



PyMOL plugin v3.0

User Guide



The CAVER 3.0 software package is distributed under the terms of the GNU General Public License v3.0. There is no warranty for the program, to the extent permitted by applicable law. Except when otherwise stated in writing the copyright holders and/or other parties provide the program “as is” without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The entire risk as to the quality and performance of the program is with you. Should the program prove defective, you assume the cost of all necessary servicing, repair or correction.

Citation:

Chovancova, E., Pavelka, A., Benes, P., Strnad, O., Brezovsky, J., Kozlikova, B., Gora, A., Sustr, V., Klvana, M., Medek, P., Biedermannova, L., Sochor, J. Damborsky, J. (2012). CAVER 3.0: A Tool for the Analysis of Transport Pathways in Dynamic Protein Structures. *PLoS Comput Biol* **8**: e1002708. doi:10.1371/journal.pcbi.1002708

Developer Team:

Human Computer Interaction Laboratory

Faculty of Informatics, Masaryk University

Botanická 68a, 602 00 Brno, Czech Republic

Webpage: <http://decibel.fi.muni.cz>

Phone: +420 549 496 939, Fax: +420 549 491 820

Loschmidt Laboratories, Department of Experimental Biology and
Research Centre for Toxic Compounds in the Environment

Faculty of Science, Masaryk University

Kamenice 5, Bld. A13, 625 00 Brno, Czech Republic

Webpage: <http://loschmidt.chemi.muni.cz>

Phone: +420 549 493 467, Fax: +420 549 496 302

Webpage: <http://www.caver.cz>

E-mail: caver@caver.cz

© Copyright 2005-2013

Human Computer Interaction Laboratory, Faculty of Informatics
and Loschmidt Laboratories, Department of Experimental Biology and
Research Centre for Toxic Compounds in the Environment, Faculty of Science

Masaryk University

Brno, Czech Republic

Contents

A. ABOUT	4
B. INSTALLATION	4
C. RUNNING CAVER PLUGIN	5
D. CALCULATION SETTINGS	6
Tunnel calculation	6
Starting point optimization	9
E. INPUTS	10
PDB file	10
Configuration file (config.txt)	10
F. OUTPUTS	11
summary.txt	11
summary_precise_numbers.csv	11
log.txt	11
warnings.txt	11
analysis <i>directory</i>	12
pymol <i>directory</i>	15
data <i>directory</i>	15
inputs <i>directory</i>	16
G. ADVANCED SETTINGS	17
Advanced calculation settings	17
Atomic radii settings	17
Visualization settings	17
H. DESCRIPTION OF CAVER PLUGIN VERSIONS	18

A. ABOUT

The CAVER PyMOL plugin v3.0 enables calculation and visualization of tunnels in PyMOL. For the calculation of tunnels, the plugin utilizes CAVER 3.0 software package.

CAVER plugin is written in the Python programming language and runs on all operating systems. We recommend you to use Python 2.7 or higher.

CAVER 3.0 is written in the Java programming language and runs on all operating systems with installed Java Runtime Environment 6.0 or higher. To check the version of Java installed on your system, follow the instructions at <http://www.java.com/en/download/help/testvm.jsp>.

PyMOL (<http://www.pymol.org/>) is a widely used molecular visualization system and is easily extensible by the Python programming language. The CAVER plugin has been tested with PyMOL v0.99 – PyMOL v1.5

Please report any incompatibilities to caver@caver.cz

B. INSTALLATION

CAVER plugin can be freely downloaded from <http://www.caver.cz/>. Download the **caver_3.0_plugin.zip** package and decompress it into any directory in your computer.

