filled, press the "Launch" button to start the AutoDock calculation.

Repeat the same chain of instructions to run the VINA calculation: **Run → Run AutoDock Vina**.

4.1.3. Analysis of the Docking Validation Process Using a Graphical Approach. After successfully running all the calculations, proceed and analyze the obtained results from both programs ADT4 and VINA. To accomplish this task, take advantage of the **Analyze** tab in ADTools GUI. Load the results obtained from ADT4 and VINA using different types of representations (Protocol 1).

4.1.4. Compound Docking Screening Campaign with ADT4 and VINA—Using the Command Line. After the presented validation process, we will use both protocols to qualitatively correlate the docking results with bioactivity data (K_i) available from ChEMBL database for several M^{pro} inhibitors. If successful, we suggest building a new database, composed of compounds with unknown affinity to the M^{pro} protein, to find for novel inhibitors.

There are several available python scripts from ADTools, which allow us to automate many steps (preparation, run, and analysis) to screen for large compound databases. However, the reader must be aware that these scripts will only be able to run using the old python 2.5 version. If this version of python is not installed in the readers' computer, one can easily install using the traditional recipes (do a simple search on the Internet), or simply use the python 2.5 version, available in the binaries folder of ADTools.

4.1.4.1. Creating and Preparing the Compound Ligand Database for Binding Screening Calculations. Prepare the target database. Nowadays, freely available compound databases are available (such as ZINC, Chembl, NCI, among others). It is out of scope of this protocol to review the steps needed to create ligand conformers. Hence, all ligand files for this protocol are provided in the abovementioned "DTB_compounds" folder in .sdf format.

As previously mentioned, to run any docking calculation, it is mandatory that each compound is in the .pdbqt file format. To achieve the necessary file conversions (.sdf → .pdbqt) we first must convert all compound files from .sdf to .mol2 file format (using for example the program unicon), followed by a .mol2 to .pdbqt format using the command line python script prepare_ligand4.py from ADTools.

Since the installation and usage of ADTools in the different operating systems can be problematic (for example, in the MacOS systems, since the Catalina version, all 32-bit programs as ADTools, became unsupported), alternative programs capable to generate the ligands and protein pdbqt's are necessary. OpenBabel (OP),[24] a well-known computational chemistry software, can be used to perform such tasks. This software can interconvert the chemical structures of ligands and proteins between different file formats (from pdb/mol2/sdf/smiles to pdbqt), and at the same time assign Gasteiger charges and define the torsion tree information necessary to define the flexibility of ligands. In our scripts, we have included all the necessary instructions in how to use OB to perform such a task.

4.1.4.2. Preparing the Protein for the Screening with ADT4 and VINA. Use the protein.pdbqt file already created in subsection 4.1.2.1, or create it directly from protein.pdb, using the python script prepare_receptor4.py from ADTools. The command sequentially adds hydrogens and assigns AutoDock atom-types and Gasteiger charges to all the atoms of the receptor.

4.1.4.3. Preparing the Grid Files for the Screening with ADT4 and VINA. Take advantage of the work previously done under the validation step to identify both the grid center and grid dimensions. Regarding the VINA runs, use the same input file (config.txt) previously prepared during the validation step. The same does not apply for ADT4 screening runs. Since we are using multiple compounds, with a different chemical composition of 13b inhibitor, additional atom types of grid maps will be required. To create them, run the prepare_gpf4.py python script from ADTools with the -d flag that allows for the identification of all atom-types found in all compounds of the screened database (located in the provided folder). Additionally, the -p option can also be used within this command to setup multiple options such as the number of points of the grid and/or grid center coordinates.

4.1.4.4. Preparing the Docking Calculations for ADT4. Create the ADT4 input files for running the docking calculation for each compound found in the target database using the python script prepare_dpf4.py from ADTools. The user can integrate this command in a bash loop to ensure the automatic creation of all required .dpf files.

4.1.4.5. Running All Screening Docking Calculations for ADT4 and VINA. After all files required to run the docking calculations are prepared, use both ADT4 and VINA programs to perform the desired screening. The specific details to run these simulations in the command line should be adapted to the number of computational resources available. Since our target database is composed of only a few compounds, the running script (see Author's Note, and accounting for all the commands and instructions necessary to prepare and run all the docking screening calculations) can account only for sequential processes (without parallelization), assuring that in an hour, all the calculations are finished even when using a laptop or a small desktop computer.

4.1.5. Analysis of the Docking Screening Campaign of a Compound Database Process. After all docking calculations are finalized, compile and organize the results according to the docking program used. As previously mentioned, our main objective is to obtain a reasonable correlation between the computational predictions and the experimental K_i data. If successful, we can expand the searchable database in future screening campaigns to help identify new promising inhibitors of the M^{pro} protein. There is a python script from ADTools (summarize_results4.py) that parses all docking output files and finds the lowest energy solution for each docking run, summarizing the energy and rank of the best solution in ADT4 runs. The -d option can be used to point to the PATH of the output log files generated in the dockings, while the -a option allows the script to append all relevant info into a single summary results file, created with the -o option. The *sort* command in bash coupled with a few "awk" instructions can organize compounds from lower to higher binding energies, allowing an easy way to quickly identify the most promising inhibitor candidates of the M^{pro} protein. A bash script can be found under the scripts folder, which integrates all this information and allows the generation of such log compilation files. Additionally, it is also possible to visualize in 3D the results using the write_lowest_energies.py python script from ADTools. When this command is applied in all ADT4 docking output log files, a .pdbqt file with the docking solution with lowest binding energy of that specific docking simulation is generated, which can then easily be visualized using PyMOL.

4.2. Results Interpretation. *4.2.1. Validation of the ADT4 and VINA Protocols—Using the GUI.* To evaluate in detail the docking solutions obtained from each docking protocol, start by looking at two complementary and useful data: the binding energy of each individual docking solution and also the cluster size defined by a RMSD-based clustering method (a RMSD cutoff is applied to define dissimilarity). Ideally, the most populated clusters show the lowest binding energies. Exceptions may appear caused by inefficient parameters, which can be fine-tuned to improve the conformational search process, or simply because the lowest binding energy solution is conformationally constrained.

To check the docking solutions obtained from each docking software, start by looking to the clusters obtained with ADT4. Load them by **Analyze → Dockings → Open** in the ADTools GUI and select the .dlg file generated by the docking run. After that, one can do **Analyze → Clusterings → Show**. A bar plot will pop-up, evidencing the clustering results obtained from the ADT4 simulations. As can be seen in Figure 15, the cluster with the lowest binding energy is the most populated, indicating good sampling in the calculation of our test case.

Look at the docking poses that represent the obtained clusters by following these instructions in the ADTools GUI: **Analyze → Conformations → Play, ranked by energy**. This allows the comparison between the different poses obtained in the ADT4

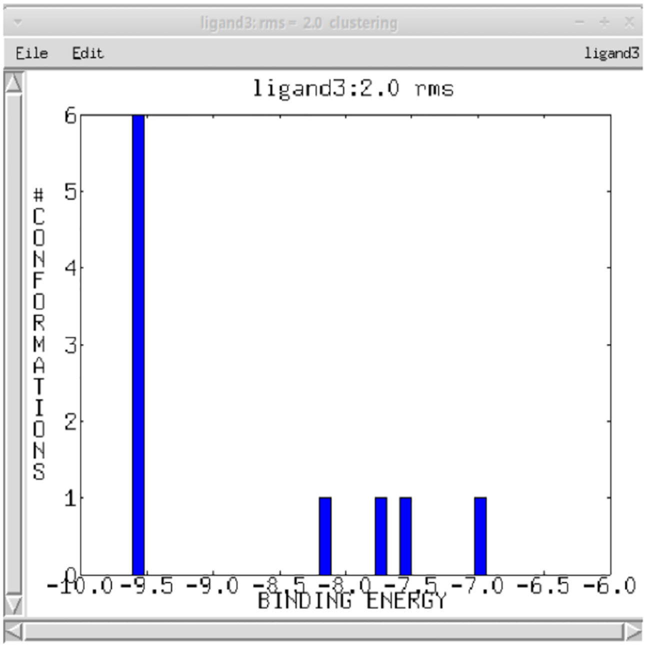


Figure 15. Clustering analysis of the docking results obtained from ADT4 calculations.

