Supplementary data

AQUA-DUCT a ligands tracking tool

Tomasz Magdziarz[1], Karolina Mitusińska[1,2], Sandra Gołdowska[1,2], Alicja Płuciennik[1,2], Michał Stolarczyk[1,2], Magdalena Ługowska[1,2] and Artur Góra[1,*]

[1]Tunneling Group, Biotechnology Centre, Silesian University of Technology, ul. Krzywoustego 8, 44-100 Gliwice, Poland. [2]Institute of Automatic Control, Silesian University of Technology, Akademicka 16, 44-100 Gliwice, Poland

*To whom correspondence should be addressed.

Contents

1 Implementation

1.1 Installation

The AQUA-DUCT (AQ) can be installed on Linux, Windows, macOS and OpenBSD systems.

Details are available at http://www.aquaduct.pl/installation/.

We recommend 64-bit SMP architecture, with at least 4 GB RAM (32 GB RAM is recommended).

1.2 Input data

The compulsory user input is minimal; by default AQ requires only MD data and definition of the *scope* and *object* areas. AQ can natively read trajectory data in Amber binary NetCDF [1] and in binary DCD formats [2]. Topology of the system can be submitted as a PRMTOP Amber parameter-topology file [3], PDB file [4], or PSF topology file [2]. Other formats compatible with AQ are currently available, for example files created via the CATDCD software that is capable of converting many formats into DCD [5].

The example data set is available at http://www.aquaduct.pl/download/.

1.3 Limitations

In general all MD simulations results which can be converted into NetCDF or DCD formats can be used as an AQ input. Also there is no limitation concerning ligands which can be tracked, however their names have to be recognized as distinct names by MDAnalysis library (e.g., WAT or HOH for water molecules). Please note the center of gravity of ligand is tracked, therefore for asymmetric ligands the results need to be carefully examined.

2 AQ calculation workflow

Visualization of the flow of data in AQ is shown on Fig. S1. The flow is linear. User data (MD simulation data and configuration files) are read and used by successive stages of calculations. Each subsequent stage uses data produced by the previous stages. Finally, all data are used to calculate statistics at the analysis stage and create visualizations. AQ implements several algorithms: convex hull (section 3.2.1), Auto Barber (section 3.4), smoothing paths (section 3.4.1), averaging paths (section 3.5), clusterization (section 3.5.3).

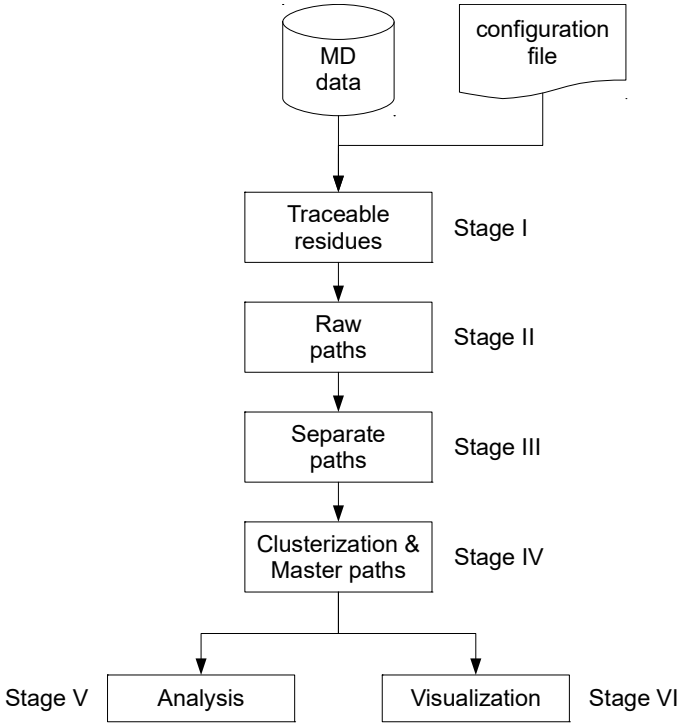

Fig. S1. The flow of data in AQ. User input, i.e. MD data and configuration file is read in subsequent stages of calculations.

2.1 Performance

The AQ performance was tested on murine epoxide hydrolase system (PDB ID: 1CQZ; 4992 atoms immersed in 8488 molecules of water). Tests were run under Linux system, Intel Core i7 CPU @ 3.50GHz machine, 64 GB RAM.

AQ calculation time depends on two main factors: length of the MD simulation (measured by number of frames) and size of the *object*. The latter factor controls number of traced molecules. The bigger *object* is the more molecules are traced.

There are two calculation stages that contribute the most to the total time: generation of raw paths (stage II), and clusterization of inlets (stage IV). However, in the case of stage IV the most time consuming part is optional calculation of average paths. Following picture compares AQ running time for 100 ns simulation with generation of average paths switched on and off.

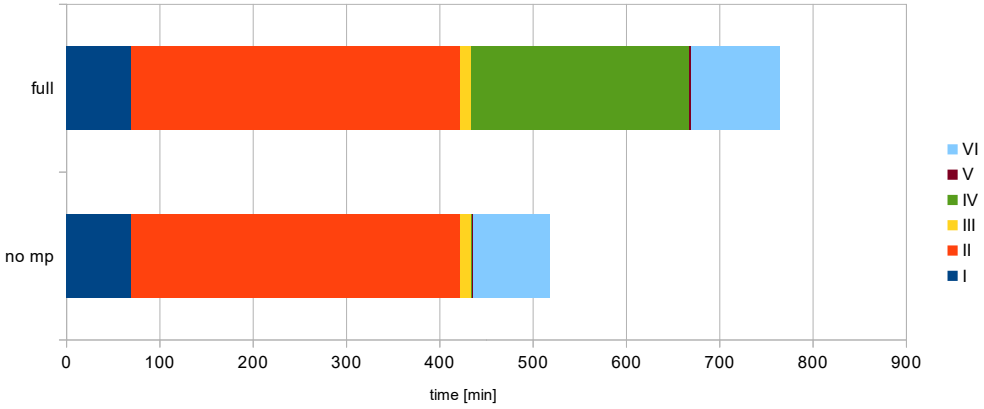

Fig. S2. Benchmark 1 Comparison of AQ running time with generation of average paths switched on and off (100 ns simulations of murine epoxide hydrolase system).