Windows

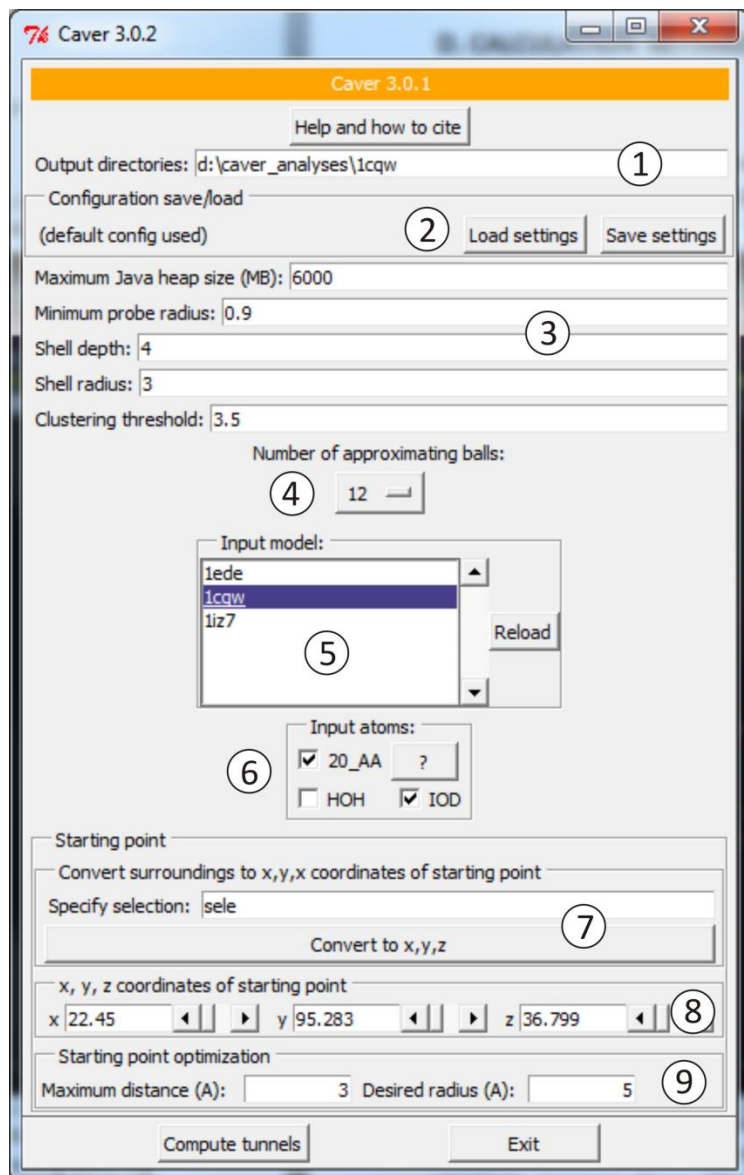
1. Run "windows/setup_301.exe" and follow the instructions. Make sure that you correctly specified the directory with the PyMOL executable.
2. If you have an appropriate version of Java and Python installed on your system, you can directly start to use CAVER plugin (see C. Running CAVER plugin).

Linux/MacOS

1. move "linux_mac/Caver3" folder to some destination (e.g.: /bin/Caver3)
2. edit "linux_mac/ Caver3_0_1.py with a text editor:
replace "directory/where/caver3/is/located" (line 64) with the path used in the previous step.
Note: check that caver.jar and config.txt files are located directly in the specified directory (e.g., /bin/Caver3/caver.jar)
3. run PyMOL as root and click "Plugin → Install plugin..." (or "Plugin → Manage plugins → Install..." in newer versions of PyMOL) and select previously edited "Caver3_0_1.py"
4. If you have an appropriate version of Java and Python installed on your system, you can directly start to use CAVER plugin (see C. Running CAVER plugin).

C. RUNNING CAVER PLUGIN

Once installed, CAVER plugin can be launched from the top menu of PyMOL: Plugin → Caver 3.0.1:



1. location of the directory for storing the calculation output files
2. buttons for loading/saving calculation configuration file
3. tunnel calculation settings
4. specification of the number of balls used for the approximation of the input structure atoms
5. selection of the input structure
6. specification of atoms to be used for the calculation
7. specification of the calculation starting point by defining the name of the PyMOL selection (this selection typically contains residues from the surroundings of the desired starting point, e.g. catalytic residues); the specified selection has to be converted to the X, Y, Z coordinates by user
8. specification of the starting point by defining its X, Y, Z coordinates in the three-dimensional space
9. starting point optimization parameters

D. CALCULATION SETTINGS

Tunnel calculation

Before you start the calculation, make sure that you have enough operating memory to allocate heap space specified by the **Maximum Java heap size** parameter. Prior to the calculation of the Voronoi diagram, all atoms of the input structure, which are larger than the smallest atom within the structure, are approximated by a user-specified number of balls with the van der Waals (VDW) radius equal to the VDW radius of the smallest atom. The **Number of approximating balls** parameter can be used to specify by how many balls will be individual atoms approximated. By specifying the **Shell radius** and **Shell depth** parameters, you can influence the definition of the protein molecular surface, while the **Minimum probe radius** parameter specifies the minimum width of tunnels you are interested in. Several tunnels with similar axes may be identified within one input structure. All identified pathways are clustered and only the cheapest tunnel (i.e., the lowest cost tunnel) is retained for each cluster. The clustering result is stored as a tree hierarchy of tunnel clusters. The size of the resulting clusters may be optimized by cutting this tree at a varying level of detail, which is specified by the **Clustering threshold** parameter.

