How to Make Molecular Movies (v1.0)

Introduction

To make molecular movies this way you will need access to the latest Molscript, Raster 3D, a few scripts I'll give you the paths for, and Adobe Imagready. Newer methods taking advantage of PyMol and ImageMagic can be found on the web.

The hardest part of the movie making process, more-or-less, is getting the initial figure. You should carefully choose an orientation for your molecule that will allow people to see the structure and any details you're highlighting clearly. For example, if you are showing H-bonds in the base-pairs of DNA, don't align the rotation axis with the double-helix axis or the interactions will be obscured. . .instead rotate perpendicular to the DNA-axis.

Color schemes and background are also up to you. I tend towards black backgrounds for projected movies. The structure looks more dramatic and it's easier for the audience to see details against black rather than washing them away in a haze with white. Also, with current projectors you can get away with primary colors whereas you can't with actual slides. Still, I tend to recommend pastel colors for anything important as they always standout despite projection method or background color.

When coloring ribbons in Molscript don't forget to set the plane2colour in addition to the planecolour. Otherwise the edges of beta strands and the insides of alpha-helices will come out gray. For still images it's okay to have a different plane2colour but for a movie there's a lot of information going by fast and I tend to feel that the excess colors become distracting.

Ok, onto the nuts and bolts of doing this:

Setting up the PDB file

1) To make your life VERY easy in this process it is imperative that you center your structure on its center of mass. Molscript sets xyz=0,0,0 at the center of the screen. However, your structure is probably nowhere near the origin of the unit cell and floats somewhere far away from 0,0,0. If you don't center it then you'd have to translate the molecule along X, Y, and Z until the center of the molecule sits very close to the y rotation axis, otherwise the structure will precess instead of rotate. This is easily accomplished with Moleman.

To start the latest Moleman (or 6d_moleman) on the cluster, type 'moleman2'. It'll scroll for a bit and prompt you for a library file. Hit enter. It'll scroll some more (showing you the commands to activate the MANY useful tools moleman has) and give you a prompt. Use the 're' command to read the file. Once it's read in use the 'xy ce' command to

center the molecule on its center of mass. The write out the file with the ‘wr’ command. Following is an example session doing this:

```
MOLEMAN2 > re
 PDB file ? (m1.pdb) mymol.pdb
 Reading from file : (mymol.pdb)
 in normal PDB format
 ignoring hydrogen atoms

 >>>>> END card encountered <<<<<

 Nr of lines read : (       5112)
 Nr of hydrogens skipped : (          0)

 Total nr of residues      : (        642)
 Nr of amino acid residues : (        639)
 Nr of nucleic acids       : (          0)
 Nr of waters              : (          0)
 Nr of metals              : (          0)
 Nr of inorganics          : (          0)
 Nr of carbohydrates       : (          3)
 Nr of organic compounds   : (          0)
 Nr of other compounds     : (          0)

 Checking for missing/extra atoms ...

 Checking "special" positions (X~Y~Z), Bs and Qs ...
 No suspicious coordinates encountered
 All atoms have B <= 100 A2
 All atoms have B >= 2.0 A2
 All atoms have Q <= 1.0
 All atoms have Q >= 0.01
 No atoms with ANISOU cards

 YASSPA results:
 Nr residues of type Non-protein          : (          3)
 Nr residues of type Loop or turn         : (        269)
 Nr residues of type Alpha helix          : (        244)
 Nr residues of type Beta strand          : (        124)
 Nr residues of type Left-handed helix    : (          2)
 %-age alpha : (  38.185)
 %-age beta  : (  19.405)
 %-age lefth : (   0.313)

 Nr of atoms now : (       5102)
 Nr of residues  : (        642)
```

```
 Select ALL atoms
 Selection history : (ALL |)
 Nr of selected atoms : (       5102)
 CPU total/user/sys :       2.0       1.9       0.1
 MOLEMAN2 > xy ce
 Moving CofG of selected atoms to (0,0,0)
 Nr of selected atoms : (       5102)
 Centre-of-Gravity : (  -1.329  -12.800  145.558)
 CofG now at (0,0,0)
MOLEMAN2 > wr
 PDB file ? (mymol.pdb) mymol_cen.pdb
 Output PDB file : (mymol_cen.pdb)
 Format : (Pdb)
 Atoms  : (ALl)
 Number of atoms to write : (       5102)
 Nr of atoms written : (       5102)
 Nr of lines written : (       5120)

 CPU total/user/sys :       1.8       1.7       0.1
 MOLEMAN2 > quit


 *** MOLEMAN2 *** MOLEMAN2 *** MOLEMAN2 *** MOLEMAN2 ***
MOLEMAN2 ***

 Version - 001229/2.8.2
 Started - Mon Apr 1 19:38:25 2002
 Stopped - Mon Apr 1 19:41:31 2002

 CPU-time taken :
 User    -      3.8 Sys    -      0.2 Total   -      4.1

 *** MOLEMAN2 *** MOLEMAN2 *** MOLEMAN2 *** MOLEMAN2 ***
MOLEMAN2 ***

 >>>>>>>>>>>>>> USF .... Uppsala Software Factory <<<<<<<<<<<<<<
 >>>>>>>>>> This program: (C) 1992-2001, G J Kleywegt <<<<<<<<<<
 >>>>>>>>>>>>>>>> E-mail: gerard@xray.bmc.uu.se <<<<<<<<<<<<<<<<
 >>>>>>>>>>>>>>>>>> http://xray.bmc.uu.se/usf <<<<<<<<<<<<<<<<<<

 *** MOLEMAN2 *** MOLEMAN2 *** MOLEMAN2 *** MOLEMAN2 ***
MOLEMAN2 ***

STOP ... Toodle pip ... statement executed
```

