

# TP pymol

## 1 Movies

### 1.1 Tutos

Sur le wiki de pymol, il y a un tuto très complet sur les possibilités de pymol pour faire des films :

<https://pymolwiki.org/index.php/MovieSchool>

Les slides d'une présentation à regarder en association de la "MovieSchool" :

[http://www.virology.wisc.edu/acp/Classes/DropFolders/Drop660\\_lectures/2012\\_660/L02\\_PyMol\\_Movies.pdf](http://www.virology.wisc.edu/acp/Classes/DropFolders/Drop660_lectures/2012_660/L02_PyMol_Movies.pdf)

Des mêmes auteurs, mais en forme de polycopié plus détaillé que les slides :

<http://www.virology.wisc.edu/acp/Classes/tutorials.html>

<http://www.virology.wisc.edu/acp/Tutorials/PyMol-animations.pdf>

### 1.2 Commandes fréquentes

commande	explication
<code>mplay</code>	joue le film actuel
<code>mstop</code>	arrête le film
<code>rewind</code>	retour au début du film
<code>mset</code>	Efface le film actuel
<code>movie.add_roll(8.0, axis='y')</code>	rotation complète autour de l'axe y en 8 secondes
<code>movie.add_rock(4.0, axis='y')</code>	rotation oscillante autour de l'axe y pendant 4 secondes
<code>movie.add_scenes(pause=4.0, rock=0)</code>	Montrer chaque scène pendant 4 secondes sans "rock"
<code>movie.add_nutate(10, 60)</code>	Rotation autour des trois axes de 60° pendant 10 secondes

### 1.3 Premiers exemples

Voici quelques exemples simples pour faire un film sous pymol. Ces exemples sont inspirés de la "MovieSchool" (voir plus haut).

#### 1.3.1 Structures RMN

1. Charger le fichier PDB 1nmr depuis Internet avec la commande `fetch` :

```
fetch 1nmr
```

Démarrer l'animation avec la commande `mplay` ou le bouton play en bas à droite. On peut arrêter l'animation avec la commande `mstop` ou le bouton correspondant.

2. Chercher à améliorer l'affichage en changeant les styles (sticks, cartoon, etc) et les couleurs.
3. Afficher toutes les structures en même temps avec :  
`set all_states, on`  
 Pour n'afficher qu'une seule :  
`set all_states, off`
4. Extraire chaque structure en un objet séparé :  
`split_states 1nmr, prefix=conf`  
 Cacher toutes ces structures avec :  
`disable`  
 Puis afficher seulement les 10 premières structures :  
`enable conf000*`  
 Alternativement on peut aussi cacher/afficher les objets en cliquant sur leur nom à droite.
5. Colorier chaque structure avec une autre couleur :  
`util.color_objs("all")`

### 1.3.2 Rotations

6. En prenant les structures précédentes, essayer ces commandes pour générer des rotations. Après chaque commande démarrer le film avec `mplay`, puis avant chaque commande effacer le film précédent avec `mset`.  
`movie.add_roll(5)`  
`movie.add_roll(5, axis='y')`  
`movie.add_rock(5)`  
`movie.add_nutate(10, 60)`

### 1.3.3 Zoom

7. En prenant les structures précédentes, on cherche à faire un zoom sur le premier résidu de la première conformation. Marquer tous les résidus numéro un :  
`show sphere, resi 1`  
 Ajuster globalement l'orientation et la position des structures avec la souris.  
 Sauvegarder cette orientation avec la commande `get_view`  
 Pour cela on doit copier&coller la sortie de cette commande dans un fichier texte (à ouvrir sous `gedit`).  
 Sauvegarder la position actuelle de la caméra :  
`scene F1, store`  
 Faire un zoom sur le résidu en question :  
`zoom resi 1 and conf0001`  
 Sauvegarder la position actuelle de la caméra :  
`scene F2, store`  
 Changer entre les deux positions de la caméra avec les touches F1 et F2.
8. Générer un film entre ces deux positions :  
`movie.add_scenes(pause=2.0, rock=0)`

## 1.4 Exemples plus complexes

Maintenant on essaie progressivement de faire des films plus complexes. Pour cela on n'utilise ici plus la ligne de commande de pymol directement, mais on utilise uniquement des scripts qu'on charge dans pymol avec :

```
@script.pymol
```

Un script est simplement un assemblage des commandes pymol dans fichier de texte qu'on peut éditer par exemple avec "gedit". Dans un script on peut ajouter des lignes de commentaires en mettant # devant.

