Timothée O'Donnell
GHARBI Houssem
M1 BiBs

Projet Docking

I) Introduction :

La formation de complexe à partir de deux protéines est un phénomène essentiel à la compréhension des interactions biologiques. Pour de nombreuses raisons ces phénomènes sont complexes. Une protéine adopte une conformation tridimensionnelle qui peut être modifiée lors de la formation d'un complexe, de plus différents types d'interaction régissent la formation d'un complexe et la diversité moléculaire des protéines complexifie toute modélisation.

Cependant il est nécessaire d'étudier ce problème, de nombreux domaines comme celui de la santé bénéficie d'une meilleure compréhension de celui-ci. En effet la capacité à prédire la formation d'un complexe entre un principe actif et sa cible permet la création de nouveaux médicament allopathique.

Les méthodes d'étude les plus précises pour visualiser la formation d'un complexe sont expérimentale. Des méthodes comme la cristallographie aux rayons X permettent la visualisation tridimensionnelle de certains complexes. Cependant cette méthode est onéreuse, relativement lente, et ne marche que si le complexe étudié est cristallisable.

Pour combler ces lacunes une utilisation d'algorithme prédictifs pour déterminer la conformation tridimensionnelle adoptée par un complexe est nécessaire. Ces algorithmes établissent un score prédictif qui évalue comparativement des conformations de docking données entre elles. Ces conformations sont obtenues par disposition itérative d'une des deux protéines autour de l'autre de sorte qu'un grand nombre de possibilités soit évaluées.

Il existe quatre grandes classes d'algorithme de scoring : empirique, statistique machine-learning et ceux basés sur des champs de force.

II) Modélisation et algorithme :

Notre projet utilise un algorithme de type champs de force. Ces méthodes de scoring tentent de donner un score proportionnel au coût énergétique du maintien du complexe par rapport aux protéines individuelles. Donc plus le score est grand,

moins la conformation est probable en tant que solution car maintenir cette conformation est énergétiquement “coûteux”.

Les évaluations de score énergétique sont basées sur la physique non quantique, dite Newtonienne, ou l’on étudie des particules avec une masse et des propriétés. Ces particules interagissent en suivant certaines lois ce qui nous donne une base pour établir des algorithmes de score.

1) Algorithme de scoring de Cornell et al. :

Notre premier algorithme utilise les lois décrivant les interactions de van der Waals et la loi de Coulomb. Il est basé sur les termes nom liés de l’algorithme décrit dans Cornell et al. 1995.

$$E_{ij} = \frac{A_{ij}}{R_{ij}^{8}} - \frac{B_{ij}}{R_{ij}^{6}} + f\frac{q_i q_j}{20R_{ij}}$$

avec f = 332.0522
i = atomes du Rec et j atomes du Lig
R_{ij} = dist entre i et j
q_i = charge de i – idem pour j
Voir Cornell et al, 1995 pour A_{ij} et B_{ij}

Figure1 : Equation de Cornell et al. (Terme non lié)

La formule originelle a été modifiée pour améliorer les résultats dans le contexte du docking, Cornell et al, 1995 s'intéressant à l’état énergétique d’une molécule et non d’un complexe.

2) Algorithme modifié :

Nous avons également utilisé deux autres algorithmes en ajoutant un terme supplémentaire : la désolvation. Pour cela nous nous sommes inspirés de la documentation d’un algorithme de docking préexistant : autodock (http://autodock.scripps.edu/)

Cet algorithme est calibré pour de petites molécules mais gère le docking protéine. Autodock utilise un algorithme composé de termes basée sur des interactions physique auxquelles il applique des coefficients obtenus par régression, ceci en fait un algorithme de machine Learning car les coefficients sont basés sur des analyse de donnés itérative. Néanmoins les termes sans les coefficients sont basés sur des équations de champs de force de physique.

