Supplementary information

Estimating **M**olecular **A**ccessible **S**urface Area and **I**naccessible **V**olume.

Analytical evaluations using MASIV

Executable, for calculations using a single processor can be downloaded from GitHub repository https://github.com/gboulougouris/MASIV. The reader has the ability to perform the same basic calculations in any system with no restrictions apart from the use of a single processor. The executables are capable of performing the analytical volume and area calculations given any two xyz files, one for the protein and one for the ligand. The program can utilize the analytical evaluation of the volume of fused spheres, developed by L.R. Dodd and D.N. Theodorou (DODD L.R., and THEODOROU D.N.,1991, Molec. Phys., 72, 1313) in the form of a compiled library, linked with the executables. The users are therefore kindly encouraged to site the Dodd and Theodorou work along with this one.

Example of a serial runs :

(avoloumgf.exe   prot.xyz lig.xyz f_R

Executable (os dependent) (For Linux using gfortran 7.5 https://github.com/gboulougouris/MASIV/releases/tag/v1.2), For Linux using gfortran 4.4 https://github.com/gboulougouris/MASIV/releases/tag/v1.1) Mac (: gfortran 10 https://github.com/gboulougouris/MASIV/releases/tag/v1.0)	prot.xyz : Protein xyz filename	lig.xyz : Ligand xyz filename

(the appropriate os dependent libvolume.a file has to be in the same directory)

f_R (optional)	Scaling factor for all radius (default is 1)

output files:

anvol.out	Main results **excluded volume, accessible area**

*Linux (Ubuntu). Build using Gfortran, Thread model: posix, gcc version 7.5.0 or version 4.4.0

Finlay is is possible to overbite the values of the minimum inter atomic interactions by pleasing in the same directory with the executable a file with the name radious.dat with the following format.

Line 1	.False. .True.
Line 2 number of atom types	8
Line 3 $f_{R,}$, $R_1, R_2, ... R_8$	1 3.677d0 3.539d0 3.593d0 3.455d0 2.895d0 2.757d0 3.033d0 2.895d0
Line 4 names of the atom types	C O N S Li Bi Cl F

Visualization using Jmol[1] or VMD[2]

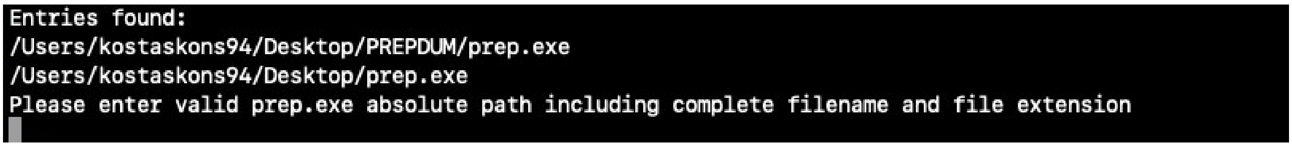
As part of the ESI we provide also two scripts ***JMOL_MASIV.sh*** and ***VMD_MASIV.sh*** scripts, which were developed for Linux and macOS based systems. However, prior to use of ***JMOL_MASIV.sh*** or ***VMD_MASIV.sh*** scripts, user should have downloaded from https://github.com/gboulougouris/MASIV the FORTRAN-based program, which prepares the set of auxiliary spheres. Note that this process results in an approximating representation since the radius of each of the auxiliary spheres is being set by an atom type, whereas in the analytical evaluation the radius of each sphere is set based on the types of two atoms one in the ligand and one in the protein. The provided ***prep.exe*** executable file in Linux must be stored in the same folder as ***JMOL_MASIV.sh*** or ***VMD_MASIV.sh***. User should also choose between VMD or Jmol molecular visualization tool and therefore either download VMD (https://www.ks.uiuc.edu/Development/Download/download.cgi?PackageName=VMD) and install it locally or download Jmol viewer (http://jmol.sourceforge.net/) and extract the downloaded .zip archive to the desired directory.

The attached ***JMOL_MASIV.sh*** and ***VMD_MASIV.sh*** scripts are identical upon deployment except for the molecular visualization process at the end which is caried out by Jmol or VMD after use of the corresponding script. The following examples of use apply to both developed scripts without any differences. Execution of either ***JMOL_MASIV.sh*** or ***VMD_MASIV.sh*** script may take from 0 up to 3 arguments as shown below.

???_MASIV.sh output_directory f_R seed_number

???_MASIV.sh :	***JMOL_MASIV.sh*** or ***VMD_MASIV.sh***
output_directory (optional)	Directory of AVOLUME_OUTPUT folder containing the generated output files (default is current directory if this argument is omitted)
f_R (optional)	Scaling factor for all radius (default is 1 if this argument is omitted)
seed_number (optional)	Random number for different ligand positioning on protein surface (default is 1234 if this argument is omitted)

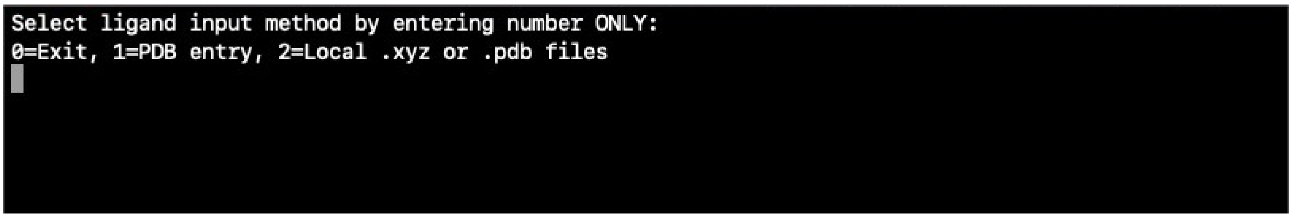
When the ***JMOL_MASIV.sh*** or ***VMD_MASIV.sh*** script is executed, the algorithm automatically searches for the installed molecular visualization software (Jmol or VMD respectively) and the executables ***avolum.exe*** and ***prep.exe***. If the automatic search returns zero or multiples instances of the aforementioned programs, the user is prompted to enter the absolute path of the corresponding file including its complete filename and file extension as in **Figure S.I. 1**.



```
Entries found:
/Users/kostaskons94/Desktop/PREPDUM/prep.exe
/Users/kostaskons94/Desktop/prep.exe
Please enter valid prep.exe absolute path including complete filename and file extension
```

Figure S.I. 1. User prompt for entering valid ***prep.exe*** absolute directory due to multiple instances of ***prep.exe*** file shown below "Entries found" line.