9. Assembler toutes les commandes pymol nécessaires pour générer le film de la question précédente. Ajouter `reinitialize` à la première ligne du script et modifier la commande `fetch 1nmr` en `fetch 1nmr, async=0` pour s'assurer que la structure est bien chargée avant l'exécution du reste du script. Pour avoir la même orientation d'affichage, insérer dans votre script la commande `set_view` telle que vous l'avez obtenue dans la question précédente avec la commande `get_view`.

### 1.4.1 HIV protéase, 1ère partie : zoom sur le ligand

Ceci est une autre protéine qui n'a rien avoir avec l'exemple précédent. On utilise la structure PDB 1hvp.

10. Télécharger et ouvrir le fichier 1hvp.pdb avec gedit. Combien de chaînes est-ce qu'on a ? A quoi correspond le résidu 200 de la chaîne B ?
11. Charger 1hvp.pdb sous pymol. Sans utiliser un script pour l'instant, colorier chaque chaîne avec une autre couleur. Montrer les chaînes sans le ligand en mode surface. Chercher une orientation qui permet de bien voir le ligand. Sauvegarder cette orientation avec `get_view`.
12. Assembler et tester un script 1hvp.pymol qui fait les mêmes étapes que dans la question précédente.
13. Ajouter à la fin de ce script des commandes pour faire un zoom sur le ligand en générant un film comme avec l'exemple 1nmr (voir plus haut).

### 1.4.2 HIV protéase, 2ème partie : bouger des objets

On voudrait maintenant faire un film où les entités du complexe 1hvp sont d'abord séparées, puis s'assemblent lors du film. Pour cela les commandes vues plus haut ne suffisent pas, il faudra utiliser la commande `mview` telle qu'elle décrite dans la "MovieSchool" sur le wiki de pymol.

Ici les scripts sont vraiment incontournables, car il y a aura beaucoup de copier&coller de commandes similaires. On assemble le film progressivement dans le script dans l'ordre des frames du film. D'abord on définit combien de frames le film doit avoir, sachant qu'on affiche en général 30 frames par seconde (fps). Ici on choisit 360 frames par exemple :

```
mset 1 x360
```

Il est important de placer cette commande avant la commande `set_view`.

14. Avant la commande `mset 1 x360` on place les commandes de style et coloriage. Il faut aussi séparer les entités du complexe dans des objets distincts, par exemple comme ceci pour la chaîne A :

`extract AA, chain A`

Puis on cachera le ligand pour l'instant avec la commande `hide`.

