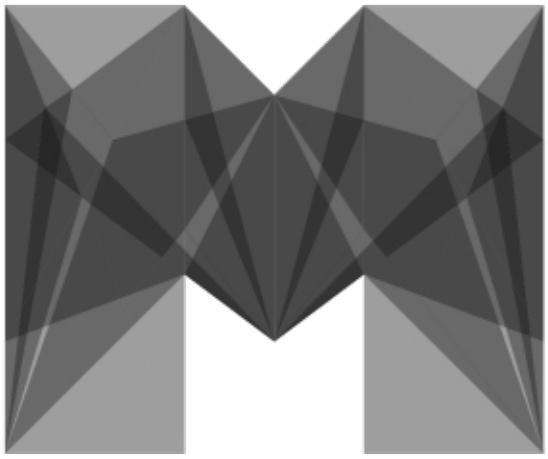

Mole 2.0

User manual

NATIONAL CENTRE NCBR for BIOMOLECULAR RESEARCH

CEITEC
Central European Institute of Technology
BRNO | CZECH REPUBLIC

Contents

1. Statement of purpose

The empty spaces in a protein or a protein complex can be classified as pockets, cavities, channels (tunnels) and pores (Figure 1). A pocket is a shallow depression on the molecular surface, and often serves as binding site for ligands or other biomolecules. A cavity is an empty space buried inside the protein structure. Cavities may communicate with the exterior environment via channels leading up to the protein surface, or may be completely isolated from the exterior environment (voids). Cavities often constitute enzymatic reaction sites, as they provide a highly controlled environment. A channel or tunnel is a ligand accessible pathway leading from the protein surface to the interior of a cavity. A pore is considered here as a channel that passes through the biomacromolecule from one point on the surface to another. Most pores serve as selective transport pathways across membranes.

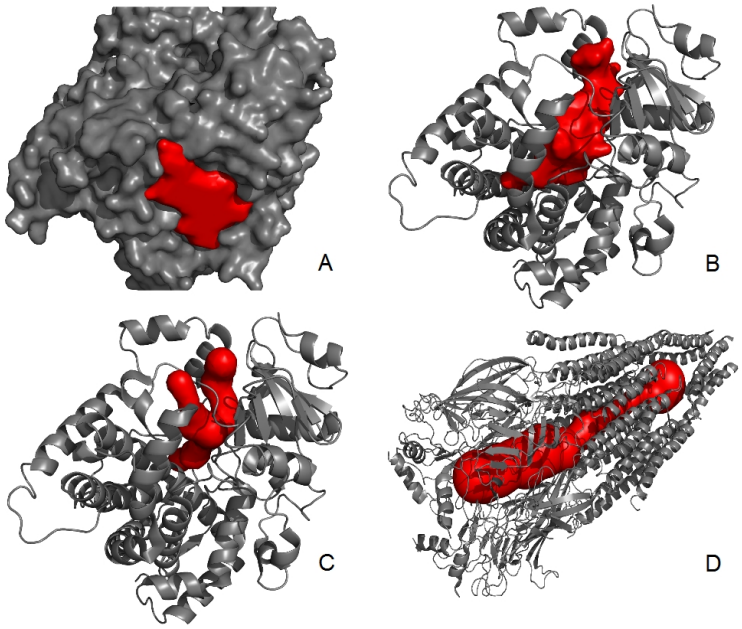

Figure 1: Types of empty spaces in proteins: A – pockets, B – cavities, C – tunnels, D – pores.

Mole 2.0 represents cutting edge software for a rapid detection and physico-chemical characterization of tunnels, pores and cavities in biomacromolecules, with emphasis on proteins and large protein-nucleic acids complexes. Mole 2.0 enables a brand new user experience, as it can be used effectively even without knowledge of the underlying algorithms. This tool comes in two versions, either as a Graphical User Interface (GUI) based application with an in-built molecular viewer, or as a standalone command-line application. The functionality of tunnel calculation and characterization is common to both versions of Mole 2.0, but there are some differences. The GUI-based application works under Windows, and allows the user to immediately visualize the results, as well as to refine the calculation in an interactive manner. The command-line application, which works under Windows, Linux and Mac, is not interactive, but the results can be exported in a form suitable for subsequent visualization in independent molecular viewers. Moreover, in combination with our PyMOL plug-in, the command line version of Mole 2.0 provides interactive functionality comparable to the GUI version under Windows, Mac OS and Linux. The command line application is ideal for batch processing, as it can be included in scripts which process large numbers of structures at a time, or which allow for automatic post-processing and integration of results.

2. How to use this manual

All you need to know in order to use Mole 2.0 effectively is covered by this material. The manual discusses the GUI based application first, where the majority of concepts, procedures and parameters are covered, and then the command line based application. Note that the part of the manual dedicated to the command line application discusses in detail only those concepts and procedures which are particular to the command line version of Mole 2.0.

Therefore, even if you plan to use only the command line application, please refer to Part I of the manual for a proper overview of functionality.

The manual follows the logical flow of operations that a user might wish to perform using Mole 2.0. For a full description of the algorithms, any questions you may have which cannot be answered by the present material, or any suggestions on how to improve Mole 2.0, please refer to our web pages at:

http://webchem.ncbr.muni.cz/Platform/App/Mole#

Enjoy working with Mole 2.0!

Part I

Mole 2.0, a Graphical User Interface based application

3. Technical details

3.1. Availability

The Mole 2.0 GUI based application is available free of charge on our website at http://webchem.ncbr.muni.cz/Platform/App/Mole#.

3.2. System requirements

The GUI based version of Mole 2.0 requires the Windows environment. More specifically, it requires the .NET framework 4.0 or above. This means that Mole 2.0 can run on any Windows operating system from Windows XP (Service Pack 3) and higher, or Windows Server 2003 (Service Pack 2) and higher.

3.3. Installation

It is not necessary to install Mole 2.0, as executable files are provided upon download. Simply download the .zip archive from our website, extract to a preferred location on your disk, and you can immediately start using Mole 2.0 by running the executable file mole.exe (Figures 2-4).

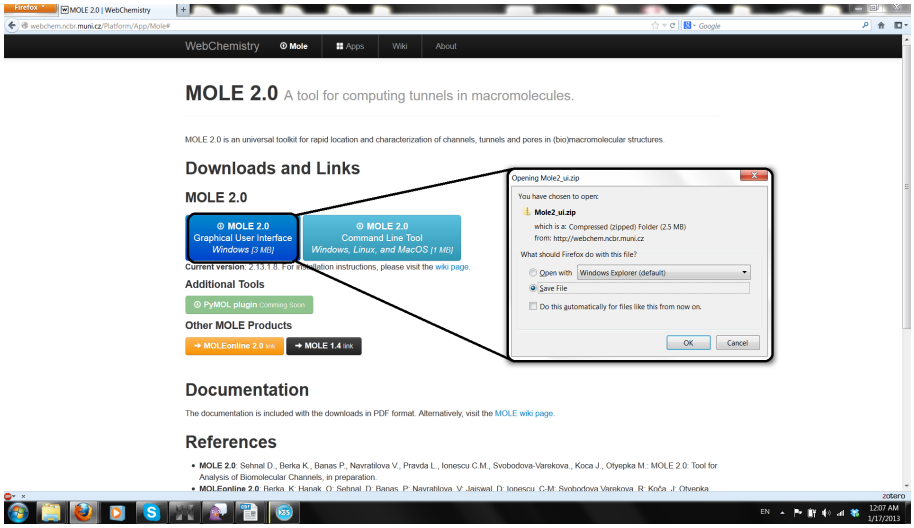

Figure 2: Download the GUI version of Mole 2.0 from our web page

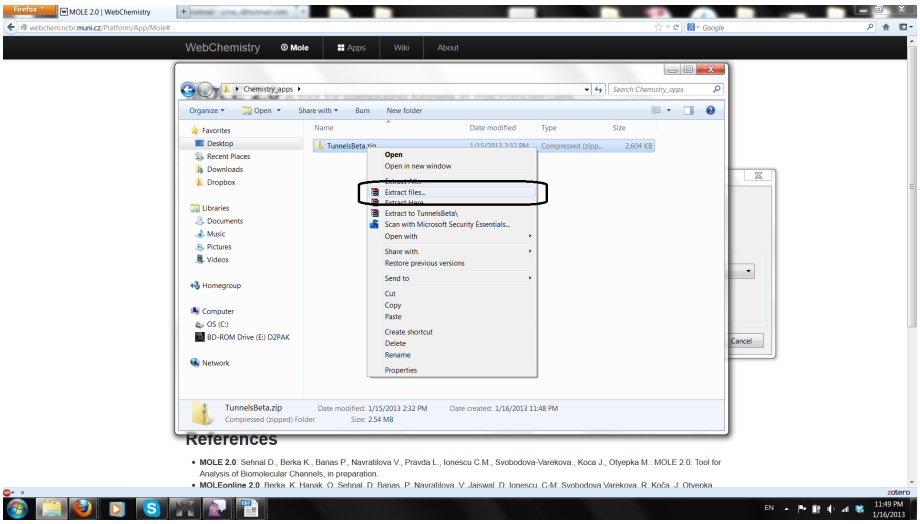

Figure 3: Unzip the files at your prefered location.

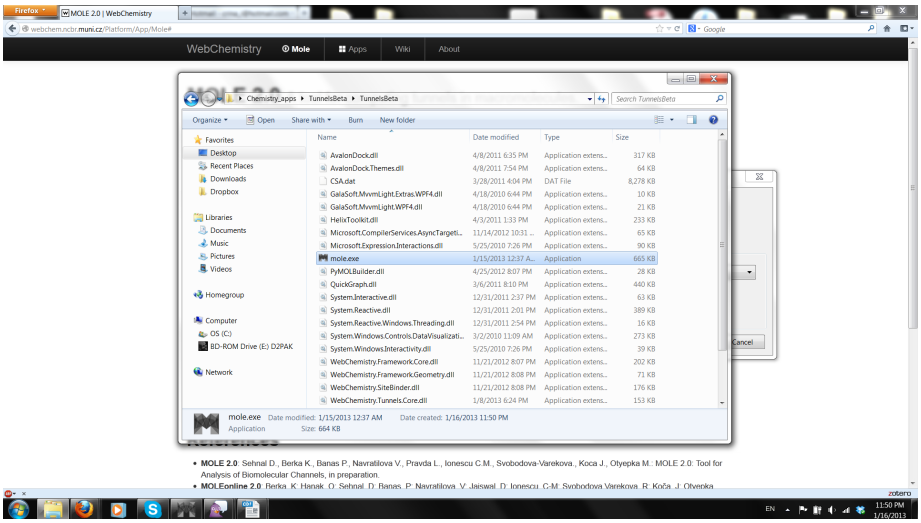

Figure 4: Start the Mole 2.0 GUI by running the file mole.exe.

After the execution of mole.exe, the starting screen should appear as seen in Figure 5. If you get an error during startup, please double check chapter 3.2. If your system meets all requirements and you still get an error, please contact us via our website at http://webchem.ncbr.muni.cz/Platform/App/Mole#.

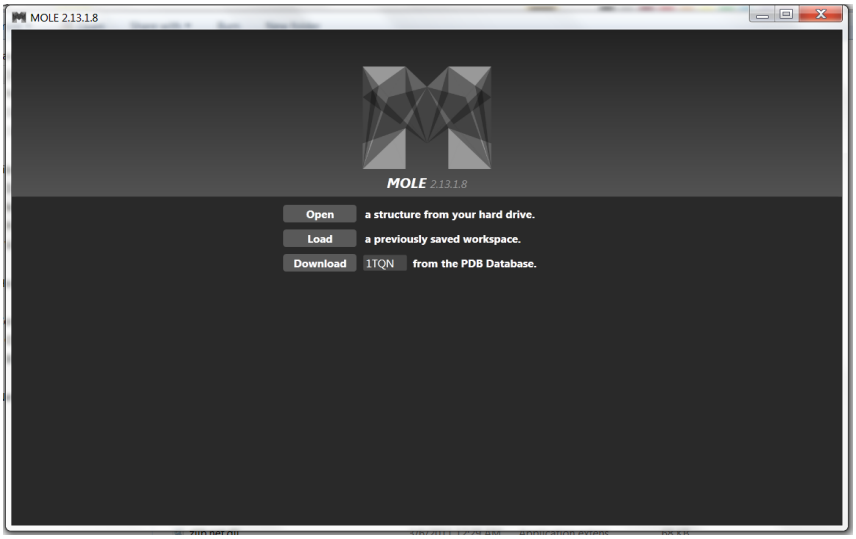

Figure 5: Start screen of the Mole 2.0 GUI.

