The
NRGsuite Guide

Written by Francis Gaudreault
Last edited on May 22, 2015

Table of Contents

1. The NRGsuite

1.1. Getting started

The NRGsuite is a suite of tools developed in the laboratory of Dr. Rafael Najmanovich in Université de Sherbrooke (Québec).

The NRGsuite, FlexAID and Process_Ligand are developed and maintained by Francis Gaudreault and Louis-Philippe Morency. The GetCleft algorithm was developed by Rafael Najmanovich but maintained by Francis Gaudreault. The volume_calc algorithm was developed by Dominic Duchêne.

This guide was written by Francis Gaudreault.

Special thanks to:
- Eugène Morin for great technical assistance
- Félix Morency for good technical tips and advices
- Pierre Lavigne for agree-ing on using the NRGsuite as an academic tool
- Students of the PHR608 course in Université de Sherbrooke for contributing in making the software better through the years.

Compatibility

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.

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. Moreover, the NRGsuite is also incompatible on Linux 32-bit with the educational and pre-compiled PyMOL build v1.3r1.

Make sure you use the latest version as listed on our website:
http://bcb.med.usherbrooke.ca/FlexAID

If you wish to view the version you are currently using, open the About menu (see section 1.9).

1.2. Installing the NRGsuite

The NRGsuite comes with pre-compiled executables, so make sure you downloaded the package for the right operating system.

Installing the NRGsuite package requires administrator rights and consists in two steps A and B.

1.2.1. Installing the NRGsuite on your hard drive

Under Windows

Double-click the NRGsuite_Win<32/64>.exe installer and install. The default location is "C:\Program Files".

Under MacOSX

Double-click the NRGsuite_MacOSX<4+/5+>.pkg installer and install the package in its default location. The default location is "/Applications/NRGsuite".

Under Linux

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

tar -xvf NRGsuite_Linux<XX>.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.

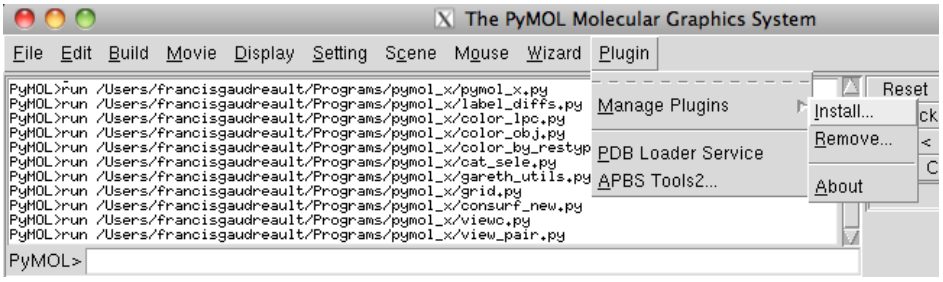
1.2.2. Installing the NRGsuite in PyMOL

*** The following step is only required on MacOSX machines.*

By default, PyMOL on MacOSX is run by clicking the application "MacPyMOL". However, you need to rename "MacPyMOL" with "PyMOLX11Hybrid" considering the X11 interface is required for the following steps.

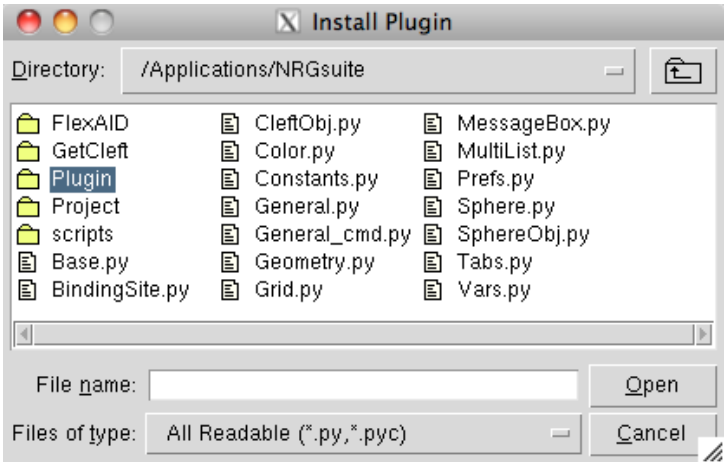
Open PyMOL and in the upper menu click the following buttons:

Plugin -> Manage Plugins -> Install…



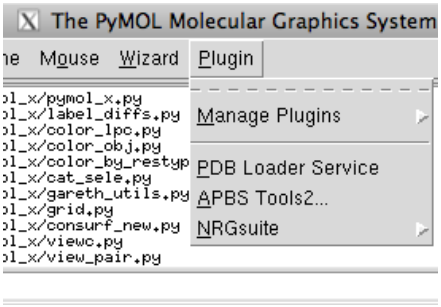
Browse to the directory in which you installed the NRGsuite on your hard drive (see step A), and double-click the following file:

NRGsuite -> Plugin -> NRGsuite.py



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

Plugin -> NRGsuite



You are now ready to work with the NRGsuite.

1.3. The project environment

The work environment of the NRGsuite is project-oriented. A project has to be activated in order to access the two main interfaces FlexAID and GetCleft.

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:

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 MacOSX

"/Users/$USER/Documents/NRGsuite"

Under Linux

"/home/$USER/Documents/NRGsuite"

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.

1.3.1. The hierarchy of a project

The NRGsuite uses a fixed hierarchy of folders to manage the content of a project:

Project/
- *Cleft/*
- *FlexAID/*
  - *.Save/*
  - *.Temp/*
  - *Binding_Site/*
  - *Ligand/*
  - *Results/*
  - *Session/*
  - *Simulation/*
  - *Target_Flexibility/*
- *GetCleft/*
  - *.Save/*
  - *.Temp/*
- *Target/*

The *Cleft/* folder contains the cleft objects. A subfolder is created for each target. Cleft objects use the extension *.nrgclf* for "Najmanovich Research Group Cleft".

The *FlexAID/Binding_Site/* folder contains the binding-site objects. A subfolder is created for each target. Binding-site objects use the *.nrgbs* for "Najmanovich Research Group Binding-Site".

The *FlexAID/Ligand/* folder contains the ligand files. The supported ligand file formats that can be processed are the following: PDB, MOL, MOL2, SDF and SMI.

The *FlexAID/Results/* folder contains the results objects. A subfolder is created for each complex (the target-ligand combination). Results objects use the extension *.nrgfr* for "Najmanovich Research Group FlexAID Results".

The *FlexAID/Results/* folder contains the session objects. Session objects use the extension *.nrgfs* for "Najmanovich Research Group FlexAID Session".

The *FlexAID/.Save/ and FlexAID/.Temp/* folders contain the session files. **The user should never edit the content of that folder.**

The *FlexAID/Simulation/* folder contains the input and output files of docking simulations. A subfolder is created for each complex from which subfolders are created using the timestamp of simulations. **The user should never edit the content of that folder.**

The *FlexAID/Target_Flexibility/* folder contains the target flexibility objects. A subfolder is created for each target. Target flexibility objects use the extension *.nrgtf* for "Najmanovich Research Group Target Flexibility".

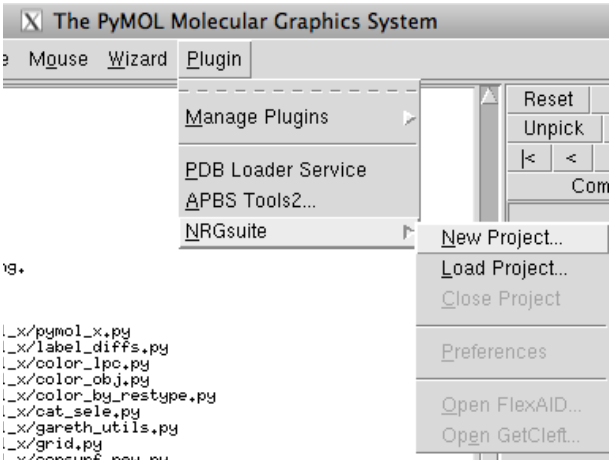
The *GetCleft/.Save/* and *GetCleft/.Temp/* folders contain the files pointed by un-saved and saved cleft objects. **The user should never edit the content of that folder.**

The *Target/* folder contains the target files. The only supported format is PDB.

1.3.2. Creating a new project

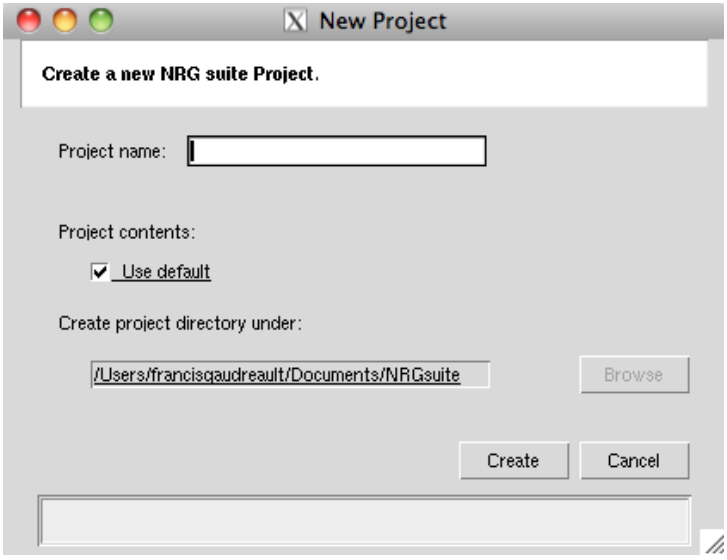
The create project interface is accessible by clicking the following button in the NRGsuite menu:

Plugin -> NRGsuite -> New Project



Name the project in the “Project name” section and click the “Create” button.
By default, a project is saved at its default location (see Section 1.3), but you can override the default location by unchecking “Use default” and clicking the “Browse” button to select your directory of choice.

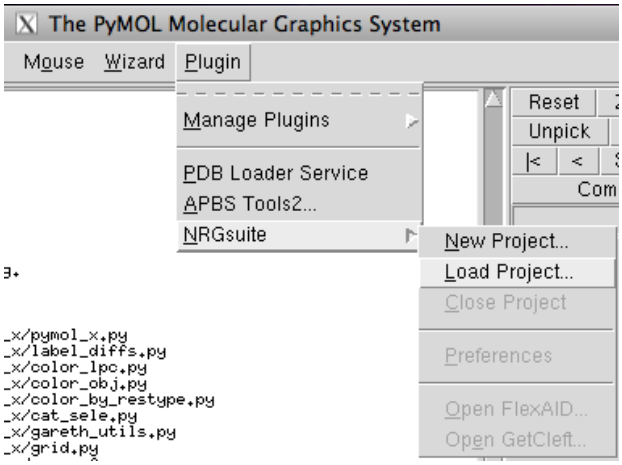
The project will automatically be loaded when created.



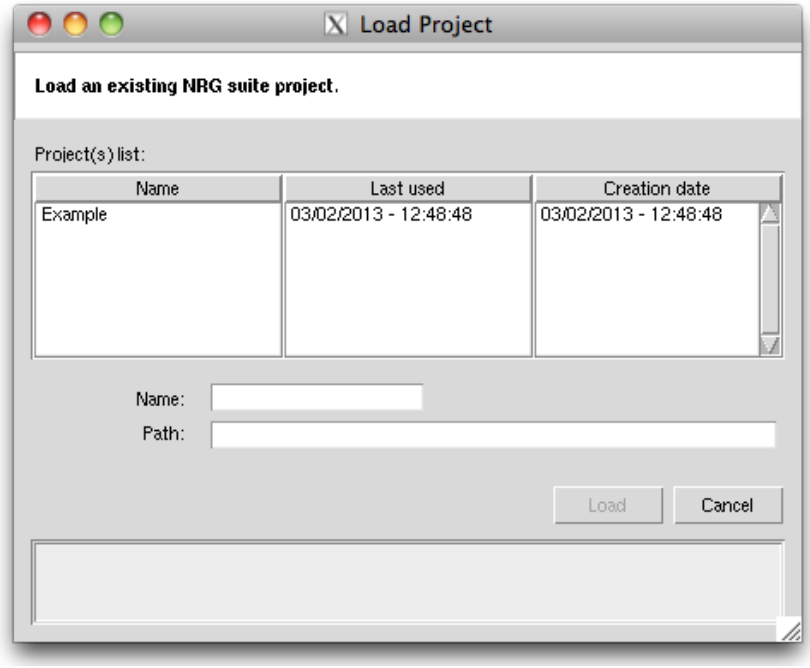
1.3.3. Loading an existing project

The load project interface is accessible by clicking the following button in the NRGsuite menu:

Plugin -> NRGsuite -> Load Project

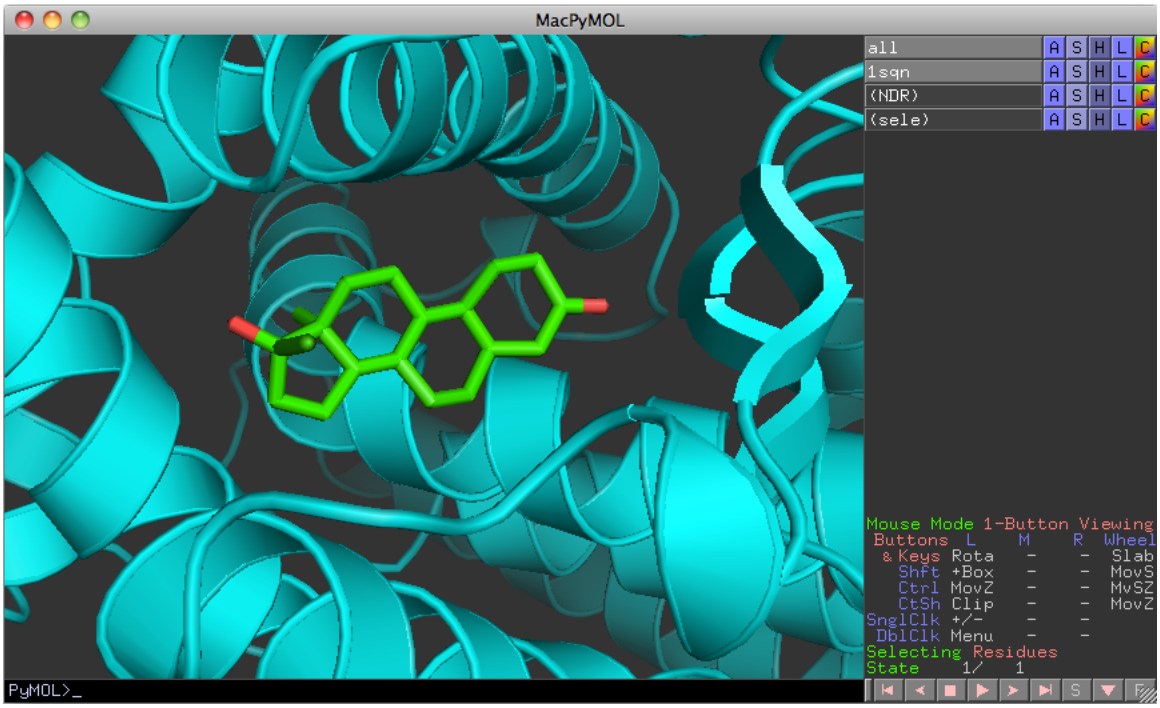


The projects of the currently logged-in user are listed in order of last time used. To load a project, simply click on the project in the list and click the "Load" button. A default project called "Example" is created the first time the NRGsuite is initialized.



1.3.3. The example project

Throughout this guide, we use as example the PDB 1SQN. The crystal structure represents the Ligand binding domain of the Progesterone Receptor (cyan) bound to an agonist called Norethindrone (green).



The residue code name of Norethindrone in 1SQN is NDR. We use that code to refer to the ligand.

1.4. Working with the FlexAID and GetCleft interfaces

Once an object has been imported into an interface do not delete, rename or modify it in the PyMOL viewer, as there is no way of tracking such events. Otherwise, such events may results in unexpected errors.

Do not delete/rename/alter/modify reserved objects created in the PyMOL viewer by the NRGsuite (capitalized and ending with '__'). You can disable/enable these objects to make them appear/disappear as you like. These reserved objects are the following:

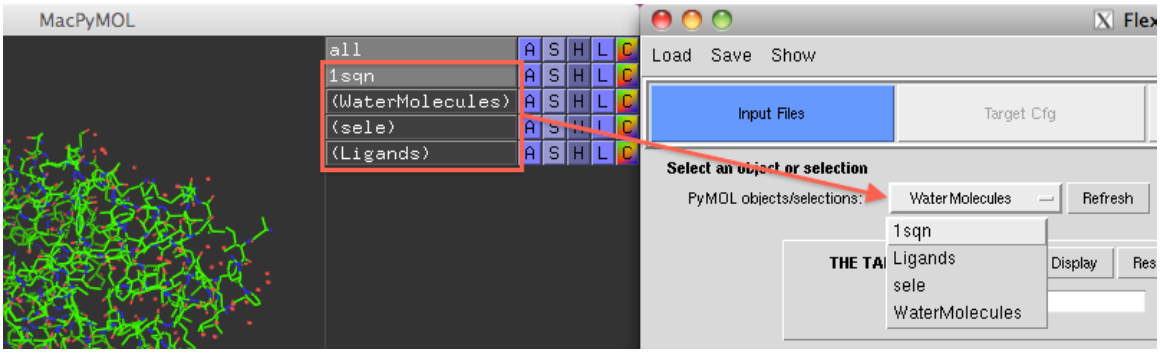
- LIGAND__
- TARGET__
- ANCHOR_LIGAND__
- FLEXIBLE_LIGAND__
- POSS_FLEX_BONDS__
- HIGHLIGHT_ATOM__
- SELECTED_BONDS__
- FLEXIBLE_SIDE_CHAINS__
- HIGHLIGHT_RESIDUE__
- CONSTRAINT_LIGAND__
- ACTIVE_CONS__
- CONS_<X>__
- DISTANCE_OBJECT__
- SPHERE__
- BINDINGSITE_AREA__
- TOP_<X>__
- RESULT_<X>__
- RESULT_<X>_H_BONDS__
- PARTITION_AREA__
- SPHERE_PT_AREA__

, where <X> is a numerical value

Do not name one of your objects with a reserved object name.

1.5. The object/selection widget

The object/selection widget lists the objects and selections (appearing with parentheses) available from the PyMOL viewer (on the left). The widget allows you to interact with the FlexAID (on the right) and GetCleft interfaces with molecules from the PyMOL viewer.



If an object does not appear in the widget, click the "Refresh" button to refresh the list of object and selections from PyMOL.