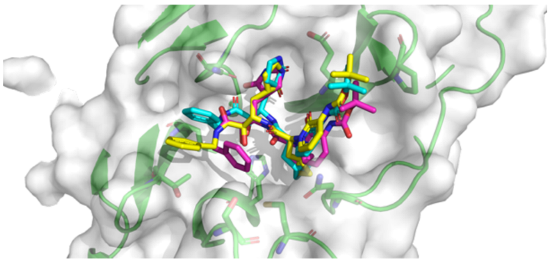


Figure 16. Superposition of the lowest binding energy docking results of the 13b inhibitor obtained from ADT4 (in cyan), VINA (in yellow), and the X-ray structure (PDB ID: 6Y2G;[3] pink) in the M^{pro} binding site. The protein surface is colored in white, while the secondary structure of the protein chain A is represented in the green cartoon. The side chains of the most important residues found at the protein binding site are represented as green sticks. This figure was built and rendered using the program PyMOL.[6]

calculations. As can be seen from Figure 16, there is a significant conformational overlap between the docking pose with the lowest binding energy and the cocrystallized conformation of 13b. Furthermore, by selecting **Analyze → Dockings → Show Interactions**, a radically different display will be obtained, in which the ligand is shown with a solvent-excluded molecular surface, and receptor atoms involved in hydrogen bonds or in close contact to the ligand are shown as spheres.

The same analysis can also be performed for the VINA docking results with **Analyze → Open AutoDock Vina result.** Loading the output results found in the output.pdbqt file we can identify the multiple solutions obtained from the docking calculations, already ranked by energy (those with lowest binding energy come first). To go through these conformations, just use the arrow keys on the keyboard. In the VINA calculations, clustering is performed internally, and all solutions presented already have an entropic correction (proportional to the size of its cluster) in their binding affinity values. Therefore, to identify the best inhibitors in the M^{pro} binding site, we can simply look at the conformations with the lowest binding energies. As can be seen in Figure 16, the best/lowest energy docking solution of 13b inhibitor obtained from the VINA and ADT4 calculations are very similar to its cocrystallized conformation. Furthermore, it should be noted that all docking methodologies used in this work (HADDOCK, Autodock 4, and Vina) were able to reproduce the X-ray pose of α-ketoamide 13b inhibitor, showing their reliability and usefulness.

4.2.2. Compound Docking Screening Campaign with ADT4 and VINA—Using the Command Line. As previously mentioned, the objective was to determine the correlation between the docking results from both docking software and the experimental K_i values (obtained from the ChEMBL database). This can be achieved by plotting the lowest binding energies of all compounds obtained with ADT4 and VINA versus their experimental K_i values. Third party's software such as DataWarrior[25]) can be used (Figure 17).

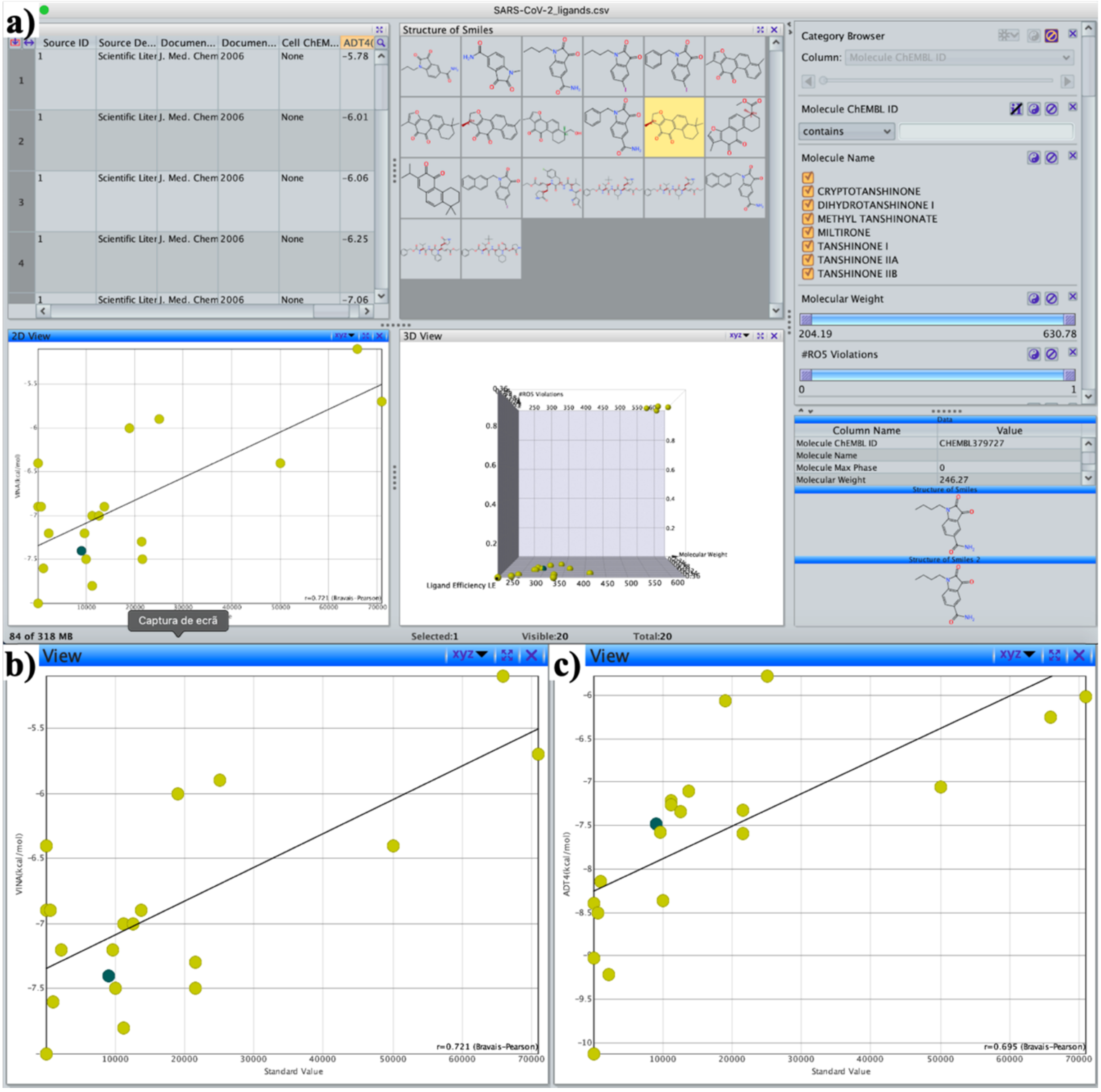
As can be seen in Figure 17b,c, good correlations were obtained between the docking scoring results and the experimental K_i values. VINA slightly outperforms ADT4 (0.72 vs 0.70), suggesting that VINA is as good, or even better, than ADT4 to perform this type of molecular docking screening calculations. Furthermore, VINA is significantly computationally efficient when compared to ADT4, which renders it the best choice to tackle more demanding screening campaigns on the quest for new SARS-CoV-2 M^{pro} inhibitors.

4.3. Retrieving Alternative Active Compounds from Alternative Databases Based on the Screening Results.

The use of docking screening campaigns is of utmost importance to identify from a database, compounds with high binding affinity toward a specific protein target. In this protocol, we have initially validated our screening protocol with a well-known inhibitor of the M^{pro} protein (compound 13b), and afterward applied it to identify from a small compound database, the ligands with higher binding affinities. However, we can even go further and try to expand our quest, to identify new promising alternative compounds from other open access or private in-house databases available to the reader. To perform such a demanding task, we can use the query capabilities of Datawarrior. This program, besides allowing the visualization and analysis of screening results (see previous section) can use previously identified hits (also referred to as templates), and search in multiple databases for new alternative compounds considering their chemical similarity, contained substructures, structure equivalence, or tautomers. In this protocol, we will not go into detail on how to perform such tasks; however, the reader can visit the help page of Datawarrior (https://openmolecules.org/help/databases.html) to get all the necessary information to run such queries. See Box 8 for critical parameters and troubleshooting for Protocol 4.

