How To Use PACUPP

Pockets **A**nd **C**avities **U**sing **P**seudoatoms in **P**roteins

molviz.org/pacupp – by Eric Martz

The term **cavities** will be used to include pockets, tunnels, and channels. **Pockets** have a single mouth to the surface of the macromolecule, and **channels** have two or more mouths. A **tunnel** connects two or more locations, and may or may not have mouths. A cavity or tunnel that has no mouths will be described as **buried**.

Please report successes, concerns, bugs, or suggestions for improvement to m0lviz (at) yahoo dot com.

Quick Start

Very briefly, you load a macromolecule into Jmol, then drop the script **3-FILL-CAVITIES.spt** onto the molecule in Jmol. Animated example results.

In a bit more detail:

Download PACUPP.zip and unzip it.

Open these instructions by double clicking **How-To-Use-PACUPP.pdf** which is in Fill_Cavities_PACUPP/0-INSTRUCTIONS.

Demonstrations of how to use PACUPP are available as a video or a slideshow.

In the downloaded folder **Fill_Cavities_PACUPP:**

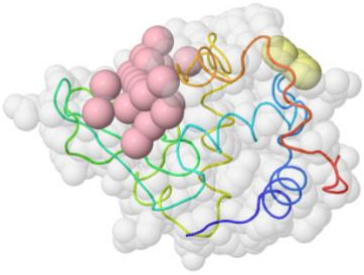
1. Double click **1-Jmol.jar**. If a black Jmol window does not appear, see 2-Load-a-Macromolecule.pdf.

2. Load a macromolecule PDB file into Jmol:
   - Drag a PDB file from your computer and drop it into the black window.
   - Or, to load from online, see 2-Load-a-Macromolecule.pdf
   - For more detailed instructions, see 2-Load-a-Macromolecule.pdf

3. After you see the macromolecule in Jmol, drag the file **3-FILL-CAVITIES.spt** and drop it onto the molecule.

If you choose the default settings, your job will proceed immediately. Or you can say *No* to the defaults, and customize the settings, following the explanations that will appear.

After your job finishes, you will see your macromolecule with its pockets and cavities represented by colored pseudoatoms. For example, at right is human lysozyme 1LZR after completion of PACUPP (using default settings). The **catalytic cleft pocket is pink**, and another smaller pocket is yellow. (Pockets with <5 pseudoatoms [PA] were hidden automatically. Hidden were 5 pockets with 2-4 PA, each and 7 with a single PA each.)



A message in the Jmol Script Console will invite you to use various PACUPP commands to visualize the results. Animated example results.

See also Jmol mouse gestures.

————————————————————

CONTENTS

———————————————————

Caution: Do not rename or reorganize folders or files in the downloaded folder Fill_Cavities_PACUPP. The PACUPP program depends on the original names and organization. You can add files (they will be ignored by PACUPP). Exceptions: you can rename files in the folders animation_kit, batches, pdb-files. Also you can rename output files as long as the four folders within output-files remain as named: pdb, pngj, reports, spreadsheet-ready-lining-lists.

Before running PACUPP:

Get familiar with your molecule. FirstGlance.Jmol.Org is strongly recommended because it tells you a great deal about your molecule, and guides you through a series of revealing views using plain language, and without knowing or using any Jmol commands. It has extensive help and examples for comparison. Before running PACUPP, you should know how many chains are in your model, which are protein and which are DNA or RNA, what ligands are present, etc. This will help you decide whether to specify any special moieties as *Inside Cavities* or *Outside Cavities*. For example, 3drf has a peptide ligand as chain B. By default, PACUPP puts protein outside of cavities. But if you want to see the cavity that contains this peptide, you need to specify 'chain=B' to be *Inside Cavities*.

Re-running Jobs

Once you have completed a PACUPP job, you can re-run PACUPP with different settings without closing and re-starting Jmol. Simply drag the file **3-FILL-CAVITIES.spt** and drop it into Jmol as many times as you want.

Similarly, after dropping a PACUPP output PDB file (or PNGJ file) into Jmol, you can run a new job on that macromolecule simply by dragging the file **3-FILL-CAVITIES.spt** and dropping it into Jmol, over and over.

If anything acts peculiar, then quit Jmol and restart Jmol.

Output PDB, PNGJ and Report Files

Each PACUPP job generates 3 files.

1. A **PDB** file containing cavity-filling pseudoatoms.
2. A **PNGJ** file containing cavity-filling pseudoatoms.
3. A **Report** listing details of the job.

These files are in the folder **output-files**, in its sub-folders *pdb* , *pngj*, and *reports*.

When PACUPP's output PDB file is **dropped into Jmol**, it is automatically rendered the same as it was at the end of the PACUPP job, and a report summarizing the job and help are displayed automatically. All PACUPP commands are available.

The PNGJ file could be shared with colleagues who are not using PACUPP. When the PNGJ file is dropped into Jmol, it will display PACUPP's initial view, which can be rotated with the mouse. Similarly, **you can save any PACUPP view to share as a PNGJ** with the command

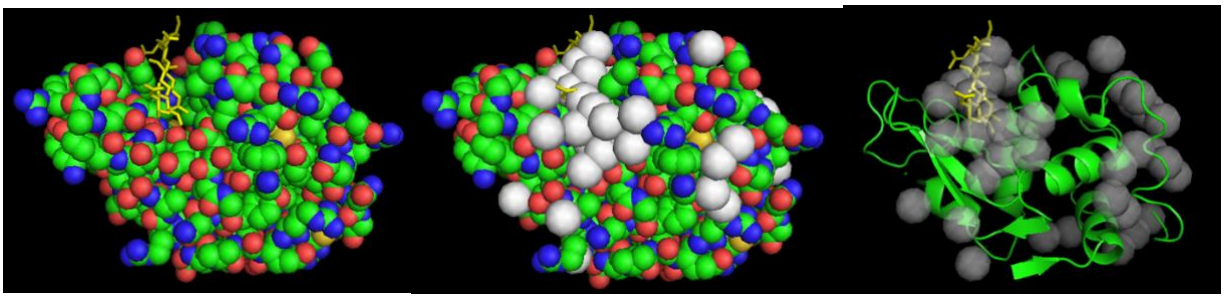
```
write filename.pngj
```

Using Jmol, PACUPP offers many convenient commands to explore and view the cavities it reveals (see below). However, PACUPP's output PDB file can be viewed in any molecular visualization software.