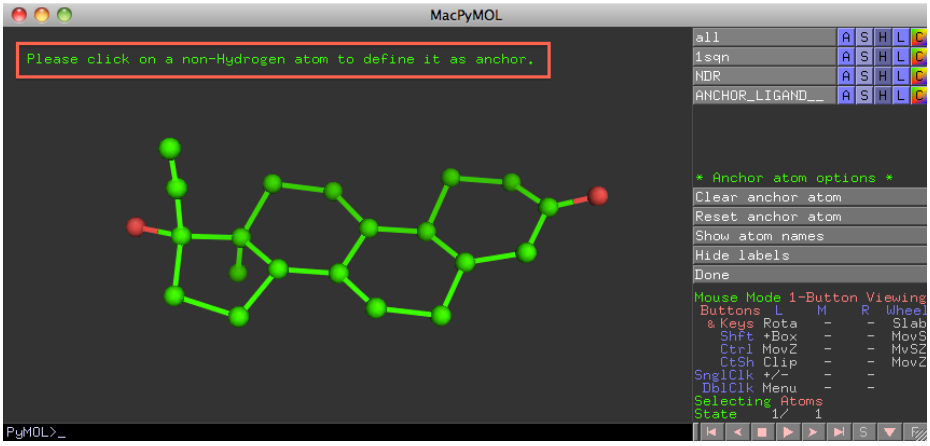
Some objects may not appear in the widget such as:

- *clefts objects containing the tag "_sph_"*
- *reserved objects used by the NRGsuite (see section 1.4)*
- *PyMOL picking selections*

1.6. The Wizards

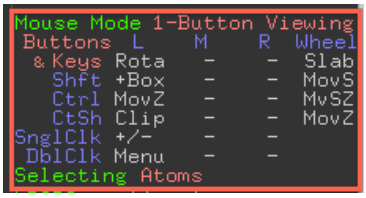
Through the FlexAID and GetCleft interfaces, you will eventually need to work with Wizards. Upon activation of a Wizard, the interface that called the Wizard will be locked and you will need to interact with the PyMOL viewer instead.

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.



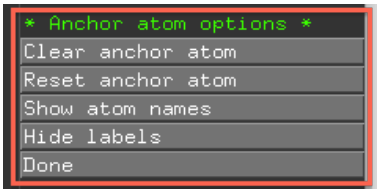
In order to accommodate the different types of input as well as to facilitate working with the Wizard, activating a Wizard may alter:

- 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 right.



The Wizard will remain active until the user clicks the “Done” button in the menu.

1.7. The Validators

Through the FlexAID or GetCleft interfaces, the user may be asked to input numerical or literal values.

The input is validated as the user modifies the value. You cannot go further through the interface whenever an error is tracked (appears as red). The error is described in the message box of the interface.

Errors may vary depending on the input of the user.

When a literal value has to be inputted, only the following characters can be used:

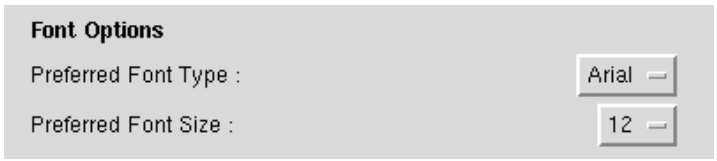
- Alpha-numeric characters (A-Z, a-z and 0-9)
- The following signs: _ - .

When the literal value is the name of a PyMOL object, the value has to begin with an alpha-numerical character and must not contain any spaces.

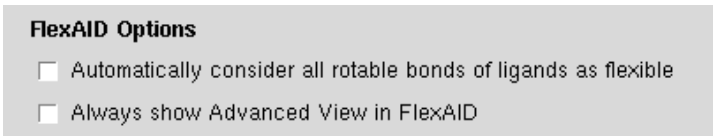
1.8. The Preferences

The Preferences menu can be accessed through the NRGsuite menu.

The NRGsuite uses a common font for all of its interfaces. Users can override the default set to it to their desired font type/size.



Users can also set options specific to the FlexAID interface.



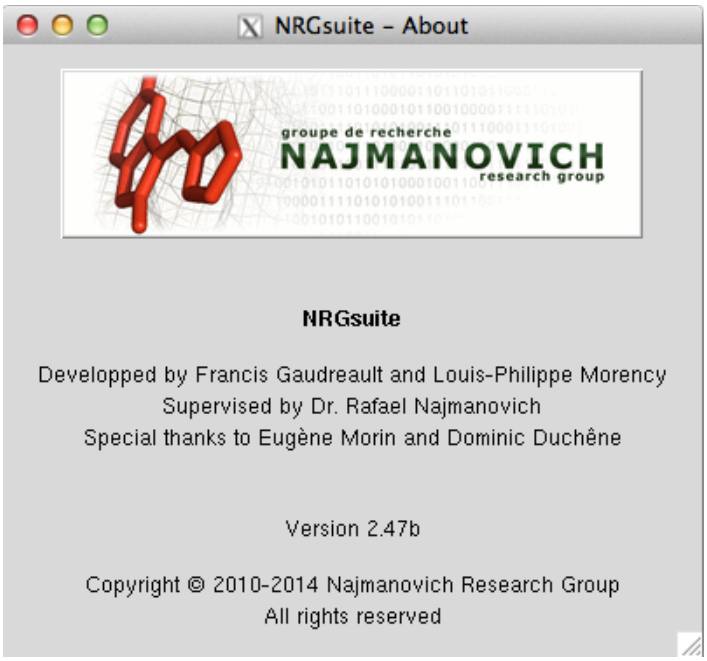
When a ligand gets processed (see section 2.3.4), all of its flexible bonds are set as rigid by default, to make them as flexible simply check the first checkbox.

By default, the advanced view of the FlexAID interface is hidden (see section 2.1). To make it visible by default check the second checkbox.

1.9. The About

The About menu can be accessed through the NRGsuite menu.

The menu displays information about the developers and the version of the NRGsuite package currently installed.

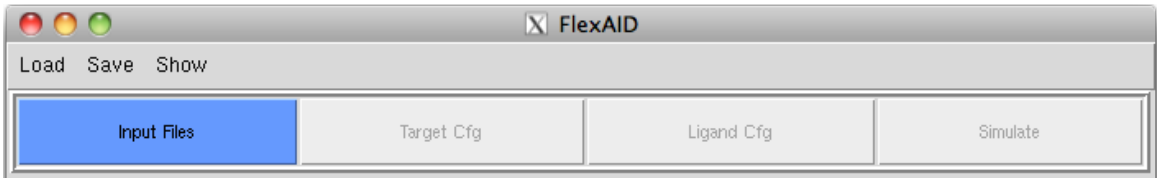


2. The FlexAID interface

2.1. Getting started

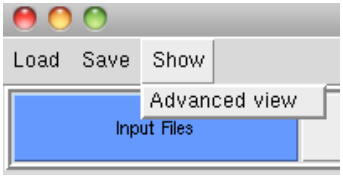
FlexAID is a docking algorithm that simulates the binding of a ligand to a target. The target can either be a protein or nucleic acid. The FlexAID interface permits to adjust the parameters of FlexAID in a convenient environment.

The FlexAID interface contains a menu from which you can save/load sessions. For more information on sessions, see section 2.2. The basic view of the interface contains 4 tabs: *Input Files, Target Config, Ligand Config* and *Simulate*. The functionalities of each tab is reviewed in details in the following sections.

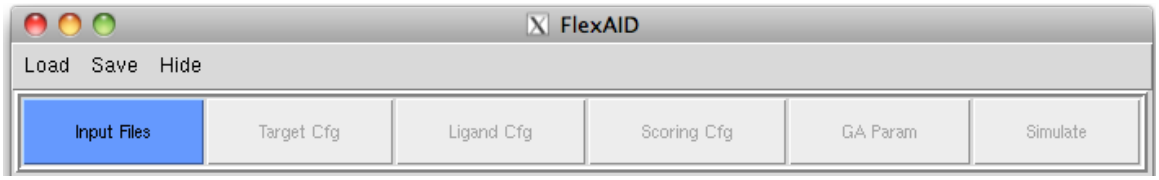


You can toggle between the Basic and Advanced views by clicking the button:

Show/Hide -> Advanced View



Two extra tabs are available in the Advanced view: *Scoring Config* and *Genetic Algorithm Params*.



A message box appears to hint you as going through the interface as well as informing of any errors that may occur.

You can reset the default values in the active tab (highlighted in blue) by clicking the "Default" button. You can close the FlexAID interface by clicking the "Close" button.

2.2. Working with sessions

A session within the FlexAID interface allows you to store and retrieve the content of all tabs (see section 2.1). 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*

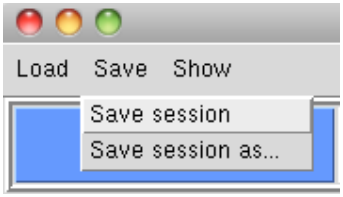
Sessions are particularly useful when you want to reproduce a simulation multiple times with the same parameters.

** Do not confuse a FlexAID interface session with a PyMOL session (using the extension .pse).

2.2.1. Saving a session

To save a session, you can either click the buttons:

Save -> Save session
Save -> Save session as…



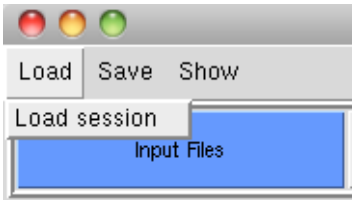
The “Save session as…” button allows you to save under a specific name. You cannot save a session elsewhere from the suggested directory.

The “Save session” button allows you to save a session under the name it was previously saved under. Thus, this will overwrite the currently saved session by the new one. In the case a session is loaded, this button will not overwrite the loaded session but rather has the same effect than “Save session as…”.

2.2.2. Loading an existing session

To load a session, click the following button:

Load -> Load session

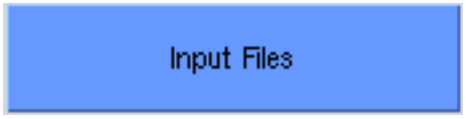


The “Load session” opens up the Session directory of the currently active project (see section 1.3). A session loaded elsewhere from the suggested directory will not be copied into your Session directory because the referenced files belong to another project.

We strongly suggest you naming correctly the objects you work with and not overwriting existing files when saving to avoid corrupting your saved sessions.

2.3. The Input Files tab

To activate the “Input Files” tab, click the corresponding tab.

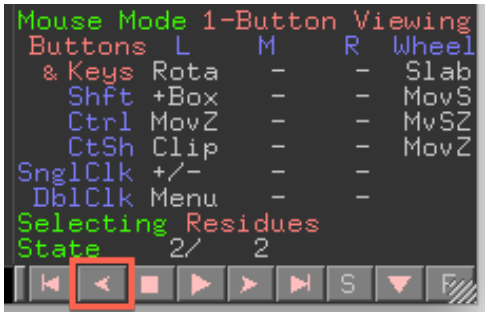


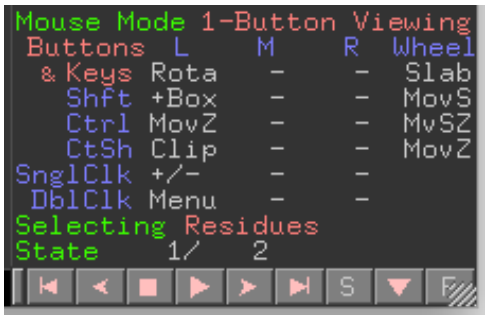
The functionalities of this tab are to:

- Import molecules (the target and the ligand)
- Displaying molecules
- Resetting molecules
- Generating 3D structures of a ligand
- Changing the anchor atom of a ligand

2.3.1. Importing the molecules

By default, new objects introduced into PyMOL are loaded into state 1. If an object of the same name already appears, it is appended to the next state. If you load a NMR solution, each model will appear into separate states. You can switch states as follows:





When importing a molecule in the FlexAID interface, make sure your object exists on the currently active state. If it does not exist, you may encounter the following error: "ERROR for object/selection '$OBJECT': The object must have at least (5) heavy atoms)".

2.3.1.1. Importing the target

FlexAID can handle a protein or a nucleic acid as target molecule. FlexAID will automatically detect the type of macromolecule imported.

There are 2 ways of importing a target into the FlexAID interface.

Saving from the object/selection widget

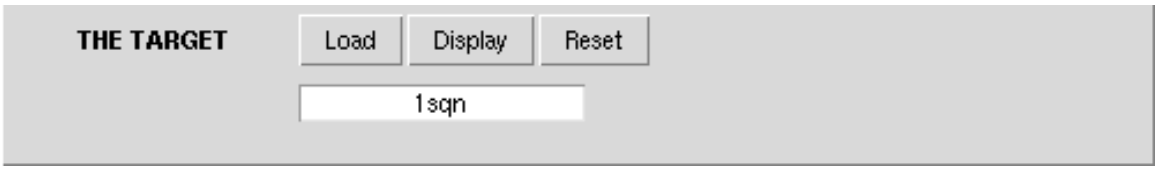
Make sure you select the right name corresponding to your target in the object/selection widget, then click the "Save as target" button.

You can rename the object/selection under a different name upon saving. When renamed, the object/selection will also be renamed under PyMOL. You cannot save a target elsewhere from the suggested directory.

Loading an existing file

The "Load" button next to "THE TARGET" opens up the Target directory of the currently active project (see section 1.3). A target loaded elsewhere from the suggested directory will be automatically copied into your Target directory.

When a target is imported successfully, its name will appear in "THE TARGET" box. This name will be referred as $TARGET throughout this guide.



2.3.1.2. Importing the ligand

FlexAID can handle all types of small molecules. However, the molecule must not exceed 100 heavy atoms (non-Hydrogen).

There are 4 ways of importing a ligand into the FlexAID interface.

Saving from the object/selection widget

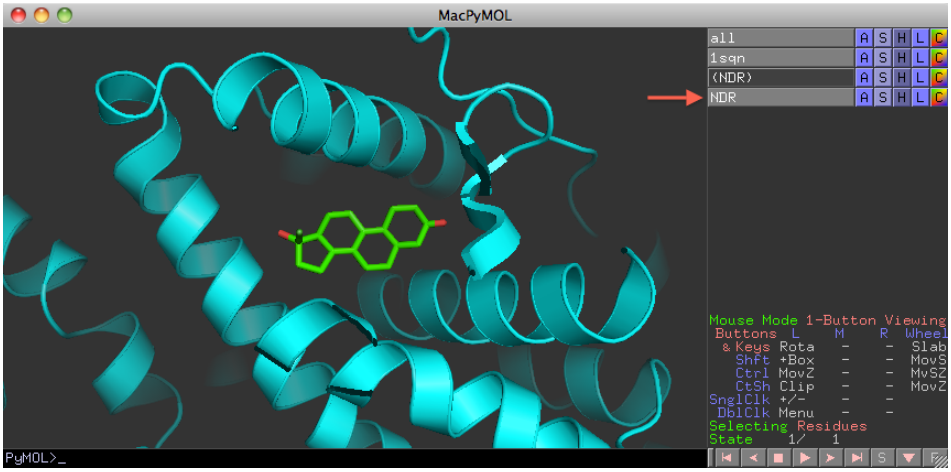
Make sure you select the right name corresponding to your ligand in the object/selection widget, then click the "Save as ligand" button.

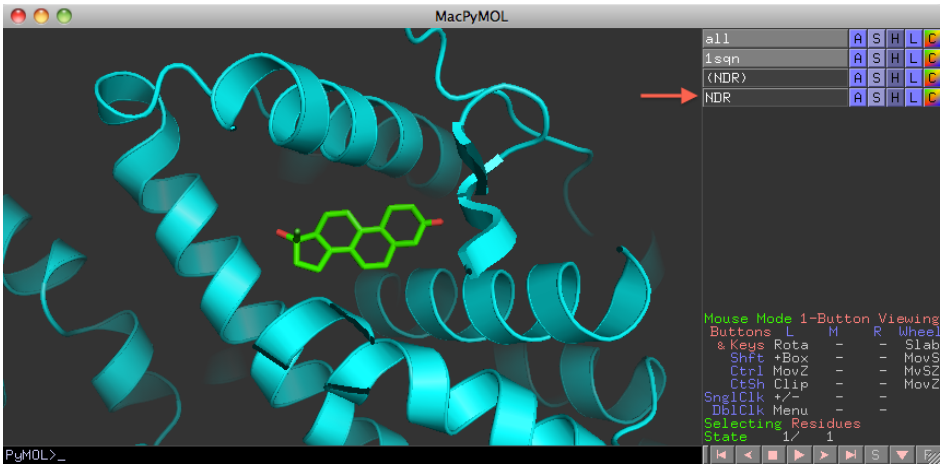
You can rename the object/selection under a different name upon saving. When renamed, the object/selection will also be renamed under PyMOL. You cannot save a ligand elsewhere from the suggested directory.

Saving an object/selection as a ligand is a valid approach in the case where the target is not bound to any ligand in the binding pocket of interest.

If you want to dock a ligand back into its binding pocket and your target is already bound to that ligand, you will not be able to do so. Thus, it is essential that the target and the ligand are dissociated and form 2 distinct objects.

In our example, by using the "Save as ligand" button on a selection of the ligand NDR, a new ligand object would be created (here also named "NDR"). However, the object "1sqn" representing our target would still be bound to the ligand.





Two options are available to us to dock a ligand back into its pocket.

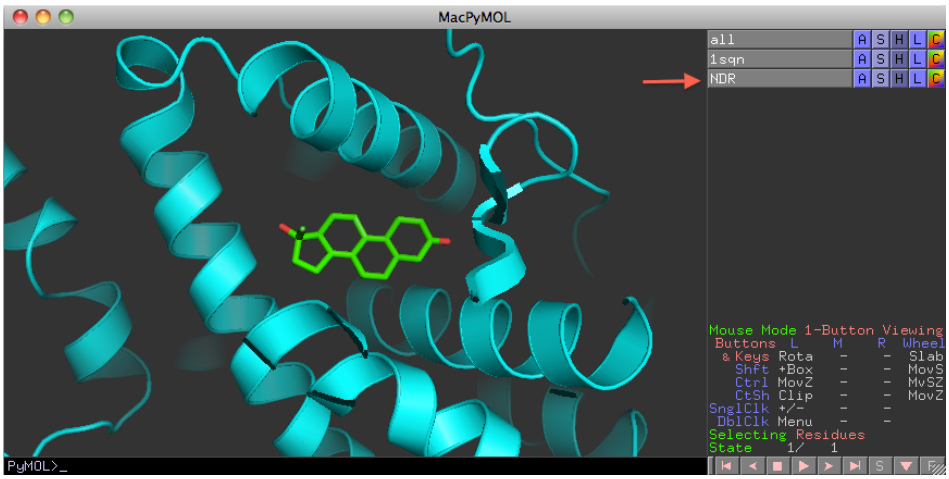
1) Use “Extract as ligand” rather than using “Saving as ligand”
2) Use the advanced feature “Include HET groups” (see section 2.6)

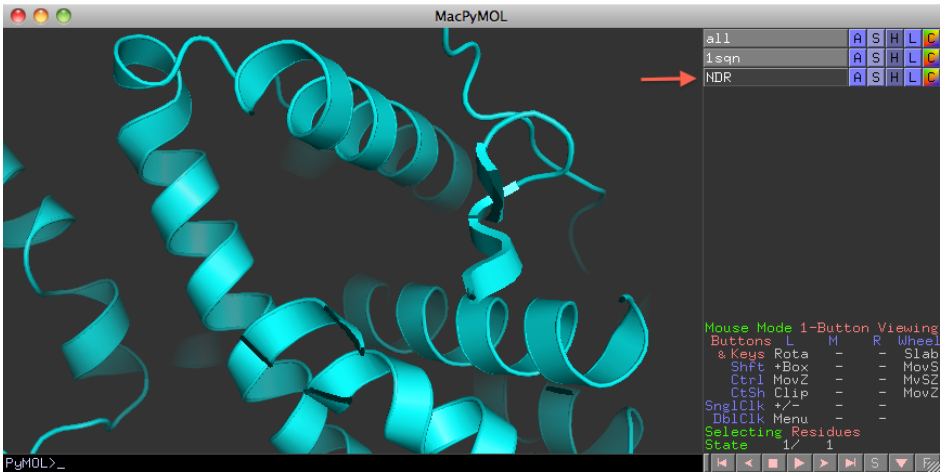
Extracting from the object/selection widget

Make sure you select the right name corresponding to your ligand in the object/selection widget, then click the “Extract as ligand” button.

