JCIM JOURNAL OF CHEMICAL INFORMATION AND MODELING

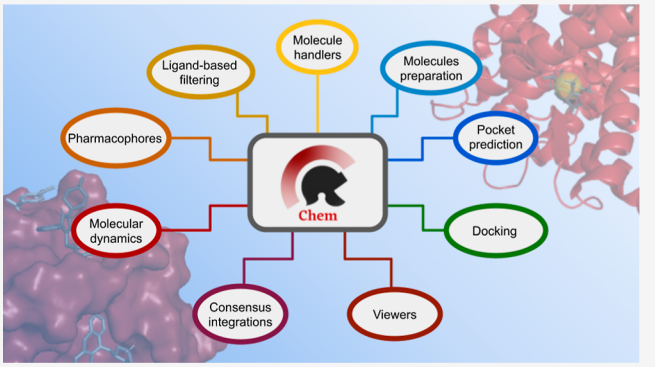
pubs.acs.org/jcim
Article

Scipion-Chem: An Open Platform for Virtual Drug Screening

Daniel Del Hoyo,* Martin Salinas, Alba Lomas, Eugenia Ulzurrun, Nuria E. Campillo, and Carlos Oscar Sorzano

Cite This: J. Chem. Inf. Model. 2023, 63, 7873−7885

ABSTRACT: Virtual drug screening (VDS) tackles the problem of drug discovery by computationally reducing the number of potential pharmacological molecules that need to be tested experimentally to find a new drug. To do so, several approaches have been developed through the years, typically focusing on either the physicochemical characteristics of the receptor structure (structure-based virtual screening) or those of the potential ligands (ligand-based virtual screening). Scipion is a workflow engine well suited for structural studies of biological macromolecules. Here, we present Scipion-chem, a new branch oriented to VDS. A total of 11 plugins have already been integrated from the most common programs used in the field. They can be used through the Scipion graphical user interface to execute and analyze typical VDS tasks. In addition, we have developed several consensus protocols that combine results from the different integrated programs to generate more robust predictions. Backstage, Scipion also facilitates the interoperability of those different software packages while tracking all of the intermediate files, parameters, and user decisions. In summary, in this article, we present Scipion-chem. This accessible, interoperable, and traceable platform provides the user with all of the tools to carry out a successful VDS workflow. Scipion-chem is openly available at https://github.com/scipion-chem.



1. INTRODUCTION

Drug discovery and development is a challenging and multistage process that involves compound identification, in vitro evaluation, hit optimization, in vivo and preclinical studies, and clinical phases, demanding substantial time and resources. Even after a promising compound is selected, several years are needed to pass the different clinical stages, and most of them are actually discarded in this process. Due to the difficulties of this process, the cost of developing a new drug is estimated to be around 2.6 billion dollars.[1] Apart from being a lengthy, costly, and complex process, it is crucial to acknowledge the vastness of the chemical space involved in drug development. This chemical space of drug-like compounds, which is estimated to consist of approximately 10^{63} molecules, remains considerably beyond our reach. Therefore, computer-aided drug discovery tools have been developed to speed up this process by filtering out nonpromising molecules based on either the molecule's characteristics or the unlikeliness of its interaction with the target of interest. Now, exploring this gigantic chemical space is still very challenging even using this computational method, but researchers keep developing new approaches and methods to improve the accuracy and speed of drug discovery tasks.

This in silico process of molecule filtering is typically referred to as virtual drug screening (VDS). It comprises different approaches, which can usually be classified as structure based (SB) or ligand based (LB) depending on the focus of the methods: either the receptor structure characteristics in SB, which is typically a protein, or the small molecules characteristics in LB.[2−4] Inside these two fields, we can find different commonly used approaches, such as molecular docking and de novo design in SB; or quantitative structure−activity relationship and pharmacophore modeling in LB, among many others.[5−8]

Throughout these last years, researchers and developers worldwide have been adding new tools and methods to approach these different tasks. Some of these programs can cover more than one step in a typical VDS workflow but usually not all of them. Among this software, we could cite Schrödinger,[9] AutoDock,[10] LePhar,[11] Fpocket,[12] and P2Rank.[13] Even though this continuous improvement helps the community to gradually solve the problem of drug discovery, for a researcher to use several different and

Received: July 27, 2023
Revised: November 7, 2023
Accepted: November 10, 2023
Published: December 5, 2023

https://doi.org/10.1021/acs.jcim.3c01085
J. Chem. Inf. Model. 2023, 63, 7873−7885

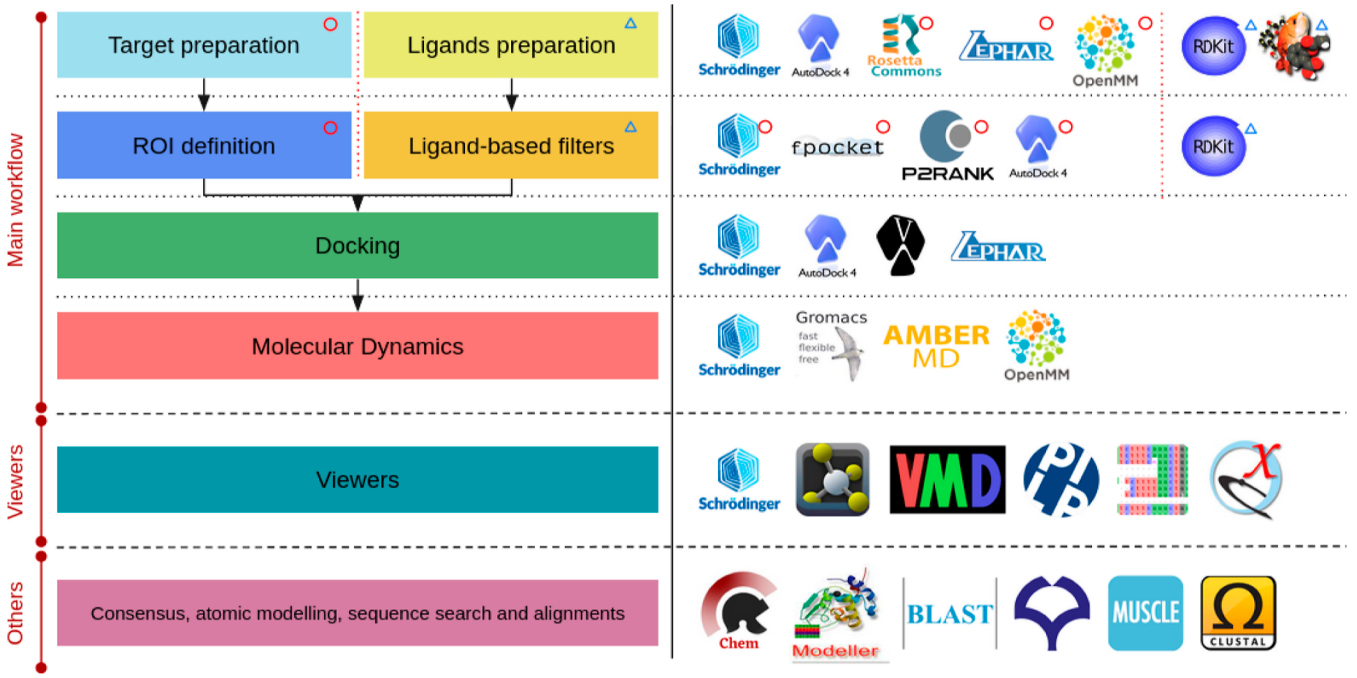


Figure 1. Scipion-chem currently integrated software, organized by the step of the typical VDS workflow in which they are involved in. Those programs that can only perform one of the tasks in the row are marked with red circles (only for target operations) and blue triangles (only for ligand operations). The viewers and other utilities available from Scipion-chem are also shown.

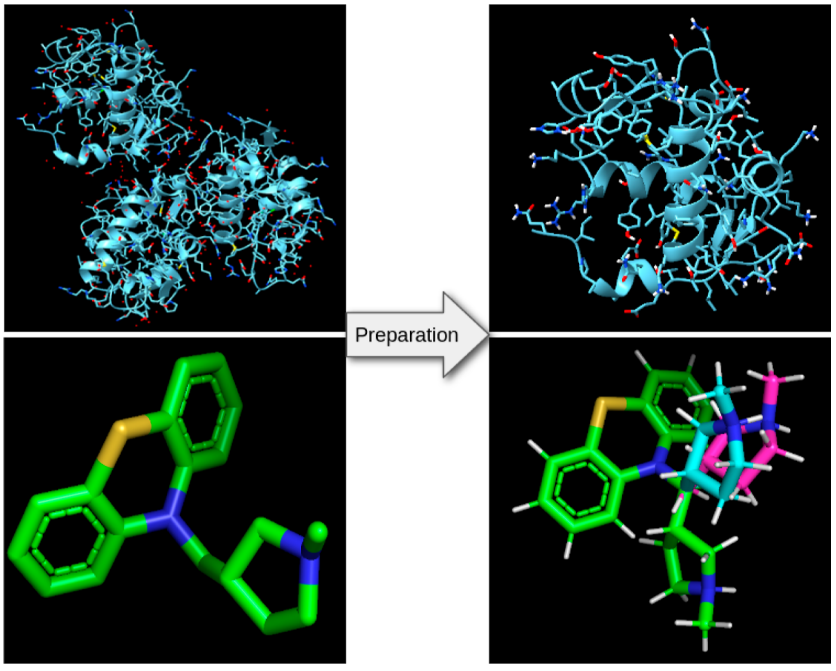


Figure 2. Example of Scipion-chem preparations for a target (PDB: 4ERF) and a ligand (ZINC480). The target is cleaned from water molecules and preexisting ligands, hydrogens are added, and only one of the three chains is generated. Hydrogens are added to the ligand, and three different conformers are generated.