Maximum Java heap size 6000 | number $n > 0$

specifies the maximum possible Java heap size (allocated memory). The values which are below the specified value and are in the following list are tested and the highest value that works in a given system is used: [500, 800, 900, 950, 1000, 1050, 1100, 1150, 1200, 1250, 1300, 1400, 1500, 2000, 3000, 4000, 5000, 6000, 8000, 10000, 14000, 16000, 20000, 32000, 48000, 64000]. This list can be edited in plugin source code: `path_to_PyMOL\modules\pmsg_tk\startup\Caver3_0_1.py`. **Important:** Increase the maximum heap size if you encounter “out of memory” error. If you want to use heap size larger than 1200 MB, you will probably need to use 64bit java.

Minimum probe radius 0.9 | number $n > 0$

specifies the minimum radius the tunnel must have to be identified (all edges of the Voronoi diagram which cannot be traversed by the probe of the specified *minimum probe radius* are removed). **Important:** To identify temporarily narrowed tunnels, the *minimum probe radius* parameter should be decreased. **Warning:** Using small values slows down the computation, and reports many irrelevant tunnels, as such tunnels exist nearly everywhere.

- *Minimum probe radius 0.9* (default): Only the tunnels with the bottleneck (minimum) radius ≥ 0.9 Å will be identified.

Shell depth 4 | number $n > 0$

specifies the maximal depth of a surface region, i.e., a part of the input structure located below the bulk solvent region. The tunnel branching is not allowed in the surface region (Figure 1). **Warning:** The tunnel branching is prohibited within the surface region. Therefore make sure that all important tunnel branches are identified under your settings. If some important tunnel or some important tunnel branch is not identified, the *shell depth* parameter is probably set to a too high value and should be decreased (Figure 1A). **Important:** This parameter was implemented to minimize

overshadowing of tunnels, and especially identification of irrelevant tunnel branches appearing as tunnels slithering on the protein surface (Figure 2). Identification of many irrelevant tunnels slithering on the protein surface indicates that the *shell depth* value is probably not set correctly and should be increased.

- *Shell depth 4* (default): The maximal depth of the surface region will be 4 Å.

