

Using PyMOL to visualise proteins

Last updated: April 22, 2021

1 Making nice images in PyMOL

1.1 Settings to change

The default settings in PyMOL do not necessarily make for nice pictures. Described below are some settings that can be changed to make images look better.

1.1.1 Background and cartoon colour

PyMOL starts with a black background and green cartoon. To change the background colour to white:

```
bg_color white
```

The sidebar can be used to change the cartoon colour.

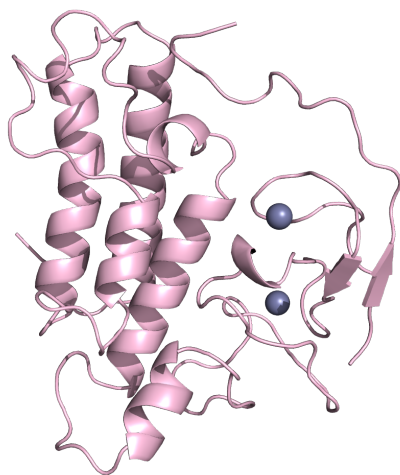
1.1.2 Ray trace settings

Images should be ray traced prior to export, e.g. when exporting to png:

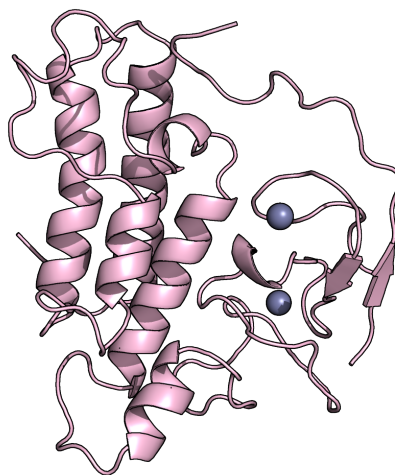
```
png my_protein.png, width=9cm, height=9cm, dpi=300, ray=1
```

4 ray trace modes are available, although modes 0 (default) and 1 are most often used. Ray trace mode 1, in particular, makes for nice images (it adds a black outline)!

```
set ray_trace_mode, 1
```



(a) Ray trace mode 0 (default)



(b) Ray trace mode 1

Figure 1: Ray trace settings 0 and 1

Modes 2 and 3 are a more niche, and are shown below:

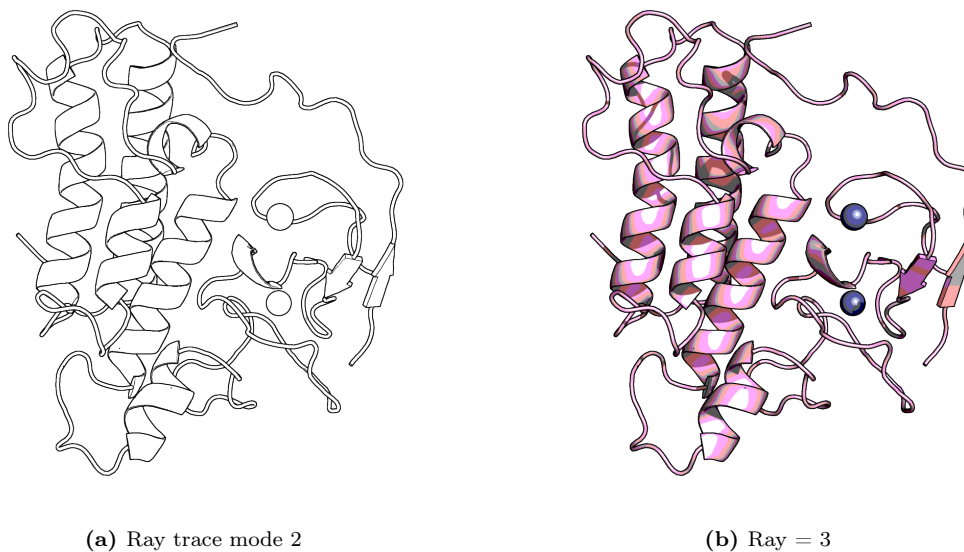


Figure 2: Ray trace mode 3

Ray shadows can be switched off, although this is very much a personal preference!

```
set ray_shadow, 0
```