$$\Delta G = \Delta G_{\text{vdW}} \sum_{i,j} \left(\frac{A_{ij}}{r_{ij}^{12}} - \frac{B_{ij}}{r_{ij}^{6}} \right) + \Delta G_{\text{hbond}} \sum_{i,j} E(t) \left(\frac{C_{ij}}{r_{ij}^{12}} - \frac{D_{ij}}{r_{ij}^{10}} + E_{\text{hbond}} \right) + \Delta G_{\text{elec}} \sum_{i,j} \frac{q_i q_j}{\varepsilon(r_{ij}) r_{ij}} + \Delta G_{\text{tor}} N_{\text{tor}} + \Delta G_{\text{sol}} \sum_{i_C,j} S_i V_j e^{(-r_{ij}^2/2\sigma^2)}$$

Figure 2 : Algorithme de l'outil autodock

Le score de desolvation cherche à mettre en valeur la différence entre l'énergie d'interaction des deux protéines avec le solvant et l'énergie d'interaction du complexe avec le solvant :

$$\Delta G_{desolv} = W_{desolv} \sum_{i,j} (S_i V_j + S_j V_i) e^{\frac{-r_{ij}^2}{2\sigma^2}}$$

Nous avons retiré le coefficient de calibrage Wdesolv de cette équation pour nous affranchir du biais et conserver uniquement la partie physique Newtonienne.

i : index of atoms in the ligand
j : index of atoms in the receptor
Wdesolv : linear regression coefficient or weight for the desolvation free energy term
Si = solvation term for atom i
Vi : atomic fragmental volume of atom i
rij : distance between atom i and atom j (in Å)
σ : Gaussian distance constant = 3.5 Å

$$S_i = a_i + k\,|q_i|$$

ai: atomic solvation parameter, ASP
k: charge-based atomic solvation parameter, QASP = 0.010 97
qi: partial atomic charge on atom i

Pour ce qui est du troisième algorithme nous avons simplement pris le terme de Van der Waals “traditionnel” (tel que le premier terme de la figure 2) car les modifications faites au exposants (8 au lieu de 12) pourrait causer une sous-estimation d’un des termes par rapport à un autre. Il nous a paru intéressant de comparer le “nouvel” algorithme de Cornell à “l’ancien” avec ce nouveau terme.

Ces trois algorithmes forment les fonctions de scoring trouvées respectivement dans les fichiers :

- NewScorringCornell.py
- NewScorringCornellAndDesolvation.py
- OldScorringCornellAndDesolvation.py

3) Programmes :

Afin de pouvoir utiliser les algorithmes sur un fichier test de 948 solutions présélectionnée de position potentielle d’un ligand vis à vis d’un récepteur fixe. Nous avons créé un programme qui permet d’implémenter les différents scoring. Pour y parvenir le programme prend un certain nombre d’arguments qu’ils lui permettent de générer des données sortantes.

3.1) RunScoring.py :

- Arguments :

Nous nous sommes limités aux arguments que nous pensions impératifs au bon fonctionnement du programme :

-in : un dossier où se trouve les fichiers correspondant aux fichiers pdb de la protéine non fixée générée autour de la protéine fixée, leurs noms doivent se terminer par “_NumeroDeLaSolution_DP.pdb”. (“NumeroDeLaSolution” étant à remplacer par le chiffre correspondant chaque solution). Nous avons conservé cette formulation pour qu’uniquement les fichiers voulus soit intégré dans l’algorithme de scoring.

-out : le dossier ou seront Assemblé tous les fichiers de données sortantes. Il peut être préexistant ou nom de telle sorte qu’il est possible de rassembler les sorties avec celles d’un autre programme. Il est cependant déconseillé de nommer les dossiers de sortie des différents programmes de la même manière car certains fichiers sortants vont en écraser d’autres.

-prog : le programme contenant l’algorithme sélectionné : (NewScoringCornell.py, NewScoringCornellAndDesolvation.py ou OldScoringCornellAndDesolvation.py). Seules ces valeurs sont possibles sauf si un programme compatible est ajouté dans le répertoire contenant runScorring.py. Plusieurs programmes peuvent tourner en parallèle.

-pdbF : le fichier pdb de la protéine définie comme fixe dans sa forme native.

-chainF : la chaîne correspondant à la protéine définie comme fixe dans les fichiers pdb (ex : B).