independent programs usually involves a series of difficulties that hamper the development of complete VDS workflows, including

1. Installation: Researchers may encounter difficulties trying to install software due to the requirements or incompatibilities each of them has.
2. File formats: Although several common formats have been established for protein and small molecule structures, some programs require a specific format, which may not coincide with the one provided by the previous step. The conversion between these formats is possible, but sometimes, it is tricky and imperfect and requires external software.
3. Command line: Many programs rely on extensive command line use. Even though this can offer high flexibility, it also implies a strong entry barrier for many

https://doi.org/10.1021/acs.jcim.3c01085
J. Chem. Inf. Model. 2023, 63, 7873−7885

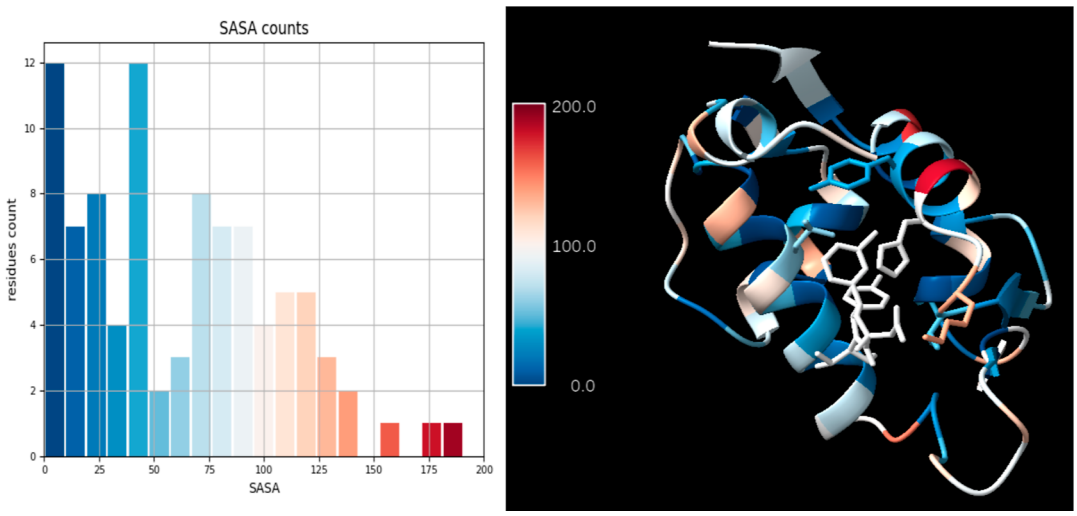


Figure 3. Example of Scipion-chem per-residue SASA calculation over a target (PDB: 4ERF-A). Histogram with the number of residues vs the SASA values is shown in the left panel. On the right panel, the structure of the complex is visualized in ChimeraX, and each residue is colored by its SASA value.

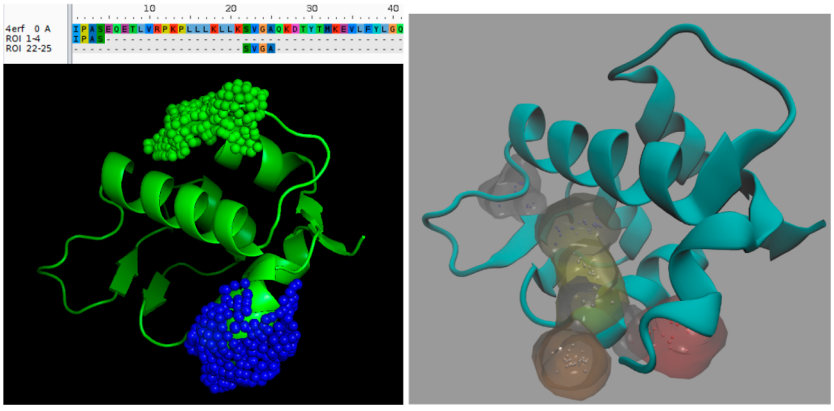


Figure 4. Example of Scipion-chem ROIs over a target (PDB: 4ERF-A). On the top left panel, two manually generated sequence ROIs are represented in AliView. On the left bottom panel, the two corresponding structural ROIs from mapping those sequence ROIs to the structure are represented as points over the surface in PyMol. Finally, on the right panel, the FPocket predicted pockets are represented as alpha spheres in VMD for the same structure.

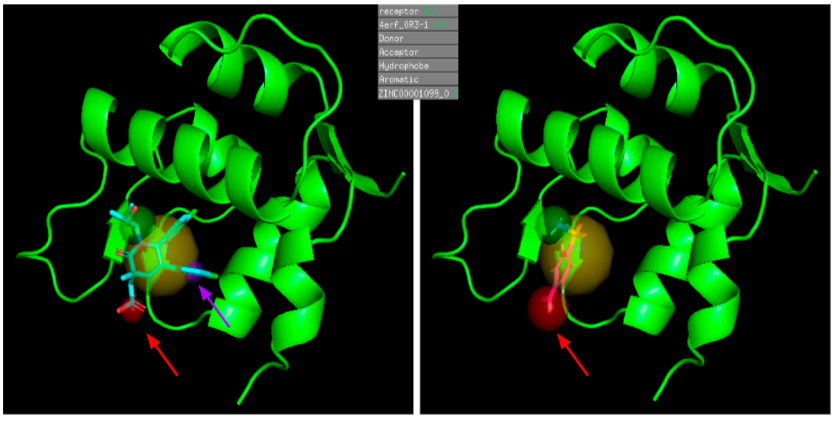


Figure 5. Example of Scipion-chem pharmacophore over a target (4ERF-A). On the left panel, actual ligand OR3 is used to generate a pharmacophore. In an intermediate step, the pharmacophore is slightly modified to make the hydrogen acceptor feature (red sphere) larger and to remove the aromatic feature (purple sphere). Then, on the right panel, molecule ZINC1099, which passed this pharmacophore filter, is plotted aligned to the pharmacophore.

users who need to learn and remember different commands for each program.

4. Traceability: VDS workflows can be composed of a considerable number of steps. When these steps are run

https://doi.org/10.1021/acs.jcim.3c01085
J. Chem. Inf. Model. 2023, 63, 7873−7885

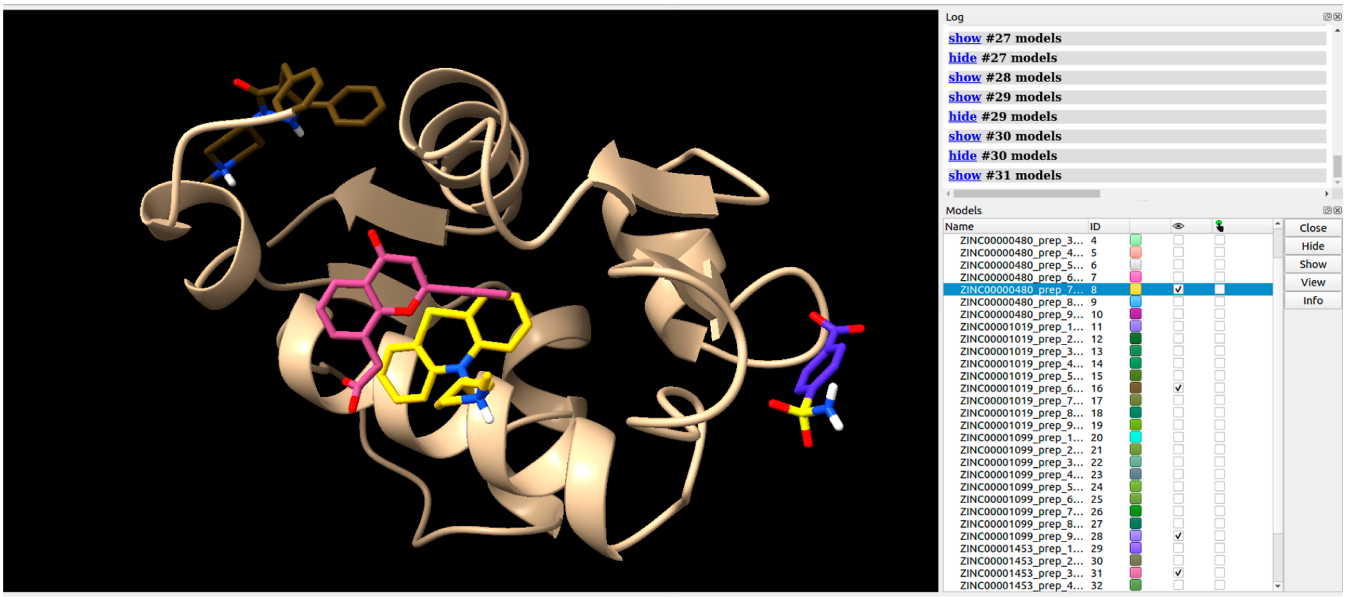


Figure 6. Example of Scipion-chem docking results in ChimeraX. Different small molecules appear to be docked onto the 4ERF-A protein. Each of these molecules stores the different metadata that the docking program provides, including the docking score or energy.