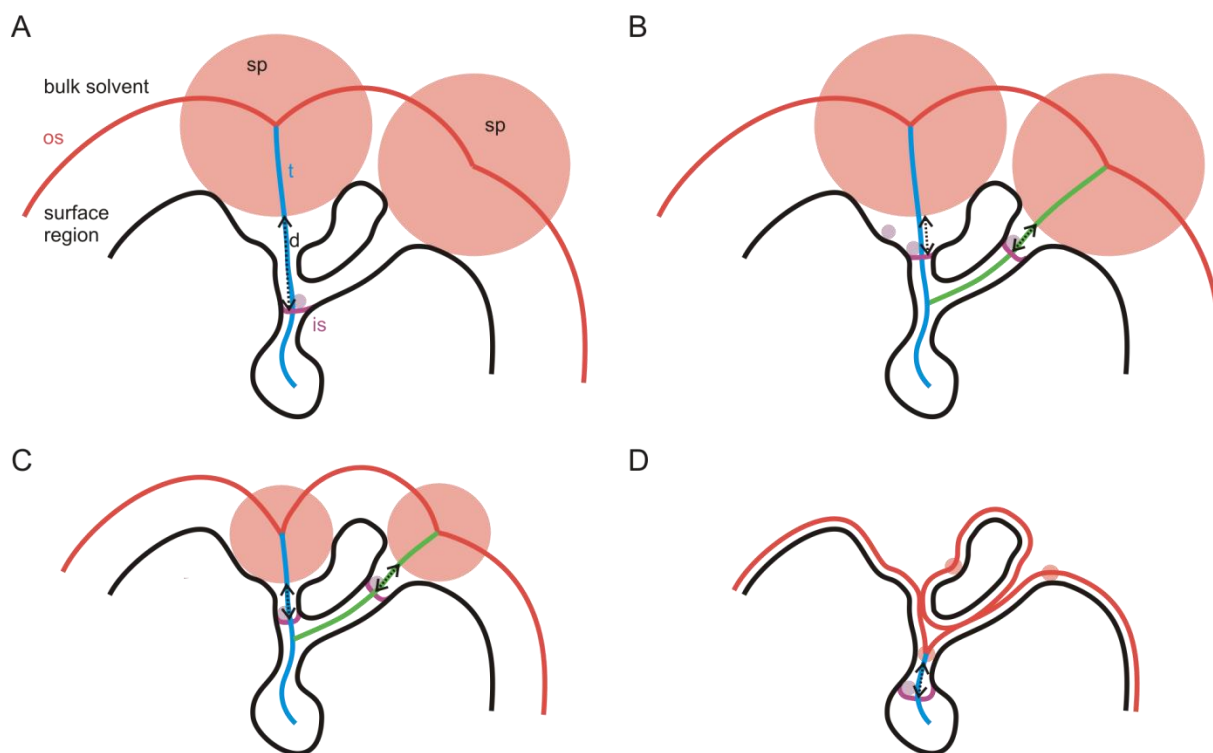


Figure 1. Different settings of the *shell radius* and *shell depth* parameters. The shell probe (**sp**; orange color) with the radius specified by the *shell radius* parameter defines the outer surface (**os**; red color) of the structure (i.e., boundary between the molecule and bulk solvent). The tunnel branching is not allowed in the surface region, which is defined as the region between the outer surface and inner surface (**is**; violet color) of the molecule. The maximal depth (**d**; dotted arrow) of the surface region is specified by the *shell depth* parameter. Note that center-lines of all identified tunnels (**t**; blue and green colors) are going through regions fully accessible to the probe (**p**; light violet) with the radius specified by the *minimum probe radius* parameter.

(A) Setting the *shell depth* parameter to a too high value may lead to overlooking of important tunnels or tunnel branches. (B) Decrease of the *shell depth* value enables identification of both important tunnels. (C) The *shell radius* should be set to a value higher than maximum radii of individual tunnels, (D) otherwise part of the molecule will be considered as the bulk solvent and misleading results will be obtained.

Shell radius 3 | number $n > 0$

specifies the radius of the shell probe which will be used to define which parts of the Voronoi diagram represent the bulk solvent (Figure 1). **Warning:** The value of *shell radius* must be higher than the bottleneck radius of each tunnel in the input structure, otherwise no tunnel will be identified (all tunnels with the bottleneck radius larger than the *shell radius*, and hence also the calculation starting

point, will be assigned as the bulk solvent). To enable identification of tunnels, the *shell radius* is in such cases set to the value of the largest bottleneck radius in a given snapshot multiplied by 2 (default setting). **Important:** The value of *shell radius* should be higher than maximum radii of individual tunnels (Figure 1C). Therefore, this value may significantly differ between individual systems and you should adjust it to suit your target system (e.g., the optimal *shell radius* for transmembrane channel proteins is usually much larger than for globular proteins).

- *Shell radius* 3 (default): The probe of 3 Å radius will be used to define the bulk solvent region (all vertices in the Voronoi diagram that can be accessed from protein exterior by this probe will be considered as the bulk solvent vertices). **Warning:** The default value is rarely suitable for transmembrane proteins or any other proteins with wide channels or tunnels.

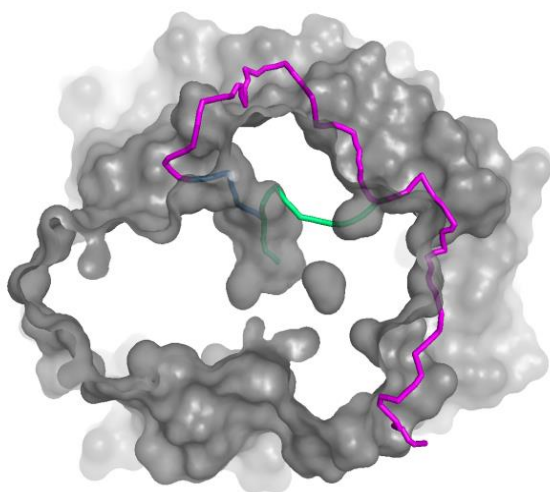


Figure 2. The example of an irrelevant tunnel branch (a side branch of the cyan tunnel) slithering on the protein surface (in violet).

Clustering threshold 3.5 | number $n \geq 0$

specifies the level of detail at which the tree hierarchy of tunnel clusters will be cut, and thus influences the size of resulting clusters. The higher the *clustering threshold*, the larger the resulting clusters.

- *Clustering threshold* 3.5 (default): the tree hierarchy of tunnel clusters will be cut at the value of 3.5

Number of approximating balls 20 | 12 | 8 | 6 | 4

specifies the number of balls that will be placed right under the surface of each larger atom to represent individual atoms in the input structure. **Important:** Approximation of larger atoms by a set of balls with equal radii allows to construct an ordinary Voronoi diagram, that captures geometry of tunnels more precisely than if one ball would be used for each atom. However, the method still leads to a certain overestimation of tunnel radii. **Warning:** Using high number of balls for the approximation increase the accuracy of results, but also the computational time and demands on

memory. Try to decrease the number of balls if you encounter problems during construction of the Voronoi diagram (may happen for large molecular systems with high number of atoms).