You can rename the object/selection under a different name upon saving. When renamed, the object/selection will also be renamed under PyMOL. You cannot save a ligand elsewhere from the suggested directory.

It is important to note that extracting a ligand will dissociate the selection from the object and create 2 distinct objects.





We suggest you import your ligand first when an extraction is required. The extraction of a ligand from a target alters the target object. Thus, if the target object was imported into the FlexAID interface before the extraction, you need to update your target object and re-save it under a proper name such as 1sqn_noNDR.

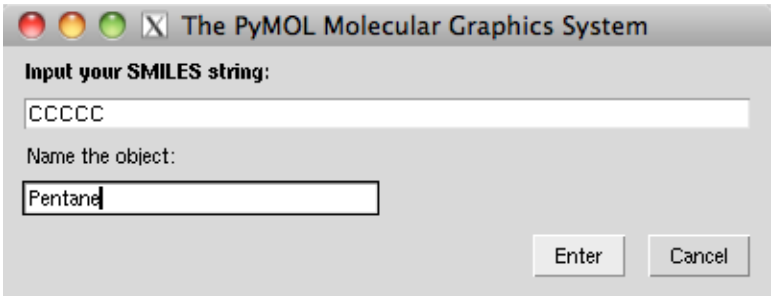
Loading an existing file

The "Load" button next to "THE LIGAND" opens up the Ligand directory of the currently active project (see section 1.3). A ligand loaded elsewhere from the suggested directory will be automatically copied into your Ligand directory.

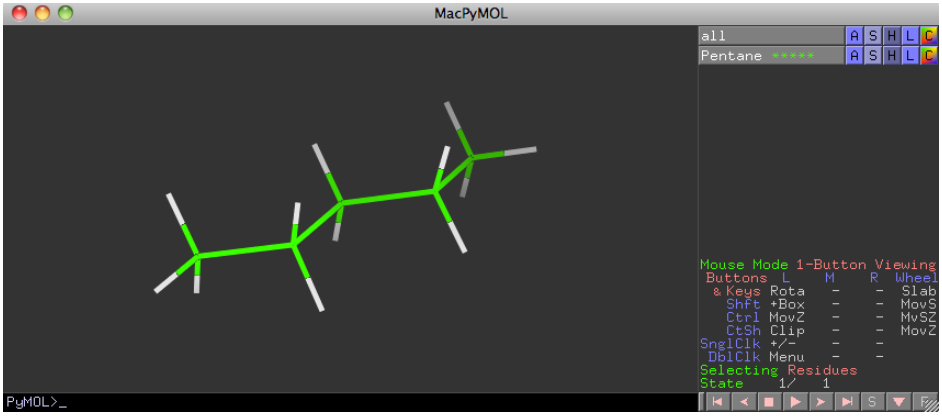
Generating a ligand from a SMILES string

It is feasible to input a SMILES to import a ligand by clicking the "Input" button.

A new window will pop up and asks you for the SMILES string as well as the name for the new object that will be created.



When you are ready to import the SMILES string, click the “Enter” button. The conversion process may take a couple of seconds to a few minutes depending on the size of the molecule because a three-dimensional conformation of the ligand is generated.



When a ligand is imported successfully, its name will appear in “THE TARGET” box. This name will be referred as $LIGAND throughout this guide.

The ligand is now ready to be processed.

2.3.2. Displaying the molecules

If your target or ligand objects do not appear anymore in the PyMOL viewer anymore, it is possible to make them re-appear.

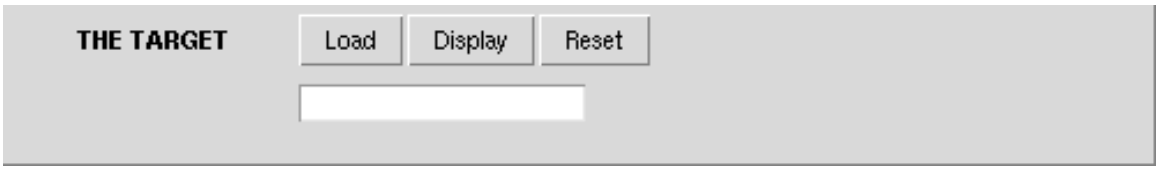
To display the target, click the “Display” button next to “THE TARGET”.

To display the ligand, click the “Display” button next to “THE LIGAND”.

If an object is already appearing, this button will simply zoom on the object.

2.3.3. Resetting the molecules

If you want to clear the target, click the "Reset" button next to "THE TARGET".



Resetting the target will also reset the configuration of the ligand (see section 2.5) except for the ligand flexibility (see section 2.5.1.2).

If you want to clear the ligand, click the "Reset" button next to "THE LIGAND".

2.3.4. Processing the molecules

Once the target and the ligand are set, you have access to the other tabs.

When the user activates another tab, the ligand and the target have to be processed. This process may take a few seconds to a few minutes 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.

Every molecule you import need to be processed. Once a ligand has been processed, you do not need to process it again except in the following cases:

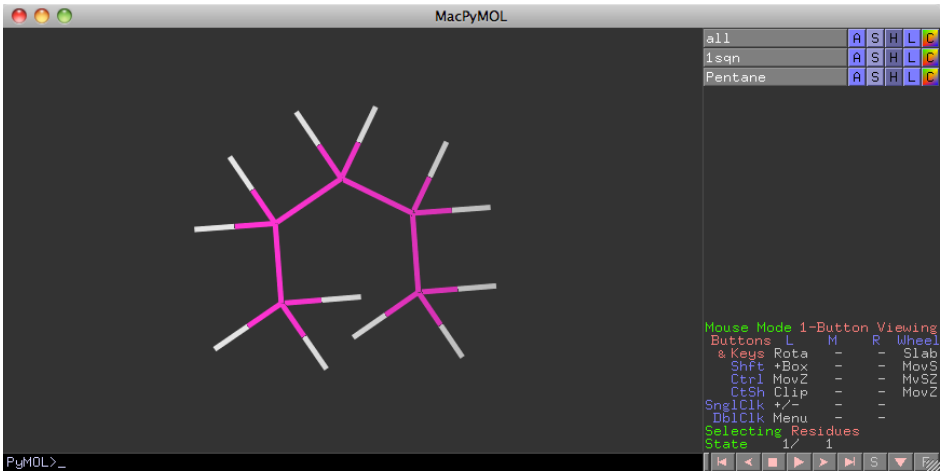
- The user toggles the 3D conformation checkbox (see section 2.3.5)
- The anchor atom of the ligand was modified (see section 2.3.6)

The processing of the ligand creates a new object called LIGAND__ (reserved name see section 1.4) with newly generated coordinates when generating a 3D structure. The processing of the target creates the object TARGET__.

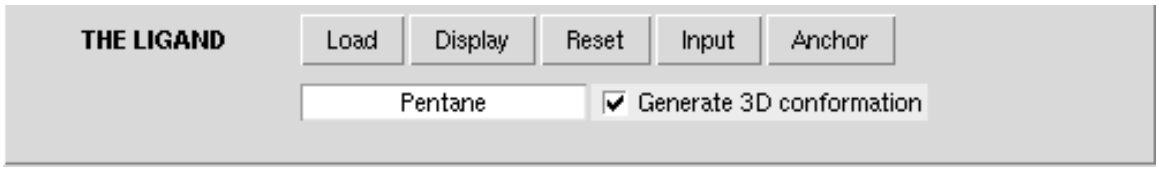
Despite that Hydrogen atoms are removed from the ligand because FlexAID do not take them into account, they are considered during the minimization when generating a 3D structure.

2.3.5. Generating a 3D conformation of the ligand

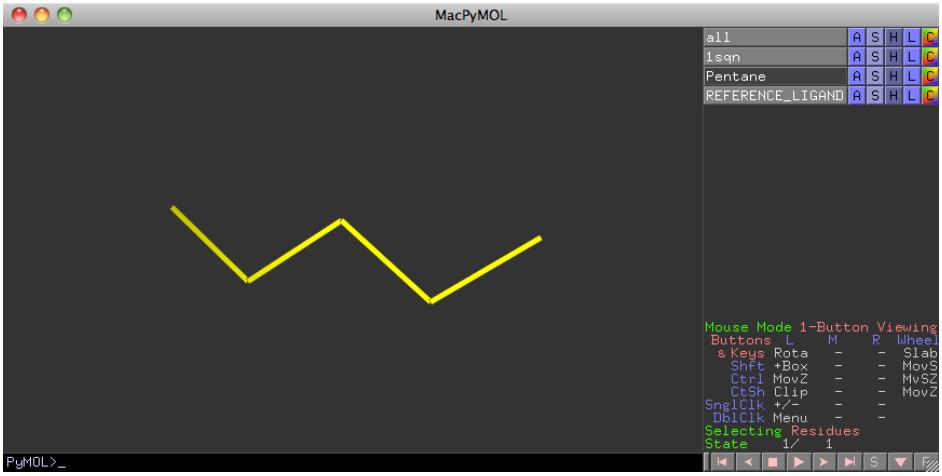
Molecules that are imported may be entirely planar and may require a minimization (using force-fields) prior docking to generate the 3D structure. In this example, we use a planar Pentane.



To generate a 3D structure of your ligand, check “Generate 3D conformation”.



During the processing of the ligand (see section 2.3.4), the ligand is minimized using force fields from OpenBabel.

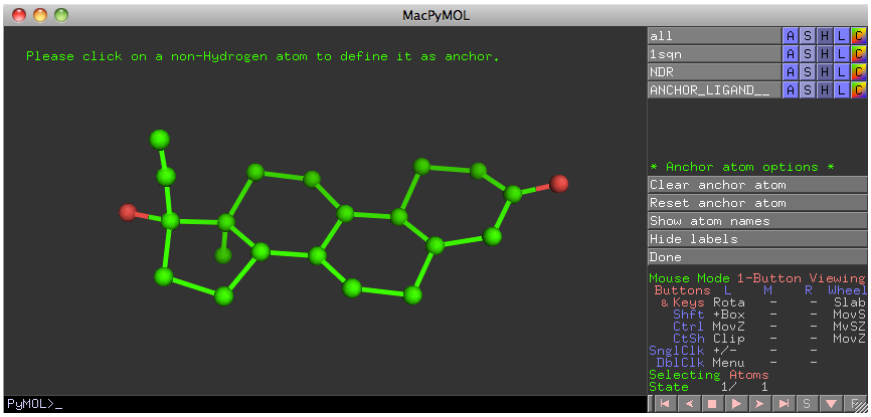


2.3.6. Changing the anchor atom of the ligand

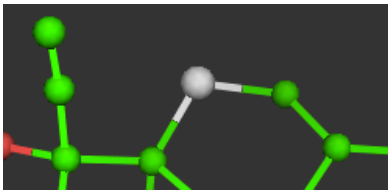
*** This feature is for advanced users only*

You can change the anchor atom of the ligand by clicking the “Anchor” button next to “THE LIGAND”.

Your ligand will appear as the object ANCHOR_LIGAND__. A wizard will open up waiting for you to click to a non-Hydrogen atom of your ligand. (see section 1.6)

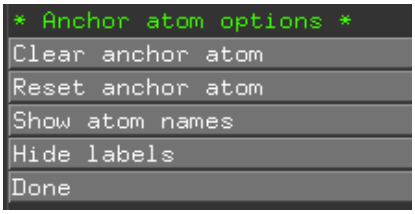


Once you activate an atom, that atom turns white.



The “Reset anchor atom” overwrites the active anchor atom with the previously saved anchor atom.

The “Clear anchor atom” button overwrites the previously saved anchor atom with the default atom.



2.4. The Target Config tab

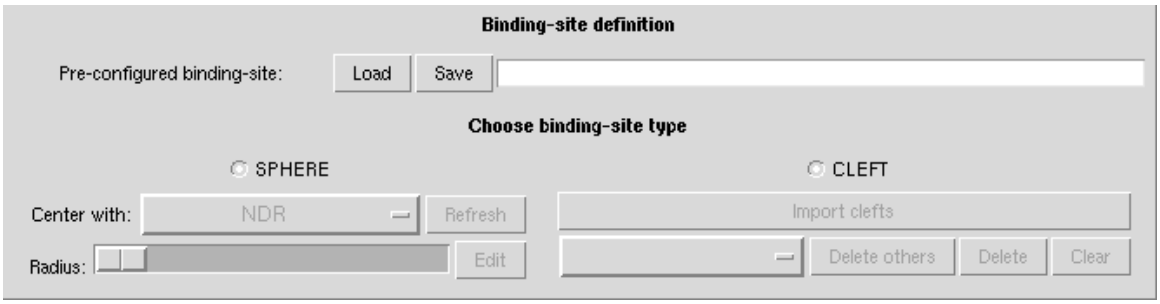
To activate the “Target Config” tab, click the corresponding tab.

The functionalities of this tab are to:

- Define a binding-site for a target
- Introduce flexibility in the target

2.4.1. Defining the binding-site of the target

The following box is dedicated to the definition of binding-sites.



FlexAID can take as input 2 different types of binding-sites.

- A spherical shape
- A combination of one or more cleft(s)

It is also possible to save your binding-site or to load an existing binding-site.

A new object called BINDINGSITE_AREA__ is created when defining a binding-site. When the user starts a simulation, a three-dimensional grid (of spacing 0.375Å) is built within the volume of the binding-site.

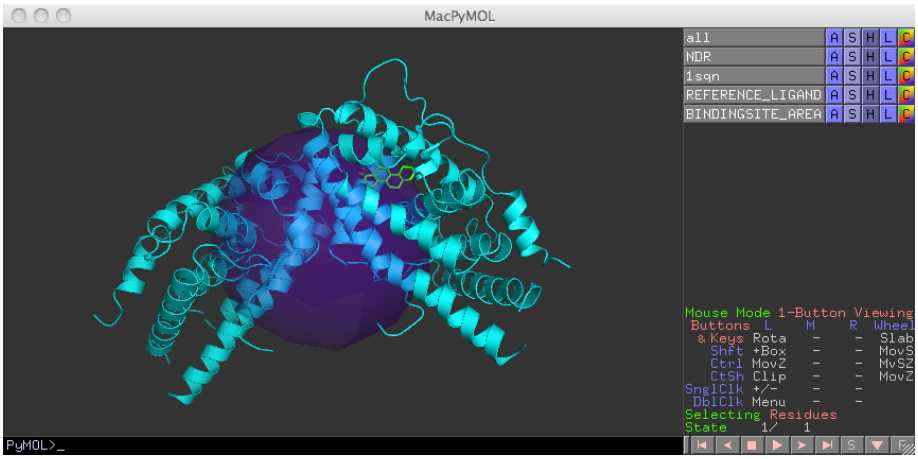
These grid vertexes serve as anchors for the anchor atom of the ligand. Thus, not necessarily all atoms of the ligand will be confined within the volume of the binding-site.

The binding-site is reset when the $TARGET (see section 2.3.1.1) is modified in the Input Files tab.

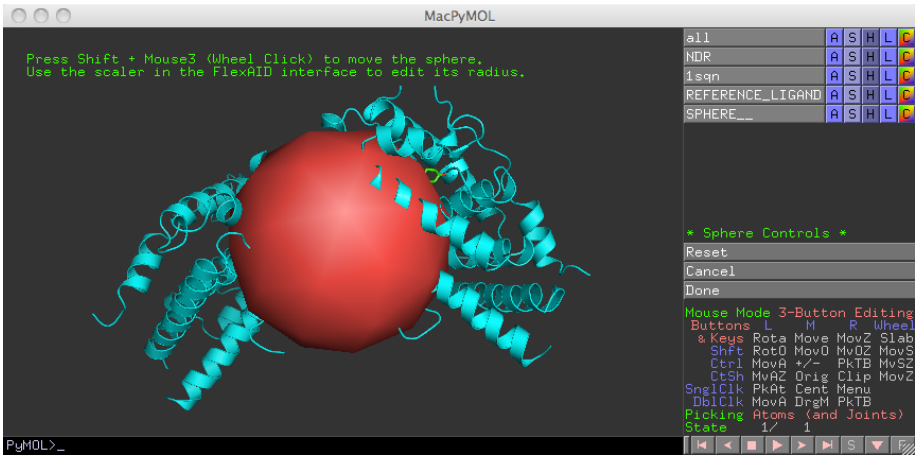
2.4.1.1. Defining the binding-site using a sphere

You can define your binding-site by using a spherical shape. You can build the binding-site by clicking the “SPHERE” radio button.

By default, a sphere centered on your target is created. A sphere appearing as transparent blue-violet is not editable.



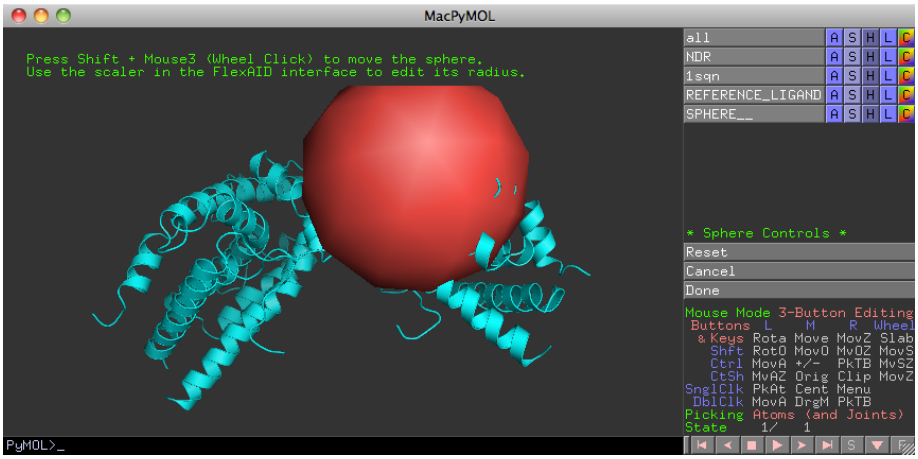
If you wish to edit the origin and radius of the sphere, click the “Edit” button. The editable sphere object called SPHERE__, will appear as opaque red. A wizard will open up waiting for you to move and resize the sphere (see section 1.6).