5. PROTOCOL 5: GROMACS—A TOOL FOR MD SIMULATION

This tutorial is expected to provide an overview of the workflow needed to perform MD simulations with the GROMACS software package.[26] It is aimed at undergrad students which are being introduced to this topic. Despite being a tutorial for beginners, it assumes that the student is acquainted with a linux command line and knows the basic principles of file manipulations. MD simulations are a very useful computational tool for sampling the conformational space of a given protein, protein−ligand, or protein−protein complex.[27,28] Atomistic



	Source ID	Source De...	Documen...	Documen...	Cell ChEM...	ADT4(
1	1	Scientific Liter	J. Med. Chem	2006	None	-5.78
2	1	Scientific Liter	J. Med. Chem	2006	None	-6.01
3	1	Scientific Liter	J. Med. Chem	2006	None	-6.06
4	1	Scientific Liter	J. Med. Chem	2006	None	-6.25
	1	Scientific Liter	J. Med. Chem	2006	None	-7.06

Category Browser
Column: Molecule ChEMBL ID

Molecule ChEMBL ID
contains

Molecule Name
☑
☑ CRYPTOTANSHINONE
☑ DIHYDROTANSHINONE I
☑ METHYL TANSHINONATE
☑ MILTIRONE
☑ TANSHINONE I
☑ TANSHINONE IIA
☑ TANSHINONE IIB

Molecular Weight
204.19 630.78

#RO5 Violations
0 1

Data

Column Name	Value
Molecule ChEMBL ID	CHEMBL379727
Molecule Name	
Molecule Max Phase	0
Molecular Weight	246.27

Structure of Smiles
Structure of Smiles 2

Figure 17. (a) DataWarrior interface. In panel b we can see the docking results obtained from VINA plotted against the K_i for each compound, while in panel c, the same correlation is plotted with the ADT4 results.

Box 8. Protocol 4 Critical Parameters and Troubleshooting

During the presented protocol, we performed the preparation, running, and analysis of the results using two different approaches: a graphic and a command line interface based fully on ADTools. Since this software is based on python's programming language, it is possible that the user experiences some problems directly linked to the python environment. Therefore, the first word of advice is to try to understand the logs obtained from each error. Since this package is widely used, it is highly likely that if you get a problem, someone already got it. Therefore, our first suggestion to face such problems is to do a web search for known solutions. At http://autodock.scripps.edu/faqs-help readers can find the answers for frequently asked questions, as well the links to the latest reference ADT4 and Vina manuals.

information on the interactions between molecules that are happening on a relatively fast time scale (nano- to microseconds) are at hand. The level of resolution combined with the time scales achieved with these methodologies is still unattainable with most experimental techniques.

A major disadvantage of MD techniques resides in the fact that the fidelity and the accuracy of the results are only as good as the model and approximations used. A correct choice of force field parameters combined with a thorough sampling of the system conformational space is pivotal to obtain meaningful results. Also, special care must be taken regarding the selection of the starting structure in MD simulations. An experimental structure is always preferred; however, it is commonly accepted to start from theoretical model structures for which there is a significantly high degree of confidence.

Please note that, in this protocol, no simulations per se are performed since these are very time-consuming. However, the input files and the results (previously simulated) are provided

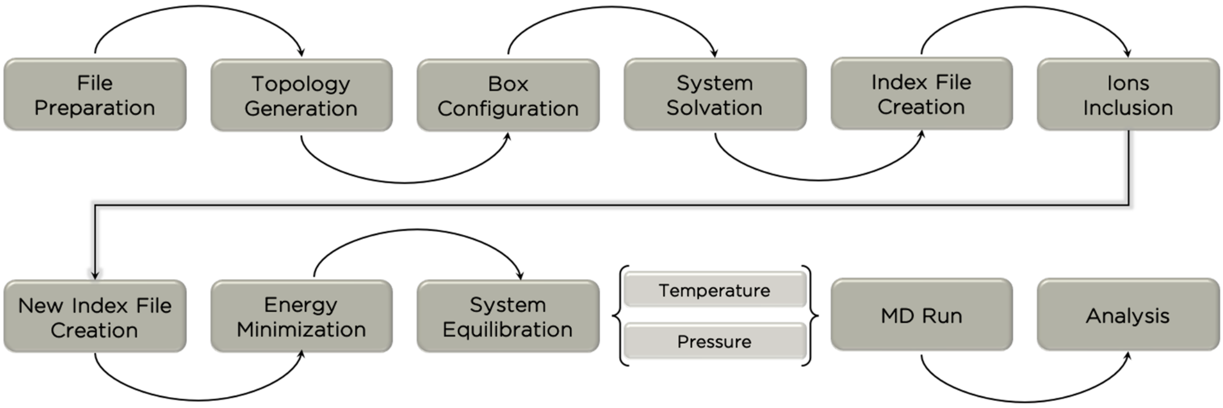


Figure 18. Workflow of the Protocol 5.

for download. Additionally, installation of the most recent GROMACS version is mandatory to replicate the results of all scripts. Please follow the workflow (Figure 18) and inspect the content of each running script. See Box 9 for necessary resources for Protocol 5.

Box 9. Protocol 5 Necessary Resources

Software: Terminal, GROMACS, PyMOL, gnuplot

Files: All input/output files and several scripts to perform the calculations.

Sample File: http://insilicotutorials.rd.ciencias.ulisboa.pt/md.zip, http://insilicotutorials.rd.ciencias.ulisboa.pt/md.html

5.1. Extract Files. Please download and decompress the md.zip file to a local folder. This will create several folders associated with each step of the MD simulations pipeline (Figure 18). In these, there are several script files, starting their names with numbers to help you run them in the correct order. There are many file types, namely,

.sh—these are the script files; you can look at them as small lists of instructions that will run like a normal program. We also provide additional information on the protocol as comment lines inside these scripts. Please open all these files in a text editor to obtain the most detailed information.

.pdb/.gro—these are the structure files, with the .gro files being a format variant of the .pdb.

.top/.itp—these are the topology files. In these files, we can store topological information about our system.

.mdp—parameter files for GROMACS, which have all the parameters needed for simulations.

.ndx—index files are needed to help GROMACS identify and distinguish atoms of the simulated system.

.log—log files created by GROMACS, which can be consulted for extra information.

.tpr/xtc/trr/edr—input/output binary files used by GROMACS.

.xvg—data files that are written by GROMACS and can be used to generate data plots.

.pml—PyMOL scripts previously created to help open and analyze the protein/ligand complex at the different steps of the protocol.

5.2. Build and Setup Our System. *5.2.1. PDB Cleanup and Input Preparation.* Enter the Preparation folder. Starting from the 6Y2G.pdb file, extract just the coordinates of the M^{pro} dimer using a simple *egrep "ATOM" 6Y2G.pdb > MPro.pdb* command. The two 13b ligands in the 6Y2G file should be ignored (for now). Their parameters required for the MD simulations will be taken from the Automated Topology Builder (ATB) and Repository (http://atb.uq.edu.au/). ATB provides GROMOS 54A7 topologies in GROMACS format for new ligands. One of those ligands was already submitted to the server and the ligand topology (ligand.itp) is available for download (http://atb.uq.edu.au/molecule.py?molid=479219). Before this file can be used, we converted the ATB custom atom types into the standard GROMOS 54A7 equivalent (see the process-files.sh script for details). This conversion loses the optimizations introduced by the ATB procedure, but allows the use of the ligands topology without adding new atom types to the standard GROMOS 54A7 force field. We also added an extra section at the end of the script to instruct GROMACS to apply position restraints also to the ligands, whenever it is called for. Addressing the specific parameters of this topology format (itp) is not in the scope of this tutorial, but a very thorough explanation of this subject is present in the GROMACS manual (http://manual.gromacs.org/documentation/2020/reference-manual/index.html). The two pdb files (one per ligand) were prepared and are available under the names: ligand{1,2}_unitedatom_original_geometry.pdb. Please note that these files need to be fully compatible with the atom ordering of the topology (ligand.itp) and the coordinates of the M^{pro} (no optimized geometries should be used). These files can be converted into the gro format using GROMACS editconf to be readily used.

