

AQUA  DUCT by T  G

TIPS & TRICKS - smoothing paths

Smoothing paths

This section will give you an insight into the method of paths smoothing that you should apply in order to smooth the inlet paths that you analyse. A trajectory of a solvent molecule inside a macromolecule's core is usually very complicated and hard to follow. The molecule during passage through the protein core can visit different cavities and meanders of the protein interior. It can be simplified by smoothing in order to improve its visualisation and approximations of the central line of a particular tunnel used by the molecule. Later on, such approximation can be utilised for energy profile construction. Current AQUA-DUCT version allows *soft* smoothing only, i.e. smoothing is used only for visualization purposes. During smoothing, the number of points (frames) in the smoothed path is the same as in a raw path. So the analysis of the smoothed path is done on the same number of points as in the raw path. However, for the purposes of visualization, some points in the path are removed. Only those points are removed which are aligned on the result smoothed path, so that no bias is introduced and removal of those points does not change the smoothed path's appearance. The smoothing is also performed proportionally, i.e. in the end, the distance between individual points will more or less give information about the time it took the molecule to travel a given section.

Studying smoothed paths can help to better understand solvent molecule's routes during the simulation, their direction and sites where they are stuck for some time due to either attraction by protein's amino acids or blocking by flexible parts of macromolecule's interior.

Generally, there are four groups of smoothing methods applied in AQUA-DUCT:

1. Window methods - `window`, `dwin`, `awin` - methods based on producing mean or median values in frames (windows) along the path.
2. Maximal step method - `mss` - method which creates an average of coordinates in a given range of path steps.
3. Combined methods - `window_mss`, `dwin_mss`, `awin_mss` - a combination of window- and mss-based methods. First, `mss` smoothing is performed and then window smoothing.
4. Savitzky-Golay filter - `savgol` - a distinct method based on a process known as convolution, which fits a polynomial to subsets of points along the path.

Case study

As an example case study we are using one of the water inlet's paths found by AQUA-DUCT during a 50 ns simulation of Cytochrome (PDB ID: 2v0m) in a water box. The raw path is visualised within the protein shown as surface (**Figures 1a, 1b**). Multiple voids and pockets are visible in the protein's interior. The chosen path leads through voids displayed on the images, creating an S-shaped trajectory (**Fig. 1a**). The chosen path leads from the protein's interior out of the object (yellow) to its exterior (outgoing part of the path - blue). A clean part of the path marks places where the molecule has quickly passed through the core and a dense, squigly and mixed up part of the path marks where it was trapped (**Fig. 1b**). The raw path is shown (**Fig. 1b**) as a reference to the results of testing of smoothing methods provided by AQUA-DUCT.

The raw path has a green, a yellow and a blue part of the path. The green part of the path marks the *within Object section of the trajectory*. It is not visible, though, since it is overlaid by the yellow one. The yellow part marks the *out of object* section of path, where the molecule left the defined *Object* but stayed within the *Scope*. The blue part marks the *outgoing* section of path, where the molecule left the *Object* and then the *Scope*.

After application of smoothing methods, the yellow part of the path changes its status to green - *within Object*, for simplification of the visualisation. The yellow part of the path marks the section of the molecule's trajectory where it leaves the *Object*, but stays within the *Scope*. Therefore, the yellow part of the path is always between two green parts - *within Object*. The purpose of smoothing paths in AQUA-DUCT is to approximate a molecule's way through the main tunnel, therefore the information about its trapping in one of the side pockets can be omitted during the process. Finally, the yellow part of the path can be removed completely and only the green one remains.

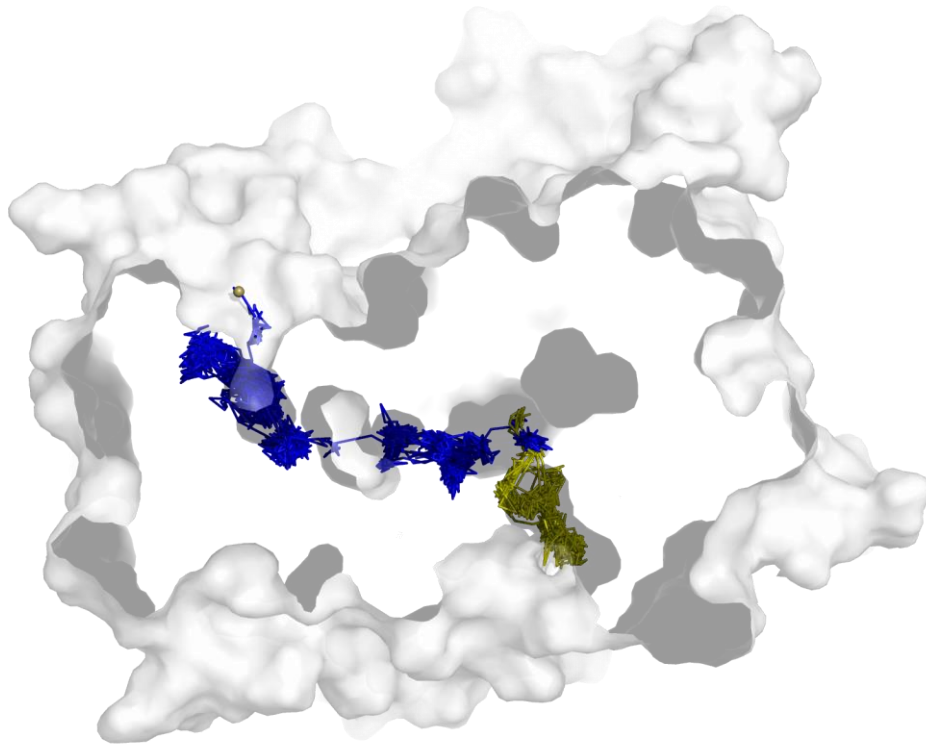


Fig. 1a. Visualisation of the entire protein shown as surface with selected water inlet's path and inlet. Inside of the protein are visible internal voids.

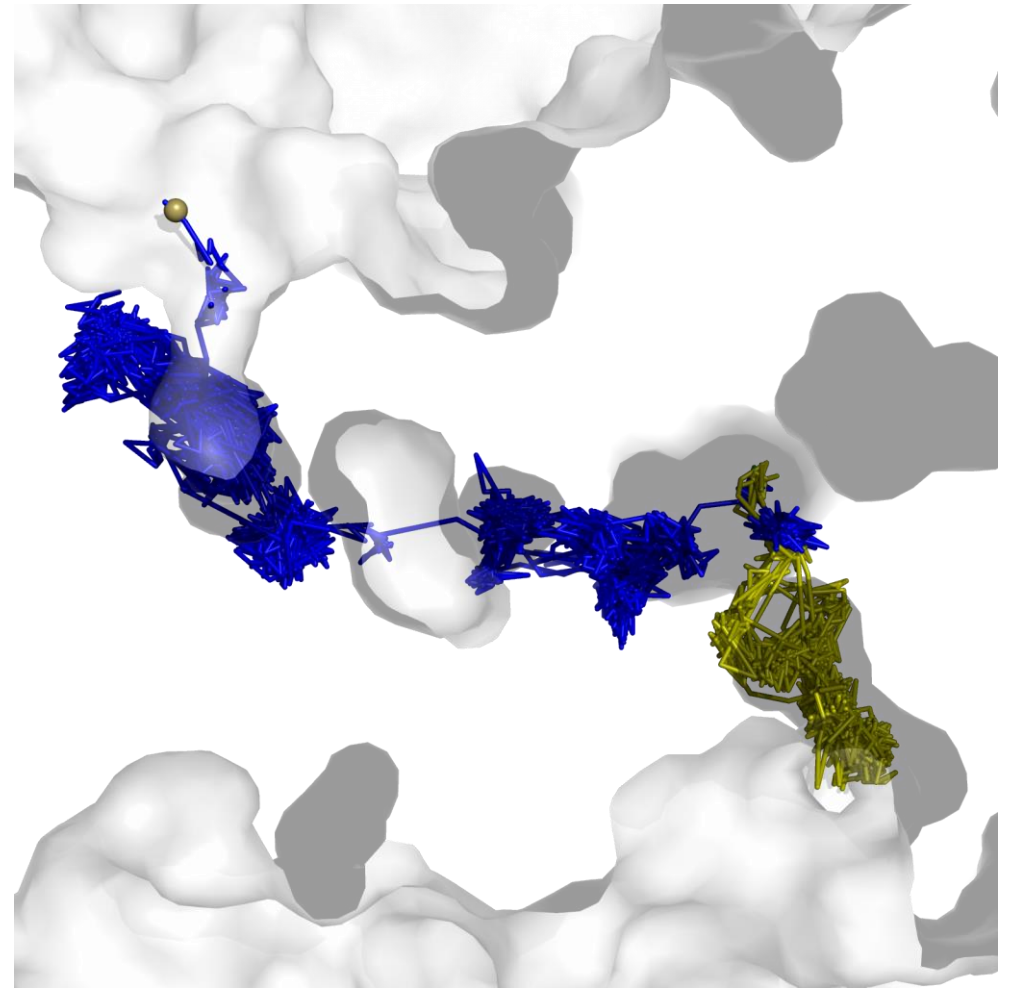


Fig. 1b. Visualisation of the part of the protein shown as surface with selected water inlet's path and inlet. Path leads through the internal voids in protein.

window

Defined size window smoothing works by creating a window of defined size (at the edges of the path the window is truncated). Next, all coordinates within a window are averaged with a function defined as `function` which can be `mean` or `median`. Those resulting values are the smoothed coordinates. The method has options `function`, `recursive` which is a number of recursive runs of smoothing method and `window` which is the window size. The default value for `window` is 5. The value of the `window` has to be an int number.

