



Applications of the NRGsuite and the Molecular Docking Software FlexAID in Computational Drug Discovery and Design

Louis-Philippe Morency, Francis Gaudreault, and Rafael Najmanovich

Abstract

Docking simulations help us understand molecular interactions. Here we present a hands-on tutorial to utilize FlexAID (*Flexible Artificial Intelligence Docking*), an open source molecular docking software between ligands such as small molecules or peptides and macromolecules such as proteins and nucleic acids. The tutorial uses the NRGsuite PyMOL plugin graphical user interface to set up and visualize docking simulations in real time as well as detect and refine target cavities. The ease of use of FlexAID and the NRGsuite combined with its superior performance relative to widely used docking software provides nonexperts with an important tool to understand molecular interactions with direct applications in structure-based drug design and virtual high-throughput screening.

Key words Computer-aided drug design, Binding mode prediction, Lead identification, Molecular docking, Molecular flexibility, Molecular recognition, Protein–ligand complex

1 Introduction

All biological processes are fundamentally guided by molecular recognition events. The study of molecular recognition aims at understanding the factors affecting the selectivity and specificity of molecular interactions. Advances in molecular biology, notably in protein identification, expression, purification, and structural determination, gave rise to an abundance of information about the three dimensional structure of macromolecules and their molecular complexes accessible through the Protein Data Bank (PDB) [1]. As a significant portion of molecular interactions may in principle be modulated by exogenous small molecules, the large-scale analysis of ligand–protein interactions helps understand the evolution of ligand binding [2] and paved the way for the development of computational techniques such as molecular docking. Docking is at the core of virtual high-throughput screening and is frequently used together with other computational techniques for a

wide range of applications in computer-assisted drug design. Molecular docking gained interest in the early stages of the drug discovery process because of its relatively low computational cost to evaluate many potential compounds [3].

Molecular docking can be used to tackle three different questions in drug design [4]: (1) Binding mode prediction, (2) Virtual high-throughput screening; and (3) Structure-based prediction of binding affinities. Binding mode prediction aims at obtaining the three-dimensional structure of the complex between a ligand of interest and the target. Binding mode prediction is used to understand the underlying interactions involved in the stabilization of the complex and can be used as a guide for structure-based drug design. Virtual high-throughput screening (vHTS) aims at detecting bioactive ligands from large compound databases. vHTS generally relies on binding-mode prediction but often requires the use of rescoring methods to determine the relative order of compounds with respect to their unknown binding affinities. The last task is the estimation of binding free-energy differences from the structure of the complex. Although it is possible to develop regression or machine learning based methods to predict binding-free energies, it is unclear how generalizable their results may be when encountering different ligands or target classes. It is generally assumed that only a structure based approach should be able to predict correctly binding free energies. A fast and accurate method to calculate binding free energy differences has not been found yet. Such a method would replace existing vHTS rescoring methods but would still require the accurate prediction of the three-dimensional structure of the ligand–protein complex.

Generally speaking, binding mode prediction explores in more detail the degrees of freedom that represent molecular flexibility [5] whereas virtual screening uses a faster search and scoring method to process more compounds and the structure-based prediction of affinities requires accurately predicted binding-modes and lengthy statistical-mechanics based calculations. However, the increase in computational power allows the use of slower docking algorithms, especially those focusing on binding mode prediction, in a high-throughput manner [6] with scoring functions precise enough to discriminate bioactive molecules from decoys [7–10]. Hence, binding mode prediction methods are standard in computer-aided drug discovery, either to explain the molecular mechanism behind the formation of a complex or to search for new bioactive ligands from virtual libraries through high-throughput virtual screening.

Recent advances in molecular docking focus on modeling the docking simulation more realistically through for example inclusion of structural water molecules and molecular flexibility [11–13]. In general, molecular flexibility of the target has at most been restricted to side-chain movements. However, attempts are being made at considering additional degrees of freedom accounting for

the target flexibility [14, 15] to account for conformational changes that occur upon binding [16], mostly because proteins exist in a multiconformational thermodynamic equilibrium [17]. Indeed, molecular rearrangements in the binding site are common [18] and, in approximately 30% of cases, are critical for ligand binding [19].

The molecular docking software FlexAID (*Flexible Artificial Intelligence Docking*) [20] has been developed as an open source binding mode prediction software with a focus on modeling molecular flexibility. Among other features, FlexAID searches the conformational landscape with a genetic algorithm, uses a soft and permissive scoring function based on the surface of contact between atoms, fully simulates ligand flexibility, uses a probabilistic rotameric approach to simulate side-chains flexibility in the target, simulates large-scale backbone movements of the target using the normal mode analysis method ENCoM [21], and allows for the simulation of covalent docking. The energy parameters utilized by the FlexAID scoring function were derived from the supervised learning (classification) of a large dataset of low energy false positive decoys (root-mean squared distance, RMSD >2 Å) and true positive decoys (RMSD ≤ 2.0 Å) for over 1300 ligand–protein complexes from the PDBbind database [22]. Notably, FlexAID outperforms many other open source molecular docking methods specialized in binding mode prediction such as Autodock Vina [23], FlexX [24], and rDock [25], particularly when molecular flexibility is crucial [20]. FlexAID is developed in C and C++, is distributed open source, and runs on all major operating systems for personal computers. FlexAID has an interactive PyMOL plugin graphical user interface, the NRGsuite [26] that greatly simplifies its use and permits the visualization in real time of the docking simulation as well as the detection, refinement, and measurement of cavities. In addition to being widely used as an educational tool, FlexAID has also been successfully used in a number of applications. Notably, to explain the binding mode of primary bile acid in a StaR-related lipid transfer domain [27], to target the germination protease and guanine riboswitch of *Clostridium difficile* as novel antibiotic targets, to analyze binding site specificity in the human serine protease Matriptase [28] and discover new bioactive molecules against the human serine protease TMPRSS6 [28], and further validate hypotheses of potential cross-reactivity targets responsible for the side-effects of approved drugs and their repurposing [29].