Okay, this is now the PDB file to use in Molscript.

Starting in Molscript

Making clean, informative, and beautiful figures in Molscript is a topic FAR beyond the scope of this guide. Scene setup (the process of making the initial scene which will be animated later) is the most time consuming part of the whole movie process, requiring in general AT LEAST 2 hours of work before animation can begin, and that's if you already know what you're doing with Molscript. There are lots of tricks you can pull in Molscript to make various kinds of motion occur in the movies but that will be included in a later section for people interested in doing more than rotations.

To make an initial input scripts for your PDB file run molauto. This has a pretty decent algorithm for determining secondary structures. I've found it to be pretty accurate but you really have to check loops as it has a tendency of assigning loops as alpha helices. This may not seem like a big deal but the smoothing algorithm for helices drastically shortens and changes the shapes of such short loops. In still figure this sometimes gets hidden but in a rotating movie EVERY mistake you make will become obvious to the audience.

Molauto has several option and they can be seen from simply typing in 'molauto':

```
citdown:~> molauto
Usage: molauto [options] pdbfile
  -notitle   do not output a title
  -nocentre  do not centre the molecule
  -cylinder  use cylinders for helices
  -turns     use turns when specified, otherwise coil
  -nice      nicer rendering; rainbow colours, more segments
  -thin      set helix, strand, coil thickness to 0
  -noligand  do not render ligand(s)
  -bonds     render ligand(s) using bonds (default)
  -stick     render ligand(s) using ball-and-stick
  -cpk       render ligand(s) using cpk
  -nocolour  do not use colour for schematics
  -ss_pdb    secondary structure as defined in PDB file (default)
  -ss_hb     secondary structure by H-bonds (DSSP-like) criteria
  -ss_ca     secondary structure by CA-geometry criteria
  -h         output this message
----- MolAuto v1.1.1, Copyright (C) 1997-1998 Per J. Kraulis
----- http://www.avatar.se/molscript/
```

For making your own figures I suggest running molauto with the nocolour flag ('molauto –nocolour mymol_cen.pdb'). Otherwise it changes the color ever residue and it's a pain to remove. You'll have to go through and assign your own colors later. The initial script

that molauto makes can be pasted into an editor and saved. Don't forget the 'end_plot' card at the end of the file.

Orientation

In the 'olden' days an orientation in molscript was setup by manually updating rotation matrices until you got more-or-less what you wanted. How PAINFUL. There are two really excellent ways to do this now, either from O or in molscript.

Option I: In O, orient the molecule the way you want. Type '*plot*'. O then writes a file in the directory from which you launched O called *'o_plot'*. Near the top of the file is a line that looks like:

transform 0.8507 -0.0260 -0.5249 -0.0489 0.9905 -0.1283 0.5233 0.1348 0.8414

Paste this rotation matrix into the *'transform by rotation'* line in your molscript input file and rerun molscript.