- *Number of approximating balls* 12 (default): Selected atoms will be approximated by 13 balls (1 central ball + 12 balls placed under the atom's surface) with the radius equal to the van der Waals radius of the smallest atom in the input structure.

Starting point optimization

The calculation starting point is initially placed into the center of gravity of the user-specified residues or atoms (specified by PyMOL selection) or a point defined by x, y and z coordinates. The calculation starting point (i.e., starting Voronoi vertex) is then identified in the vicinity of the initial starting point based on the criteria specified by the **Maximum distance** and **Desired radius** parameters.

Maximum distance 3 | $n \geq 0$

specifies the maximal distance of the calculation starting point (i.e. the starting Voronoi vertex) from the initial starting point. If no vertex is found within the user-specified distance, the default value of *Maximum distance* is used. If still no vertex is found, the vertex closest to the initial starting point is selected as the calculation starting point. **Warning:** Simultaneous setting of the *Maximum distance* and *Desired radius* parameters to high values may displace the calculation starting point outside of the input structure.

- *Maximum distance 3* (default): The calculation starting point should be localized within the 3 Å distance from the initial starting point. If no vertex is found, the vertex closest to the initial starting point will be selected as the calculation starting point.
- *Maximum distance 1.5*: The calculation starting point should be localized within the 1.5 Å distance from the initial starting point. If no vertex is found within the 1.5 Å distance, the calculation starting point will be searched within the 3 Å distance. If no vertex is found within the 3 Å distance, the vertex closest to the initial starting point will be selected as the calculation starting point.

Desired radius 5.0 | $n \geq 0$

The closest Voronoi vertex to the initial starting point, which is located within the specified distance (see the *Maximum distance* parameter) from the initial starting point and at least *Desired radius* far from the balls representing the input structure, will be used as the starting point for the calculation of tunnels. If no such vertex exists, then the vertex with maximal distance to the balls representing the input structure will be selected from all vertices which are located within the specified distance from the initial starting point. **Tip:** Setting of *Desired radius* to a high value will place the starting point to the center of the largest free space within the *Maximum distance* area. **Warning:** Simultaneous setting of the *Maximum distance* and *Desired radius* parameters to high values may displace the calculation starting point outside of the input structure.

- *Desired radius 5* (default): The minimum distance between the starting Voronoi vertex and the balls representing the input structure should be 5 Å. If no such Voronoi vertex can be found, the vertex located within the specified distance from the initial starting and the maximal distance to the input structure will be used as the starting Voronoi vertex.

E. INPUTS

PDB file

As an input, you have to provide a structure in the PDB format. The atoms that you do not want to include in the CAVER 3.0 analysis (e.g, ligands, water molecules, etc.) can be removed manually beforehand or by checking/unchecking a respective box in the plugin. If alternate conformations of residues are detected, they are either automatically removed by the program before the analysis (for each residue, the alternate conformation with the highest occupancy is retained), or in some specific cases, you will be asked to remove them manually. **Important:** The name of the PDB file is used for initialization of the pseudo-random number generator. If you want to reproduce the results of previous analyses precisely, you have to use the input files with exactly same names.

Configuration file (config.txt)

The main calculation parameters can be set directly in the plugin (see D. Calculation settings). Besides that, you can adjust many other calculation parameters by editing the calculation configuration file (the description of all CAVER 3.0 parameters is provided in CAVER 3.0 user guide which is part of the CAVER 3.0 command line package downloadable at <http://www.caver.cz/>). For saving or loading configuration files, use the “Save settings” and “Load settings” buttons in the plugin. Alternatively, you can edit the default configuration file (see G. Advanced settings).

F. OUTPUTS

The results are stored in the "**caver_output/calculation_id**" directory created in the directory specified in the "Output directories" field of the plugin.

summary.txt

Location: `caver_output/calculation_id/summary.txt`

File with the summary information about all identified tunnels.