-chainV : la chaîne correspondant à la protéine présente dans les solutions multiples

-pdbV : Argument optionnel qui correspond au fichier pdb de la conformation native de la protéine variable dans le complexe alignée sur le fichier pdb de la protéine fixe. Si cet argument n'est pas rempli on ne peut pas calculer les différents RMSD entre le complexe avec le meilleur score et le complexe natif. Ceci est utile uniquement pour évaluer l'algorithme de scoring utilisé car la structure tridimensionnelle du complexe est déjà connue. Deux autres sorties supplémentaires sont également présente un fichier pdb contenant l'interface native colorée grâce au bfactor ainsi que les mêmes acides aminés colorés similairement. Ces deux fichiers permettent de comparer la position de l'interface réelle dans le complexe avec le meilleur score et dans le véritable complexe obtenu expérimentalement.

- Sorties systématiques :

- Un dossier contenant les fichiers pdb, fichiers créés par le programme, des complexes de toutes les solutions envisagé est créé. Ce format de fichier permet de simplifier le programme d'analyse. On conserve ce dossier pour pouvoir visualiser les complexes de chaque solution.

- Le pdb du complexe avec le meilleur score est copié et collé dans le dossier de sortie fourni en -out. Cette sortie évite à l'utilisateur d'avoir à fouiller dans le dossier mentionné précédemment pour identifier le pdb de la solution avec le meilleur score.

- Le fichier Scoring regroupe les identifiants et les score du meilleur au moins bon, c'est à dire du score énergétique le plus petit au plus grand.

- Sorties optionnelles :

Remarque : les sorties suivantes sont présentes uniquement si l'on peut comparer à un complexe de référence c'est à dire si l'argument -pdbV est utilisé :

- RMSD.out regroupe les RMSD sur les carbones alpha et celles sur tous les atomes du complexe entier du ligand seul et sur uniquement les atomes de l'interface.

- Les fichiers pdb du complexe natifs et du meilleur score sont également copiés et modifiés de telle sorte à pouvoir observer le déplacement de l'interface du complexe de référence sur le fichier avec le meilleur score. Ceci est fait en modifiant le bfactor des atomes dans le fichiers pdb, toutes valeurs précédentes sont donc perdues et si l'on souhaite les observer il faut se référer à la sortie systématique du complexe avec le meilleur score.

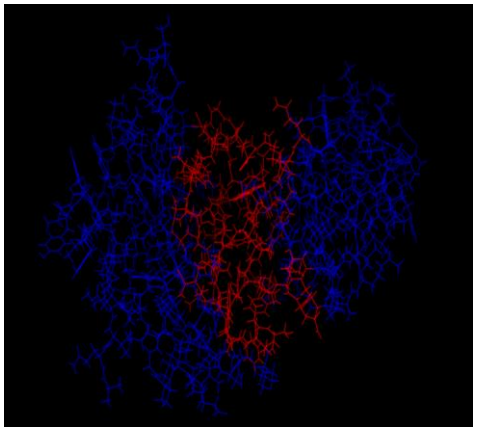

Illustration de complexe de référence

3.2) runRMSDFull.py

Le fichier RMSDFull.py prend les mêmes arguments que la fonction précédente mais sans l'argument -prog, de plus tous les arguments sont obligatoires car un complexe de référence est nécessaire pour le RMSD.

Ce programme rend un fichier dans lequel les solutions ont un score qui correspond à un RMSD de comparaison entre carbones alphas de deux complexes entiers. Les solutions sont classées de la meilleure à la moins bonne c'est à dire par RMSD croissant. Cette étape nous permet par comparaison de savoir si notre programme a sélectionné la meilleure solution parmi celles qui lui sont proposées.