Option II: Type 'molauto mymol_cen.pdb |molscript2 –gl' where mymol_cen.pdb is your centered pdb file of interest. This will bring up a window with an automatically generated ribbon diagram. You can move the molecule using the mouse. To output the coordinates, right click in the window and choose the option that writes out the coordinates. This will write a 3x3 rotation matrix in the window you ran the molauto program from. Paste this matrix in the same way as for method (1) and rerun molscript (you can do this by right clicking in the window again and clicking 'reread input file'). **NOTE**: If you reread the input file, change the orientation of the structure and decide you'd rather have that one instead, you may output a new rotation matrix. HOWEVER, this new matrix is the rotation away from the orientation specified by the rotation in the input file. Therefore, you must ADD this new matrix to the molscript file, not replace it. Molscript will be happy to reorient your molecule as many times as you'd like it to (I have one input file that contains about 10 matrices to get it aligned properly).

You can check the progress of your changes to the molscript input file by rereading it in the open-gl window. However, this representation is a bit far from what it will look like in the final rendered format. You can get a better idea by looking at a postscript cartoon of the file by running:

citdown:~> molscript2 <myinput.in> myinput.ps &

This creates a postscript file called myinput.ps which can be viewed on the SGI using xpsview.

citdown:~>xpsview myinput.ps &

If you're satisfied it's time to render the image in raster3d (a ray tracing program). At current time raster3d only works on the alphas. You should be running the jobs there

anyway as it's nauseatingly slow on the SGIs. I suggest opening a second window on the SGI, rlogin into the alpha of your choice (I prefer the dual processor ones as generally there is a free processor on it even if someone else is running a job, that way you don't interfere with someone's refinement). Use the following commands there to make an initial rendered file:

twister:~> molscript2 –r <myinput.in> myinput.r3d &

This creates a raster3d input file. If you take a look at it in a text editor you'll notice it has a header with scene information. Unless you specified the background color in your molscript input file then it has been set by default to black. You can change this to white now. You can also opt to have shadows cast. I suggest you only do this for more 'artistic' movies or still images as the default shadows are very distracting when looking for real information. Save any changes you made to the r3d file and then type:

twister:~> render –tiff myimage.tiff < myinput.r3d &

The computer will churn for a while and eventually write out the file (usually within 30 seconds). You can look at it on the SGIs using the xv program.

citdown:~> xv myimage.tiff &

Currently, the raster3d on the alphas will ONLY make tiff files. If the image you saw wasn't big enough change the number of tiles and pixels per tile in the r3d file header. Generally the maximum is 50x50 tiles and 30x30 pixels (meaning 300x300 dpi). That will great a huge file and take a while to render and load.

Now that you see it rendered you may want to go back and make some changes. If you're happy, proceed to setting up the movie.

A note about multiple rotations:

You may find it useful to have more than one molecule or image of your molecule present in a single animation (for example you'd like to see the molecule rotate about a vertical AND a horizontal axis at the same time). In this case it turns out to be better to render the WHOLE scene rather than making two movies and trying to make PowerPoint play both of them at the same time. Two movies playing in the same frame become VERY jittery. In this case you'd read in two copies of your molecule and separate them by some distance and apply the Y rotation and X rotation separately at the end of scene setup. I've found this particularly helpful for DNA where each rotation has something different to offer, or in the case of the famous slide showing all the Class I MHC homologues rotating together.

Setting up the movie

There's a few things left to do to the molscript input file before you can render. **The single most important one is setting the window size.** Window size in molscript is akin to zoom. The smaller the number the bigger your molecule gets, and it can even be cropped this way. If you run molscript without first setting the window size it will set it automatically to include every detail at the edge of the scene. This is great except if you run the animation script it will set this number for EVERY frame. If the observed dimensions of your molecule change as it's rotated (and they will if you're rendering anything other than a single sphere) then the 'zoom' will change each frame and your molecule appear to fly away from and towards your audience as it spins. Do not do this unless you have enough Dramamine on hand for every member of your audience**.** To fix this you must set the window size INSIDE the molscript input file. The best way to get the number itself is to run molscript a couple of times with the molecule in various orientations (go ahead and set the Y rotation, for vertical spins, to several values and rerun molscript). At the end of the molscript log will be the new window and slab (slab is also important or your molecule could get clipped as it turns). Generally you need to find the orientation where the molecule has the greatest left→right dimensions and choose that window size (and probably add about 5 to it just to be safe). The best slab choice would be taken from a rotation about 90 degrees away. Frankly though, slab doesn't tend to affect things until it's too small so setting it to something large will generally work (like twice the largest dimension of your protein).

Window and slab info goes at the top of the script before molecule information is read:

```
plot

window 180;
slab 120;

! This is some useful info about the following script

 read moleculename "mymol_cen.pdb" ;
 read 2ndmolecule "anothermol.pdb" ;
```

Ok, only one more modification to go:

The script looks for the following line somewhere within your molscript file:

INSERT Y ROTATION

It will then replace this line with the appropriate information contained in the animation script. This line is generally best left as a separate module at the end of the general orientation information:

```
Transform in moleculename by rotation
INSERT Y ROTATION
;
```

As in all things UNIX case does matter.

