

TP modélisation par homologie - Partie pymod

1 Installer pymod

pymod est un plugin pour pymol pour faire de la modélisation par homologie sous pymol en intégrant plusieurs outils, comme BLAST, Clustal et modeller. pymod s'installe dans le home-directory ("dossier personnel") de chaque utilisateur, c'est pour cela que pymod n'est pas encore installé sur les PC en salle de TP. Une fois que vous êtes connecté avec votre compte (numéro d'étudiant + mot de passe UPMC), suivre ces instructions pour installer pymod :

1. Télécharger le "Linux Installer Bundle" ici :
`http://schubert.bio.uniroma1.it/pymod/download.html`
Sur cette page vous pouvez aussi télécharger le "User's Guide"
2. Ouvrir un Terminal (Ctrl+Alt+T), puis taper ces commandes (sans tout taper, on peut compléter les noms de fichiers ou dossiers avec la touche tabulation) :
`cd Téléchargements`
`unzip linux_pymod_2.0_installer_bundle.zip`
`pymol -c install_all.py`
Normalement vous devez voir à la fin "Installation successful"
3. Lancer ensuite pymol depuis le Terminal en tapant simplement :
`pymol`
4. Ouvrir le menu "Plugin->Plugin Manager", puis aller sur l'onglet "Settings". Cliquer sur "Add new directory..."
Puis choisir le dossier :
`/tmp/guest-Er6kqB/.pymod/pymol/plugins_pymod`
(la partie `/tmp/guest.../` sera différent, ce sera votre dossier personnel ("home directory"))
5. Fermer (Alt+F4) et relancer pymol
6. Aller de nouveau dans le menu Plugin, il devrait maintenant y apparaître une entrée "PyMod 2.0". Cliquer dessus, puis une fenêtre devrait s'ouvrir avec le titre "PyMod first session".
Lire le texte puis cliquer sur Ok.
Sans changer le chemin de dossier qui est votre home directory par défaut, cliquer sur "Submit".
Ensuite vous pouvez donner un nom à votre projet, puis cliquer sur "Submit".
Une fenêtre "PyMod 2.0" devrait s'ouvrir.

2 Premier exemple avec pymod

2.1 Les projets pymod

Il faut savoir que pymod 2.0 vient d'être mise-en-ligne et n'a pas encore toutes les fonctionnalités qu'on pourrait espérer. Notamment même si les projets sont sauvegardés sous `~/pymod`, la version actuelle de pymod ne permet pas de les ouvrir

ultérieurement. Mais le développeur principal de pymod m'avait dit que cette fonctionnalité devrait être implémentée dans pymod vers fin octobre 2016. En attendant il faut faire attention de ne pas fermer les fenêtres pymod, pymol et Terminal avant d'avoir terminé un projet, sinon il faudrait recommencer à zéro.

2.2 Importer la séquence cible (= "target sequence")

Ce premier exemple est pris du User's Guide de pymod (chapitre 5 Usage Examples). Il s'agit de générer des modèles par homologie d'une seule chaîne, ici la "dihydrofolate reductase of Mycobacterium avium".

1. Récupérer la séquence en format FASTA depuis le site www.uniprot.org avec l'identifiant O30463. Une fois sur la page de la séquence en question il faut cliquer sur le bouton bleu "Format" puis cliquer-droit sur "FASTA".
2. Sur cette même page d'uniprot chercher s'il y a déjà des structures 3D résolues pour cette protéine et noter leur code PDB. Cliquer aussi sur le lien vers le "ProteinModelPortal" et noter avec quel template 3D le modèle a été obtenu.
3. Encore sur la page d'uniprot chercher la partie "Sequence" puis noter la longueur de la séquence.
4. Créer un nouveau projet pymod : sous pymol aller dans le menu "Plugin -> PyMod 2.x". Dans la fenêtre pymod importer la séquence via "File -> Sequences -> Open from File"