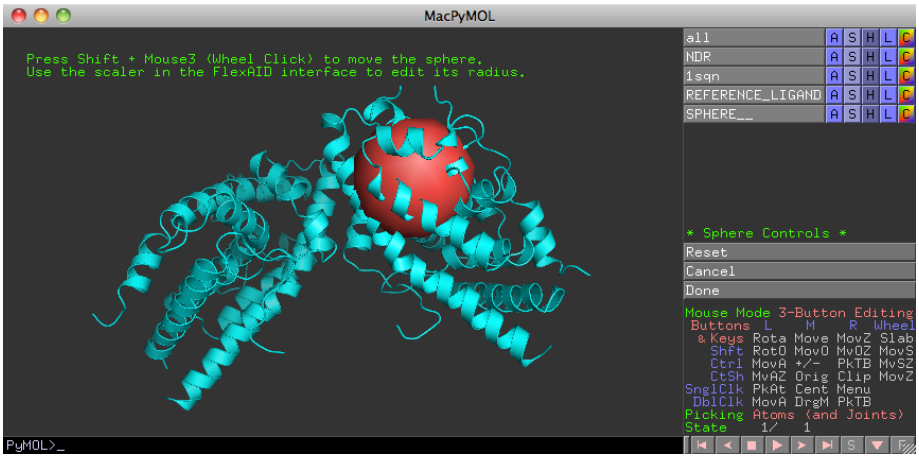
You can move the sphere manually by holding:

- With a mouse: “Left Shift” key on your keyboard and Mouse 3 button on the mouse (usually the Wheel click).
- With a trackpad: “Left Ctrl” key and Left click on the trackpad.

You can center the sphere on an existing PyMOL object/selection by clicking its name in the widget (see section 1.5).



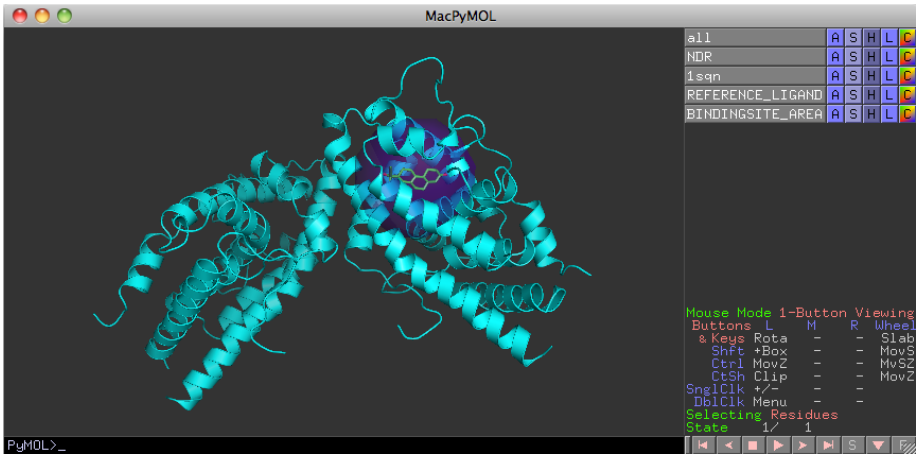
Dragging left and right the scale changes the radius of the sphere.



You can reset the origin and radius of the sphere to the values when the Wizard was activated, by clicking the “Reset” button in the Wizard menu.

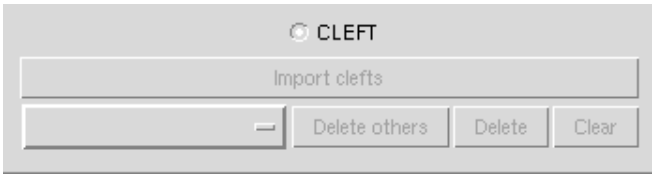
You can cancel the changes that were made and rebuild the binding-site using the previously saved binding-site by clicking the “Cancel” button.

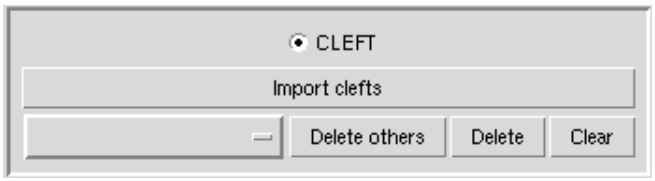
You can accept the changes that were made and build the new binding-site by clicking the “Done” button.



2.4.1.2. Defining the binding-site using clefts

You can define your binding-site using a combination of one or more clefts. Clefts are generated using the GetCleft interface (see section 3.3.2). You can build the binding-site by clicking the "CLEFT" radio button.

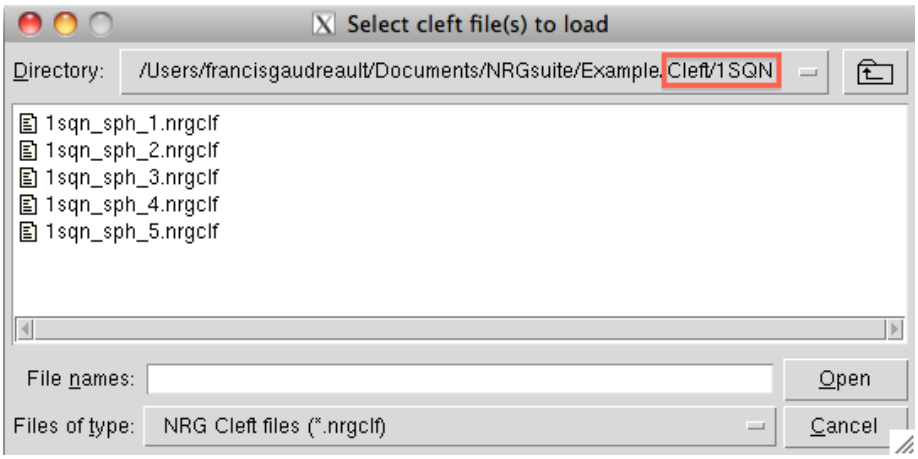




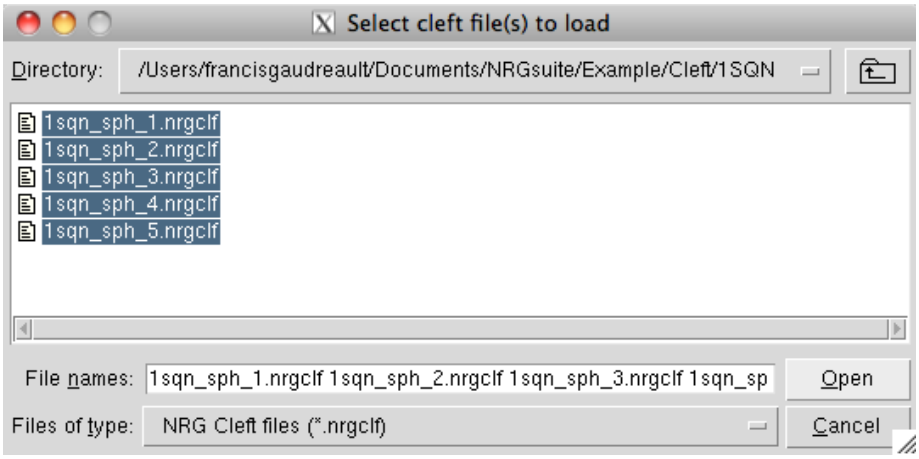
To build on the binding-site using clefts, you need to import at least one cleft by clicking the "Import clefts" button.

The "Import clefts" button opens up the Cleft directory of the currently active project (see section 1.3). However, if clefts were previously saved (see section 3.2.2) for your $TARGET (see section 2.3.1.1), the sub-directory Cleft/$TARGET is opened.

Clefts loaded elsewhere from the suggested directory will not be copied into your Cleft directory.

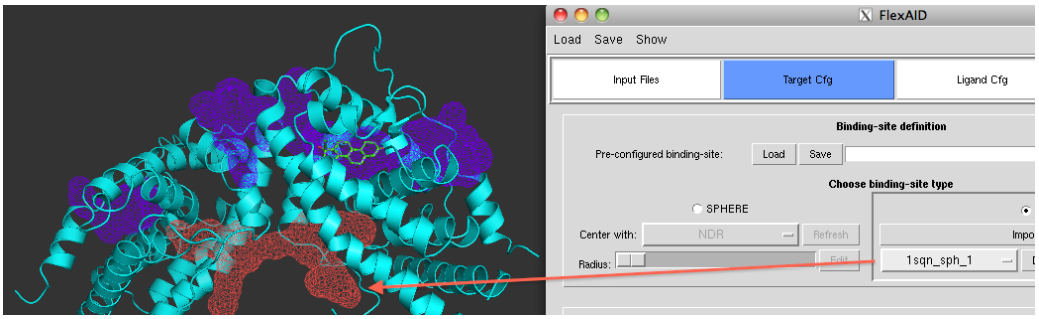


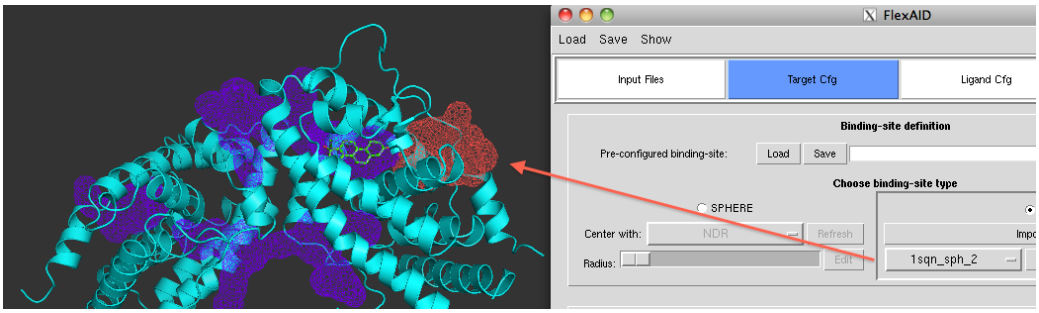
You can import multiple clefts at once. Once you selected your clefts, click the "Open" button to import them.



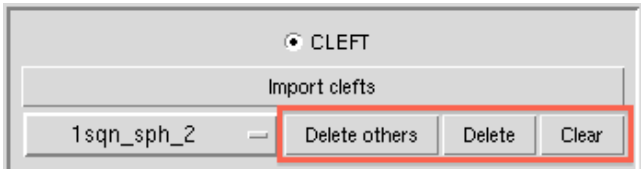
Once imported, the clefts will appear in the clefts list:

The active cleft is highlighted in red while other clefts are blue-violet. It is important to note that docking of the ligand will take place in all clefts in the clefts list and not only the active cleft.



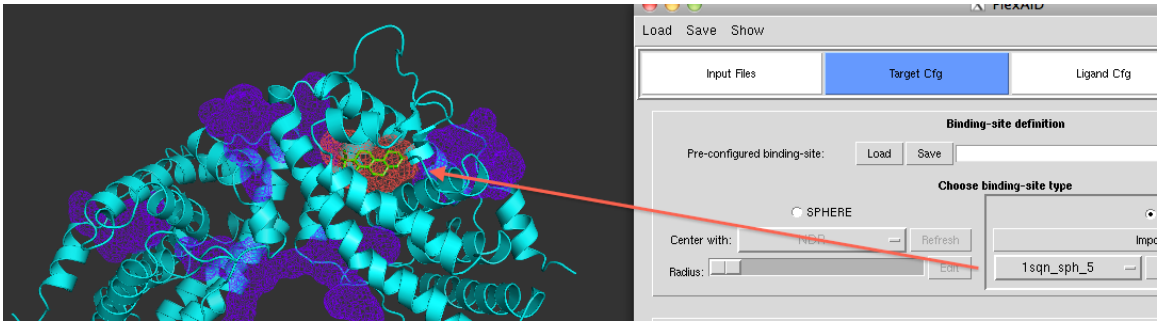


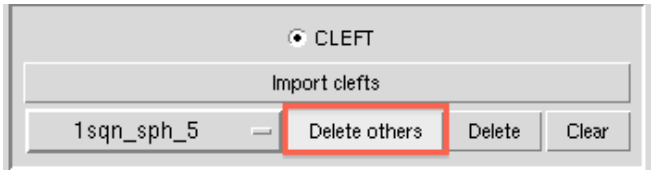
You can manipulate the clefts by using the buttons in the “CLEFT” box.

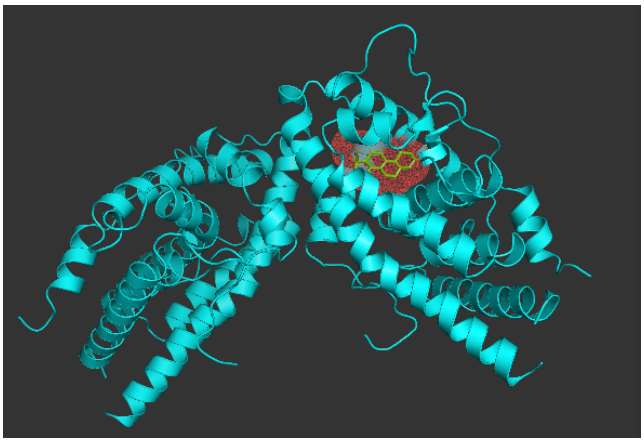


“Clear” will delete all clefts in the clefts list.
“Delete” will delete the active cleft.
“Delete others” will delete all clefts except for the active cleft.

In our example, if we wanted to dock back the NDR into its binding pocket, we could do the following:

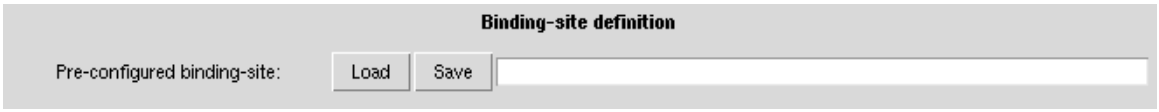






2.4.1.3. Pre-configured binding-sites

You can save your binding-site or load an existing one.



Saving the binding-site

If you do not want to rebuild your binding-site every time you work with your target, you can save it by clicking the "Save" button.

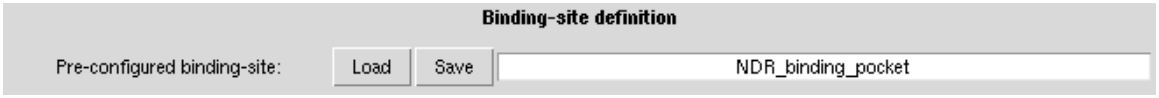
You can rename the binding-site under a different name upon saving. The default directory when saving a binding-site is Binding_Site/$TARGET (see section 2.3.1.1) of the currently active project (see section 1.3). You cannot save a binding-site elsewhere from the suggested directory.

Loading an existing binding-site

You can load an existing binding-site by clicking the "Load" button.

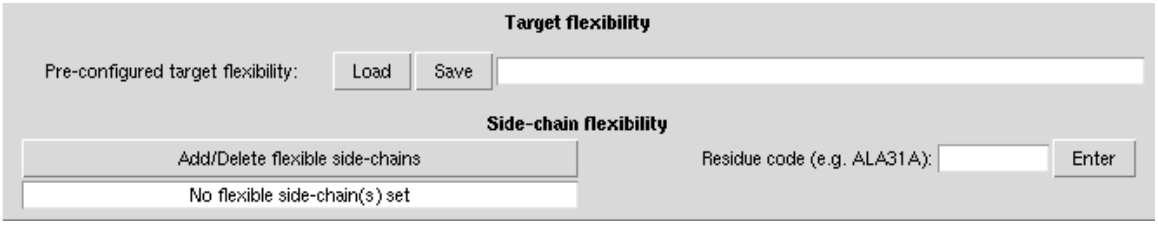
The default directory when loading a binding-site is Binding_Site/$TARGET (see section 2.3.1.1) of the currently active project (see section 1.3). A binding-site loaded elsewhere from the suggested directory will not be copied into your Binding_Site directory.

Upon successful saving or loading, the binding-site name will appear:



2.4.2. Introducing flexibility in the target

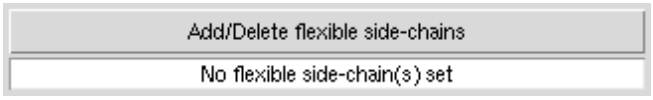
Target flexibility can be introduced in the following box:



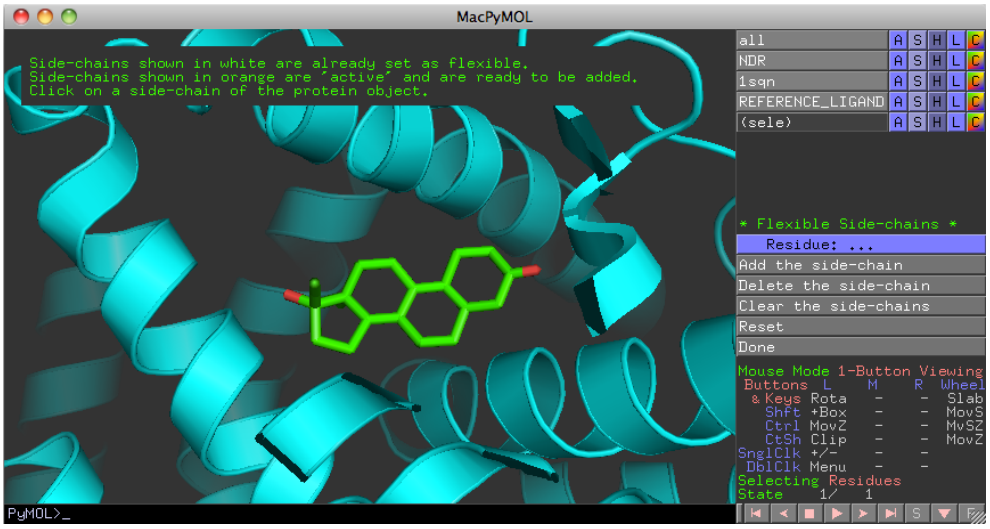
You can render side-chains of a protein as flexible by clicking the “Add/Delete flexible

2.4.2.1. Adding a side-chain from the PyMOL viewer

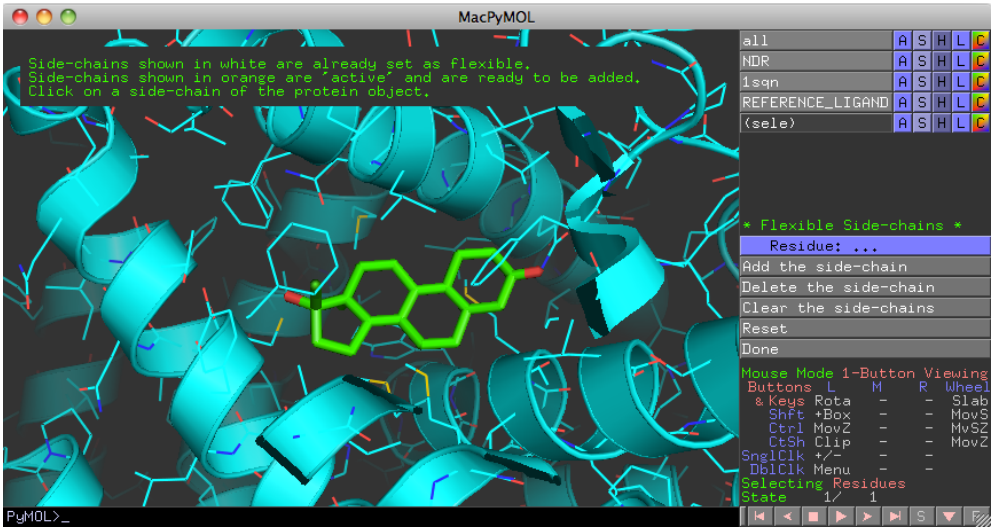
You can add side-chains by clicking the “Add/Delete flexible side-chains” button.