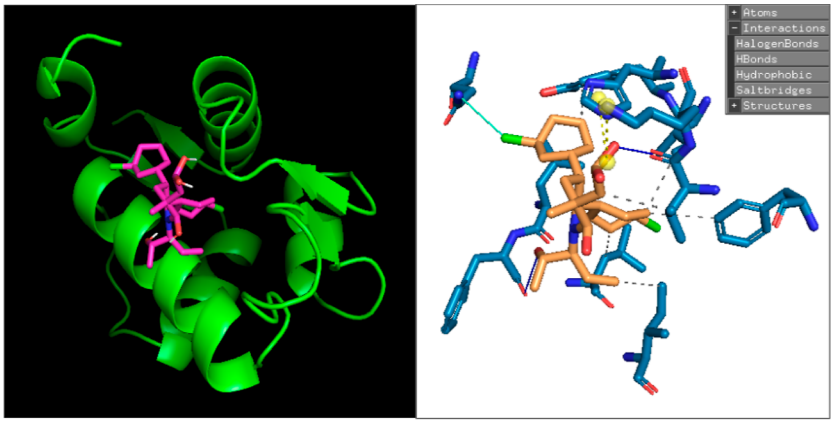


Figure 7. Example of Scipion-chem docking over a target (4ERF-A) with its actual ligand (OR3). The left side shows the target in cartoon representation with one of the nine positions found for the molecule in PyMol using Vina. On the right side, the same docked molecule is visualized in PyMol using PLIP, plotting and listing all the noncovalent interactions between the ligand and the target, only representing the interacting residues.

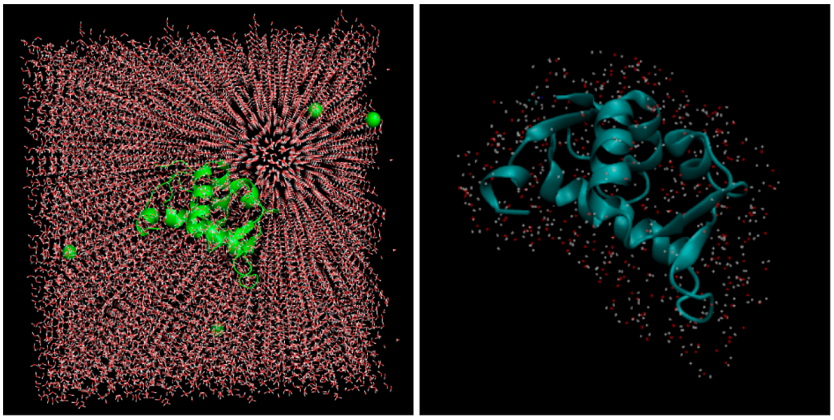


Figure 8. Example of the Scipion-chem solvated MD system (4ERF-A). The left panel shows the protein in a cubic water box and a number of ions (green spheres) counter the charges of the protein using PyMol. A snapshot of the same system is visualized on the right panel by using VMD, showing only the closest water to the protein.

https://doi.org/10.1021/acs.jcim.3c01085
J. Chem. Inf. Model. 2023, 63, 7873−7885

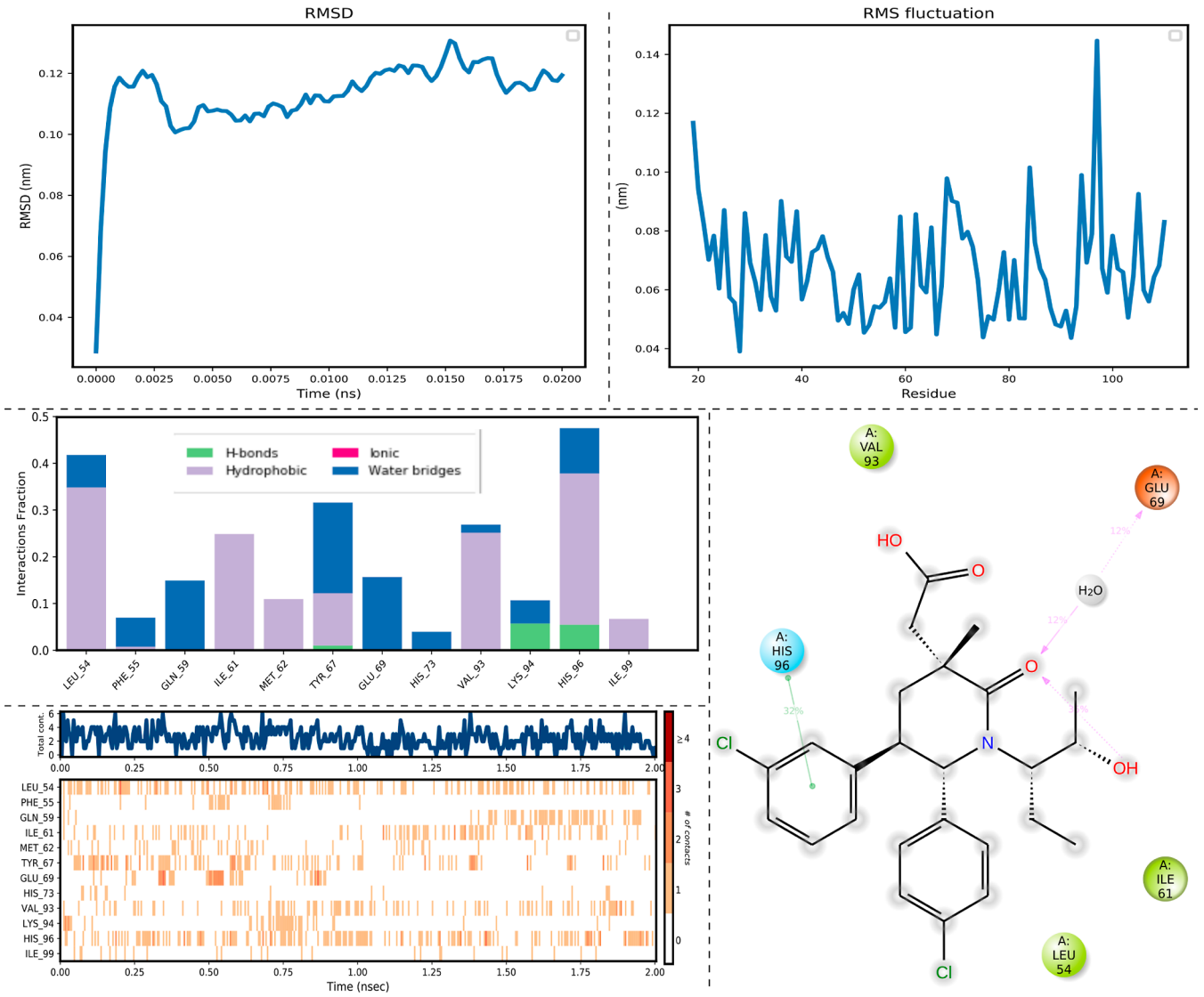


Figure 9. Example of Scipion-chem MD analysis that can be performed on the trajectory. On the top left and right panels, different analyses on the mobility of the protein (4ERF) throughout the trajectory are performed using Gromacs. On the bottom panels, Schrödinger is used to analyze the trajectory of 4ERF together with the OR3 ligand, and the images show representations of the target−ligand contacts.

separately, it might be difficult to keep track of all the decisions, parameters, or intermediate files, especially if these intermediate files sometimes need conversions or modifications.

5. Comparison: Even though having several software options is mostly helpful, it also adds the problem of deciding which results should be trusted or how to compare them.

Scipion-chem is an extension of the Scipion workflow engine[14−16] that has been designed to get rid of these complications by (1) automatically installing the software of interest and building any necessary environment to avoid conflicts; (2) automatically converting the intermediate files using OBabel,[17] RDKit,[18] or Biopython[19] when necessary; (3) offering a graphic user interface (GUI) to design and manage the workflows and running the programs; (4) saving and organizing all the parameters, workflows, and files in a straightforward folder structure; and (5) providing intuitive viewers and consensus protocols to compare and extract the most relevant information out of results from different software in equivalent steps.

2. VDS WORKFLOW

In this section, we follow the typical VDS pipeline steps to characterize promising ligands for a protein target, explaining each step and the available programs and tools in Scipion-chem. The PDB structure 4ERF[20] will be used as target throughout the workflow to give examples and figures of the outputs of each step. In Figure 1, we show the software integrated by Scipion-chem and the steps in which they are used. Moreover, the viewers included along the pipeline are also shown along with some extra functionalities that can be used from the platform.