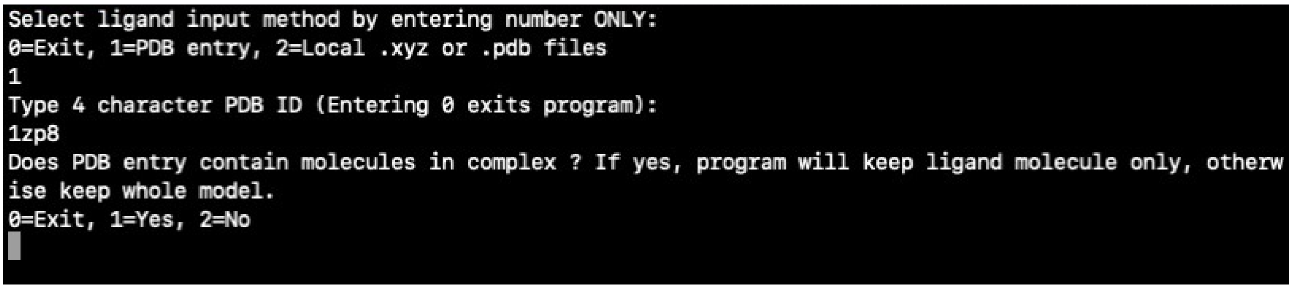
Afterwards, the user is prompted to select input method and molecular structure of preference, consecutively for ligand and protein molecules. All user prompts of this script have similar selection patterns for both ligand and protein molecules as well as provide the option to terminate/exit the program at any time by simply entering number 0. Regarding input method prompt, the user is able to examine either online PDB entries or local .xyz and .pdb molecular structure files by entering the assigned selection number respectively as in **Figure S.I. 2**.



```
Select ligand input method by entering number ONLY:
0=Exit, 1=PDB entry, 2=Local .xyz or .pdb files
```

Figure S.I. 2. User prompt for input method selection.

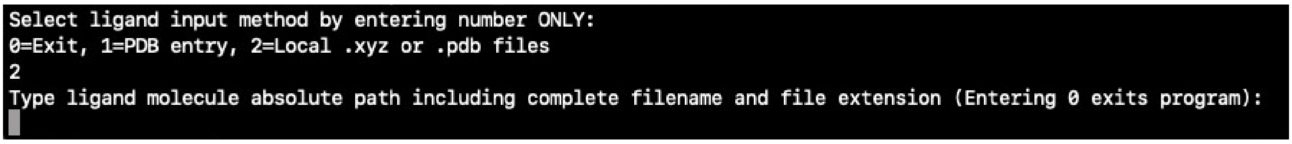
If "PDB entry" input method is selected, the user is allowed to type the 4-character PDB ID below the displayed message "Type 4 character PDB ID (Entering 0 exits program)". Entering the PDB ID of preference triggers the model splitting section of the algorithm where the user may choose between discriminating ligand-protein molecules if found in complex at PDB or considering the selected PDB entry as a whole input model (**Figure S.I. 3**). Preferably, the user should know beforehand if the input PDB entry possesses the examined molecules in complex.



```
Select ligand input method by entering number ONLY:
0=Exit, 1=PDB entry, 2=Local .xyz or .pdb files
1
Type 4 character PDB ID (Entering 0 exits program):
1zp8
Does PDB entry contain molecules in complex ? If yes, program will keep ligand molecule only, otherw
ise keep whole model.
0=Exit, 1=Yes, 2=No
```

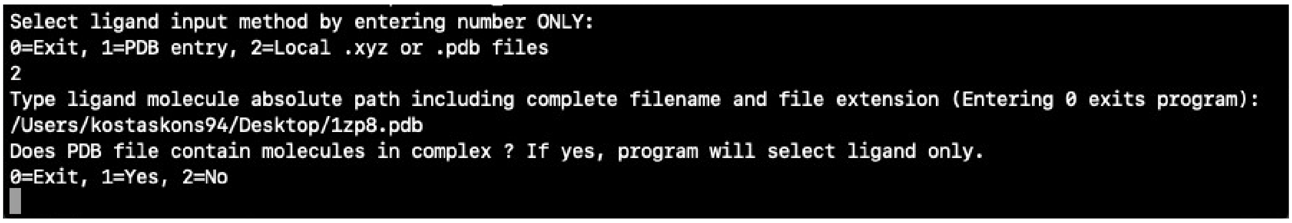
Figure S.I. 3. User prompt for PDB entry model splitting selection after entering PDB ID of choice, here 1zp8.

Alternatively, if the "Local .xyz and .pdb files" input method is selected, the user may type the absolute directory which is required to locate the .xyz or .pdb ligand/protein file of preference, under the message "Type ligand/protein molecule absolute path including complete filename and file extension (Entering 0 exits program):" as in **Figure S.I. 4**. If local .pdb file directory is entered, the model splitting section of the algorithm is triggered, similar to the online PDB entry selection process mentioned in the previous paragraph, where the user is prompted to either separate ligand-protein molecules if found in complex inside the molecular configuration file or manipulate the .pdb file as a whole input model (**Figure S.I. 5**).



```
Select ligand input method by entering number ONLY:
0=Exit, 1=PDB entry, 2=Local .xyz or .pdb files
2
Type ligand molecule absolute path including complete filename and file extension (Entering 0 exits program):
```

Figure S.I. 4. Molecular configuration file directory prompt after "Local .xyz or .pdb files" input method selected.

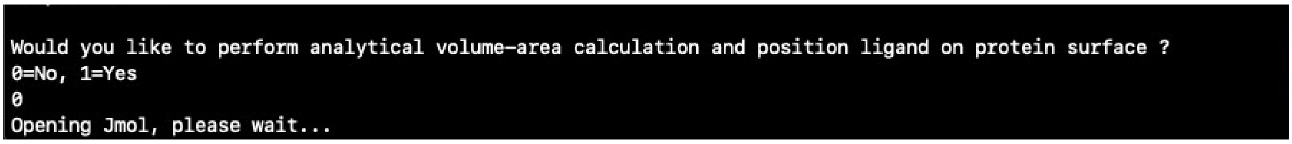


```
Select ligand input method by entering number ONLY:
0=Exit, 1=PDB entry, 2=Local .xyz or .pdb files
2
Type ligand molecule absolute path including complete filename and file extension (Entering 0 exits program):
/Users/kostaskons94/Desktop/1zp8.pdb
Does PDB file contain molecules in complex ? If yes, program will select ligand only.
0=Exit, 1=Yes, 2=No
```

Figure S.I. 5. User prompt for local .pdb file model splitting selection after entering the .pdb file directory of preference, here the absolute path of 1zp8.pdb file.

After the desired selections are made, the algorithm proceeds to loading the ligand and protein molecules of preference to Jmol or VMD visualization software where the user is notified by the message "Loading data to Jmol/VMD, please wait…" upon execution of the corresponding script. This step is crucial for the proper modification and storage of the selected molecules inside the created AVOLUME_OUTPUT folder which will later contain all the generated output files of ***JMOL_MASIV.sh*** and ***VMD_MASIV.sh*** scripts. A new subdirectory is created inside the AVOLUME_OUTPUT folder each time the scripts are executed, named after the tested molecules accompanied by the visualization method (Jmol/VMD) and f_R that was used as follows: ***protein_P-ligand_L_VMD_fR#*** or ***protein_P-ligand_L_VMD_fR#***. The ligand and protein molecules of preference are saved in this new subdirectory acquiring the appropriate .pdb or .xyz file extension depending on the user input method Due to the fact that ***prep.exe*** and ***avolum.exe*** accept only .xyz files, the selected ligand and protein molecules are also necessarily stored there in .xyz format with filename suffices "_L" and "_P" for ligand and protein respectively.