Following table shows AQ calculation time versus simulation length. The time of calculations can be reduced approximately by 30% when generation of average paths is switched off.

Table S1. Benchmark 2. AQ calculation time versus simulation length of mouse epoxide hydrolase system.

Simulation length [ns]	Number of traced molecules	AQ calculation time generation of average pathways off	AQ calculation time generation of average pathways on
1	45	1m 24s	2m 26s
2	73	3m 1s	5m 4s
4	123	6m 45s	11m 18s
8	189	15m 43s	25m 49s
10	215	20m 6s	33m 4s
15	293	32m 39s	51m 56s
16	308	36m 41s	57m 11s
20	395	50m 7s	1h 15m 28s
32	599	1h 37m 52s	2h 17m 53s
50	862	2h 47m 8s	4h 15m 45s
64	1058	4h 3m 13s	6h 2m 6s
100	1588	8h 37m 48s	12h 43m 49s
128	1721	11h 41m 19s	17h 20m 43s

Dependency between calculation time and number of traced molecules is close to linear (Fig. S3). Stage 1 and partially III and V do not depend strongly on the number of traced molecules.

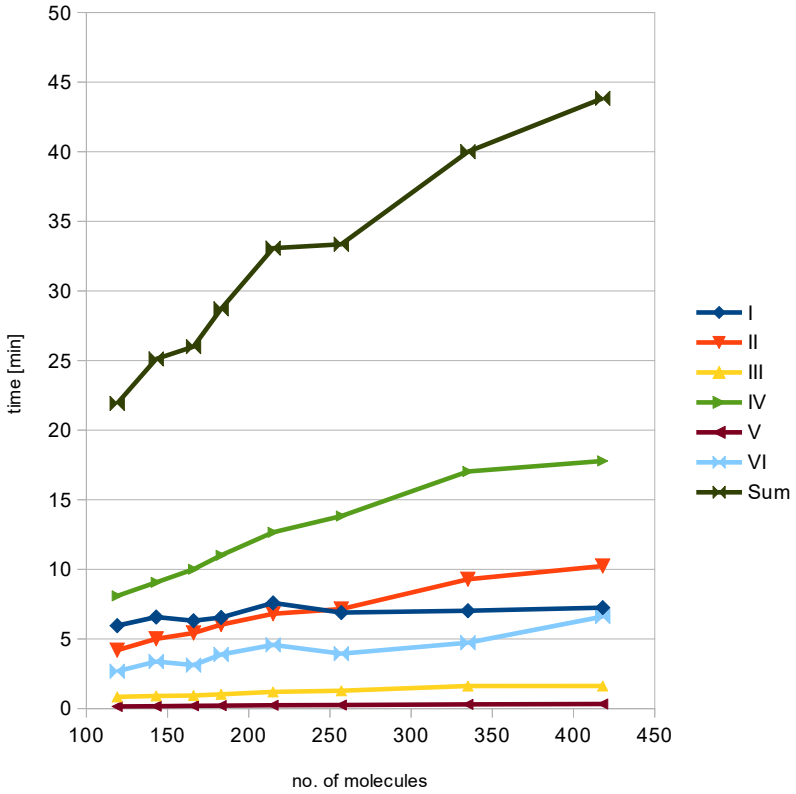

Fig. S3. Benchmark 3 Comparison of AQ running time with different number of traced molecules (10 ns simulations of murine epoxide hydrolase system).

3 AQ configuration file - explanation

Calculations in AQ are performed according to options in the configuration file, which is a plain text file with its content divided into sections. The **global** section allows the user to setup paths of data files. The rest of the options, including the *scope* and *object* definitions, are in sections that correspond to their respective stages of AQ calculations. Smoothing and clusterization detailed options are in separate sections.

The following is the content of a configuration file used to perform calculations described in chapter 4 in user case.

3.1 Global settings

```
[global]
top = 1cqz.prmtop
trj = 1cqz.nc
```

Options **top** and **trj** hold paths to MD data files. AQ accepts PSF (CHARMM/NAMD/XPLOR files), PRMTOP (Parm7 Amber), and PDB files as topologies. Trajectory data can be NC (Amber NetCDF) or DCD (CHARMM or NAMD, or LAMMPS).

3.2 Traceable residues

The first stage of AQ calculations finds all molecules that should be traced and creates the list of traceable molecules. The search of the molecules is done according to user provided definitions of the *scope* and *object* areas. These definitions have to be placed in the **traceable_residues** section. The *scope* is the area to which AQ limits its analysis. The *object*, on the other hand, is an area of special interest, for example the active site, which has to be penetrated by traceable molecules in at least one frame of the simulation.

```
[traceable_residues]
scope = name CA
scope_convexhull = True
```

The *scope* defined as interior of a convex hull of alpha carbon atoms.

The *scope* can be defined as interior of a convex hull of atoms of any molecular object. It can be the protein backbone (**backbone**), entire protein (**protein**), or any other atoms. The *scope* can also be defined directly. In that case, **scope_convexhull** should be set to **False** and the definition should include the names of molecules to be traced and spatial constraints, for example: **resname WAT around 2.0 protein**.
Please note, residue name of the water molecules is dependent on the topology file. Here it is **WAT** but it can also be **HOH** or another name.

```
object = (resname WAT) and (sphzone 6.0 (resnum 99 or resnum 147 or resnum 231
or resnum 261 or resnum 289))
```

The *object* was defined as water molecules that are within a 6 Å spherical zone around center of masses of residues that build the active site.

