

MOVIE MORPHING AND MAKING USING LSQMAN, PYMOL, AND QUICKTIME PRO
by Logan Ahlstrom (edits by Sue)

PREPARING A PDB FILE FOR MORPHING WITH LSQMAN

First, we need to edit the .pdb files. This is the most tedious part, but must be done correctly.

1) Replace any HETATM designations with ATOM

```
vi 1X8P.pdb
:1,$s/HETATM/ATOM / The two spaces after ATOM are important
:wq 1X8P_II.pdb
:q!
```

2) Remove anisotropic Bs (get Unix abbrev.....) temperature lines

```
cat 1X8P_II.pdb | grep -v ANIS > 1X8P_III.pdb
```

3) Remove unwanted CONECT records

For example, "CONECT 1674 1631" (This connects the heme to a water molecule in original .pdb file. It must be removed, or else the morphing will reveal unwanted and progressively lengthening bonds.

4) Remove the LINK records.

These may be found just before the atomic coordinates are listed. For example,
LINK FE HEM 185 NE2 HIS A 59

5) Remove unwanted waters.

Simple... just delete them in the text editor.

6) Remove alternate conformations.

Should not be too many... probably easiest just to do this by hand.

Or use `ccp4i` to remove alternate conformations

7) Add-in the molecules you want at the positions you want.

This will require some work. For example, the 1X8P.pdb file of NP4 contains water bound to the heme, but we need to add an NO outside of the protein so it will move into binding position as the movie continues. So, using PyMol you can use a guess-and-check method to come-up with some good x,y,z coordinates to place the NO. This is tricky because you have to make sure that the moving molecules don't pass through, not through each other, any residue side chains, etc.. Just play with the coordinates a bit.

Once you have discovered some reasonable coordinates, append these into the most recent .pdb file. For instance:

```
ATOM 2000 N NO B 186 25.410 30.406 19.162 1.00 5.47 N
ATOM 2001 O NO B 186 24.936 29.951 18.295 1.00 9.30 O
ATOM 2002 O HOH C 187 11.010 19.129 34.726 1.00 10.89 O
ATOM 2003 O HOH C 188 7.402 18.626 33.536 0.60 16.72 O
ATOM 2004 O HOH C 189 9.292 16.225 31.133 1.00 14.29 O
ATOM 2005 O HOH C 190 12.367 17.646 31.810 1.00 45.23 O
ATOM 2006 N HSM D 191 30.904 28.356 22.236 1.00 25.17 N
ATOM 2007 CA HSM D 191 30.275 28.706 23.525 1.00 18.93 C
ATOM 2008 CB HSM D 191 29.638 30.077 23.476 1.00 20.26 C
ATOM 2009 CG HSM D 191 28.866 30.358 24.778 1.00 16.16 C
ATOM 2010 ND1 HSM D 191 27.588 29.825 24.964 1.00 12.39 N
ATOM 2011 CD2 HSM D 191 29.122 31.173 25.867 1.00 14.58 C
ATOM 2012 CE1 HSM D 191 27.132 30.300 26.114 1.00 17.20 C
ATOM 2013 NE2 HSM D 191 28.037 31.112 26.681 1.00 12.35 N
```

Notice how the NO is specified as chain B, the waters as chain C, and the histamine (HSM) as chain D. Also, to make life a bit more simple (although it is not essential) I changed the atom numbers of the these chains starting from 2000. Save these .pdb files as something like 1x8P_good.pdb.

I suggest superimposing the pdb files before continuing to morphing using coot, SSM, lsqkab or whatever program you prefer. If you have more than one segment in your movie, this will minimize jumping between segments.

Once the above alterations have been made to the .pdb files, you are finally ready to morph.

MORPHING

I used LSQMAN (Uppsala Software Factory) to accomplish the task of morphing one .pdb file into another. See http://xray.bmc.uu.se/usf/lsqman_man.html for the manual. Before I give the script, I would like to outline, conceptually, thinking behind the process.

I had three basic stages of my morphing objectives:

- 1) NO comes into pocket, replacing the water that was bound
- 2) HSM comes into the pocket (with a water molecule), replacing the NO that was bound
- 3) The remaining water molecules come back into the pocket, replacing the HSM that was bound.

So, this was really a cyclic process that would require three .pdb files, (bound_H2O).pdb, (bound_NO).pdb, (bound_HSM).pdb. Thus, three different morphings, representing the three steps above, would be required.

In the shell under your home directory (or wherever LSQMAN was installed), run the following:

```
lx_lsqman
#Read in the two files
#The files do not have to be in your working directory; you may even just drag-and-drop them into the appropriate
#space in the shell window.
re m1 1X8P.pdb
re m2 1X80.pdb
#Tell LSQMAN to use all atoms (however, this excludes hydrogen atoms)
at no
#Fix silly nomenclature problems (this just makes sure that there are no problems reading the .pdbs, which can be in
#various forms)
nomen m1
nomen m2
#Next we will specify the atoms/chains present in our system
fix
m1
A1-185 B186 C187-190 D191
m2
A1-185 B186 C187-190 D191
strict
seq
torsion
#Morph between the two structures
#IMPORTANT NOTE!!! It is easiest to just specify each part of your system as a separate chain because you may
#simply call upon the chain designations to move different atoms/molecules at different stages of a morphing.
#Remember in the starting .pdb files for NP4 we specified residues 1-185 as chain A, 186 (the NO) as chain B, 187-190
#(the waters) as chain C, and 191 (the Histamine) as chain D. This can be seen above under the "fix" command, too. It
#was just easier this way.
morph
m1
A1-185 B186 C187-190 D191
m2
A1-185 B186 C187-190 D191
#Next we specify the number of morphing steps we desire. I found that 60 is a good number. This gives enough
#frames so that the movie can play at a decent speed - not too fast! - so that all important movements can be easily
#visualized. Furthermore, you can always adjust this speed when reading the image sequence of .png files into
#QuickTime.
60
#Specify the root name of the frame output files. I used morph_I for the first part, morph_II for the second part, and
#morph_III for the third part.
morph_I
#Choose Cartesian coordinates
c
m
#Superposition the desired atoms. In this case, A1-185 were superimposed. COOT may also be used for this step.
#However, if you are to repeat a morphing with LSQMAN, make sure to re-name your input .pdb files. If not, then
```

```
#your structures will already be superimposed from the start of the script, which can lead to many unexpected morphing
#events.
A1-185
999
# At this point you receive instructions to execute (@) a “.lsqmac” file that was generated. This file will complete the
#superpositioning... do not forget to run it!
morph_I.lsqmac
quit
```