Increasing `recursive` results in gradually increasing the smoothness of path. The smoothed path is more clear and there are less squiggly parts where the molecule was trapped. Simultaneously, the original shape of the path is preserved resulting in a clearer, yet very informative path. Application of increased `recursive` (**Fig. 2a-f**) as well as `window` (**Fig. 2a, 2g-i**) results in a more smoothed path. The greater the value, the smoother the path. However, setting `function` to `median` did not result in observable differences (**Fig. 2e, 2f, 2i**).

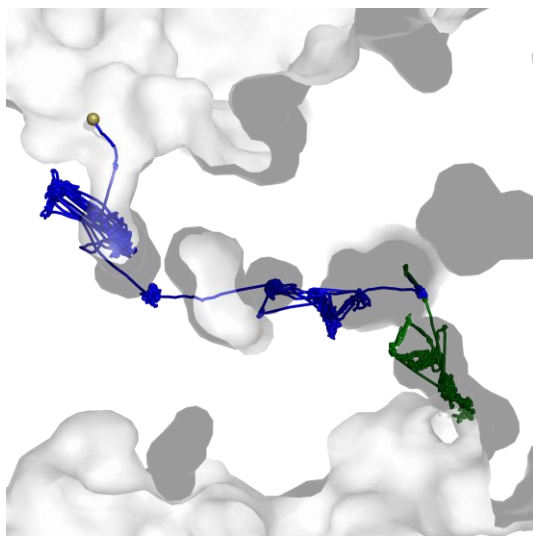


Fig. 2a. `[smooth]`
`method = window`
`recursive = 1`
`window = 5`
`function = mean`

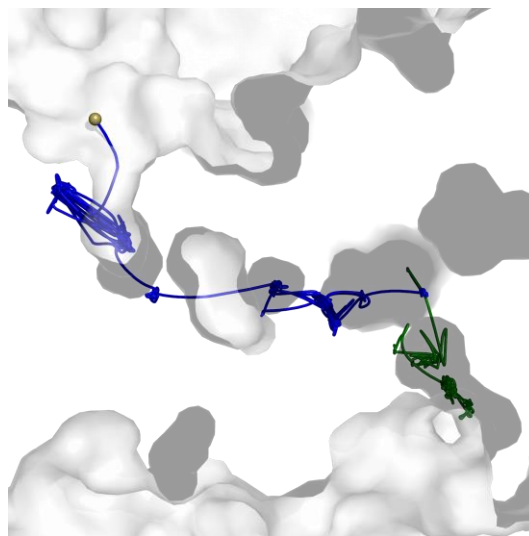


Fig. 2b. `[smooth]`
`method = window`
`recursive = 5`
`window = 5`
`function = mean`

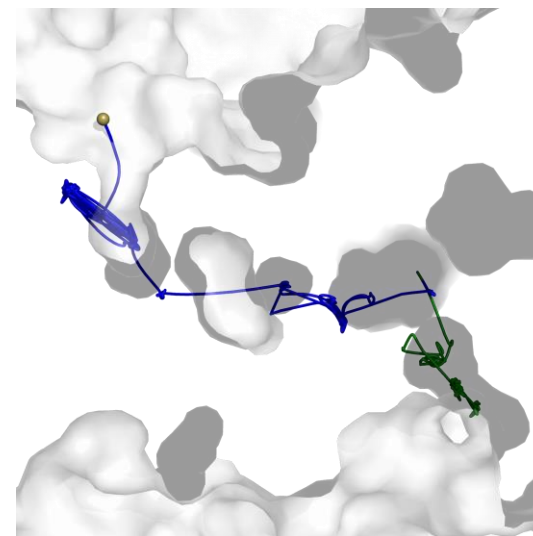


Fig. 2c. `[smooth]`
`method = window`
`recursive = 25`
`window = 5`
`function = mean`

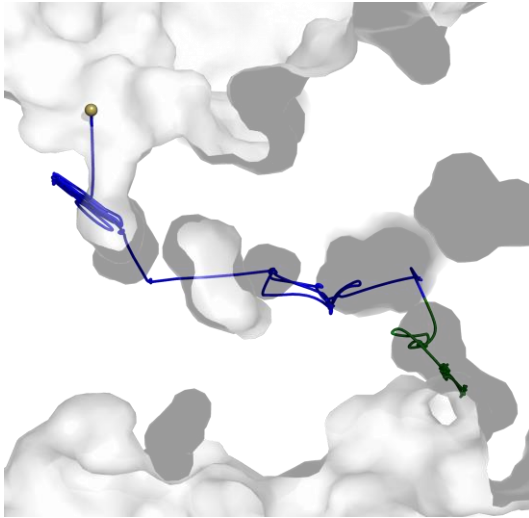


Fig. 2d. [smooth]
method = window
recursive = 100
window = 5
function = mean

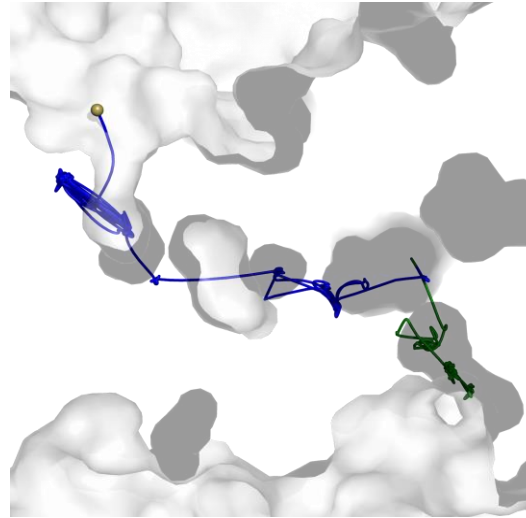


Fig. 2e. [smooth]
method = window
recursive = 25
window = 5
function = median

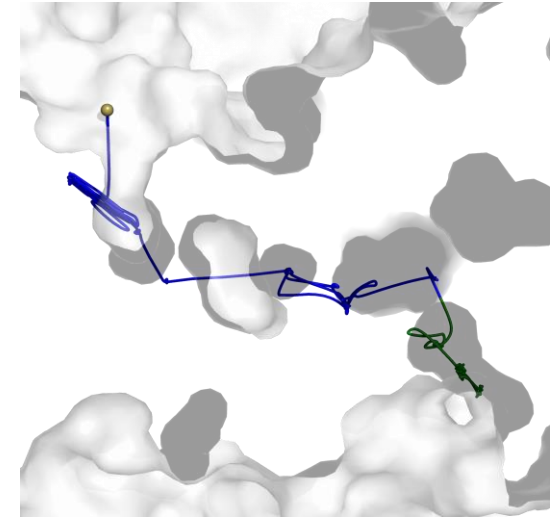


Fig. 2f. [smooth]
method = window
recursive = 100
window = 5
function = median

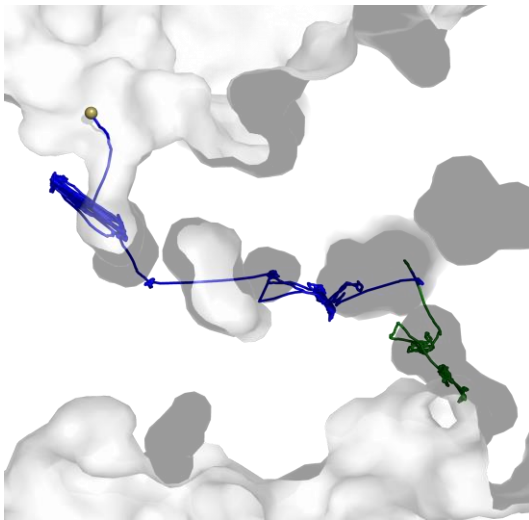


Fig. 2g. [smooth]
method = window
recursive = 1
window = 25
function = mean

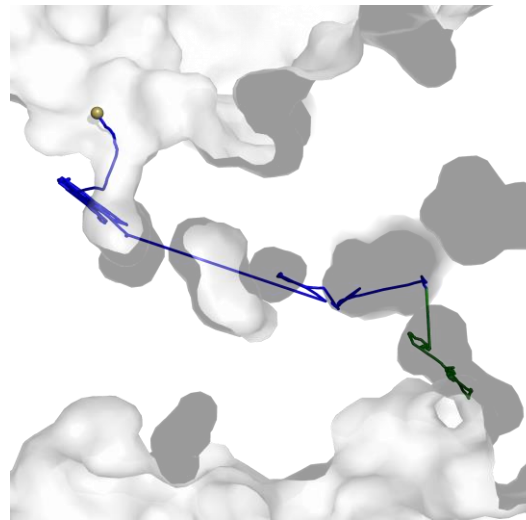


Fig. 2h. [smooth]
method = window
recursive = 1
window = 100
function = mean

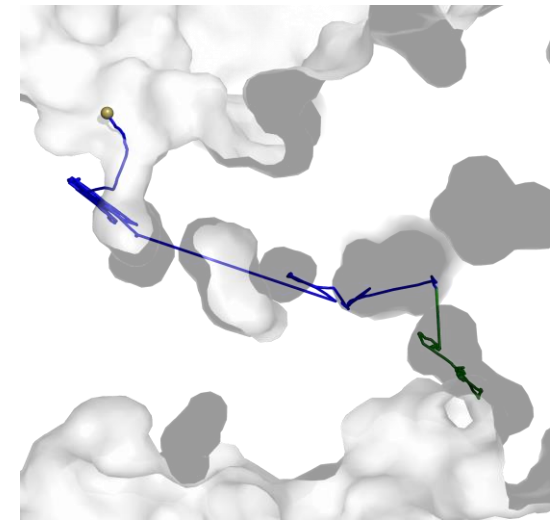


Fig. 2i. [smooth]
method = window
recursive = 1
window = 100
function = median

dwin

Distance defined size window smoothing is a modification of `window` method. The difference is the definition of the window size, which is here an average distance between points of input coordinates. In the first step, an average distance between all points is calculated and this value is used to calculate the actual window size. In the next step, for each coordinate a window of size calculated in the first step is created. Finally, all coordinates within a window are averaged with a function defined as `function` which can be `mean` or `median`. Those resulting values are the smoothed coordinates. The method has options `function` and `recursive` which is a number of recursive runs of smoothing method.