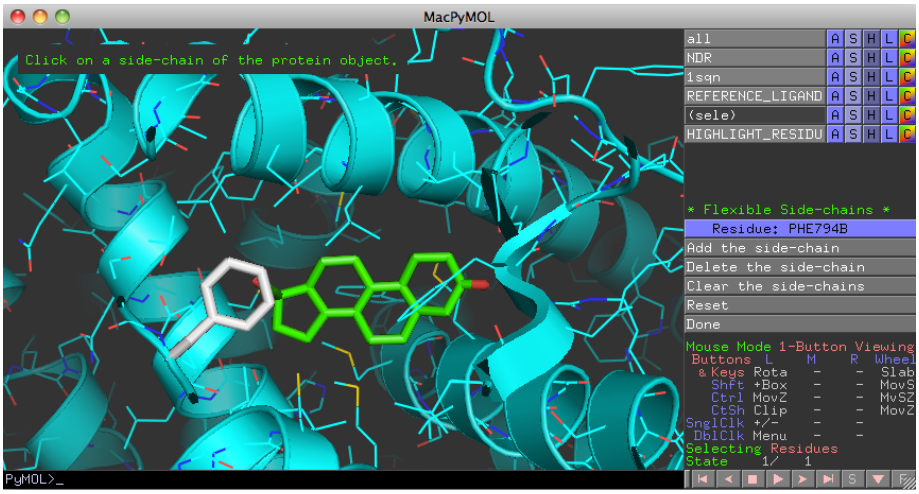
A wizard will open up waiting for you to click on side-chains on your target protein to make them as flexible (see section 1.6).



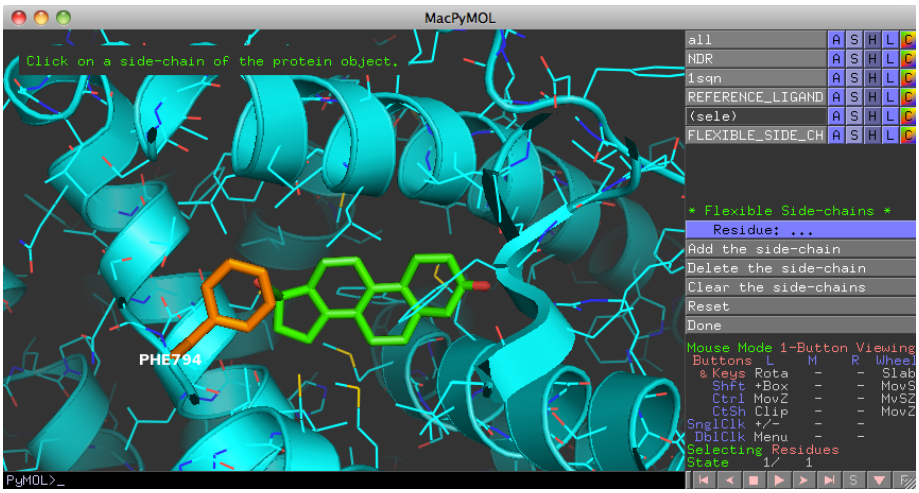
Make sure you show the ‘lines’ display for your target protein object so you can click on the side-chain atoms.



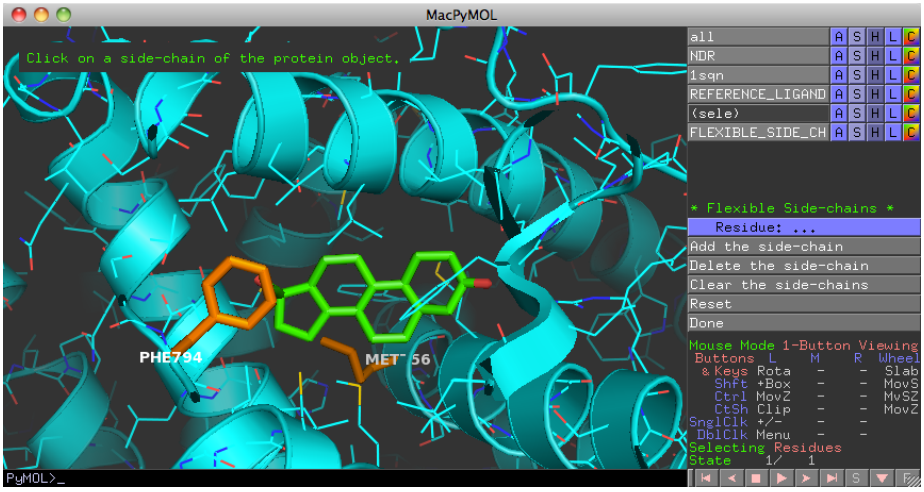
Clicking on a side-chain will activate that side-chain (in white), meaning you can interact with it via the controls of the Wizard menu.



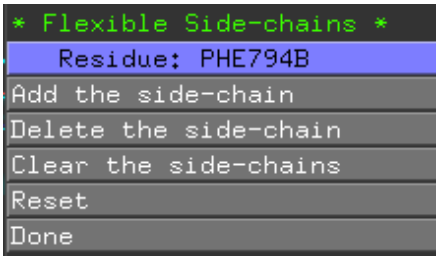
You can make the side-chain as flexible during docking simulations by clicking the "Add the side-chain" button. Once a side-chain has been added, the side-chain will appear in orange and its residue name will be displayed.



Multiple side-chains can be added within the Wizard.



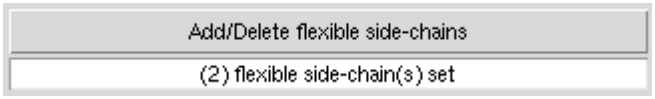
If you wish to remove a side-chain, activate it then click the “Delete the side-chain” button.



The “Reset” button inactivates the active side-chain.

The “Clear the side-chains” button will delete all side-chains set by the user.

When the Wizard is inactivated, the number of flexible side-chains set by the user will be displayed.



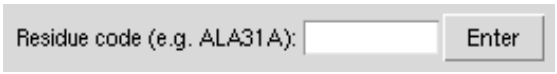
2.4.2.2. Adding a side-chain by its residue code

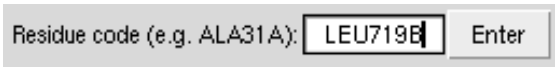
Each protein residue has a unique residue code consisting of:

- The three-letters-code of the amino acid
- The residue number
- The chain identifier of the residue

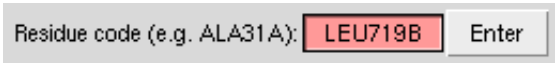
In the case where the chain identifier is null, use the hyphen identifier (e.g. ALA31-).

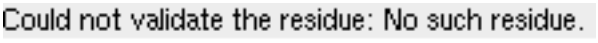
When you already know the residue code of the side-chain, type its residue code in the following area and click “Enter”.



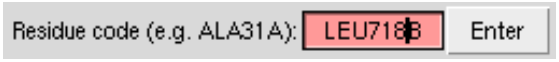


The residue code gets validated to ensure the residue exists and does not miss atoms. If the validation fails, a description of the error appears in the message box.



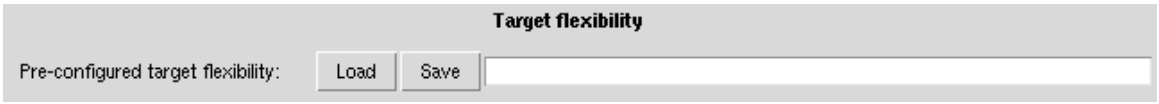


Make the appropriate changes when the validation fails then click the “Enter” button again. When the residue is added, your input residue code will disappear and the number of flexible side-chains set by the user is displayed on the left.



2.4.2.3. Pre-configured target flexibility

You can save your target flexibility or load an existing one.



Saving the target flexibility

If you do not want to set the flexibility of a target every time you work with your target, you can save it by clicking the "Save" button.

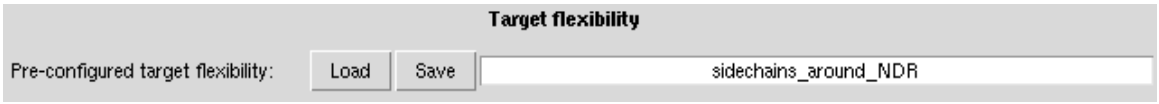
You can rename the target flexibility under a different name upon saving. The default directory when saving a target flexibility is Target_Flexibility/$TARGET (see section 2.3.1.1) of the currently active project (see section 1.3). You cannot save a target flexibility elsewhere from the suggested directory.

Loading an existing target flexibility

You can load existing target flexibility by clicking the "Load" button.

The default directory when loading a target flexibility is Target_Flexibility/$TARGET (see section 2.3.1.1) of the currently active project (see section 1.3). A target flexibility loaded elsewhere from the suggested directory will not be copied into your Target_Flexibility directory.

Upon successful saving or loading, the target flexibility name will appear:



2.5. The Ligand Config tab

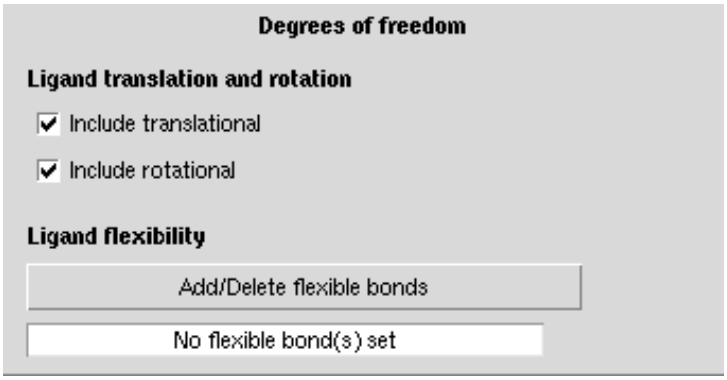
To activate the “Ligand Config” tab, click the corresponding tab.

The functionalities of this tab are to:

- Include degrees of freedom for the ligand
- Use the ligand as reference
- Add constraints

2.5.1. Including degrees of freedom for the ligand

The following box controls the degrees of freedom that can be introduced during the docking simulation:



The different degrees of freedom that can be introduced are:

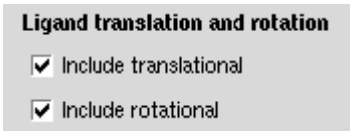
- Translational degree of freedom
- Rotational degree of freedom
- Torsional bonds of the ligand

The degrees of freedom are reset when the $LIGAND (see section 2.3.1.2) is modified in the Input Files tab.

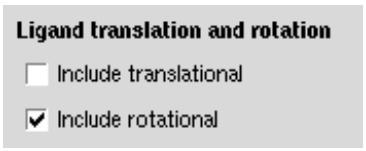
By default, all ligand flexible bonds are treated as non-rotatable (rigid). To make them as flexible, see section 1.8.

2.5.1.1. Including ligand translation and rotation

By default, translational and rotational degrees of freedom are included.

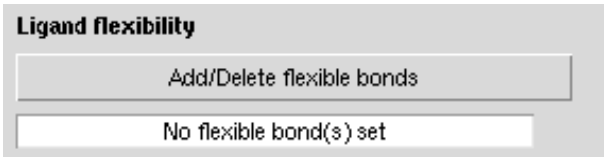


To exclude the translational or rotational degree of freedom, simply uncheck the corresponding checkbox:



2.5.1.2. Including ligand flexibility

When a ligand is imported, the molecule is considered as rigid during docking unless the user explicitly includes ligand flexibility. To include ligand flexibility, click the "Add/Delete flexible bonds" button.

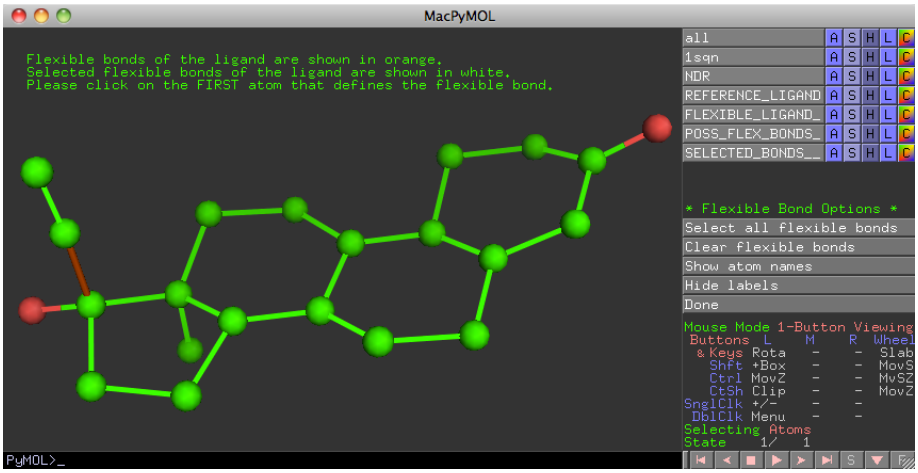


Your ligand will appear as the object FLEXIBLE_LIGAND__.
The following objects are also created:

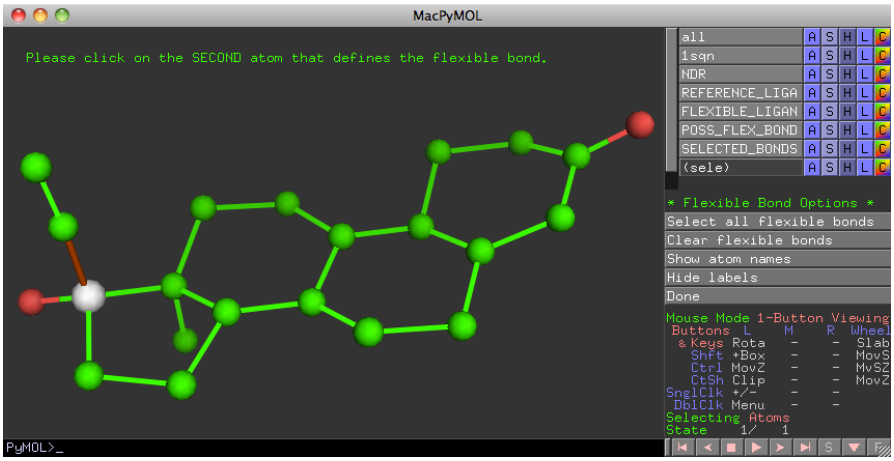
- POSS_FLEX_BONDS__ (to display the possible flexible bonds of the ligand)
- SELECTED_BONDS__ (to display the selected flexible bonds of the user).

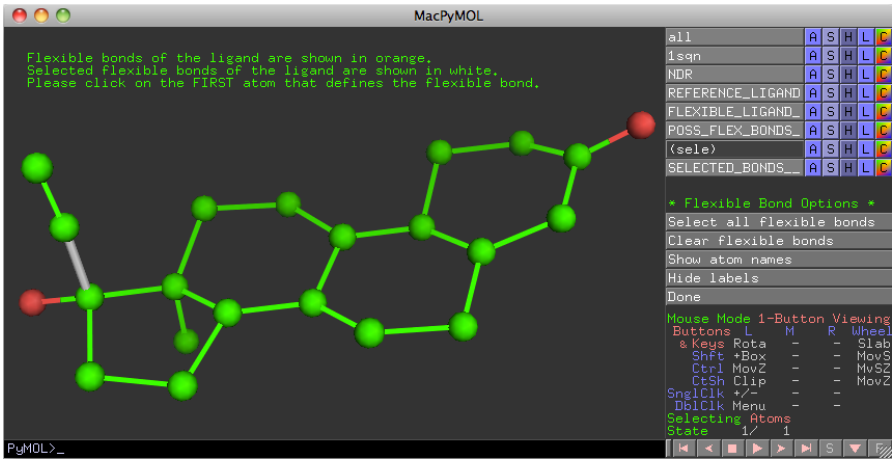
Do not interact with these last 2 objects.

A wizard will open up waiting for you to click on the 2 atoms that define the flexible bond (see section 1.6).



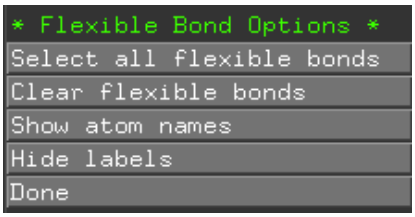
Flexible bonds appear in orange. Clicking on the 2 atoms that define the flexible bond the bond will render this bond as flexible during the docking simulation. The selected bonds of the user appear in white.



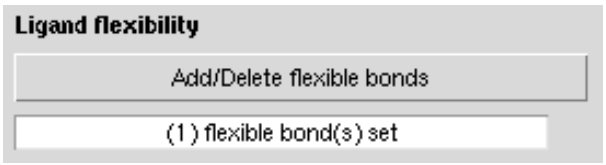


Rather than selecting all bonds manually, you can quickly select all flexible bonds by clicking the “Select all flexible bonds” button in the Wizard menu.

The “Clear flexible bonds” removes all selected flexible bonds of the user.



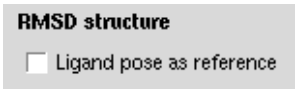
When the Wizard gets inactivated, the number of selected flexible bonds will be shown in the interface.



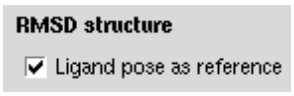
2.5.2. Using the ligand as reference

In some situations, the user may wish to use the LIGAND__ object (see section 2.3.4) as reference pose during the docking simulation. This may particularly be the case when the user extracts a ligand from a bound complex and wants to dock the ligand back into its binding pocket (most commonly referred as self-docking).

By default, the Root-Mean-Square Deviation (RMSD) from a reference pose is inactivated.



If you wish to calculate the RMSD during the docking simulation, simply check "Ligand pose as reference".