2.1. Molecule Import. This step, previous to those presented in Figure 1, involves adding either a new target or ligand structures into a Scipion project. The import protocols allow the user to intuitively choose the origin of the structures and download the corresponding files if necessary.

https://doi.org/10.1021/acs.jcim.3c01085
J. Chem. Inf. Model. 2023, 63, 7873−7885

2.1.1. Target Import. The targets for VDS are mainly proteins whose structure can be characterized by diverse methods such as X-ray crystallography or cryo-electron microscopy. These methods ultimately generate the files that contain the information on the structure. The most well-known structure file types are pdb and cif, usually downloaded from the PDB database.[21] From Scipion, the user can choose whether to download an existing structure from the PDB using its unique identifier or import the structure from a local file on his/her computer. In addition, Scipion includes protocols for similarly importing sequences and using modeling programs such as AlphaFold[22,23] in order to get their corresponding structures.

2.1.2. Ligand Import. The hypothetical ligands are usually small molecules that can interact, fit on the protein surface, and change the target's functionality. The ligands can be imported using different formats depending on the type of ligand representation, 2D or 3D, such as sdf, mol2, or pdb, and others simply contain atomic information on the molecule, without specifying any coordinates, such as smi. Many of these formats are supported by Scipion-chem using OpenBabel and RDKit. The user can choose to import the ligand structures from

1. Local file(s): Import the ligand(s) structures from local file(s).
2. Predefined libraries: Throughout the years, the drug screening community has defined several libraries of ligands based on specific criteria. Scipion-chem allows downloading some of these default libraries (e.g., ZINC[24] and ECBL[25]).
3. Database IDs: Import ligand(s) by their ID in several databases such as PubChem,[26] ChEMBL,[27] BindingDB,[28] or ZINC.[24] Some of these databases group ligands based on some criteria. The IDs of these groups can also be used to download the components of the group.
4. Atomic structures: Some atomic structure files from PDB include small molecules that can be extracted.
5. Draw molecules: Scipion-chem includes a protocol calling JChemPaint,[29] a Java program that allows users to draw small molecules manually.

2.2. Molecule Preparation. Usually, both the protein target and the small molecules used as hypothetical ligands must be prepared to have some characteristics needed by the posterior software in the pipeline. Figure 2 contains a graphical example of target and ligand preparations.

2.2.1. Target Preparation. Depending on the method and procedure for resolving the protein target structure, the resulting structure file might have slightly different characteristics that we want to modify. The main modifications include adding missing hydrogens, removing water molecules or other heteroatoms, and removing irrelevant chains. Some of the included programs can also perform more advanced preparations such as assigning new partial charges to the atoms, performing energy minimization, or adding missing atoms in the structure. Moreover, Scipion also includes programs specifically oriented to modeling and modifying the atomic structures which can also be used to accomplish this step.[23] In Figure 1, the different options for target preparation are shown. These protocols generate an output of type AtomStruct, containing the prepared atomic structure.

2.2.2. Ligand Preparation. As for the target structure, some modifications may need to be performed over the ligand structures to make them compatible with the following steps. Similarly, ligands can be treated to add hydrogens or assign charges, among other options. Another preparation specific to ligands is the optional generation of conformers, alternative energetically favorable conformations that the molecule can acquire. Depending on the downstream pipeline programs to use, the generation of conformers might or might not be necessary. In Figure 1, the different options for the ligand preparation are shown. These protocols generate an output of type SetOfSmallMolecules, containing the prepared ligand structures.

2.3. Molecule Filtering. The steps described in this section reduce the computations the posterior docking step needs. It includes the definition of regions of interest (ROIs) on the target structure, on one hand, to reduce the docking space to explore and, on the other hand, to reduce the number of ligands to dock by discarding poor, unpromising ligands based only on their own characteristics.

2.3.1. ROI Definition. In Scipion-chem, we define the structural ROIs as groups of atoms in the target that are of concern for some reason. Depending on their origin, they can be represented as points over target atoms, over the surface of the target, or near it (Figure 3). From a molecular docking perspective, structural ROIs reduce the total space to explore from the complete target to a few promising or interesting sites. However, they are ultimately groups of atoms that can be of interest for any other topic inside the platform. There are a number of intuitive protocols to define the structural ROIs in Scipion-chem.

1. Manually: The user might want to manually define the structural ROIs directly from coordinates, specific residues, existing ligands, interchain protein−protein interfaces, or patterns of near residues in the target.
2. Based on target characteristics: By examining certain structural properties of the target and its corresponding atoms and residues, the structural ROIs can be identified based on predetermined thresholds. In Scipion-chem, we can, for example, define the structural ROIs for sites with a high SASA (solvent-accessible surface area), which can be visualized in the same protocol (Figure 3).
3. Predicted pockets: There are programs specifically designed to predict those sites of the target that are more likely to interact with a ligand. The exact procedure for determining these regions differs depending on the program. Still, they are usually concave regions or pockets where the ligands find themselves more surrounded by the target atoms, and therefore, the interaction can be energetically more favorable. In Scipion-chem, we have incorporated FPocket, P2Rank, AutoSite, and SiteMap programs, and their results are all parsed as the structural ROIs.
4. Sequence ROIs: Parallel to the structural ROIs, we defined sequence ROIs, which are defined as a group of consecutive residues on a protein sequence. There are also several ways to define sequence ROIs, such as manually, based on conservation, or existing natural variants or mutations. These sequence ROIs can be mapped to the structural ROIs based on the alignment of their sequence and the sequence of a protein structure.

At this point of the pipeline, Scipion-chem offers a consensus tool to compare the structural ROIs obtained

https://doi.org/10.1021/acs.jcim.3c01085
J. Chem. Inf. Model. 2023, 63, 7873−7885

from different sources and extract those shared among them. It works by clustering the structural ROIs based on the residue overlap. A set of contact residues and atoms ultimately defines each structural ROI. These clusters define whether a pair of structural ROIs approximately defines the same region. Once the clusters are built, only those containing a determined number of ROIs are kept, which might be useful in different situations. For example, we might want to keep the most promising protein pockets by running the consensus over several sets of pockets obtained from different tools. This way, only the most druggable pockets according to both programs will remain. Another more trivial but still useful example is to run the consensus protocol over a set of predicted pockets and a structural ROI defined by hand on a residue of interest, for example, a residue involved in the enzyme reaction of the protein. Using this procedure, we will extract only the pocket(s) where the residue of interest is found (Figure 4).

2.3.2. Ligand-Based Filtering. We can filter the ligands before the docking step using an LB step. Scipion-chem includes several filters based on only the characteristics of the ligand. To do so, we use the RDKit package to include the following filters based on

1. Chemical features: We include two protocols based on the molecule features. The first one uses the ADME (absorption, distribution, metabolism, and excretion) filter to evaluate which molecules have desirable pharmacokinetic properties. The second one applies the PAINS (pan-assay interference compounds) filter to remove the ligands containing molecular patterns that would lead to highly unspecific interactions.
2. Molecule shape: This protocol applies a filter that takes out those molecules whose shape is not similar enough to the shape of another molecule. For example, this filter is used when we know the structure of a reference ligand, and we would like to find more ligands with a similar shape, but that might have different chemical compositions. We support shape comparison with RDKit or Shape-it, using Tanimoto, Protrude, or rmsd distances.[30]
3. Molecule fingerprints: Molecular fingerprints are encoded representations of molecules. They typically represent the absence or presence of specific molecular patterns in the molecule. As with the previous filter, this one will remove those molecules whose fingerprints are not similar enough to that of another molecule. The user can choose among the Morgan or MACCS fingerprints[31] and whether to use Tanimoto or Dice similarity coefficients.
4. Pharmacophore: Pharmacophores are a type of molecular representation.[32] They extract molecular features of one or more molecules and represent them in 3D space, usually as spheres (see Figure 5). For another molecule to pass a pharmacophore filter, it must be possible to align and match its own features to those in the pharmacophore, with some degree of error. Scipion-chem includes protocols to (1) generate pharmacophores out of a group of small molecules, (2) define and manually modify RDKit pharmacophores, and (3) run the filter over a set of small molecules. Pharmacophores can be powerful tools for determining the most relevant features a ligand must have to interact with a receptor site. The user might, for example, want to build a pharmacophore out of the best results of a structural-based docking to perform a pharmacophore-based docking later. Scipion-chem currently supports pharmacophore generation and filter protocols through RDKit.

2.4. Docking. Molecular docking is usually the main process in an SB VDS workflow. It calculates the pose of the desired small molecules over the surface of the target, trying to find the position with the most favorable energy, which, hopefully, will coincide with the actual position of the ligand in the case this interaction really occurs. Usually, every docking method relies on a scoring function that ranks the positions provided by a search method and the different tested ligands. The scoring function and the search method are variable among the numerous methods the scientific community has developed. In Scipion-chem, we have integrated six docking protocols: AutoDock4, Vina, AutoDock-GPU (all from Autodock), LeDock (from LePhar), DARC (from Rosetta), and Glide (from Schrödinger). The user can choose among these methods whether to dock a set of small molecules over the whole protein or a set of structural ROIs and choose the different parameters that the software allows. The results can be visualized grouped by ROIs or small molecule and either in ChimeraX or in PyMol (see Figure 6).