5.2.2. Generation of a Topology (Top) from a Structure File (pdb). Enter the Setup folder. The first GROMACS module used is **pdb2gmx** which generates a topology (.top file) defining our molecule according to a force field (we chose GROMOS 54a7), including parameters about bonds, angles, dihedrals. The water model used was SPC. The input PDB file (Mpro.pdb) should only contain protein atoms. You can open and view the script run_pdb2gmx.sh using a text editor. The key command line executes the **pdb2gmx** module with several options. The "-ignh" will ignore the hydrogen atoms, if present in the pdb file, and the "-merge all" flag tells GROMACS to merge all chains into a single topology. With the -renum option the final topology and coordinate files will have their atoms/residues renumbered starting from 1 (default). The "-ter" argument allows choosing the termini ionization. The *pdb2gmx* module asks the user for the protonation/ionization states of the termini. There are 4 termini in the two chains of M^{pro} and we have chosen the ionized states (option 0) for all groups.

The addition of the two ligands to the system requires a little hacking to both the topology and the coordinate .gro file. We need the Cartesian coordinates (available in the original PDB files) of the two ligands in the GROMACS format (.gro) to be concatenated into the final system (MPro-ligs.gro). Also, we need to add the ligand topology (ligand.itp) being invoked inside the protein topology and, finally, the two ligand molecules to the molecules listing section at the end of the topology (MPro-ligs.top). All these steps of scripting can be found inside the run_pdb2gmx.sh script file.

5.2.3. Generating a System Box and Adding Solvent. Use the module **editconf** (box.sh) to generate the box surrounding our protein, which, as the name suggests, edits structure files. Among several box configurations, the rhombic dodecahedron is the most regular space-filling unit cell available in GROMACS. The "-c" option centers the solute in the box and "-d" specifies the distance between the solute and the box. We used a cutoff of 1.4 nm cutoff for nonbonded interactions, thus a 1.6 nm distance (0.8 + 0.8, distances to two opposite sides of the box) between them is usually sufficient. The *-resnr 1* allows a complete renumbering (starting from 1) of atoms/residues in the system.

Having defined our solute and the system box which contains it, we now proceed to add the solvent molecules (water in this case) using the **solvate** module. The -cs option requires a solvent configuration file to be used. We will use SPC water which is a simple and efficient three-point charge water model. The number of water molecules to be added are automatically calculated by the program to fill the empty volume in the box. Since we are adding new molecules to our system, the program also needs to update the topology in the "molecules" section, which is why we provide it to the program.

Note: a PyMOL script (pml) is available in the current folder to help evaluate the conformational evolution between steps.

5.2.4. Creating Index Files. The index file (created in make_index.sh) is required for assigning atoms to specific groups, such as Protein, Solvent, or simply SOL, which can include everything but the solute. The **make_ndx** GROMACS module starts from either a gro or pdb file as input and creates an index file with named entries (ex.: Protein, Solvent) thus designating the respective atoms to different groups. Since we will be mainly discussing the protease and the ligands, we will delete all the default groups that are not needed using "del 2−12" and "del 3" commands (note that the group list is updated upon each action), then we will rename the ligand group (13b) by selecting the second index "name 2 13b", the protease (MPro) group "name 1 Mpro", and the group with atoms from both MPro+13b is designated Protein with "name 4 Protein". This is just an intermediary index file to prepare the system for the next step, and the final more complete index will be created after adding counterions.

5.2.5. Adding Ions. After setting up our final system composed by M^{Pro}, the two ligands, and solvent, an imbalance between anionic and cationic residues was obtained. A surplus of four anionic residues per monomer results in a total charge of −8, which needs to be neutralized with the addition of ions (cations in this case). The **genion** GMX tool can be used (see add_ion.sh), but it requires a generic .tpr input file of our system, which needs to be created "a priori". The **grompp** module creates this .tpr file that comprises the atom coordinates, topology information, index, and simulation parameters. All files are available, and as well, the parameter file which can be created empty (all default values) is available. The flag "-maxwarn 10" increases the number of acceptable warnings before the program exits, which is not problematic in this case and should be ignored. The **genion** module adds ions to the system by changing a molecule from the chosen group (usually solvent) to an ion. The group is selected from the index file "-n index.ndx" and, in our case, we selected "SOL", which contains the water molecules. We need cationic residues; hence, we chose eight sodium atoms (Na+) to reach the overall system neutrality. The positive cations can be selected with flags "-pname NA", for name, and "-np 8", for the amount. Alternatively, the *genion* module has a flag (-neutral) that automatically adds enough counterions to obtain full system neutrality.

Finally, a new index file (make_new_index.sh) is now required where the solvent group should include the counterions just added. Also, there are a series of index groups that can be created (see the script) to help later on with the analysis process, namely, the individual ligands, each M^{pro} monomer, the α-carbons of each monomer, etc.

5.2.6. Energy Minimization. Enter the Minimization folder. Now, the system is solvated and electroneutral; however, before performing any MD simulation, we need to remove any high-energy interactions that may cause numerical instabilities or even the collapse of the simulation. Hence, we perform a two-step procedure (minimization.sh) to help the system reach a minimum of energy, ideally, approaching the global energy minimum of the system. The minimization steps use the same protocol and the same GROMACS modules and flags; however, the .mdp files (MD parameters) differ from each other. Therefore, before running this script, inspect them. The **grompp** module prepares a .tpr file with all the compiled simulation settings of our system, incorporating the information from the parameters (-f .mdp), the topology (-p .top), the coordinates (-c .gro), and the index (-n .ndx). It writes a complete output parameter file with the currently used .mdp parameters (-po .mdp), the ones supplied by the user, and the remaining default ones. It also writes a processed topology (-pp .top) with all topological information of our system, completely independent of the force field files. Finally, the "-maxwarn 1000" overrides the halt order when multiple warnings occur. The **mdrun** is the GROMACS module that runs simulations according to the parameters specified in the mdp file which, as we have discussed, has been compiled by **grompp** into the binary .tpr file.

In this minimization protocol, we have only used the *steepest descent* algorithm, but other algorithms and combinations could have been adopted. In our two steps, we only changed the constraints algorithm (LINCS) which is turned off in the first step to allow an increased plasticity in the system and turned on in the second step to bring all bonds to their minimum distances.

Note: a PyMOL script (pml) is available in the current folder to help evaluate the conformational evolution between steps.

5.2.7. Initialization of Temperature/Pressure. Enter the Initialization/folder. The initialization procedure (Initialization.sh) is an equilibration step that introduces temperature and pressure to a "frozen" still system. Be careful when first allowing movement in our system due to large interaction energies inside the box. Once the system heats up on the first segment of equilibration, the solvent/solute interactions are not optimal and thus the position of the complex atoms (M^{Pro}+13b ligands) must be restrained while water molecules accommodate themselves. The position restraints are specified in *posre.itp* (Setup) and the *ligand.itp* (Preparation) files. A restraining force

(1000 kJ mol^{-1} nm^{-2}) is assigned to the M^{pro} heavy atoms and all atoms of the ligand, penalizing their movement.

5.2.8. Heating up the System. The first step of the equilibration (init1) is done at a constant number of particles, volume, and temperature (**NVT ensemble**). This is defined in the init1.mdp file by the following parameters:

- Tcoupl = v-rescale; turning on the temperature coupling with the v-rescale method.
- ref_t = 310.0; setting the temperature to 310.0 K.
- Pcoupl = no; since pressure coupling is not being done, the volume of the box will be fixed.

Please note that the integrator flag is set to *md* and that *gen_vel = yes*, which means initial velocities will be randomly generated for the atoms, according to the temperature requested (*gen_temp = 310*).

5.2.9. Turning on the Pressure Coupling. After the temperature has been stabilized, the pressure coupling is turned on. The system will now sample from the **NPT ensemble** (constant number of particles, pressure, and temperature). This is the ensemble that most resembles experimental conditions which is why this is the most used ensemble in MD simulations of biomolecules. In init2.mdp, Pcoupl is now stating the pressure coupling method used (Parrinello-Rahman) and *gen_vel* has been turned off so that velocities are read from the trajectory resulting from the previous step.

A final step (init3) was created to equilibrate the integrator step. In the previous steps, an integrator step of 1 fs (0.001 ps) was used in order to obtain better stability in the simulations. However, since the MD simulations will be performed with a 2 fs step, this parameter must be adjusted, preferentially still under position restraints on the protein.