Increasing `recursive` results in quick simplification of the path. When `recursive` ≥ 5 (**Fig. 3c-f**) the path no longer gives information about sites where the molecule got stuck and does not even clearly show through which voids it passed. Using the `dwin` smoothing method with option `recursive` you have to be careful not to lose too much data. Setting `function` to `median` did not result in observable differences (**Fig. 3c, 3f**).

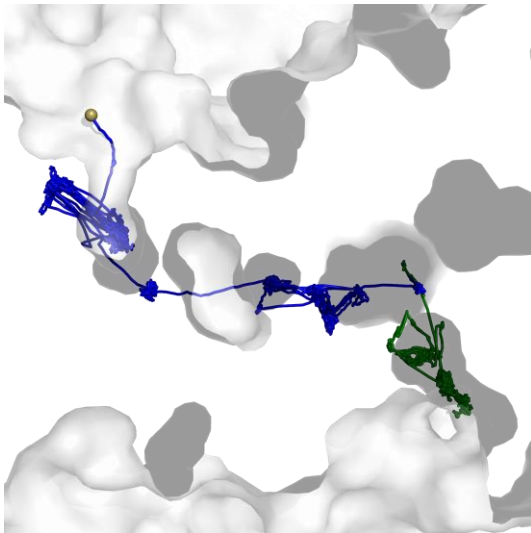


Fig. 3a. `[smooth]`
`method = dwin`
`recursive = 1`
`function = mean`

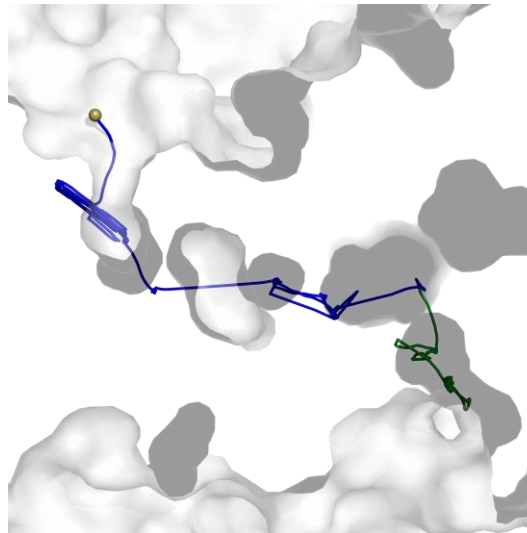


Fig. 3b. `[smooth]`
`method = dwin`
`recursive = 2`
`function = mean`

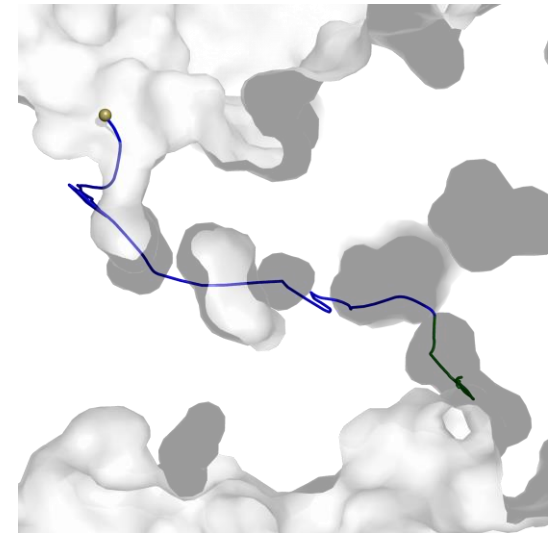


Fig. 3c. `[smooth]`
`method = dwin`
`recursive = 3`
`function = mean`

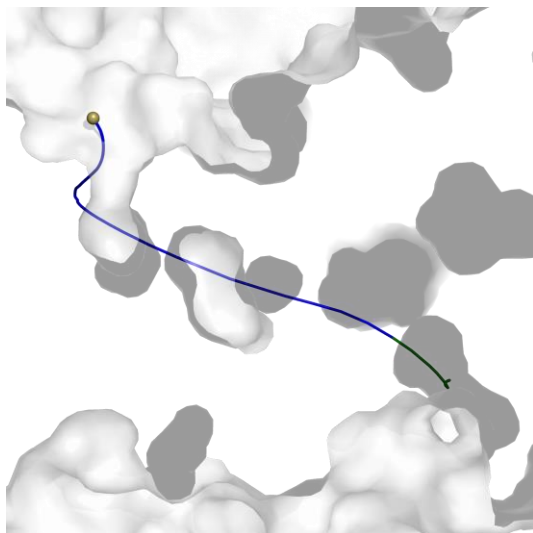


Fig. 3d. [smooth]
method = dwin
recursive = 5
function = mean

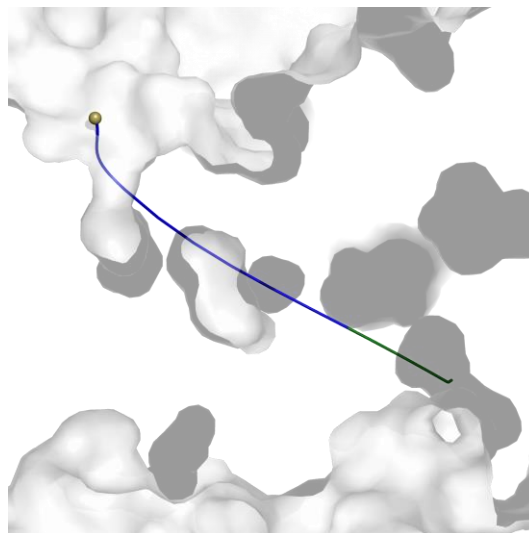


Fig. 3e. [smooth]
method = dwin
recursive = 15
function = mean

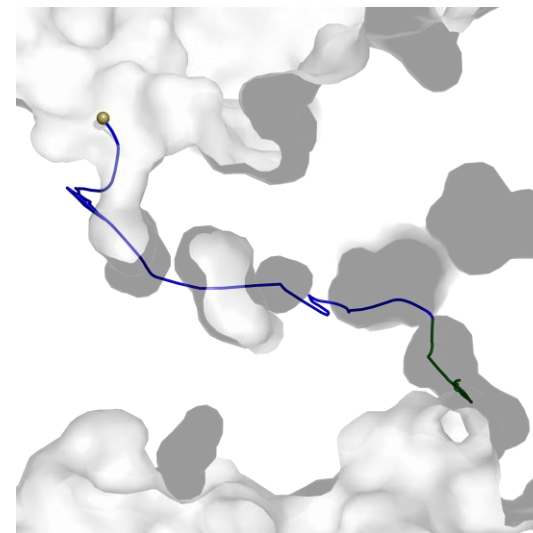


Fig. 3f. [smooth]
method = dwin
recursive = 3
function = median

awin

Active size window smoothing is a method similar to `dwin`, since the window size is defined also as a distance. The difference is that the actual window size is calculated for each point separately. In the first step, for each coordinate the window is calculated by examining the distance differences between points. In the next step, all coordinates within the window are averaged with a function defined by `function` which can be `mean` or `median`. Those resulting values are the smoothed coordinates. The method has options `function` and `recursive` which is a number of recursive runs of smoothing method.

In this method also, compared to method `dwin`, increasing the value of `recursive` results in quite abrupt simplification of the path. When `recursive` ≥ 5 (**Fig. 4c-f**) the path no longer gives information about sites where the molecule got stuck and does not even clearly show through which voids it passed. Using `awin` smoothing method with option `recursive` you have to be careful not to lose too much data. Setting `function` to `median` did not result in observable differences (**Fig. 4c, 4f**).