In the following sections, we demonstrate how to use FlexAID and the NRGsuite to predict the binding mode of a common drug to its molecular target. As an example of binding mode prediction, FlexAID will be used to predict the binding mode of the influenza neuraminidase inhibitor zanamivir (Fig. 1). We encourage the user to repeat the methodology with different ligands and parameters to

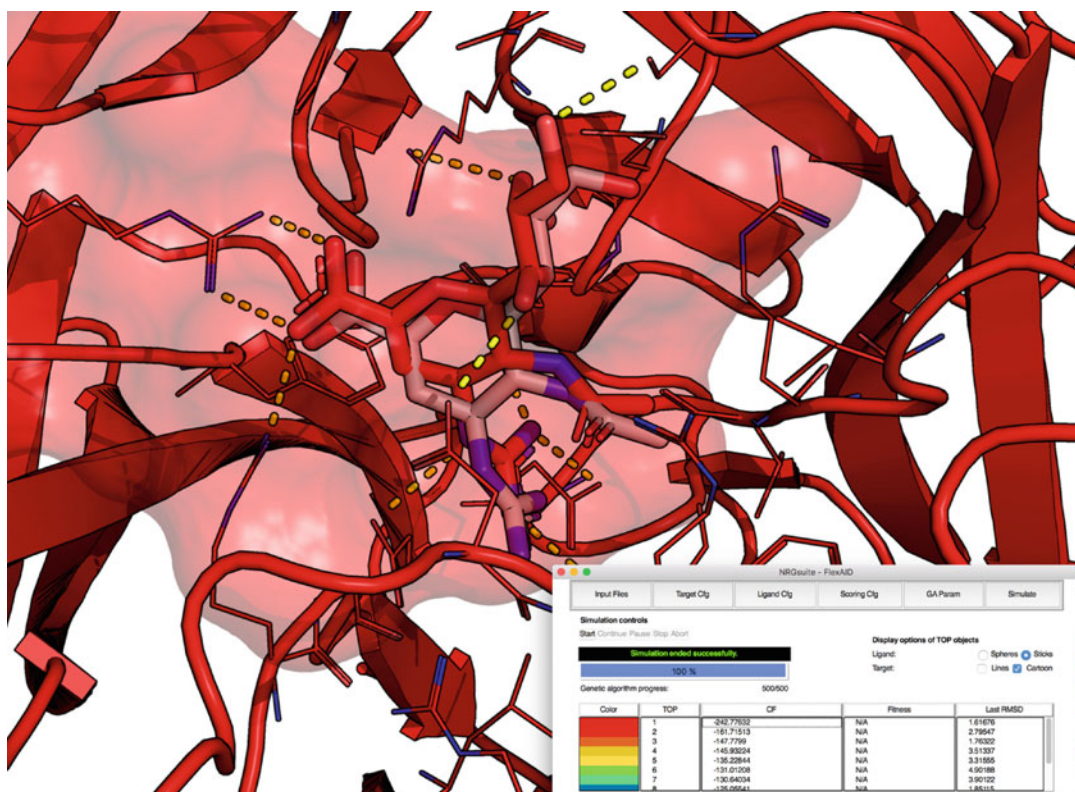


Fig. 1 The binding mode of zanamivir predicted by FlexAID (RESULT_1) is shown in red colored sticks with its predicted hydrogen bonds in yellow dashed lines. The pose of reference is shown in white sticks. By default, however, the target is color-coded using the colors in the table and is shown as “Cartoon.” The ligand and other optimized residues (including side-chains) are shown in white as sticks surrounded by its residues shown as lines. FlexAID Simulate tab (shown at the bottom right) displays information in a table format about the coloring scheme of the molecules for different results (column 1), the rank of the results (column 2) an evaluation of their energy of binding (column 3) and the RMSD from the reference (column 5)

compare results. The methodology explained here will allow the user to repeat the experiment with any complex and could easily be adapted for high-throughput applications by more advanced users by means of scripting the use of the input files created with the NRGsuite for a given target.

2 Materials

For this tutorial, it will be necessary to install the molecular visualization system PyMOL (*see Note 1*) and the NRGsuite software package that contains every tool to be used in this tutorial (*see Note 2*). All the software is available for the major operating systems, i.e., Linux, macOS, and Windows, but it is important to install the appropriate version of the NRGsuite for your system (*see Note 3*).

After the installation of both PyMOL and the NRGsuite, the last step required is to open PyMOL and install the NRGsuite as a plugin (*see Note 4*).

3 Methods

During this tutorial, the main objective will be to retrieve the experimentally observed binding mode of a drug in complex with its target. Although several experimental methods can now solve high resolution (*see Note 5*) tridimensional structures of these complexes, e.g., liquid and solid state nuclear magnetic resonance (NMR) and electron microscopy (EM), X-ray crystallography remains the most popular technique used in structure determination (*see Note 6*). The experimental structure of the complex is called the reference, and it will be used as a control to validate the accuracy of the predictions made during the molecular docking simulations (*see Note 7*).

The prediction of the binding mode of a drug-like molecule to a biologically relevant target of interest, mostly a chain of amino or nucleic acids (*see Note 8*), is entirely performed inside PyMOL and using the graphical user interface NRGsuite for the molecular docking software FlexAID. The following protocol can be executed on any standard personal computer and does not require any specialized hardware.

As a case study during this protocol, we will be interested in the drug zanamivir, commonly distributed under the name Relenza by Gilead Sciences. Zanamivir has been developed as a neuraminidase inhibitor to treat and prevent the transmission of infections by influenza A and B viruses. The neuraminidase enzymes (acylneuraminyl hydrolase, E.C. 3.2.1.18) hydrolyzes the glycosidic bond linking a terminal sialic acid and other sugar molecules located on a host cells' surface and/or on the hemagglutinin [30]. The cleavage of the sialic acid bond, specifically the α -ketosidic linkage between terminal sialic acid and adjacent sugar at the surface of the infected cell, eases the diffusion of viral particles from infected host cells toward new cells, thus facilitating the viral infection [31]. The determination of the tridimensional structure of the glycoprotein neuraminidase [32] in 1983 allowed the structure-based design of antiviral agents targeting the conserved structure of its binding site [33]. Interestingly, computer-aided drug discovery greatly contributed to the development of compounds leading to the FDA approval of zanamivir as a first-in-class neuraminidase inhibitor in 1999 [34].

The first step is to select a structure of zanamivir in complex with its target, the neuraminidase, in the PDB (*see Note 9*). The structure that will be used is a crystal of zanamivir bound to the

H1N1 neuraminidase solved by X-ray crystallography at a resolution 1.45 Å (PDB: 3B7E) (*see Note 10*).

All the subsequent steps will be performed in PyMOL. Text in italic following the “>” symbol represent commands to be entered in the PyMOL command line interface.