Execution of ***prep.exe*** takes place next as the message "Executing prep.exe" suggests using the aforementioned ***protein_P.xyz*** and ***ligand_L.xyz*** files as input in order to output the ***protein_P-ligand_L_dummy.xyz*** file in the new subdirectory created inside AVOLUME_OUTPUT folder containing the generated set of auxiliary spheres from the tested molecules. Following execution of ***prep.exe***, the user is prompted for the analytical accessible surface area and inaccessible volume calculation as well as ligand positioning on protein surface (**Figure S.I. 6**). Unless f_R and seed_number arguments are present, the above notification process which involves running ***avolum.exe*** is not triggered. Furthermore, ***positioned_ligand_Lxyz*** file with the xyz coordinates of the positioned ligand molecule on the protein surface and ***anvol.out*** file possessing the analytical values of molecular accessible surface area (MASA) and molecular inaccessible volume (MIV) are not created. Entering 0 for No or 1 for Yes, leads to the final stage of the algorithm where Jmol/VMD is fully deployed after the message "Opening Jmol/VMD, please wait…" .



```
Would you like to perform analytical volume-area calculation and position ligand on protein surface ?
0=No, 1=Yes
0
Opening Jmol, please wait...
```

Figure S.I. 6. User prompt for performing analytical accessible surface area and inaccessible volume calculation as well as ligand positioning on protein surface. This prompt is not displayed if f_R and seed_number arguments are omitted.

At the last stage of ***JMOL_MASIV.sh*** and ***VMD_MASIV.sh*** scripts, the visualization of ligand and protein models takes place using either Jmol (**Figure S.I. 7a**) or VMD. In the case of Jmol (***JMOL_MASIV.sh***), a gray colored isosurface around the loaded set of auxiliary spheres is formed constituting the graphical representation of molecular accessible surface. For more convenient, the generated set of pink colored auxiliary spheres from the ***protein_P-ligand_L_dummy.xyz*** file is hidden and therefore not displayed in the final state of Jmol (**Figure S.I. 7b**). The selected ligand molecule will also be positioned on the protein surface at the proper annotated minimum distance if ***avolum.exe*** is executed with f_R or seed_number arguments present (**Figure S.I. 7c**). The Jmol visualization process is completed when the two messages below are displayed (**Figure S.I. 8**), which signal overall completion of the executed ***JMOL_MASIV.sh*** algorithm. According to the displayed message of **Figure S.I. 8a,** the input ligand and protein molecules are stored in a new subdirectory inside AVOLUME_OUTPUT folder named after them alongside Jmol's final state ***protein_P-ligand_L.spt*** script for easier future access and regeneration of molecular accessible surface in Jmol. The values of molecular accessible surface area and inaccessible volume are then retrieved from Jmol's built-in variables based on the loaded ***protein_P-ligand_L_dummy.xyz*** model. The calculated molecular accessible surface area and inaccessible volume by Jmol and the corresponding analytical estimation by ***avolum.exe*** if applicable are displayed and saved in the ***results.out*** file as prompted (**Figure S.I. 8b**).

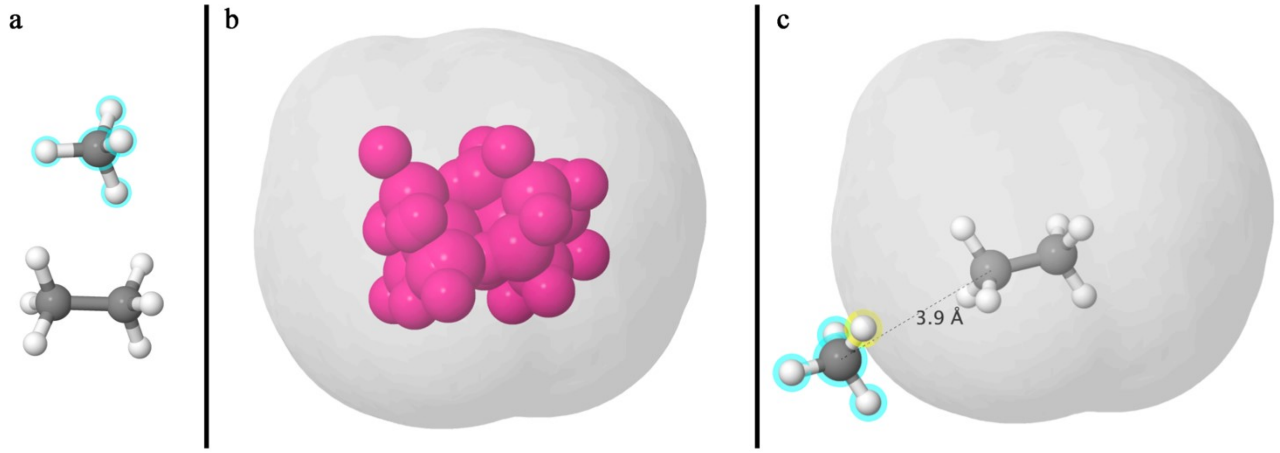
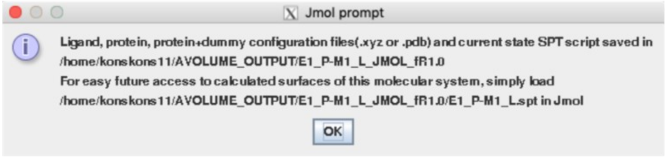


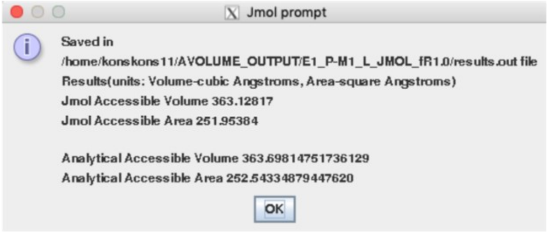
Figure S.I. 7 (a) Jmol visualization output after execution of ***JMOL_MASIV.sh***. A showcase of the test system consisting of two molecules, an ethane representing the protein molecule of the system and a methane displaying the inserted ligand molecule distinguished by its cyan haloed atoms. **(b)** An intermediate stage of the Jmol visualization process with the generated pink auxiliary spheres surrounded by their gray 3D isosurface. The set of auxiliary spheres is loaded from the created ***protein_P-ligand_L_dummy.xyz*** file and is later hidden for representation purposes. Point of view rotated on y axis by 45 degrees. **(c)** An illustration of the final state of Jmol where ***avolum.exe*** was executed with user-defined f_R=1.0 and seed_number=3000 resulting in the methane ligand molecule being positioned on the ethane protein surface. Point of view rotated on y axis by 45 degrees. Also depicted here is the excluded volume around the ethane molecule of the test system, where the generated gray surface points coincide with the center of the first atom (yellow highlighted sphere) of the inserted methane ligand in opaque color and cyan halos positioned at the annotated minimum interatomic distance (=3.9 Å) without penetrating the created gray isosurface.