Ce classement permet une appréciation du modèle mais de doit pas être utilisée pour effectuer un calibrage du modèle. Le problème étant que l'on pourrait introduire un biais c'est à dire que le modèle deviendrait performant pour trouver des solutions dans des données de protéines issus d'une population similaire à celle qui ont été utilisée pour le calibrage. Ceci est l'une des raisons pour lesquelles runRMSDFull.py est un fichier séparé l'autre étant que le programme principal n'oblige pas à fournir un complexe de référence.

```
1  rec_nat_lig_74.pdb     RMSD Complex entier (CA) : 32.002201
2  rec_nat_lig_866.pdb    RMSD Complex entier (CA) : 32.230526
3  rec_nat_lig_291.pdb    RMSD Complex entier (CA) : 32.615096
4  rec_nat_lig_33.pdb     RMSD Complex entier (CA) : 32.737591
5  rec_nat_lig_392.pdb    RMSD Complex entier (CA) : 32.808714
```

III) Implémentation et analyse :

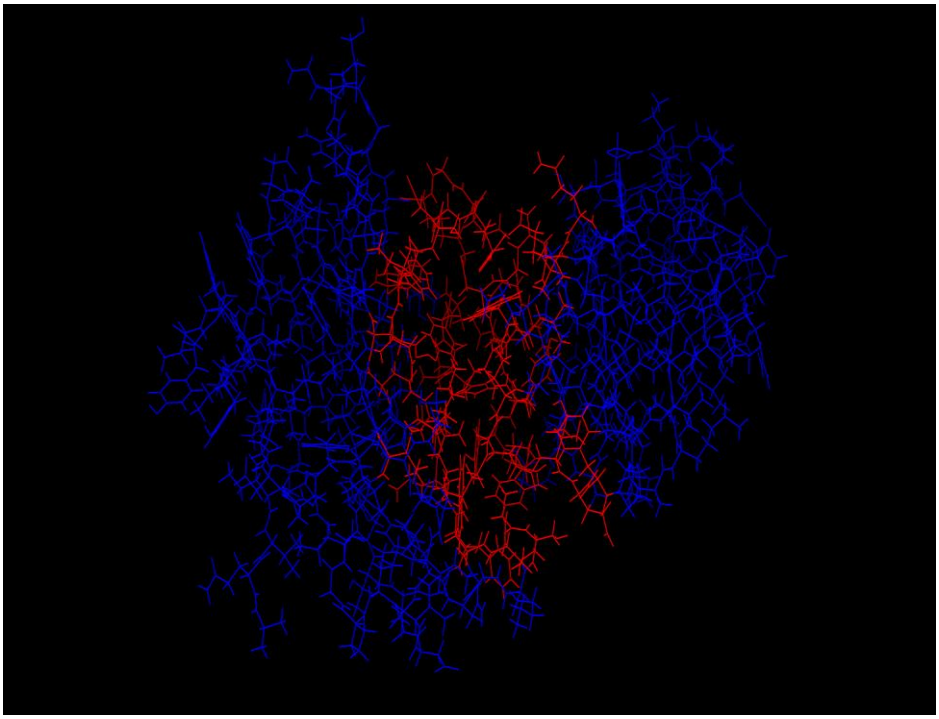

Figure 3 : Image issue de pymol du complexe de référence

1) Implémentation de NewScoringCornell.py :