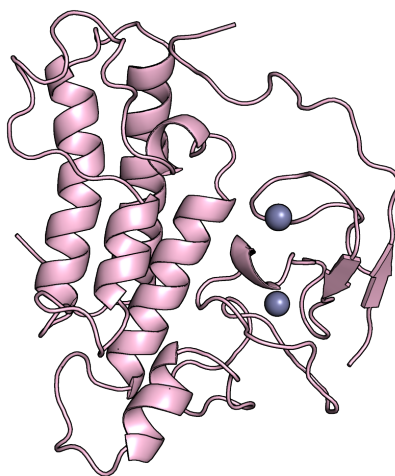


Figure 3: Ray trace mode 1, ray shadow 0

1.1.3 Ambient lighting

The default lighting setting (0.1) can be a bit dim. To increase brightness, change the ambient setting (number from 0 to 1):

```
set ambient, 0.3
```

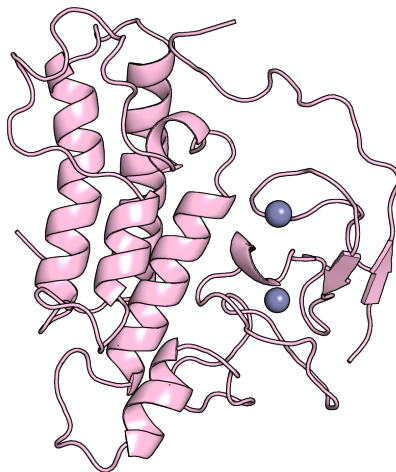


Figure 4: Ambient 0.3, ray trace mode 1, ray shadow 0

1.1.4 Cartoon settings

Two settings can be changed to make cartoons look prettier, especially in proteins with lots of α -helices:

```
set cartoon_fancy_helices, 1
set cartoon_highlight_color, gray70
```

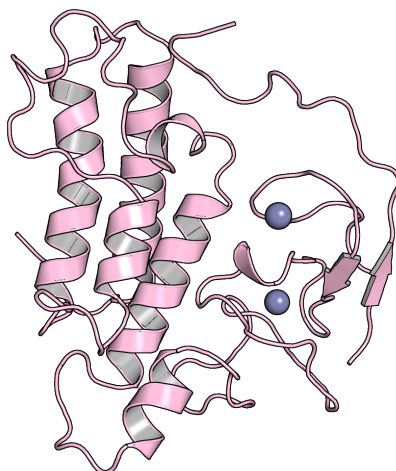


Figure 5: Cartoon fancy helices 1, cartoon highlight color gray70, ambient 0.3, ray trace mode 1, ray shadow 0

1.1.5 Orthoscopic mode

This is off by default, i.e. the default is perspective mode. To switch on orthoscopic viewing:

```
set orthoscopic
```

1.1.6 Other settings

To view the full list of settings, go to Setting » Edit All. Settings can be changed through this menu, or via the command line if the name of the setting is known.

1.2 pymolrc

A `.pymolrc` file (placed in `$HOME`) can be used to load custom settings on startup.

A sample `.pymolrc` is shown below.

```
bg_color white
set orthoscopic
set line_width, 6
set ambient, 0.3
set line_as_cylinders, on
set opaque_background, off
set dash_color, black
set dash_transparency, 0.2
set ray_shadow, 0
set ray_trace_mode, 1
set cartoon_fancy_helices, 1
set cartoon_highlight_color, gray70
```

Apart from settings, aliases can also be included in `.pymolrc`, e.g. I have an alias to ray trace a 1000×1000 pixel image, to check how it looks like before exporting.

```
alias snapshot, ray 1000,1000
```

2 Useful commands in PyMOL

2.1 Storing views

After finding a nice view, it can be stored with:

```
view my_view_name, store
```

To retrieve a view (e.g. after playing around with the protein), use:

```
view my_view_name
```

To delete a view:

```
view my_view_name, clear
```

To see all views that are stored, type `view`, followed by a space, then press tab.

To get the current view information:

```
get_view
```

This prints out the view information in a format that can then be pasted into a script.

```
### cut below here and paste into script ###
set_view (\
  -0.617719293,  -0.294218957,  -0.729284823,\
  -0.756448865,  -0.031187352,   0.653309822,\
  -0.214960709,   0.955228984,  -0.203296438,\
   0.000000000,   0.000000000, -150.259384155,\
  37.276630402,  43.539588928,   17.712219238,\
  118.465644836, 182.053131104,   20.000000000 )
### cut above here and paste into script ###
```

3 Other resources

There are lots of resources available online, which are listed below.

Webpages:

- [PML scripting in PyMOL](#)
- [Making pretty pictures with PyMOL](#)
- [A quick reference guide for intermediate users](#)

For a fairly recent PyMOL tutorial, see [this workshop by Garrett Morris](#).