4. Functionality

The Mole 2.0 GUI based application allows for the rapid and automated calculation of cavities, tunnels and pores in biomacromolecules, from the smallest polypeptides to the largest protein/nucleic acid complexes found in the Protein Data Bank. Moreover, it comes with a built-in molecular viewer, allowing for immediate visualization of results, as well as interactive refinement of the calculation. A remarkable advantage of Mole 2.0 is the fact that it allows you to interactively tweak the tunnel detection algorithm, such that the results are immediately available for inspection and comparison. Another useful feature of the GUI based version of Mole 2.0 is that it contains a lot of tool tips to help you along. If you are unsure about the meaning of a button, parameter or value, simply hover with the mouse cursor over it in order to get a basic explanation regarding that particular item.

4.1. Loading a structure

Immediately at startup, you have the opportunity to load a structure of interest (Figure 5). Molecular structures can be loaded into Mole 2.0 in the RCSB Protein Data Bank format (*.**pdb**). If the .pdb file contains more structural models, as is the case with structures determined by NMR experiments, only the first model will be used. If you wish to use a different model, you need to copy and save only the particular model of interest into a separate file, and load it separately into Mole 2.0. Alternatively, Mole 2.0 is able to find, retrieve and load any structure directly from the Protein Data Bank website if you provide it with the PDB ID and you are connected to the Internet. Only the first model will be loaded in the case of .pdb files containing more structural models. Note that you can load and work with more than one structure at a time. Structures can be loaded into Mole 2.0 at any time using the appropriate button (more about this in chapter 4.3.).

If you have used Mole 2.0 before and saved a workspace, you may load the saved workspace and all related content, including structures, at this time. A detailed explanation regarding workspaces can be found in chapter 4.3.

4.2. Automatic detection of cavities, tunnels and pores

In most cases, Mole 2.0 provides relevant results in a fully automated manner, without user intervention or the need to understand the algorithms involved in computing empty spaces in biomolecules. Simply loading a structure will trigger the automatic detection procedure (Figure 6), and the results will be available for visualization in several seconds. Once the structure is loaded, you can simply click the **Auto** buttons next to Tunnels and Pores, on the right hand side of the screen, and you will be able to instantly access the results as described below.

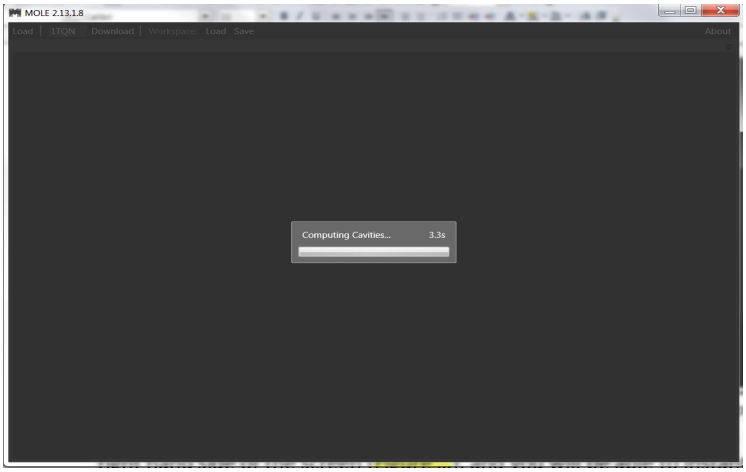

Figure 6: Automatic detection of cavities, tunnels and pores using Mole 2.0.

4.3. Visualizing results

First, it is important to mention that Mole 2.0 uses the concept of workspaces to define the visible area of the screen where results are displayed, along with all the loaded structures, the results themselves and various action

buttons (Figure 7). Any state of the workspace can be saved into a special file (***.wtw**), and used at a later time. The .wtw file stores all your results, parameter settings and even camera position, so that you can resume your work with Mole 2.0 at any time.

The area of the workspace can be divided into several main parts, namely the menu, in-built molecular viewer, residue selection bar, refinement panel, results panel and status bar.

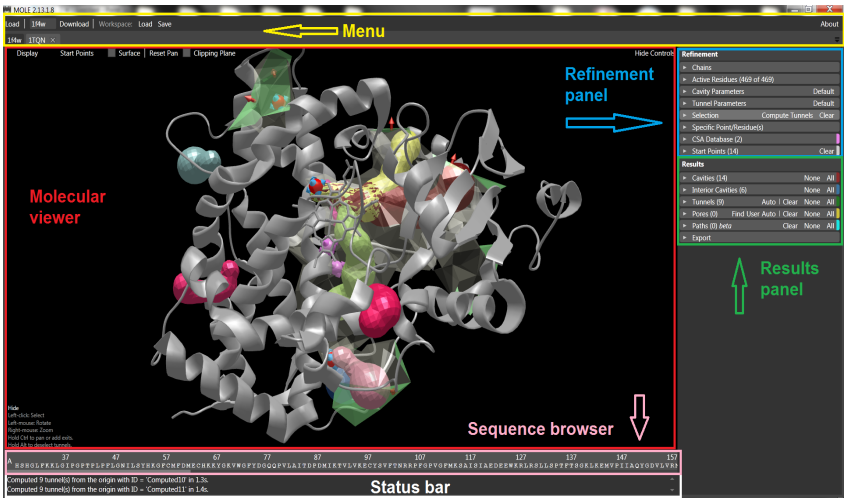

Figure 7: Workspace of the Mole 2.0 GUI.

4.3.1. Menu

The topmost part of the workspace contains the Mole 2.0 menu (Figure 8), which allows to perform general operations such as loading or downloading structures, and saving or loading workspaces. If more structures are loaded into the Mole 2.0 GUI-based version, each of them will appear in its own tab, and you can work with each structure completely independently.

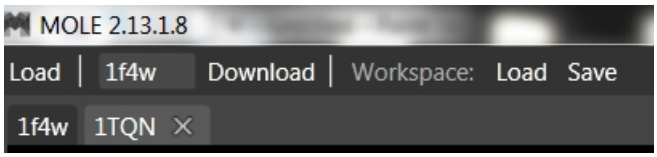

Figure 8: The menu of Mole 2.0 GUI contains commands to load or download structures, load or save workspaces, and switch between different structure tabs.

4.3.2. Molecular viewer

The largest part of the workspace contains the in-built molecular viewer and buttons related to visualization options (Figure 9). Here you can visualize various graphical elements such as loaded structures or any cavities, tunnels, and pores that Mole 2.0 has found. Additional graphical elements include the molecular surface, along with tunnel start points as balls with arrows, and tunnel exit points as tetrahedrons with arrows. At the bottom left corner of the viewer you can find a quick help box regarding the usage of the mouse inside the viewer (select, rotate, zoom, etc.). Do not forget to look for the tool tips whenever you feel uncertain.

The built-in molecular viewer also allows you to select residues (amino acids, ligands, solvent) by clicking on them directly inside the viewer. Such selections are useful for differentiated graphical representation, but also to refine the tunnel detection algorithms (for a complete discussion on tunnel starting points, please see chapter 4.6.). Note that whenever you hover over a residue in the viewer, that residue becomes highlighted in yellow, and its name and number are displayed at the bottom right corner of the viewer.

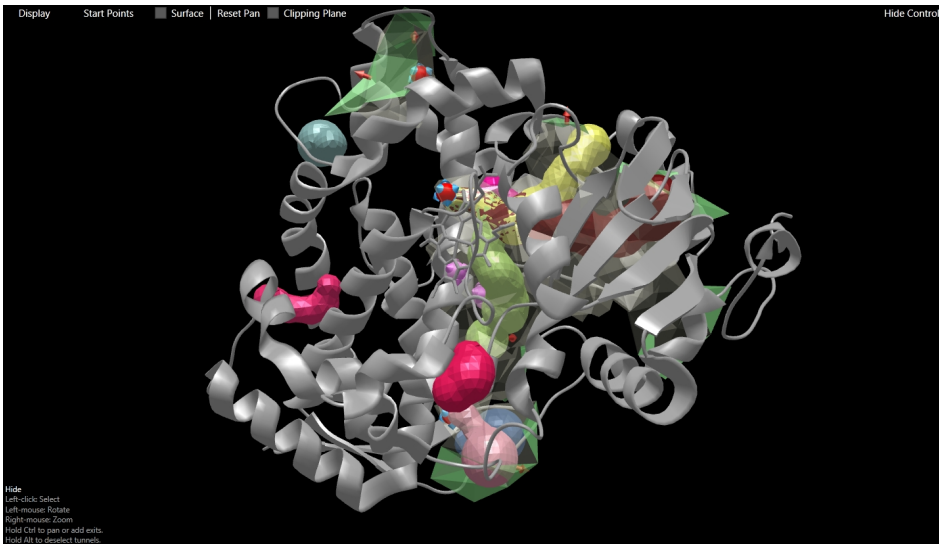

Figure 9: Molecular viewer of the Mole 2.0 GUI.

At the top of the molecular viewer there are various options for visualization (Figure 10). The **Display** menu contains controls for the type of molecular structure representation (e.g., cartoon, sticks), coloring scheme (e.g., by atom, by residue, etc.) and background color. The **Display** menu also allows to visualize non-protein atoms such as ligands, cofactors or solvent molecules. Additionally, the **Display** menu enables the representation of cavities as solid volumes, and the representation of tunnels as volumes or traces. The **Start Points** menu controls which types of tunnel start points will be displayed (for a complete discussion on tunnel starting points, please see chapter 4.6.). Additional visualization options that can be found at the top of the molecular viewer include the toggle for the display of molecular **surface**, and a **clipping plane** tool with a scroll bar which allows to explore three-dimensional structures in steps. Take some time to explore all visualization options.

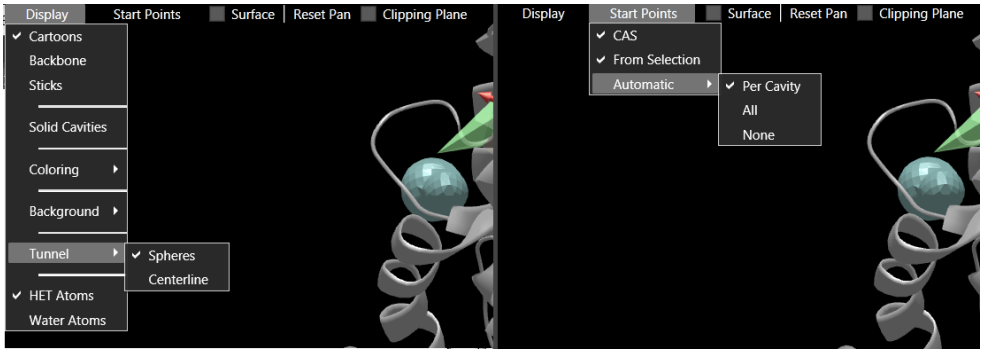

Figure 10: Detailed visualization options can be found at the top of the molecular viewer.

4.3.3. Sequence browser

Right under the in-built viewer there is a sequence browser (Figure 11), which contains the primary sequence of the loaded structure. Residues (amino acids, ligands, solvent) can be selected by clicking on their one-letter codes in the sequence. Such selections are useful for differentiated graphical representation, but also to refine the tunnel detection algorithms (for a complete discussion on refinement techniques, please see chapter 4.6.). Note that whenever you hover over a residue in the sequence browser, that residue becomes highlighted in yellow in the in-built molecular viewer.

Figure 11: The sequence browser of the Mole 2.0 GUI contains the list of residues in the structure.

4.3.4. Refinement panel

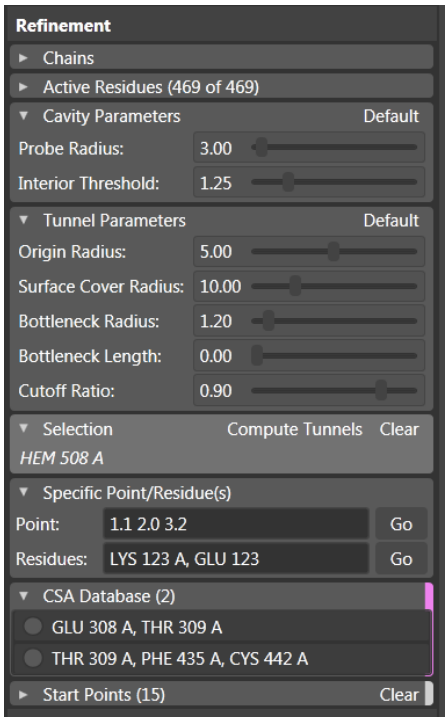

Figure 12: Refinement panel of the Mole 2.0 GUI.