2.3 Recherche et sélection de template 3D

5. Dans la fenêtre pymod : Sélectionner la séquence cible précédemment importé en cliquant sur son nom à gauche. La couleur devrait changer de rouge en vert. Sélectionner maintenant le menu "Tools -> Database Search -> BLAST". Cette interface permet de changer quelques options de BLAST, puis de faire une requête au serveur de BLAST (connexion Internet obligatoire). Ici on ne change pas les options, on remarque juste que pymod a déjà sélectionné par défaut la banque "PDB", c'est-à-dire les séquences des structures 3D de la PDB. Si vous faites un BLAST directement sur le serveur de la NCBI, alors la banque "nr" serait sélectionné par défaut, c'est-à-dire une très grande banque de séquences de protéines pour la plupart sans structure 3D associé. Cliquer sur "Submit" pour lancer BLAST avec votre séquence cible. Pymod peut rester figé le temps de la requête qui peut durer quelques minutes. A la fin une fenêtre "NCBI BLAST Output" s'ouvre et affiche la liste des séquences similaires trouvées (agrandir si nécessaire pour voir toutes les colonnes).
6. La première entrée a une identité de séquence de 100%, ce qui vient du fait qu'une structure 3D a déjà été résolue pour notre séquence cible. On n'utilisera bien sûr pas cette structure comme template, mais on pourra l'utiliser pour vérifier nos modèles générés. Pour comprendre la signification des colonnes "Query span" et "Subject span" on refait ce run de BLAST via le site de la NCBI directement.
7. Aller sur le serveur de BLAST de la NCBI et choisir un protein-BLAST. Pour faire une recherche on peut donner directement le code uniprot de notre séquence, puis il faut faire attention de bien choisir la bonne "database". Sur la page des résultats on voit d'abord des informations sur notre séquence "query", comme sa longueur et sa décomposition possible en domaines. En dessous il y a un graphique avec des lignes horizontales colorées. Chaque ligne correspond à une séquence similaire au query.

Qu'est-ce qu'on peut dire sur la partie C-terminal de notre séquence (les dix derniers résidus environ) ? En cliquant sur une ligne l'alignement qui se trouve plus bas de la page est affiché. Chercher l'entrée 4KL9. En regardant cet alignement que veut dire "Subject span" dans la fenêtre pymod ? Aller sur la RCSB PDB puis chercher l'entrée 4KL9, puis choisir l'onglet "Sequence". Est-ce que la numérotation des résidus dans la structure correspond au "Subject span" ?