The cavity-filling pseudoatoms are element holmium (Ho; think "holes"). Each contiguous group of pseudoatoms is given a cavity number, which are the "sequence" numbers of the Ho atoms. Layer 1 (the surface layer of pseudoatoms) has occupancy 0.01. Deeper layers 2, 3, etc. have occupancies 0.02, 0.03, etc. Cavities with no surface exposure ("buried") have occupancy 0.00.

PyMOL

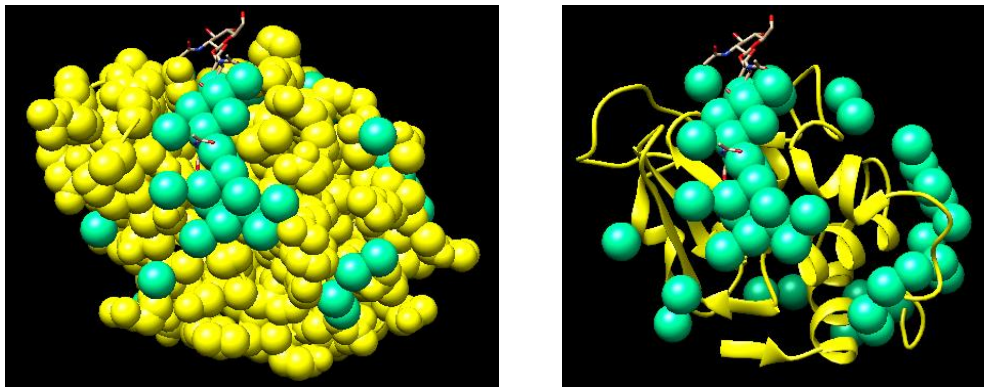
For example, here is PACUPP's PDB file for lysozyme (1LZR) in PyMOL. Substrate is yellow sticks. PACUPP's cavity-filling pseudoatoms are hidden at the left, opaque white in the middle, and translucent white at the right.



There are more pseudoatoms (white/gray) in these images than in the Jmol image above because, by default, PACUPP hides smaller clumps of pseudoatoms (in this case, it hid clumps with <5 pseudoatoms).

Chimera

Here is PACUPP's PDB file for lysozyme (1LZR) in UCSF Chimera. Cavity pseudoatoms are green. Protein is yellow. Substrate is rendered as sticks colored by element.



There are more pseudoatoms (white/gray) in these images than in the Jmol image above because, by default, PACUPP hides smaller clumps of pseudoatoms (in this case, it hid clumps with <5 pseudoatoms).

PACUPP Commands

When a PACUPP command is mentioned below for the first time, it is highlighted in yellow, like this: **v** (opens the **V**iews Menu). Commands are entered in the Jmol Script Console window (see 2-Load-a-Macromolecule.pdf for how to open this window). Complete lists of commands are available in these documents (all in the folder 0-INSTRUCTIONS):

- command-details.pdf
- commands-alphabetic.pdf

Animations for Slides

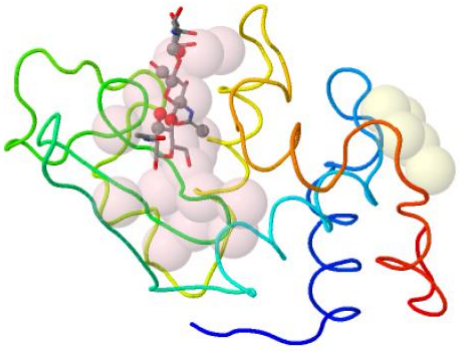
PACUPP includes an animation kit. Any PACUPP view can be easily rendered as a movie (GIF format) file that can be dropped into a presentation slide. Here is a slideshow with Animated example results. (If that link to Google Slides is not working, you can download the slideshow as a ~100 MB PowerPoint file.)

Views

Instructions for visualizing the pockets and cavities in various ways are displayed automatically by PACUPP. Entering the command **v** (for Views) displays a menu with many options, including to show the atoms inside (contents of) each pocket or cavity.

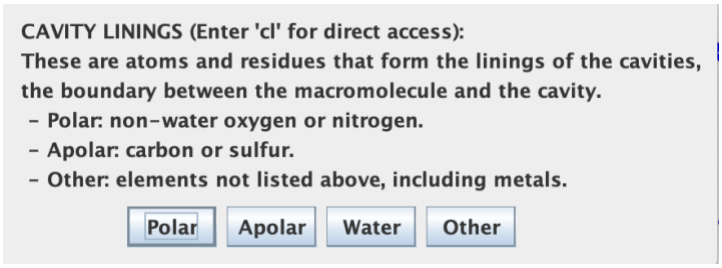
Atoms Inside Cavities

At right you see the **Inside Cavities** (cavity contents) view for lysozyme, showing the substrate in the pink pocket.

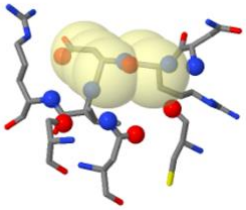


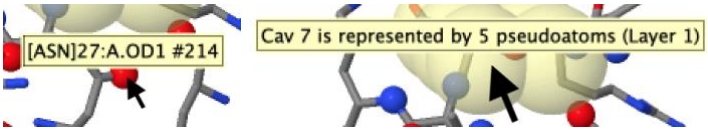
Cavity Lining Atoms

Other views show atoms clashing with the pseudoatoms (not shown here), or atoms outside a cavity that define its surface, **Cavity Linings**.



As a simple example, the polar contacts to the smaller pocket in lysozyme are shown at right. Balls are the atoms that line the cavity; sticks are for context. Touching an atom pops up its identification





while clicking a protein atom gives a more detailed report:

[ASN]27:A.OD1 (:A means Chain A; OD means Oxygen Delta=4th)

SPREADSHEET READY Lists of Cavity Lining Atoms

After displaying the polar or apolar macromolecule atoms that line a cavity, the command **L** **Lists** those **L**ining atoms, and writes them into a **spreadsheet-ready file**. (Lower case **l** also works.) Here is the list for the above example:

Atoms defining the lining of cavity 7 for
3drf_cavities_fine_small_3_5_20_24.
11 POLAR macromolecule atoms within
5 Å of the pseudoatoms:

Atom	AltLoc	Res	SeqNo	Chain
OG		SER	24	A
OD1		ASN	27	A
ND2		ASN	27	A
O		CYS	116	A
N		ASN	118	A
N		ARG	119	A
O		ARG	119	A
N		ASP	120	A
OD1		ASP	120	A
N		VAL	121	A
N		ARG	122	A

Writing list to
output-files/spreadsheet-ready-lining-lists/ 3drf_cavities_fine_small_3_5_20_24_cav7_POLAR_list.txt
This file is SPREADSHEET READY!