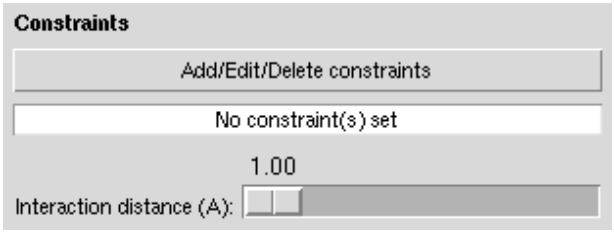


2.5.3. Adding constraints

FlexAID allows the use of constraints to drive the genetic algorithm. 2 types of constraints can be added:

- Intra-molecular constraints (ligand-ligand or target-target)
- Inter-molecular constraints (ligand-target)

The following box permits to manage the constraints. To add constraints click the "Add/Edit/Delete constraints" button.

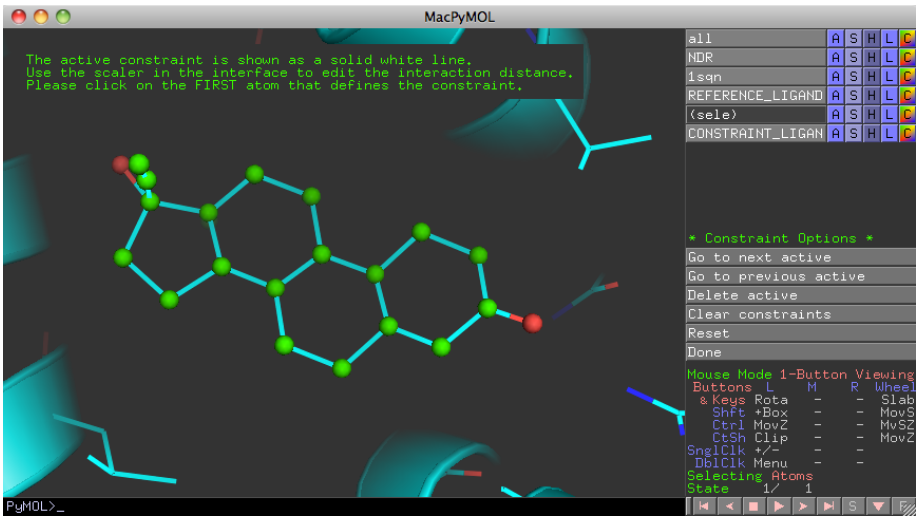


Your ligand will appear as the object CONSTRAINT_LIGAND__.
The following objects are also created:

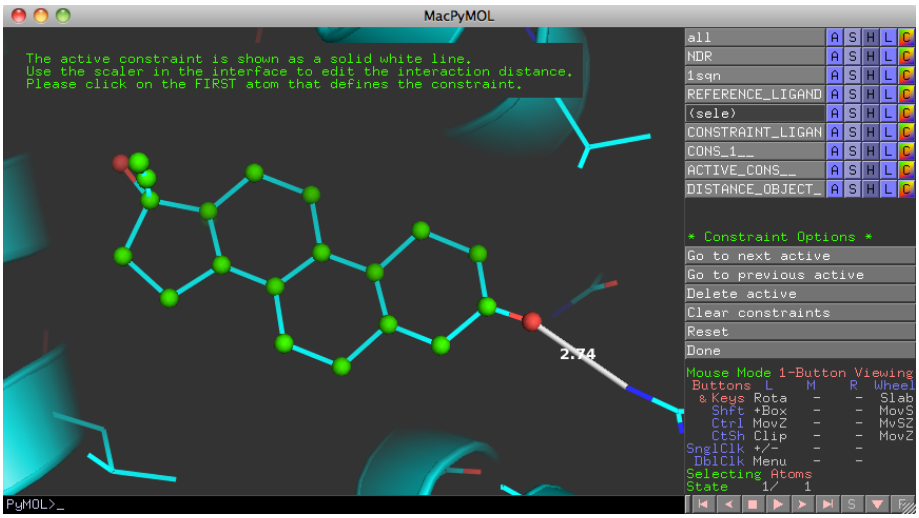
- ACTIVE_CONS__ (to display the active constraint)
- CONS_X__ (to display each constraint separately)
- DISTANCE_OBJECT__ (to display the constraint distance)

Do not interact with these last 3 objects.

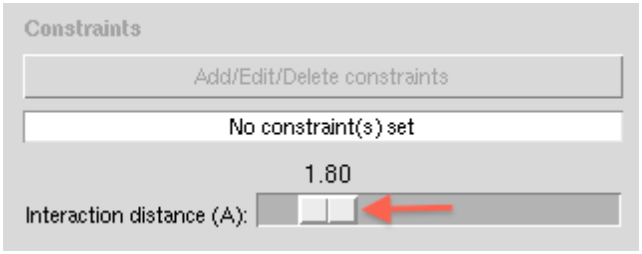
A wizard will open up waiting for you to select 2 atoms that define the constraint (see section 1.6). Adding constraints between a ligand not yet bound to our target can be particularly tricky considering the molecules may be far away from each other. To overcome this issue, we suggest you preliminary dock your ligand into your binding pocket and extract the ligand from the result (see section 2.3.1.2).

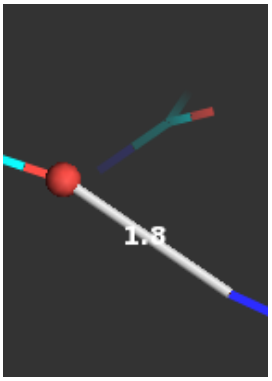


When the user selects 2 atoms to define a constraint, a constraint is created. The default constraint distance is the distance between the 2 atoms or 2.5Å if the distance exceeds the limit of detection of contacts of 7.0Å (in FlexAID).

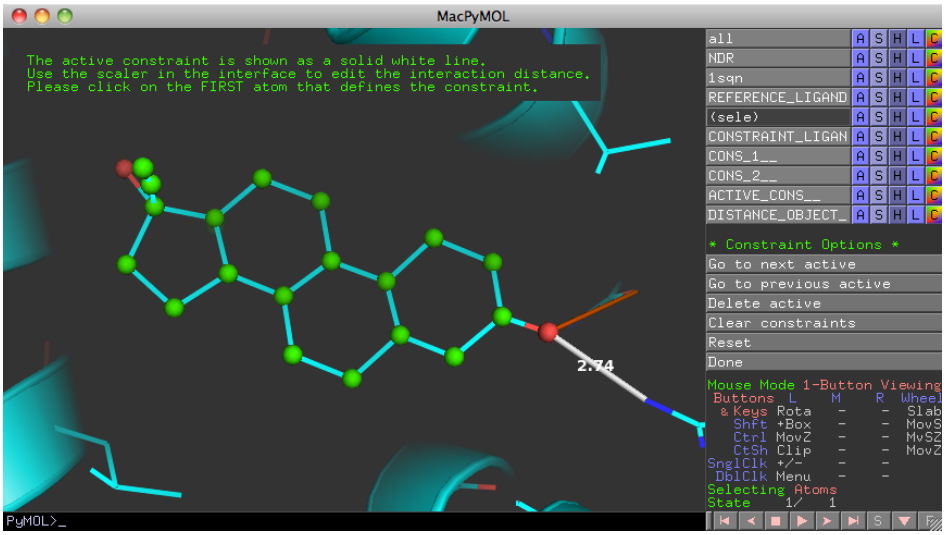


You can drag left or right the scale in the FlexAID interface to adjust the constraint distance of the active constraint.

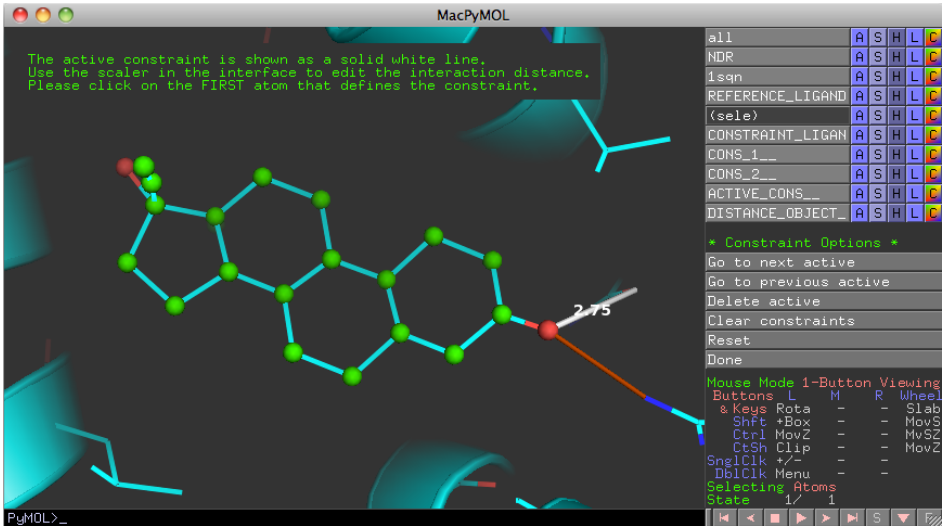




You can add multiple constraints within the Wizard.



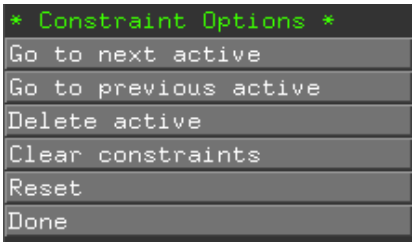
The active constraint is shown in white while other constraints are in orange. You can move through the constraints by using the “Go to next active” and “Go to previous active” buttons.



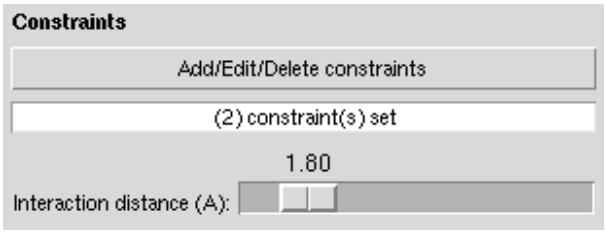
From the Wizard menu, you can delete the active constraint by clicking the “Delete active” button.

You can delete all constraints set by the user by clicking “Clear constraints”.

The “Reset” button will reset the atom selection back to the first atom.



When the Wizard gets inactivated, the number of constraints will be shown in the interface.



2.6. The Scoring Config tab

*** This tab is for advanced users only*

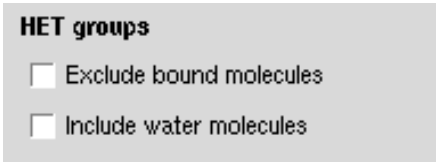
To activate the "Scoring Config" tab, click the corresponding tab.

The functionalities of this tab are to:

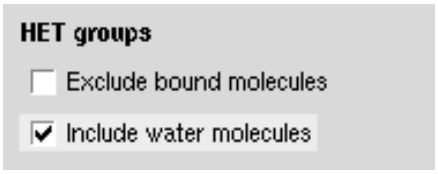
- Exclude atoms that will be scored
- Exclude and include bound molecules of the target
- Adjust the permeability for soft-docking
- Adjust the properties of the solvent
- Alter the size of the search space

2.6.1. Including the heteroatoms of the target

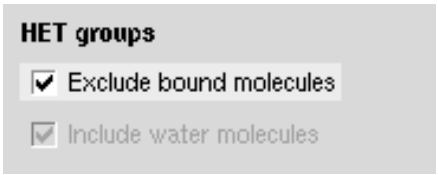
The $TARGET may be bound to HET atoms (typing defined in the PDB as HETATM) including molecules such as 1) metals, 2) modified amino acids, 3) water molecules and 4) other ligands. It is possible to dock on the target while omitting or considering those HET groups using the following menu.



By default, water molecules are excluded during docking. Ordered water molecules may be important for bridging interactions. To include them, simply check the corresponding checkbox.



Other HET groups are included by default, to exclude them check “Exclude bound molecules”.



It is important to note that excluding bound molecules will exclude metals and ligands. However, we do not suggest using this option considering whole residues of the proteins (modified amino acids) are also excluded.

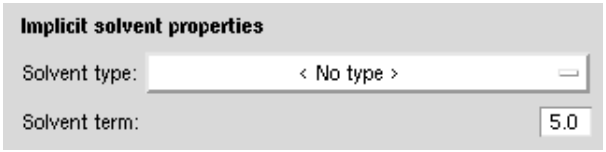
2.6.2. Adjusting the soft-docking properties

Steric clashes occur when the distance between two atoms (non-bonded covalently) is smaller than the sum of their Van der Walls radii. The permeability allows the steric clashes to be slightly tolerated and not penalizing (referred as Soft-docking). By default, a permeability of 10% (0.1) is allowed before computing steric clashes.

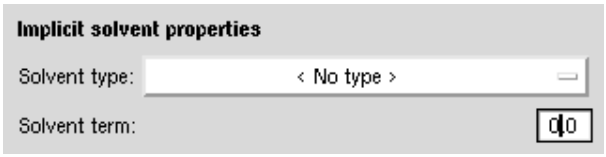
You can modify the percentage of permeability as you wish. We do not suggest using permeability over 20%.

2.6.3. Adjusting the properties of the solvent

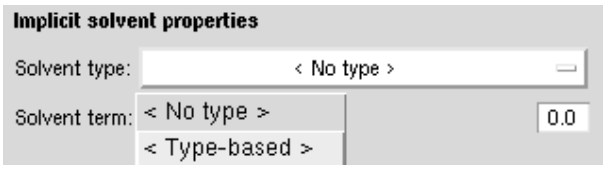
FlexAID uses an implicit solvent model, i.e. areas that are not in contact with other atoms are considered being in contact with the solvent. You can adjust the properties of the solvent. Solvent type can either be “No type” or “Type-based”. By default, the solvent type is set to “No type”.



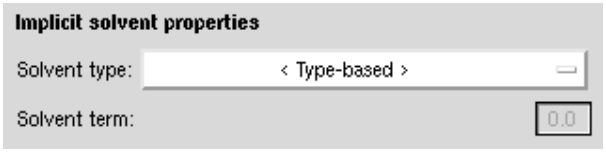
As “No type”, the solvent forms interactions independently of the atoms type in contact with. By default, the solvent term is set to a positive value (5.0), thus all interactions with the solvent are unfavourable. This forces the ligand to anchor deeply into the target cavities. A negative value makes all interactions with the solvent as favourable. The solvent would have no impact with a null value (0.0). You can adjust the solvent term as you wish.



It is possible to toggle to solvent type “Type-based”.

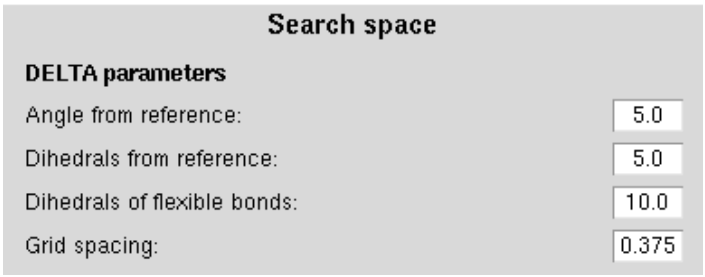


Contrary to “No type”, the solvent forms interactions, which depend on the atoms type in contact with.



2.6.4. Changing the resolution of docking

When the rotational degree of freedom of the ligand is included (see Section 2.5.1.1), the ligand rotates around a reference point in space defined by the grid vertexes (see section 2.4.1).

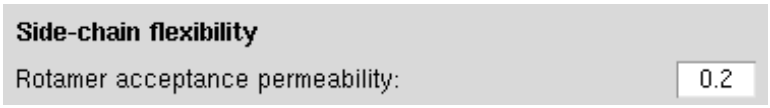


You can increase the resolution of the rotational degrees of freedom by reducing the degrees of angles and dihedrals.

Alternatively, you can also increase the resolution of the translational degrees of freedom by reducing the grid spacing.

2.6.5. Changing the sampling of rotamers

The users have the choice to add side-chains that will be made as flexible during the simulations.



In the initial steps of FlexAID, rotamers from the Penultimate rotamer library are evaluated. Only those rotamers that fulfill the van der Waals rotamer acceptance permeability will be accessible during the simulations.

More rotamers can be accessible if the user can allow more permeability by increasing the acceptance parameter. However, those rotamers will have more internal steric clashes with their surroundings and be penalized during docking.

Rather than using rotamers, users can choose to use rotamer instances representing side-chain conformers taken from a non-redundant dataset of protein-ligand holo structures. To use conformers, check 'Use rotamer instances'.

2.7. The Genetic Algorithm Params tab

*** This tab is for advanced users only*

To activate the "Genetic Algorithm Params" tab, click the corresponding tab.

The functionalities of this tab are to:

- Adjust the number of energy evaluations (length of simulations)
- Adjust the genetic operators
- Adjust the PyMOL visual display of docking simulations
- Change the fitness model
- Change the reproduction model

2.7.1. Adjusting the number of energy evaluations

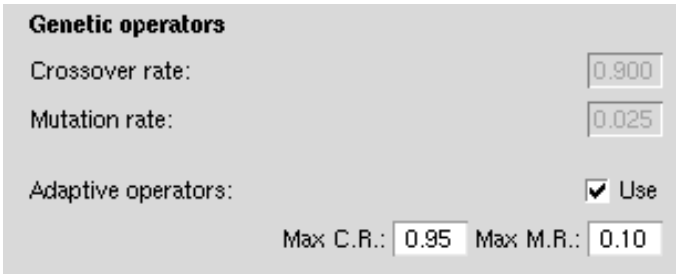
FlexAID uses genetic algorithms for covering the search space. By default, FlexAID evaluates 90,000 complexes (300 evaluations per generation).

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. You can increase the number of chromosomes and generations by modifying the corresponding values:

Alternatively, when the search space is small, you can decrease these values.

2.7.2. Adjusting the genetic operators

A genetic algorithm uses the crossover and mutation operators to sample the conformational space efficiently. You can modify the values of these operators in the following menu:

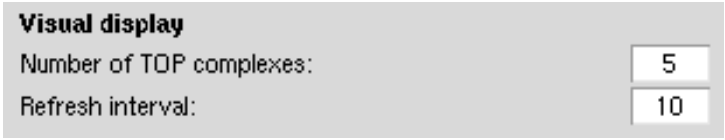


By default adaptive operators are used, i.e. the crossover and mutation rates are increased to a maximum (referred as Max C.R. and Max M.R.) when the whole population of chromosomes has converged. You can deactivate this feature by unchecking the "Use" button.