8. Retour sur la fenêtre "NCBI BLAST Output" de pymod : Dans la longue liste de templates possibles on choisit de prendre deux templates de deux espèces différentes : *Bacillus anthracis* et *Moritella profunda*. Pour chaque espèce on prend la première entrée de la liste, c'est-à-dire les PDBs 3JW3 et 2ZZA. Cocher la case de ces deux entrées de la liste, puis cliquer sur "Submit". L'alignement de séquences s'affiche dans fenêtre de pymod. Regarder cet alignement tout le long de la séquence en utilisant la barre de défilement horizontal. Noter quelle partie de la séquence cible est alignée avec la séquence de 3JW3 et de 2ZZA, respectivement. A quoi correspondent les symboles *, . et : (voir cours) ? Colorier la séquence query avec "Display -> Color all Sequences -> By residue properties -> Polarity". Puis faire la même chose pour les deux séquences alignées en faisant un clique-droit sur le nom de chaque séquence, puis choisir le menu "Color" dans le menu local qui s'ouvre.
9. Importer les structures 3D des templates 3JW3 et 2ZZA en faisant un clique-droit sur le nom de chaque séquence, puis choisir "Structure -> Fetch PDB file" dans le menu local. On doit choisir si on veut importer tout le fichier PDB, c'est-à-dire toutes les chaînes avec ligands et cofacteurs ou seulement la partie qui est alignée avec la séquence query. Ici on choisit "Import all chains", comme on voudrait créer un modèle avec le cofacteur NADP. Si on aurait des templates multi-domaines, alors il aurait fallu choisir l'autre option pour ne garder que la partie des templates qui correspond au domaine à modéliser. Une fois les deux templates 3JW3 et 2ZZA importés on voit que les séquences sont remplacées par les séquences des fichiers PDB sans alignement avec notre séquence cible. Chaque PDB a deux chaînes, A et B. Est-ce qu'il s'agit de dimères biologiques ici ? Pour répondre regarder l'annotation sur la RCSB PDB pour les deux templates. Pour mieux comprendre la différence entre *Asymmetric Unit* et *Biological Assembly* lire ceci : pdb101.rcsb.org/learn/guide-to-understanding-pdb-data/biological-assemblies Comme on veut créer un monomère comme modèle, on peut effacer les deux chaînes B avec le menu local "Sequence -> Delete sequence".
10. Dans la fenêtre pymol on peut voir la structure 3D des quatre chaînes, coloriées par la couleur de la séquence dans la fenêtre pymod. Pour afficher les deux fenêtres en même temps, utiliser sous Ubuntu les raccourcis clavier Ctrl+Alt+numéro, avec numéro du pavé numérique : 4 = moitié gauche, 6 = moitié droite, 8 = moitié haut, 2 = moitié bas et 5 = plein écran. Pour centrer la vue 3D sur une chaîne, il suffit de cliquer sur son nom dans la fenêtre pymod pour l'activer (le nom devient vert) puis faire un clique avec le bouton du milieu (= la molette) de la souris. Pour cacher une chaîne on utilise aussi le bouton du milieu, mais sur un nom de chaîne désactivé, c'est-à-dire en rouge. Pour sélectionner un résidu d'une chaîne, on clique avec le bouton du milieu sur le résidu en question dans la fenêtre pymol.

2.4 Alignement

Après la sélection des templates, il faut encore les aligner correctement avec notre séquence cible. Cet alignement peut être guidé par un profil généré en superposant en 3D les templates.

11. Générer ce profil "structural" sélectionner les deux chaînes 3JW3_Chain_A et 2ZZA_Chain_A (elles doivent être en vert) puis choisir le menu "Tools -> Structural alignment -> CE Alignment". Laisser les options par défaut et cliquer sur "Submit". Les structures vont être superposées dans la fenêtre pymol et l'alignement qui en résulte au niveau séquence est montré dans la fenêtre pymod. Les symboles *, : et . se réfèrent maintenant à l'alignement de deux séquences sans notre séquence cible. Inspecter dans la fenêtre pymol l'alignement structural. Quelles sont les zones qui sont moins bien superposées ? Pour répondre à cette question activer l'affichage des séquences dans pymol avec le menu "Display -> Sequence". En sélectionnant un résidu dans la structure, celui-ci s'affiche aussi dans les séquences.
12. (Avancé) Pour répondre à la question précédente d'une manière plus quantitative, on peut aussi calculer le RMSF (Root Mean Square Fluctuation) sur les atomes C_α des deux templates superposés. Pour chaque résidu une valeur de RMSF est calculée et donne la distance spatiale entre ces deux résidus. En général on calcule le RMSF sur tout un ensemble de conformations, comme par exemple sur les milliers de conformations générées par une simulation de dynamique moléculaire, d'où le calcul d'une moyenne des fluctuations spatiales. Ici on n'a que deux conformations, donc on calcule simplement une distance spatiale par résidu entre les deux conformations. Pour effectuer le calcul on doit copier le script "rmsf_states.py" dans votre dossier du projet pymod actuel (on peut afficher le chemin avec `pwd` sous pymol). Ce script vient d'ici : <http://pldserver1.biochem.queensu.ca/~rlc/work/pymol/>
Puis on charge le script sous pymol avec la commande :
- ```
run rmsf_states.py
```
- Pour vérifier s'il est bien chargé, on fait appel à son aide intégrée :
- ```
help rmsf_states
```
- Comme on peut lire dans cette aide, on doit fournir un objet "multi-state". Pour faire cela, on sauvegarde les positions des C_α de nos deux templates :
- ```
save 3JW3_CA.pdb, 3JW3_Chain_A and name CA and resi 2-161
save 2ZZA_CA.pdb, 2ZZA_Chain_A and name CA and resi 2-161
```
- On a ici seulement sauvegardé les résidus à partir du résidu 2 jusqu'au résidu 161, comme cette sélection correspond à l'alignement sans gap qu'on avait obtenu avec CE-alignment sous pymod. Vérifier cela en inspectant l'alignement à l'aide de la souris : on mettant la souris au dessus d'un résidu, son numéro et nom s'affiche en bas dans la fenêtre pymod.
- Ensuite il y a encore une correction à faire sur le fichier 3JW3\_CA.pdb : Il y a deux atomes  $C_\alpha$  pour le résidu MET-37, il s'agit ici de deux conformations alternatives. Ouvrir ce fichier PDB avec un éditeur de texte (gedit par exemple, pas LibreOffice !), puis effacer une des deux lignes MET-37.
- Maintenant on charge dans pymol ces deux fichiers PDB pour créer un seul objet "multi-state" qu'on nomme "mov" :
- ```
load 3JW3_CA.pdb, mov
load 2ZZA_CA.pdb, mov
```
- Pour n'afficher que l'objet "mov" on clique dans pymol sur "all", puis sur "mov". Ensuite, comme on n'a ici uniquement les atomes C_α de pymol, les affichages classiques (cartoon, lines etc) ne marchent pas. Essayer l'affichage "spheres" :
- ```
show spheres
```
- Puis cliquer sur le bouton "C" à côté de "mov" pour choisir une coloration. Choisir "spectrum -> b-factors". On voit que pour l'instant il n'y a pas de valeurs B-facteur, ce qui donne une seule couleur pour tous les résidus.
- Lancer maintenant le script rmsf\_states sur l'objet "mov" :