The list file can be dropped onto the Excel icon to create a spreadsheet.

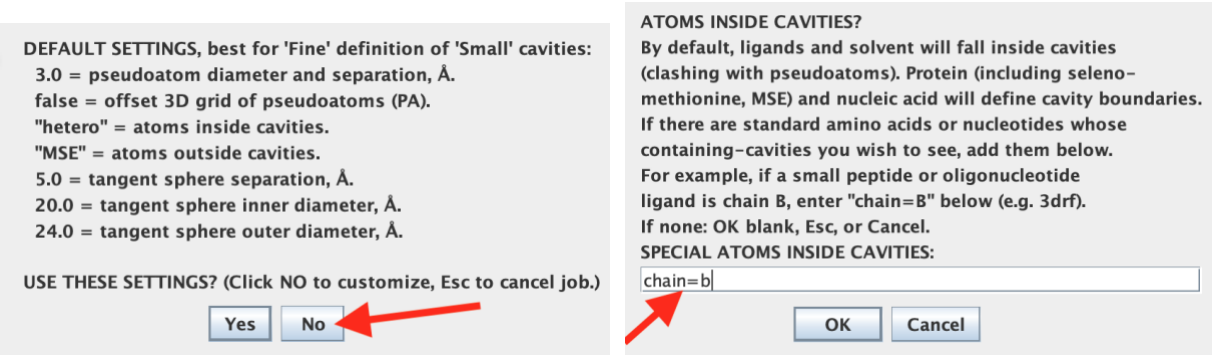
Cavity Mouths

The *Cavity Mouths* view emphasizes the mouths of cavities. It also lets you see that the pocket-filling pseudoatoms don't bulge above the protein surface.

As an example, consider oligopeptide-binding protein A from *Lactococcus lactis* (3drf). This protein is about 80 kDa, about 600 amino acids. It has, in its closed conformation, a buried cavity that promiscuously binds peptides. 3drf contains an **octapeptide in the cavity**.

Customizing Atoms Inside Cavities

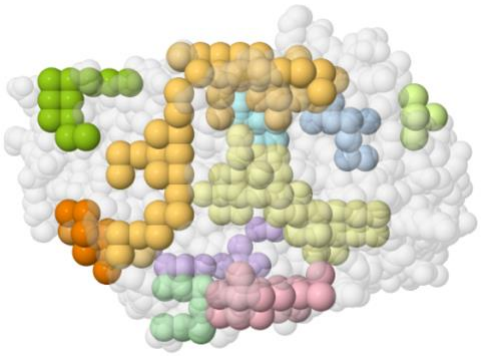
In order to include this peptide within the pseudoatoms representing this cavity, when setting up the PACUPP job, one must specify "chain=b" as "Inside Cavities".



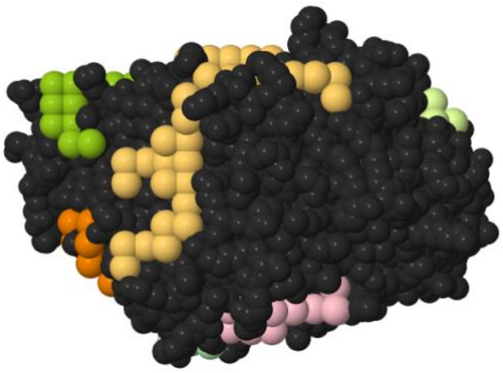
(A similar dialog lets you specify atoms to be *Outside* of cavities. By default, protein and nucleic acid are outside, defining cavity boundaries. Because seleno-methionine is often used to solve crystallographic phases, MSE is also *Outside* cavities by default. As an example, when exploring cytochromes [e.g. 3oue], if one wishes to see pockets excluding the spaces occupied by 'hetero' hemes, one can specify HEM to be *Outside* cavities.)

Here are the 10 discrete cavities (discrete clumps of contiguous pseudoatoms [PA]) that have 5 or more PA/cavity for 3drf (default PACUPP settings). The yellow cavity is buried. That means that no PA on the surface of the macromolecule touches any PA in the yellow cavity. It means that any mouth that the yellow cavity has to the surface is not much wider than a single PA, and that no PA happened to fit into any such mouth. By rotating the Mouths View, one can look for openings to the yellow cavity.

[Continued on next page.]

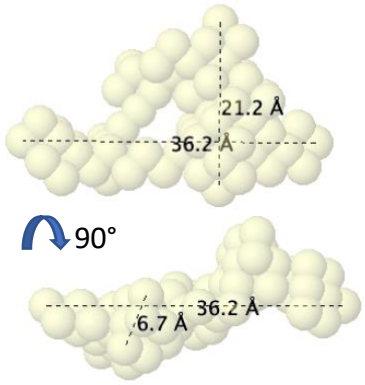


Protein: light gray translucent atoms. Cavities are each a distinct color.



Mouths View: Protein is dark gray. Cavities are each a distinct color.

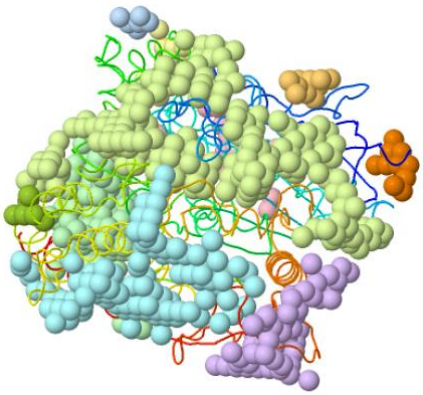
Measuring Cavities



When you click the button *Measure Cavities* in the Views menu (shown above), the visible cavities become translucent. This makes it easy to double-click on each of two atoms to display the distance between them. (When the PA are opaque, you can't see the displayed distance because it is typically inside the PA.) At right are measurements for the yellow buried cavity in 3drf.

Add the clash distance to get the outside dimensions: PA are created wherever they don't clash with the macromolecule. The measurements shown are atom center-to-center distances. To get the outside size of the cavity, add half of the clash distance to each end of the measured distance; that is, add the clash distance to the measurement. The clash distance is shown in the report.txt file near the bottom, and near the top of the messages generated when you drop the output PDB file into Jmol, under *Settings*. For the example here (default settings), it is 3.0 Å. So the outside longest dimension in this example is 36.2 + 3.0 = 39.2 Å.