15. Puis on déplace les objets en dehors de l'écran avec `translate`, comme par exemple :  
`translate [-100,0,0],object=AA`  
qui bouge l'objet AA de -100 sur l'axe x.
16. Comme le ligand est caché et doit apparaître plus tard dans le film, il faut d'abord enregistrer sa représentation "caché" avec :  
`scene F1, store`  
Ensuite après avoir définie la longueur du film (voir plus haut) on programme les frames où il y a un événement. Ceci commence avec une commande :  
`frame 30`  
où ici on se place au frame numéro 30. Puis on doit enregistrer la position des objets à ce frame comme ceci :  
`mview store, object=AA`  
On doit également enregistrer la représentation des objets :  
`mview store, scene=F1`  
Si on a changé la position d'un objet au frame 180 par exemple, alors on doit laisser `mview` générer la transition entre le frame 30 et 180 ici. Pour cela on ajoute en plus de la commande avec "store" une deuxième commande avec "interpolate" ou plutôt "reinterpolate" pour les frames suivants :  
`mview store, object=AA`  
`mview interpolate, object=AA`
17. Sans bouger les objets enregistrer le premier frame et le frame 30 avec les objets et la scène F1. Ceci fera une seconde de pause au début du film.
18. Enregistrer un frame 180 où les chaînes de la protéine (sans ligand) sont de nouveau assemblées. Pour cela il faut utiliser `translate` avant les commandes `mview`. Ne pas oublier les commandes avec "interpolate" en plus du "store".
19. Ajouter encore une pause d'une seconde.
20. Au frame suivant afficher le ligand et sauvegarder cette représentation sous la scène F2, puis appliquer cette scène avec `mview`.
21. Quelques secondes plus tard, replacer le ligand à sa position d'origine sans oublier d'appliquer toujours la scène F2 avec `mview`.

### 1.4.3 Exemples de la Movie School

Essayer les exemples de la movie school sur le wiki de pymol.

## 1.5 Exporter un film

Pour exporter un film en MPEG directement avec pymol il faut installer "freemol", voir ici :

[https://pymolwiki.org/index.php/MovieSchool\\_6](https://pymolwiki.org/index.php/MovieSchool_6)

Pour activer le ray-tracing des frames il y a la commande :

```
set ray_trace_frames, 1
```

C'est plus jolie, mais prend un peu de temps. Pour le désactiver on utilise la même commande, mais avec un zéro :

```
set ray_trace_frames, 0
```

## 1.6 Trouver les commandes qui correspondent au menu

Le menu "Movie" de pymol contient des sous-menus très facile à utiliser pour réaliser des opérations assez complexes. Malheureusement il n'est nul part documenté à quel ligne de commande pymol chaque sous-menu de "Movie" correspond. Donc on a dû mal à les intégrer dans un script pymol. Heureusement que pymol est open-source et qu'on peut trouver les deux fichiers source qui permettent de faire le lien entre menu et commande. Les noms des sous-menus sont définies ici des sources de pymol :

```
modules/pmg_tk/skins/normal/___init___py
```

Puis toutes les fonctions pour un "Movie" sont dans :

```
modules/pymol/movie.py
```

et peuvent être appelées depuis la ligne de commande pymol via :

```
movie.commande(arguments)
```

Pour par exemple faire un

```
Movie->Program->Scene Loop->Steady
```

il faudrait utiliser cette commande (ici 4 secondes de pause sur chaque scène) :

```
movie.add_scenes(pause=4.0, rock=0)
```

On peut y ajouter un "rock" en mettant `rock=-1` pour utiliser la valeur actuel de `sweep_mode` dans pymol ou alors spécifier le `sweep_mode` avec `rock` positif, car dans ce cas là : `sweep_mode=rock-1`. Le paramètre `sweep_mode` peut avoir ces valeurs : 0 = axe x, 1= axe y, 2 = axe z, 3 = nutation xy, voir ici :

```
http://pymol.org/dokuwiki/doku.php?id=setting:rock
```

## 2 Plugins pour pymol

Il existe un très grand nombre de plugins pour pymol, voir ici pour la liste :

```
https://pymolwiki.org/index.php/Category:Plugins
```

En plus il existe une importante collection de scriptes, voir ici :

```
https://pymolwiki.org/index.php/Category:Script\_Library
```

Une grande partie des plugins et une partie des scriptes peuvent être installés tous à la fois via "git", voir ici pour les étapes à faire :

```
https://pymolwiki.org/index.php/Git\_install\_scripts
```

## 3 Liens vers d'autres tutos

```
http://www-cryst.bioc.cam.ac.uk/members/zbyszek/figures\_pymol
```

### 3.1 De Schrödinger

Liste de toutes les commandes pymol, par thème ou alphabétique :

```
http://pymol.org/dokuwiki/doku.php?id=command
```