Malo Leprohon
M2BI

Projet court :
Analyse de la communication allostérique à l'aide d'un alphabet structural

Introduction :

Les protéines possèdent une structure tridimensionnelle complexe que l'on peut diviser en une strucutre primaire, secondaire, tertiaire et quaternaire. Les éléments de la structure secondaire les mieux connus sont les hélices alpha et les brins bêta. Cependant ce que l'on appelle « random coil » est beaucoup moins bien connu et caractérisé. La création d'alphabet structuraux a pour but d'essayer de simplifier et caractériser de manière plus précise les différentes structures secondaires. Cela, en se basant sur l'arrangement local de la chaîne polypeptidique, notamment via l'étude des angles dièdres autour des liaisons peptidiques. Barnoud et al (1). ont notamment développé un alphabet structural de 16 lettres se basant sur les angles dièdres sur une fenêtre de 5 acides aminés.

L'allostérie est un phénomène que l'on pourrait résumer à l'influence d'un changement conformationnel (pouvant être dû à l'interaction avec un ligand) au niveau d'une région sur la structure de régions distantes de la même protéine.
Le but du projet est de servir de l'alphabet structural Blocs Protéiques pour essayer d'observer une possible allostérie, via le développement d'un programme qui mesurerait l'information mutuelle entre les différentes position d'une séquence protéique. En effet l'information mutuelle est une mesure de la dépendance entre deux variables aléatoires. Si dans une séquence de Blocs Protéiques on observe fréquemment des changements simultanés et cohérents à deux positions différentes alors l'information mutuelle entre ces deux positions sera élevée. Si de plus ces positions sont éloignées spatialement alors on peut supposer qu'il y a un mécanisme d'allostérie derrière ces changements de structure locale.

Matériel et méthodes :

Versions des modules utilisés et données de départ:

Le programme pb_seq_mi.py a été développé sur python 3.7.4 et avec les packages suivant : pbxplore 1.3.8, numpy 1.17.2, pandas 0.25.1, matplolib 3.1.1 et seaborn 0.9.0.

Le programme est disponible sur git hub : https://github.com/leprohonmalo/projet_court

Les données d'example sont issues de trois dynamiques moléculaires du domaine Calf-1 sous trois phénotypes différents : Wild Type, L653R, L721R. Les données sont disponibles dans le fichier joint au rapport, dans le répertoire raw_data.

Fonctionnement du programme :

Pour fonctionner le programme à besoin de données de dynamiques moléculaires sous la forme d'un fichier de données de trajectoire et d'un fichier de données de topologie.
Le programme utilise ensuite le module pbxplore développé pour pouvoir exploiter ces données et extraire des séquences de Blocs Protéiques de la dynamique moléculaire pour des frames réparties à

intervalles réguliers. Différentes fréquences d'occurrence des Blocs Protéiques aux différentes positions de la protéine sont ensuite calculées afin de pouvoir calculer l'information mutuelle entre chaque position de la séquence analysée. Le programme écrit ensuite les différentes tables de fréquences et d'informations mutuelles dans divers fichiers dans un répertoire de sortie spécifié par l'utilisateur.

Calcul de l'information mutuelle :

L'information mutuelle est une mesure de la dépendance entre deux variables aléatoires X et Y. On peut la calculer dans le cas de variables discrètes avec la formule suivante :

$$I(X;Y) = \sum_{x,y} P(x,y) \log \frac{P(x,y)}{P(x)\,P(y)}$$

Ou I(X,Y) est l'information mutuelle entre les variables X et Y, P(x) la probabilité de l'évènement x , P(y) la probabilité de l'évènement y et P(x,y) la probabilité de l'évènement (x et y). Dans le cas présent les variables X et Y représentent le contenu en Bloc Protéique des positions i et j. Les évènements x et y représentent la présence d'un Bloc Protéique précis aux positions i et j. On ne connaît pas les lois de probabilités qui régissent ces évènements et on s'appuie donc sur les fréquences observées de ces évènements pour le calcul de l'information mutuelle bien que la fiabilité de cette estimateur repose beaucoup sur le nombre séquences disponibles. Pour nos données on dispose de 501 séquences de Blocs Protéiques par dynamique moléculaire.