- **ID**: The ID of a given tunnel (= ranking of a given tunnel based on the Priority).
- **No**: The total number of tunnels belonging to a given cluster (not relevant for the analysis of static structures).
- **No_snaps**: The number of snapshots with at least one tunnel with a radius larger or equal to the value specified by the *minimum probe radius* paramter (not relevant for the analysis of static structures).
- **Avg_BR**: The bottleneck radius of a given tunnel.
- **Max_BR**: The maximum bottleneck radius in a given tunnel cluster (not relevant for the analysis of static structures).
- **Avg_L**: The tunnel length.
- **Avg_C**: The tunnel curvature (*curvature = length/distance*, where *length* is the length of the tunnel and *distance* is the shortest possible distance between the calculation starting point and the tunnel ending point).
- **Priority**: The priority of a given tunnel
- **Avg_throughput**: The tunnel throughput
- **Avg_up_E_BR**: The upper error bound of bottleneck radius estimation.
- **Avg_up_E_TR**: The upper error bound of tunnel profile radii estimation.
- **Max_up_E_BR**: The maximal upper errors bound of bottleneck radii estimation (not relevant for the analysis of static structures).
- **Max_up_E_TR**: The maximal upper error bound of tunnel radii estimation.
- **SD**: The standard deviation of a value listed in the column preceding the SD column.

summary_precise_numbers.csv

Location: `caver_output/calculation_id/summary_precise_numbers.csv`

File containing precise numbers of Priority and throughput of individual tunnels (these numbers are rounded in the summary.txt file).

log.txt

Location: `caver_output/calculation_id/log.txt`

File containing the log of the calculation.

warnings.txt

Location: `caver_output/calculation_id/warnings.txt`

File containing warning messages from the calculation.

analysis directory

tunnel_characteristics.csv

Location: `caver_output/calculation_id/analysis/tunnel_characteristics.csv`

The CSV file containing information about characteristics of individual tunnel:

- **Snapshot:** The name of the input structure
- **Tunnel cluster:** The ID of a tunnel cluster to which a given tunnel belongs (not relevant for the analysis of static structures).
- **Tunnel:** ID of a given tunnel.
- **Throughput:** The throughput of a given tunnel (throughput = e^{-cost}).
- **Cost:** The cost of a given tunnel.
- **Bottleneck radius:** The radius of the bottleneck, i.e., the narrowest part, of a given tunnel.
- **Bottleneck R error bound:** The upper error bound for estimation of the bottleneck radius of a given tunnel, i.e., the maximal possible overestimation of the bottleneck radius due to approximation of input structure.
- **Length:** The length of a given tunnel.
- **Curvature:** The curvature of a given tunnel; $curvature = length/distance$, where *length* is the length of the tunnel (distance from the calculation starting point to the tunnel ending point calculated along the tunnel axis) and *distance* is the shortest possible distance between the calculation starting point and the tunnel ending point.

tunnel_profiles.csv

Location: `caver_output/calculation_id/analysis/tunnel_profiles.csv`

The CSV file with data for plotting the profiles of individual tunnels:

- **Snapshot:** The name of the input structure.
- **Tunnel cluster:** The ID of a tunnel cluster to which a given tunnel belongs (not relevant for the analysis of static structures).
- **Tunnel:** ID of a given tunnel.
- **Throughput:** The throughput of a given tunnel (throughput = e^{-cost}).
- **Cost:** The cost of a given tunnel.
- **Bottleneck radius:** The radius of the bottleneck, i.e., the narrowest part, of a given tunnel.
- **Average R error bound:** Average upper error bound for estimation of radii in a given tunnel profile.
- **Max. R error bound:** The maximal upper error bound for estimation of radii in a given tunnel profile.
- **Bottleneck R error bound:** The upper error bound for estimation of the bottleneck radius of a given tunnel, i.e., the maximal possible overestimation of the bottleneck radius.
- **Curvature:** The curvature of a given tunnel; $curvature = length/distance$, where *length* is the length of the tunnel (distance from the calculations starting point to the tunnel ending point

calculated along the tunnel axis) and *distance* is the shortest possible distance between the calculation starting point and the tunnel ending point.

- **Length:** The length of a given tunnel.
- **Axis:** Characteristics reported for each point of the tunnel profile.
- **Values...**
 - Axis X: x coordinate of a given point of the tunnel axis.
 - Axis Y: y coordinate of a given point of the tunnel axis.
 - Axis Z: z coordinate of a given point of the tunnel axis.
 - Axis distance: shortest possible distance between a given point of the tunnel axis from the calculation starting point.
 - Axis length: distance between a given point of the tunnel axis from the calculation starting calculated along the tunnel axis.
 - Axis R: tunnel radius in a given point of the tunnel axis, i.e., the radius of the largest possible ball that can be centered at a given point of the tunnel axis without colliding with the input structure.
 - Axis Upper limit of R overestimation: the upper error bound for estimation of the tunnel radius in a given point of the tunnel axis, i.e., the maximal possible overestimation of the tunnel radius in a given point.