a



```
Ligand, protein, protein+dummy configuration files(.xyz or .pdb) and current state SPT script saved in
/home/konskons11/AVOLUME_OUTPUT/E1_P-M1_L_JMOL_fR1.0
For easy future access to calculated surfaces of this molecular system, simply load
/home/konskons11/AVOLUME_OUTPUT/E1_P-M1_L_JMOL_fR1.0/E1_P-M1_L.spt in Jmol
```

b



```
Saved in
/home/konskons11/AVOLUME_OUTPUT/E1_P-M1_L_JMOL_fR1.0/results.out file
Results(units: Volume-cubic Angstroms, Area-square Angstroms)
Jmol Accessible Volume 363.12817
Jmol Accessible Area 251.95384

Analytical Accessible Volume 363.69814751736129
Analytical Accessible Area 252.54334879447620
```

Figure S.I. 8 Overall completion of the executed ***JMOL_MASIV.sh*** upon the molecular test system of **Figure 7** (protein ethane - ligand methane) notified by two Jmol prompt messages. **(a)** Jmol prompt message informing user that input ligand and protein molecules are also stored in the created subdirectory inside the AVOLUME_OUTPUT folder, here ***/home/username/AVOLUME_OUTPUT/E1_P-M1_L_JMOL_fR1.0*** alongside Jmol's final state script, e.g. ***E1_P-M1_L.spt*** for corresponding molecular test system. **(b)** Jmol prompt message displaying the calculated values of MASA and MIV. The "Jmol Accessible Volume" and "Jmol Accessible Area" lines refer to Jmol calculations while the last "Analytical Accessible Volume" and "Analytical Accessible Area" lines belong to ***avolum.exe*** estimations. Results are also saved in the ***results.out*** file of ***/home/username/AVOLUME_OUTPUT/E1_P-M1_L_JMOL_fR1.0*** subdirectory as shown.

However, in the case of VMD (***VMD_MASIV.sh***) no isosurfaces are formed around the loaded set of auxiliary spheres. The ***protein_P-ligand_L_dummy.xyz*** file is additionally converted to ***protein_P-ligand_L_dummy.vtf*** file which is mainly utilized for the correct van der Waals radii assignment to the generated dummy atoms and calculation of MASA. For more convenient, the generated set of auxiliary spheres from the ***protein_P-ligand_L_dummy.vtf*** file is hidden and therefore not displayed in the final state of VMD. The selected ligand molecule is positioned on the protein surface at the proper minimum distance if ***avolum.exe*** is executed with f_R or seed_number arguments present (**Figure S.I. 9**). VMD provides only calculation of the accessible surface area by built-in commands and its output value is saved in the ***results.out*** file inside AVOLUME_OUTPUT folder together with the analytically estimated MASA and MIV by ***avolum.exe*** if applicable (**Figure S.I. 10**). For easier future access and recalculation of molecular accessible surface area in VMD, the ***protein_P-ligand_L-VMD_script.txt*** file is created and stored in the new subdirectory inside AVOLUME_OUTPUT folder.

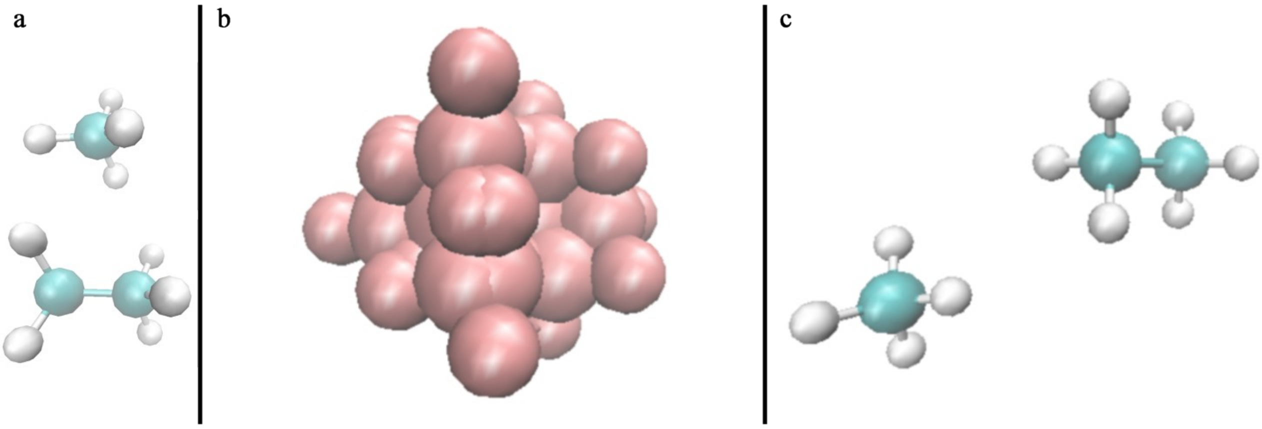
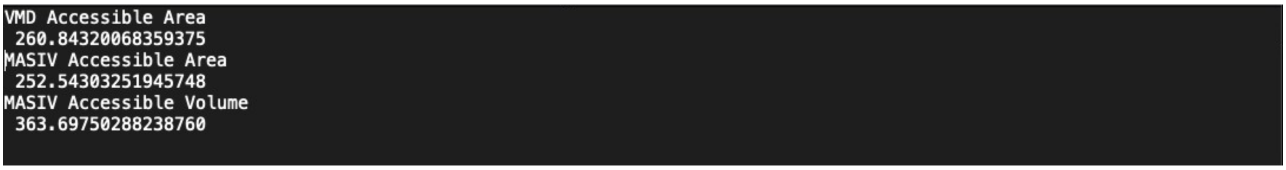


Figure S.I. 9 (a) VMD visualization output after execution of ***VMD_MASIV.sh***. A showcase of the same test system as in **Figure 7** consisting of two molecules, an ethane representing the protein molecule of the system (lower molecule) and a methane displaying the inserted ligand molecule (upper molecule). **(b)** The generated set of auxiliary spheres is loaded from the created ***protein_P-ligand_L_dummy.vtf*** file which is later hidden for representation purposes. Point of view rotated on y axis by 45 degrees. **(c)** An illustration of the final state of VMD where ***avolum.exe*** was executed with user-defined f_R=1.0 and seed_number=3000 resulting in the methane ligand molecule being positioned at the minimum interatomic distance on the ethane protein surface. Point of view rotated on y axis by 45 degrees.



```
VMD Accessible Area
 260.84320068359375
MASIV Accessible Area
 252.54303251945748
MASIV Accessible Volume
 363.69750288238760
```

Figure S.I. 10 Contents of ***results.out*** VMD output file. The "VMD Accessible Area" value refers to VMD calculation while "MASIV Accessible Volume" and "MASIV Accessible Area" values belong to ***avolum.exe*** estimations.

Examples of use:

Some examples using both scripts are included in the section below applying various input methods. Following step-by-step these examples should yield the exact same results by any user.