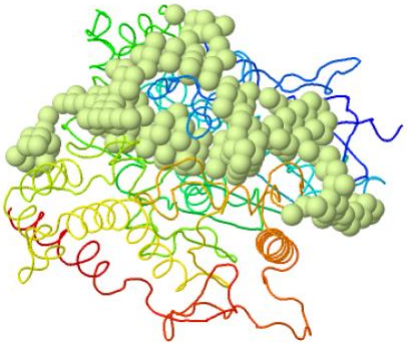
Isolating Cavities of Interest



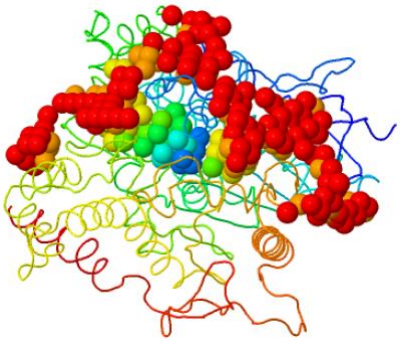
At right are the cavities in acetylcholinesterase (1EA5) of the Pacific electric ray (Torpedo californica; cavity detail: very fine; cavity size: small). Cavities with <5 pseudoatoms (PA) are hidden automatically; this leaves 10 discrete cavities (clumps of PA) visible. The one of interest is light green, because (as you will see) it includes the deep catalytic pocket (gorge).

Click a Cavity to Isolate It

Simply click on the light green cavity to isolate it. (Click on it again to toggle back all the other cavities.)



Color by Depth



Next it will be useful to color the PA by depth from the surface. The command for this is **d** (for Depth). **Surface PA are red**, and the **deepest PA are blue**, with intermediate layers in a spectral sequence. The deep catalytic pocket is now discernable as the yellow, green and blue PA.

Hiding the Macromolecule

The command **m** (for MacroMolecule) toggles visibility of the translucent gray atoms of the macromolecule. These were already hidden in some of the 1EA5 images above. The command **b** (for BackBone) toggles visibility of the colored backbone trace.

Trimming A Cavity

PACUPP provides two sets of tools for trimming cavities to remove unwanted fenestrations or connections to shallow surface pockets.

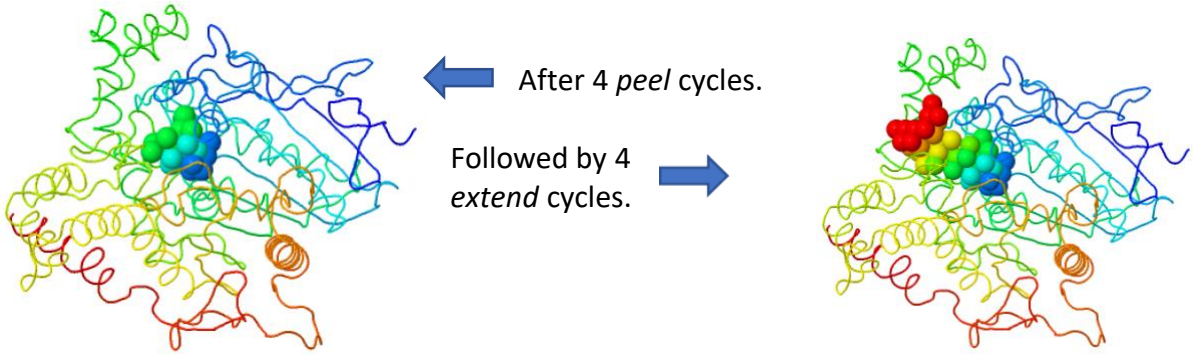
Undo

As you use either of these methods, if you make a mistake, Jmol provides a comprehensive undo button that will undo the last step, or multiple steps. It is in a row buttons at the bottom of Jmol's Script Console window (the window where PACUPP reports are displayed, and where you enter commands):

Peel Away the Surface

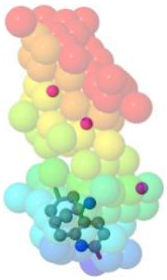
As we have already seen above, the mouth of the catalytic pocket of acetylcholinesterase connects with a sprawling confluence of shallow pockets (**red**, image above right). We can peel

these away by repeatedly entering the command **p** (for Peel). We can then extend back out by repeatedly entering the command **e** (for Extend).



Extending back out from the peeled core **does not restore the sprawling confluence** because each "extend" cycle adds only the PA in the next outer layer that *contact already visible PA*. Thus, in the last "extend" cycle, the red PA added are only those that contact already visible orange PA. In this case, that result was pleasing. In other cases, one might "fatten" the cavity (see the **f** and **f1** commands in command-details.pdf).

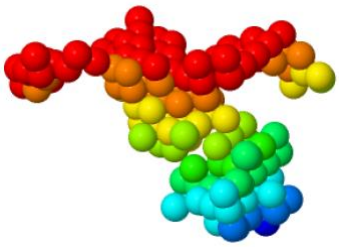
Now that we have a pleasing representation of the catalytic pocket, let's see what is inside. We'll use 1vot for this because there is nothing but water in the catalytic gorge of 1ea5.



After processing 1vot (cavity detail: extra fine, cavity size: small), hiding the macromolecule (mm) and backbone (bb), coloring by depth (d), isolating the catalytic gorge cavity by clicking on it, peeling (p), extending back out (e), and fattening it (f) plus one cycle of layer 1 fattening (f1), in the Views menu (v), *Inside Cavities* was clicked, and we see the inhibitor **huperzine** inside the deep catalytic pocket, along with 3 water oxygens.

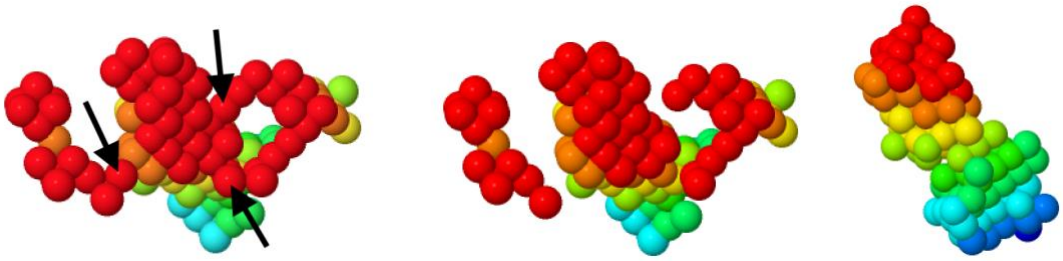
Trimming Pseudoatoms by Clicking

The Peel/Extend tools are one way to trim a cavity. Another way provided by PACUPP is **clicking on pseudoatoms to hide them**. To demonstrate this, we'll again use acetylcholinesterase **1vot**. The initial default settings were declined so that different levels of cavity detail could be tried (see next section). *Extra fine* detail gave the best result.