The upper right part of the workspace (Figure 12) contains a set of controls that can be used to refine the cavity, tunnel and pore detection algorithms. We provide here only a brief overview of the purpose of each section. A full description of the concepts and procedures can be found in chapter 4.6.

In the sections **Chains** and **Active residues** you can restrict the areas of the protein where Mole 2.0 will search for cavities, tunnels and pores (Figure 13). The automatic algorithm uses the whole structure, but you may unselect whole chains or specific residues if you wish, and click the corresponding **Update** button to apply the changes. Note that whenever you hover over a chain or residue in the **Chains** or **Active residues** lists, those elements become highlighted in yellow in the in-built molecular viewer.

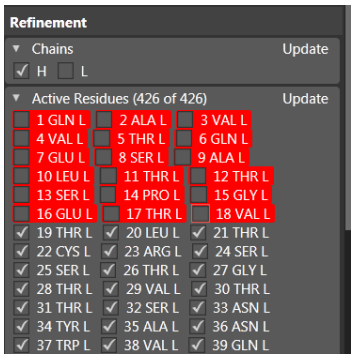

Figure 13: You may exclude part of the structure by unselecting whole chains or specific residues

The sections **Cavity parameters** and **Tunnel parameters** can be used to fine tune the detection algorithm. Details on how these parameters affect the algorithm and the results are given in chapter 4.6. The section called **Selection** displays the residues that you select via the sequence browser or directly inside the molecular viewer, and allows to easily clear the whole selection. Additionally, it allows to recompute tunnels once the settings have been adjusted (see chapter 4.6. for details). The sections **Specific point/Residue(s)** and **CSA Database** contain further controls for tunnel starting points, which will be discussed in detail in chapter 4.6.

The section **Start Points** contains a list of all potential tunnel start points generated by the settings in all the above mentioned sections. This list isn't actually used to refine the calculation, but it is a tool which allows you to closely follow all the changes you achieve during the refinement. For now it is important to note that tunnel **start points** are displayed in the molecular viewer as balls of different colors, depending on their type and whether they are active or not (Figure 14).

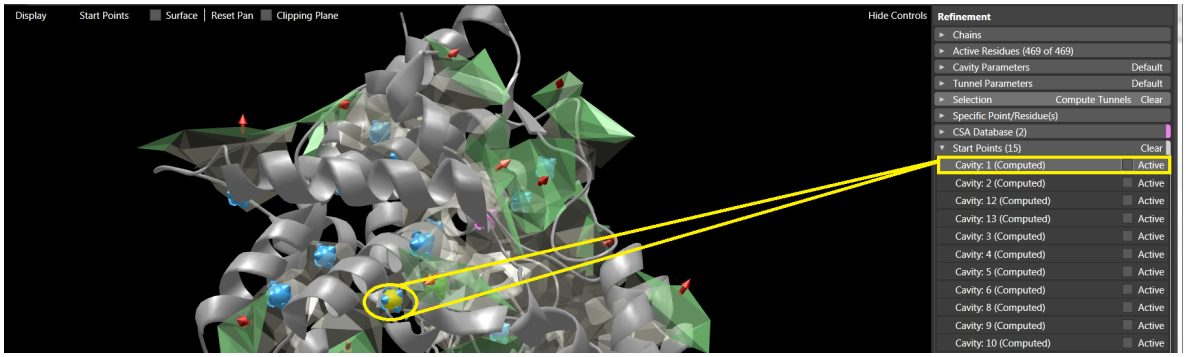

Figure 14: Potential tunnel start points appear as colorful balls in the viewer.

Not all start points constitute the origin of relevant tunnels. If you tick the **Active** box of a start point, you will either notice that one or more tunnels become displayed in the viewer (Figure 15 A), or that an error message ap-

pears in the **Status bar** under the **Sequence browser** (Figure 15 B). You can achieve the same effect by clicking on any start point (colorful ball) directly inside the molecular viewer.

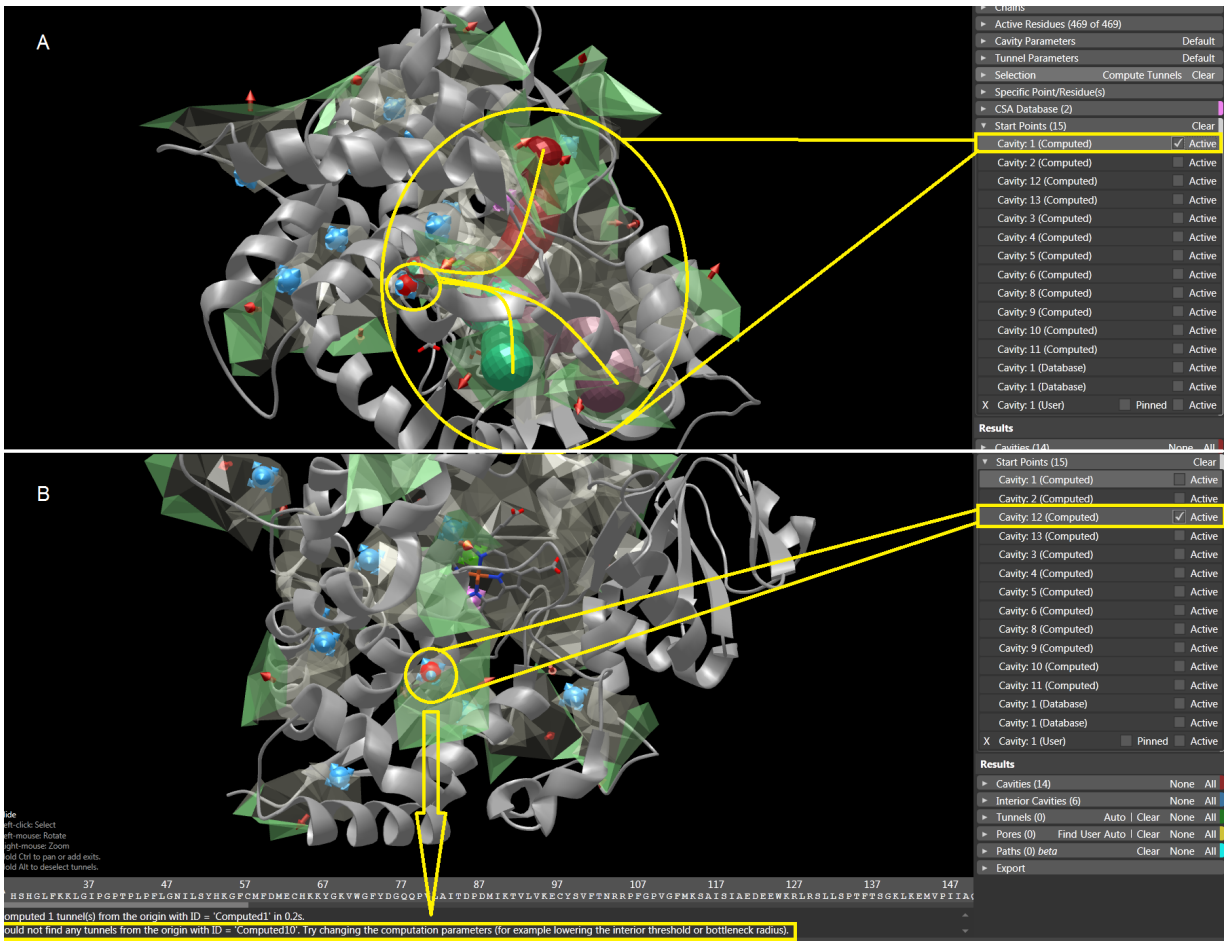

Figure 15: A) - In order to look for any tunnels that start from a specific point, make the point active by either ticking the Active box in the list of Start Points, or clicking directly on the colorful ball in the viewer. B) - If no tunnel is found, an error message is displayed in the Status bar.

Note, however, that sometimes not all start points are visible in the viewer, in which case you need to change the visualization options at the top of the viewer to Start Points → Automatic → All.

4.3.5. Results panel

The lower right part of the workspace (Figure 16) contains all graphical elements that can be displayed by the in-built molecular viewer, such as cavities, tunnels and pores. If you wish to display one such element, simply expand its list and tick the item of interest. The buttons **All** and **None** can be used to easily show or hide all graphical elements of a certain type. The button **Clear** completely removes all graphical elements of a certain type from the results.

Cavities are displayed in the viewer as transparent volumes, unless the **Solid cavities** visualization option is marked at the top of the viewer. Note that by default only the three largest cavities are selected and displayed in the viewer, and if you would like to display other cavities you need to tick the appropriate box on their left. **Interior cavities** are simply cavities which do not communicate with the surface of the protein, and thus Mole 2.0 does not look for tunnels in interior cavities. **Tunnels** and **pores** can be automatically generated using the **Auto** button. Note that tunnels can also be made visible by directly clicking on their start point (colorful ball) inside the molecular viewer, or by ticking the **Active** box of their start point on the **Refinement panel**. Only tunnels starting from active start points will be available in the list of **Tunnels** on the **Results panel**.

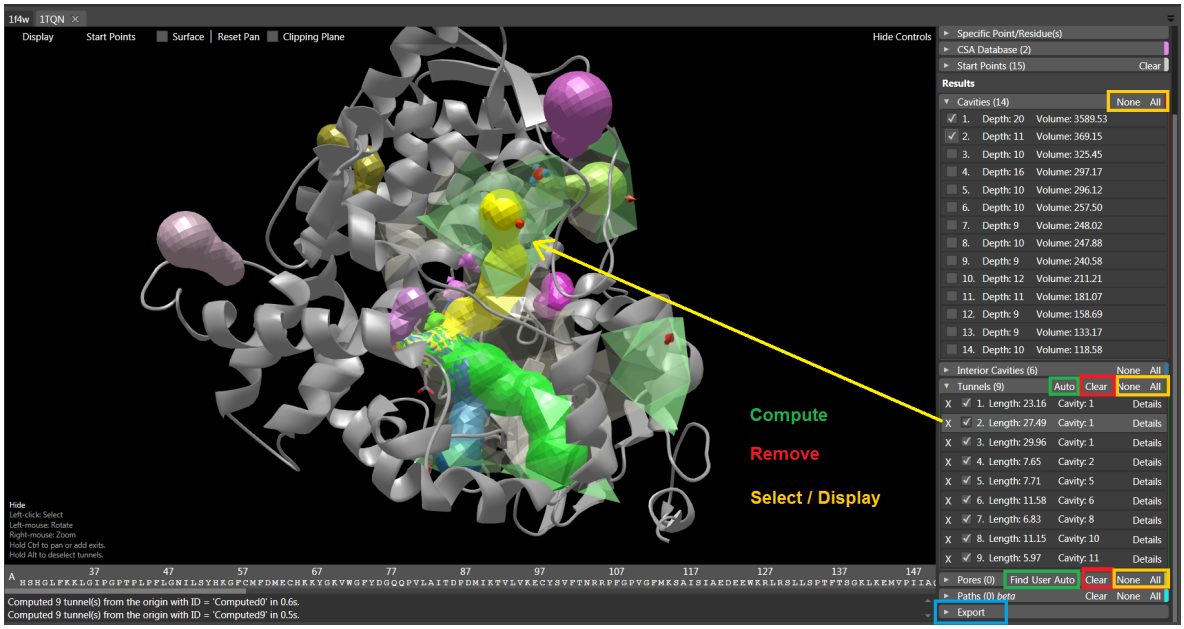

*Figure 16: The **Results panel** contains the elements that can be displayed in the viewer (cavities, tunnels, pores).*

The additional section **Paths** allows for the detection of channels between two given points on the protein surface. Such points can be specified as coordinates in the 3D space, or as the geometrical centre of a group of residues. Nevertheless, the **Paths** section is still in beta state, and should be used with care.

Note that whenever you hover over a certain graphical element in any list in the **Results panel**, the element is displayed in the viewer in yellow. Also note that whenever cavities are connected to the surface of the protein, a green tetrahedron and one or more red arrows are displayed in that area. These additional graphical elements are there to mark exit points, which are important in detecting tunnels and pores. Such elements cannot be operated with in the **Results panel**, but they can be adjusted during the refined detection procedure. More details on exit points can be found in chapter 4.6.