Note: a PyMOL script (pml) is available in the current folder to help evaluate the conformational evolution between steps.

5.3. Production MD. Enter the "MD" folder and check out the files already provided. After the initialization steps, the system is ready for production. An important feature of a production run is the fact that the system is completely unrestrained. In the MPro-ligs.mdp file, the total simulation is set as a function of two parameters:

- dt, the integrator time step, which we used 0.002 (ps).
- nsteps, the number of integration steps that defines the total simulation time. We chose 5 × 10^{7} steps, which can be multiplied by the 0.002-time step, leading to 10^{8} fs, or 100 ns.

Now the writing frequency of the system coordinates is defined to our .xtc file (nstxout-compressed). Coordinates were written every 100 ps, which should render 1000 frames at the end of the 100 ns simulation. In the command line to start the simulation, the -ntomp flag (number of threads, or CPU cores) can be substituted by other equivalent flags in order to perform the simulation using CPUs and GPUs. Running the simulation results in the generation of trajectories of position (xtc) and energies (edr), which can be analyzed with some detail. The final system configuration was also written (the gro file).

5.4. Result Interpretation. After performing the MD simulation, the complete output files will be available and ready for processing. In the Analyses folder, there are several subfolders, one per analysis, containing running scripts and the output files:

Trajectory. Inside the traj folder there is a script (extract_traj.sh) to process the trajectory (removes water molecules and ions) and center the protein in the simulation box. Since M^{Pro} and the 13b ligands consist of four separate molecules, we need to apply a custom procedure to correct the periodic boundary conditions (PBC). This consisted in giving a single atom located at the center of the complex (e.g., Cα of Tyr126) for the centering procedure. The final processed trajectory (traj_all.xtc) can be used in the following analysis. We also created a smaller PDB trajectory with only 100 frames which can be easily explored with PyMOL (open-traj-pymol.pml) and help to identify which regions of the complex are changing the most.

Root-Mean-Square-Deviation. Inside the rmsd folder, there is a script that calculates the RMSD of the full complex, the individual M^{pro} monomers and the individual 13b ligands. This property measures the deviation of the trajectory to the initial conformation. When simulating homodimers or any other symmetric complexes, we can calculate the level of symmetry of the different monomers. It is a simple protocol, where PyMOL is used to align one image to the other and report the best RMSD obtained. In the symmetry folder, we calculated the symmetry RMSD deviations between the M^{pro} monomers and between the ligands. RMSD values can be used to evaluate the stability and convergence of the MD simulations.

Radius of Gyration. Inside the gyration folder, there is a script that calculates the radius of gyration (R_g) of the full complex and M^{Pro} alone, without ligands. This property provides an estimation of the size of the protein and is often used to assess stability and the equilibration of the protein conformational space.

Secondary Structure. Inside the DSSP folder, there is a script that calculates and processes the secondary structure of each residue in each M^{Pro} monomer. The total helicity and β-content can be easily obtained by summing over the two monomers. This is another important property that provides data on the protein stability and convergence.

Interface Solvent Accessible Surface Area. Inside the sasa folder, there is a script that calculates the solvent-accessible surface area (SASA) of all molecules in the system under different partners. With this information, we can estimate the interfacial SASA of each 13b ligand over time. This is a very powerful property and can be designed and used to assess both convergence and stability of interfaces.

Hydrogen-Bond Number. Inside the H-bond folder we calculated the number of hydrogen bonds over time between the M^{pro} and each of the two ligands. This property provides an estimation of these polar contacts contribution to the stability of the complex.

Molecular Mechanics/Poisson–Boltzmann Surface Area (MM-PBSA). Inside the MM-PBSA folder, there is a script that calculates and processes the binding free energy over time of the complex between M^{pro} and each 13b ligand. The calculations were performed using an in-house implementation of the MM-PBSA method (PyBindE). This method calculates the binding free energy as the summation of four distinct energetic terms: two related to molecular mechanics in vacuum (Van der Waals and Coulombic); and two related to the solvation contribution to the binding energy. The polar solvation term is calculated using a Poisson–Boltzmann solver (DelPhi),[29] while the apolar term is calculated using a SASA-only model. This implementation calculates all partial contributions, but for the purposes of this tutorial, we only present the final binding free energies. The outputs are available for inspection and processing using PyMOL and Gnuplot (already processed PDFs are also provided). Figure 19 provides a graphical overview of the

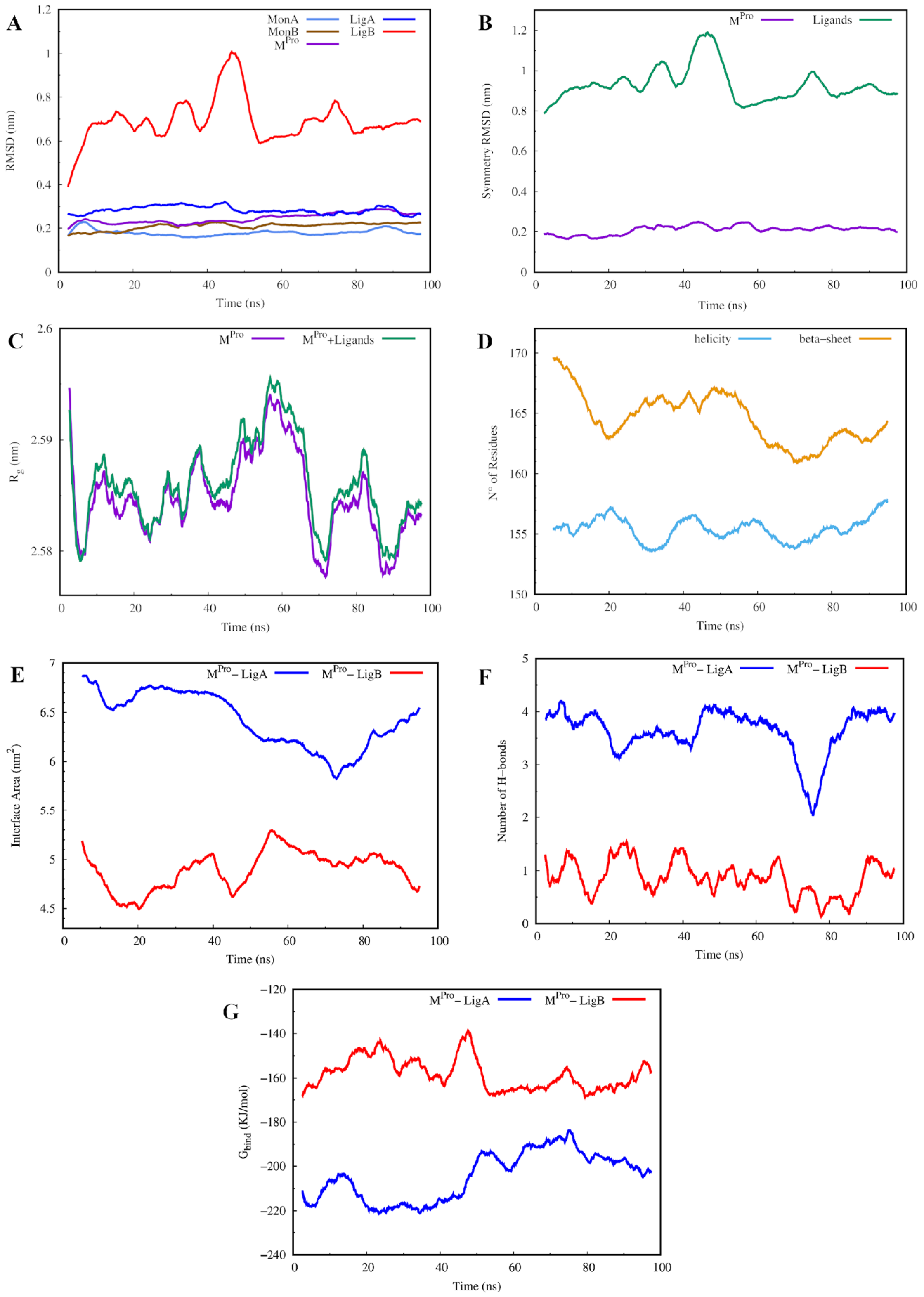


Figure 19. Graphical representation of the results obtained in this protocol. The data were obtained for the RMSD (A), symmetry RMSD (B), radius of gyration (C), secondary structure (D), interface area (E), H-bond number (F), and binding free energies (G).