To ensure the comparability of docking results obtained from each of these methods, the molecules that have been docked onto the target can be rescored using a consistent scoring function. In Scipion-chem, we have integrated two protocols for docking rescoring: the first uses the AutoDock4 scoring function and the second uses the ODDT package. This package rescores the docking positions using a number of scoring functions, and the user can choose a list of these that will be stored in each molecule metadata. Therefore, once the docking poses are rescored using the same function, we developed a consensus protocol to extract the most relevant poses from a set of different docking results. To do so, the docking positions are clustered based on their rmsd, and only clusters containing a determined number of poses are considered part of the consensus. Then, the molecule with the best chosen score or energy is used to represent the cluster. This consensus does not need the previous rescoring since the clustering process would proceed similarly. Still, thanks to it, we can compare the different poses in the cluster and choose the most favorable.

Finally, another tool is integrated to analyze the docking poses in detail: PLIP (protein−ligand interaction profiler)[33] (Figure 7). This tool plots in PyMol the docking pose and the different noncovalent interactions that the target and the ligand would have based on the distance and the atom types.

2.5. Molecular Dynamics. Molecular dynamics (MD) is a broad and complex field of science that tries to simulate at the atom level the interactions and dynamics of molecules, trying to represent them in a closer way to reality, compared to usual docking methods. Therefore, MD calculations are usually computationally expensive, and it may not be worth using them for VDS studies compromising many hypothetical ligands. However, under some circumstances, simulating the interactions of a limited number of specific target−ligand pairs might be interesting. For example, in our case, if our filters during the VDS pipeline are strict enough, we could extract the best few docked poses and then simulate these pairs. This way, we would further check how strong their interactions are and whether the ligand would actually stay in the receptor pocket

https://doi.org/10.1021/acs.jcim.3c01085
J. Chem. Inf. Model. 2023, 63, 7873−7885

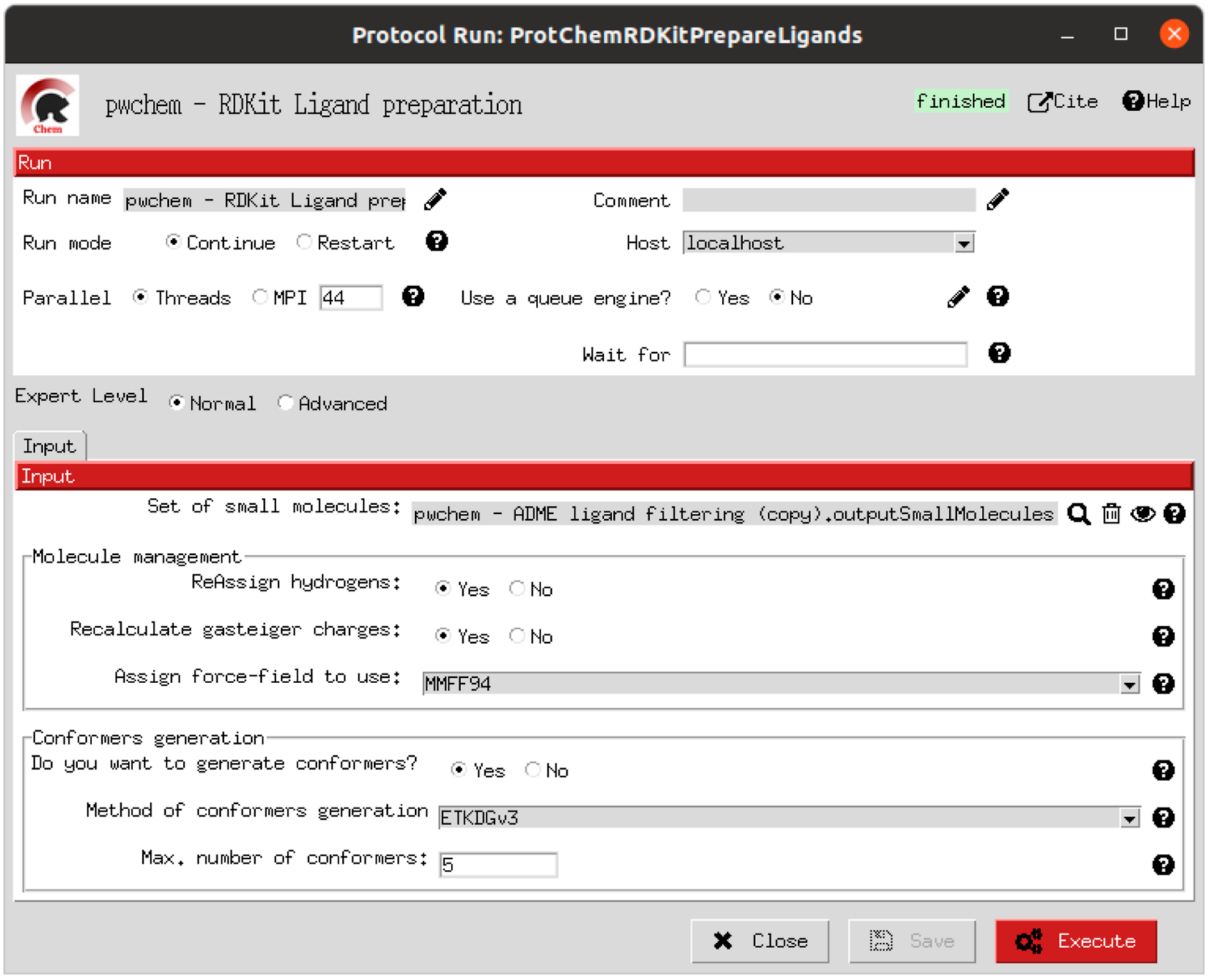


Figure 10. Scipion-chem RDKit ligand preparation form. The figure shows the Scipion forms corresponding to the RDKit ligand preparation described. The form contains an upper section where technical (run) parameters can be set (protocol name, number of threads, queue system, etc.) and a second section (input) where the user can define the input set of molecules and the different parameters RDKit must use for the preparation.

enough time to act as a drug. Moreover, MD might be useful for many other purposes involving simulation of the receptor or the ligand separately to study their behaviors freely. In Scipion-chem, we provide several packages for preparing and running full MD simulations. These include Gromacs,[34] AmberTools,[35] OpenMM,[36] and Desmond (from Schrödinger). For the case of Desmond, in addition to the standard viewers, the Maestro viewer is also available. These plugins contain protocols for

1. System preparation: The protocol takes the structure of either a receptor or a docked ligand to prepare a solvated system (Figure 8). The different parameters accepted to tune these systems by each software are provided in the protocol GUI, including the force fields to use, size and form of the water box, whether to add ions and in which quantities or concentrations, etc. The resulting solvated systems can be visualized using PyMol from the Scipion project.
2. Simulation: The user can define a list of steps to be performed consecutively, choosing the parameters for each of them and whether to store or not the trajectory out of each step simulation. The steps included can perform energy minimization, equilibration, or just production simulations. The list of steps is modified in the protocol GUI using wizards to ensure the correctness of the defined steps. The results of these protocols are the molecular systems after the simulation and, if stored, the trajectories followed by these system atoms through the simulation. Both can be visualized using PyMol and VMD, and several trajectory analyses are included to explore the simulation results further (see Figure 9).
3. Trajectory modifications: A protocol to perform trajectory or system modifications is included to allow for standard operations such as removing water from a trajectory or subsampling it.

3. SHOW CASE

In this section, we executed the main VDS workflow in Scipion-chem, including the consensus docking protocol, in a bigger data set. This example, with code FABP4, was extracted from the DUD-E database[37] and contains 47 active molecules, each of them with around 50 decoys, making a total of 2749 decoy molecules. Both the active, decoy libraries and the receptor structure are imported to the Scipion GUI from the files provided by DUD-E. Then, the main VDS workflow is executed, including the use of an LB filter, a receptor binding site prediction, and 3 different docking programs (AutoDock-GPU, AutoDock Vina, and LeDock). Finally, their results are combined by using ODDT rescoring and our consensus

https://doi.org/10.1021/acs.jcim.3c01085
J. Chem. Inf. Model. 2023, 63, 7873−7885

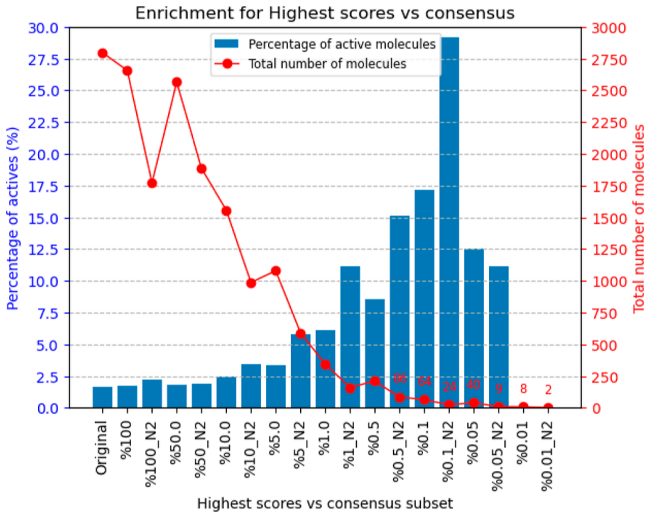


Figure 11. Scipion-chem consensus protocol enrichment. The graph shows the percentage of actives (blue bars) and the total number of molecules (red dots) for each of the subsets generated in the workflow. The subset "Original" represents the original set imported from DUD-E; "% 100" represents the subset of molecules remaining after the described LB filtering (which slightly improves the enrichment), and then each of the consensus subsets generated by applying a best ODDT score ranking filter for the top % x and consensus docking with parameter N2.