### **rmsf\_states mov**

Les couleurs devraient avoir changé en arc-en-ciel en indiquant en bleu des RMSF faibles et en rouge des RMSF fortes. Repérer les zones de RMSF fortes dans la séquence et noter-les. Pour cela on doit activer l'affichage de la séquence, via le menu pymol "Display -> Sequence". Par contre on voit que la séquence affichée de "mov" est un mélange des séquences des deux templates. Pour les afficher séparément, on doit créer deux objets séparés :

### **split\_states mov**

Pour sauvegarder les valeurs RMSF traduites en B-facteurs, sauvegarder ces deux objets :

**save 3JW3\_Bfac.pdb, mov\_0001**

**save 2ZZA\_Bfac.pdb, mov\_0002**

Pour revenir à l'affichage en cartoon de la question précédente on clique sur "all" puis "3JW3\_Chain\_A" et "2ZZA\_Chain\_A", puis :

**hide everything**

**show cartoon**

13. Pour aligner notre séquence cible à l'alignement des deux templates, on utilise des méthodes d'alignement séquence-profil, comme celles qui sont implémentées dans pymol : ClustalW, MUSCLE ou Clustal Omega. Le "profil" est ici simplement l'alignement obtenu lors de la question précédente. On sélectionne donc notre séquence cible et les deux séquences des templates. Puis, on choisit ici Clustal Omega avec le menu "Tools -> Profile alignment -> Clustal Omega". Laisser les options par défaut, puis cliquer sur "Submit".

Sauvegarder l'alignement obtenu en ouvrant le menu local (clique-droit sur "Alignement") et en sélectionnant "Edit Cluster -> Save Alignment to File". Sauvegarder l'alignement dans le dossier de votre projet pymol.

Charger l'alignement sauvegardé pour dupliquer l'alignement, afin de le modifier manuellement tout en gardant "l'original" : Choisir "File -> Alignment -> Open From File" puis choisir le fichier qu'on vient de sauvegarder.