The *object* definition has to comprise of the name of the molecules to trace and some kind of spatial constraints. The *object* area should be inside of the *scope*.

3.2.1 Convex hull of macromolecule atoms

AQ uses quickhull algorithm for convex hulls calculations [6]. Convex hull concept is used to check if traced molecules are inside of the macromolecule. Convex hull can be considered as a very rough approximation of molecular surface. Its interior does not depend much on conformational changes, whereas, for example the use of the solvent-excluded molecular surface (SES) for molecular surface representation could result in unwanted removal of tracked molecules due to large caves opening upon relatively small conformational changes (Fig S4).

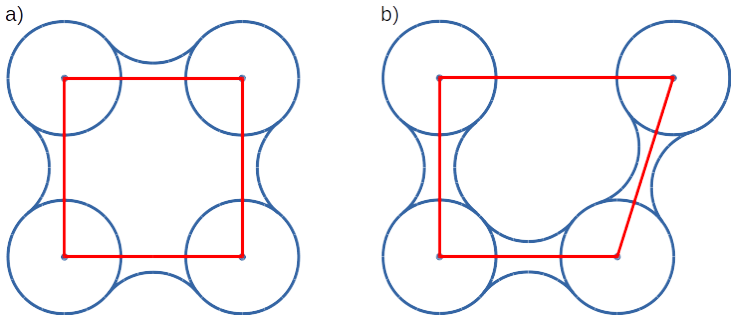

Fig S4. Comparison of the approximation of molecular surface by convex hull (red lines) and solvent-excluded molecular surface methods (blue lines) during local conformational changes, a) closed conformation, b) opened conformation.

3.3 Raw paths

The second stage of calculations uses the list of all traceable molecules from the first stage and finds coordinates of the center of masses for each frame as long as molecules are within the *scope*. Definitions of the *scope* and *object* are used in the first and second stages of AQ calculations.

```
[raw_paths]
scope = name CA
scope_convexhull = True
object = (resname WAT) and (sphzone 6.0 (resnum 99 or resnum 147 or resnum 231
or resnum 261 or resnum 289))
```

The *scope* and *object* definitions are the same as in the **`traceable_residues`** section.

```
clear_in_object_info = False
```

AQ allows the user to redefine the *scope* and *object* areas in the **`raw_path`** section. By default, the *object* occupancy data calculated in the first stage are reused in the second stage. However, upon redefinition of the *object* it is recommended to also switch on the option **`clear_in_object_info`**. This will force recalculation of the *object* occupancy data.

3.4 Separate paths

The third stage of AQ calculations creates separate paths using the raw data generated in the second stage. Options related to this stage should be placed in the **`separate_paths`** section of the configuration file.

```
[separate_paths]
sort_by_id = True
```

By default, AQ sorts separate paths by the ID of the corresponding residue. Paths can also be sorted by order of appearance by switching off the **sort_by_id** option.

```
discard_short_paths = 1
```

Paths comprising of 1 frame or shorter are removed.

Short paths usually are not of great importance and can be removed. This option accepts any positive natural number.

```
apply_smoothing = False
apply_soft_smoothing = True
```

Permanent (hard) smoothing is switched off, soft smoothing is switched on. See also **smooth** section.

Paths identified on the basis of raw data can be used either as they are or can be smoothed. There are two modes of applying smoothing: permanent or soft. Permanent smoothing replaces the original paths' traces to smoothed. All further calculations will use smoothed traces.
On the other hand, soft smoothing generates smooth traces and keeps them along with the originals. In this case, smoothed traces can be used for visualization only or can also be used to generate averaged paths in the fourth stage.

```
auto_barber = protein
auto_barber_mincut = 4.5
auto_barber_maxcut = None
auto_barber_tovdw = False
```

Auto Barber procedure is set to **protein**. Minimal sphere used to cut is 4.5 Å, maximal size is not set. Correction to van der Waals radius of closest atom is switched off.

Both methods of the *scope* definitions, direct and convex hull way, are not capable to detect the macromolecule surface correctly. It is possible, however, to set the **auto_barber** option, which trims paths down to the approximated surface of the macromolecule or other molecular entity defined by this option. This trimming is done by creating collection of spheres that have centers at the ends of paths and radii equal to the distance from the center to the nearest atom of user defined molecular entity. Next, parts of raw paths that are inside these spheres are removed and separate paths are recreated. Option **auto_barber_mincut** allows to define minimal radius of spheres used in trimming. Option **auto_barber_maxcut** allows to define maximal radius of spheres used in trimming. Option **auto_barber_tovdw** allows to correct spheres' radii by van der Waals radius of the nearest atom.

3.4.1 Smoothing method

Smoothing methods used in AQ have to satisfy one criterion, namely the number of points in smoothed trajectory have to be preserved. Default method, is well known Savitzky-Golay filter

(**savgol**) [7]. In addition 3 different moving window methods are available: Window, ActiveWindow, and DistanceWindow; (**window, awin, dwin** respectively). They differ by window size definition. Window method uses number of frames, ActiveWindow and DistanceWindow use window length in Å which is transformed to number of frames in two different ways. Additional available method is MaxStep (**mss**) which finds "cardinal points" in the trajectory that are separated by the distance defined with step parameter. Next, all cardinal points and points of linear interpolation between cardinal points are returned as smoothed coordinates. Number of interpolated points is in accordance to points skipped between cardinal points.

Window methods and MaxStep method can be combined which results in 3 additional methods in which MaxStep smoothing is run before Window, ActiveWindow, and DistanceWindow smoothing respectively (**window_mss, awin_mss, dwin_mss**).