3.1 Preparation of the Work Environment

First, launch the version of PyMOL you installed for your system (*see Note 1*) and make sure that the NRGsuite is installed (*see Note 2*). The command line below will download the crystal structure of the 1918 H1N1 strain neuraminidase in complex with zanamivir.

```
> fetch 3b7e
```

1. The influenza virus neuraminidase glycoprotein is a homotetramer with fourfold circular symmetry. The selected PDB entry is a crystal of two monomers assembled into a homodimer. The experiment in this protocol will use a single monomer to model the complex. This command line removes the second monomeric biological assembly found in the chain B of the structure.

```
> remove 3b7e and chain B
```

2. The complex contains both the target, the neuraminidase, and the reference position of the ligand, i.e., the solution we want to retrieve with molecular docking. The command line below will extract zanamivir, the residue named ZMR, from the complex and create a separate object in PyMOL named ZMR. This object will use as the ligand for docking and as the reference that will be used to evaluate the precision in binding mode prediction.

```
> extract ZMR, 3b7e and resn ZMR
```

The work environment of the NRGsuite is project-oriented. A project must be activated to access the two main interfaces GetCleft (to detect, refine, and measure the volume of cavities) and FlexAID (to perform the docking simulations). When a project is activated, the objects you work with can only be saved within that active project. Only Target and Ligand object types are automatically saved in your project folder when loaded elsewhere on the computer (*see Note 11*). Create a new project for this tutorial by clicking on the *Plugin* menu item in the PyMOL main window (*see Note 12*).

```
Plugin → NRGsuite → New Project... → ‘your_project_name’  
(see Note 13).
```

3.2 Definition of the Target Binding Site with the GetCleft Interface

The definition of the binding site, i.e., the specific area(s) on the surface of the target that will be searched during the optimization process, is an important step to insure the success of a molecular docking experiment. Although it could be possible to use all the

accessible surface of the target as a potential binding site for the ligand during a molecular docking simulation, it is highly suggested to refine the conformational search of the ligand to specific areas of interest, hot spots, or, as it will be referred to further, clefts.

GetCleft, one of the two main tools integrated in the NRGsuite, is an algorithm that identifies surface accessible cavities (or clefts) in the target (also internal cavities depending on the parameters used) as potential binding-sites. The GetCleft interface allows you to adjust the parameters of GetCleft as well as to partition existing clefts and calculate their volume. The GetCleft interface contains a menu from where you can save/load clefts. The interface contains three tabs: *Generate*, *Partition*, and *Volume*.

1. Open the GetCleft interface from the Plugin menu found in the main window of PyMOL

Plugin → *NRGsuite* → *Open GetCleft...*

2. The *Generate* tab is activated by default upon GetCleft's opening and it allows the generation of clefts for a target, the filtering of the clefts generated and the measurement of their volumes. There are several parameters of interest that can be used here: the minimal and maximal radius of the spheres that will be used to locate and generate the cleft (this parameter controls the general shape of the cleft) (*see Note 14*), a given residue that must be included in the cleft (*see Note 15*), and the maximal number of clefts desired by the user. For this experiment, there is no need to modify these parameters.

First, select a target structure from the scrolling list of "PyMOL objects/selections" by clicking on the drop-down menu and selecting 3b7e (*see Note 16*). Upon selection of the target, simply click *Start* to retrieve available clefts for the neuraminidase. When clefts are generated, they are automatically loaded into the PyMOL viewer using unique color coded surface representations and unique object names (*see Note 17*). Specifically, clefts are in decreasing order of the number of spheres needed to build them, which is somewhat correlated to the volume of the clefts.

3. (*Optional*) Although this section is written as optional for this specific molecular docking experiment, it is important to keep in mind that most other biological targets will require a more precise definition of the binding site to enhance the precision of the predictions.

The *Partition* tab is used to partition the volume of an existing cleft when it is too large for the need of the user. The process is divided in three distinct steps:

STEP 1: Select a parent cleft to partition. To partition a cleft, simply select the cleft of interest from the list of clefts. Only objects

containing the *_sph_* tag are displayed in the list. The selected cleft will flash in the PyMOL viewer. Generally, the largest cleft, numbered 1 and colored red, is the one that will be used for partitioning.

STEP 2: Add one or more sphere(s) to partition. To partition a cleft, you need to add one or multiple partition sphere(s). The cleft volume contained into the volume of the spheres is conserved.

To add a sphere, click the “Add” button. A wizard will activate and a partition sphere will appear centered onto your cleft. The name of the sphere object is SPHERE_X__, where X is a numerical value of the index of the partition spheres inserted.

Once you set the size and location of the partition sphere properly, click the “Done” button in the Wizard (*see Note 18*). The partition sphere will appear in the Spheres objects list. If you wish to modify this sphere simply click the “Edit” button. A new wizard will activate. You can also delete partition spheres by clicking the “Delete” button (*see Note 19*).

Finally, the greyed-out volume of the cleft represents the volume that will be conserved when the portioned cleft is created.

STEP 3: Name the partitioned child cleft. Click the *Create* button and the partitioned cleft will appear as a new object in the PyMOL viewer with an added suffix “_pt” to its name.

4. (*Optional*) The *Volume* tab of the GetCleft interface calculates the volume of the clefts generated above. It is particularly useful for the user to compare the total available volume of a cleft and the estimated volume of a small molecule as it hints at the relative size of the space to search.

The clefts that appear are color-coded and are displayed as a list in the volume tab. If your cleft does not appear, try pressing the “Refresh clefts” button at the bottom. A cleft with an uncalculated volume has a null volume (0.0). You can estimate the volume of all clefts (“ALL” button), uncalculated clefts (“Remaining” button) or a selected cleft (“Selected” button). The calculated volume is displayed in Å³ upon completion.

5. Clefts are not automatically saved when generated. Thus, the user needs to explicitly save the clefts before importing them into the FlexAID interface. You can save clefts as long as they were generated in the same GetCleft session. If you generate clefts, then close the GetCleft interface without saving and reopen it, the clefts will not be available for saving.

To save the clefts, click the *Save* menu button in the upper left corner of the GetCleft window and save the files at the default location suggested in the file window. The NRGsuite cannot save a cleft elsewhere than the suggested directory (*see Note 20*).

6. Close the GetCleft interface window by clicking on the *Close* button in the lower right corner.

3.3 Configure, Perform, and Analyze the Molecular Docking Experiment with the FlexAID Interface

The FlexAID interface permits users to adjust the parameters of FlexAID in a convenient environment. The basic view of the interface contains four tabs: *Input Files*, *Target Config*, *Ligand Config*, and *Simulate*. The functionalities of each tab are reviewed in detail in the following sections. The FlexAID interface contains three menu items in the top left hand corner: *Load* and *Save* sessions (see **Note 21**), and *Show*, which turns on the display of two additional tabs: *Scoring Cfg* (scoring function configuration) and *GA Params* (genetic algorithm parameters). These tabs are needed for the *optional* sections of the following protocol.