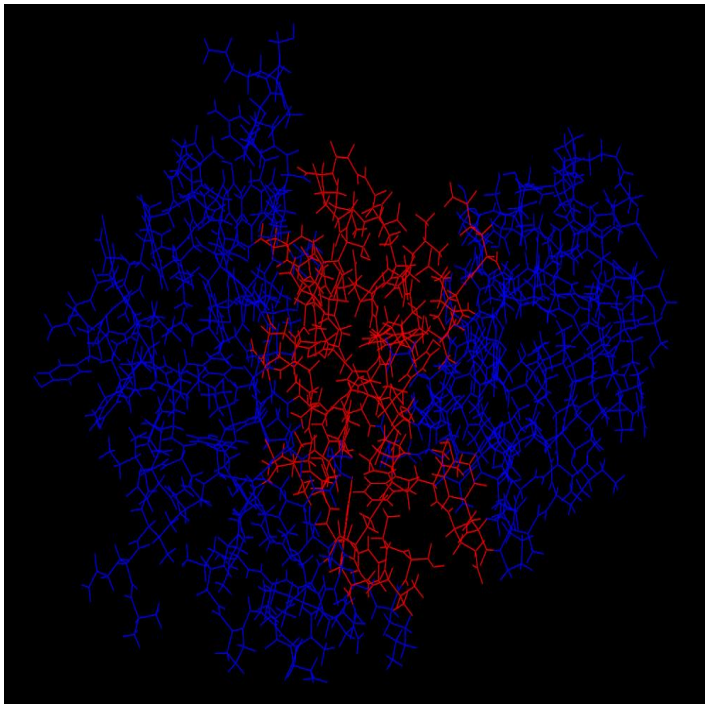

Figure 3 : Image issue de pymol du complexe de prédit par NewScoringCornell.py

```
Meilleur Solution: rec_nat_lig_28.pdb

-RMSD Complex entier (CA): 0.642360
 -RMSD Complex entier (tous les atomes): 0.785420

-RMSD Ligand (CA): 0.966612
 -RMSD Ligand (tous les atomes): 1.173359

-RMSD Interface native (CA): 0.558701
 -RMSD Interface native (tous les atomes): 0.636650
```

Le complexe trouvé est très proche du complexe natif particulièrement au niveau du squelette de carbone Alpha de l'interface dans lequel on trouve un RMSD extrêmement faible De Plus l'implémentation de la fonction runRMSDFull.py nous indique que cette solution est la meilleure dans l'échantillon disponible.

L'ajout d'un terme à cette fonction ne pourras donc pas nous permettre de sélectionner un meilleur individu dans cet échantillon. Si une solution moins bonne est sélectionnée cela sera dû au nouveau terme car la seule différence entre l'algorithme utilisé ici et le suivant est l'addition du score lié à la desolvation.

Le complexe trouvé est très proche du complexe natif particulièrement au niveau du squelette de carbone Alpha de l'interface dans lequel on trouve un RMSD extrêmement faible De Plus l'implémentation de la fonction runRMSDFull.py nous indique que cette solution est la meilleure dans l'échantillon disponible.

L'ajout d'un terme à cette fonction ne pourras donc pas nous permettre de sélectionner un meilleur individu dans cet échantillon. Si une solution moins bonne est sélectionnée cela sera dû au nouveau terme car la seule différence entre l'algorithme utilisé ici et le suivant est l'addition du score lié à la desolvation.

2) Implémentation de NewScoringCornellAndDeslovation.py :

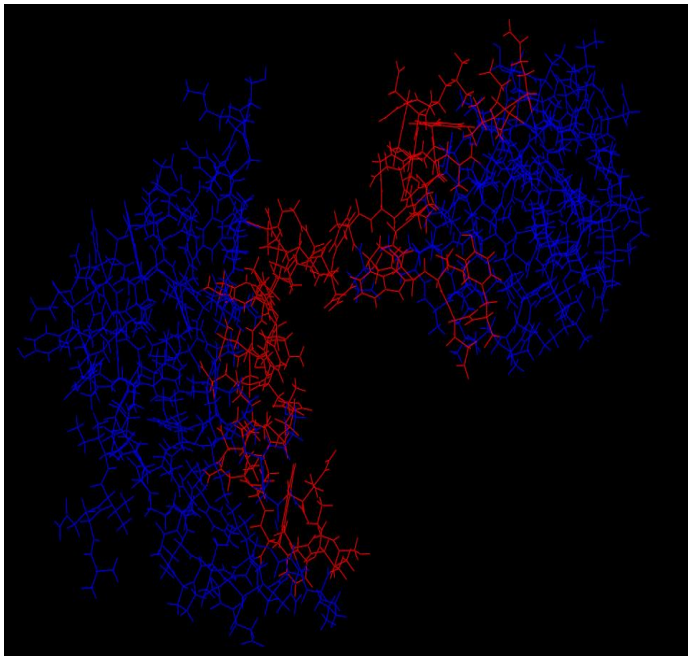

Figure 4 : Image issue de pymol du complexe de prédit par NewScoringCornellAndDesolvation.py

```
Meilleur Solution: rec_nat_lig_375.pdb

-RMSD Complex entier (CA): 12.087043
 -RMSD Complex entier (tous les atomes): 12.097480

-RMSD Ligand (CA): 18.188360
 -RMSD Ligand (tous les atomes): 18.072727

-RMSD Interface native (CA): 14.607142
 -RMSD Interface native (tous les atomes): 14.130967
```