Executing JMOL_MASIV.sh with online[3] PDB molecules as input

For our first example, the script is utilized with all of its 3 arguments passed in order to experience full deployment of ***JMOL_MASIV.sh*** as follows :

```
./JMOL_MASIV.sh    ./         1.0       1000
```

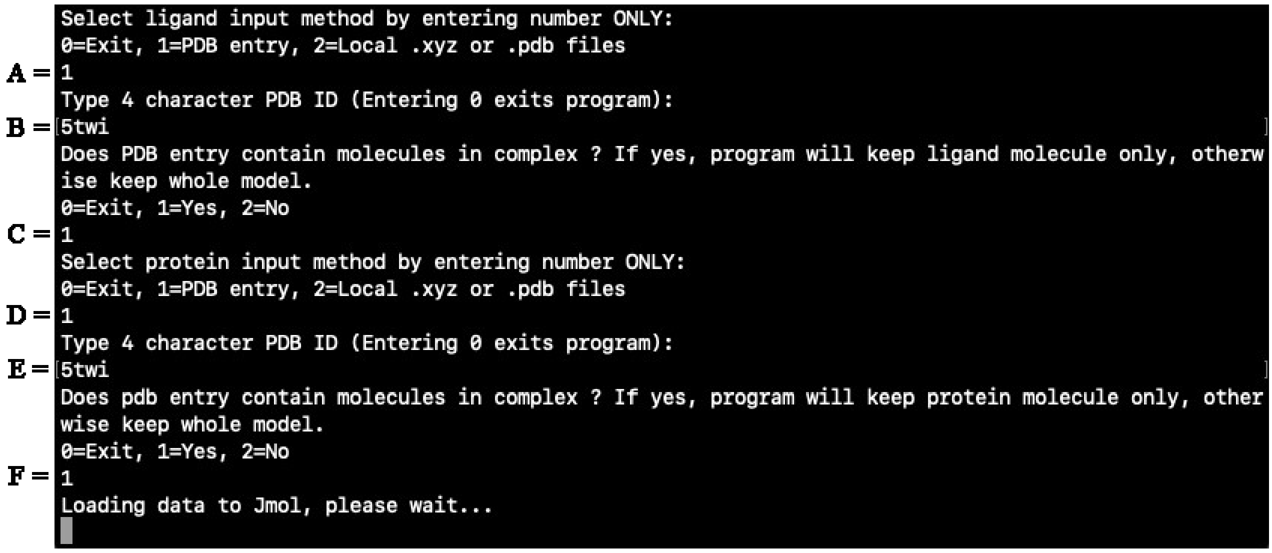
where

./ is the output_directory ,

1.0 is the f_R , and

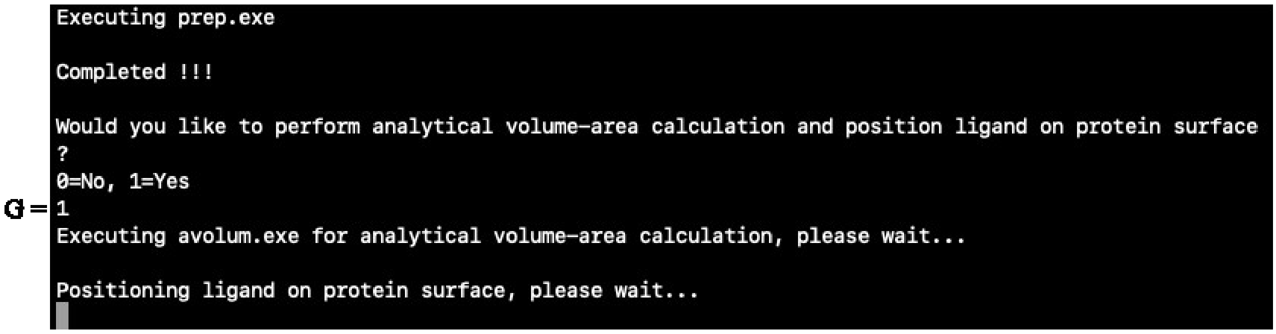
1000 is the seed_number .

Afterwards, the algorithm scans the user system for installed versions of Jmol, ***prep.exe*** and ***avolum.exe*** and will inform the user if zero or multiple instances of those programs are found in order to enter the corresponding valid absolute directories. A new folder named ***AVOLUME_OUTPUT*** is created in the output_directory (./**).** "Online PDB" and "local .pdb/.xyz" input methods are similar and guide the user effortlessly through the different stages of the algorithm. In this example, an online PDB entry is used which possesses ligand and protein molecules in complex. For this reason, the selected PDB entry is processed and split into separate ligand and protein molecules by the algorithm. The tested online PDB entry here is ***5twi***[3] which splits into its ligand and protein molecules with 20 and 44 atoms respectively. The ***5twi*** PDB entry is loaded and split if A=1, B=5twi, C=1, D=1, E=5twi and F=1 like below:



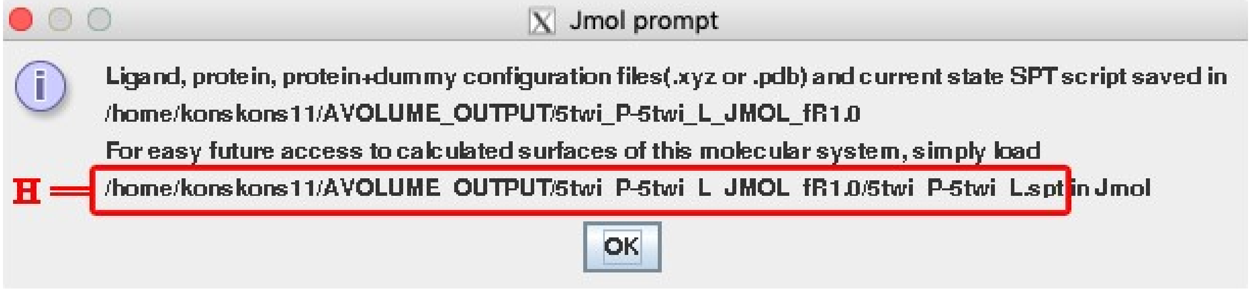
```
Select ligand input method by entering number ONLY:
0=Exit, 1=PDB entry, 2=Local .xyz or .pdb files
1
Type 4 character PDB ID (Entering 0 exits program):
[5twi                                                                              ]
Does PDB entry contain molecules in complex ? If yes, program will keep ligand molecule only, otherw
ise keep whole model.
0=Exit, 1=Yes, 2=No
1
Select protein input method by entering number ONLY:
0=Exit, 1=PDB entry, 2=Local .xyz or .pdb files
1
Type 4 character PDB ID (Entering 0 exits program):
[5twi                                                                              ]
Does pdb entry contain molecules in complex ? If yes, program will keep protein molecule only, other
wise keep whole model.
0=Exit, 1=Yes, 2=No
1
Loading data to Jmol, please wait...
```

The message "Loading data to Jmol, please wait.." is displayed next and the molecules are processed by Jmol in order to separate ligand and protein molecules as well as save them locally in .pdb and .xyz format. A new subdirectory inside ***AVOLUME_OUTPUT*** folder is created named ***5twi_P-5twi_L_JMOL_fR1.0*** and the ***5twi_P.pdb, 5twi_P.xyz, 5twi_L.pdb*** and ***5twi_L.xyz*** files are stored in it. The ***5twi_P.xyz*** and ***5twi_L.xyz*** are used by ***prep.exe*** for generating the set of auxiliary spheres which are saved as ***5twi_P-5twi_L-dummy.xyz*** in the ***5twi_P-5twi_L_JMOL_fR1.0*** folder after successful execution. The user is then prompted if he/she would like to perform analytical MASA - MIV calculation and position ligand molecule on protein surface. If G=1 like below the ***avolum.exe*** is executed and will take a few seconds to complete.