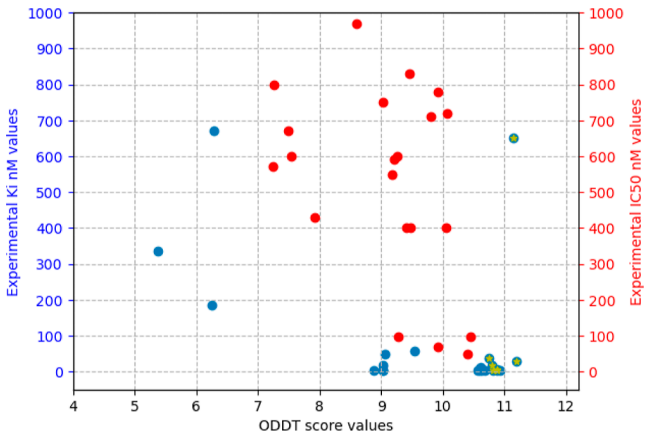


Figure 12. Experimental interaction values (nM) of actives in K_i (blue) or IC_{50} (red) against the corresponding values in the ODDTScore Vina. The yellow stars specify actives found in the best consensus set.

docking protocol. Thanks to the Scipion workflow engine, all of these VDS screening steps can be scheduled and run in parallel as a complete workflow, where the user can specify how to manage the resources devoted to each of the steps and software. Moreover, for each of these steps or protocols, Scipion GUI forms are accessible to easily modify and explore the different parameters involved in their execution (see Figure 10).

In the following lines, we will go through the steps involved in this VDS workflow. A more detailed description of these steps, together with the corresponding forms, the actual workflows, and the data generated, can be accessed in our GitHub docs page: https://scipion-chem.github.io/docs/publications/scipion-chem_vds.

1. Import: In this example, we imported the structures directly from the pdb files (for the receptor, corresponding to PDB 2NNQ)[38] and sdf files (for the ligands) provided by DUD-E. The forms provided by Scipion allow the user to choose the origin of the structure and, in the case of the small molecules, the molecule handler (RDKit or OpenBabel) to use and if a 3D reconstruction is needed.
2. Preparation: Once the structures are imported into the Scipion workflow, separate preparation steps are performed for the receptor and ligand libraries. In this case, we used the protein preparation protocol in the OpenMM plugin, which uses PDBFixer for the receptor protein, and RDKit for the preparation of the ligands. In each of the forms, the user is asked about the preparation parameters desired, such as removing undesired atoms (water and other nonprotein entities) or adding missing atoms in the receptor, which force fields to use, and whether to generate conformers (5 for our example) in the preparation of the ligands (shown in Figure 10).
3. ROI definition: In this particular example, P2Rank is used to predict the most promising pockets in our receptor, which will become those ROIs where we will direct the docking processes. Then, this P2Rank protocol is followed by a filter protocol to extract only the 2 best pockets predicted, in order to speed up the downstream workflow.
4. LB filtering: On the ligand side, a filtering step is used by passing the ADME LB filter protocol to our active and decoy molecules. After this filter, we continue the processing with 46 active and 2611 decoy molecules, discarding 1 and 138 molecules, respectively.
5. Docking: This step involves the execution of 3 independent docking programs (AutoDock-GPU, AutoDock Vina, and LeDock) over the 2 defined ROIs and both the active and decoy prepared libraries. In practice, this is the slowest step of the workflow and therefore becomes the usual bottleneck in its execution, so it is important to choose appropriate resources for them. In our case, the forms allow us to define the number of threads and GPUs (only for AutoDock-GPU) to allocate for each of the executions. Moreover, as the previous cases, the forms also include the parameters that the user can tweak to define the docking processes, such as the number of docking poses to generate for each of the molecule conformers. In our case, we chose five, resulting on 96226, 109139, and 76830 docking poses for each of the cited docking programs.
6. Rescoring: In order to combine and compare the docking poses generated by each of the software, we need to first evaluate those poses using the same scoring function. In this case, we use the ODDT score protocol to rescore all the docking poses with its Vina score function.
7. Filter and consensus: Finally, the rescored poses can be combined and ranked, and the consensus protocol can be applied to cluster and extract the most promising docking positions. In our example, different ranking filters and consensus parameters were used to evaluate the results. Nine filtered subsets of our docked molecules containing the 0.01, 0.05, 0.1, 0.5, 1, 5, 10, 50, and 100% of the highest-scored poses were

https://doi.org/10.1021/acs.jcim.3c01085
J. Chem. Inf. Model. 2023, 63, 7873−7885

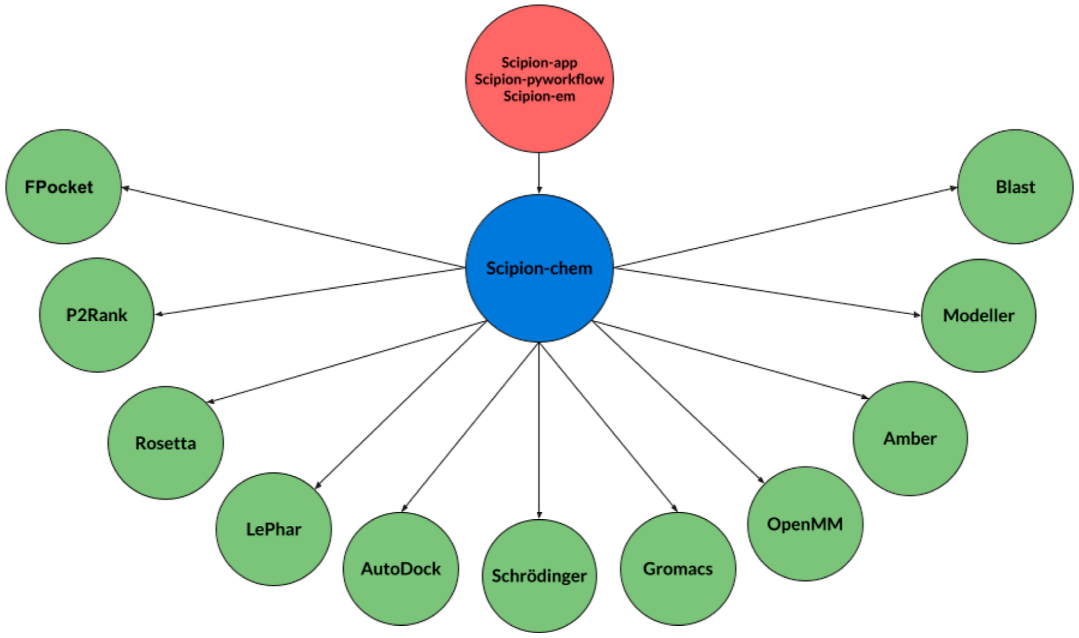


Figure 13. Scipion-chem plugin schema. The red circle represents the Scipion workflow engine, which is made of three parts. The Scipion-chem core plugin is a blue circle containing the main functionalities and tools needed by the rest of the plugins, in green, which depend on it.

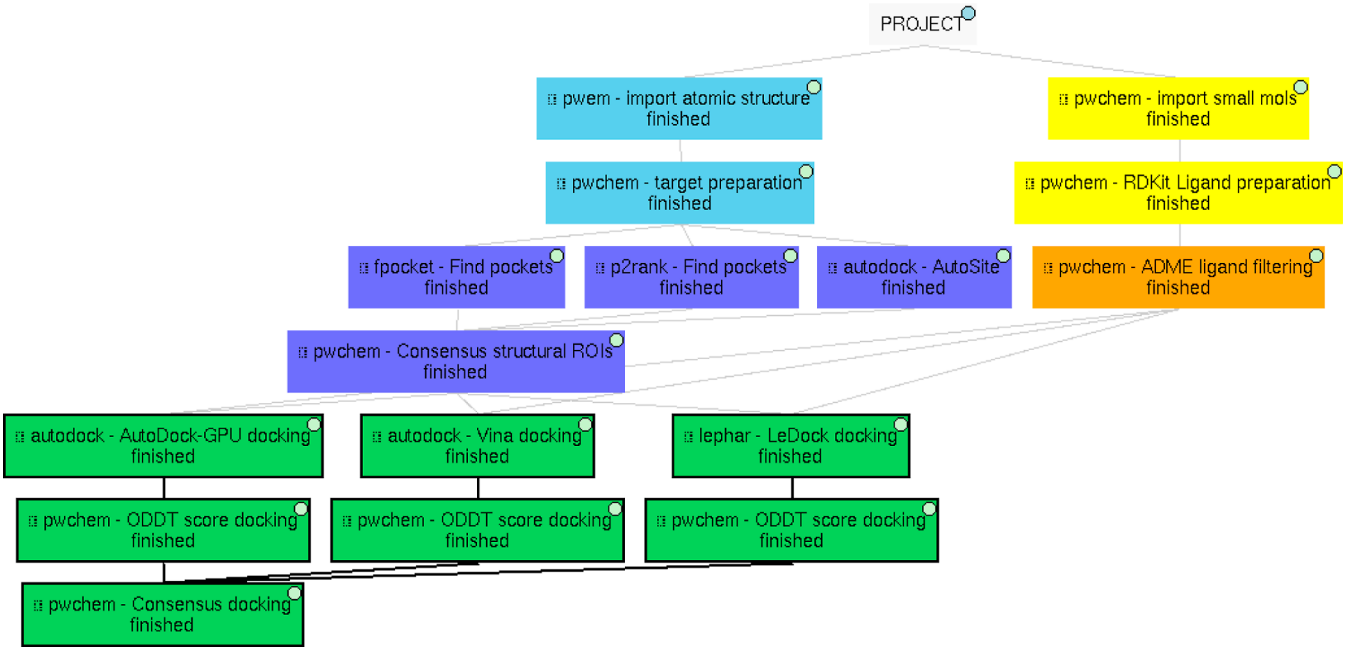


Figure 14. Scipion-chem VDS workflow example following the color convention in Figure 1. The results of this workflow are small molecules that have passed the ligand filters and docked to the predicted pockets (structural ROIs) from the receptor. The most promising pockets and docked poses are extracted using the consensus protocols described above.