Then, as above, the macromolecule and backbone were hidden, and the pseudoatoms colored by depth. This made it easy to see which of many cavity entrances on the surface extends deeply into the protein (blue pseudoatoms), and clicking on that region isolated the view at right. The two red "wings" are shallow surface pockets.

If we want to hide the wings, the first step is to **disconnect** them from the main catalytic gorge. After entering the command **a** (Atom), each pseudoatom you click will be hidden. In this case, hiding 4 pseudoatoms (black arrows) disconnects both wings. After entering the command **k**, clicking on a pseudoatom hides the entire **K**luster of pseudoatoms that contacts it.



With the cavity isolated and trimmed, we can now enter **v** (Views) to see what is inside the cavity (as above), see and list the macromolecule atoms lining the cavity, examine the mouth of the cavity, and measure the dimensions of the cavity.

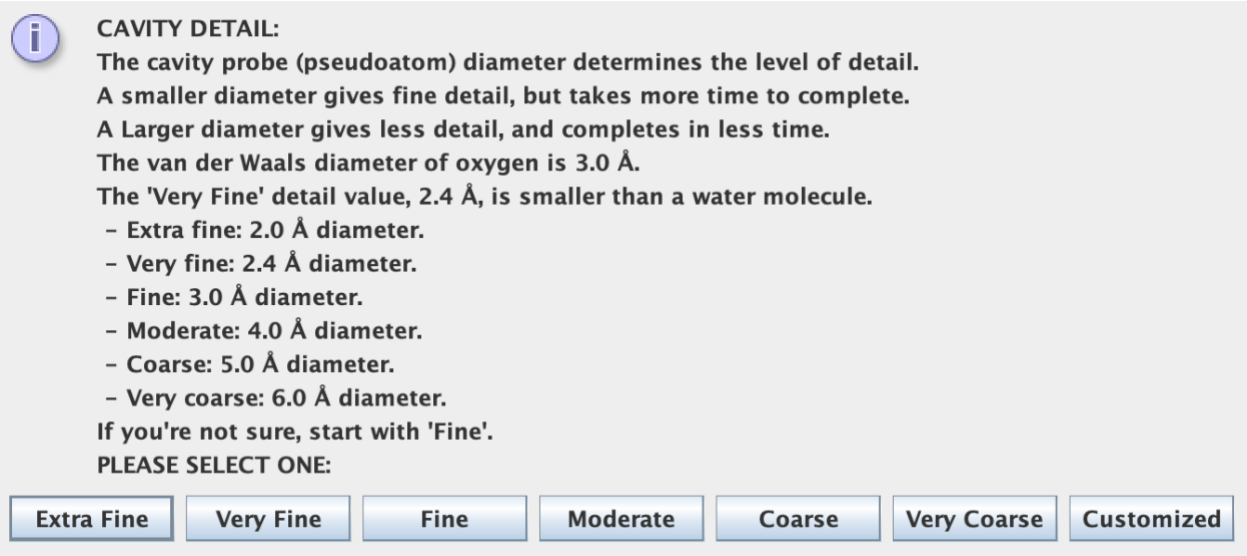
Saving Trimmed Cavities in PNGJ Files

After you have trimmed a cavity, it may be useful to save the trimmed view in a PNGJ file. When a PNGJ file is dropped into Jmol, you will see the exact view present at the time the file was saved, but you can rotate the molecule and modify the view. Furthermore, if you drop the file menu/menu.spt onto the view in Jmol, you will have all of PACUPP's commands available. You can save a PNGJ file whenever you wish by entering this command into Jmol:

write filename.pngj

The file will be written into the folder containing Jmol.jar. You can then move it to a suitable location.

Cavity Detail



Generally speaking, too much detail may reveal tiny crevices connecting interior cavities to the surface -- crevices that may seem too small to be functionally important. (Of course it is important to bear in mind that a static crystallographic or cryo-EM structure fails to represent thermal motion of proteins. Tiny channels or areas with no PA might form enlarged openings periodically. For example, recall that crystal structures of hemoglobin have no opening large enough to explain how molecular oxygen rapidly binds to and is released from the buried heme iron.)

As an example, again consider oligopeptide-binding protein A from *Lactococcus lactis* (3drf). Above are images from default PACUPP settings (cavity detail: fine), which show a buried cavity that has no pseudoatom "mouths" on the macromolecular surface. When cavity detail is set to *Very Fine*, a single PA in the interior cavity touches a single PA in a nearby surface pocket, so the interior cavity is no longer strictly "buried". With cavity detail set to *Extra Fine*, additional minor contacts with surface PA occur.

Reducing cavity detail excludes minor crevices radiating out from a major cavity, and also reduces the detail of the cavity surface.

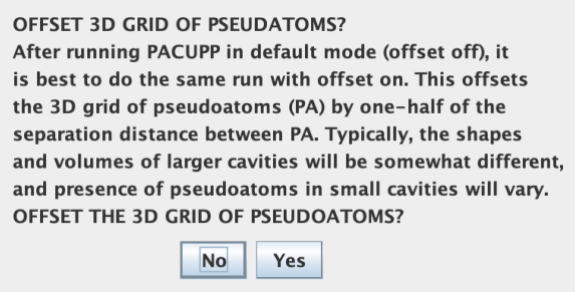
Offset: Hit or Miss & Cavity Volume

Whenever concerned about the volume of a cavity, or the existence of a very small cavity, it is important to do **two jobs with all settings identical, except** one with **offset true**, and one with **offset false**. Below is explained why.

Detection of tiny cavities is "hit or miss".

When cavities or pockets are no larger than a few PA, whether or not PACUPP will create PA to represent them depends on the accident of how the 3D grid of PA aligns with the cavity boundaries. It is possible for all PA in the vicinity to clash, so no PA are created to represent a small cavity. Equally, one or more PA may happen to avoid clashes, and thus be created to represent the cavity. Thus, **detection of very small cavities**, the size of a few atoms, is "hit or miss".