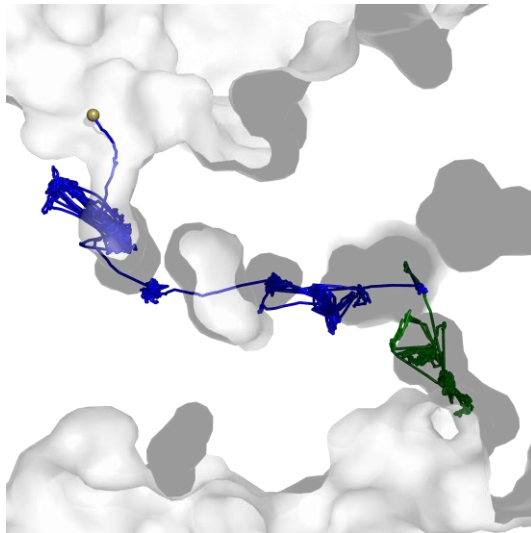


Fig. 4a. [smooth]
method = awin
recursive = 1
function = mean

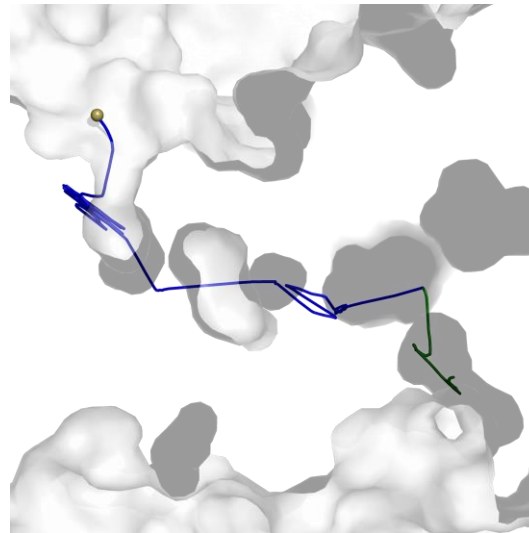


Fig. 4b. [smooth]
method = awin
recursive = 3
function = mean

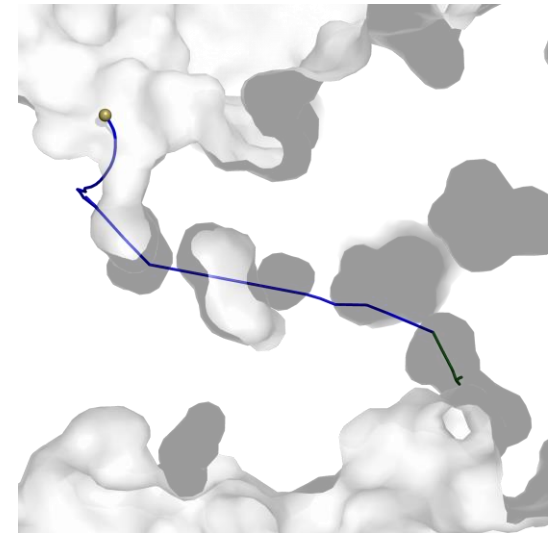


Fig. 4c. [smooth]
method = awin
recursive = 5
function = mean

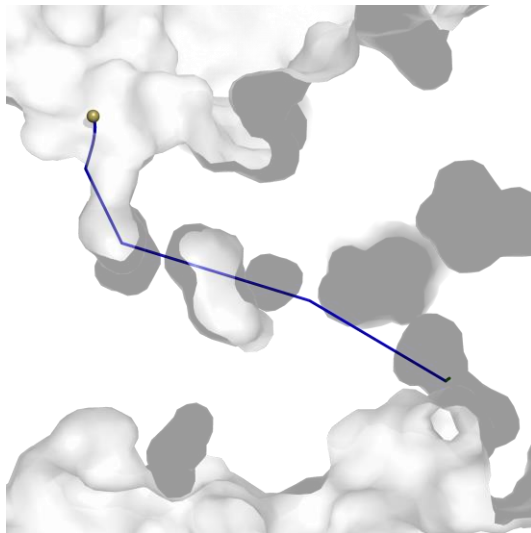


Fig. 4d. [smooth]
method = awin
recursive = 15
function = mean

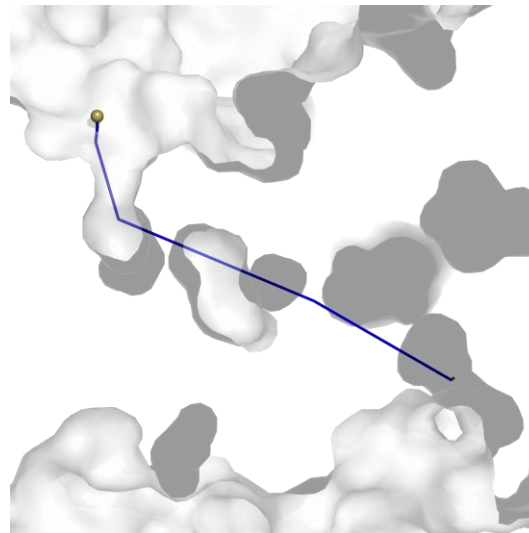


Fig. 4e. [smooth]
method = awin
recursive = 25
function = mean

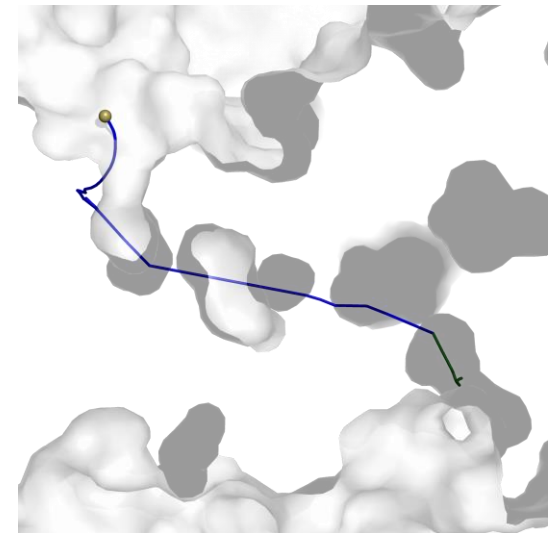


Fig. 4f. [smooth]
method = awin
recursive = 5
function = median

mss

Maximal Step smoothing is a method that moves through the coordinates and calculates distance over the traversed path. In the first step, if a point is found within the `step`, then it is added along with the beginning and end of the path to the set of “cardinal points”. Next, all cardinal points and points of linear interpolation between cardinal points are returned as smoothed coordinates. Number of interpolated points is in accordance with points skipped between cardinal points. The method has options `recursive` which is a number of recursive runs of smoothing method and `step` which defines the size of the step.

Increasing `recursive` without changing the `step` does not result in any changes (**Fig. 5b, 5c**). This is due to the fact that the default value for `step=1` and all the points are taken into consideration as cardinal points in every iteration and increasing the number of iterations of the algorithm will not affect the results.

Increasing only `step`, however, results in simplification of the path, but not necessarily its smoothing. The output path is very shattered, since parts of the path between the cardinal points are interpolated by a line (**Fig. 5d-f**). You have to be careful not to set `step` to a too great a value, since it may bias the result. On **Fig. 5f** the path is too simplified and has a pointy end leading to the inlet through the protein core rather than the void that was actually used for the transport of this molecule.

Combining `recursive` and `step` also resulted in shuttered output path (**Fig. 5g-i**) This method does not really smoothen the path, rather simplifies it and transforms it into points connected by straight lines. Using `mss` smoothing method you have to be careful not to lose too much data.