1. Open the FlexAID interface from the Plugin menu found in the main window of PyMOL.

Plugin → *NRGsuite* → *Open FlexAID*...

2. The *Input* tab is displayed by default upon FlexAID's opening and allows configuration of the target and ligand (see **Note 22**). First, select "3b7e" from the PyMOL objects/selection drop-down menu and click "Save as target" (see **Note 20**). Then, select "ZMR" from the drop-down menu and click "Save as ligand" (see **Note 20**).
3. Click on the *Target Cfg* (target configuration) tab in the FlexAID interface window to access the parameters of the target (see **Note 23**). After selecting a target and a ligand, the definition of the binding site is the third mandatory input to perform a simulation. In the "Binding-site definition" upper box, under "Choose binding-site type," click on "CLEFT" and then click on the button "Import clefts." Select the first cleft which should be named "3b7e_sph_1" (see **Note 24**). A new object named "BINDINGSITE_AREA_" will be created in the PyMOL viewer and the red cleft will appear with the mesh, a grid like, representation.

(*Optional*) Although there is no need to consider target flexibility for the docking of zanamivir into the neuraminidase, it is crucial to keep in mind that the docking of any ligand that has not been crystallized with the target might benefit from the defining of key residues on the target in the binding site that interacts with the ligand as flexible. Even when flexibility is not critically needed to retrieve the binding mode of reference, the accuracy of the simulation may improve when highly mobile side chains can change conformations (see **Note 25**). There are two options to include flexibility in the target in the "Side-chain flexibility" section of the "Target flexibility" box: (1) the "Add/Delete flexible side-chains" wizard button, which allows the interactive selection of specific side-chains by the user in the PyMOL viewer, and (2) the "Residue code" text entry form, which quickly activates the flexibility for a given residue of the target. In our current case, the key

residues ARG152A, GLU276A, ARG118A, ARG292A, and ASP151A (*see Note 15*) specifically contribute to bind zanamivir (*see Note 26*).

4. Click on the *Ligand Cfg* (ligand configuration) tab in the FlexAID interface window to access the parameters of the ligand. The functionalities found in this tab include: the inclusion/exclusion of the ligand's degrees of freedom (rotational, translational, and flexible bonds' dihedrals), the computation of the RMSD compared to the reference's position, and the imposition of geometric constraints between a specific pair of atoms from the target and ligand, which can be useful to simulate covalent docking.

(*Optional*) The “Degrees of freedom” box found on the left side of the interface controls the degrees of freedom that can be introduced for the zanamivir ligand. Ligand translational and rotational degrees of freedom are enabled by default and it allows FlexAID to translate and rotate the ligand anywhere in the binding site where the anchor atom of the ligand can be placed (*see Note 27*).

(*Optional*) In the recent version of the NRGsuite, all the rotatable bonds are considered flexible by default. However, the “Ligand flexibility” section includes a “Add/Delete flexible bonds” button that opens a PyMOL Wizard (*see Note 18*) that displays all the flexible bonds within the ligand using a ball-and-stick representation: the ones selected as flexible are shown in white while the ones considered rigid are shown in orange (*see Notes 28 and 29*).

Check the “RMSD structure” checkbox found next to the right in the interface window (show above) to order FlexAID to compute the RMSD between the position of the atoms of the ligand in the reference (zanamivir position in the 3b7e PDB entry) and its predictions. This measure is used to determine whether FlexAID's predictions are successful or not in cases where the known experimental binding pose is known (*see Note 8*).

(*Optional*) The “Constraints” section located directly below the “RMSD structure” box allows FlexAID to specify a desired distance between two atoms to drive the optimization during the genetic algorithm. There are two types of constraints that can be added: intramolecular constraints (the specific distance between two atoms of the ligand itself or two atoms of the target itself) and intermolecular constraints (the specific distance between one atom of the target with one atom of the ligand). This is useful to simulate the covalent binding of an atom of the ligand to an atom of the target with the exact distance of a given bond length. The “Add/Delete constraints” button opens a PyMOL wizard allowing you to select two atoms to constrain. Once two atoms are selected,

the distance slider allows the user to set the desired distance for a given constraint. It is possible to sequentially add multiple constraints in the same manner.

5. (*Optional*) The *Scoring Cfg* (scoring function configuration) tab allows you to change the inclusion/exclusion of atoms in the scoring, the inclusion of water molecules, the definition of the solvent, and the permeability of soft-docking. Unless you are interested in including water molecules from the crystal in the simulation, there is no need to modify this section. Including water molecules will tell FlexAID to assign the oxygen atoms in the crystal to the one of water and will include it in its evaluation. Check the “Include water molecules” checkbox of the “HET groups” section to consider water molecules (*see Note 30*). The checkbox “Exclude bound molecules” will, once checked, exclude any HET atom that is part of the target structure, which is useful if one wants to remove small organic molecules used during crystallization.
6. (*Optional*) The *GA Params* (genetic algorithm parameters) tab allows you to parametrize the number of energy evaluations (length of simulations), the genetic algorithm operator, and the visual display of the simulation in the PyMOL viewer window. FlexAID uses a genetic algorithm to cover the search space, which uses an initial number of chromosomes in the population that will evolve for several generations. By default, there are 500 chromosomes that will evolve during 500 generations leading to 250,000 energy evaluations. These values can be modified in the “Genetic parameters” section of the *GA Params* tab (*see Note 31*).

(*Optional*) It is possible to modify the visual display of real-time docking in the PyMOL viewer with the corresponding section in the *GA Params* tab. The “Number of TOP complexes” indicate the number of complexes that are displayed in the PyMOL viewer. The TOP complexes shown are the ones with the lowest energy according to the scoring function. The “Refresh interval” represents the interval, in number of generations, at which the display of TOP complexes is refreshed. The progress of the genetic algorithm is shown in the Simulate tab (*see step 7*). The FlexAID algorithm waits for the NRGsuite to display the TOP complexes before processing the next generation. Thus, we strongly suggest not displaying too many TOP complexes or too short refresh intervals as it may drastically increase the computational runtime for a simulation.