The 3D grid of PA starts at a corner of the boundbox (the corner with the lowest values of X, Y, and Z), and potential PA are positioned every PA diameter in the X, Y, and Z directions (to fill the boundbox). PA are created at those grid positions only when they don't clash with the macromolecule. The **offset = true** option starts the 3D array half a PA diameter offset



from that corner, in the negative X, Y, and Z directions.

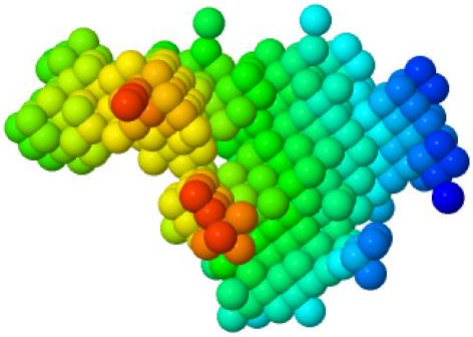
An example occurs when 3drf is run with cavity detail: very fine, cavity size: small, and offset = false. A tiny buried cavity of **3 PA** occurs where the sidechain of Asn451 defines part of the cavity boundary. When the job is re-run with the only change being offset = true, **no PA** are created near Asn451 because all of the 3D grid positions in the vicinity clash with protein atoms.

Similarly for 3drf (detail: very fine), with offset = false a seven PA cavity occurs with boundaries being defined in part by the sidechains of Asp70, Thr125, and Leu200. When re-run with offset = true, the largest cluster of PA near those sidechains has only 2 PA.

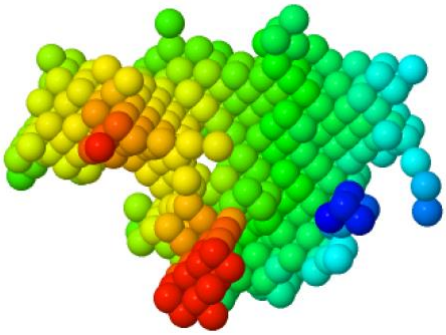
Uncertainty in Cavity Volumes

The same "hit or miss" character of how the 3D grid of pseudoatoms aligns with cavity boundaries affects estimation of the volume of a cavity. PACUPP estimates the volume as the volume of a cube the diameter of a pseudoatom times the number of PA in the cavity cluster. For example, if the PA diameter is 3.0 Å (default, cavity detail: fine), the volume of a cube of that diameter is 3 x 3 x 3 = 27 $Å^3$. The volume of a cavity represented by 10 PA would thus be estimated as approximately 270 $Å^3$.

There is usually a small difference in volumes estimated with offset true vs. offset false. Sometimes the difference is > 5%. Such a case occurs with 3drf, cavity detail: extra fine, chain B inside cavities, cavity size: small. Offset false vs. true makes an 8% difference in the estimate of the cavity volume, and substantial differences in the shape of the PA cluster.

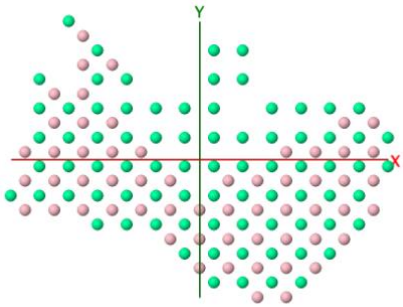


Offset false.
460 PA of diameter 2.0 Å.
Volume: 3,680 $Å^3$.



Offset true.
497 PA of diameter 2.0 Å.
Volume: 3,976 $Å^3$.

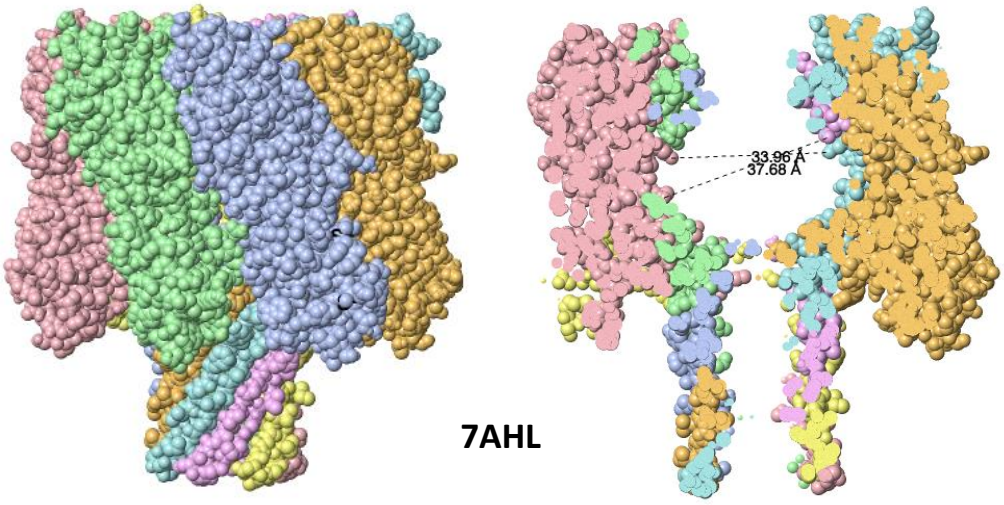
The image at right shows the PA grid positions that don't clash with the macromolecule for the major interior cavity of 3drf, when the Z axis is perpendicular to the screen. **Pink** positions are with **offset false**, and **green positions** are with **offset true**. There are more non-clashing positions with the latter (green).



Cavity Size

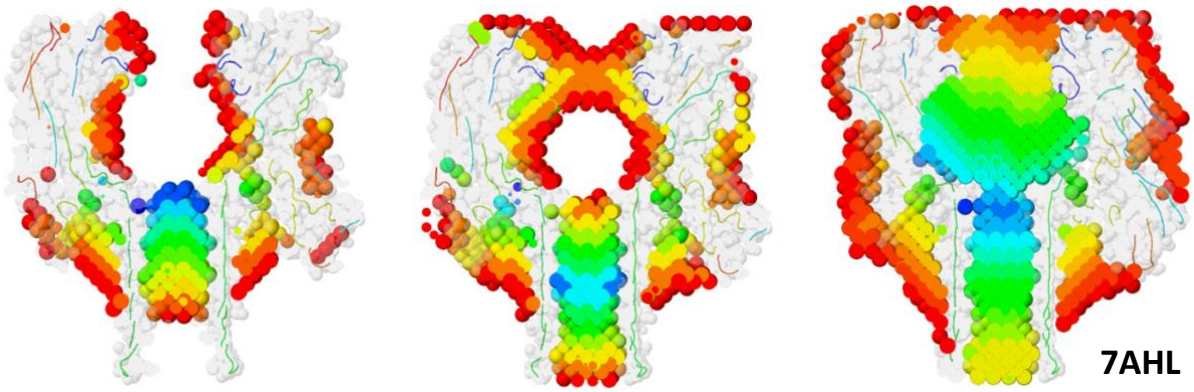
PACUPP uses T*angent Spheres* to define a smoothed surface of the macromolecule (see Methods). PA that are **inside the smoothed surface**, and that **don't clash** with the macromolecule, define the pockets and cavities. The default diameter of the tangent spheres is 20 Å. When a cavity is large enough that tangent spheres can fit inside it, its inner surface becomes part of the macromolecular surface, so the PA filling it are **hollow**. In that case, increasing the diameter of the tangent spheres produces a **solid** filling of the large cavity. Increasing the tangent sphere diameter also makes a smoother surface, which fills the pockets with more PA.