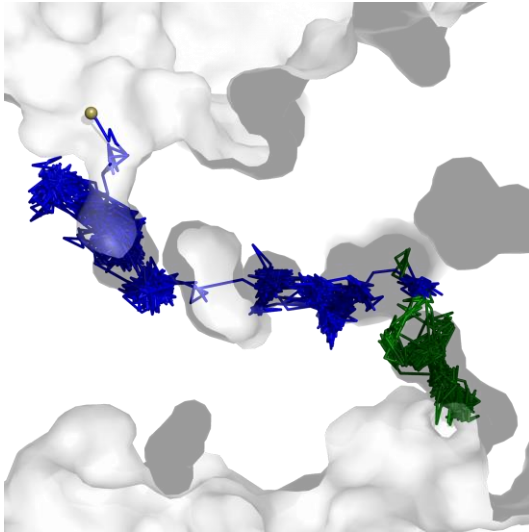


Fig. 5a. [smooth]
method = mss
recursive = 1
step=1

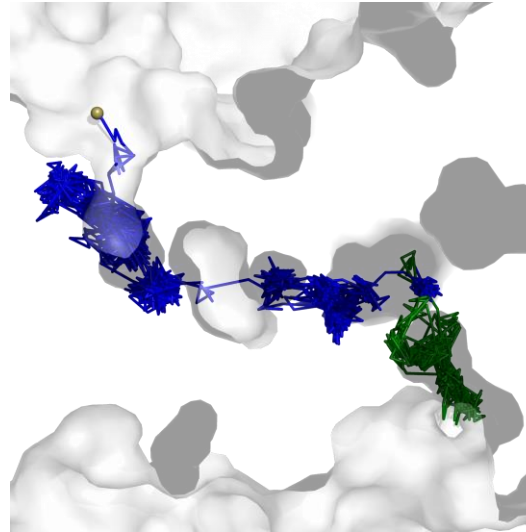


Fig. 5b. [smooth]
method = mss
recursive = 5
step=1

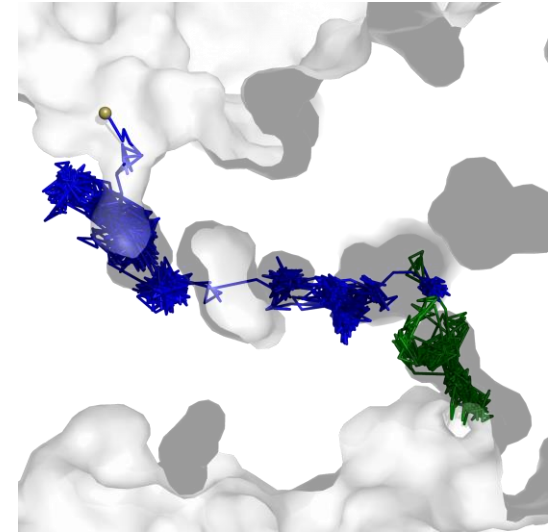


Fig. 5c. [smooth]
method = mss
recursive = 100
step=1

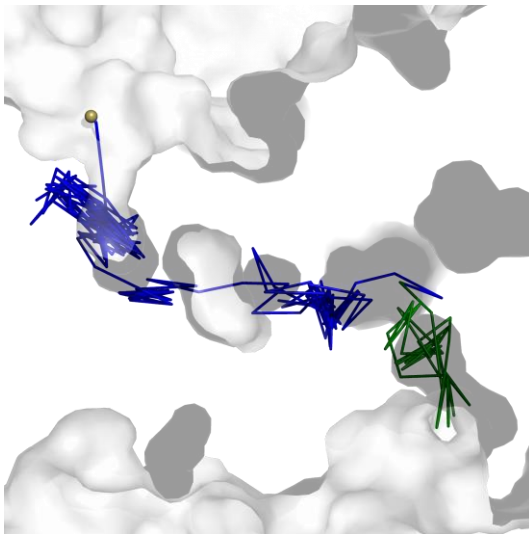


Fig. 5d. [smooth]
method = mss
recursive = 1
step = 3

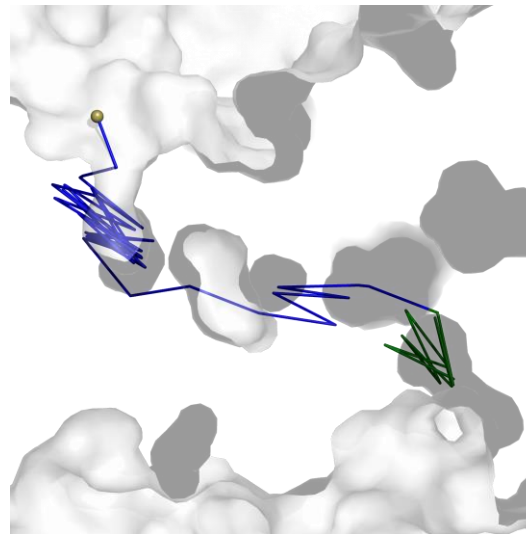


Fig. 5e. [smooth]
method = mss
recursive = 1
step = 5

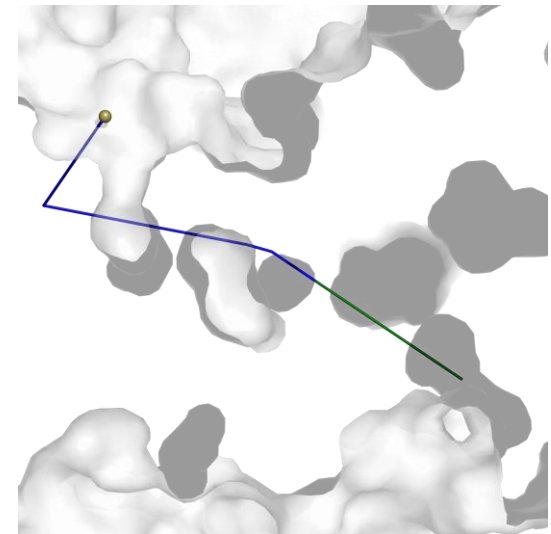


Fig. 5f. [smooth]
method = mss
recursive = 1
step = 15

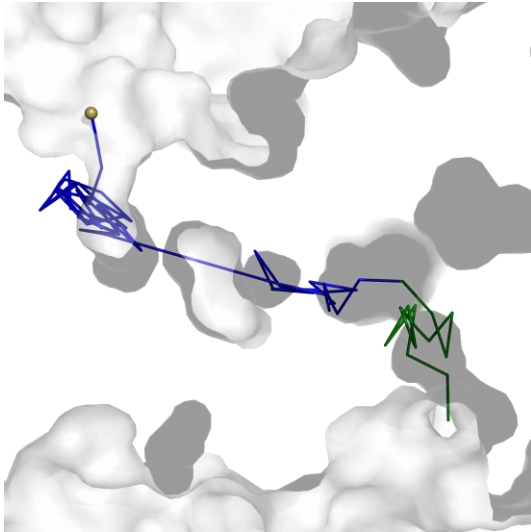


Fig. 5g. [smooth]
method = mss
recursive = 3
step = 3

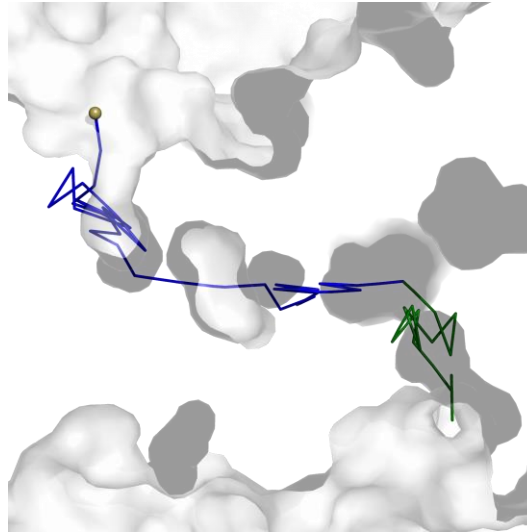


Fig. 5h. [smooth]
method = mss
recursive = 15
step = 3

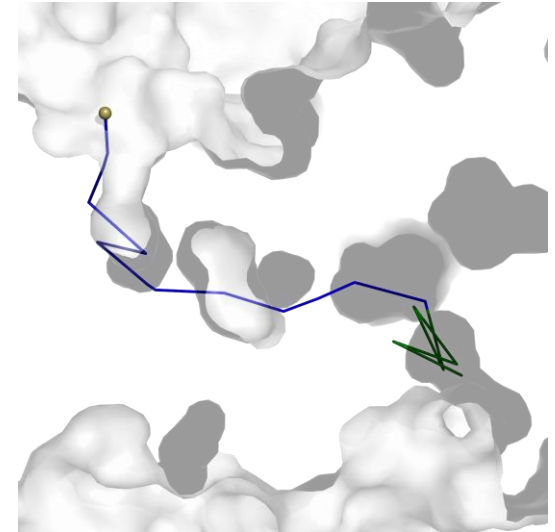


Fig. 5i. [smooth]
method = mss
recursive = 15
step = 5

window_mss

Window over Max Step smoothing is a combination of methods `window` and `mss` and performs window smoothing over maximal step smoothing. This method has an option `recursive` which is a number of recursive runs of smoothing method.

Increasing the value of `recursive` results in greater smoothness of path (**Fig. 6a-f**). Method `window_mss` allows to smooth the trajectory almost completely, still preserving the trajectory shape, although losing information about places where the molecule was trapped (**Fig. 6d-f**).