7. Click on the *Simulate* tab to initiate the docking simulation. To start a docking simulation, simply click the “Start” button. Once a simulation has successfully started, you have access to the “Pause,” “Stop,” and “Abort” buttons. The “Pause”

button allows you to pause the simulation while looking at real-time results in the PyMOL viewer. The “Stop” and “Abort” buttons both end the simulation. However, stopping a simulation generates final results while aborting does not. Stopping a simulation can be particularly useful when the docking optimization has converged already to its minimum energy. FlexAID also offers a way to continue a simulation from previously generated results. When results are activated, the “Continue” button may be clickable depending on the context (*see Note 32*). It is possible to see these buttons at the top left corner of the *Simulate* tab of the FlexAID interface.

Click on the “Start” button to initiate the simulation. When the genetic algorithm is under way, FlexAID updates the values of fitness and scoring functions for the best individuals in terms of scoring function. During the simulation, the PyMOL viewer is updated with reserved objects (named TOP_<X>__, where X is the Xth result predicted by FlexAID) displaying the actual best results evaluated by FlexAID. These objects are temporary and color-coded according to the “Color” column shown in the FlexAID *Simulate* tab (*see* the interface panel at the bottom right of Fig. 1). The NRGsuite offers a quick way of changing the display of TOP_<X>__ complexes objects all at once instead of having to manually update the display for the newly created objects. By default, the target is displayed as “Cartoon” and the ligand as “Sticks.” You can change these parameters as you like in the “Display options of TOP objects” in the top right corner of the *Simulate* tab of FlexAID. Whenever a reference is used (*see step 4* above), the “Last RMSD” column is updated at every refresh interval (*see step 6* above about the *GA Params* tab).

When a simulation is finished, or stopped, the reserved objects RESULT_<X>__ and RESULT_<X>_H_BONDS__ are created in the PyMOL viewer. RESULT_<X>__ are the complex’s objects while the RESULT_<X>_H_BONDS__ objects are distances objects displaying the h-bonds for the result. Their according energy value (referenced as CF) and RMSD from the reference (when one is used) of the results are displayed in the table shown at the bottom right of Fig. 1.

In the case where constraints are applied, an extra column, Apparent CF, will appear which is a modified value of the CF to reflect whether the predicted pose respects the desired constraints imposed by the user.

8. *Analysis*—Finally, look at the RESULT__X objects RMSD in the last column of the FlexAID *Simulate* results table. Remember that this docking experiment serves as control with which further docking experiments, e.g., different small molecules, will be compared. FlexAID can retrieve RMSDs below 2.0 Å on

this specific case easily when this protocol is used (*see Note 33*). Both the specific interactions (H_BONDS in the PyMOL viewer objects) and the CF score can be used to compare different poses or different ligands (this should be used with caution with ligands of very different in size) during your next experiments.

4 Notes

1. PyMOL is freely distributed as source code or precompiled binaries for academic users at <http://pymol.org/educational>. For the best results, please make sure to use the latest PyMOL version available for your system.

- (a) As a side note, PyMOL uses its own version of Python to function and the NRGsuite requires at least Python version 2.5 to work properly. Therefore, PyMOL versions 1.0/1.1 are currently unsupported by the NRGsuite.

2. The NRGsuite is available at <http://biophys.umontreal.ca/nrg> as package installers for Windows and macOS and in the form of an install script for Linux. At the time of this writing, the version of the NRGsuite is 2.48j, but it is recommended to use the latest version available on our website.

All software employed in the protocols are free at least for nonprofit users. FlexAID and the NRGsuite are free for everyone and distributed under the GNU General Public License. The NRGsuite is available for Linux (32/64-bit), macOS, and Windows (32/64-bit).

- (a) The suite has been extensively tested on Linux 32/64-bit, Windows 32/64-bit, and macOS 64-bit machines with free-for-academic PyMOL versions 1.2/1.3 as well as PyMOL versions 1.6/1.7/1.8.

3. This is particularly important for operating environments running under Windows, where you need to select either the 32-bit or 64-bit version. It is strongly suggested to install the 64-bit version if you run Windows 7 or any more recent distribution. Here is how to install the NRGsuite for your system:

Windows

Double-click the NRGsuite_2.48i_Win<32/64>.exe installer and install. The default location is “C:\Program Files”.

macOS/Mac OS X

Double-click the NRGsuite_2.48i_MacOSX64.pkg. installer and install the package in its default location. The default location is “/Applications/NRGsuite”.

Linux

Open up a terminal and move to the directory in which you downloaded the archive `NRGsuite_2.48i_Linux<32/64>.tar`, and execute the following commands:

```
> tar -xvf NRGsuite_2.48i_Linux<32/64>.tar
> sh install.sh NRGsuite.tar.gz
```

The default location is “`/usr/local/NRGsuite`”. Therefore, superuser privileges (`sudo`) are required for installing the NRGsuite. To override the default location, open `install_linux.sh` with any text editor and change the value of `INSTALL_PATH` with the desired location.

4. Open PyMOL and in the upper menu click the following buttons:

Plugin → *Manage Plugins* → *Install...*

Browse to the directory in which you installed the NRGsuite on your hard drive (*see step A*), and double-click the following file:

NRGsuite → *Plugin* → *NRGsuite.py*

Restart PyMOL and the NRGsuite menu should appear in the Plugin menu:

Plugin → *NRGsuite*

You are now ready to use the NRGsuite.

5. In structural biology, the resolution of a structure is a measure of the quality of the data collected. As the resolution of a structure gets smaller, the confidence in the location of atoms is higher. A resolution of 2.0 Å and below is generally considered as “high resolution” for a structure in that the position of all heavy atoms (i.e., nonhydrogen atoms) is well established.
6. All of the publicly available experimental structures of biological complexes can be parsed and obtained directly from the Protein Data Bank (PDB) at <http://rcsb.org>. There are multiple ways to parse the PDB and the user can search using specific criteria, e.g., the desired resolution, a specific drug, or the experimental method used to solve the structure. Statistically and historically speaking, the process of drug discovery mostly involved structures solved using X-ray crystallography because of its compatibility with different families of proteins and its better resolution, but structures solved by other methods with high resolution are sometimes available and can be used for molecular docking.
7. In molecular docking, a pose predicted with a root-mean squared distance (RMSD) below 2.0 Å from the experimental pose of reference is considered a success. The resolution of the reference structure has a significant impact on the accuracy of the predictions. For this reason, it is suggested to choose a “high resolution” structure solved at 2.0 Å or lower when it is available.