Important: The tunnel profile is usually plotted as the tunnel radius over the tunnel length (i.e., the distance between the point and the calculation starting point measured along the tunnel axis). Alternatively, you can plot the tunnel radius over the shortest possible distance of the profile point from the calculation starting point. By this way, you usually obtain profiles that are better aligned to each other than profiles calculated along the tunnel axis.

atoms.txt

Location: `caver_output/calculation_id/analysis/atoms.txt`

The list of tunnel-lining atoms (atom serial numbers). For each tunnel, all atoms located within the specified distance from some ball of a given tunnel are reported.

residues.txt

Location: `caver_output/calculation_id/analysis/residues.txt`

The list of tunnel-lining residues. For each tunnel cluster, all residues located within the specified distance from some ball of a given tunnel are reported.

- **C:** ID of a chain to which a given residue belongs.
- **res:** Index of a given residue in the PDB file.
- **AA:** Residue name.
- **N:** Number of snapshots in which a given residue lined a given tunnel (not relevant for the analysis of static structures).
- **sideN:** Specifies whether at least one side chain atom of a given residue lined a given tunnel (1 = yes, 0 = no). For this purpose, all atoms except those named H, N, C, O, CA or HA are considered as side chain atoms.

- **atoms:** Tunnel-lining atoms. Format is <number of snapshots – not relevant for the analysis of static structures>:<atom name>_<atom serial number> (e.g., 2:O_582—the atom 582(O))

bottlenecks.csv

Location `caver_output/calculation_id/analysis/bottlenecks.csv`

The list of bottleneck residues of individual tunnels. All residues located within the specified distance from the bottleneck of a given tunnel are reported as its bottleneck residues.

- **Snapshot:** The name of the input structure.
- **Tunnel cluster:** The ID of a tunnel cluster to which a given tunnel belongs (not relevant for the analysis of static structures).
- **Tunnel:** ID of a given tunnel.
- **Throughput:** The throughput of a given tunnel ($\text{throughput} = e^{-\text{cost}}$).
- **Bottleneck X:** x-coordinate of the bottleneck of a given tunnel.
- **Bottleneck Y:** y-coordinate of the bottleneck of a given tunnel.
- **Bottleneck Z:** z-coordinate of the bottleneck of a given tunnel.
- **Bottleneck R:** The radius of the bottleneck of a given tunnel.
- **Bottleneck residues:** residues located within the specified distance from the bottleneck of a given tunnel; bottleneck residues are ordered from the closest to the most distant ones. Format is chain:residue index (e.g., B:152—residue 152 from the chain B).

profile_heat_maps directory

Location: `caver_output/calculation_id/analysis/profile_heat_maps`

The directory containing heat maps, i.e., figures visualizing the time evolution of the profiles of individual tunnels.

- **cl_CLUSTER_ID_profile_heat_map.png** (e.g., `cl_000001_profile_heat_map.png`): Heat map showing the profile of a given tunnel. The color of an element with coordinates x and y expresses the radius of a given tunnel in $x-1^{\text{th}}$ snapshot and y^{th} distance interval (Figure 3).
- **average_images:** Directory containing the average profile heat map of each tunnel cluster (not relevant for the analysis of static structures)
- **csv:** Directory containing source data for plotting of the profile heat maps.

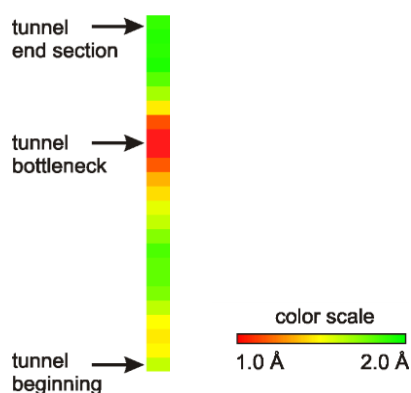


Figure 3. Heat map visualizing the tunnel profile. The color of an element with coordinates y expresses the radius of a given tunnel in y^{th} distance interval.

pymol directory

view_plugin.py

Location: `caver_output/calculation_id/pymol/view_plugin.py`

The script for re-opening of the results in PyMOL. The script opens the files from the `out/data/clusters_timeless/` directory together with the `out/data/origins.pdb` and `out/data/v_origins.pdb` files. You can run the script either by clicking the `view_plugin.py` file (you have to associate the extension “.py” with PyMOL first) or run the script via the PyMOL menu (File > Run > `view_plugin.py`) or type “run `view_plugin.py`” in the PyMOL command line. **Important:** The script opens the calculated tunnels but does not open the input structure, so you have to load it manually.