Résultats:

Les différentes matrices d'information mutuelle pour les trois phénotypes ont une distribution de valeurs entre 0 et 0,35 (Figures non disponibles voir en annexe). Dans les trois cas on observe les valeurs les plus élevées (supérieures à 0,25) aux niveau de positions proches. Cela s'explique par le fait qu'il reste plus probable qu'un changement de conformation locale est un impact plus important sur la région locale plutôt que sur une région plus éloignée dans la séquence. Dans aucun cas on n'observe dans aucun cas des valeurs aussi élevées pour des positions éloignés ou l'information mutuelle ne dépasse pas 0,15-0,20. Cependant ces valeurs restent plus élevés que l'information mutuelle entre la plupart des positions éloignées et ne sont observables qu'au niveau de régions précises (par exemple entre les régions 20-26 et 104-110 dans le cas de Calf-1 Wild Type). On peut donc penser que ces valeurs sont suffisamment élevées pour un cas biologique pour pouvoir supposer des relations faibles d'allostéries dans ces régions. Il faut toutefois vérifier que ces régions sont belle et bien distantes spatialement.

On observe également que les profils d'informations mutuelles des phénotypes mutants sont assez différents de celui du profil Wild-Type. Cela suppose que les mutations de ces phénotypes peuvent avoir un impact assez important sur la dynamique de la structure de Calf-1.

Discussion :

De manière générale on observe pas réellement de valeur d'informations mutuelles importantes. Cela peut être due à la complexité de la réalité des systèmes biologiques. Cependant on arrive à observer quelques tendances dans les différents profils que ce soit au niveau locale ou entre régions éloignées. Cela montre que l'information mutuelle peut être une donnée intéressante à exploiter

pour étudier et détecter des dynamiques structurales, et peut être de l'allostérie. À noter toutefois que les fréquences utilisées pour le calcul de l'information mutuelle ne sont pas d'excellents estimateurs de probabilité si les effectifs (nombres de séquence) ne sont pas importants.

Bibliographie :

1. Barnoud J, Santuz H, Craveur P, Joseph AP, Jallu V, de Brevern AG, Poulain P, PBxplore: a tool to analyze local protein structure and deformability with Protein Blocks *PeerJ* 5:e4013 https://doi.org/10.7717/peerj.4013 (2017).

Annexe

Usage:

Le dossier projet_court contient un fichier .yml contenant les différents prérequis et pouvant être exploité par conda pour créer un environnement viable pour le programme.
Le dossier raw_data contient des données pour trois dynamiques moléculaires du domaine Calf-1 .
Le dossier results contient les exemples de résultats pour les trois dynamiques.
Le dossier doc contient le présent rapport.
Le dossier src contient le programme pb_seq_mi.py

Exemple d'utilisation sur la dynamique moléculaire Calf-1 WT à partir du dossier projet_court :

python3 src/pb_seq_mi.py --traj raw_data/DM_Calf-1_WT/production_long1/md.trr --topo raw_data/DM_Calf-1_WT/production_long1/md.gro -o results/DM_Calf-1_WT/

--traj correspond au fichier de trajectoire de la dynamique moléculaire.
--topo correspond au fichier de topology de la dynamique moléculaire.
-o (ou –output) correspond au répertoire de sortie ou seront écrits les différentes tables et figures.

Le paramètre -h ou –help permetra d'afficher l'aide à l'utilisation.
La documentation des différentes fonction est disponible dans le script pb_seq_mi.py sous forme de docstring.

Difficultés rencontrées :

Le programme devait initialement renvoyer une heatmap et un réseau représentant les différentes positions reliés entre elles par des liens ayant pour poids leur information mutuelle.

Le programme renvoi bien une heatmap mais celle ci possède de nombreux défauts visuels (dimensions, manque de titres) et n'a donc pas pu être intégrée au rapport. Cela est due aux contraintes de temps et aux manques de connaissances et d'expériences sur plusieurs modules utilisés dans ce programme (matplotlib, seaborn, pandas). Pour ces même raisons le réseau n'a pu être réalisé.

Bien que simple au premier abord, l'implémentation du calcul de l'information mutuelle s'est avérée être plus difficile que prévu. Cela en a toutefois fais une bonne pratique.

Le programme aurait pu être plus développé sur plusieurs aspects :
- Implémentation d'une option pour pouvoir également exploiter des données de fichiers pdb
- Implémentation peut être utile d'une gestion des différentes chaînes polypeptidiques sur plusieurs tables dans le cas d'hétérodimères pour ne pas fausser les fréquences de Blocs Protéiques en

combinant les différentes chaînes d'une même table (pas sur que le programme survivrait si les longueurs des chaînes sont en plus différentes).

Toujours pour ces même raisons la documentation est simplement sous forme de docstring dans le script.

Fonctionnement du programme :

Schéma workflow du programme :