14. Inspecter l'alignement obtenu sur l'alignement original ("pas imported").  
*ATTENTION : On peut manuellement changer l'alignement avec la souris : on sélectionne un résidu avec la souris et on reste appuyé avec le bouton gauche, tous les résidus à partir de ce résidu sont déplacés avec la souris. Faire ces opérations uniquement sur l'alignement dupliqué ("imported")*

Quelles zones ont une faible identité de séquence ? Pour quelles zones de notre séquence cible on n'a aucune information structurale à partir des templates ? Vérifier les positions des gaps dans la structure, soit avec clique-milieu sur les résidus dans pymol, soit en changeant la couleur de ces résidus dans pymol, par exemple :

**color blue, resi 51-52**

Est-ce que cela sera facile d'insérer les résidus de notre séquence cible sur ces deux structures ?

15. **(Avancé)** Essayer d'améliorer manuellement l'alignement avec la souris sur l'alignement dupliqué ("imported") en gardant l'alignement original inchangé.
16. Les derniers 13 résidus de notre séquence cible n'ont aucune correspondance dans l'alignement avec les templates. Ces résidus seront trop difficiles à modéliser, pour cela on préfère de les enlever : Cliquez-droit sur le nom de notre séquence cible, puis "Sequence -> Edit Sequence". Effacer les 13 derniers résidus.

## 2.5 Génération du modèle

En prenant l'alignement original cible <-> templates obtenu par Clustal Omega, on génère un modèle avec MODELLER.

17. Sélectionner la séquence cible dans pymod puis choisir le menu "Tools -> Homology modeling -> MODELLER". Dans les options on sélectionne d'abord les deux templates dans "Use as Template", puis pour 3JW3 on sélectionne "Do not use any heteroatomic residue" et on n'inclut pas les molécules d'eau de 3JW3. Pour 2ZZA par contre on sélectionne "Select single heteroatomic residues, puis on coche "NAP". On coche aussi pour les 157 molécules d'eau à inclure. Voir le "User's Guide" de pymod pour plus de détail sur ces options (partie 5.1.4 Model building). Avant de lancer Modeller, on choisit encore quelques options dans l'onglet "Option" : On met le "Optimization Level" à "High" et on choisit le "DOPE Score" dans "Color models by". Après vérification de toutes les options on clique sur "Submit". Le calcul peut prendre quelques minutes.

## 2.6 Analyse du modèle

Une fois la génération du modèle terminé correctement, les résultats seront affichés dans pymod et pymol. Dans la fenêtre pymol on voit le modèle qui est coloré par le score DOPE. Le modèle est superposé au templates qui sont également évalués par le score DOPE. Ces scores par résidu sont aussi affichés dans une fenêtre à part. En cliquant sur un endroit de la courbe, le résidu est sélectionné sous pymol.

18. Identifier les zones avec grandes valeurs du score DOPE, c'est-à-dire les zones potentiellement mal prédites. Regarder ces zones dans les structures. Comparer ces zones par rapport à votre inspection de l'alignement obtenu avec Clustal Omega (position des gaps, etc).
19. Comparer le modèle obtenu avec la structure expérimentale de notre cible : Taper dans pymol :  
**fetch 2W3W**  
Importer la séquence de cette structure dans pymod avec le menu "File -> Sequences -> Import PyMOL Objects". Puis pour superposer cette structure de référence avec notre modèle, on sélectionne la séquence cible et la séquence 2W3W dans pymod et on choisit "Tools -> Structure alignment -> Superpose".  
Dans pymol n'afficher que le modèle et 2W3W, puis inspecter l'accord. Est-ce que le score DOPE a bien prédit les zones mal prédites ?
20. **Pour aller plus loin** : revenez sur l'étape d'alignement cible <-> templates et essayer de trouver un alignement qui améliore les zones mal prédites. C'est-à-dire pour chaque nouveau alignement il faut régénérer un modèle avec Modeller, puis le comparer avec la référence.

## 3 Deuxième exemple avec pymod

Le "User's Guide" de pymod contient un deuxième exemple comme tutoriel : la modélisation par homologie d'un complexe de protéines. Suivre les instructions (en anglais) du "User's Guide", partie 5.2.