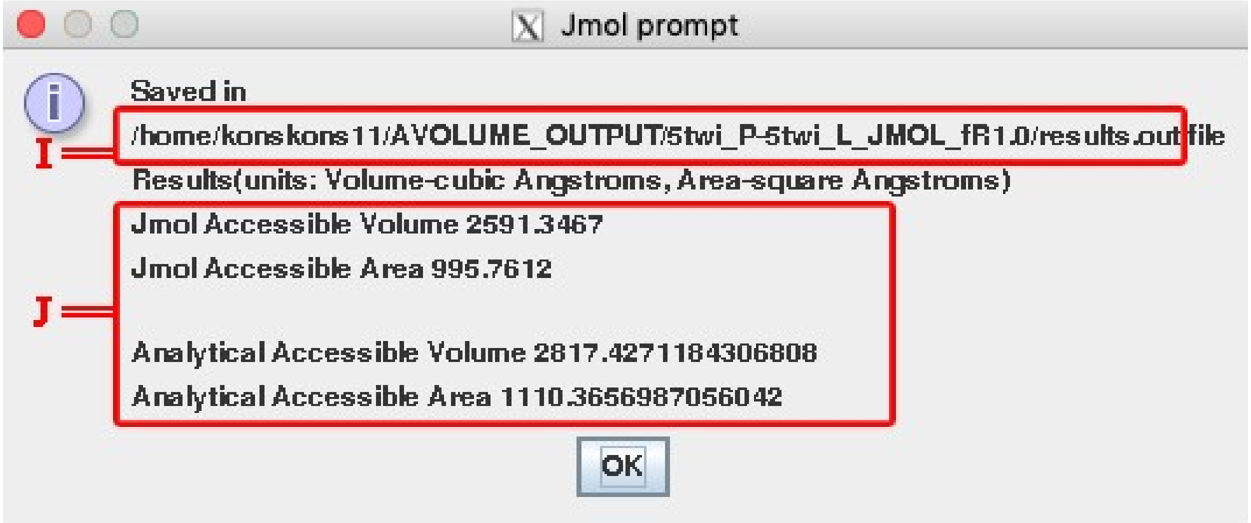


```
Executing prep.exe

Completed !!!

Would you like to perform analytical volume-area calculation and position ligand on protein surface
?
0=No, 1=Yes
1
Executing avolum.exe for analytical volume-area calculation, please wait...

Positioning ligand on protein surface, please wait...
```

Upon completion, ***positioned_5twi_Lxyz*** file with the xyz coordinates of the positioned ligand molecule on the protein surface and ***anvol.out*** file possessing the analytical estimated values of molecular accessible surface area (MASA) and molecular inaccessible volume (MIV) are created inside the ***5twi_P-5twi_L_JMOL_fR1.0*** folder. The message "Opening Jmol, please wait..." is displayed next and Jmol viewer starts opening the app user-interface window accompanied by its console. The ***5twi*** ligand (***5twi_L.xyz***), protein (***5twi_P.pdb***), dummy (***5twi_P-5twi_L-dummy.xyz***) and positioned ligand (***positioned_5twi_Lxyz***) models are loaded in Jmol and the visualization process takes place creating a gray 3D isosurface around the generated set of auxiliary spheres of the ***5twi_P-5twi_L-dummy.xyz*** model. Two messages are displayed upon completion of the ***JMOL_MASIV.sh*** script as shown below.



Ligand, protein, protein+dummy configuration files(.xyz or .pdb) and current state SPT script saved in
/home/konskons11/AVOLUME_OUTPUT/5twi_P-5twi_L_JMOL_fR1.0
For easy future access to calculated surfaces of this molecular system, simply load
/home/konskons11/AVOLUME_OUTPUT/5twi_P-5twi_L_JMOL_fR1.0/5twi_P-5twi_L.spt in Jmol



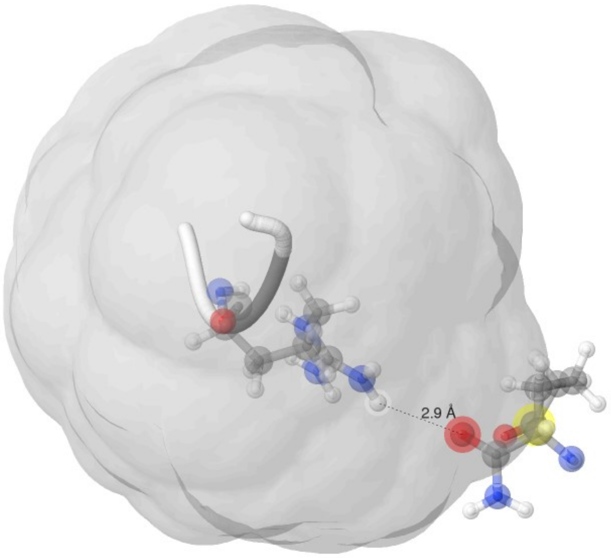
Saved in
/home/konskons11/AVOLUME_OUTPUT/5twi_P-5twi_L_JMOL_fR1.0/results.out file
Results(units: Volume-cubic Angstroms, Area-square Angstroms)
Jmol Accessible Volume 2591.3467
Jmol Accessible Area 995.7612

Analytical Accessible Volume 2817.4271184306808
Analytical Accessible Area 1110.3656987056042

In the first one, the user is notified that all of the loaded models in Jmol are saved in the output_directory/***AVOLUME_OUTPUT/5twi_P-5twi_L_JMOL_fR1.0*** directory. A script containing all the necessary commands to reproduce Jmol's final state and calculations is created named ***5twi_P-5twi_L.spt*** (absolute directory H = output_directory/***AVOLUME_OUTPUT/5twi_P-5twi_L_JMOL_fR1.0/5twi_P-5twi_L.spt***). The second message however holds the calculation results of MASA and MIV which are also exported to ***results.out*** file (absolute directory I = output_directory/***AVOLUME_OUTPUT/5twi_P-5twi_L_JMOL_fR1.0/results.out***) apart from being displayed at section J of the prompt message. The "Jmol Accessible Volume" and "Jmol Accessible Area" lines refer to Jmol calculations while the last "Analytical Accessible Volume" and "Analytical Accessible Area" lines belong to ***avolum.exe*** estimations. The results of MASA and MIV for the ***5twi*** test molecular system as input here should be:

```
Results(units: Volume-cubic Angstroms, Area-square Angstroms)
Jmol Accessible Volume 2591.3467
Jmol Accessible Area 995.7612

Analytical Accessible Volume 2817.4271184306808
Analytical Accessible Area 1110.3656987056042
```

Clicking OK on the prompt messages reveals the final state of Jmol to the user which is presented below.



