McVol - Calculating Protein Volume and Identifying Cavities in Proteins

1 Introduction

McVol is originally written to calculate the Van der Waals volume and molecular volume of proteins. The volume integration is solved by a Monte Carlo algorithm. Based on this integration, also internal cavities and surface clefts are identified an filled with water molecules. Additionally, a membrane of dummy atoms can be placed around the molecule.

2 File preparation

Two input files are necessary to run McVol. The first file is a pqr file of the protein. The format is like the pdb format, but following the three columns with the xyz coordinates, there is one column with the charge and the last column with the radius of the atom. It is important to check the radii of all atoms, since these radii are taken for all calculations within McVol. To prevent problems, atom radii larger than 5 Å will cause an error message to be displayed. This is mainly to avoid an exhaustive memory usage since the box dimensions are influenced by the maximal radius of all atoms. If an atom got a large radius, the box will grow extremely large and you will run out of memory soon.

The second file is the setup file, called like the pqr file, but with setup as file extension. So for example, these two files have to be named molname.pqr and molname.setup. The setup file contains all flags which are adjustable by the user. These are in detail:

- **nmc** Number of Monte Carlo steps per $Å^3$ of the box volume
- **surfPT** Number of surface points per atom used for dot surface calculations
- **probe** Probe sphere radius

- **membZmin** Minimum Z coordinate for the membrane creation
- **membZmax** Maximum Z coordinate for the membrane
- **startgridspacing** Grid spacing for the first cavity search, for the cleft search and the membrane creation
- **cavgridspacing** Grid spacing for the cavity refinement
- **minVol** Minimum volume for cavities. Cavities or clefts smaller than this volume are discarded.
- **waterVol** Volume of one water molecule
- **DummyRad** Radius of the dummy atoms placed as membrane
- **blab** Debug output level
- **MembDim** Thickness of the membrane in x/y direction
- **CoreZMin** Membrane Core region minimum Z coordinate
- **CoreZMax** Membrane Core region maximum Z coordinate
- **CoreDim** Allowed distance of membrane points not in core region
- **CleftDim** Box dimension for cleft search
- **CleftRel** Percentage of points which need to be cleft or protein to define a cleft point
- **CleftMethod** Method for cleft detection. Use 2!
- **SurfaceCluster** Maximal distance of two neighboring dot surface points. Neighboring criteria for clustering.

Please read the paper **??** to learn more about the theory of the program before you change the default values of these parameters.

Some of these variables still need some explanation.

The three variables starting with "**Core**" implement a feature for the membrane creation for protein channels. The membrane created by McVol tends to "flood" over the channel borders into the channel, since the channel interior can not be separated from the real cytoplasmic or ectoplasmic space. We will call this channel artifact from now on. We added a feature which should represent the headgroups of a lipid bilayer. The core region is therefore defined as the hydrophobic region of the bilayer. To reduce the channel artifact described above, one can define the core region as a part of the membrane. Therefore, **CoreZMin** should be larger than **MembZmin** and **CoreZMax** should be smaller than **membZmax**. If define correctly, this core region defines the real properties of the membrane and should not not be higher or lower as the channel borders. Points, which are in the two regions above or below the core regions are only defined as membrane, if they are at least **CoreDim** Å away from the closest point in the core region. All membrane parts flooding over the channel borders into the channel are removed from the membrane and marked as surface cleft. This reduces the channel artifact significantly.

The **CleftDim** variable defines the dimesion of the box which is placed on each surface point during cleft detection. If **CleftRel** percent of this box are protein or other cleft points, the surface point is marked as cleft point.

The **SurfaceCluster** variable is used during the clustering of the dot surface. Two surface dots are defined as neighboring and therefore as belonging to one graph, if they are less than **SurfaceCluster** Å apart from each other.