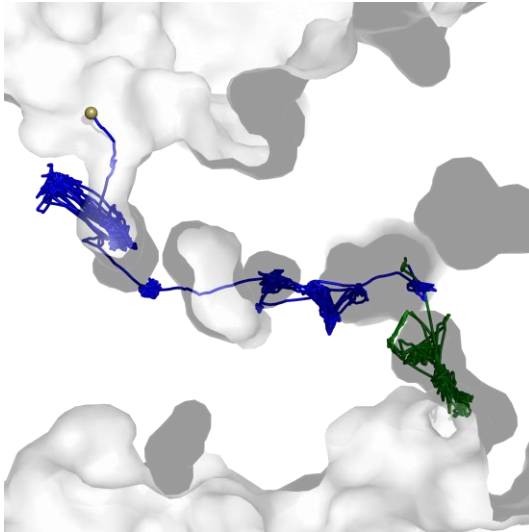


Fig. 6a. [smooth]
method = window_mss
recursive = 1

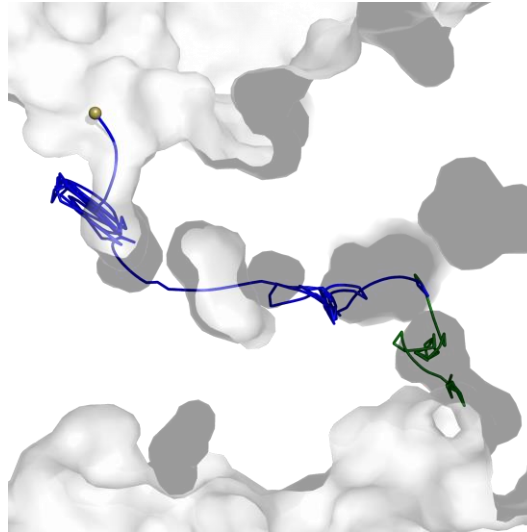


Fig. 6b. [smooth]
method = window_mss
recursive = 3

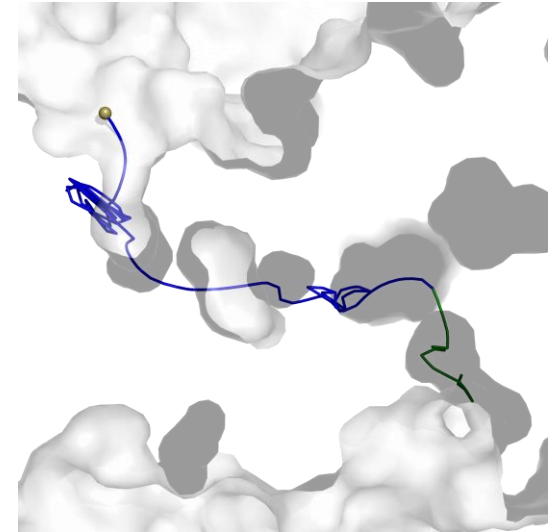


Fig. 6c. [smooth]
method = window_mss
recursive = 5

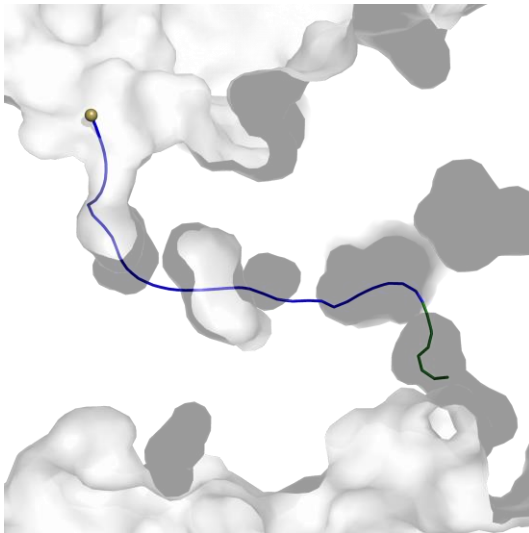


Fig. 6d. [smooth]
method = window_mss
recursive = 15

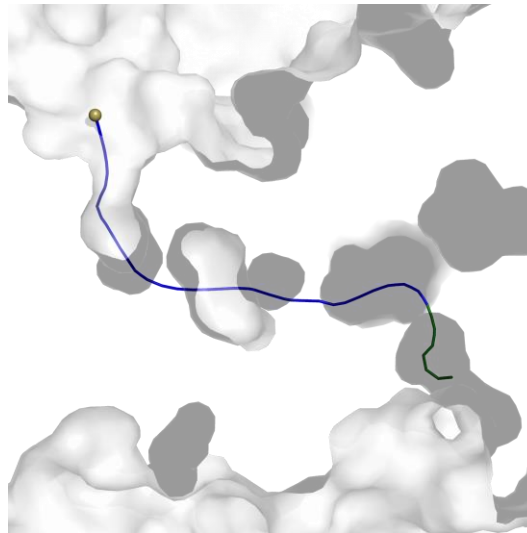


Fig. 6e. [smooth]
method = window_mss
recursive = 25

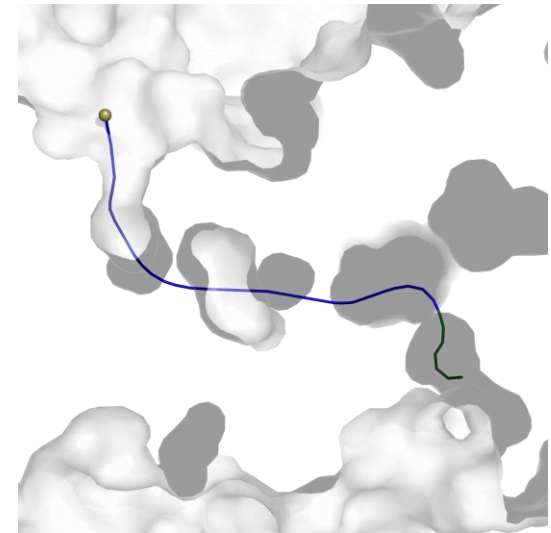


Fig. 6f. [smooth]
method = window_mss
recursive = 100

dwin_mss

Distance Window over Max Step smoothing is a combination of methods `dwin` and `mss` and performs distance window smoothing over maximal step smoothing. This method has an option `recursive` which is a number of recursive runs of smoothing method.

Increasing the value of `recursive` results in greater smoothness of path (**Fig. 7a-f**). This method, however, results in very quick smoothness of path, even with `recursive=3` the path is completely smoothed (**Fig. 7b**). This method, compared to `win_mss` method, can lose information about the actual trajectory of the molecule - that is, information about the voids that the molecule visits (**Fig. 6f, 7b**). Setting `recursive \geq 15` resulted in complete loss of information about the actual trajectory (**Fig. 7d-f**).

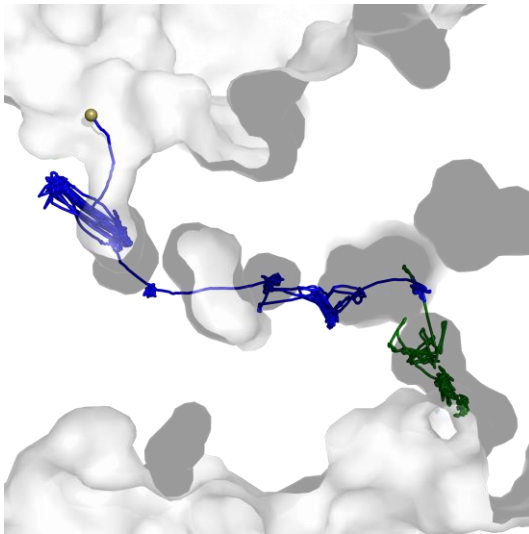


Fig. 7a. [smooth]
method = dwin_mss
recursive = 1

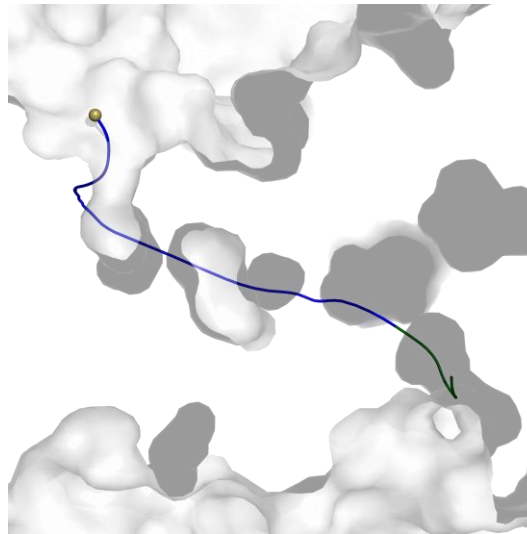


Fig. 7b. [smooth]
method = dwin_mss
recursive = 3

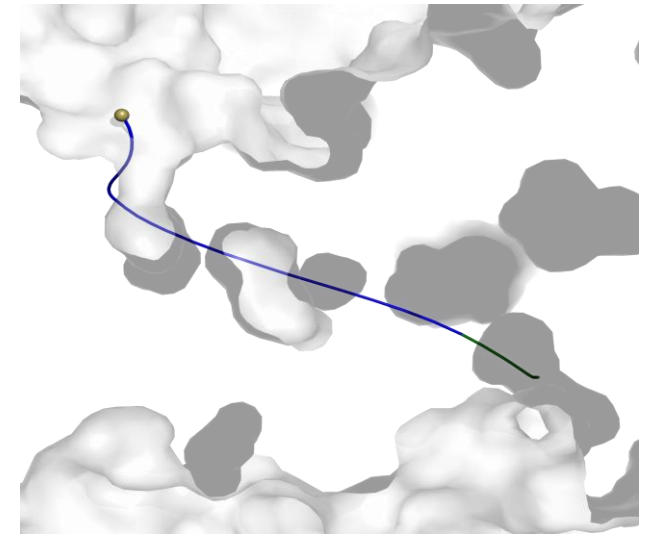


Fig. 7c. [smooth]
method = dwin_mss
recursive = 5

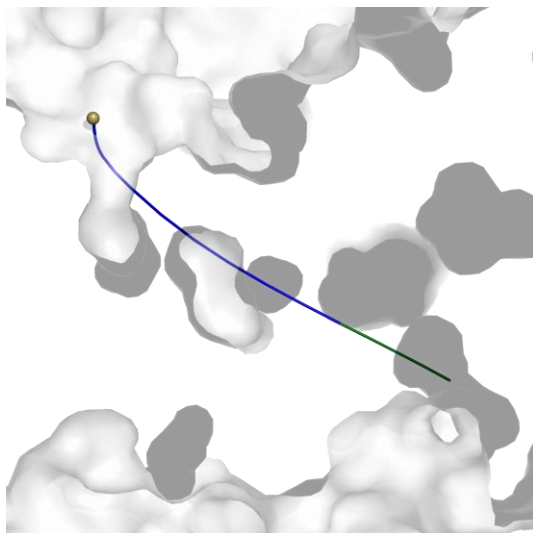


Fig. 7d. [smooth]
method = dwin_mss
recursive = 15

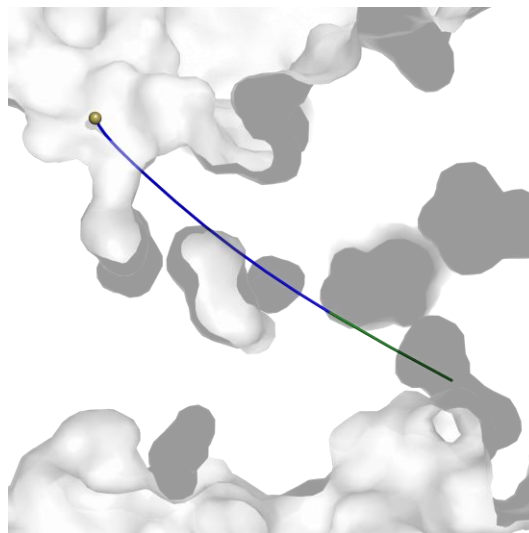


Fig. 7e. [smooth]
method = dwin_mss
recursive = 25

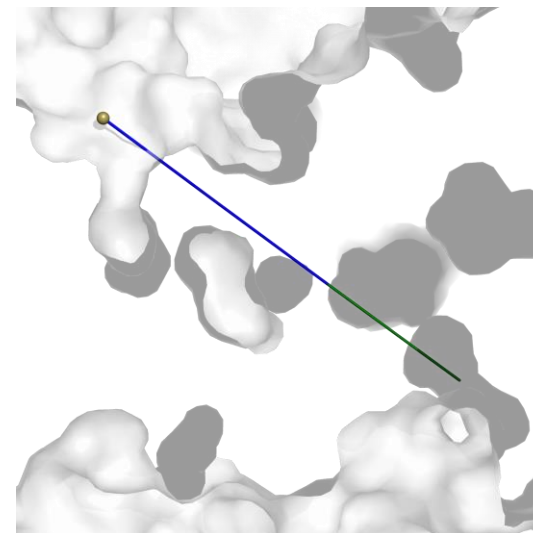


Fig. 7f. [smooth]
method = dwin_mss
recursive = 100

awin_mss

Active Window over Max Step smoothing is a combination of methods `awin` and `mss` and performs active window smoothing over maximal step smoothing. This method has an option `recursive` which is a number of recursive runs of smoothing method.