For more convenient, point of view is rotated on y axis by 45 degrees. This is the graphical illustration of the final state of Jmol where ***avolum.exe*** was executed with user-defined f_R=1.0 and seed_number=1000 resulting in the ***5twi_L.xyz*** ligand molecule being positioned on the ***5twi_P.pdb*** protein surface. The loaded ***5twi_P-5twi_L-dummy.xyz*** and ***5twi_L.xyz*** models are hidden for representation purposes. However, depicted here is the excluded volume around the ***5twi_P.pdb*** protein molecule of the test system based on the gray isosurface of ***5twi_P-5twi_L-dummy.xyz*** model, where the generated gray surface points coincide with the center of the first atom (yellow highlighted sphere) of the ***positioned_5twi_Lxyz*** ligand positioned at the annotated minimum interatomic distance (=2.9 Å) without penetrating the created gray isosurface. The minimum interatomic distance is equal to the sum of the corresponding atomic radii multiplied by the applied f_R. The atoms found in minimum interatomic distance are the hydrogen atom (white halo) of ***5twi_P.pdb*** and the oxygen atom (red halo) of ***positioned_5twi_Lxyz*** which possess 1.2 Å and 1.7 Å atomic radii respectively, so (1.2 Å + 1.7 Å) * 1.0 = 2.9 Å minimum interatomic distance.

Executing VMD_MASIV.sh with local .xyz molecules as input

For our second example, the ***VMD_MASIV.sh*** script is utilized with 2 of its 3 arguments passed as follows :

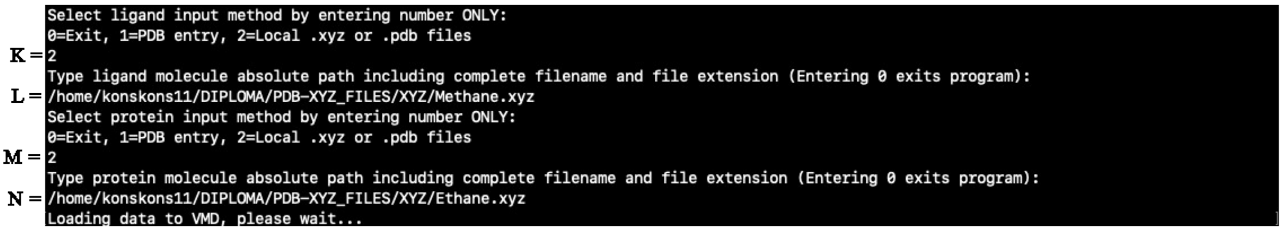
```
./VMD_MASIV.sh    ./        0.5
```

where

./ is the output_directory , and

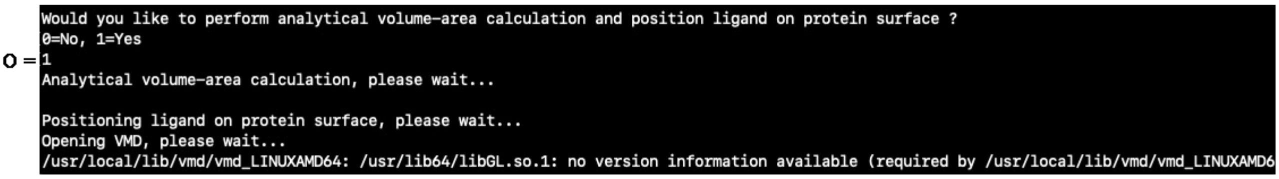
0.5 is the f_R .

Afterwards, the algorithm scans the user system for installed versions of VMD, ***prep.exe*** and ***avolum.exe*** and will inform the user if zero or multiple instances of those programs are found in order to enter the corresponding valid absolute directories. A new folder named ***AVOLUME_OUTPUT*** is created in the output_directory (./). “Online PDB” and “local .pdb/.xyz” input methods are similar and guide the user effortlessly through the different stages of the algorithm. In this example, a local .xyz files are used for both ligand and protein molecules. The tested local .xyz files here are the ***Methane.xyz***[4] and ***Ethane.xyz***[5] acting as ligand and protein molecules respectively. The .xyz files should be first downloaded locally and then loaded by entering K=2, L={absolute path of ***Methane.xyz*** including filename and extension}, M=2 and N={absolute path of ***Ethane.xyz*** including filename and extension} like below:



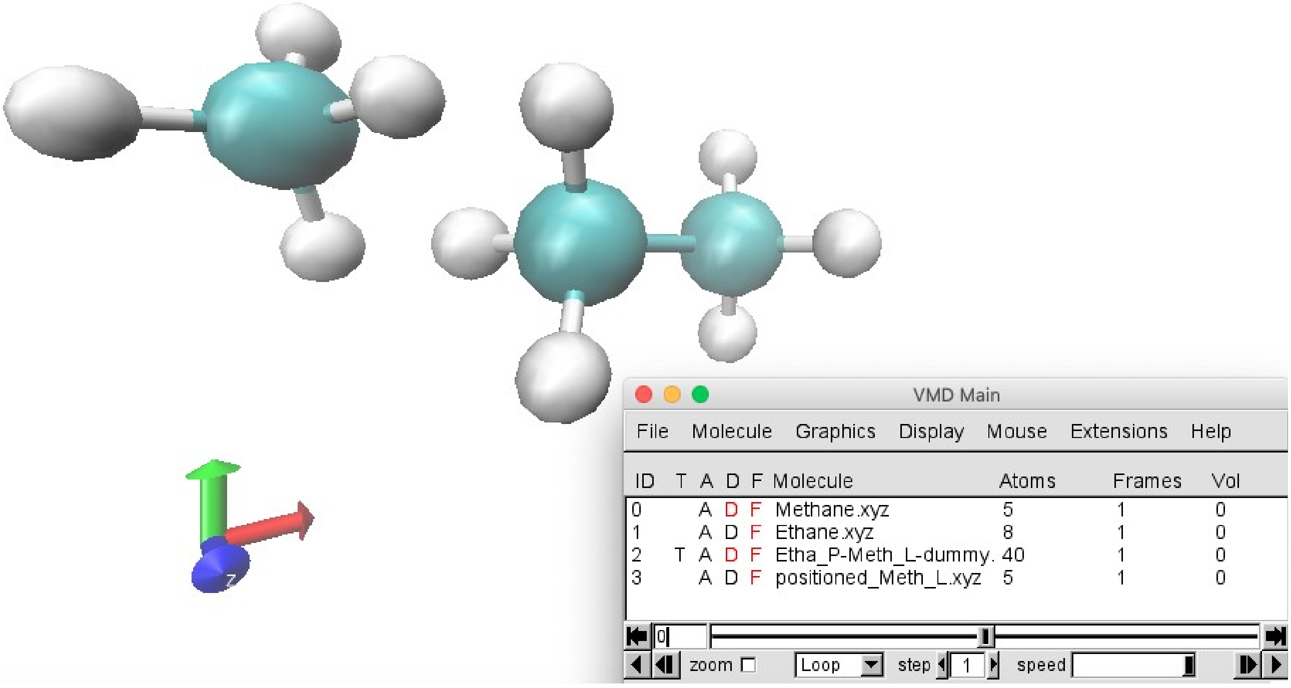
```
    Select ligand input method by entering number ONLY:
    0=Exit, 1=PDB entry, 2=Local .xyz or .pdb files
K = 2
    Type ligand molecule absolute path including complete filename and file extension (Entering 0 exits program):
L = /home/konskons11/DIPLOMA/PDB-XYZ_FILES/XYZ/Methane.xyz
    Select protein input method by entering number ONLY:
    0=Exit, 1=PDB entry, 2=Local .xyz or .pdb files
M = 2
    Type protein molecule absolute path including complete filename and file extension (Entering 0 exits program):
N = /home/konskons11/DIPLOMA/PDB-XYZ_FILES/XYZ/Ethane.xyz
    Loading data to VMD, please wait...
```