One last thing:

When molscript runs it puts header info into the r3d file. It can either write this anew for each file or it can take it from a header.r3d file that is present in the same directory. You HAVE to use a header.r3d file otherwise you can't specify the size of the final image. Below is an example header.r3d file that contains information that will result in an image just slightly larger than a PowerPoint window after cropping. Making files much larger result in wasted time transferring files, difficulties handling them on the Mac, and larger QuickTime files which will only gum up your presentation:

```
render (Raster3D) input file, MolScript v2.1.2, Copyright (C) 1997-1998 Per J. Kraulis
45 45      tiles in x,y
30 30      pixels (x,y) per tile
3          antialiasing level (1,2,3)
1 1 1      background colour
F          no shadows cast (T = shadows cast)
25.6       Phong power (specular highlights)
0.25       secondary light contribution
0.05       ambient light contribution
0.25       specular reflection component
0          eye position (0 = no perspective)
1 1 1      main light source position
1 0 0 0    view matrix: input coordinate transformation
0 1 0 0
0 0 1 0
0 0 0 164.715
3          mixed objects
*          (free format triangle and plane descriptors)
*          (free format sphere descriptors)
*          (free format cylinder descriptors)
```

You can change any of these numbers but I'd recommend leaving the others alone (especially antialiasing. You don't want jagged images do you!?)

The animation script

The following script was written by Art Chirino and I have several modified versions of it that handle translations, X, Z, or arbitrary axis rotations, and separate scripts that handle the running of Bobscript (and extension to molscript that allows you to do some fancier drawings like H-bonds, planes through aromatic residues, DNA backbones, etc. and then passes them through to render on a different machine since bobscript doesn't run on Alphas and render doesn't run on SGIs). The script is shown in totality below with the salient features highlighted.

```
#!/bin/csh -f
#
# batch script to generate semi-continuous rotating
# Molscript images: A.J. Chirino April, 2000
#
# starting Molscript template file; Make sure to
# edit it and insert "INSERT Y ROTATION" *after*
# all transformations but *before* the final ";"
#
set molscript_template="finalinput.in"
#
# All files generated will have this base in their file names
set base_filename="molname"
#
# raster3d graphics file format:
# if running on SGI/IRIX, set to "sgi"
# (you will then also have to convert these to tiff or some other
#  format that Photoshop, etc likes)
# if running on ALPHA/TRU64 or LINUX, set to "tiff"
#
set graphics_format="tiff"
#As of right now you don't have a choice, always use tiff
# This is the rotational increment (in degrees)
@ increment = 2
#
##########################################################################
######
@ counter = 0
@ degrees = $counter * $increment
while ( $degrees < 361 )
#
# Pad zeros
if ( $counter < 10 ) then
        set num="00"${counter}
else if ( $counter >= 10 && $counter < 100 ) then
        set num="0"$counter
else if ( $counter >= 100 && $counter < 1000 ) then
        set num=$counter
endif
#
set out_molscript_filename=${base_filename}${num}.in
set out_graphics_filename=${base_filename}${num}.${graphics_format}
#
echo "Rendering Image #"$counter
sed "s/INSERT Y ROTATION/by rotation y $degrees.0 /" $molscript_template > \
$out_molscript_filename
```

```
#
molscript2 -r  < $out_molscript_filename | render -${graphics_format} \
$out_graphics_filename
#
@ counter = $counter + 1
@ degrees = $counter * $increment
#
end
```

In this example the rotation increment is set to 2 degrees. This makes a very smooth rotation with a manageable number of frames. All the animations on the website are 5 degree rotations and look a little jagged. I have modified the counter to stop at 360 but this results in an extra file which you'll delete later.

So, quick check: You must have in the same directory
1) Your pdb file(s)
2) You molscript script with the INSERT Y ROTATION flag in it and *not* commented out (180 images of the same orientation is a very boring movie)
3) Your header.r3d file
4) The batchrot.csh script.

You run the script like you would any other C-shell script, and in the same directory as your input file and where you'd like the output to go. You should run this on one of the Alphas as both molscript2 and raster3d run there. Depending on the size of your structure this could take anywhere from 10 minutes to an hour to make all 180 files. When it is done you're going to have a directory containing all the input files molscript made where the 'INSERT Y ROTATION' line has been replaced with 'by rotation #' where # is the new rotation value. You'll also have a bunch of tiff files sequentially numbered. You can go ahead and look at a couple of the files to make sure they came out correct.