Details of the smoothing method can be defined in the **smooth** section.

```
[smooth]
method = dwin_mss
step = 4.2
window = 4.2
```

Smoothing with **dwin_mss** method with options **step** set to 4.2 Å and **window** set to 4.2 Å.

Very strong smoothing can results in trajectory which looks very well but it may lack of some features which are crucial for correct interpretation.

3.5 Clusterization of inlets

Each of the separate paths has a beginning and an end and if they are at the boundaries of the *scope* they are considered as inlets, i.e., points that mark where the traceable molecules enter and/or leave the *scope*. Clusters of inlets mark endings of tunnels or ways in the system that was simulated in the MD.

General clusterization options are grouped in the **inlets_clusterization** section. Detailed settings of the clusterization method are located in a separate **clusterization** section. Methods that are used in optional reclusterization of outliers defined as a non-clustered inlets are defined in a separate **reclusterization** section.

```
[inlets_clusterization]
detect_outliers = Auto
```

Automatic detection of outliers is switched on.

If the option **detect_outliers** is switched on, the procedure of detecting outliers is performed. This step can be executed in two modes:

1) Automatic mode: Option **detect_outliers** is set to **Auto**. An automatic procedure is executed and points which are far from the centroid of the cluster are annotated as outliers. This is based on the experimental method and is subject of change. More information can be found on the AQ home page.

2) Threshold mode: Option **detect_outliers** is set to the value of the threshold in Å. In this mode, a point is considered to be an outlier if its minimal distance from any other point in the cluster is greater than the threshold.

recluster_outliers = True

Reclusterization of outliers is switched on. The method defined in the reclusterization section will be used in this step.

singletons_outliers = 2

Annotation of small clusters as outliers is set to 2.

Option **singletons_outliers** can be set to remove small clusters. If the cluster size is less or equal to this setting, it is removed and its points are annotated as outliers.

max_level = 2

Maximal level of recursive clusterization is set to 2.

Both initial clusterization and reclusterization can be performed in a recursive manner. Clusters generated in one step can be submitted for further clusterization. By default, AQ allows for 5 levels of such recurrence.

create_master_path = True

Generation of averaged paths that correspond to the phenomena of transition from one cluster to another is switched on. More information can be found in documentation.

3.5.1 Main clusterization method

Details of the initial clusterization method are located in the separate section **clusterization**.

```
[clusterization]
method = meanshift
```

MeanShift method is selected [8].

```
cluster_all = True
bandwidth = Auto
```

Options **cluster_all** and **bandwidth** are specific for the MeanShift method. More details can be found in documentation.

recursive_clusterization = clusterization

Clusterization method defined in the **clusterization** section will be reused in the recursive clusterization.

To setup recursive clusterization, appropriate sections should have the **recursive_clusterization** option set to a section name that holds the clusterization

method definition. This definition is used at the next level of clusterization. In the case of initial clusterization, the **recursive_clusterization** option is set by default to **clusterization**. This means that by default, the method defined in the **clusterization** section is reused in all levels of initial clusterization.

```
recursive_threshold = >0.9
```

Only clusters greater than 0.9 of the total possible size will be submitted for recursive clusterization.

The user can also exclude clusters of a certain size from recursive clusterization. This is possible by setting the **recursive_threshold** option to a desired threshold. Definition of the threshold is comprised of an operator and a value. Operator can be any one of the following: >, >=, <=, <. Value has to be expressed as a floating number and must be in the range of 0 to 1, where 0 denotes an empty cluster and 1 denotes a cluster of all possible points.

3.5.2 Reclusterization method

The following section defines the clusterization method used in the reclusterization of outliers.

```
[reclusterization]
method = meanshift
cluster_all = False
bandwidth = Auto
```

MeanShift method is selected and some options specific for this method are used.

3.5.3 Other clusterization methods

AQ implements clusterization methods using Scikit-learn [9]. Currently, there are 5 methods available: MeanShift [8], DBSCAN [10], Affinity Propagation [11], Birch [12], and K-Means [13]. The proper choice of the clusterization method strongly depends from geometry of analyzed system, spatial distribution of inlets, number of inlets, etc. The Mean Shift method is recommended as a starting method since it is dedicated for analysis of many clusters with uneven cluster size and it is applicable for non-flat geometry. Alternatively, DBSCAN method can also be applicable. Other methods are recommended for reclusterization rather since they possess several limitations (e.g., they are recommended for flat geometry, they perform well for limited number of clusters etc.).

3.6 Analysis and statistics calculations

Once the fourth stage of AQ calculations is completed, a statistical summary of traced molecules is generated. This summary is saved as a plain text file.

```
[analysis]
dump_config = True
```

Configuration will be included in the header of the statistical summary file.

3.7 Visualization options

The sixth stage of AQ calculations visualizes the results calculated in stages 1 to 4.

By default AQ does not prepare any visualizations. To switch on this feature, appropriate options should be added to the **visualize** section.

```
[visualize]
all_paths_raw = True
```

All of the separate paths will be displayed using raw (original) coordinates.

```
all_paths_smooth = True
```

All of the separate paths will be displayed using smoothed coordinates.

```
all_paths_split = True
```

All of the separate paths will be split into incoming, object, and outgoing parts.

```
all_paths_raw_io = True
```

Points where the incoming parts of traces begin and where the outgoing parts end are indicated by small arrows (cones). Directions are calculated for raw paths.

```
all_paths_smooth_io = True
```

Points where the incoming parts of traces begin and where the outgoing parts end are indicated by small arrows (cones). Directions are calculated for smooth paths.

```
paths_raw = True
```

Separate visualizations of raw paths.

```
paths_smooth = True
```

Separate visualizations of smooths paths.

```
paths_states = True
```

Separate visualizations of raw and smooth paths created as one object with multiple states.