results obtained. In the 100 ns simulation, the RMSD of M^{pro} is relatively low, indicating that the protein is not undergoing major conformational transitions (Figure 19A,B). This is also corroborated by the R_g values (Figure 19C) and the secondary structure variation (Figure 19D), which show a very stable protein homodimer. However, when focusing on the individual 13b ligands, there is an obvious difference in behavior, where Ligand A retains most features of the X-ray conformation with relatively low RMSD values (Figure 19A), a significantly high interface SASA with M^{pro} (Figure 19E), and a high H-bond content (Figure 19F); while Ligand B undergoes a major conformational transition (Figure 19A) deviating from Ligand A binding mode (Figure 19B) and slightly detaching from M^{pro} (Figure 19E,F). These results show that despite being a homodimer, the X-ray structure of M^{pro} is not fully symmetrical and the two binding modes are not completely equivalent. As a result, we observe two significantly different binding energy profiles (Figure 19G), where we estimate a binding energy penalty of ~40 kJ/mol for Ligand B compared to that for Ligand A. Longer MD simulations and additional replicate simulations would be required to better assess the convergence of both binding sites. From the X-ray structure, we suggest selecting chain A to perform any type of structural analyses, including molecular docking calculations. See Box 10 for critical parameters and troubleshooting for Protocol 5.

Box 10

Protocol 5 Critical Parameters and Troubleshooting

- Atom Y in residue XXX N was not found in rtp entry XXX while sorting atoms (Example: Atom HB3 in residue SER 1 was not found in rtp entry SER with 8 atoms). This error occurs when atom Y, typically a hydrogen, is present in the initial structure bound to residue XXX number N with a different nomenclature than the one used in the force field residue database parameters (.rtp) file. This leads to a mismatch between the information provided in the input and the force field. To solve this problem, normally the -ignh flag is used when running the pdb2gmx module. This flag informs GROMACS to ignore the hydrogen atoms in the input structure.
- No force field found. This error appears when pdb2gmx cannot find the specified force field. A possible explanation is that the GROMACS installation was moved to another folder. This can be fixed by reinstalling GROMACS or by moving the installation to its original folder. Another possible explanation is that a custom force field was being used in the working directory and is now missing.
- Residue 'XXX' not found in RTP. custom molecule XXX, such as 13b, is either not present as an entry in the force field residue database parameters (.rtp) file, incorrectly named in the .rtp file or in the "include" statement. The force field is built upon the information about atom types, bonds, angles, and bonded and nonbonded interactions for both the proteins and the ligands. Either this information is added to the force field or, as done in the tutorial, be called upon as an "include" statement inside the topology file (e.g., include "../1_Preparation/ligand.itp"). In this case, the ligand structure should not be submitted to pdb2gmx, but rather added manually after this module.
- System has nonzero charge. This requires more counter-ions to neutralize the system box. You can use the "genion" module to add the necessary positive/negative counterions. A fully neutral system is required when using Particle Mesh Ewald (PME) as the long-range electrostatics method. If Reaction Field theory were used, the system could remain charged.
- The cutoff length is longer than half the shortest box vector. This error appears when the cutoff length for nonbonded interactions leads to atoms interacting with their periodic image (when using periodic boundary conditions), which is unrealistic. To solve this issue, either increase the box size (running the "editconf" module) or decrease the cutoff length (beware that the cutoff length may be optimal for a given force field).
- Number of coordinates in coordinate file does not match topology. In this issue, the total number of atoms described in the topology does not match the atoms in the given gro or pdb file. Check if your topology file is updated after adding solvent or counterions or if there is a number mismatch in the "molecules" entry in your topology. Sometimes, the number difference reported in the error is a good hint to the source of the mismatch.
- LINCS/SETTLE/SHAKE warning. When performing runs using "mdrun", the simulation might halt due to warnings from the constraint's algorithm. Owing to an unstable system, the constraint algorithm is unable to adjust the bonds and angles, leading to the system "blowing up". Check your trajectory and pdb files to see if there are high energy interactions occurring that might destabilize the system.

6. CONCLUSIONS

Through the combination of five computer-based protocols, this tutorial shed light on the key interactions of SARS-CoV-2 M^{pro} with α-ketoamide 13b inhibitor, also providing a SBVS strategy to identify new potential SARS-CoV-2 M^{pro} inhibitor candidates. Therefore, this tutorial constitutes an important approach which can be useful to identify and full characterize the interactions between different drug candidates and several targets with an emergent role in several impacting diseases, such as COVID-19, in a less expensive and easier way, accessible to all scientific community.

■ DECLARATIONS

Availability of Data and Requirements. *Sample Files:* All files and data related to each Protocol are available at http://insilicotutorials.rd.ciencias.ulisboa.pt/.

Support Protocols. Protocol 1 (PyMOL Installation). Although PyMOL is not free to develop, maintain, and support, there is an Open-Source PyMOL version that includes all functionalities described in Basic Tutorial 1. Open-Source PyMOL can be freely downloaded from https://github.com/schrodinger/pymol-open-source. The installation tips in different distributions (Windows, Linux, MacOSX) can be found at https://pymol.org/2/support.html?#installation.

Protocol 2 (MODELER Installation). MODELER can be freely downloaded and installed from https://salilab.org/modeller/download_installation.html. You need to register to access the password for download. We would advise the installation by conda if possible:

```
conda config --add channels salilab
conda install modeler
```

Make sure you add the correct license to the file at your personal path: */yourpath/modeler-9.25/modlib/modeler/config.py*.

Readers can obtain further information about MODELER installation in the paper published by Webb and Sali.[7]

Protocol 4 (ADT4 and VINA). All calculations in the SBVS campaign can be performed by running the provided bash scripts in the provided zip file. These scripts execute all the necessary steps of the protocols within a Linux environment using both graphical and command line interfaces. The scripts can be easily edited and adapted to run in other OS systems. In any case, the protocols require the correct installation of the following software/programs:

.**Text editor**. Depending on the OS, one can use different flavors of text editors to edit the provided files. The authors suggest the use of Emacs, which can be installed in all OS distributions and fulfill all the necessary tasks.

. **AutoDock 4.2.6** and **AutoDock Vina** can be freely downloaded from https://autodock.scripps.edu/downloads/. Windows, Linux, MacOSX, and source files are available.

. **AutoDockTools (ADTools)** can be downloaded from https://ccsb.scripps.edu/mgltools/. ADTools will be used for the visual setup, run, and analysis of all the calculations of ADT4 and VINA, both via its graphical and command line approaches.

. **DataWarrior** can be downloaded from https://openmolecules.org/datawarrior/. DataWarrior is a software traditionally used by MedChem scientists which allows the calculation and visualization of physicochemical properties and structure activity relationships.

. **Unicon** can be downloaded from https://www.zbh.uni-hamburg.de/forschung/amd/software/unicon.html. This command line software is able to generate compound conformers and perform file conversion of the different compounds in a screening database.

. **SplitSDFiles.pl from MayaChemTools**: This pearl script is part of the MayaChemTools package (that can be downloaded from http://www.mayachemtools.org/), and that allows, among other functions, to split a single .sdf file into multiple individual files.

OpenBabel can be downloaded from http://openbabel.org/wiki/Main_Page. This software is able to interconvert different file formats and can additionally be used to prepare protein and ligand files to the docking screening simulations.

Protocol 5 (GROMACS Installation). The GROMACS software package can be easily installed in Ubuntu and other Debian based operating systems by running simply: *apt install gromacs* (as sudo/root). For other linux distros, there are similar recipes. For Windows users, the installation of the Linux Bash Shell on Windows 10 is highly recommended. To install a specific GROMACS version, please refer to the "Quick and dirty installation" section of the GROMACS online manual at https://manual.gromacs.org/documentation/2020/install-guide/index.html.

Internet Resources: Protocol 1.