A note about speed

The NFS mount on our cluster is slow. If you render from your usual user directory the Alpha will be sending information back and forth over the network to the disk stack. This makes WOEFUL use of processor time. I once noted that on the fastest Alphas only about 25% of the processor time was being used to render while the rest of the time was spent waiting for data transfer. Go ahead and transfer the required files to the temp directory on the appropriate alpha. The local disks have much faster access and will probably cut the rendering time by 75%. Using the settings in the above header.r3d file each image should be about 2.4 Mb.

Moving the files:

You must now transfer your files to a Mac to make the final movie. Now that all computers are behind the firewall this is easier. When you ftp, use 'ftp –i' as the

command. The '-i' flag prevents the computer from asking you if you want to put EACH individual file there and instead transfers them in batch when you use mget or mput. This is going to take a while.

Finally, assembling the movie!!!

On your Mac of choice, open Adobe ImageReady. You may want to make sure that the program has access to most of the machine's memory and that all the other programs are closed. From the file menu choose 'Import→folder as frames'. Find your folder, single click on it, and then click the choose button in the new window's menu bar. This will begin the process of importing ALL image files in that directory sequentially. The status bar will slow down as it gets towards the end. One way to make this easier is to go ahead and use a script in Photoshop to open each file, crop is appropriately, and save it out as a jpeg file. This will drastically reduce the size of the dataset without much change in quality. Look at the manual if you want to know how to write Photoshop scripts.

If it went successfully:
You are now faced with a full picture of your first image and a film strip at the bottom of the screen. You need to scroll all the way to the end of the strip and click on the last few images which will update the image in the main window. What you're looking for it the last image where a two degree rotation will bring you to the 1st image in the strip. You don't want the last and first image to be the same, else the movie will pause on that orientation as it plays it twice. It keeps your movie from being smooth and annoys obsessive compulsive people like me ☺. Delete any extraneous frames (the little right facing arrow key at the edge of the film strip has a 'delete frames' feature. You can highlight multiple frames at once using the control key.

Now, due to the way raster3d renders you probably have lots of white (or black) space around your molecule that you don't need. You should clip this off to make the movie file smaller. You can set crop boundaries just like in Photoshop, but here it will apply the same crop boundaries to ALL the images. Of course, since your molecule has moved around it may fall outside the crop boundaries you set on a particular frame. Check several other orientations to make sure you don't crop your molecule too.

Now that you have 358 degrees of rotation represented and cropped, you need to set the timing. If you don't do this your structure will go whirling by like a tornado, and again you may have to pass out Dramamine. Highlight the entire film strip (using shift-click) and then right click on the time indicator of any frame (found near the bottom of the frame). For a movie with 2 degree increments I've found a timing of 0.075 to 0.09 seconds per frame works pretty well. If you are using fewer frames up the timing accordingly.

Now you can write the final file. Under 'file' choose 'export original'. This gives you a menu of choices. Choose a file name and then set the file type to 'QuickTime movie'. Next it'll ask you for a jpeg compression scheme. I generally use the default of 'jpeg photo' and a compression ratio of 65%. The computer will now SLOWLY write out the

file. You can double click on the icon for it to view the movie in a QuickTime player to verify it's what you want.

Import into PowerPoint

From a blank PowerPoint slide (with the same color background as your movie) import the 'movie from file'. The movie should appear and be a little bigger than the frame. Go ahead and use the corner sliders to drag it down to size (using the shift key to keep the aspect ratio constant and avoid distorting the image).

Now, click on your movie and select from the pull down menus 'slide show→custom animation'. From inside 'custom animation' choose 'more options' and click on 'loop continuously until stopped'. This allows your rotation to be played as one continuous rotation.

Also, from within 'custom animation' I suggest you select the timing tab and set the animation to start automatically, either on loading the slide or after a short delay. It's distracting in talks when people interrupt their talk and their slide to drag the mouse over the image and click on it to play. Also, they tend to forget to move the mouse away and on some platforms the mouse icon stays over your movie. This gives a much cleaner effect to the presentation. Usually a 2-3 second delay is enough for the audience to be pleasantly surprised that the image they thought was static is actually a movie.

Note on labeling:

Labels of your molecule can be added in PowerPoint but rotating labels are HIGHLY distracting. If you need to label, go ahead and shrink the movie down more and add them as text in PowerPoint where the placement and color of the text is vertically aligned with what you're labeling.