This option allows to investigate each detected trajectory separately.

```
paths_raw_io = True
```

Points where the incoming parts of traces begin and where the outgoing parts end are indicated by small arrows (cones). Directions are calculated for raw paths of separated visualizations.

```
paths_smooth_io = True
```

Points where the incoming parts traces begin and where the outgoing parts end are indicated by small arrows (cones). Directions are calculated for smooth paths of separated visualizations.

```
ctypes_raw = True
```

Separate paths will be displayed using raw (original) coordinates and for each cluster-to-cluster transition a separate object is created.

```
ctypes_smooth = True
```

Separate paths will be displayed using smooth coordinates and for each cluster-to-cluster transition a separate object is created.

```
inlets_clusters = True
```

Clusters are displayed as scatter plots of points.

```
show_molecule = protein
show_molecule_frames = 0
```

Protein structure is displayed for frame 0.

```
show_chull = name CA
show_chull_frames = 0
```

Convex hull of alpha carbon atoms is displayed for frame 0.

```
show_object = name * and (sphzone 6.0 (resnum 99 or resnum 147 or resnum 231 or
resnum 261 or resnum 289))
show_object_frames = 0
```

Convex hull of all atoms within the spherical zone of 6 Å from the center of mass of residues 99, 147, 231, 261, and 289 is displayed for frame 0.

This options works exactly the same as **`show_chull`** but is meant for showing the shape of the *object*.

```
save = visualization_script.py
```

Visualization objects will be saved as a Python script and archive file.

By default, AQ creates visualization objects as archive files and creates Python scripts that read the archive, starts PyMOL, and loads all objects. It can optionally save PyMOL session or the session can be saved later.
It is also possible to skip the archive creation and visualization script and generate PyMOL session directly in AQ. To do so, option **`save`** must be used and be set to the session file name (with **`.pse`** extension). The **`save`** option can also be used to alter the default name of the visualization script. In this case, it has to end with the **`.py`** extension.

4 USER CASE

As a user case, an example of the analysis of 10 ns of murine soluble epoxide hydrolase is presented. AQ calculations took 33 minutes to complete on Intel Core i7 @ 3.5 GHz machine.
The example data set (for 1 ns MD simulation) is available at http://www.aquaduct.pl/download/. The 10 ns data set is available on request.

4.1 MD simulation

Results of a short classical MD simulation (10 ns) of the murine soluble epoxide hydrolase structure (PDB ID: 1CQZ) were used as the input data for AQ analysis. Prior to MD calculations run by AMBER 14 [14], counterions were added and the protein was immersed in a truncated octahedral box of TIP3P water molecules. Water molecules were also placed in cavities inside the protein using a combined method of the 3D-RISM [14] and Placevent algorithm [15]. In total, 8488 molecules of water were added to the system. The coordinates were saved at intervals of 1 ps, and the MD trajectory was saved as an Amber NetCDF binary file.

4.2 AQ complete configuration file

```
[global]
top = 1cqz.prmtop
trj = 1cqz.nc

[traceable_residues]
scope = name CA
scope_convexhull = True
object = (resname WAT) and (sphzone 6.0 (resnum 99 or resnum 147 or resnum 231
or resnum 261 or resnum 289))

[raw_paths]
scope = name CA
scope_convexhull = True
object = (resname WAT) and (sphzone 6.0 (resnum 99 or resnum 147 or resnum 231
or resnum 261 or resnum 289))
clear_in_object_info = False

[separate_paths]
sort_by_id = True
discard_short_paths = 1
apply_smoothing = False
apply_soft_smoothing = True
auto_barber = protein
auto_barber_mincut = 4.5
auto_barber_maxcut = None
auto_barber_tovdw = False

[smooth]
method = dwin_mss
step = 4.2
window = 4.2

[inlets_clusterization]
detect_outliers = Auto
recluster_outliers = True
singletons_outliers = 2
max_level = 2
create_master_paths = True
```

```
[clusterization]
method = meanshift
cluster_all = True
bandwidth = Auto
recursive_clusterization = clusterization
recursive_threshold = >0.9

[reclusterization]
method = meanshift
cluster_all = False
bandwidth = Auto

[analysis]
dump_config = True

[visualize]
all_paths_raw = True
all_paths_smooth = True
all_paths_split = True
all_paths_raw_io = True
all_paths_smooth_io = True
paths_raw = True
paths_smooth = True
paths_states = True
paths_raw_io = True
paths_smooth_io = True
ctypes_raw = True
ctypes_smooth = True
inlets_clusters = True
show_molecule = protein
show_molecule_frames = 0
show_chull = name CA
show_chull_frames = 0
show_object = name * and (sphzone 6.0 (resnum 99 or resnum 147 or resnum 231 or
resnum 261 or resnum 289))
show_object_frames = 0save = visualization_script.py
```

4.3 Visual inspection

Prior to analysis, it is recommended to check the visualization to confirm clusterization correctness. View of AQ results limited to smoothed paths is shown on Fig. S5. The entry/exits of traceable molecules should be clustered in space, and single separated clusters should be grouped as outliers.

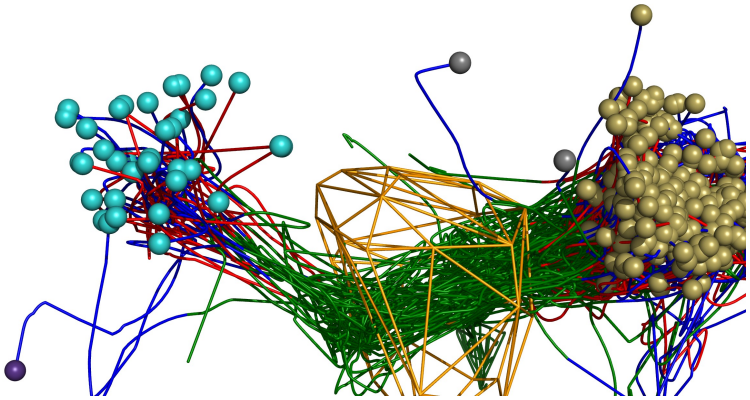

Fig. S5. Plain visualization of smoothed paths. Incoming parts shown in red, object parts in green, and outgoing parts in blue. Clusters of inlets points representing tunnels exits shown as scattered color balls. Main clusters 1 and 2 shown in sand and cyan, respectively. They show both, the entry and egress of water molecules through tunnel T1 and T2,

respectively. Small cluster 3 (identifying tunnel T3) in purple and outliers in grey show water molecules egress only. The *object* is marked as orange shape.