- https://github.com/schrodinger/pymol-open-source: webpage on GitHub to download the Open-Source PyMOL version.
- https://pymol.org/2/support.html?#installation: webpage with PyMOL installation types required for different distributions (Windows, Linux, MacOSX).
- https://sourceforge.net/projects/pymol/lists/pymol-users: PyMOL Users Mailing List webpage, which includes the exchange of ideas, tips, and information with other knowledgeable users, as well as the update on the most recent PyMOL news.
- https://pymolwiki.org/index.php/Main_Page: PyMOL-Wiki webpage, which is a user knowledge database and could also guide the reader into tutorials, plugins or answers for questions often asked.

Protocol 2.

- https://www.rcsb.org/: Protein Data Bank webpage.
- https://salilab.org/modeller/download_installation.html: webpage that includes the files required for downloading and installation MODELER software for different distributions (Windows, Linux, MacOSX).
- https://www.uniprot.org/uniprot/P0DTD1: UniProt webpage to download the sequence of the M^{pro} (3C-like proteinase).

Protocol 3.

- https://www.bonvinlab.org/education/: webpage with several valuable resources such as online lectures for a more detailed use of HADDOCK web server.
- https://ambermd.org/tutorials/pengfei/: webpage with LEaP program tutorials.

Protocol 4.

- http://autodock.scripps.edu/downloads: webpage with the required source files for downloading and installing ADT4 and VINA software for different distributions (Windows, Linux, MacOSX).
- http://mgltools.scripps.edu/downloads: webpage with the required source files for downloading and installing ADT4 and VINA software for different distributions (Windows, Linux, MacOSX).
- http://www.openmolecules.org/datawarrior/: webpage with the required source files for downloading and installing DataWarrior software for different distributions (Windows, Linux, MacOSX).
- https://www.zbh.uni-hamburg.de/forschung/amd/software/unicon.html: webpage with the required source files for downloading and installing UNICON command-line tool for different distributions (Windows, Linux, MacOSX).
- http://www.mayachemtools.org: webpage with the required source files for downloading the Perl script SplitSDFiles.pl from MayaChemTools.
- http://autodock.scripps.edu/faqs-help: webpage with the answers for the most frequently asked questions, as well as the links to latest reference ADT4 and Vina manuals.
- http://openbabel.org/wiki/Main_Page: webpage with the required source files for downloading and installation of Openbabel software (Windows, Linux, MacOSX).

Protocol 5.

- http://manual.gromacs.org/documentation/2020/reference-manual/index.html: GROMACS online reference manual.
- http://atb.uq.edu.au/: Automated Topology Builder (ATB) and Repository.
- https://github.com/mms-fcul/mmpbsa: MM-PBSA code (PyBindE). Please install *git* in the command line and follow the PyBindE dependencies (see README file).

■ AUTHOR INFORMATION

Corresponding Authors

Miguel Machuqueiro − *BioISI—Biosystems & Integrative Sciences Institute, Faculty of Sciences, University of Lisboa,*

Lisboa 1749-016, Portugal; orcid.org/0000-0001-6923-8744; Email: machuque@ciencias.ulisboa.pt

Irina S. Moreira − *University of Coimbra, Center for Neurosciences and Cell Biology, Department of Life Sciences, Coimbra 3000-456, Portugal;* orcid.org/0000-0003-2970-5250; Email: irina.moreira@cnc.uc.pt

Authors

Nícia Rosário-Ferreira − *Coimbra Chemistry Center, Chemistry Department, Faculty of Science and Technology, University of Coimbra, Coimbra 3004-535, Portugal; CNC—Center for Neuroscience and Cell Biology, University of Coimbra, Cantanhede 3060-197, Portugal;* orcid.org/0000-0002-7225-9287

Salete J. Baptista − *CNC—Center for Neuroscience and Cell Biology, University of Coimbra, Cantanhede 3060-197, Portugal; Centro de Ciências e Tecnologias Nucleares, Instituto Superior Técnico, Universidade de Lisboa, Bobadela 2695-066, Portugal*

Carlos A. V. Barreto − *CNC—Center for Neuroscience and Cell Biology, University of Coimbra, Cantanhede 3060-197, Portugal; PhD Programme in Experimental Biology and Biomedicine, Institute for Interdisciplinary Research (IIIUC), University of Coimbra, Coimbra 3000-456, Portugal;* orcid.org/0000-0003-1459-7680

Filipe E. P. Rodrigues − *BioISI—Biosystems & Integrative Sciences Institute, Faculty of Sciences, University of Lisboa, Lisboa 1749-016, Portugal*

Tomás F. D. Silva − *BioISI—Biosystems & Integrative Sciences Institute, Faculty of Sciences, University of Lisboa, Lisboa 1749-016, Portugal;* orcid.org/0000-0003-4608-2673

Sara G. F. Ferreira − *BioISI—Biosystems & Integrative Sciences Institute, Faculty of Sciences, University of Lisboa, Lisboa 1749-016, Portugal*

João N. M. Vitorino − *BioISI—Biosystems & Integrative Sciences Institute, Faculty of Sciences, University of Lisboa, Lisboa 1749-016, Portugal*

Rita Melo − *CNC—Center for Neuroscience and Cell Biology, University of Coimbra, Cantanhede 3060-197, Portugal; Centro de Ciências e Tecnologias Nucleares, Instituto Superior Técnico, Universidade de Lisboa, Bobadela 2695-066, Portugal;* orcid.org/0000-0003-1056-1007

Bruno L. Victor − *BioISI—Biosystems & Integrative Sciences Institute, Faculty of Sciences, University of Lisboa, Lisboa 1749-016, Portugal*

Complete contact information is available at:
https://pubs.acs.org/10.1021/acssynbio.1c00368

Author Contributions

[#]N.R.-F. and S.J.B. are co-first authors. Conceptualization, resources, and funding, I.S.M. and M.M.; methodology, software, validation, formal analysis, data curation and visualization, all authors; writing-original draft preparation, N.R.F., S.J.B., C.A.V.B., F.E.P.R., T.F.D.S., S.G.F.F., R.M., B.L.V.; writing-review and editing, N.R.F., S.J.B., I.S.M., M.M., supervision and project administration, I.S.M., M.M. All authors have read and agreed to the published version of the manuscript.

Funding

This work was funded by the COMPETE 2020—Operational Programme for Competitiveness and Internationalisation and Portuguese national funds via FCT—Fundação para a Ciência e a Tecnologia, under projects LA/P/0058/2020, UIDB/04539/2020, UIDP/04539/2020, POCI-01-0145-FEDER-031356, PTDC/QUI-OUT/32243/2017, DSAIPA/DS/0118/2020, and PTDC/BIA-BFS/28419/2017. N.R.-F., C.B., and T.S. acknowledge support from FCT, under the doctoral fellowships PD/BD/135179/2017, SFRH/BD/145457/2019, and SFRH/BD/140886/2018, respectively. M.M. also acknowledges support from FCT-CEECIND/02300/2017.

Notes

The authors declare no competing financial interest.

ACKNOWLEDGMENTS

The authors would like also to acknowledge COST Action CA17104: STRATAGEM—New diagnostic and therapeutic tools against multidrug-resistant tumors.

ABBREVIATIONS

3D	Three-Dimensional
ADT4	AutoDockTools 4
ADTools	AutoDockTools
AI	Artificial Intelligence
AIRs	Ambiguous Interaction Restraints
ATB	Automated Topology Builder
COVID-19	COronaVIrus Disease 2019
DOPE	Discrete Optimized Protein Energy
Eelec	ELECtrostatic intermolecular Energy
Evdw	Van Der Waals intermolecular Energy
FCC	Fraction of Common Contacts
GROMACS	GROningen MAchine for Chemical Simulations
GUI	Graphical User Interface
HADDOCK	High Ambiguity Driven protein−protein DOCKing
K_i	Inhibitory constant
MD	Molecular Dynamics
MERS-CoV	Middle East Respiratory Syndrome Corona-Virus
Mpro	Main protease
NMR	Nuclear Magnetic Resonance
PDB	Protein DataBank
PME	Particle Mesh Ewald
RMSD	Root-Mean Square Deviation
SARS-CoV-1	Severe Acute Respiratory Syndrome CoronaVirus 1
SARS-CoV-2	Severe Acute Respiratory Syndrome CoronaVirus 2
SASA	Solvent-Accessible Surface Area
SBVS	Structure-Based Virtual Screening
SBVS	Structure-Based Virtual Screening
TAD	Torsion Angle molecular Dynamics
UNIPROT	Universal Protein Resource
VINA	AutoDock VINA
α-ketoamide 13b	tert-butyl (1-((S)-1-(((S)-4-(benzylamino)-3,4-dioxo-1-((S)-2-oxopyrrolidin-3-yl)-butan-2-yl)amino)-3-cyclopropyl-1-oxopropan-2-yl)-2-oxo-1,2-dihydropyridin-3-yl)-carbamate

REFERENCES

(1) Olubiyi, O. O.; Olagunju, M; Keutmann, M; Loschwitz, J; Strodel, B. High Throughput Virtual Screening to Discover Inhibitors of the Main Protease of the Coronavirus SARS-CoV-2. *Molecules* **2020**, *25* (14), 3193 DOI: 10.3390/molecules25143193.