Increasing the value of `recursive` results in greater smoothness of path (**Fig. 8a-f**). This method, compared to `win_mss` method, can lose information about the actual trajectory of the molecule - that is, information about the voids that the molecule visits (**Fig. 6f, 8c**). Setting `recursive` ≥ 15 resulted in complete loss of information about the actual trajectory (**Fig. 8d-f**). Also, Setting `recursive` ≥ 25 resulted in losing information about character of the path - the path became mostly green (**Fig. 8e-f**).

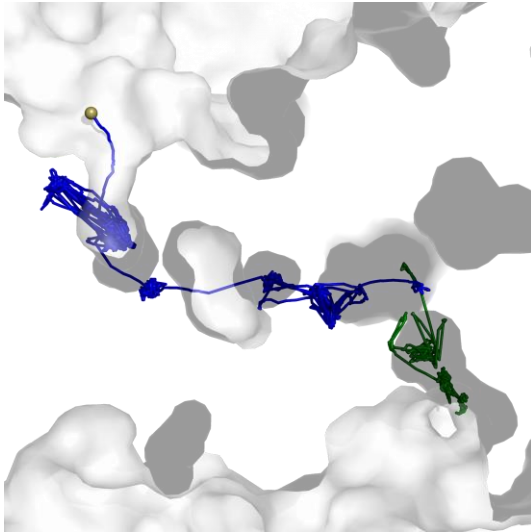


Fig. 8a. [smooth]
method = awin_mss
recursive = 1

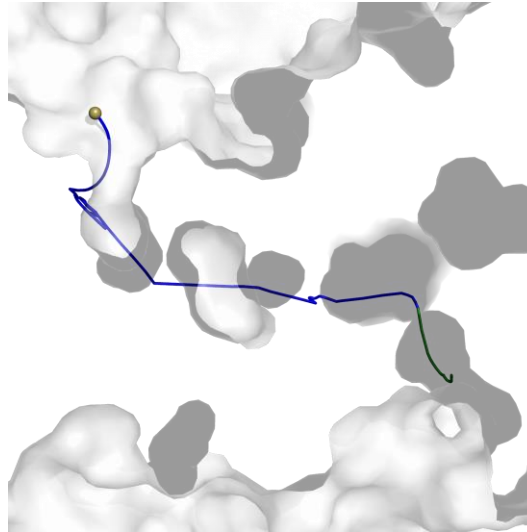


Fig. 8b. [smooth]
method = awin_mss
recursive = 3

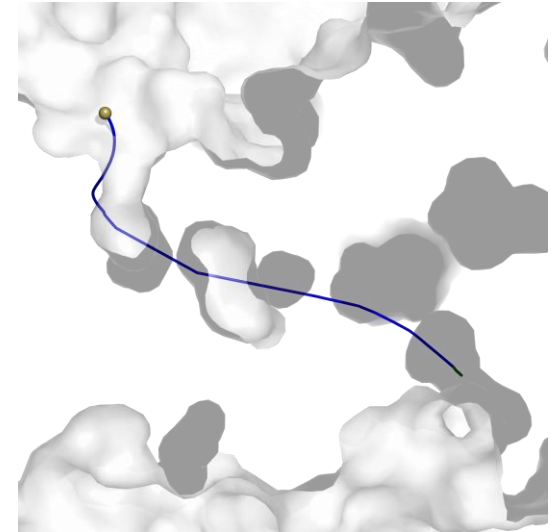


Fig. 8c. [smooth]
method = awin_mss
recursive = 5

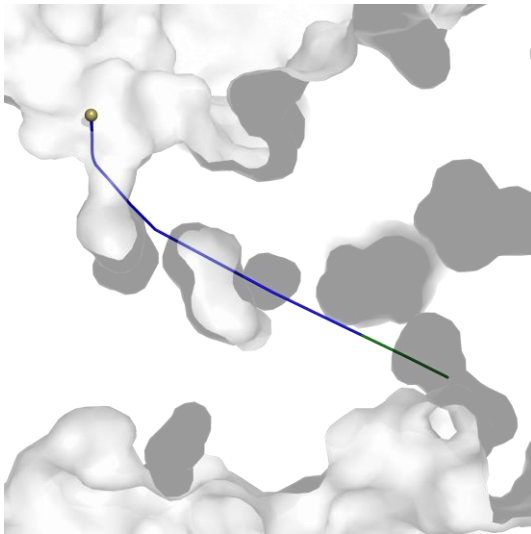


Fig. 8d. [smooth]
method = awin_mss
recursive = 15

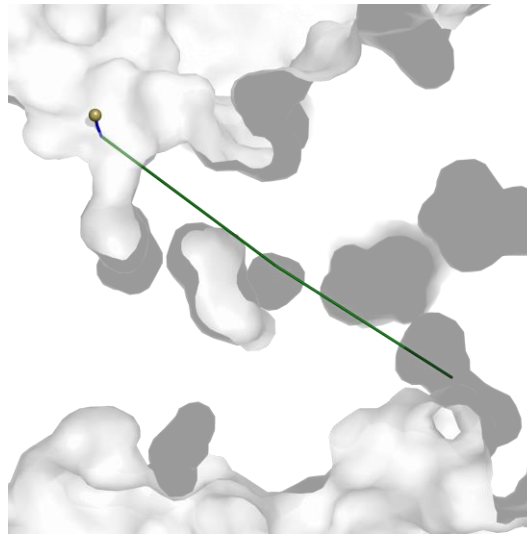


Fig. 8e. [smooth]
method = awin_mss
recursive = 25

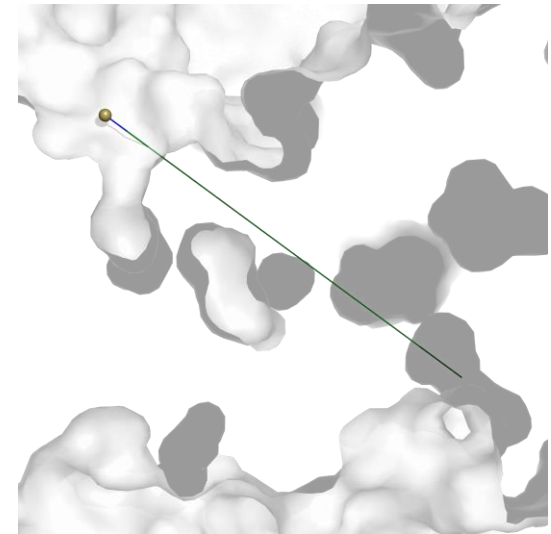


Fig. 8f. [smooth]
method = awin_mss
recursive = 100

savgol

Savitzky-Golay filter is a process known as convolution, which fits in a subset of trajectory points defined by `window` with a low-degree polynomial by the method of linear least squares. This method uses 1D filter that is applied for each dimension separately. The method has options `window` which is the window size and `polyorder` which is polynomial order. The `window` parameter has to be a positive odd number. Its default value is 5. The `polyorder` attribute has to be less than `window` and also cannot be too big. Its default value is 2.

The method `savgol` itself does not strongly smooth the path (**Fig. 9a**). It requires higher values for `window`, however, increasing this parameter too much may result in “detachment” of one of paths ends from the inlet (**Fig. 9d**). Increasing `polyorder` does not increase smoothness of paths (**Fig. 9c, 9e, 9f**). What is more, too great a value of `polyorder` may have a completely opposite effect to smoothing (**Fig. 9f**).