8. The NRGsuite is developed to allow molecular docking simulations involving any biological target. Although this tutorial will use a protein target to predict the binding mode of the drug of interest, it could be replaced by any other protein or nucleic acid target for which a tridimensional structure is readily available.
9. It is easy to identify all available structures by querying the PDB using the “Zanamivir” keyword and then refining by molecular target and organism. Subsequently, it is possible to sort the structure by resolution in order to choose the best available resolution.
10. The structure information page can be consulted at: <http://www.rcsb.org/pdb/explore/explore.do?structureId=3B7E>. The PDB file can be found at: <http://files.rcsb.org/view/3B7E.pdb>.
11. The directory-tree “Documents/NRGsuite” is automatically created in the home folder upon initialization of the NRGsuite plugin in PyMOL. This folder represents the default location in which the different projects of \$USER are held (where \$USER refers to the currently logged-in user on the system). Depending on the operating system, this folder is located at the following location in your file system:

Under Windows

Vista and higher: “C:\Users\%USER%\Documents\NRGsuite”

Older: “C:\Documents and Settings\%USER%\Documents\NRGsuite”

****Do not confuse “Documents” with “My Documents”**

Under macOS or Mac OS X

“/Users/%USER%/Documents/NRGsuite”

Under Linux

“/home/%USER%/Documents/NRGsuite”

12. PyMOL has two windows: one that is named “PyMOL Molecular Graphics System” and is referred to as the main window. The second window is the “PyMOL Viewer” and will be referred as such.
13. It is preferred to *avoid using whitespaces in any filenames that will be used with the NRGsuite*. Although most of the configurations will allow, it is known to cause bugs on some operating systems, especially for Windows users.
14. GetCleft detects the cavities of a target by introducing spheres between pairs of target atoms and reducing the radii of the spheres until there are no clashes. The radius range can be used to control the volume of generated clefts. By default, the range goes from a Minimum of 1.50 Å to a Maximum 4.00 Å. Reducing the minimum radius allows one to insert spheres in

deeply buried cavities. Thus, clefts generated with a lower minimum are larger.

15. The NRGsuite uses a string representation to identify unique residues by their residue name (written using the three capital letters form), a residue number and the residue chain formatted as *RESIA*. For example, the first methionine, numbered 1, from the chain A will be formatted *MET1A*.
16. The list of PyMOL objects/selections shows every object found in the PyMOL viewer window. If your object does not appear in the list, hit the refresh button on the right.

Some objects will not be shown in the list, e.g., objects which name contains a double underscore “__” (reserved for Python and PyMOL only) and objects whose names contains an extension “*.*”.

17. Each cleft object is named *\$TARGET_sph_X* where *\$TARGET* is the name of the target (3b7e in this protocol) and X is a numerical value from 1 to the maximum number of clefts desired by the user.
18. Through the FlexAID and GetCleft interfaces, you will eventually need to work with Wizards. Upon activation of a Wizard, the interface that called the Wizard will be locked and you will need to interact with the PyMOL viewer instead. Each Wizard is unique and waits for a different input from the user. The type of input is clearly written on top of the PyMOL viewer.

To accommodate the different types of input as well as to facilitate working with the Wizard, activating a Wizard may alter:

- The selection mode (“Selecting”)
- The mouse mode (“Mouse Mode”)
- The PyMOL view
- The masking of your objects (un-clickable)

However, these will be reset to your original values once the Wizard turns inactive. The user can also interact with the Wizard menu on the bottom right corner of the PyMOL viewer. The Wizard will remain active until the user clicks the “Done” button in the menu.

19. It is possible to add multiple individual spheres to accurately define the binding site. Whenever a second partition sphere is added, it is also appended to the sphere objects list. To help you distinguish which sphere corresponds to which in the PyMOL viewer, you can refer to the *SPHERE_PT_AREA__* object. The object is shown as a transparent violet sphere and refers to the currently selected sphere in the sphere objects list.

20. The NRGsuite has been designed to load and save files in the project directory initially defined by the user upon installation. Therefore, it is best if the user save and load files at the suggested location whenever any file window pops up.
21. A session within the FlexAID interface allows you to store and retrieve the content of all tabs (*see* Subheading 3.3). The content of a session includes:

- *file references pointing to the target, the ligand*
- *the binding-site definition*
- *the target flexibility*
- *the results of a simulation*
- *values in each of the tabs*

Sessions are particularly useful when you want to reproduce a simulation multiple times with the same parameters. ****Do not confuse a FlexAID interface session with a PyMOL session (using the file extension *.pse*).**

22. There are two ways of importing the target: the first one involves the PyMOL object/selection drop-down menu that allows the selection of an object in PyMOL, while the second allows you to load previously used targets in your project from the Load button located to the right of the “THE TARGET” menu box.

There are three ways of importing the ligand: the first one involves the drop-down menu that allows the selection of a PyMOL object/selection, the second one uses the *Load* button found to the right of the “*THE LIGAND*” menu box and the last uses the *Input* button to generate a 3D structure of your ligand from a SMILES [35] representation. Please note that the “Generate 3D conformation” checkbox must be checked to generate a 3D structure of your ligand. Finally, the *Anchor* button allows the definition of the anchor atom in a ligand (*see Note 27*).

23. When the user activates another tab from the Input tab, the ligand and the target must be processed. This process may take a few seconds to a minute depending on the size of the molecules. This process is necessary to derive (1) the atom types of the molecules, (2) the flexible bonds of the ligand, and (3) to build the necessary input files that are required for executing a docking simulation with FlexAID.

The processing of the ligand creates a new object called LIGAND__ (reserved name *see Note 16*) with newly generated coordinates when generating a 3D structure.

The processing of the target creates the object TARGET__.

24. There are cases where multiple individual clefts need to be used to define the binding site, e.g., when the active site is

undetermined or when multiple allosteric sites need to be queried altogether. It is possible to import multiple clefts at once by selecting multiple cleft files while holding “Shift” (selects all files from first to last) or “Control” (add the clicked file to the list of currently selected ones) key. Once you have selected your clefts, click the “Open” button to import them. “Delete others” will delete all clefts except for the active cleft.

Once imported, the clefts will appear in the clefts list. The active cleft is highlighted in red in the PyMOL viewer while the others are blue-violet. It is important to note that docking of the ligand will take place in all clefts in the list and not only the active one. You can manipulate the clefts by using the buttons in the “CLEFT” box. “Clear” will delete all clefts in the clefts list. “Delete” will delete the active cleft.