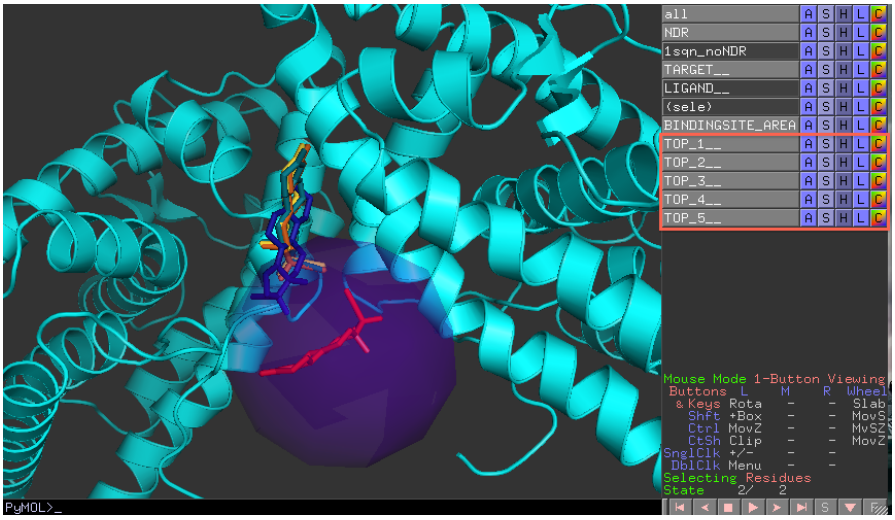
Constant operators will be used throughout the simulations independently of the status of the population.

2.7.3. Adjusting the PyMOL visual display of real-time docking

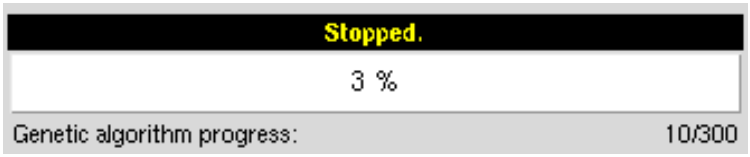
You can modify the visual display of real-time docking in PyMOL via the following menu:



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 lowest energy according to the scoring function.



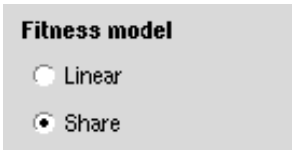
The "Refresh interval" represents the interval, in number of generations, at which the display of TOP complexes is refreshed. The progress of the genetic algorithm is shown in the Simulate tab (see section 2.8).

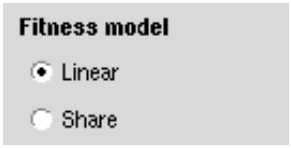


The FlexAID algorithm waits for the NRGsuite to display the TOP complexes before processing the next generation. Thus, we strongly suggest not displaying too many TOP complexes as it may drastically increase the computational runtime for a simulation.

2.7.4. Changing the fitness model

The selection of individuals (before reproduction, see Section 2.7.5) from the whole population during the genetic algorithms obeys to a fitness function. You can toggle the “Linear” and “Share” fitness models.

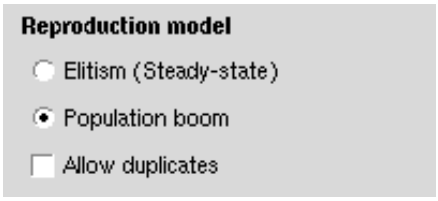




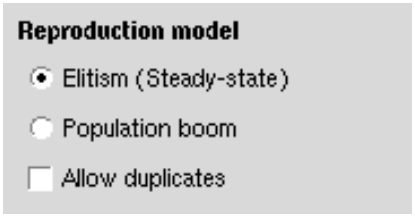
In the “Linear” model, the individuals with lowest energy are given a higher probability of being selected. In the “Share” model, individuals closely related in the search space are given a lower probability of selection. Thus, this model tends to favour unexplored areas of the search space. By default, the fitness model is “Share”.

2.7.5. Changing the reproduction model

Genetic algorithms generate new individuals by reproducing parent individuals selected by the fitness function (see Section 2.7.4). You can toggle between the "Population boom" and "Elitism" reproduction models in the following menu:

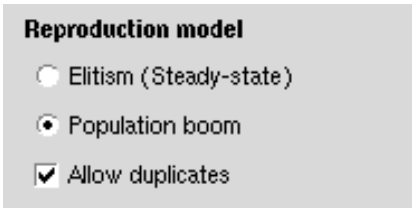


The default model "Population boom" generates extra N individuals from the current population (where N is the number of chromosomes, see Section 2.7.1), and conserves the best N individuals from the current and new populations for the next generation.



The "Elitism" model generates extra N individuals from the current populations and uses the new population for the next generation.

By default, FlexAID ensures that none of the individuals generates are duplicated, i.e. represents the same complex. If you want to allow duplicates, simply check "Allow duplicates". This might result in a slight gain in speed.



2.8. The Simulate tab

To activate the “Simulate” tab, click the corresponding tab.

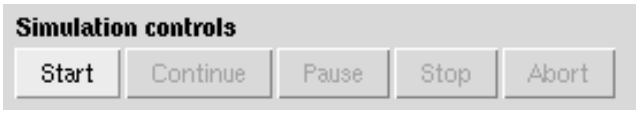
The functionalities of this tab are to:

- Control docking simulations
- Change the visual display of complexes
- Manage docking results

2.8.1. Executing a docking simulation

2.8.1.1. Controlling a docking simulation

To start a docking simulation, simply click “Start” button.

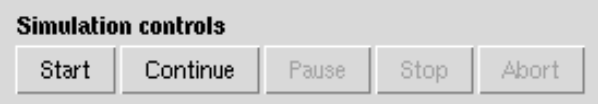


Once a simulation has successfully started, you have access to the “Pause”, “Stop” and “Abort” buttons.

The “Pause” button allows you to hold 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 of energy.

2.8.1.2. Continuing a docking simulation

FlexAID offers a way to continue a simulation from previously generated results. For more information on how to load existing results see Section 2.8.3. When results are activated, the "Continue" button may be clickable depending on the context.



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.

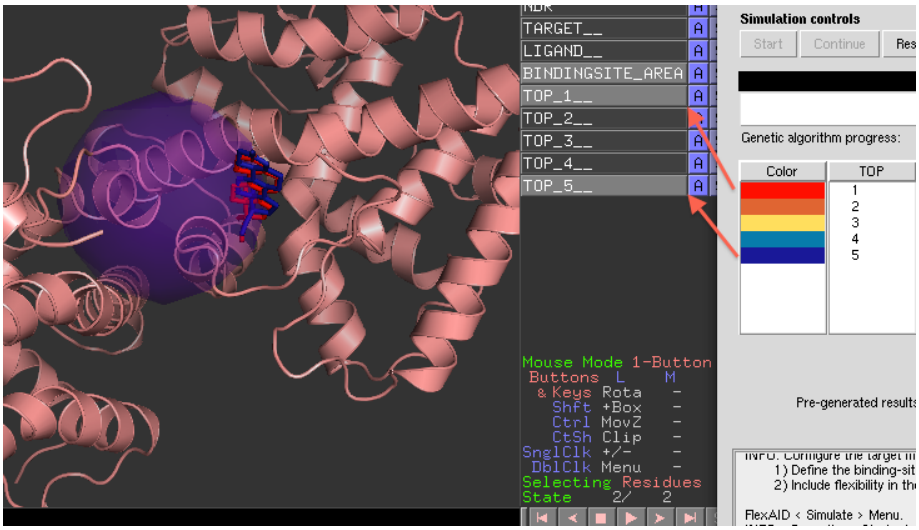
You can modify the parameters of the genetic algorithms and still be able to continue a simulation.

2.8.1.3. Viewing docking in real-time

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. When a reference is used (see Section 2.5.2), the RMSD is updated at every refresh interval (see Section 2.7.3).

Color	TOP	CF	Fitness	Last RMSD
	1	-184.831	15.421	N/A
	2	-147.600	15.019	N/A
	3	29.808	4.360	N/A
	4	239.428	5.703	N/A
	5	310.633	9.095	N/A

Reserved objects TOP_<X>__ (see Section 1.4) are created in order to visualize the docking simulation in real-time. The complexes are colored in the PyMOL viewer using the color code of the table.



2.8.1.4. Viewing the status of a simulation

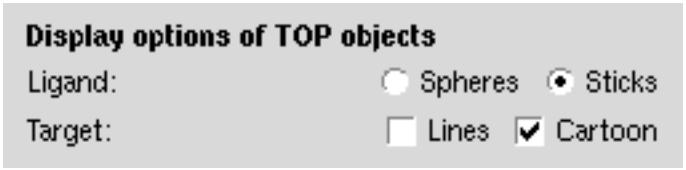
At all time, you may refer to the status bar to learn more on the status of the simulation…

and on errors that may occur (always in red).

2.8.2. Changing the visual display of complexes

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.



2.8.3. Viewing the docking results

When a simulation is finished or stopped, the reserved objects RESULT_<X>__ and RESULT_<X>_H_BONDS__ are created. RESULT_<X>__ are the complexes objects while the RESULT_<X>_H_BONDS__ objects are distances objects displaying the h-bonds for the result.

Their according energy value and RMSD from the reference (when one is used) of the results are displayed in the table:

Color	TOP	CF	Fitness	Last RMSD
	1	-188.75814	N/A	N/A
	2	-8.51084	N/A	N/A
	3	-3.00145	N/A	N/A
	4	28.63538	N/A	N/A
	5	100.69229	N/A	N/A
	6	142.05573	N/A	N/A
	7	159.09238	N/A	N/A
	8	172.52778	N/A	N/A
	9	214.92108	N/A	N/A

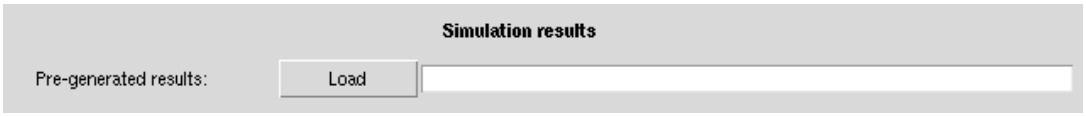
By default, the target is color coded using the colors in the table and is shown as cartoon. The ligand and other optimized residues (including side-chains) are shown in white as sticks surrounded by its residues shown as lines.

When constraints are applied, the extra column Apparent CF will appear.

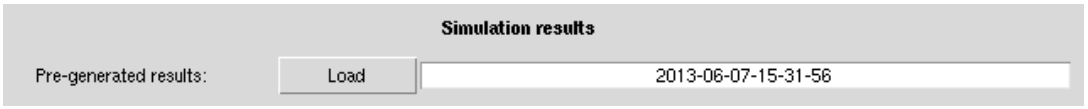
2.8.4. Loading existing docking results

The simulate tab also allows you to load previously generated results. To load existing results, simply press the "Load" button. The default directory when loading a result is FlexAID/Result/$COMPLEX (see section 3.3.1) of the currently active project (see section 1.3). $COMPLEX is the uppercase value of the combination of $TARGET and $LIGAND (see Sections 3.3.1 and 2.3.1.2).

A result loaded elsewhere from the suggested directory will not be copied into your Result directory.



Once a simulation has successfully ended, results are automatically saved with the starting date and time of the simulation. The results are now set as active. The results are saved into the FlexAID/Results/$COMPLEX folder.



You can view the report of a result by clicking the following button:

A report gives you a detailed list of the parameters of a given result. The report is opened with Notepad (Windows), Gedit or Kedit (Linux), and the text-associated application (MacOS X).

```
Target name                     1sqn_noNDR
Target file                     /Users/francisgaudreault/Documents/
NRGsuite/Example/Target/1sqn_noNDR.pdb
Ligand name                     BTN
Ligand file                     /Users/francisgaudreault/Documents/
NRGsuite/Example/FlexAID/Ligand/BTN.pdb
Method optimization             Genetic algorithms
Scoring function                Complementarity function
Binding-site definition         Sphere
Sphere center (x,y,z)           11.419500351,2.3710000515,-11.3724994659
Sphere radius                   4.0A
Binding-site volume             268.0832A^3
Binding-site resolution         0.375A
Target flexibility              No
Ligand flexibility              Yes
Number flexible bonds           1
Interaction constraints         No
Translational degree of freedom Yes
Rotational degree of freedom    Yes
Pose as reference               No
Atom types definition           Sybyl
Pairwise energy matrix          MC_10p_3.dat
Van der Waals permeability      10.0%
Delta angle                     2.5 degrees
Delta dihedrals                 2.5 degrees
Delta flexible dihedrals        10.0 degrees
Solvent type                    No-type
Solvent term                    5.0
Maximum docking results         10
```

3. The GetCleft interface

3.1. Getting started

GetCleft is an algorithm that identifies the cavities (or clefts). The GetCleft interface allows to adjust the parameters of GetCleft as well as to partition existing clefts and calculating their volume.

The GetCleft interface contains a menu from which you can save/load clefts. The interface contains 3 tabs: *Generate, Partition* and *Volume*. The functionalities of each tab is reviewed in details in the following sections.

A message box appears to hint you as going through the interface as well as informing of any errors that may occur.

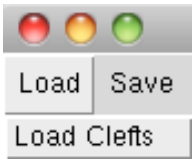
You can reset the default values in the active tab (highlighted in blue) by clicking the "Default" button. You can close the GetCleft interface by clicking the "Close" button.

3.2. Working with clefts

Cleft objects represent the cavities of a target. Clefts are particularly useful when defining the binding-site within the FlexAID interface (see section 2.4.1.2) as clefts represent potential binding-sites for a ligand.

3.2.1. Loading clefts

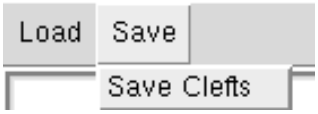
You can load previously saved clefts by clicking the “Load Clefts” button in the menu.



The default directory when loading a binding-site is Clefts/$TARGET (see section 3.3.1) of the currently active project (see section 1.3). A cleft loaded elsewhere from the suggested directory will not be copied into your Cleft directory.

3.2.2. Saving clefts

Clefts are not automatically saved when generated (see section 3.3.1). 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 and re-open it, the clefts will not be available for saving.



The default directory when saving a cleft is Clefts/$TARGET (see section 3.3.1) of the currently active project (see section 1.3). You cannot save a cleft elsewhere from the suggested directory.

3.3. The Generate tab

To activate the “Generate” tab, click the corresponding tab.

The functionalities of this tab are to:

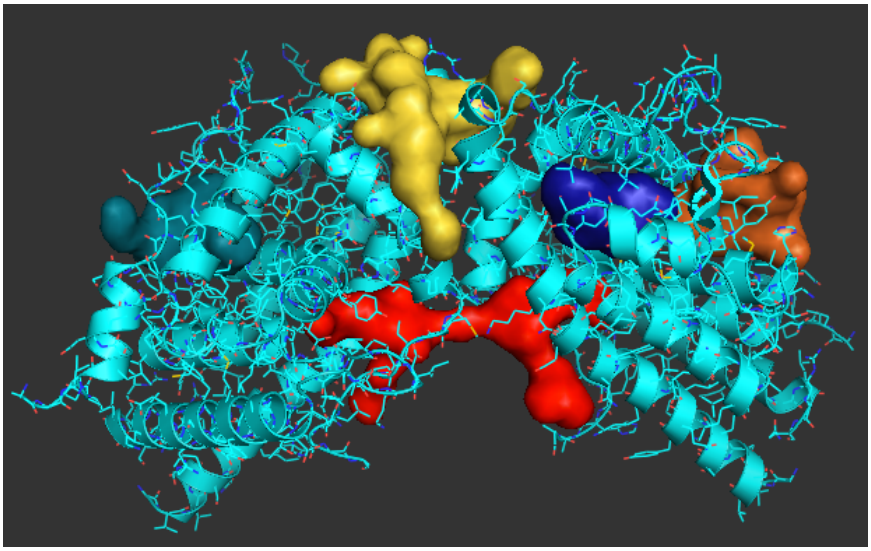
- Generate the clefts of a target
- Filter the generated clefts
- Control the volume of the clefts

3.3.1. Generating clefts

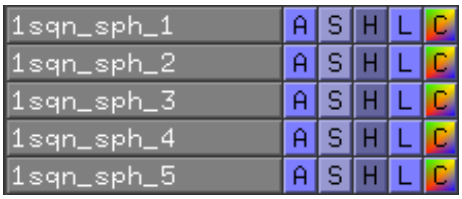
The initial step in generating clefts is to select the target of interest from the existing PyMOL object/selection by clicking its name in the widget (see section 1.5). This target is referred as $TARGET.

Upon selection of the target, simply click the "Start" button.

When clefts are generated, they are automatically loaded into the PyMOL viewer using a unique color code as well as object name.



Each object is named $TARGET_sph_<X>, where <X> represents a numerical value.



Specifically, clefts are ordered in decreasing number of spheres inserted (see section 3.3.4), which can somewhat be correlated in terms of volume.

3.3.3. Filtering generated clefts

You can apply filters before generating the clefts to:

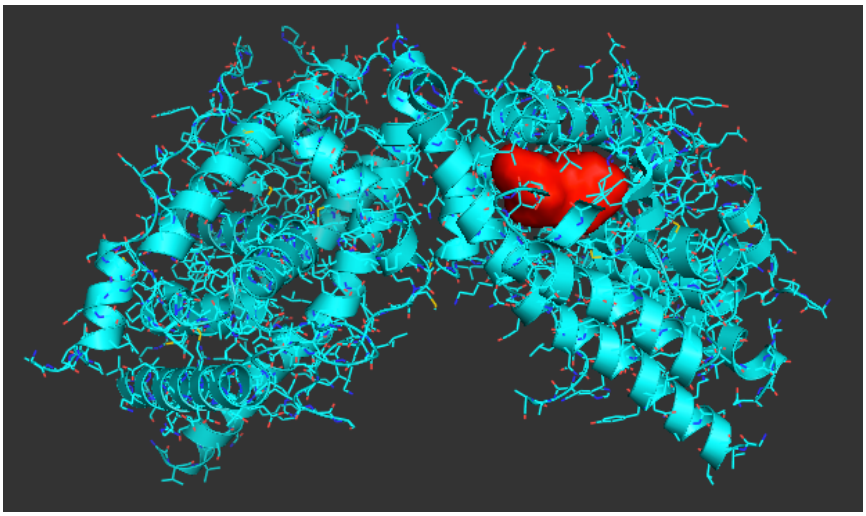
- Select only clefts in contact with a residue
- Limit the number of clefts

3.3.3.1. Filtering generated clefts using a residue code

You can filter clefts by selecting only those in contact with a residue. To do so, simply enter its residue code (see section 2.4.2.2).

A residue in contact is not restricted an amino acid or nucleic acid, as it may also specify a metal, a ligand or a water molecule (e.g. MG1A, HOH103C).

Applying this filter may return no clefts if none of the clefts are in contact with the residue. Otherwise, only clefts in contact are shown.