Consider *Staphlococcal* hemolysin 7ahl, a homo-heptamer "mushroom" with a transmembrane stalk (hydrophobic surface) and a large channel.



7AHL

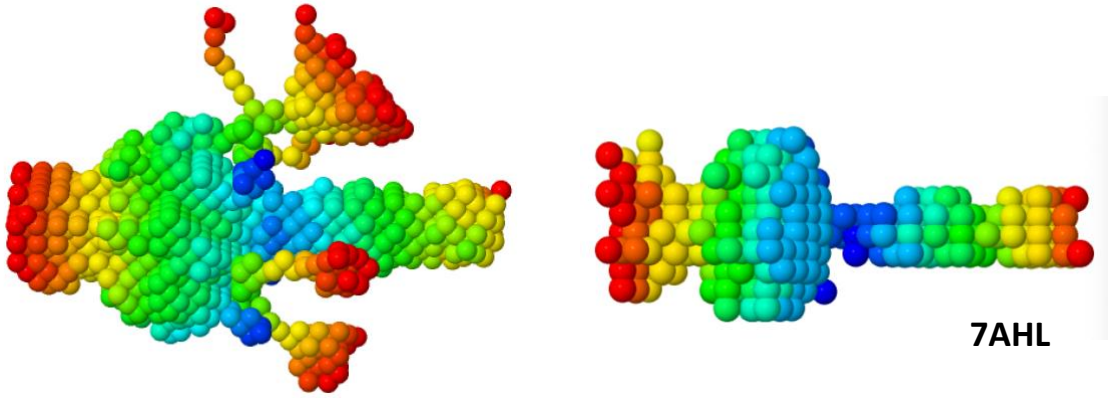
The widest part of the channel is about 35 Å in diameter. Here are PACUPP results for 7ahl shown as slabs of 6% thickness (PACUPP command **s**) with PA colored by depth (red=surface, blue=deepest).



Cavity detail: Fine Cavity size: **Small**	Cavity detail: Fine Cavity size: **Medium**	Cavity detail: Fine Cavity size: **Large**

PACUPP's preset cavity size "large" uses a tangent sphere inner dimeter of 44 Å. But in the case of 3KX9 (**empty ferritin**), this is **not large enough** to prevent the pseudoatoms filling the main cavity (which has a diameter of ~60 Å) from being hollow. Even a tangent sphere inner diameter of 50 Å produces a hollow filling. A tangent sphere inner diameter of **60 Å produces a solid filling**. By choosing to **customize** settings, you can specify a suitable tangent sphere inner diameter.

With **Fine** cavity detail (PA diameter 3.0 Å), many meandering small tunnels connect to the major central channel of 7AHL. The result below (left) was peeled, and then extended back out to eliminate many connections to the meandering confluence of surface pockets. Reducing cavity detail to **Moderate** (PA diameter 4.0 Å), a simplified representation emerges. Peeling was unnecessary. Simply clicking on the central channel to isolate it gave the result below (right).



Cavity detail: **Fine** Cavity size: Large "Peeled" & "Extended"	Cavity detail: **Moderate** Cavity size: Large "Peeling" unnecessary.

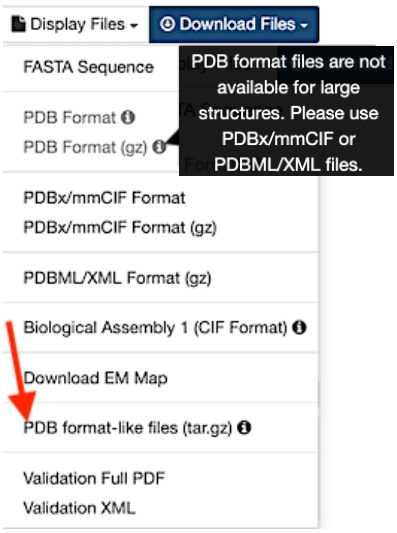
Biological Assemblies

When the PDB file contains specifications for a biological assembly (biological unit, biounit, probable quaternary structure) different from the asymmetric unit, PACUPP **alerts the user** and recommends processing the biological assembly. A link is provided that enables downloading the biological assembly in the most useful format. (Biological assemblies downloaded from the RCSB Protein Data Bank are problematic: duplicated chains all have the same name, and are separated into multiple models. It is much more practical for each chain to have a unique name and for all chains to be in a single model.) When exploring a PDB entry in FirstGlance in Jmol you will be offered a link to view or download this most useful format of the biological assembly (example: 1hho).

Large PDB Files & Batch Processing

PDB format, not mmCIF

The Protein Data Bank uses mmCIF as its primary data format. However, due to its popularity, the much older and more restrictive PDB format is offered whenever possible.



PACUPP can process **only PDB format**. If you load an mmCIF file, PACUPP will explain that it cannot process it. About 1% of entries in the Protein Data Bank are too large for the PDB format (>99,999 atoms or >62 chains). However, subsets of these huge models are available from RCSB.org to download in PDB format. See the Ribosome example. (Jmol is mmCIF capable. It would likely be possible to extend PACUPP to mmCIF format, but would likely take more than a week's work, as the mmCIF format is more complicated than the PDB format.)

Batch Processing

PACUPP has batch processing. **Examples/templates are in the folder *batches*.** This means you can specify, in a batch Jmol script file, multiple sets of parameters for a single PDB file, or multiple PDB files, each with one or more sets of parameters. Batch files run without user intervention. Before execution begins, the commands in the batch file are checked for syntactic validity, and PDB files specified are checked for availability. Thus, if there is a typographical error in a command or a PDB code, you can fix it before you walk away to let the bactch file process. You will need a plain text editor to configure batch files.