25. The frequency of side-chain rearrangements correlates with their entropy; thus the most entropic side-chains often benefit from flexibility during a docking simulation. Studying the frequency of side-chain rearrangements can help you determine which ones have the greatest probability of changing conformation upon binding [19].
26. A quick way to identify which residues interact with the ligand in the chosen complex is to refer to the “Poseview Image” of the ligand of interest in the PDB (*see Note 6*). This is found in the “Small Molecules” section of a given PDB entry. In the “2D Diagram & Interaction,” a visual representation of the ligand is shown along with the residues of the target that contribute to binding.

An alternative method to find these residues for a target that is not found in the PDB, like a homology model, would be to look at the most conserved residues in a multiple sequence alignment of proteins from the same Pfam [36] family.

27. The anchor atom is the first atom to be placed in the binding site and is the only atom that absolutely needs to be inside the volume of the cleft(s) used for the molecular docking experiment.
28. When a ligand is imported, the molecule is considered rigid during docking unless the user explicitly includes ligand flexibility. To include ligand flexibility, click the “Add/Delete flexible bonds” button. Your ligand will appear as the object FLEXIBLE_LIGAND__.

The following objects are also created (do not interact with these PyMOL objects):

POSS_FLEX_BONDS__ (to display the possible flexible bonds of the ligand) SELECTED_BONDS__ (to display the selected flexible bonds of the user).

A wizard will open waiting for you to click on the two atoms that define the flexible bond. Flexible bonds appear in

orange. Clicking on the two atoms that define the flexible bond will render this bond as flexible during the docking simulation. The selected bonds of the user appear in white.

Rather than selecting all bonds manually, you can quickly select all flexible bonds by clicking the “Select all flexible bonds” button in the Wizard menu.

The “Clear flexible bonds” button removes all selected flexible bonds.

When the Wizard gets inactivated, the number of selected flexible bonds will be shown in the interface.

29. At the time of the writing, the latest version of PyMOL running on Windows 10 contains an unresolved bug where PyMOL crashes when the user tries to open the “Add/Delete flexible bonds” wizard of the *Ligand Cfg* tab in the FlexAID interface. For this reason, all rotatable bonds of the ligand are now enabled by default in the *Preferences* panel of the NRGsuite found in the *Plugin* → *NRGsuite* → *Preferences* menu item of the PyMOL main window. Please note that this bug will be fixed in a future version of the NRGsuite available at our website (*see Note 2*).
30. Most of the time, including all water molecules when docking the reference during the control experiment will ease the docking problem because water molecules will fill most of the binding site and the ligand could be fit using geometric criteria. To avoid this, it is suggested to keep only the strongly bound water molecules that directly interact in the binding site. The B-factor column of the PDB file could also be a good indicator of which water molecules must be kept; a low B-factor signifies that this atom is well ordered and there is not a large uncertainty on its position. It is suggested to keep the water molecules with the lowest B-factors as well as those who directly interact with the ligand in the reference.
31. The search space is increased when you include additional degrees of freedom in the optimization (e.g., flexible bonds of the ligand flexible and/or flexible side-chains). With these additional degrees of freedom, more energy evaluations are required to converge to a minimum of energy. When the control simulation is not able to retrieve the conformation of reference, you can increase the number of chromosomes and generations by increasing the corresponding values. It is suggested to initially simulate using default ones.
32. You can continue a simulation if and only if the active results were generated with the same parameterization, i.e., the same target and ligand, binding-site definition, flexibility of the molecules, etc. Specifically, the content of the following tabs must be the same: *Input Files*, *Target Cfg*, *Ligand Cfg*, and

Scoring Cfg. You can modify the parameters of the genetic algorithms and still be able to continue a simulation.

33. Docking is an optimization problem and it is possible to have a bad initial population for a simulation (which is randomly generated) or it is also possible that the optimization got stuck in a local minimum of energy. Sometimes, simply repeating the experiment gets you better results.

Acknowledgments

R.J.N. is part of PROTEO (the Québec network for research on protein function, structure, and engineering) and GRASP (Groupe de Recherche Axé sur la Structure des Protéines). The authors would like to thank the users of FlexAID and the NRGsuite for numerous bug reports and feedbacks, thus contributing to their development, and Florence Min for critical reading of the manuscript.

Funding: L.P.M. is the recipient of a Ph.D. fellowship from the Fonds de Recherche du Québec—Nature et Technologies (FRQ-NT).

References

1. Berman HM, Westbrook J, Feng Z, Gilliland G, Bhat TN, Weissig H et al (2000) The Protein Data Bank. *Nucleic Acids Res* 28:235–242
2. Najmanovich RJ (2017) Evolutionary studies of ligand binding sites in proteins. *Curr Opin Struct Biol* 45:85–90. <https://doi.org/10.1016/j.sbi.2016.11.024>
3. Gohlke H, Klebe G (2002) Approaches to the description and prediction of the binding affinity of small-molecule ligands to macromolecular receptors. *Angew Chem Int Ed* 41:2645–2676
4. Meng X-Y, Zhang H-X, Mezei M, Cui M (2011) Molecular docking: a powerful approach for structure-based drug discovery. *CAD* 7:146–157. <https://doi.org/10.2174/157340911795677602>
5. Kitchen DB, Decornez H, Furr JR, Bajorath J (2004) Docking and scoring in virtual screening for drug discovery: methods and applications. *Nat Rev Drug Discov* 3:935–949. <https://doi.org/10.1038/nrd1549>
6. Jacob RB, Andersen T, McDougal OM (2012) Accessible high-throughput virtual screening molecular docking software for students and educators. *PLoS Comput Biol* 8:e1002499. <https://doi.org/10.1371/journal.pcbi.1002499>
7. Neudert G, Klebe G (2011) DSX: a knowledge-based scoring function for the assessment of protein-ligand complexes. *J Chem Inf Model* 51:2731–2745. <https://doi.org/10.1021/ci200274q>
8. Li Y, Liu Z, Li J, Han L, Liu J, Zhao Z et al (2014) Comparative assessment of scoring functions on an updated benchmark: 1. Compilation of the test set. *J Chem Inf Model* 54:1700–1716. <https://doi.org/10.1021/ci500080q>
9. Englebienne P, Moitessier N (2009) Docking ligands into flexible and solvated macromolecules. 4. Are popular scoring functions accurate for this class of proteins? *J Chem Inf Model* 49:1568–1580. <https://doi.org/10.1021/ci8004308>
10. Grudin S, Kadukova M, Eisenbarth A, Marillet S, Cazals F (2016) Predicting binding poses and affinities for protein–ligand complexes in the 2015 D3R Grand Challenge using a physical model with a statistical parameter estimation. *J Comput Aided Mol Des* 30(9):791–804. <https://doi.org/10.1007/s10822-016-9976-2>