The **Results panel** also contains an explicit **Export** section, which will be discussed in detail in chapter 4.5.

Each graphical item in the **Results panel** contains also the values of properties of interest, such as depth (in Å) and volume (in Å^3) of cavities, and length (in Å) of tunnels and pores. Moreover, clicking the **Details** button opens a new visualization window, where a more detailed analysis can be performed for each tunnel (see chapter 4.4.)

4.3.6. Status bar

The bottom part of the workspace represents the status bar (Figure 17), which gives information regarding the type and duration of all operations performed in Mole 2.0. The **status bar** provides a useful log of all calculations with Mole 2.0, including possible errors. To read through all log messages in the **status bar**, simply use the arrows in the right corner.

*Figure 17: The **Status bar** of the Mole 2.0 GUI provides a log of all calculations performed.*

4.4. Interpreting results

In the case of tunnels and pores, the GUI version of Mole 2.0 also offers a useful way to immediately visualize and export the physico-chemical properties of the identified voids. These properties influence the size and type of ligands that can pass through a tunnel, the direction of travel and how long it takes for the ligand to go through.

Such information becomes available via the **Details** button of each tunnel or pore from the **Results panel**. For each tunnel, you can open a window containing its physical properties (Figure 18). The window consists of two tabs that you can switch between. At the top of the window you can find the **export property** menu. Various kinds of properties (see below) can be copied to clipboard in different formats, and used for further analyses. Such data can be exported also via the **Export** section in the **Result panel** (for details, please see chapter 4.5.).

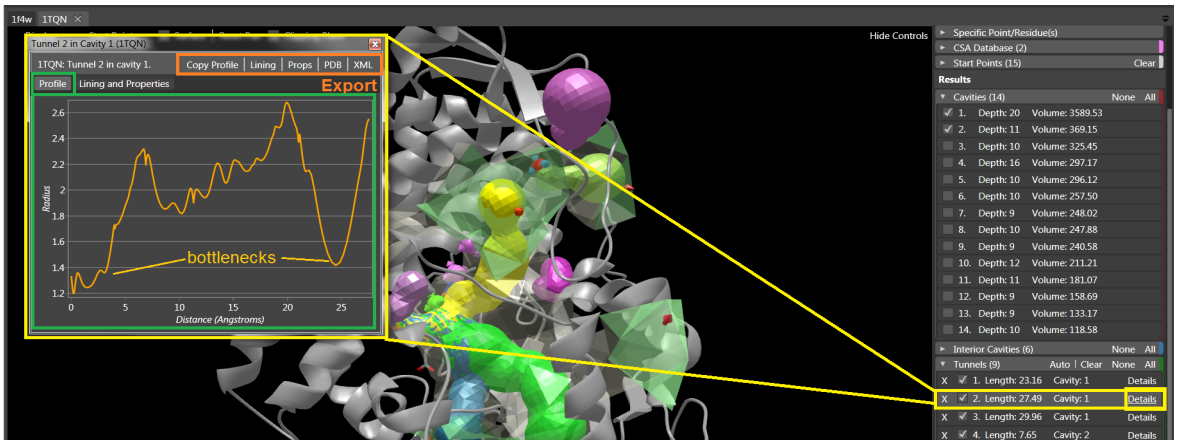

Figure 18: Click on the Details button to open a window which contains the properties of the tunnel, which can be exported in various formats. The Profile tab describes the thickness of the tunnel along its length.

The **Profile** tab contains a plot of the tunnel thickness along the length of the tunnel. The X axis follows the length of the tunnel, while the thickness is evaluated and represented on the Y axis as the **radius** descriptor, which is half of the diameter (thickness) of the tunnel. A narrow part of the tunnel or pore, meaning an area where the radius is small, represents a tunnel **bottleneck**. Both the radius and length are expressed in Å. Such graphs give an idea regarding the landscape of the tunnel or pore, and help in predicting how a ligand may proceed through.

The **Lining and Properties** tab contains a list of residues lining the tunnel surface, together with the physico-chemical properties of the tunnel and its segments. Here we must mention that Mole 2.0 does not define segments as identical units of length, but rather as **layers**. Each layer is defined by the residues lining it. A new layer starts whenever there is a change in the list of residues lining the tunnel along its length.

Residue	Charge	Hydropathy	Hydrophobicity	Polarity	Mutability
ALA	0	1.8	0.02	0	100
ARG	1	-4.5	-0.42	52	83
ASN	0	-3.5	-0.77	3.38	104
ASP	-1	-3.5	-1.04	49.7	86
CYS	0	2.5	0.77	1.48	44
GLU	-1	-3.5	-1.14	49.9	77
GLN	0	-3.5	-1.1	3.53	84
GLY	0	-0.4	-0.8	0	50
HIS	0	-3.2	0.26	51.6	91
ILE	0	4.5	1.81	0.13	103
LEU	0	3.8	1.14	0.13	54
LYS	1	-3.9	-0.41	49.5	72
MET	0	1.9	1	1.43	93
PHE	0	2.8	1.35	0.35	51
PRO	0	-1.6	-0.09	1.58	58
SER	0	-0.8	-0.97	1.67	117
THR	0	-0.7	-0.77	1.66	107
TRP	0	-0.9	1.71	2.1	25
TYR	0	-1.3	1.11	1.61	50
VAL	0	4.2	1.13	0.13	98

Table 1: Physico-chemical properties of single amino acid residues.

Several physico-chemical properties are of interest here. Such properties are defined per amino acid residue, and can be found in Table 1. In the **Lining and Properties** tab, Mole 2.0 reports the physico-chemical properties per group of residues making up layers of the tunnel, or the complete tunnel. An overview of the physico-chemical properties reported by Mole 2.0 is given below. Note that the properties considered here overlap in physical meaning, and are highly correlated (e.g., more polar residues are less hydrophobic). They are nevertheless presented as separate properties for your convenience.

4.4.1. Charge

Mole 2.0 considers that the **charge** of a residue is given by its protonation state. Therefore, each amino acid has a formal charge. The charge of the tunnel represents the algebraic sum between the charges on positively and negatively charged residues which form the surface of the tunnel. However, note that residues are included in this count only when their side chains, and not their backbones, form the surface of the tunnel. Such residues are displayed in the list without a dot next to their one-letter code.

4.4.2. Hydropathy

The **hydropathy** index (**Hdry**) quantifies the hydrophobic or hydrophilic character of a residue's side chain. Larger values of **Hdry** indicate higher hydrophobicity (e.g., 4.5 for isoleucine, 4.2 for valine) , while lower values of **Hdry** indicate higher hydrophylicity (e.g., -4.5 for arginine, -3.9 for lysine). The hydropathy of the tunnel or tunnel layer is calculated as the average of the **Hdry** indices of all lining amino acid residues.

4.4.3. Hydrophobicity

The **hydrophobicity** index (**Hdph**) provides a useful way of evaluating the hydrophobicity of some residues in comparison to others. More positive values of **Hdph** indicate more hydrophobic amino acids (e.g., 1.81 for isoleucine, 1.71 for tryptophan), while more negative values of **Hdph** indicate less hydrophobic amino acids (e.g., -1.14 for glutamic acid, -1.04 for aspartic acid). The hydrophobicity of the tunnel or tunnel layer is calculated as the average of the **Hdph** indices of all lining amino acid residues.

4.4.4. Polarity

The **polarity** index (**Pol**) is a way to quantify the displacement of positive and negative charges inside a residue. Large residues with charged side chains have higher polarity, while small, neutral residues have minimal polarity. The polarity of the tunnel or tunnel layer is calculated as the average of the **Pol** indices of all lining amino acid residues.

4.4.5. Mutability

The **mutability** index (**Mut**) of an amino acid residue relates to the probability of that particular amino acid mutating into another amino acid. High **Mut** values indicate relatively high propensity for mutation, while low **Mut** values indicate lower propensity for mutation. The mutability of the tunnel or tunnel layer is calculated as the average of the **Mut** indices of all lining amino acid residues which contribute to the tunnel surface with their side chains.

The **Lining and Properties** tab is organized in a table. Please note that when the physico-chemical properties window first opens, it is not large enough to display all the data it contains in the **Lining and Properties** tab. You therefore have to enlarge the window in order to see all the data in the table (Figure 19).

At the top of the table, the values of the properties are given as calculated for the whole tunnel. The total charge, along with the average hydropathy, hydrophobicity, polarity and mutability are given. You can also find here the values computed per layer, and then averaged over the whole tunnel while weighting according to the length of each layer.

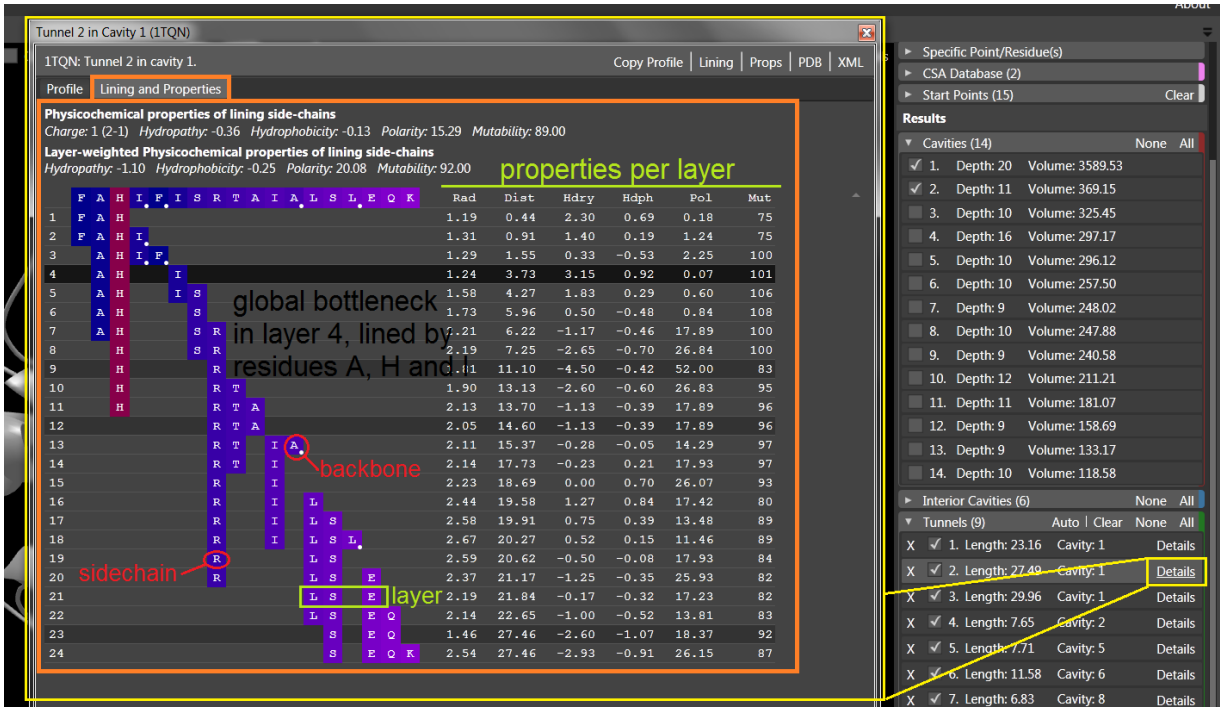

Figure 19: The Lining and Properties tab of the properties window needs to be enlarged to be able to display all information. Each line in the table contains the description of a layer in the tunnel.

Further, the lining residues of the tunnel are given as a list of their one-letter codes, in the order in which they contribute to the tunnel surface, starting from the bottom of the cavity (tunnel starting point), and expanding towards the surface of the protein (tunnel exit point). The tunnel is characterized layer by layer. Remember that a tunnel layer is defined by the residues lining it. Therefore, the order of residues in this list is not given by the amino acid sequence directly, but by the overall three-dimensional arrangement of the entire protein structure, and how the residues come together from distant parts of the sequence to form the surface of the tunnel. Hover over any of the residues to get complete information regarding its position in the sequence. The residues which contribute to the surface of the tunnel by their backbone part are marked with a dot next to their one-letter code. Note that if a residue contributes with both its side chain and its backbone, it appears twice in the list of residues.