The message "Loading data to VMD, please wait.." is displayed next and the molecules are processed by VMD. A new subdirectory inside ***AVOLUME_OUTPUT*** folder is created named ***Etha_P-Meth_L_VMD_fR0.5*** and the ***Ethane.xyz, Etha_P.xyz, Methane.xyz*** *and* ***Meth_L.xyz*** files are stored in it. The ***Etha_P.xyz*** and ***Meth_L.xyz*** are used by ***prep.exe*** for generating the set of auxiliary spheres which are saved as ***Etha_P-Meth_L-dummy.xyz*** in the ***Etha_P-Meth_L_VMD_fR0.5*** folder after successful execution. The ***Etha_P-Meth_L-dummy.xyz*** is additionally converted into the ***Etha_P-Meth_L-dummy.vtf*** file for later use by VMD as it contains the correct van der Waals radii assigned to the generated dummy atoms for calculation of MASA. The user is then prompted if he/she would like to perform analytical MASA - MIV calculation and position ligand molecule on protein surface. If O=1 like below the ***avolum.exe*** is executed and will take a few seconds to complete.



```
    Would you like to perform analytical volume-area calculation and position ligand on protein surface ?
    0=No, 1=Yes
O = 1
    Analytical volume-area calculation, please wait...

    Positioning ligand on protein surface, please wait...
    Opening VMD, please wait...
    /usr/local/lib/vmd/vmd_LINUXAMD64: /usr/lib64/libGL.so.1: no version information available (required by /usr/local/lib/vmd/vmd_LINUXAMD6
```

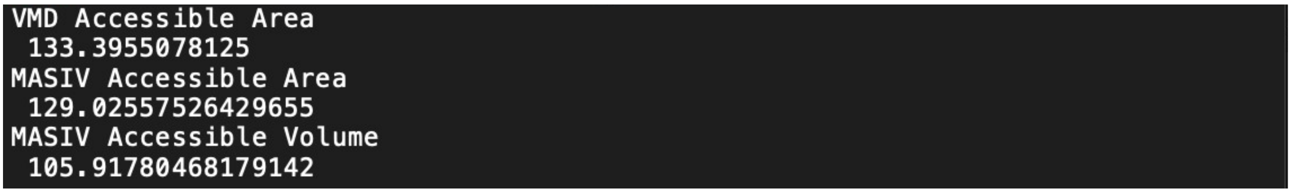
Upon completion, ***positioned_Meth_L xyz*** file with the xyz coordinates of the positioned ligand molecule on the protein surface and ***anvol.out*** file possessing the analytical estimated values of molecular accessible surface area (MASA) and molecular inaccessible volume (MIV) are created inside the ***Etha_P-Meth_L_VMD_fR0.5*** folder. The message “Opening VMD, please wait...” is displayed next and VMD starts opening the app user-interface window accompanied by a smaller window titled “VMD Main”. The final state of VMD as viewed by the user is presented below.



ID	T	A	D	F	Molecule	Atoms	Frames	Vol
0		A	D	F	Methane.xyz	5	1	0
1		A	D	F	Ethane.xyz	8	1	0
2	T	A	D	F	Etha_P-Meth_L-dummy.	40	1	0
3		A	D	F	positioned_Meth_L.xyz	5	1	0

The ***Methane*** ligand (***Methane.xyz***), ***Ethane*** protein (***Ethame.xyz***), dummy (***Etha_P-Meth_L-dummy.xyz***) and positioned ligand (***positioned_Meth_Lxyz***) models are loaded in VMD as shown inside the “VMD Main” titled window and the MASA calculation process takes place. For more convenient, point of view is rotated on y axis by 45 degrees. This is the graphical illustration of the final state of Jmol where ***avolum.exe*** was executed with user-defined f_R=0.5 resulting in the ***Methane.xyz*** ligand molecule being positioned on the ***Ethane.xyz*** protein surface. The loaded ***Etha_P-Meth_L-dummy.xyz*** and ***Methane.xyz*** models are hidden for representation purposes. However, depicted here is the centered ***Etha_P.xyz*** protein molecule of the test system with the ***positioned_Meth_Lxyz*** molecule positioned at the minimum interatomic distance. A script containing all the necessary commands to reproduce VMD’s final state and calculations is created at the output_directory***/AVOLUME_OUTPUT/Etha_P-Meth_L_VMD_fR0.5/*** named ***Etha_P-Meth_L-***

VMD_script.txt. A report holding the calculation results of MASA is also exported to ***results.out*** file inside the ***AVOLUME_OUTPUT/Etha_P-Meth_L_VMD_fR0.5/*** folder. The contents of the ***results.out*** file are displayed below and should be the same if tested by the user for ***Ethane.xyz – Methane.xyz*** molecular system:



```
VMD Accessible Area
 133.3955078125
MASIV Accessible Area
 129.02557526429655
MASIV Accessible Volume
 105.91780468179142
```

VMD Accessible Area

133.3955078125

MASIV Accessible Area

129.02557526429655

MASIV Accessible Volume

105.91780468179142

The "VMD Accessible Area" value refers to VMD calculations while "MASIV Accessible Volume" and "MASIV Accessible Area" values belong to ***avolum.exe*** estimations.

References:

(1) Jmol: an open-source Java viewer for chemical structures in 3D http://jmol.sourceforge.net/ (accessed 2021 -03 -24).

(2) VMD Software Downloads https://www.ks.uiuc.edu/Development/Download/download.cgi?PackageName=VMD (accessed 2021 -07 -20).

(3) Bank, R. P. D. RCSB PDB - 5TWI: Alpha Helix Nucleation by a Simple Cyclic Tetrapeptide https://www.rcsb.org/structure/5TWI (accessed 2021 -07 -27).

(4) Gilbert, A. *Methane from Nutjunkie/IQmol*; 2021.

(5) Gilbert, A. *Ethane from Nutjunkie/IQmol*; 2021.