AQ implements a method for the optional trimming of paths down to the protein surface, a so-called Auto Barber procedure. Consequently, the inlets points of paths are located very close to the protein surface, which allows for more precise clusterization. This is especially useful in cases when the tunnel entrance is located within the pocket. Without the Auto Barber procedure in such cases, paths inlets are widely scattered and difficult to cluster. Fig. S6 shows an example of nicely trimmed paths where the tunnel T1 entrance is located inside the pocket. It is recommended to always use the Auto Barber procedure, however, there are situations when it is not necessary. See example on Fig. S7 – tunnel T2.

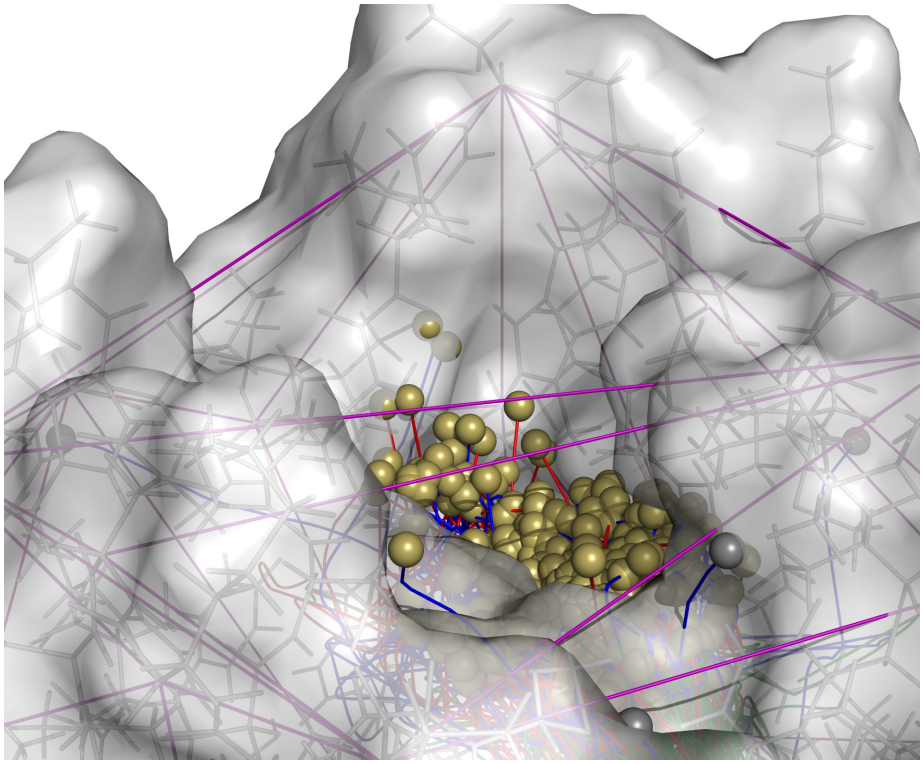

Fig. S6. Close up view of the tunnel T1 entrance located inside a pocket. Paths were trimmed by the Auto Barber procedure. Cluster of paths inlets is shown as a scatter of sand balls.

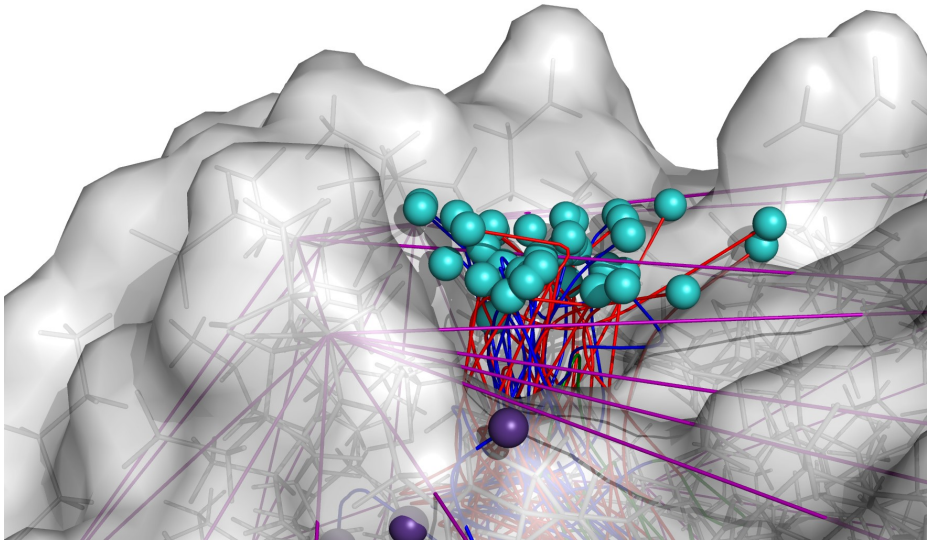

Fig. S7. Close up of tunnel T2 entrance. Cluster of paths inlets is shown as a scatter of cyan balls.