generated. Then, two consensus protocols were executed for each of these subsets with a difference in a vital parameter. First, both consensus runs will produce the same pose clusters; however, one of the consensus executions will only consider sufficient those clusters containing at least one pose from each of the three docking software (N3), while the other, more permissive one, will consider sufficient those that contain at least poses from 2 docking software (N2). This way, we intend to generate sets enriched in active molecules and smaller than the original set of 2796 molecules. The results of this experiment comparing the filtering vs N2 consensus are shown in Figure 11, where we can observe the enrichment of actives vs decoys of the output subsets and the total number of molecules kept for each of them. Subsets labeled % *x* show the enrichment for the sets generated only passing the score filter, while those labeled % *x*_N2 represents the corresponding set generated after passing the score filter plus the consensus protocol. A similar image with the results for the N3 consensus can be found in our GitHub documentation page.

https://doi.org/10.1021/acs.jcim.3c01085
J. Chem. Inf. Model. 2023, 63, 7873−7885

As we can infer from the graphs, both strategies lead to a considerable enrichment of the original data set as the percentage of actives (the blue bars) is generally enhanced, while the number of total molecules in the subset (the red line) is reduced. For our FABP4 example, from the original 2796 (47 actives to 2749 decoys) molecules (1.68% of actives), we obtained considerable enrichment in both the filtered and filter plus consensus subsets. For instance, we obtained a subset of 64 molecules where 11 actives were kept (17.19%) for the 0.1% filtered subset, or once this same subset is passed through the N2 consensus, we further enriched it to keep 7 actives out of just 24 molecules (29.17%).

Therefore, we were able to reduce the total number of molecules in the original set while significantly enhancing the proportion of actives. However, the user must be careful not to reduce too much the number of docking poses with the score filter since we can observe that subsets below 0.05% lose all or most of the active molecules.

Additionally, in Figure 12, the experimental values for the interaction of the active molecules and the receptor are represented and plotted based on their experimental K_i (blue) or IC_{50} (red) values and their best pose ODDTScore. Those points with a yellow star correspond to the active molecules present in the best resulting consensus data set (% 0.1_N2). As we can observe, the ODDTScore seems to correlate relatively well, and most of the highest ODDTScores represent the best experimental affinities, which are captured in the consensus.

In summary, a complete VDS workflow was executed from the Scipion GUI on the FABP4 data set, which allowed to run all the different software involved from the same program, while maintaining the flexibility of those program executions since the user can easily modify both the functional and resource management parameters. In addition, the Scipion API allowed for the automation and parallelization of all of the tasks involved in the workflow. Finally, the use of new Scipion-chem own tools, such as the consensus docking protocol, provides the user with further resources and, together with the viewers and filter protocols, can be helpful for the VDS process. In future works, we will keep developing and integrating new tools. Specifically, we focus on improving our consensus protocols in order to generate robust and reliable results from the combination of several outputs.

4. IMPLEMENTATION

Scipion-chem is a Python-based workflow package designed to work as a plugin for Scipion, which is therefore necessary for its installation. Then, Scipion-chem works as the core plugin for the rest of the chem plugins. Today, 11 different plugins plus the core are available in the Scipion-chem website (see Figure 13), with around 80 different protocols to be executed. Still, we expect to continue to incorporate more plugins and protocols in the future.

These plugins integrate the previously described programs into the workflow, which are currently mainly related to VDS and MD. Inside Scipion, the user can visualize the workflows as a set of connected boxes, each representing a protocol like the one we previously described. These protocols can be found in various lists on the left-hand side of the screen or by using a search operation. In Figure 14, an example of a VDS workflow is shown following the schema in Figure 1. Inside each of these protocol boxes, the different parameters governing the execution of the program can be modified in the protocol form.

5. CONCLUSIONS

The number of tools related to VDS keeps growing each year, with new or updated scripts and algorithms that try to solve specific parts of the drug discovery process accurately. While this increasing number of software offers high flexibility and possibilities, it also poses a high difficulty barrier to users who must learn how to use each of these programs, usually in combination. Hereby, we have presented Scipion-chem, an open and interactive platform for VDS that eases access to this software and provides interoperability among them. Scipion-chem is built over the workflow engine Scipion, which provides complete traceability and automation of the workflows and software installation. Advanced users may work with a python API that enhances the automatic and programmatic access to Scipion-chem tools. Finally, the user can use different consensus tools that compare the results from different software to extract the most relevant pieces of common information.

Even though Scipion-chem was born focused on VDS, its name suggests that it can be projected to a range of chemoinformatic fields. From the Scipion team, we are already involved in expanding Scipion-chem to MD (with already useable tools) and quantum mechanics, and we hope to release new plugins related to these soon. We will keep working on maintaining and offering new VDS-related tools such as ligand retrosynthetic analysis or fragment-based screening.

AUTHOR INFORMATION

Corresponding Author

Daniel Del Hoyo − *National Center of Biotechnology (CNB-CSIC), Madrid 28049, Spain;* orcid.org/0009-0005-3321-1810; Email: ddelhoyo@cnb.csic.es

Authors

Martin Salinas − *National Center of Biotechnology (CNB-CSIC), Madrid 28049, Spain*

Alba Lomas − *National Center of Biotechnology (CNB-CSIC), Madrid 28049, Spain*

Eugenia Ulzurrun − *Center for Biological Research (CIB-CSIC), Madrid 28040, Spain*

Nuria E. Campillo − *Center for Biological Research (CIB-CSIC), Madrid 28040, Spain; Institute of Mathematical Sciences (ICMAT-CSIC), Madrid 28049, Spain;* orcid.org/0000-0002-9948-2665

Carlos Oscar Sorzano − *National Center of Biotechnology (CNB-CSIC), Madrid 28049, Spain*

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.jcim.3c01085

Notes

The authors declare no competing financial interest.

All discussed Scipion-chem plugins can be found openly available in https://github.com/scipion-chem as GitHub repositories. We encourage users to use them, check the code, look for bugs, and actively provide us any feedback or suggestions via GitHub or mail. Also, all the data discussed so far in this paper are documented and can be found in the corresponding GitHub page https://scipion-chem.github.io/docs/publications/scipion-chem_vds, which contains the importable workflows, the input data from DUD-E, and all the results generated. Finally, a list of the software installed by

https://doi.org/10.1021/acs.jcim.3c01085
J. Chem. Inf. Model. 2023, 63, 7873−7885

the referenced Scipion-chem plugins is detailed in this same documentation, together with their versions and sources.

■ ACKNOWLEDGMENTS

This research work was funded by the European Commission—NextGenerationEU (Regulation 2020/2094), through CSICs Global Health Platform (PTI Salud Global) and an FPU (Formación de Profesorado Universitario) grant from the Spanish Ministry of Education. In addition, we would like to express our gratitude to Pedro Febrer, Aida Pinacho and Carlos Roca for their contributions to the molecular dynamics section of this project that provided a starting point for our work.