Each line in the table describes a layer of the tunnel, and it contains the list of residues contributing to its surface, along with a few descriptors. The first descriptor is **Rad**, the minimum tunnel radius along that particular layer. Remember that the thickness of the tunnel is twice the value of **Rad**. The second descriptor is **Dist**, the distance from the bottom of the cavity (tunnel starting point) to the farthest atom in that particular layer. The third descriptor is **Hdry**, the average hydropathy index of the group of residues making up the surface of the tunnel layer. The fourth descriptor is **Hdph**, the average hydrophobicity index of the group of residues making up the surface of the layer. The fifth descriptor is **Pol**, the average polarity index of the group of residues making up the surface of the layer. The last index is **Mut**, the average mutability index of the group of residues making up the surface of the layer. Note also that the shade of the background of each line in the table indicates whether there is a bottleneck in the tunnel in that particular layer. Darker shades indicate local bottlenecks, while black indicates the presence of a global bottleneck.

It is important to note that such results represent static pictures of the identified cavities, tunnels and pores. During protein dynamics, the shape profile might be modified, the side chain of some amino acid residues might change position, or a tunnel leading out from a cavity might open or close completely. Depending on the case, such changes might be associated with the protein performing its biological function, or on the contrary, might indicate a pathological mutation or even improper conditions of the simulation. Additionally, remember that all

quantitative evaluations are fairly approximate, since they are based on experimental or modeled molecular structures of varying quality.

4.5. Exporting results

To export your results, expand the **Export** section in the **Results panel** (Figure 20), and click on the appropriate button. The graphical representation of tunnels, pores and paths, expressed as sets of co-ordinates in the three-dimensional space, can be exported in PDB format. Mole 2.0 will produce a PDB file with a content similar to:

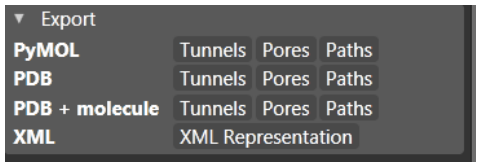

Figure 20: You may export your results in the Export section of the Results panel.

```
REMARK ATOM  NAM RES   TUNID     X       Y       Z    Distnm RadiusA
HETATM    1  X   TUN H   1     -17.113 -23.724  -5.514  0.00  1.33
HETATM    2  X   TUN H   1     -17.119 -23.746  -5.473  0.05  1.28
HETATM    3  X   TUN H   1     -17.127 -23.764  -5.433  0.09  1.25
                                          ...
HETATM  706  X   TUN H   4     -24.207 -27.310 -32.884  7.55  3.02
HETATM  707  X   TUN H   4     -24.227 -27.318 -32.847  7.59  3.02
HETATM  708  X   TUN H   4     -24.251 -27.329 -32.799  7.64  3.02
```

Moreover, Mole 2.0 allows you to generate PyMOL scripts for comfortable visualization and production of high quality pictures using PyMOL. Mole 2.0 will produce a .py file with a content similar to:

```
def Tunnels1():
   model = Indexed()
   at = Atom()
   at.name = "0"
   at.vdw = 1.3251938904398
   at.coord = [-17.113096738422, -23.7241725168357, -5.51369483241423]
   model.atom.append(at)
                                          ...
   for a in range(len(model.atom)-1):
      bd = Bond()
      bd.index = [a,a+1]
      model.bond.append(bd)
   cmd.set("surface_mode",1)
   cmd.set("mesh_mode",1)
   cmd.load_model(model,"Tunnels1")
   cmd.hide("everything","Tunnels1")
   cmd.set("surface_color","pink","Tunnels1")
   cmd.show("surface","Tunnels1")
Tunnels1()
                                          ...
def PdbLoadCommand():
   pdbCode = '1TQN'
   pdbCode = pdbCode.upper()
   cmd.fetch(pdbCode)
PdbLoadCommand()
```

The properties of the identified tunnels, as discussed in chapter 4.4. , can be exported in XML format. Mole 2.0 will produce an XML file with a content similar to:

```
<?xml version="1.0" encoding="UTF-8"?>
-<Tunnels>
     -<Tunnel Cavity="1" Auto="1" Id="1">
     <Properties Mutability="84" Polarity="13.99" Hydropathy="-0.22" Hydrophobicity="-0.07" Num-
Negatives="1" NumPositives="2" Charge="1"/>
               -<Profile>
                        <Node   Z="8.660635"   Y="-0.096446"   X="2.175627"   Distance="0.000000"
T="0.000000" Radius="1.325194"/>
                        <Node   Z="8.701494"   Y="-0.118397"   X="2.169900"   Distance="0.046734"
T="0.005405" Radius="1.284251"/>
                        <Node   Z="8.741024"   Y="-0.136001"   X="2.161753"   Distance="0.090767"
T="0.010811" Radius="1.247336"/>
                                          ...
               </Profile>
               -<Layers>
```

```
                        <ResidueFlow>302 PHE A,305 ALA A,508 HEM A,301 ILE A Backbone,302 PHE A
Backbone,301 ILE A,119 SER A,212 ARG A,105 ARG A,108 PHE A,119 SER A Backbone,106 ARG A
Backbone,120 ILE A Backbone,120 ILE A,107 PRO A Backbone,122 GLU A,107 PRO A,111 VAL A</Residue-
Flow>
                        <LayerWeightedProperties Mutability="87" Polarity="15.51" Hydropathy="-
0.42" Hydrophobicity="-0.08"/>
                        -<Layer LocalMinimum="0" EndDistance="0.22977" StartDistance="0.00000"
MinRadius="1.22635">
                                <Residues>302 PHE A,305 ALA A,508 HEM A</Residues>
<FlowIndices>0,1,2</FlowIndices>
                                <Properties Mutability="75" Polarity="0.18" Hydropathy="2.30"
Hydrophobicity="0.69" NumNegatives="0" NumPositives="0" Charge="0"/>
                        </Layer>
                                                ...
                </Layers>
        </Tunnel>
</Tunnels>
```

Another way to export the properties of each tunnel is to use the **export property** menu at the top of the window containing the physico-chemical properties of the respective tunnel. Simply click the **Details** button on the right side of the tunnel of interest in the **Results panel**, and then click the appropriate format at the top right corner of the newly opened window (Figure 18). The results will be exported to clipboard in your chosen format, and then you may paste them in whichever type of file you prefer.

4.6. Refined detection of cavities, tunnels and pores

In most cases, Mole 2.0 provides relevant results in a fully automated manner, without user intervention or the need to understand the algorithms involved in computing empty spaces in biomolecules. Nevertheless, having even a general impression of the basic steps can help in tailoring the functionality of the program to specific cases, and thus enhancing the research experience. Therefore, we first give a brief overview of the procedure of computing empty spaces in proteins, as well as the definitions of the most important concepts. We then explain how to use these concepts to refine your calculation, along with examples to show how proper tweaking of Mole 2.0 parameters can improve your results. Remember that a great advantage of Mole 2.0 is the fact that it allows you to interactively tweak the tunnel detection algorithm, such that the results are immediately available for inspection and comparison.

4.6.1. Overview of algorithm and concepts

As soon as the structure of a biomolecule is loaded, its atomic structure is represented by Van der Waals spheres centered on the atoms. Further, the atomic structure is translated into a molecular graph representation, which allows for the computation of the molecular surface and its features in a straightforward manner (Figure 21 i,ii). The molecular surface is approximated as the surface accessible to a sphere, the size of which is a parameter (**Probe Radius**) which can be adjusted in order to control the level of detail provided by the molecular surface. Subsequently, it is possible to identify cavities on the molecular surface (Figure 21 iii). A particular concave feature of the molecular surface can be considered a cavity if its width is larger than some minimum value (**Interior Radius**). As a consequence, this parameter basically controls the definition of cavities.

Further, Mole 2.0 looks for tunnels which lead from the identified cavities to the surface of the protein. The calculation of tunnels as paths between two points (e.g., between the bottom of a cavity to protein surface) requires that the start and end points be defined prior to the calculation (Figure 21 iv,v). Tunnel start points can be assigned automatically at the bottom of each identified cavity, or they can be user-defined. The user may specify a point in the three-dimensional space, or group of amino acid residues. Tunnel start points will be tested within a certain distance around this point. This distance is controlled by an adjustable parameter (**Origin Radius**), allowing an effective exploration of an area of interest. Tunnel exit points can be automatically assigned at the outer boundary of each cavity, or can be user-defined. Many exit points are generated and tested at each cavity boundary. The density of exit points that will be tested at cavity boundaries can be controlled by an adjustable parameter (**Surface Cover Radius**). Note that even if the tunnel start and end points are user defined, Mole 2.0 will

find tunnels only if the tunnel start point lies inside some cavity, and the exit point lies at the boundary of some cavity.

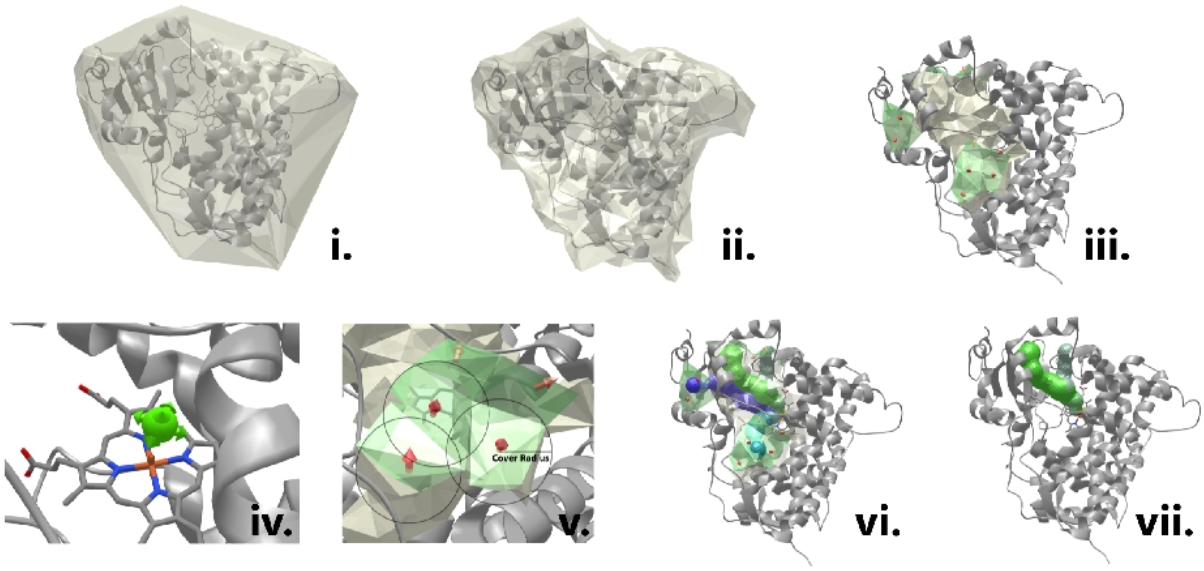

Figure 21: Overview of tunnel calculation procedure: i, ii) building the molecular surface; iii) identifying cavities; iv) identifying potential tunnel start points; v) identifying potential tunnel exit points; vi) computing tunnels; vii) filtering tunnels.

Once the tunnel start and end points have been identified, the tunnel itself is computed as the shortest distance between the two points on the molecular graph representation (Figure 21 vi). The visual representation of the tunnel is a surface volume between the start and end point. The thickness of the tunnel at each point corresponds to the empty space between the surrounding Van der Waals spheres which represent the atoms of the amino acid residues lining the tunnel.

Many tunnels are generated by the above described procedure, but not all might be relevant. It is possible to filter out tunnels that are not of interest by setting a few parameters (Figure 21 vii). In order to ensure that the reported tunnel is wide enough for ligands of interest to pass through, the minimum requirements for the bottlenecks can be set by specific parameters (**Bottleneck Radius** , **Bottleneck Length**). If two identified tunnels follow the same channel leading out of a cavity in too similar a manner, only the shorter tunnel will be reported. The critical level of similarity can also be adjusted by a parameter (**Cutoff Ratio**).

It is also possible to identify pores in the protein structure, and in principle it is done by finding connections between two tunnel exit points. Depending on the selected technique (see below), you may decide exit points will be tested when looking for pores. The restrictions applied on the exit points are the same as for tunnels. It is possible to filter out uninteresting pores by adjusting the same parameters as for tunnels.

4.6.2. Refinement techniques

The tools needed in this step can be found in the **Refinement Panel** at the top right corner of the workspace (Figure 12). It is important to remember that Mole 2.0 first identifies cavities, and only afterwards looks for tunnels and pores. Therefore any operation which affects the way cavities are found and characterized will reflect upon the subsequent tunnels and pores.

In the sections **Chains** and **Active residues** you can restrict the areas of the protein where Mole 2.0 will search for cavities (see also chapter 4.3.4. and Figure 13). After you have made your changes, click **Update**, and then recalculate all tunnels and pores by using the corresponding **Auto** buttons in the **Results panel**.