4.4 Statistical analysis

Statistical data are gathered in 5_analysis_results.txt file (Fig. S8).

```
method = dwin_mss
================================================================================
Number of traceable residues: 151
Number of separate paths: 206
================================================================================
Number of inlets: 380
Number of clusters: 3
Outliers: yes
================================================================================
Clusters summary - inlets
----------------------------------------
     Nr Cluster    Size INCOMING OUTGOING
----------------------------------------
      0       0       4        0        4
      1       1     337      172      165
      2       2      35       19       16
      3       3       4        0        4
================================================================================
Separate paths clusters types summary - mean lengths of paths
--------------------------------------------------------------------------------
     Nr   CType    Size      Inp   InpStd      Obj   ObjStd      O
--------------------------------------------------------------------------------
      0     1:1     149    134.9   189.69    328.1   582.76    156
      1     1:2       5    179.1   168.15    646.5   266.76    423
      2     1:3       2    132.5    58.89    414.0   213.78    200
      3     1:N      12     88.3    69.50   2138.7  2151.66      n
      4     1:0       4     79.2    51.12    147.1   162.56    832
```

Fig. S8. Screenshot of the block of the analysis_results.txt file with main statistical data.

On the basis of the tables produced by AQ, summary Table S3 was created.

Table S3. Summary of incoming and outgoing paths group by clusters. Rows correspond to incoming clusters, columns correspond to outgoing cluster. Column SUM shows sum of incoming paths. Values in the table correspond to amount of ligands that enter by exit indicated by row cluster and leave by exit indicated by column cluster. N means the core of the protein. Percent values in parentheses are the ratios calculated in regard to values in SUM column and show the percentage contribution of given exit.

	1		2		3		N		outliers		SUM
1	149	(87%)	5	(3%)	2	(1%)	12	(7%)	4	(2%)	172
2	6	(32%)	9	(47%)	2	(11%)	2	(11%)	---		19
3	---		---		---		---		---		---
N	10	(67%)	2	(13%)	---		3	(20%)	---		15
outliers	---		---		---		---		---		---

Several conclusions can be drawn based on the results from the table:

- Tunnel cluster 1 is the most frequently used tunnel exit (172/206 = 83% of all traffic),
- Most of the water molecules that enter the active site by the T1 tunnel exit leave the active site by the same exit (diagonal 1.1 87%), others leave by another exits (T2 or T3 or outliers 6%) or stay in the active site (7%).
- About 10% of water molecules that were identified in the active site pass through the T2 tunnel,
- T3 tunnel exit was used only by outgoing paths,
- 3 water molecules stayed in the active site during whole analysis

4.5 Analysis of visualization

Visualization opened in the PyMOL software can provide additional details about water traffic in the analyzed system. The sample of capabilities is presented on the picture below.

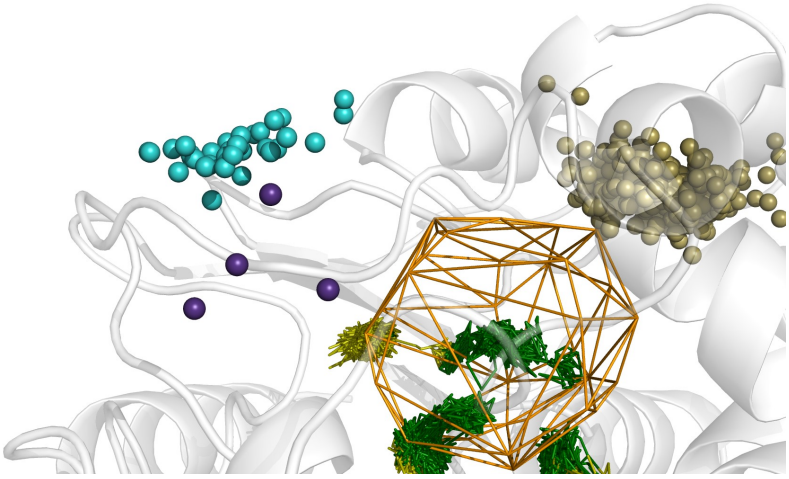

Fig. S9. Visualization of raw paths of water molecules that do not leave the *scope* for the entire simulation. 4 small pockets are visible. Raw paths visualization allows one to distinguish object parts into two categories. Green color marks positions in which molecules were strictly inside the *object* area. Yellow color marks positions in which molecules left the *object* but stayed in the *scope*.

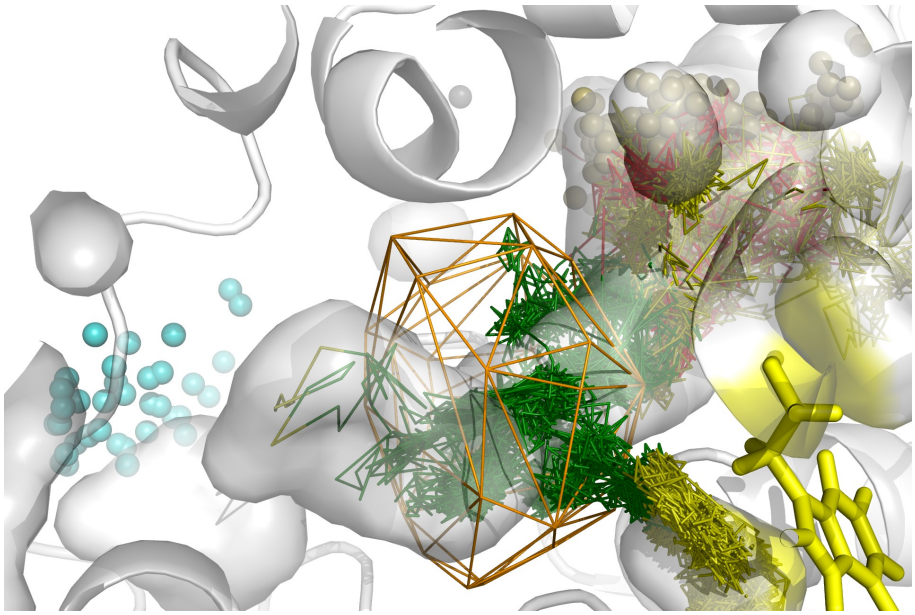

Fig. S10. Close up visualization of W230, shown in yellow sticks, that hinders and blocks the movement of water molecules in one of the small pockets located close to the *object* area.