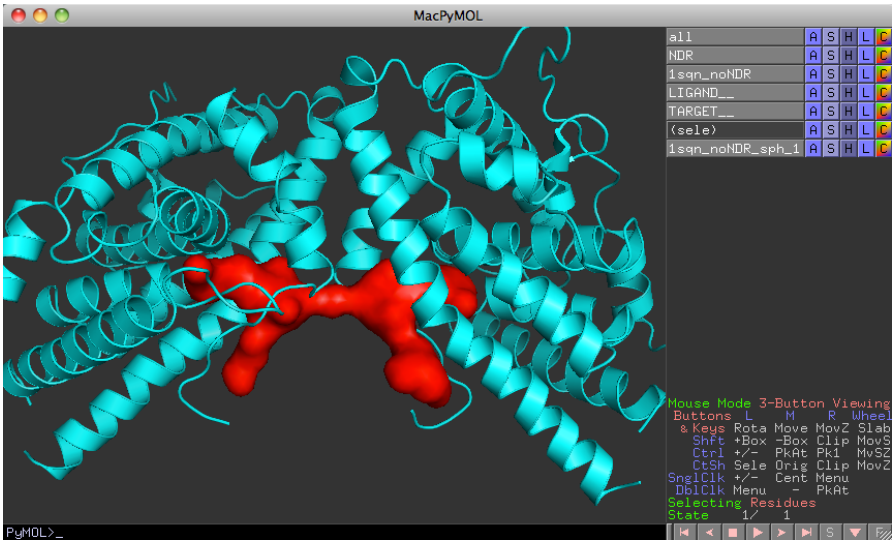
3.3.3.1. Limiting the number of clefts

You can limit the number of clefts that are generated by modifying the following parameter:

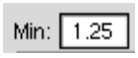
When the value inputted exceeds the apparent number of clefts, all apparent clefts are shown.

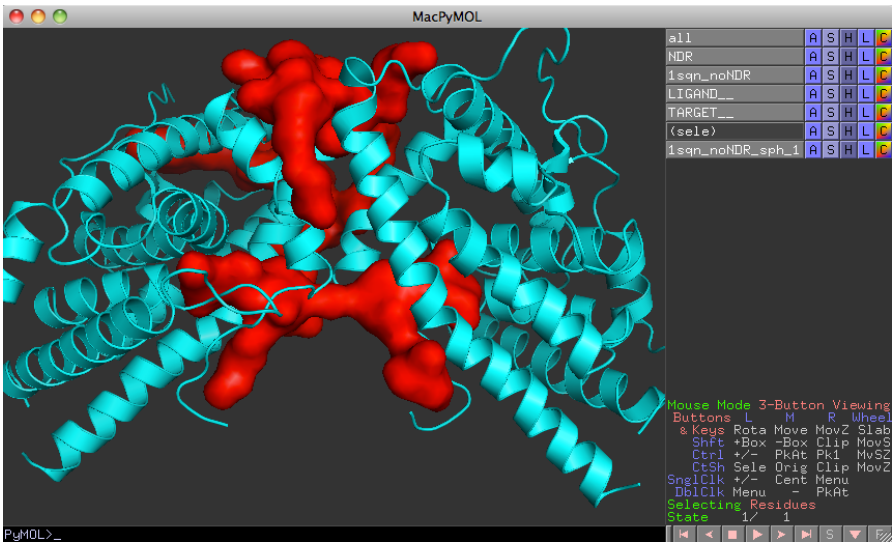
3.3.4. Controlling the volume of generated clefts

GetCleft detects the cavities of a target by introducing spheres between pairs of target atoms and reduces 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 1.50A to a Maximum 4.00A.



Reducing the minimum radius allows to insert spheres in deeply buried cavities. Thus, clefts generated with lower minimum are larger.





3.4. The Partition tab

To activate the “Partition” tab, click the corresponding tab.

The functionalities of this tab are to:

- Partition a cleft

3.4.1. Partitioning a cleft

It is possible to partition the volume of an existing cleft in the case where the cleft is too large for our needs.

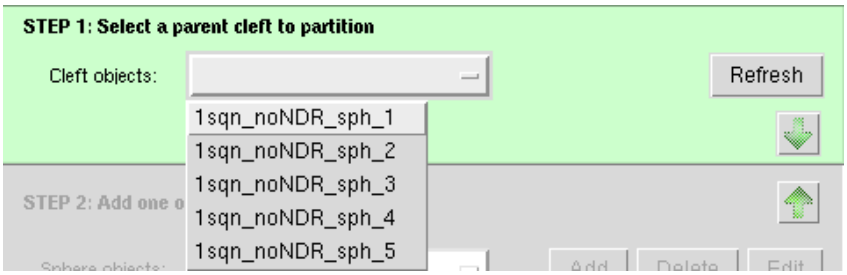
Partitioning a cleft consists in 3 easy steps:

- Selecting the cleft to partition
- Adding spheres to partition
- Creating the partition

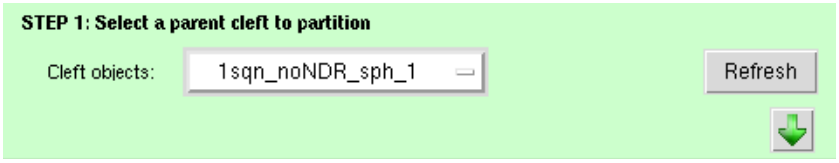
You can toggle between the differents steps by using the green arrows:

3.4.1.1. Selecting a 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.

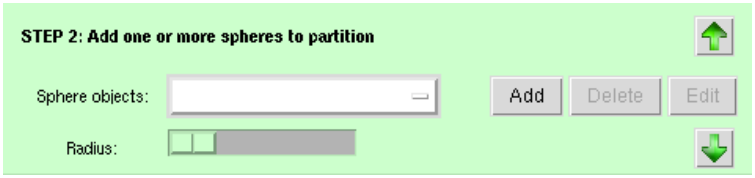


The selected cleft will flash in the PyMOL viewer.

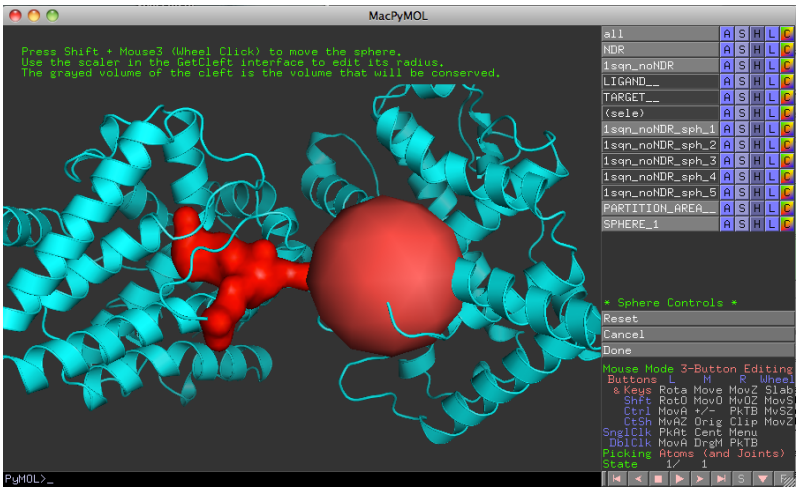


3.4.1.2. Adding spheres to partition

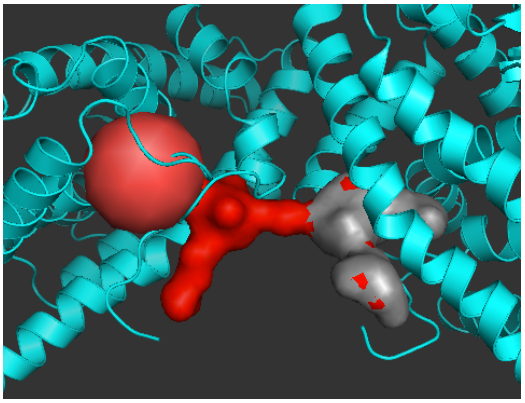
To partition a cleft you need to add one or multiple partition sphere(s). The cleft volume contained into the volume of the spheres is conserved.



To add a sphere, click the "Add" button. A wizard will active 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. To move and resize the sphere, refer to section 2.4.1.1.

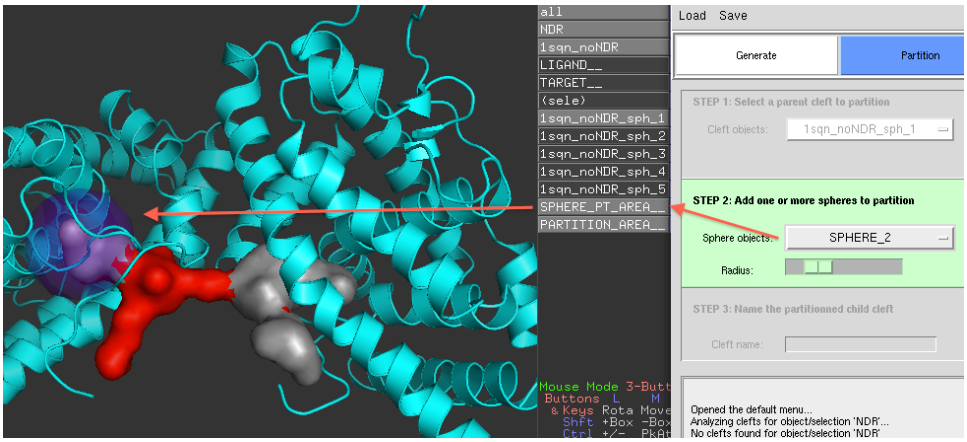


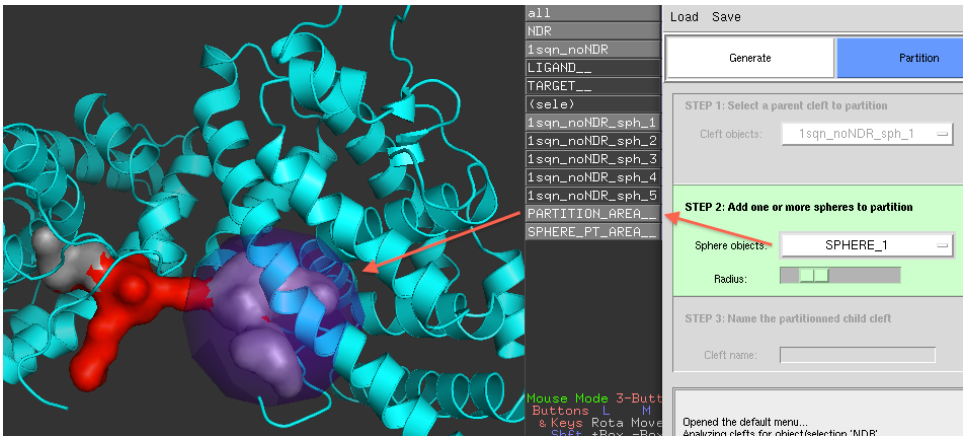
Once you set the size and location of the partition sphere properly, click the "Done" button in the Wizard. The partition sphere will appear in the Spheres objects list. If you wish to modify this sphere simply click the "Edit" button. A new wizard will activate. You can also delete partition spheres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



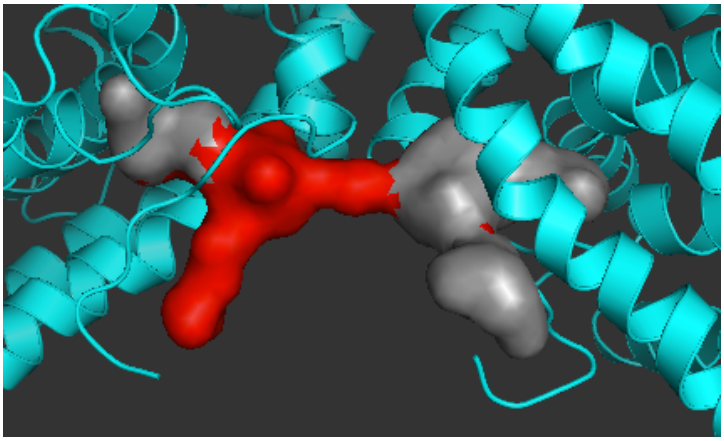
A second partition sphere is added. The sphere is also appended to the sphere objects list.

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.



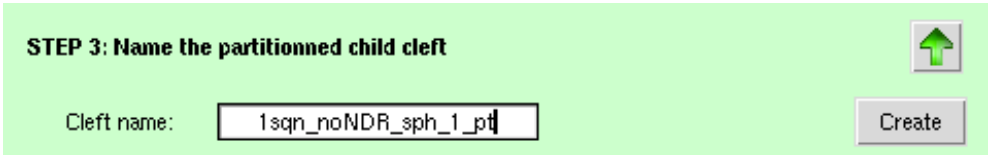


The greyed out volume of the cleft represents the volume that will be conserved when the partitioned cleft is created.



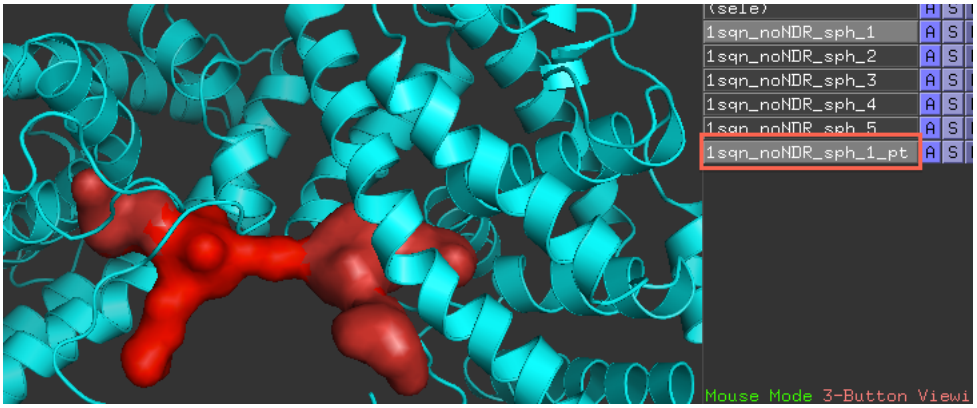
3.4.1.3. Creating the partition

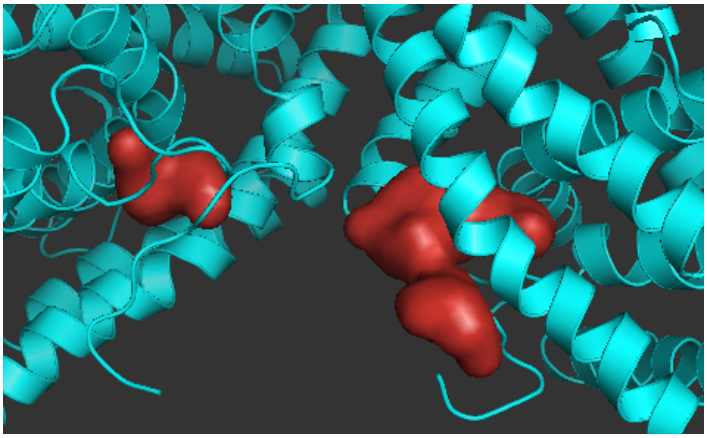
You can create a partition from the following menu:



Name your partition and click “Create”. The partitioned cleft will appear as a darker color than its parent cleft.

** You can no longer rename the partition in recent versions of the NRGsuite.





3.5. The Estimate tab

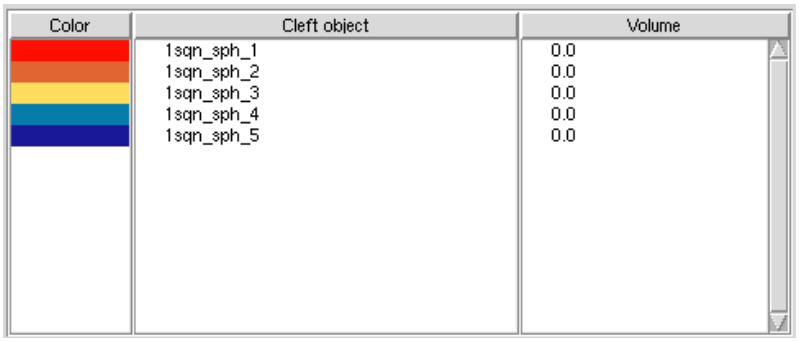
To activate the “Estimate” tab, click the corresponding tab.

The functionalities of this tab are to:

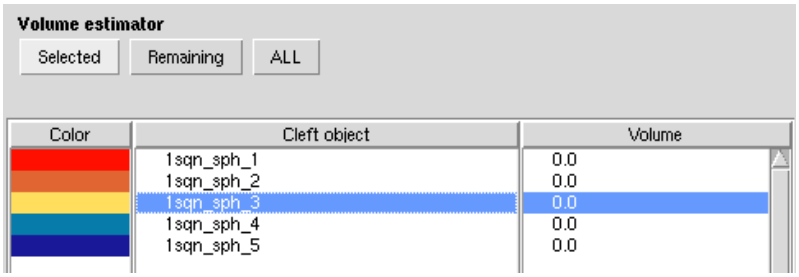
- Calculate the volume of clefts

3.5.1. Estimating the volume of clefts

The clefts 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 uncalculated volume has a null volume (0.0).



You can estimate the volumes of all clefts (“ALL” button), uncalculated clefts (“Remaining” button) or of a selected cleft (“Selected” button), e.g.:



The calculated volume is displayed upon completion.

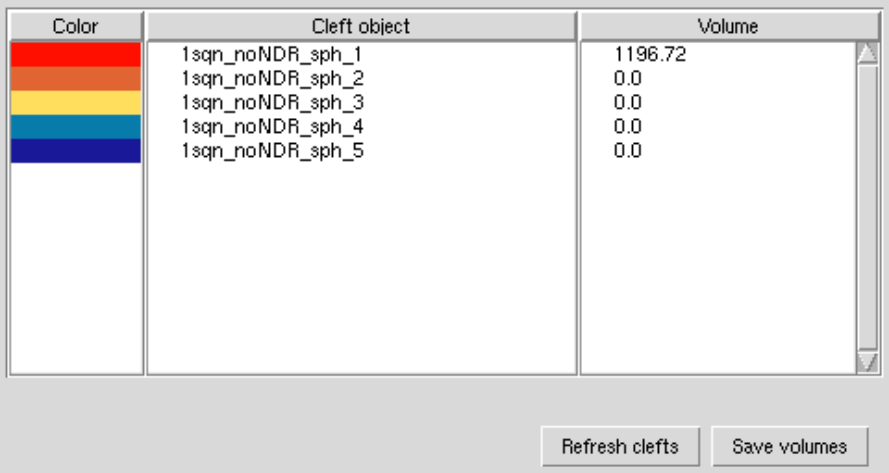
3.5.1.1. Increasing the resolution of calculations

The calc_volume algorithm uses an incremental procedure to determine the volume of clefts. You can increase the level of precision by increasing the iterations. You can adjust the number of iterations by modifying the following field:

However, the computational runtime is increased with higher number of iterations. On less-performing systems, you may decrease that number.

3.5.1.2. Saving the volumes

If you wish to save the calculated volumes for later without having to re-calculate them, you can explicitly save them by clicking “Save volumes”.



Saving the volumes requires you to save the clefts before. For more information on how to save clefts, see section 3.2.2.