Adjusting the **Probe Radius** parameter in the **Cavity Parameters** section allows to refine the level of detail of the molecular surface (Figure 22). A lower value of **Probe Radius** will produce a rough surface, following closely on the Van der Waals contour. A higher value of **Probe Radius** will produce a smooth surface, with little detail. If you use a higher **Probe Radius** you may find that some tunnels prolong artificially outside the van der Waals area. Therefore, the default value is usually suitable for smaller, more compact structures, while a higher value might be useful in the case of large structures or when looking for tunnels with broad profiles.

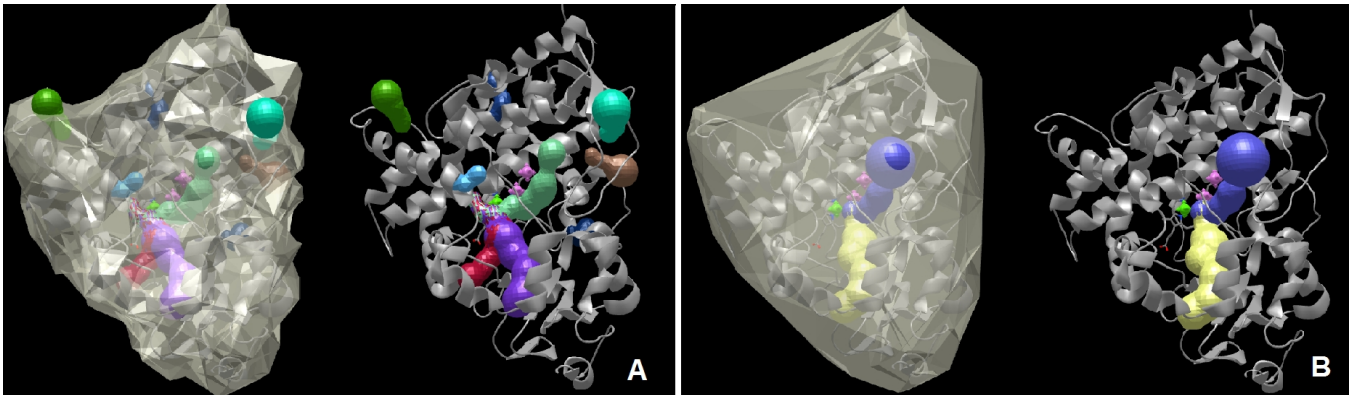

Figure 22: The parameter Probe radius controls the level of detail of the surface. A - Probe radius 3, B - Probe radius 12.

Adjusting the **Interior Threshold** parameter in the **Cavity Parameter** section allows to identify all voids wider than double the **Interior Threshold** (Figure 23). A small **Interior Threshold** will therefore result in the identification of more cavities, and probably also more tunnels. Sometimes it is useful to see even tunnels which seem too narrow for the ligand of interest, if you suspect that the protein moves significantly before or during ligand binding. On the other hand, if the protein contains wide voids, using a larger **Interior Threshold** will make it easier to see only those tunnels which are biologically relevant.

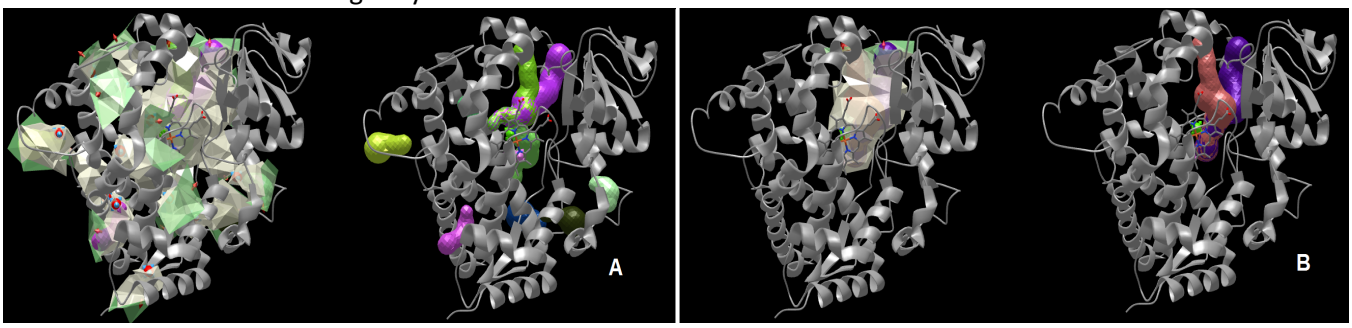

Figure 23: The parameter Interior threshold controls the minimum definition of cavities. A - Interior threshold 1.25, B - Interior threshold 1.90.

Remember, after you have made your changes to the **Cavity parameters**, you need to tell Mole 2.0 to generate and display the new tunnels and pores by using the corresponding **Auto** buttons in the **Results panel**.

Further, you may filter out tunnels which you presume will not be biologically relevant, or which you are not interested in at that time. Tunnel filtering can be achieved by adjusting parameters which directly define the tunnel, and which can be found in the **Tunnel parameters** section of the **Refinement panel**. Note that making changes to the Tunnel parameters does not require you to recompute the the tunnels by using the **Auto** button. The reason is that **Tunnel parameters** affect simply which tunnels will be reported to you, and not which tunnels Mole 2.0 is able to find.

We have already mentioned that Mole 2.0 calculates each tunnel as the path between the tunnel start and exit point, and thus imposing restrictions on these points is an easy way to filter out tunnels. There are several possibilities for identifying tunnel start points (Figure 24).

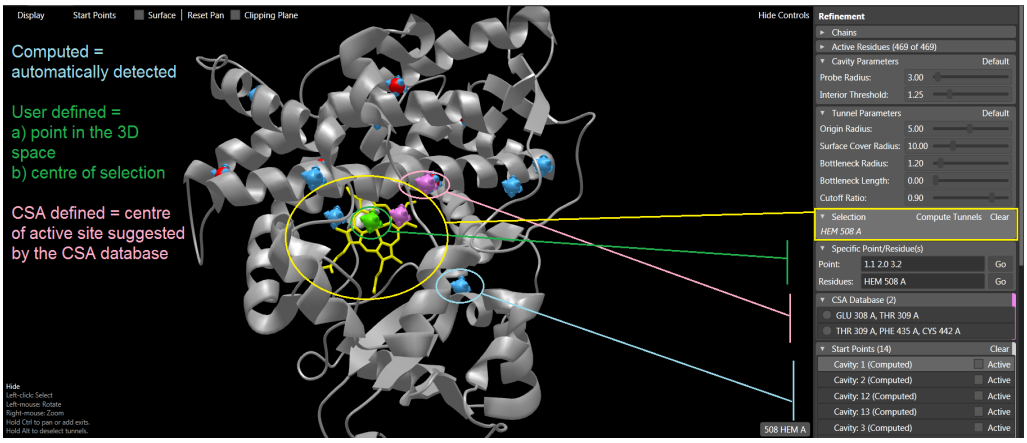

Figure 24: Potential tunnel start points can be automatically generated, user defined or inferred from documented active sites.

a) Mole 2.0 can automatically detect possible tunnel start points at the bottom of all identified cavities. These automatically detected start points appear as small blue spheres in the molecular viewer, though they are only visible if their parent cavity is active (i.e., if there is a tick in the **Active** box of that cavity in the list of cavities on the **Results panel**). Automatically detected start points can be found in the list of **Start Points** on the **Refinement panel**, and are marked by the label **Computed**.

b) Tunnel start points can also be completely user defined. In the **Specific Point/Residue(s)** section, you may specify a point in the three-dimensional space by its Cartesian coordinates. Alternatively, in the same section you may specify a group of residues, the center of which will be used as a potential tunnel starting point. Once you have specified some point (by coordinates or group of residues), click the corresponding **Go** button in order to generate a user-defined start point. If this point lies inside some cavity, a green ball will appear in the viewer, and a new item will be added to the list of **Start Points** on the **Refinement panel**, marked by the label **User**. Note also that the **Selection** section is cleared and updated with the specified residues. A user start point can be generated also by simply clicking on residues directly inside the viewer, but first you will have to clear the selection manually by using the **Clear** button at the right of the **Selection** section on the **Refinement panel**. Next, click the **Compute tunnels** button in the **Selection** section, or tick the **Active** box of the newly added point in the **Start Points** list, or click directly on the green ball in the molecular viewer. If any tunnels with that start point are found, they will be displayed in the viewer as colorful volumes or traces (depending on your display options), and added at the top of the **Tunnels** list in the **Results panel**. If no tunnel is found, an error message will appear in the **Status bar**. Do not worry about the accuracy of your specification of the tunnel start point. By default, Mole 2.0 will test several start points within a certain radius of your initial specification. If you would like to test a wider area of space around your specified point, simply increase the value of the parameter **Origin Radius** in the **Tunnel Parameters** section of the **Refinement panel**. If no points within **Origin Radius** of your initial specification lie inside a cavity, Mole 2.0 will not display any green ball in the viewer, and will not search for tunnels with such start points. If you would like to remove all user-defined start points, use the **Clear** button in the **Start Points** section of the **Refinement panel**. If you want to remove only some (not all) user-defined start points, first tick the **Pinned** box of the points you wish to keep, and then hit the **Clear** button.

c) Additionally, Mole 2.0 can retrieve information from the Catalytic Site Atlas (CSA), a database of known catalytic sites. If you have a working Internet connection and you load your protein with its PDB ID, Mole 2.0 will look up the CSA database for any entries related to this PDB ID. If a catalytic site is found in CSA, the amino acid residues which make up this catalytic site will be listed as an item in the **CSA database** section of the **Refinement panel**, and the geometrical centre of the catalytic site will appear in the molecular viewer as a pink ball. Moreover, an additional item will appear at the bottom of the **Start Points** list on the **Refinement panel**, marked by the label

Database. You may use this point as a potential tunnel start point. Simply click on the pink ball in the viewer, or tick the corresponding **Active** box in the **Start Points** list. One or more tunnels may be generated, or you may receive an error message in the **Status bar**, depending on whether the chosen point is a relevant tunnel start point. Alternatively, you may click on the definition of this point in the **CSA database** section of the **Refinement panel**. This will cause the corresponding amino acid residues to be selected, and the pink ball will to be displayed in green, as any other user-defined start point. Then click **Compute tunnels**, or click on the ball to see if any tunnel is found at that position.

You may also wish to test as many tunnel exit points as possible. In order to do so, decrease the value of the **Surface Cover Radius** parameter in the **Tunnel Parameters** section on the **Refinement panel**. Nonetheless, remember that tunnel exit points are displayed in the molecular viewer as red arrows. So, once you have identified the optimal values for the other parameters, it is better to revert to a smaller **Surface Cover Radius** for better visualization.

The **Tunnel Parameters** section provides the opportunity to filter out tunnels based not only on their start and end points, but also on their properties (Figure 25). The parameter **Bottleneck Radius** defines the minimum radius of the tunnel at any point along its length. Setting **Bottleneck Radius** a lower value will return more tunnels, and its effect is somewhat similar to that of setting **Interior Threshold** in the **Cavity Parameter** section. Tunnels which are narrower than twice the **Bottleneck Radius** at any point will not be reported if the parameter **Bottleneck Length** is set to 0. Nonetheless, if you would like to allow for some tolerance, you may ask Mole 2.0 to report also tunnels which are narrower than twice the **Bottleneck Radius** over a length of maximum **Bottleneck Length**. Additionally, it is possible to minimize redundancy by filtering out tunnels that are too similar. If two tunnels are too similar over a proportion of their length higher than **Cutoff Ratio**, only the shorter tunnel will be reported. If **Cutoff Ratio** is set closer to 1, the similarity criterion is weak, and more tunnels will be reported. If **Cutoff Ratio** is set closer to 0.5, the similarity criterion is strong, and fewer tunnels will be reported.