MAKING THE MOVIE

This is accomplished by generated a .pse file in PyMol that incorporates all the frames, in order, of any executed morphs.

```
#For the first part of the NP4 simulation, lsqman generated
morph_I_1.pdb
morph_I_2.pdb
...
morph_I_60.pdb
#For the second part,
morph_II_1.pdb
...
morph_II_60.pdb

#For the third part,
morph_III_1.pdb
...
morph_III_60.pdb.

#In the command line in PyMol I typed
load morph_I_1.pdb, mov, 1
#This sets the first .pdb of the first part of the morphing as the first frame of a movie (entitled “mov”). Continuing #in
this manner,
load morph_I_60.pdb, mov, 60 .
#I then began to read in the next set of 60 morph_II files, starting with the 61st frame of the movie
load morph_II_1.pdb, mov, 61
...
load morph_II_60.pdb, mov, 120
#I then entered the set of 60 morph_III files
load morph_III_1.pdb, mov, 121
...
load morph_III_60.pdb, mov, 180
#Each .pdb (all 180!) had to be entered separately so as to ensure they were read in the correct frame order by PyMol.
#I then used the “mset” command in the PyMol GUI command line to generate a nice movie that would hold the frames
#at which the water, NO, and HSM (frames 1, 60, and 120, respectively) were bound to the heme.
mset 1 x30 1 -60 60 x30 61 -120 120x30 121 -180
# This holds the first frame (water bound) for 30 frame counts, plays frames 1-60, holds frame 60 (NO bound) for 30
#frame counts, plays frames 61 -120, holds frame 120 (HSM bound), then plays frames 121 -180. There is no need to
#hold frame 180 for 30 frame counts because it is the same as frame 1. In total, the movie becomes 270 frames (the
#original 180, +30+30+30 for the bound states = 270).

#To make a ray traced movie type into the command line
set ray_trace_frame=1
```

Under the File menu, select “Save Movie.” This will generate 270 .png files in the directory you specify and will take 10-15 minutes. If the computer you are working on does not have a movie-making program installed, save these .png

files to a flash drive or sftp them to a computer that has a program that can read them into a movie-making program. I used the Mac OS X QuickTime Pro for this task.

QUICKTIME PRO

Make sure that all of the .png files are in numeric order with the same root name in the directory you chose. Open QuickTime Pro and under the File Menu select Open Image Sequence. Select the first .png file from the directory with all 270. You will then be asked how many frames per second you wish your movie to be played at; I chose 23.67, which seemed to work quite well and created an eleven second-long movie. Within a couple of seconds your movie will be made.

EXTRAS

Transparency

For transparent effects here is an example of how I made loop A (residues 30-37) of NP4 transparent.

```
create new_obj, (resi 30:37)
```

```
#This creates a duplicate representation of this loop
```

```
#Then simply hide these residues in the initially present structure.
```

```
#Set the transparency
```

```
set cartoon_transparency, 0.5, new_obj
```

```
#Transparency values 0-1.
```

```
#Can be done for other representations, such as sticks, which can be important for side chains
```

```
set stick_transparency, 0.5, resi 36
```

*Important note: the residue specification of the desired transparent region may have to be more or less residues than the loop/chain/residues of interest, depending if alternate conformations were left in the original .pdb file.

Representing hydrogen bonds and van der Waals interactions

The easiest way to do this is to use the "Measurement" tool under the Wizard Menu. This needs to be done at each frame in which you would like to show such interactions. For the above NP4 movie, I needed to create three bound states (water, NO, and HSM, respectively, in the pocket).

Click on the the two atoms you want, then the measurement will appear as a separate item in the list of objects present. Hide the numeric label and change the color. White and yellow work well, and can be used to distinguish between hydrogen bonds and van der Waals interactions. When you are done with this image, ray trace it and SAVE IT WITH THE SAME ROOT NAME AS THE .PSE MOVIE FILE YOU PLAN TO CREATE. This avoids re-naming issues later for QuickTime. Also, you must keep the same view within PyMol to create the rest of the movie (i.e. do not create the bound states showing any desired bonds and then changed the point of view of the molecule).

*Note: Make life simple and create all of your bonds for each bound state at once. You can hide the ones that are not to be shown during a specific bound state. This way you can quickly render an image of each state as a .png file and then run the "Save Movie" option under the File Menu. This also reduces the likelihood of accidentally changing the point of view of the molecule while trying to create new bonds for a new bound state because there is no need to click on the image screen (just the side menu).