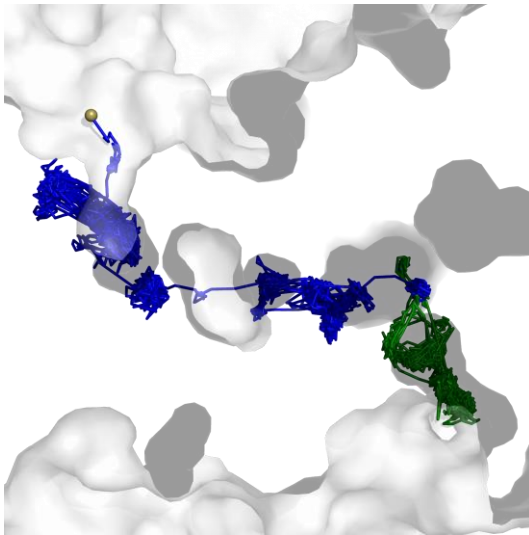


Fig. 9a. `[smooth]`
`method = savgol`
`window = 5`
`polyorder = 2`

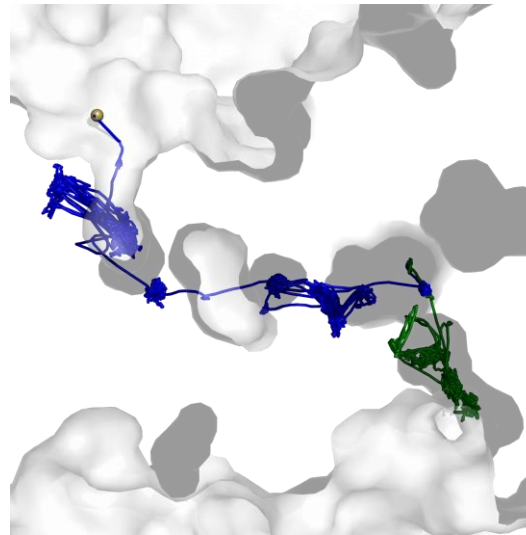


Fig. 9b. `[smooth]`
`method = savgol`
`window = 15`
`polyorder = 2`

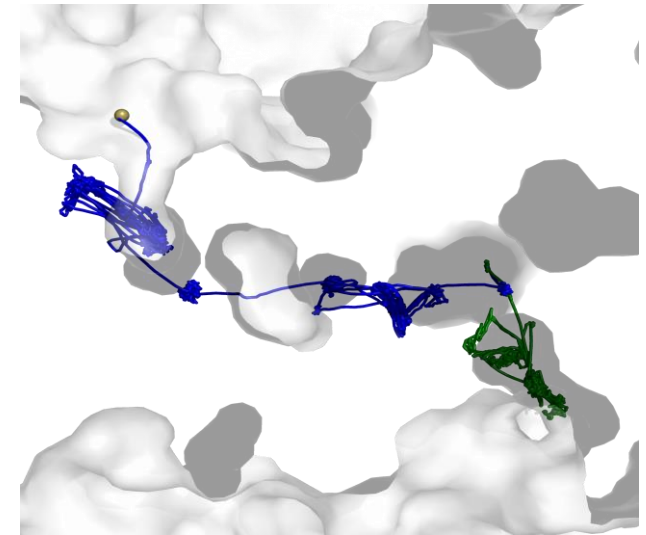


Fig. 9c. `[smooth]`
`method = savgol`
`window = 21`
`polyorder = 2`

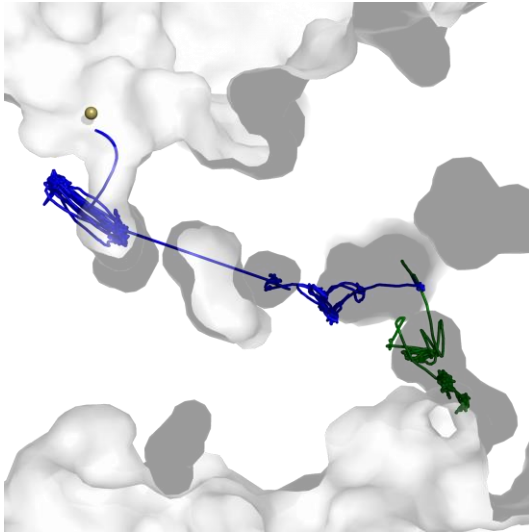


Fig. 9d. [smooth]
method = savgol
window = 101
polyorder = 2

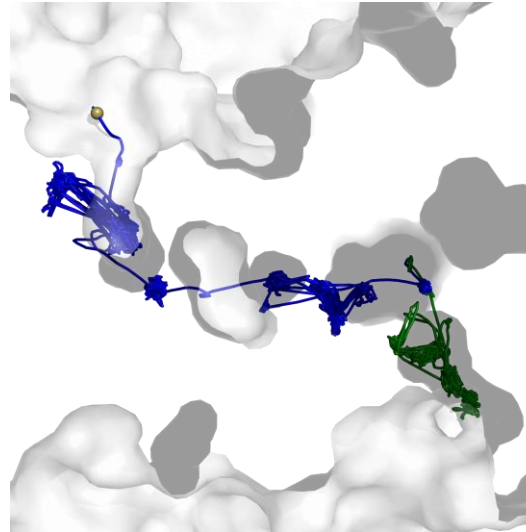


Fig. 9e. [smooth]
method = savgol
window = 21
polyorder = 5

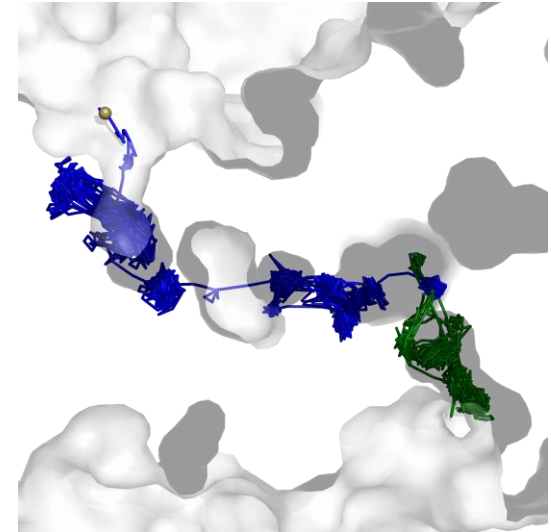


Fig. 9f. [smooth]
method = savgol
window = 21
polyorder = 15

Our tips

- All of the smoothing methods except for `savgol` have an option `recursive`, which is a number of recursive runs of the smoothing method. Generally, increasing this value results in greater smoothness of path. However, the same value of `recursive` may work differently with different methods.
- Different recursive values give different effects. For `window` it has a smaller effect, for `dwin` and `awin` it is quite strong. For example, for `dwin` and `awin` methods `recursive` results in a very smooth path even with one digit numbers, whereas for `window`, it has to be a three digit number. For combined methods with `mss`, `recursive` is even stronger.
- When choosing the smoothing method, you have to ask yourself what effect you want to achieve.
- Be careful not to lose too much of the data by too strong simplification of your paths!
- To obtain a smoother and more readable path, yet one that would still resemble the raw path and store all information about places where the molecule was trapped, you should choose window methods - `window`, `dwin`, `awin`, `savgol`.
- To obtain a path that would be simplified to points connected by straight sections, you should choose `mss`.
- To obtain a very strongly smoothed path without the information about places where the molecule was trapped, but one that would show the exact route of the molecule, you should choose a combined method - `win_mss`. The other two, `dwin_mss` and `awin_mss`, should also do the job, although they are much stronger and you have to be careful not to set too high `recursive` and not lose information about the actual molecule's route.

TIPS & TRICKS – smoothing paths

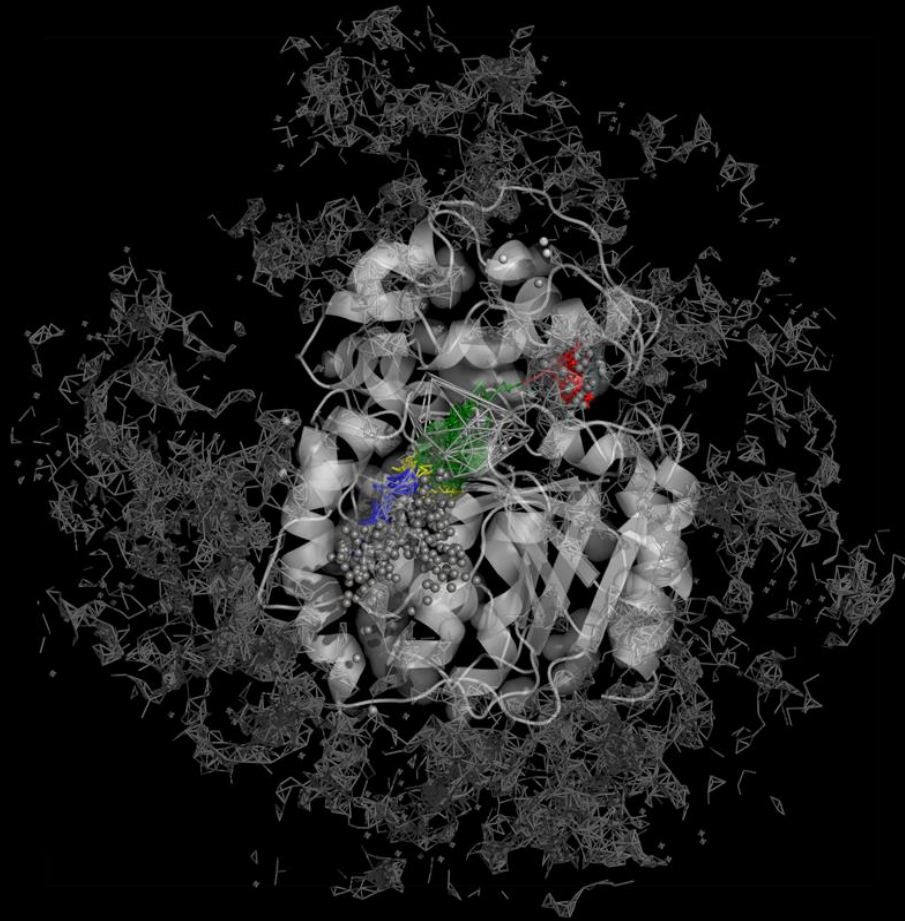


Installation and guide:

<http://www.aquaduct.pl>

More info:

info@aquaduct.pl



Tunneling Group
Biotechnology Centre
Silesian University of Technology
ul. Krzywoustego 8
44-100 Gliwice
POLAND