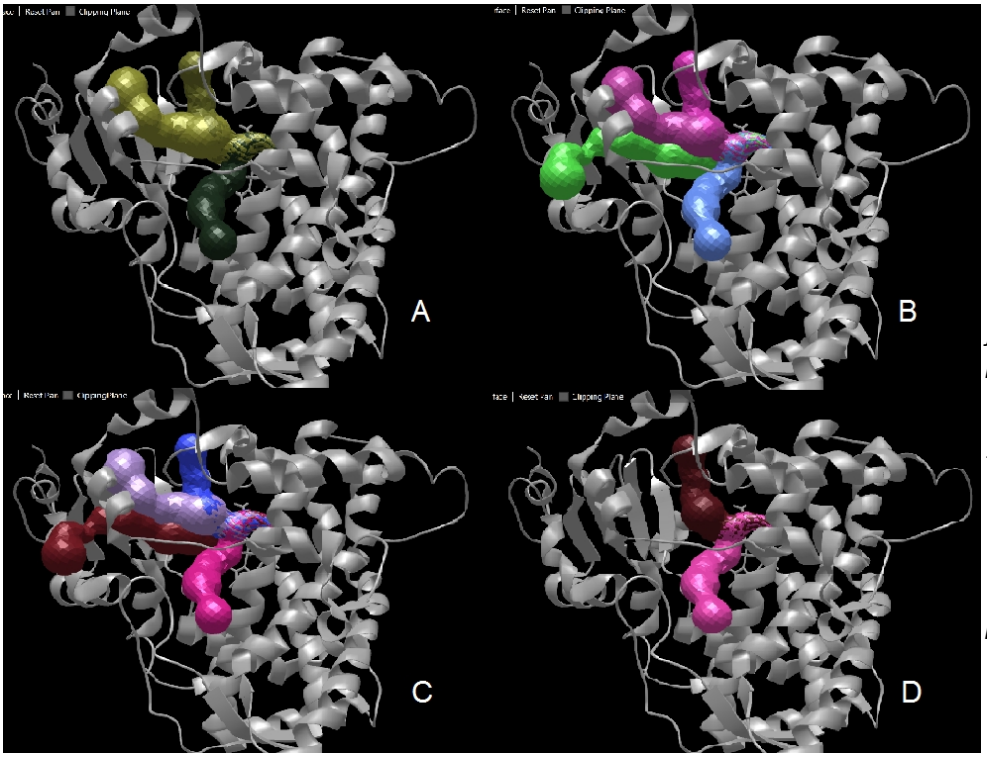

Figure 25: By tuning the Tunnel parameters on the Refinement panel, you can apply filters on which tunnels will be reported. A) Default parameters allow you to see 3 tunnels leading out of a given caivty. B) By decreasing the Bottleneck radius parameter, a fourth tunnel becomes visible, but note that this tunnel has a narrow segment towards its exit. C) The fourth tunnel can be made visible also by increasing the tollerance parameter Bottleneck length. D) Tightening the similarity criterion Cutoff ratio from 0.9 to 0.5 removes two of the tunnels on the grounds that they share at least 50% of their length with one of the remaining tunnels.

Pores can be computed and displayed only after the tunnels have been identified. The reason is that pores are defined as channels which connect two tunnel exit points. All parameters which affect tunnels thus affect pores in the same way (Figure 26). Moreover, special importance is given here to exit points. We have already seen that exit points can be defined only at the outer boundaries of cavities. Therefore, parameters which affect the molecular surface (**Chains**, **Active residues**, **Probe radius**), the definition of cavities (**Interior threshold**) or the density of possible exit points at the cavity boundary (**Surface Cover Radius**) will influence the detection of pore exit points. Additionally, Mole 2.0 provides several algorithms which allow a more specific selection of exit points to be tested (Figure 16). The **Auto** button computes the pores as channels between all pairs of potential exit points automatically generated at all cavity boundaries. Note that not all potential exit points (i.e., red arrows) are the exit points of actual tunnels. The **Find** button limits the search to exit points of already identified tunnels, and it is by this algorithm that you are likely to obtain relevant results. The **User** algorithm allows you to define exit points by CTRL+click on the molecular surface in the viewer. Make sure to first display the molecular surface by ticking the **Surface** box at the top of the viewer. It is possible to filter out uninteresting pores by adjusting the same parameters as for tunnels.

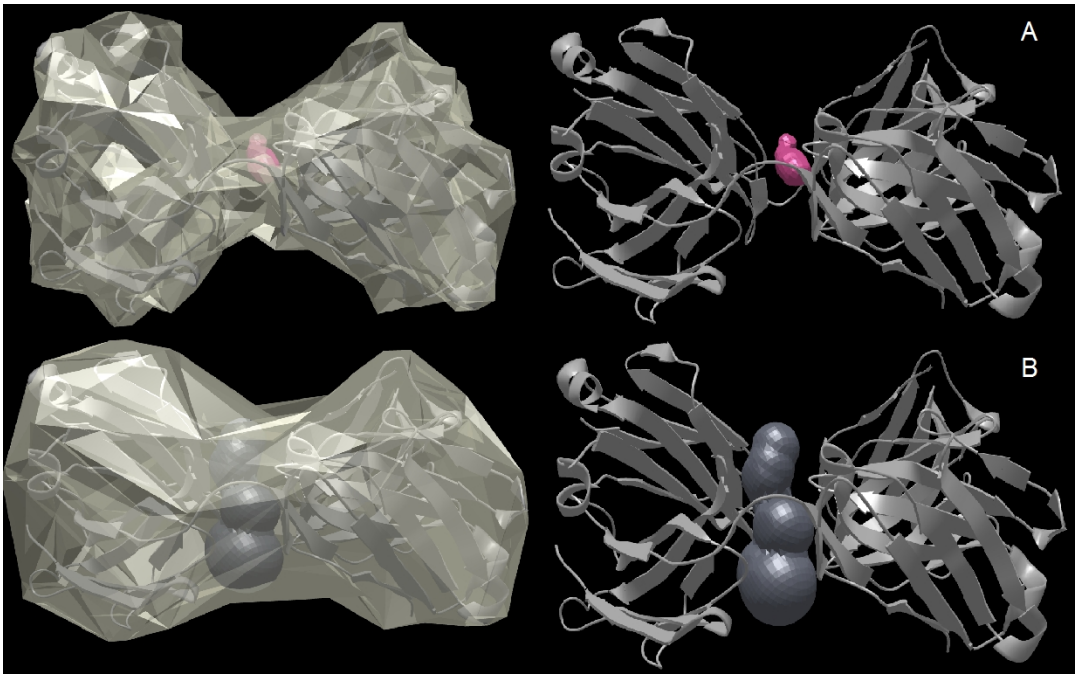

Figure 26: Example of optimal pore computation with the Mole 2.0 GUI: A) Using default parameters, the location of the pore is found, but the profile of the pore is not realistic. B) Increasing the parameter Probe radius from 3 to 9 allows for a smoother and bulkier molecular surface, which leads to a proper identification and description of the pore.

Part II

Mole 2.0, a commandline-based application

5. Technical details

5.1. Availability

The command line based version of Mole 2.0 is available free of charge on our website at http://web-chem.ncbr.muni.cz/Platform/App/Mole#.

5.2. System requirements

The command-line version of Mole 2.0 can be run on Windows, Mac OS or Linux. Windows users will need the .NET framework 4.0 or above, meaning that Mole 2.0 will run on any Windows operating system from Windows XP (Service Pack 3) and higher or Windows Server 2003 (Service Pack 2) and higher. Linux and Mac OS users need to install the Mono running environment (http://www.mono-project.com), version X or higher.

5.3. Installation

It is not necessary to install Mole 2.0, as executable files are provided upon download. Simply download the .zip archive from our website (Figure 27), extract to a preferred location on your disk, and you can immediately start using Mole 2.0 by running the executable file mole2.exe. This executable file takes as input a single XML file which includes all information about the molecular structures of interest, the parameters of the calculation, and export options for the results.

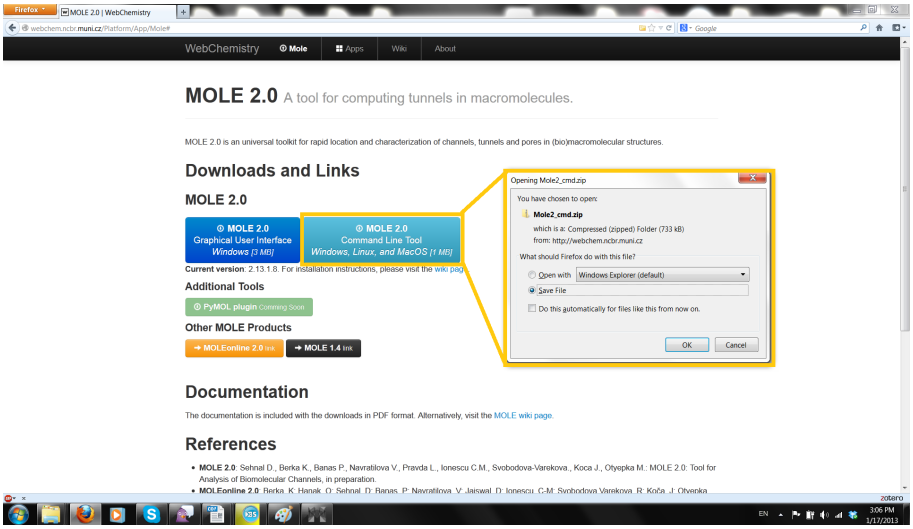

Figure 27: Download the command line version of Mole 2.0 from our web page

In order to run the command line version of Mole 2.0 on Windows, you need to open a command line prompt, then type inside the path to the executable, and the path to the xml file as below:

```
path/to/mole2.exe path/to/xml.xml
```

For example, if you have already navigated to the folder where you extracted all necessary files of Mole 2.0, and you have copied in this folder also the xml input file called input.xml, as well as the molecule you would like to work with, you need simply run:

```
mole2.exe input.xml
```

On any non-Windows operating systems with the mono environment installed, you just need to open a terminal and run:

```
mono path/to/mole2.exe path/to/xml.xml
```

If you get an error, please double check section 5.2. . If your system meets all requirements and you still get an error, please contact us via our website at http://webchem.ncbr.muni.cz/Platform/App/Mole#.

6. Functionality

Compared to the GUI based version, in the command line version of Mole 2.0 cavities and tunnels are computed the same way, and the same properties can be exported. However, unlike the GUI application, the command line version of Mole 2.0 does not provide immediate visualization of the identified cavities, tunnels and pores, but it does allow for subsequent visualization in PyMOL via automatically generated scripts. Moreover, in combination with our PyMOL plug-in, the command line version of Mole 2.0 provides interactive functionality comparable to the GUI version under Windows, Mac OS and Linux. The command line version of Mole 2.0 is ideal for processing a large number of molecular structures at the same time, as the call to Mole 2.0 can be easily be included in scripts.

The command-line version of Mole 2.0 employs the same basic parameters as the GUI version. Therefore please refer to Part I of this manual for details regarding the interpretation of results (chapter 4.4.), and basic algorithms and concepts (chapter 4.6.).

It is important that you read through Part I of this manual even if you plan to use only the command line version of Mole 2.0!

There are a few additional parameters involved in operating with the command line version of Mole 2.0. Nonetheless, tunnel calculation can still proceed in an automatic or refined fashion, according to your preference.

The input file contains XML tags, which function as categories or settings for your calculation using Mole 2.0. Depending on how you set the tags, you may run the automatic or refined tunnel detection algorithms, and you may decide about which data is exported and how. You do not need to adjust or even understand all the parameters included in the XML input file. The most important parameters are discussed in the subsections below, while the rest are briefly described in the Annex of this manual.

6.1. Loading structures

The **Input** tag allows to specify the path to an input PDB file. If your structure of interest is in the file 1HTQ.pdb, and this file is at the same place as the location from where you are running the Mole 2.0, your **Input** tag should look like:

```
<Input>1HTQ.pdb</Input>
```

6.2. Automatic detection of cavities, tunnels and pores

If you would like to run the automatic tunnel detection algorithm, just use the file **testinput_auto.xml**, provided together with the Mole 2.0 executables, in which you simply adjust the **Input** tag accordingly.

```
<?xml version="1.0" encoding="UTF-8"?>
-<Tunnels>
     <Input>1HTQ.pdb</Input>
     <WorkingDirectory>./auto_output/</WorkingDirectory>
     <Params OriginRadius="5" SurfaceCoverRadius="10" InteriorThreshold="1.4" ProbeRadi-
us="3"/>
     <Export PDB="1" PyMol="1" MeshDensity="1.33" Cavities="1" MeshGz="1" Mesh="1"/>
     <Origin Auto="1"/>
</Tunnels>
```

6.3. Exporting results

The data and format that will be exported is controlled at the level of the **WorkingDirectory** and **Export** tags. The **WorkingDirectory** tag should contain the path to the folder (directory) where the output data are going to be stored after the calculation is done. If the folder does not exist, Mole 2.0 will create such a folder. All properties of all tunnels are exported by default, and the functionality is equivalent to that described in chapter 4.5. of Part I in this manual. In order to export the PDB representation of the identified tunnels, make sure that the attribute **PDB** of the **Export** tag has the value 1. Otherwise, set it to 0. To export also the cavities, set the **Cavities** attribute of the **Export** tag to 1 too.