3 Running McVol

Running McVol is done by calling the binary and passing the molecule name as an argument.

```
McVol example
```

The two files described in the last section are named example.pqr and example.setup in this case. The output of the program containing the general informations and volumes of all cavities, clefts and the whole program are written to stdout. Atom representations of the cavities, clefts and the membrane are written to files in a directory **cav**. The output as well as the generated

files are explained in the next section.

4 Output

The output of the program starts with the variables you specified in the input. These are written out as read from the setup file for debugging purpose. The dot Surface calculation algorithm writes the exact number of surface dots used for this calculation. This number could differ from the number given as input parameter since the tessellation calculation secures an equal distribution of points on the atoms spheres and adjust the number of surface points given in the input. The total solvent accessible surface (SAS) of the protein follows in the output. This surface is the total SAS of the protein, also containing internal surfaces, which are the SASes of cavities. The detailed SASes of all internal cavities follow if **SurfaceCluster** is set to a value above 0. The SAS of cluster one is also the SAS of the protein without the surface of internal cavities.

The output of the Monte Carlo volume integration follows:

`Inside atoms`: Monte Carlo points placed inside atom volume

`Inside voids`: Monte Carlo points placed inside void volume

`Inside envelope`: Monte Carlo points placed inside the envelope volume

`Inside solvent`: Monte Carlo points placed inside the solvent volume (including envelope region)

`Molecular Volume`: Molecular Volume (VdW Volume + void volume)in $Å^3$

`Molecular Volume + Envelope Volume`: The volume enclosed by the solvent accessible Surface $Å^3$

`VdW Volume`: Van der Waals volume in $Å^3$

The cavity definition is done afterwards and the number of cavities found is printed. If a cleft search is asked (CleftMethod set to 2), the number of surface points defined as cleft points in each iteration is listed. The number of cavities including clefts is printed afterwards.

If a membrane creation is asked, the details about this creation are printed in the following. The reduction of atoms means the ratio between the membrane created by McVol and a membrane build as a rectangle box. Since McVol defines a membrane with the shape of the Protein with a given thickness, this reduces the number of dummy atoms necessary for this membrane.

Details about the cleft refinement follow and are of minor importance. Cavity volume is recalculated and the grid representation is refined, cavities with a volume smaller than the **minVol** are discarded.

Some checks for artifacts follow. If a cavity touches the border of the box it was enclosed during the refinemend calculation, something with the cavity definition went wrong. If we have defined overlapping cavities, these are detected and merged. From the Check for duplicates listing you can also see the final volume of each cavity or cleft. The file names for each cavity are also listed. Dot representations of the cavities are printed to these files. Be aware, that the numbering of the cavities is artificial and done by the order in which they are defined by the graph search algorithm. This numbering can vary between two runs of the program, since the Monte Carlo algorithm uses random numbers to place points inside the protein containing box. Cavity numbers are also not correlated to the numbering of the dot surface clusters. Finally, the number of water molecules placed in each cavity is listed.

Dummy atom representations of all cavities and clefts are written to the files listed in the output. All these files are in the cav directory. The dummy atom representation of the membrane is stored in the **memb.pqr** file. The **memb_rest.pqr** file contains atoms located in the protein sourrounding but discarded during the membrane creation. In fact this is the protein sourrounding without the membrane. In the **memb_cutof.pqr** file is everything which is separated from the membrane during the processing of the core region. If all Core parameters are set to 0, this is one slice at the top and bottom of the membrane. The water oxygen positions are written to the water.pdb file. All water molecules of all cavities are combined here. You will find two files starting with **surface_** in the main directory. In the **surface_clusters_surf.pdb** file, the dot surface of the protein is written as an atom dummy representation. Depending on the number of surface points you have chosen, these files could grow really big. The **surface_clusters_cav.pdb**

file contains the dot surface representations of the internal cavities, if **SurfaceCluster** is set to a value above 0. The residue number of the dummy atoms in **surface_clusters_cav.pdb** corresponds to the cluster number assigned in the output of the surface clustering. Therefore you can assign the SAS of each surface in **surface_clusters_cav.pdb**.