(2) Jiménez-Alberto, A; Ribas-Aparicio, R. M.; Aparicio-Ozores, G; Castelán-Vega, J. A. Virtual screening of approved drugs as potential

SARS-CoV-2 main protease inhibitors. *Comput. Biol. Chem.* **2020**, *88*, 107325.

(3) Zhang, L; Lin, D; Sun, X; Curth, U; Drosten, C; Sauerhering, L; et al. Crystal structure of SARS-CoV-2 main protease provides a basis for design of improved α-ketoamide inhibitors. *Science* **2020**, *368* (6489), 409−12.

(4) Fischer, A; Sellner, M; Neranjan, S; Smieško, M; Lill, M. A. Potential Inhibitors for Novel Coronavirus Protease Identified by Virtual Screening of 606 Million Compounds. *Int. J. Mol. Sci.* **2020**, *21* (10), 3626.

(5) Banerjee, R; Perera, L; Tillekeratne, V. Potential SARS-CoV-2 main protease inhibitors. *Drug Discovery Today* **2021**, *26* (3), 804−16.

(6) *The PyMOL Molecular Graphics System*, ver. 2.4; Schrödinger, LLC, 2020. https://pymol.org/2/.

(7) Webb, B; Sali, A. Comparative Protein Structure Modeling Using MODELLER. *Curr Protoc Bioinformatics.* **2016**, *54*, 5.6.1−5.6.37.

(8) Gouy, M; Guindon, S; Gascuel, O. SeaView Version 4: A Multiplatform Graphical User Interface for Sequence Alignment and Phylogenetic Tree Building. *Mol. Biol. Evol.* **2010**, *27* (2), 221−4.

(9) Wiederstein, M; Sippl, M. J. ProSA-web: interactive web service for the recognition of errors in three-dimensional structures of proteins. *Nucleic Acids Res.* **2007**, *35*, W407−10.

(10) Cristobal, S; Zemla, A; Fischer, D; Rychlewski, L; Elofsson, A. A study of quality measures for protein threading models. *BMC Bioinf.* **2001**, *2*, 5.

(11) van Zundert, G.C.P.; Rodrigues, J.P.G.L.M.; Trellet, M.; Schmitz, C.; Kastritis, P.L.; Karaca, E.; Melquiond, A.S.J.; van Dijk, M.; de Vries, S.J.; Bonvin, A.M.J.J.; et al. The HADDOCK2.2 Web Server: User-Friendly Integrative Modeling of Biomolecular Complexes. *J. Mol. Biol.* **2016**, *428* (4), 720−5.

(12) Lensink, M. F.; Brysbaert, G; Nadzirin, N; Velankar, S; Chaleil, R. A. G.; Gerguri, T; et al. Blind prediction of homo- and hetero-protein complexes: The CASP13-CAPRI experiment. *Proteins: Struct., Funct., Genet.* **2019**, *87* (12), 1200−21.

(13) Graziani, D; Caligari, S; Callegari, E; De Toma, C; Longhi, M; Frigerio, F; et al. Evaluation of Amides, Carbamates, Sulfonamides, and Ureas of 4-Prop-2-ynylidenecycloalkylamine as Potent, Selective, and Bioavailable Negative Allosteric Modulators of Metabotropic Glutamate Receptor 5. *J. Med. Chem.* **2019**, *62* (3), 1246−73.

(14) Sethi, A; Joshi, K; Sasikala, K; Alvala, M. Molecular Docking in Modern Drug Discovery: Principles and Recent Applications [Internet]. *Drug Discovery and Development - New Advances*; Intech, 2020. DOI: 10.5772/intechopen.85991.

(15) Zhang, W; Hu, Z. Faculty Opinions recommendation of Crystal structure of SARS-CoV-2 main protease provides a basis for design of improved α-ketoamide inhibitors. *Faculty Opinions*, 2020. https://facultyopinions.com/prime/737592020.

(16) de Vries, S. J.; Bonvin, A. M. J. J. CPORT: a consensus interface predictor and its performance in prediction-driven docking with HADDOCK. *PLoS One* **2011**, *6* (3), No. e17695.

(17) Preto, A. J.; Moreira, I. S. SPOTONE: Hot Spots on Protein Complexes with Extremely Randomized Trees via Sequence-Only Features. *Int. J. Mol. Sci.* **2020**, *21* (19), 7281.

(18) Rodrigues, J. P. G. L. M.; Trellet, M.; Schmitz, C.; Kastritis, P.; Karaca, E.; Melquiond, A. S. J.; Bonvin, A. M. J. J.; et al. Clustering biomolecular complexes by residue contacts similarity. *Proteins: Struct., Funct., Genet.* **2012**, *80* (7), 1810−7.

(19) Maia, E. H. B.; Assis, L. C.; de Oliveira, T. A.; da Silva, A. M.; Taranto, A. G. Structure-Based Virtual Screening: From Classical to Artificial Intelligence. *Front. Chem.* **2020**, *8*, 343.

(20) Morris, G. M.; Huey, R; Lindstrom, W; Sanner, M. F.; Belew, R. K.; Goodsell, D. S.; et al. AutoDock4 and AutoDockTools4: Automated docking with selective receptor flexibility. *J. Comput. Chem.* **2009**, *30* (16), 2785−91.

(21) Trott, O; Olson, A. J. AutoDock Vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. *J. Comput. Chem.* **2009**, *31* (2), 455−61.

(22) Forli, S; Huey, R; Pique, M. E.; Sanner, M. F.; Goodsell, D. S.; Olson, A. J. Computational protein-ligand docking and virtual drug screening with the AutoDock suite. *Nat. Protoc.* **2016**, *11* (5), 905−19.

(23) Wang, Z; Sun, H; Shen, C; Hu, X; Gao, J; Li, D; et al. Combined strategies in structure-based virtual screening. *Phys. Chem. Chem. Phys.* **2020**, *22* (6), 3149−59.

(24) O'Boyle, N. M.; Banck, M; James, C. A.; Morley, C; Vandermeersch, T; Hutchison, G. R. Open Babel: An open chemical toolbox. *J Cheminform.* **2011**, *3* (1), 1−14.

(25) Sander, T; Freyss, J; von Korff, M; Rufener, C. DataWarrior: an open-source program for chemistry aware data visualization and analysis. *J. Chem. Inf. Model.* **2015**, *55* (2), 460−73.

(26) Abraham, M. J.; Murtola, T; Schulz, R; Páll, S; Smith, J. C.; Hess, B; et al. GROMACS: High performance molecular simulations through multi-level parallelism from laptops to supercomputers. *SoftwareX* **2015**, *1−2*, 19−25.

(27) Rahman, M. M.; Saha, T; Islam, K. J.; Suman, R. H.; Biswas, S; Rahat, E. U.; et al. Virtual screening, molecular dynamics and structure-activity relationship studies to identify potent approved drugs for Covid-19 treatment. *J. Biomol. Struct. Dyn.* **2021**, *39*, 1−11.

(28) Hollingsworth, S. A.; Dror, R. O. Molecular Dynamics Simulation for All. *Neuron* **2018**, *99* (6), 1129−43.

(29) Li, L; Li, C; Sarkar, S; Zhang, J; Witham, S; Zhang, Z; et al. DelPhi: a comprehensive suite for DelPhi software and associated resources. *BMC Biophys.* **2012**, *5* (1), 1−11.