6.4. Interpreting results

Since immediate visualization cannot be achieved directly by the command line application, you may wish to visualize the cavities and tunnels using PyMOL once your Mole 2.0 calculation is done. In this case, check that the attribute **PyMol** in the **Export** tag of the XML file be set to 1. In this case, Mole 2.0 will generate a PyMOL script which you can then run in PyMOL (for more details, please see chapters 6.6. and 6.7. below). All exported properties can be analyzed as described chapter 4.4. of Part I in this manual.

6.5. Refined detection of cavities, tunnels and pores

As previously mentioned, adjusting the way cavities are detected directly affects the tunnels that Mole 2.0 will be able to identify in a structure.

Please read through chapter 4.6. of Part I in this manual before proceeding.

It is possible to restrict the area of the protein where the cavity search will be performed by employing the **SpecificChains** attribute in the **Input** tag. For example, if you would like to include only chains A, B and C of the structure from the file 1JJ2.pdb, then your **Input** tag looks like:

```
<Input SpecificChains="ABC">1JJ2.pdb</Input>
```

Note that the attribute **SpecificChains** is not mandatory.

The additional tag **NonActiveResidues** allows for further specification regarding which areas of the structure will be excluded from the calculation. The space previously occupied by the excluded residues will be considered as empty, which may be useful, for instance, in studying the protein even if the ligand is present in the cavity or tunnel.

```
<NonActiveResidues>
  <Residue Name="HEM" Chain="A" SequenceNumber="508" />
</NonActiveResidues>
```

Remember that the molecular structure is represented at some point by the van der Waals radii of the atoms. Moreover, the tunnel size is computed according to the distances between van der Waals spheres. The command line version of Mole 2.0 allows you to customize the van der Waals radii of selected elements via the **CustomVdW** tag. In this case, you must provide the van der Waals radii of all elements in the following format:

```
-<CustomVdw>
   <Radius Value="1" Element="H"/>
 </CustomVdw>
```

The **Params** tag provides ample regulatory control over the calculation. This tag is equivalent to the **Refinement panel** in the GUI version of Mole 2.0 (see the chapter 4.3.4. and 4.6. from Part I in this manual).

As in the GUI version of Mole 2.0, the command line application allows user defined tunnel start points within the **Origin** tag. If the attribute **Auto** is set to 1, then Mole 2.0 automatically detects tunnel start points at the bottom

of the cavities. In order to define a start point, first set the attribute **Auto** to 0. Then, identify the start point as a **residue** or group of residues:

```
<Origin Auto="0">
   <Residue Chain="A" SequenceNumber="308" />
   <Residue Chain="A" SequenceNumber="309" />
</Origin>
```

or as a **point** in the three-dimensional space, defined by its Cartesian coordinates:

```
<Origin Auto="0">
    <Point X="1.01" Y="-2.35" Z="15.4" />
</Origin>
```

Remember that all user defined starting points will be optimized according to the **Origin Radius** parameter. If you do not want such an optimization to take place, you can use the **ExactPoint** tag:

```
<Origin Auto="0">
    <ExactPoint X="1.01" Y="-2.35" Z="15.4" />
</Origin>
```

Multiple starting points can be delivered to Mole 2.0 in a single calculation. In this case, the **Origin** tag will contain a separate **Pinned** tag for each user defined start point:

```
<Origin Auto="0">
  <Pinned><Residue Chain="X" SequenceNumber="123" /></Pinned>
  <Pinned><Point X="1" Y="2" Z="3" /></Pinned>
</Origin>
```

6.6. Visualization in PyMOL

While the command line version of Mole 2.0 does not allow for immediate visual inspection of the results, it provides you the possibility to do so using PyMOL, a free and versatile molecular visualization package. Once you have PyMOL installed on your computer, start it. Then go to File → Run, and navigate to the directory where your Mole 2.0 calculation has produced output. This is the folder that you have set via the tag **WorkingDirectory** in your input XML file. If you had set the **PyMol** attribute in the **Export** tag to 1, you should find a file called **tunnels.py**. Load this file into PyMOL, and you should be able to immediately see the molecule and tunnels.

Note that it takes a bit of time to get used to working with PyMOL (choosing different visualization modes, making selections, etc.). Therefore, if you are a first time PyMOL user, it is advised that you spend a bit of time exploring the works of PyMOL itself before you attempt to visualize Mole 2.0 output.

6.7. Interactive tunnel calculation with Mole 2.0 inside PyMOL

If you are using Mac OS or Linux, you cannot run the GUI based version of Mole 2.0. Nevertheless, by using our PyMOL plug-in, you have the opportunity to interactively refine your Mole 2.0 calculation inside PyMOL in a similar manner as you would inside the Mole 2.0 GUI.

If you have already downloaded and unpacked the command line version of Mole 2.0, simply download the Mole 2.0 PyMOL plug-in from our website at http://webchem.ncbr.muni.cz/Platform/App/Mole#. Then start PyMOL, go to Plugins → Plugin Manager → Install new plug-in, navigate to the location where you have downloaded the Mole 2.0 plug-in, and load this file into PyMOL. Now the Mole 2.0 plug-in should be available in the list of available PyMOL plug-ins. If you do not find it in the list, try restarting PyMOL. In order to use the Mole 2.0 plug-in, simply double click on it in the list. A separate window will open, where you will be able to setup your Mole 2.0 calculation (Figure 28). At the top of the plug-in window there are a few tabs.

In the **Compute tunnels** tab, you may set the most basic requirements of your calculation (Figure 28 A). First, if you have more than one molecule loaded into PyMOL, here you may select which one you will use in your Mole

2.0 calculation. Next, you can define potential tunnel start points as a list of residues. If you do not specify any start points, Mole 2.0 will automatically generate start points at the bottom of all identified cavities. In the **Compute tunnels** tab you may also specify the location of the output files by the **Save output to** button. Additionally, if you would like to get start point suggestions from the CSA database, you must specify the location of the CSA database file by the **Select CSA.dat file** button. This file was provided to you together with the download of the command line version of Mole 2.0.

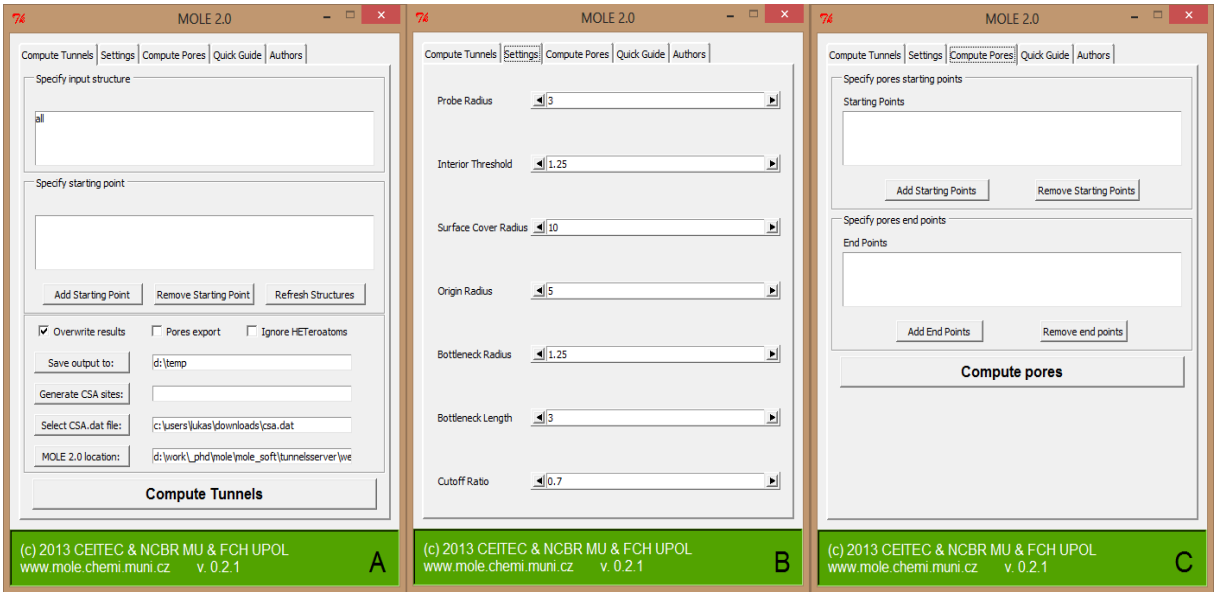

Figure 28: The Mole 2.0 plug-in for PyMOL allows you to run Mole 2.0 calculations from inside PyMOL. Basic computation settings can be found in the tabs Compute tunnels and Compute pores. Fine tuning can be achieved by adjusting the parameters in the Settings tab.

Remember that you need to tell PyMOL where to look for the Mole 2.0 executable file. In order to specify the location of the Mole 2.0 command line application, click the **Mole 2.0 location** button and navigate to the appropriate folder where you downloaded and extracted the files from our website. You may now start the calculation by using the **Compute tunnels** button.

The **Settings** tab (Figure 28 B) allows you to refine the calculation in a manner equivalent with the **Cavity parameters** and **Tunnel parameters** described in chapters 4.3.4. and 4.6. from Part I in this manual. The **Compute pores** tab gives you precise control over which points are used during the pore calculation procedure (for details please see chapter 4.6.).

Once you have set the appropriate parameters, start the Mole 2.0 calculation via the **Compute tunnels** button. You will be able to visualize all tunnels inside PyMOL directly, and moreover all physico-chemical properties will be available in the working directory as described in the chapters 4.4. and 4.5.

If you experience any issues in working with the PyMOL plug-in, please contact us via our website at http://web-chem.ncbr.muni.cz/Platform/App/Mole#.

Again, remember that while the Mole 2.0 plug-in offers lots of tool tips to guide you through setting up your Mole 2.0 calculation inside PyMOL, working with PyMOL itself is not very straightforward for complete beginners. Thus, if you are a first time PyMOL user, it is strongly advised that you spend a bit of time exploring the works of PyMOL itself (especially choosing different visualization modes, making selections, etc.) before you attempt to run Mole 2.0 calculations using the Mole 2.0 PyMOL plug-in.

Annex

List of adjustable parameters in Mole 2.0

Input parameters

Parameter	Mole 2.0 version	Type	Default value	Function
Probe Radius	GUI, CL	Double	3	Regulates level of detail of the molecular surface. Higher Probe Radius produces less detail
Interior Threshold	GUI, CL	Double	1.25	Minimum radius of void inside the protein structre, so that the void would be considered a cavity.
Origin Radius	GUI, CL	Double	5	If the user defined a tunnel start point, expand the search for tunnel start points to a sphere of radius Origin Radius.
Surface Cover Radius	GUI, CL	Double	10	Regulates the density of exit points tested at each outer boundary. Higher Surface Cover radius produces a lower density of exit points.
CustomVdW	CL	Double	per element	Allows to adjust the van der Waals radius of various atoms found in protein structures.
WorkingDirectory	CL	String	./output/	Path to the results folder
Origin	CL	Boolean	1	Controls the type of the tunnel start points used. If 1, then automatically detected start points are used. If 0, other start points must be defined.

Filtering parameters

Parameter	Mole 2.0 version	Type	Default value	Function
Bottleneck Radius	GUI, CL	Double	1.25	Minimal radius of a valid tunnel if Bottleneck Length is 0.
Bottleneck Length	GUI, CL	Double	0	Maximum length of a valid tunnel for which the radius is less than Bottleneck Radius
Cutoff Ratio	GUI, CL	Double	0.9	Maximum degree of similarity between two tunnels before one tunnel is discarded.

Export parameters

Parameter	Mole	Type	Default	Function

	2.0 version		value	
Mesh	CL	Boolean	0	Controls storing information about the mesh of detected tunnels, for subsequent visualization in PyMol.
MeshGz	CL	Boolean	0	Controls storing of mesh information in a zip file archive.
Mesh Density	CL	Double	1.33	Level of detail of mesh surface. The higher the Mesh Density, the lower the level of detail in the visualization.
Cavities	CL	Boolean	1	Controls whether cavities will be exported
PyMol	CL	Boolean	1	Controls whether a PyMol script will be generated, for subsequent visualization in PyMol.