On observe sur ce résultat par rapport au précédent que les interfaces se font toujours faces mais qu'elles se sont éloignées. Il est probable que le terme d'énergie de désolvation soit surexprimé. On peut d'ailleurs observer l'augmentation du score pour la meilleure solution par rapport à la précédente. Les RMSD ont tous augmentés ce qui est observable sur la figure tridimensionnelle qui est nettement plus distante du complexe de référence que la solution de l'algorithme précédent.

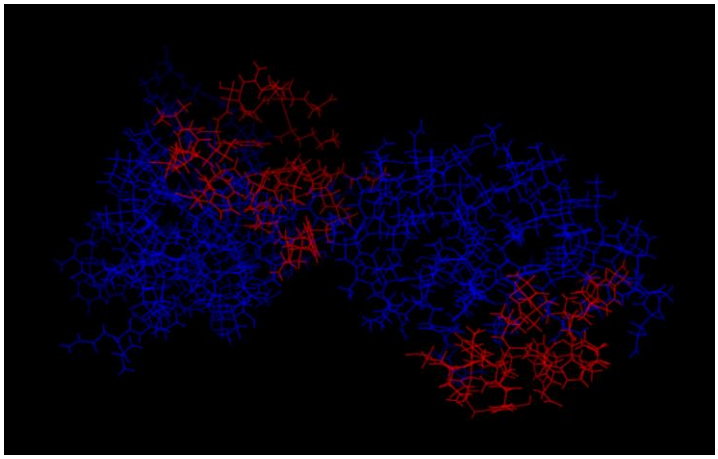

Figure 5 : Image issue de pymol du complexe de prédit par OldScoringCornellAndDesolvation.py

```
Meilleur Solution: rec_nat_lig_593.pdb

-RMSD Complex entier (CA): 20.631256
 -RMSD Complex entier (tous les atomes): 20.620103

-RMSD Ligand (CA): 31.045536
 -RMSD Ligand (tous les atomes): 30.804886

-RMSD Interface native (CA): 23.766994
 -RMSD Interface native (tous les atomes): 23.273659
```

En augmentant le facteur de puissance lié à la distance, le score de le RMSD de la meilleure solution est augmenté. Le terme énergétique de van der Waals a pris une importance trop grande par rapport aux deux autres termes. On observe ceci notamment car les liaisons ioniques présentent dans l'interface du complexe de référence qui ajoutaient un terme de score négatif sont devenu négligeable. En conséquence, les interfaces ne sont plus en face l'une de l'autre.

Les RMSD calculés de cette solution sont les plus mauvais parmi les trois solutions, ce n'est pas surprenant car les deux protéines sont distantes et tournées par rapport à la référence.

IV) Conclusion :

Le meilleur des trois algorithmes de prédiction pour l'échantillon de 948 solutions fournis est le premier. Cet algorithme a sélectionné la meilleure solution parmi celles qui était disponible. Le fait d'ajouter le terme de desolvation ou de rebasculer sur la formule d'origine du terme de Van Der Waals (Cornnel et al. 1995) ont réduit la capacité du modèle à trouver la meilleure solution.

Les termes de scores étant additifs, on peut déduire que le terme de désolvation nous a éloigné de la bonne solution pour cet échantillon. Si on généralise les observations sur cet échantillon, on sélectionnerait le premier algorithme comme le meilleur des trois présentés.

Pour conserver le terme de deslovation, on pourrait introduire des coefficients de régression. Cependant, ces coefficients seront produits à partir d'échantillon pour lesquelles on a complexe cristallisé. Les protéines cristallisables étant présumément issue d'une population non homogène à celle des protéines en générale ceci introduirait un biais sur les modèles.