In macOS, with batch files that take hours, it is crucial to prevent the computer from going to sleep. Open Terminal and enter the command **caffeinate**. There will be no report in Terminal,

but this keeps the computer awake until that Terminal window is closed, or the caffeinate process is terminated.

Processing Time

Batch files are useful when you want to process larger numbers of PDB files, many sets of parameters, or very large PDB files that take a long time to process. You can start the batch, then go to bed or do something else while it processes. PACUPP completes most PDB files in less than minute. Extremely large PDB files with fine cavity detail can take nearly an hour.

PDB	Molecule	Non-Hydrogen Atoms	Cavity		**PACUPP Execution Time***
			Detail	Size	
1LZR	Lysozyme	1,251	Fine	Small	**0.05 min (3 sec)**
1EA5	Acetylcholinesterase	5,156†	Fine	Small	**0.1 min (10 sec)**
Two-thirds of entries in the Protein Data Bank have < 5,500 non-H atoms.					
3HYC	Phosphatase	10,767	Fine	Medium¶	**0.6 min (34 sec)**
7AHL	Strep hemolysin	17,207‡	Fine	Large¶	**1.3 min**
1PMA	Proteasome	45,892 §	Fine	Medium	**9 min**
"	"	"	Moderate	Large¶	**4.3 min**
5NP6	50S Ribosome (PDB)	94,519	Coarse	Small	**23.5 min**
3J9Q	Contractile nanotube	99,648 Δ	Fine	Small	**54 min**
"	"	"	Moderate	Medium	**19 min**
"	"	"	Moderate	Large¶	**15.7 min**

* On a mid-2014 MacBook Pro (2.2 GHz Intel Core i7).
† Among the largest third of entries in the Protein Data Bank. Median entry has 3,500 non-H atoms.
‡ Among the largest 6% entries in the Protein Data Bank. 13,000 is largest 90%.
¶ Smallest cavity size that avoided hollow cavities.
§ Among the largest 2% of entries in the Protein Data Bank.
Δ Among the largest 0.7% of entries in the Protein Data Bank (including mmCIF entries). 99,999 atoms are the maximum that can fit in the PDB data format.

PDB File Header Is Preserved

In the PDB file format, all lines preceding the first ATOM line are called the **"header"**. In fact, the very first line of a PDB file has record type HEADER (always a single line), but the term "header" is usually used to refer to all lines before the first ATOM line. The header lines are numerous. For example, 1LZR (lysozyme) has 304 header lines, 1029 ATOM lines, 223 HETATM lines (N-acetyl glucosamine and water), and 56 other lines (ignored by Jmol, mostly CONECT lines). A very large PDB file, 3j9q, has 2,429 header lines and 99,695 ATOM lines (no HETATM and thus no CONECT lines).

Jmol can write PDB files (write filename.pdb), and this is handy because Jmol includes only the currently selected atoms in the file it writes. However, PDB files written by Jmol contain only

ATOM records with **no header whatsoever**. Such files can be processed by PACUPP. This could be used to extract a single model from a multiple-model PDB files. The command to select a single model is e.g. select model=11.

Because the header contains a great deal of useful information, **PACUPP preserves the entire header**, when present in the initially loaded file, in its output PDB and PNGJ files. In fact, those files differ from the originals only in the addition of a block of REMARK Jmol commands inserted into the header, and a block of pseudoatoms (element holmium) ATOM records added to the end.

Hydrogen Atoms Ignored

PACUPP ignores hydrogen atoms when present in the model. That is, they are ignored when placing PA so as to avoid clashes with the macromolecule, and they are not shown in any PACUPP Views. However, they are preserved in the output PDB and PNGJ files in order to be faithful to the original.

Multiple Models

There are ~12,000 multiple-model entries in the Protein Data Bank. Most of those are NMR results, but there are more than 100 X-ray crystallographic results with multiple models. When there are multiple models (e.g. 2bbn, 21 NMR models), PACUPP processes **only the first model**. If you want to process another model, you must use a text editor to create a PDB file containing only that model. (If you use Jmol's command to load, for example, only model 7 from 2bbn, load =2bbn 7, the results cause technical problems and PACUPP will reject the job, explaining how to prepare model 7.)

Alternate Locations

About 40% of the entries in the Protein Data Bank specify multiple (alternate) locations for some atoms. For example, ~3% of the atoms in 3drf, shown above, have alternate locations with occupancies of 40-60%. (FirstGlance in Jmol provides in-depth information about alternate locations and occupancies, including spreadsheet-ready tabulations.)

PACUPP excludes alternate locations from its cavity detection. That is, atoms in alternate locations are ignored when placing PA that don't clash with the macromolecule. Also, the various Views offered by PACUPP hide atoms in alternate locations. However, PACUPP's output PDB and PNGJ files preserve the alternate locations in order to be faithful to the original entry.

Examples

Further examples are animated in the slideshow PACUPP Sample Results. If you cannot view the slideshow from that link to Google Slides, you can download a 100 MB PowerPoint file.

Many of the design specifications have already been described above. In addition, the following situations have been coded for and tested. If you find any misbehavior or bugs, please send details, sufficient to reproduce the problem, to m0lviz (at) yahoo.com.

1. The loaded model **must be a PDB file**. It must contain at least one ATOM record for element C, O, N, or P. CIF files are rejected.
2. The PDB file **need not have any header**. The first line can be an ATOM line, as it will be when selected atoms are written into a PDB file from Jmol. When a header is present, it is preserved in the output PDB and PNGJ files.
3. A **PDB ID code** is sought as follows:
    a. From the end of the first line, if its field type is HEADER.
    b. From the end of the first line if it is a REMARK from MakeMultimer.py (Biological Assembly).
    c. The first 4 characters of the loaded filename, if they obey PDB ID syntax (numeral followed by 3 alphanumeric characters).
    d. If none of the above succeed, the PDB ID is blank.
4. The **output filenames** (PDB, PNGJ, and report) begin with the full name of the loaded PDB file (excluding .pdb). Ideally, the name of the loaded PDB file begins with the PDB ID code. If not, and if the PDB ID code was determined, it is prepended onto the output filenames.

See also the separate document **PACUPP Methods**. There, the steps in a PACUPP job are explained in the order in which they are executed.

Please report successes, concerns, bugs, or suggestions for improvement to m0lviz (at) yahoo dot com.