11. Mobley DL, Dill KA (2009) Binding of small-molecule ligands to proteins: “what you see” is not always “what you get”. *Structure* 17:489–498. <https://doi.org/10.1016/j.str.2009.02.010>
12. Wong SE, Lightstone FC (2011) Accounting for water molecules in drug design. *Expert Opin Drug Discov* 6:65–74. <https://doi.org/10.1517/17460441.2011.534452>
13. Corbeil CR, Moitessier N (2009) Docking ligands into flexible and solvated macromolecules. 3. Impact of input ligand conformation, protein flexibility, and water molecules on the accuracy of docking programs. *J Chem Inf Model* 49:997–1009. <https://doi.org/10.1021/ci8004176>
14. Limongelli V, Marinelli L, Cosconati S, La Motta C, Sartini S, Mugnaini L et al (2012) Sampling protein motion and solvent effect during ligand binding. *Proc Natl Acad Sci U S A* 109:1467–1472. <https://doi.org/10.1073/pnas.1112181108>
15. Therrien E, Weill N, Tomberg A, Corbeil CR, Lee D, Moitessier N (2014) Docking ligands into flexible and solvated macromolecules. 7. Impact of protein flexibility and water molecules on docking-based virtual screening accuracy. *J Chem Inf Model* 54:3198–3210. <https://doi.org/10.1021/ci500299h>
16. Zhao S, Goodsell DS, Olson AJ (2001) Analysis of a data set of paired uncomplexed protein structures: new metrics for side-chain flexibility and model evaluation. *Proteins Struct Funct Genet* 43:271–279
17. Boehr DD, Nussinov R, Wright PE (2009) The role of dynamic conformational ensembles in biomolecular recognition. *Nat Chem Biol* 5:789–796. <https://doi.org/10.1038/nchembio.232>
18. Najmanovich RJ, Kuttner J, Sobolev V, Edelman M (2000) Side-chain flexibility in proteins upon ligand binding. *Proteins Struct Funct Genet* 39:261–268
19. Gaudreault F, Chartier M, Najmanovich RJ (2012) Side-chain rotamer changes upon ligand binding: common, crucial, correlate with entropy and rearrange hydrogen bonding. *Bioinformatics* 28:i423–i430. <https://doi.org/10.1093/bioinformatics/bts395>
20. Gaudreault F, Najmanovich RJ (2015) FlexAID: revisiting docking on non-native-complex structures. *J Chem Inf Model* 55:1323–1336. <https://doi.org/10.1021/acs.jcim.5b00078>
21. Frappier V, Najmanovich RJ (2014) A coarse-grained elastic network atom contact model and its use in the simulation of protein dynamics and the prediction of the effect of mutations. *PLoS Comput Biol* 10:e1003569. <https://doi.org/10.1371/journal.pcbi.1003569>
22. Wang R, Fang X, Lu Y, Yang C-Y, Wang S (2005) The PDBbind database: methodologies and updates. *J Med Chem* 48:4111–4119. <https://doi.org/10.1021/jm048957q>
23. Trott O, Olson AJ (2010) AutoDock Vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. *J Comput Chem* 31:455–461. <https://doi.org/10.1002/jcc.21334>
24. Rarey M, Kramer B, Lengauer T, Klebe G (1996) A fast flexible docking method using an incremental construction algorithm. *J Mol Biol* 261:470–489. <https://doi.org/10.1006/jmbi.1996.0477>
25. Ruiz-Carmona S, Alvarez-Garcia D, Foloppe N, Garmendia-Doval AB, Juhos S, Schmidtke P et al (2014) rDock: a fast, versatile and open source program for docking ligands to proteins and nucleic acids. *PLoS Comput Biol* 10(4):e1003571. <https://doi.org/10.1371/journal.pcbi.1003571>
26. Gaudreault F, Morency L-P, Najmanovich RJ (2015) NRGsuite: a PyMOL plugin to perform docking simulations in real time using FlexAID. *Bioinformatics* 31:3856–3858. <https://doi.org/10.1093/bioinformatics/btv458>
27. Letourneau D, Lorin A, Lefebvre A, Frappier V, Gaudreault F, Najmanovich R et al (2012) StAR-related lipid transfer domain protein 5 binds primary bile acids. *J Lipid Res* 53(12):2677–2689. <https://doi.org/10.1194/jlr.M031245>
28. Duchêne D, Colombo E, Désilets A, Boudreault P-L, Leduc R, Marsault E et al (2014) Analysis of subpocket selectivity and identification of potent selective inhibitors for matrilipase and matriptase-2. *J Med Chem* 57:10198–10204. <https://doi.org/10.1021/jm5015633>
29. Chartier M, Morency L-P, Zylber MI, Najmanovich RJ (2017) Large-scale detection of drug off-targets: hypotheses for drug repurposing and understanding side-effects. *BMC Pharmacol Toxicol* 18:1046. <https://doi.org/10.1186/s40360-017-0128-7>
30. Seto JT, Rott R (1966) Functional significance of sialidase during influenza virus multiplication. *Virology* 30:731–737
31. Moscona A (2005) Neuraminidase inhibitors for influenza. *N Engl J Med* 353:1363–1373. <https://doi.org/10.1056/NEJMra050740>

32. Varghese JN, Laver WG, Colman PM (1983) Structure of the influenza virus glycoprotein antigen neuraminidase at 2.9 Å resolution. *Nature* 303:35–40. <https://doi.org/10.1038/303035a0>
33. Itzstein von M, Wu WY, Kok GB, Pegg MS, Dyason JC, Jin B et al (1993) Rational design of potent sialidase-based inhibitors of influenza virus replication. *Nature* 363:418–423. <https://doi.org/10.1038/363418a0>
34. Talele T, Khedkar S, Rigby A (2010) Successful applications of computer aided drug discovery: moving drugs from concept to the clinic. *Curr Top Med Chem* 10:127–141. <https://doi.org/10.2174/156802610790232251>
35. O’Boyle NM (2012) Towards a universal SMILES representation – a standard method to generate canonical SMILES based on the InChI. *J Chem* 4:22. <https://doi.org/10.1186/1758-2946-4-22>
36. Finn RD, Bateman A, Clements J, Coggill P, Eberhardt RY, Eddy SR et al (2014) Pfam: the protein families database. *Nucleic Acids Res* 42:D222–D230. <https://doi.org/10.1093/nar/gkt1223>