view_timeless.py

Location: `caver_output/calculation_id/pymol/view_timeless.py`

The script for opening the results in PyMOL. The tunnels are by default visualized as their center-lines, the input structure as cartoon. The script opens the files from the `out/data/clusters_timeless/` directory together with the `out/data/origins.pdb` and `out/data/v_origins.pdb` files. You can run the script either by clicking the `view_timeless.py` file (you have to associate the extension “.py” with PyMOL first) or run the script via the PyMOL menu (File > Run > `view_timeless.py`) or type “run `view_timeless.py`” in the PyMOL command line.

atoms.py

Location: `caver_output/calculation_id/pymol/atoms.py`

The script for the selection of tunnel-lining atoms in PyMOL. You can run the script via the PyMOL menu (File > Run > `atoms.py`) or type “run `atoms.py`” in the PyMOL command line.

data directory

clusters_timeless directory

Location: `caver_output/calculation_id/data/clusters_timeless`

The directory with PDB files, each representing one tunnel. All PDB files may be opened at once in PyMOL using the `caver_output/calculation_id/pymol/view_timeless.py` or `caver_output/calculation_id/pymol/view_plugin.py` script.

tree.txt

Location: `caver_output/calculation_id/data/tree.txt`

The file containing a tree hierarchy of identified tunnels (i.e., clustering results).

times.txt

Location: `caver_output/calculation_id/data/times.txt`

The file containing the calculation times of individual computation steps.

origins.pdb

Location: `caver_output/calculation_id/data/origins.pdb`

The PDB file with the initial starting point (specified by the user). The initial starting point is, together with the tunnels, visualized in PyMOL using the `caver_output/calculation_id/pymol/view_timeless.py` or `caver_output/calculation_id/pymol/view_plugin.py` scripts.

v_origins.pdb

Location: `caver_output/calculation_id/data/v_origins.pdb`

The PDB file with the calculation starting point. The calculation starting point is, together with the tunnels, visualized in PyMOL using the `caver_output/calculation_id/pymol/view_timeless.py` or `caver_output/calculation_id/pymol/view_plugin.py` scripts.

inputs *directory*

This directory contains the input PDB file and configuration file which was used for the calculation.

G. ADVANCED SETTINGS

Advanced calculation settings

By default, the plugin loads the calculation settings from the config.txt file stored in Caver3 directory. If you want to change the predefined calculation settings, go to the Caver3 directory (location of the directory is specified during the installation of the plugin) and edit the config.txt file. The description of all CAVER 3.0 parameters is provided in the CAVER 3.0 user guide which is part of the CAVER 3.0 command line package downloadable at <http://www.caver.cz/>.

Atomic radii settings

Atomic radii are stored in the atom_radii.csv file located in the Caver3/bin directory (location of Caver3 is specified during the plugin installation). Edit this file if you want to specify radii of atoms that are not set or change the predefined settings.

Visualization settings

Template files for generation of heat maps and visualization scripts are stored in the Caver3/bin directory (location of Caver3 is specified during the plugin installation).

Visualization templates for PyMOL

- **view_timeless.py**: template script for the visualization of the input structure and tunnel center-lines in PyMOL. Edit this file if you want to change the predefined visualization.
- **view_plugin.py**: template script for the visualization of identified tunnels in PyMOL. Edit this file if you want to change the predefined visualization.
- **rgb.py**: template script for the coloring of tunnels in PyMOL. Edit this file if you want to change the predefined coloring scheme.

Heat map palettes

- **palette.png**: color scale to be used for generating the profile heat map.

H. DESCRIPTION OF CAVER PLUGIN VERSIONS

Version 3.01

- Calculation of tunnels performed by CAVER 3.01
- Starting point and optimized starting vertex are displayed even when no tunnels are found.
- Changed meaning and label of field: Java heap size -> Maximum Java heap size. Maximum possible memory is allocated below specified value. Default maximum value is 6000. Only the values which are below specified value and are in the following list are tested, the highest value that works is used: [500, 800, 900, 950, 1000, 1050, 1100, 1150, 1200, 1250, 1300, 1400, 1500, 2000, 3000, 4000, 5000, 6000, 8000, 10000, 14000, 16000, 20000, 32000, 48000, 64000]. This list can be edited in plugin source code, which can be found e.g. in C:\Program Files\DeLano Scientific\PyMOL\modules\pmg_tk\startup.
- The starting point optimization procedure was corrected. Previously, the starting vertex was sometimes identified in too narrow cavity due to error in implementation. The change can lead to minor changes in the geometric properties of tunnels, similarly as if the parameter seed is modified. Return to the previous behavior is possible by setting 'correct_voronoi_diagram' to 'no' (this is not recommended).

Version 3.0 BETA

- Calculation of tunnels performed by CAVER 3.0