■ REFERENCES

(1) Leelananda, S. P.; Lindert, S. Computational Methods in Drug Discovery. *Beilstein J. Org. Chem.* **2016**, *12*, 2694−2718.
(2) Lin, X.; Li, X.; Lin, X. A Review on Applications of Computational Methods in Drug Screening and Design. *Molecules* **2020**, *25*, 1375.
(3) McInnes, C. Virtual Screening Strategies in Drug Discovery. *Curr. Opin. Chem. Biol.* **2007**, *11*, 494−502.
(4) Varela-Rial, A.; Majewski, M.; De Fabritiis, G. Structure Based Virtual Screening: Fast and Slow. *Wiley Interdiscip. Rev.: Comput. Mol. Sci.* **2021**, *12*, No. e1544.
(5) Park, H.; Jeon, J.; Kim, K.; Choi, S.; Hong, S. Structure-Based Virtual Screening and De Novo Design of PIM1 Inhibitors With Anticancer Activity From Natural Products. *Pharmaceuticals* **2021**, *14*, 275.
(6) Klenner, A.; Hartenfeller, M.; Schneider, P.; Schneider, G. "Fuzziness" in Pharmacophore-Based Virtual Screening and De Novo Design. *Drug Discovery Today: Technol.* **2010**, *7*, e237−e244.
(7) Lill, M. A. Multi-Dimensional QSAR in Drug Discovery. *Drug Discovery Today* **2007**, *12*, 1013−1017.
(8) Neves, B. J.; Braga, R. C.; Melo-Filho, C. C.; Moreira-Filho, J. T.; Muratov, E. N.; Andrade, C. H. QSAR-Based Virtual Screening: Advances and Applications in Drug Discovery. *Front. Pharmacol.* **2018**, *9*, 1275.
(9) Repasky, M. P.; Shelley, M.; Friesner, R. A. Flexible Ligand Docking With Glide. *Curr. Protoc. Bioinf.* **2007**, *18*, 8−12.
(10) 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*, 905−919.
(11) Liu, N.; Xu, Z. Using LeDock as a Docking Tool for Computational Drug Design. *IOP Conference Series: Earth and Environmental Science*, 2019; p 012143.
(12) Le Guilloux, V.; Schmidtke, P.; Tuffery, P. Fpocket: An Open Source Platform for Ligand Pocket Detection. *BMC Bioinf.* **2009**, *10*, 168.
(13) Krivák, R.; Hoksza, D. P2Rank: Machine Learning Based Tool for Rapid and Accurate Prediction of Ligand Binding Sites From Protein Structure. *J. Cheminf.* **2018**, *10*, 39.
(14) de la Rosa-Trevín, J.; Quintana, A.; del Cano, L.; Zaldívar, A.; Foche, I.; Gutiérrez, J.; Gómez-Blanco, J.; Burguet-Castell, J.; Cuenca-Alba, J.; Abrishami, V.; et al. Scipion: A Software Framework Toward Integration, Reproducibility and Validation in 3D Electron Microscopy. *J. Struct. Biol.* **2016**, *195*, 93−99.
(15) Conesa, P.; Fonseca, Y. C.; de la Morena, J. J.; Sharov, G.; de la Rosa-Trevín, J. M.; Cuervo, A.; Mena, A. G.; de Francisco, B. R.; Carazo, J. M.; Sorzano, C. O. S. Scipion3: A Workflow Engine for Cryo-Electron Microscopy Image Processing and Structural Biology. *Biol. Imaging* **2023**, *3*, 1−22.
(16) Krieger, J. M.; Sorzano, C. O. S.; Carazo, J. M. Scipion-EM-ProDy: A Graphical Interface for the ProDy Python Package within the Scipion Workflow Engine Enabling Integration of Databases, Simulations and Cryo-Electron Microscopy Image Processing. *Int. J. Mol. Sci.* **2023**, *24*, 14245.
(17) O'Boyle, N. M.; Banck, M.; James, C. A.; Morley, C.; Vandermeersch, T.; Hutchison, G. R. Open Babel: An Open Chemical Toolbox. *J. Cheminf.* **2011**, *3*, 33.
(18) Landrum, G. RDKit: A Software Suite for Cheminformatics, Computational Chemistry, and Predictive Modeling. *Greg Landrum* **2013**, *8*, 31.
(19) Cock, P. J. A.; Antao, T.; Chang, J. T.; Chapman, B. A.; Cox, C. J.; Dalke, A.; Friedberg, I.; Hamelryck, T.; Kauff, F.; Wilczynski, B.; et al. Biopython: Freely Available Python Tools for Computational Molecular Biology and Bioinformatics. *Bioinformatics* **2009**, *25*, 1422−1423.
(20) Rew, Y.; Sun, D.; Gonzalez-Lopez De Turiso, F.; Bartberger, M. D.; Beck, H. P.; Canon, J.; Chen, A.; Chow, D.; Deignan, J.; Fox, B. M.; et al. Structure-Based Design of Novel Inhibitors of the MDM2-p53 Interaction. *J. Med. Chem.* **2012**, *55*, 4936−4954.
(21) Berman, H. M. The Protein Data Bank. *Nucleic Acids Res.* **2000**, *28*, 235−242.
(22) Jumper, J.; Evans, R.; Pritzel, A.; Green, T.; Figurnov, M.; Ronneberger, O.; Tunyasuvunakool, K.; Bates, R.; Žídek, A.; Potapenko, A.; et al. Highly Accurate Protein Structure Prediction With AlphaFold. *Nature* **2021**, *596*, 583−589.
(23) Martínez, M.; Jiménez-Moreno, A.; Maluenda, D.; Ramírez-Aportela, E.; Melero, R.; Cuervo, A.; Conesa, P.; del Caño, L.; Fonseca, Y. C.; Sánchez-García, R.; et al. Integration of Cryo-EM Model Building Software in Scipion. *J. Chem. Inf. Model.* **2020**, *60*, 2533−2540.
(24) Irwin, J. J.; Shoichet, B. K. ZINC—A Free Database of Commercially Available Compounds for Virtual Screening. *J. Chem. Inf. Model.* **2005**, *45*, 177−182.
(25) Meiners, T.; Stechmann, B.; Frank, R. EU-OPENSCREEN—Chemical Tools for the Study of Plant Biology and Resistance Mechanisms. *J. Chem. Biol.* **2014**, *7*, 113−118.
(26) Kim, S.; Thiessen, P. A.; Bolton, E. E.; Chen, J.; Fu, G.; Gindulyte, A.; Han, L.; He, J.; He, S.; Shoemaker, B. A.; Wang, J.; Yu, B.; Zhang, J.; Bryant, S. H. PubChem Substance and Compound Databases. *Nucleic Acids Res.* **2016**, *44*, D1202−D1213.
(27) Gaulton, A.; Hersey, A.; Nowotka, M.; Bento, A. P.; Chambers, J.; Mendez, D.; Mutowo, P.; Atkinson, F.; Bellis, L. J.; Cibrián-Uhalte, E.; et al. The ChEMBL Database in 2017. *Nucleic Acids Res.* **2016**, *45*, D945−D954.
(28) Liu, T.; Lin, Y.; Wen, X.; Jorissen, R. N.; Gilson, M. K. BindingDB: A Web-Accessible Database of Experimentally Determined Protein-Ligand Binding Affinities. *Nucleic Acids Res.* **2007**, *35*, D198−D201.
(29) Krause, S.; Willighagen, E.; Steinbeck, C. JChemPaint - Using the Collaborative Forces of the Internet to Develop a Free Editor for 2D Chemical Structures. *Molecules* **2000**, *5*, 93−98.
(30) Taminau, J.; Thijs, G.; De Winter, H. Pharao: Pharmacophore Alignment and Optimization. *J. Mol. Graphics Modell.* **2008**, *27*, 161−169.
(31) He, K. Pharmacological Affinity Fingerprints Derived From Bioactivity Data for the Identification of Designer Drugs. *J. Cheminf.* **2022**, *14*, 35.
(32) Horvath, D. *Chemoinformatics and Computational Chemical Biology*; Bajorath, J., Ed.; Humana Press: Totowa, NJ, 2011; pp 261−298.
(33) Salentin, S.; Schreiber, S.; Haupt, V. J.; Adasme, M. F.; Schroeder, M. PLIP: Fully Automated Protein−Ligand Interaction Profiler. *Nucleic Acids Res.* **2015**, *43*, W443−W447.
(34) Van Der Spoel, D.; Lindahl, E.; Hess, B.; Groenhof, G.; Mark, A. E.; Berendsen, H. J. GROMACS: Fast, Flexible, and Free. *J. Comput. Chem.* **2005**, *26*, 1701−1718.
(35) Salomon-Ferrer, R.; Case, D. A.; Walker, R. C. An Overview of the Amber Biomolecular Simulation Package. *Wiley Interdiscip. Rev.: Comput. Mol. Sci.* **2013**, *3*, 198−210.
(36) Eastman, P.; Swails, J.; Chodera, J. D.; McGibbon, R. T.; Zhao, Y.; Beauchamp, K. A.; Wang, L.-P.; Simmonett, A. C.; Harrigan, M. P.; Stern, C. D.; et al. OpenMM 7: Rapid Development of High

https://doi.org/10.1021/acs.jcim.3c01085
J. Chem. Inf. Model. 2023, 63, 7873−7885

Performance Algorithms for Molecular Dynamics. *PLoS Comput. Biol.* **2017**, *13*, No. e1005659.
(37) Mysinger, M. M.; Carchia, M.; Irwin, J. J.; Shoichet, B. K. Directory of Useful Decoys, Enhanced (DUD-E): Better Ligands and Decoys for Better Benchmarking. *J. Med. Chem.* **2012**, *55*, 6582−6594. Accessed: 2023-09-24
(38) Jacobson, B. L. *Crystal Structure of Human Adipocyte Fatty Acid Binding Protein in Complex with ((2′-(5-ethyl-3, 4-diphenyl-1H-pyrazol-1-yl)-3-biphenylyl)oxy)acetic Acid*, 2007. .

https://doi.org/10.1021/acs.jcim.3c01085
J. Chem. Inf. Model. 2023, 63, 7873−7885