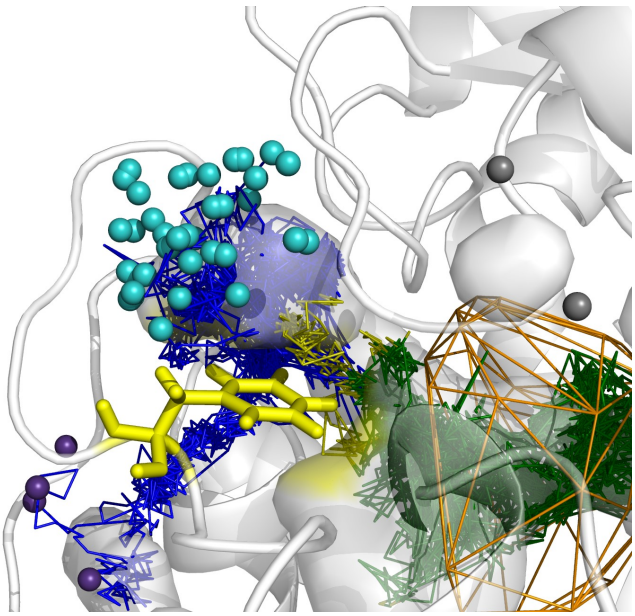

Fig. S11. Trajectories that leave the protein by T2 and T3 tunnels exits are separated by F181 shown in yellow sticks. F181 residue is probably also responsible for hindering movements of water in this direction.

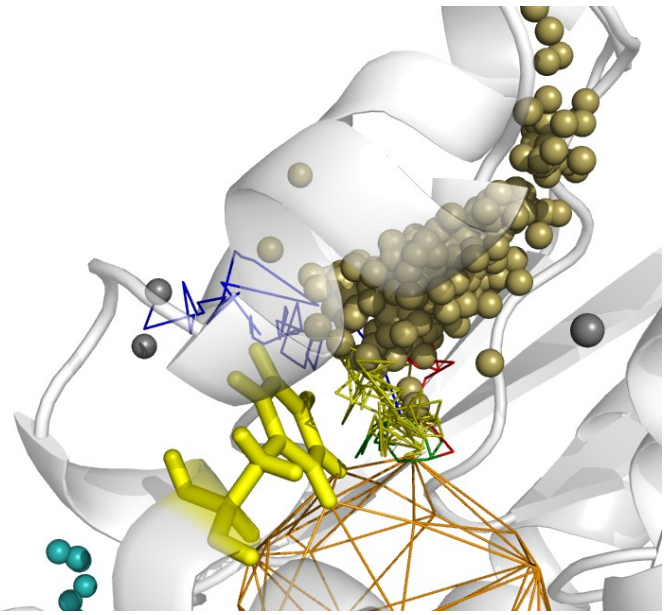

Fig. S12. Analysis of flux between tunnel T1 exit and outliers revealed that outliers are in fact branches of the main trace. F145 (sticks) seems to block water movements in the flux between tunnel T1 and outliers (path no 64 from raw paths).

5 REFERENCES

[1] AMBER NetCDF Trajectory/Restart ConventionVersion 1.0, Revision C, http://ambermd.org/netcdf/nctraj.xhtml, accessed 2016.11.21.
[2] CHARMM, (Chemistry at HARvard Macromolecular Mechanics), https://www.charmm.org/, accessed 2016.11.21; LAMMPS Molecular Dynamics Simulator http://lammps.sandia.gov/, accessed 2016.11.21.
[3] AMBER "PARM" parameter/topology file specification, http://ambermd.org/formats.html#topology, accessed 2016.11.21.
[4] PDB File Format, http://www.rcsb.org/pdb/static.do?p=file_formats/pdb/index.html, accessed 2016.11.21.
[5] CatDCD - Concatenate DCD files, http://www.ks.uiuc.edu/Development/MDTools/catdcd/, accessed 2016.11.21.
[6] Barber, C.B., *et al.*, (1996). The Quickhull algorithm for convex hulls. ACM Transactions on Mathematical Software, *22 (4)*, 469-483.
[7] Savitzky, A. and Golay, M. J. E., (1964). Smoothing and Differentiation of Data by Simplified Least Squares Procedures. Anal. Chem., *36 (8)*, 1627–1639.
[8] Comaniciu, D., and Meer, P., (2002). Mean shift: a robust approach toward feature space analysis, IEEE Transactions on Pattern Analysis and Machine Intelligence *24(5)*, 603-619.
[9] Pedregosa, F., *et al.*, (2011). Scikit-learn: Machine Learning in Python. JMLR *12*, 2825-2830.
[10] Ester, M., *et al.* (1996) A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise Published in Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96), 226-231.
[11] Frey, B.J., and Dueck, D., (2007). Clustering by Passing Messages Between Data Points. Science, *315(5814)*, 972-976.
[12] Zhang, T., *et al.*, (1996). BIRCH: An efficient data clustering method for large databases Proceedings of the 1996 ACM SIGMOD international conference on Management of data 103-114 and Perdisci, R., JBirch - Java implementation of BIRCH clustering algorithm (https://code.google.com/archive/p/jbirch).
[13] David, A., and Vassilvitskii, S., (2007). k-means++: The advantages of careful seeding. Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms, Society for Industrial and Applied Mathematics 1027-1035.
[14] Case, D.A., *et al.*, (2014). AMBER 14, University of California, San Francisco.
[15] Sindhikara, D. J., *et al.*, (2012). Placevent: An algorithm for prediction of explicit solvent atom distribution—Application to HIV-1 protease and F-ATP synthase. J. Comput. Chem. *33*, 1536–1543.