

# Protein Interactions Calculator implémentation

Léo Maury, Raphaël Bodin

## 1) Abstract :

Protein Interactions Calculator<sup>1</sup> (PIC; <http://pic.mbu.iisc.ernet.in>) est un serveur web qui permet le calcul des différents types d'interactions importantes pour la structure protéique (liaisons disulfure, interactions hydrophobes, les interactions ioniques, etc.). L'objectif était d'implémenter la méthode utilisée sur le site dans le langage de programmation python. Nos résultats sont relativement proches de ceux observés sur PIC. L'utilisation d'une méthodologie différente explique ces écarts.

## 2) Introduction :

Les différents mécanismes d'interactions protéines-protéines sont en grande partie responsable de la complexité cellulaire. En effet, les processus biologiques reposent sur des complexes protéines-protéines stables et spécifiques. Ces interactions sont notamment impliquées dans la transduction du signal, la régulation du cycle cellulaire, la réponse immunitaire, etc. Complémentairement aux interactions intraprotéines, les interactions interprotéines permettent aussi la reconnaissance entre les bons partenaires dans des environnements cellulaires denses et complexes.

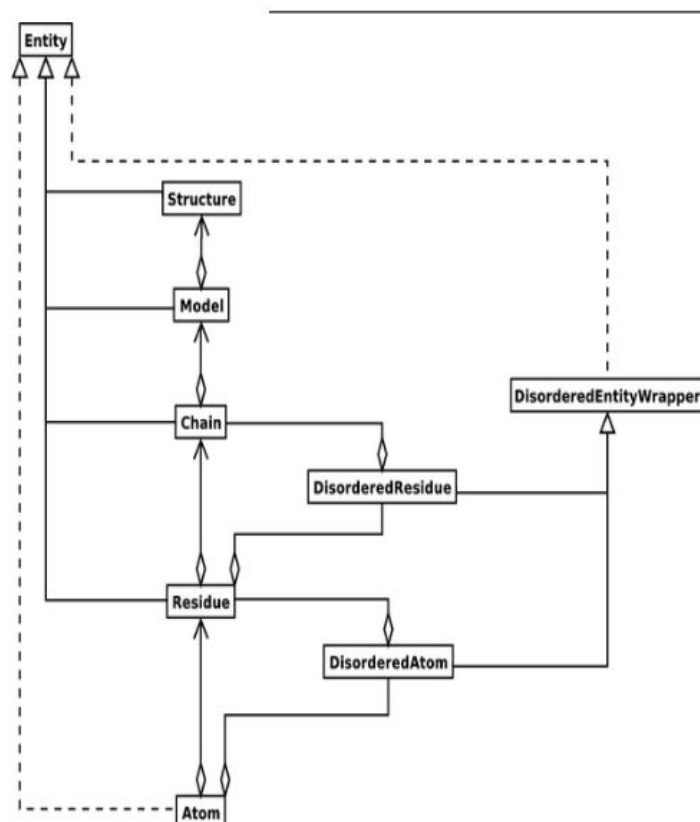
Ces dernières années, des progrès significatifs ont été réalisés dans de nombreux aspects de la biologie computationnelle (l'amarrage, la modélisation des calculs d'affinité, la sensibilité aux changements environnementaux et les mutations ponctuelles)<sup>2</sup>. Malgré ces progrès, de nombreux défis restent à relever et une compréhension complète des forces agissant au sein ou entre les protéines n'a pas encore été atteinte.

Le service Web, PIC (Protein Interactions Calculator), permet de faciliter la reconnaissance et l'analyse de divers types d'interactions dans les structures tertiaires des protéines et les structures des complexes protéine-protéine. Les interactions identifiées entre les résidus peuvent être visualisées à l'aide d'une interface RasMol et Jmol. L'avantage du serveur PIC est l'accessibilité aux calculs d'interaction inter-résidus sur un seul site.

L'objectif du projet était d'implémenter les méthodes utilisées par PIC dans le langage de programmation python afin de déterminer les interactions interprotéines et intraprotéines à partir d'un fichier Protein data Bank (PDB).

### 3) Methods :

Ce projet a été réalisé grâce à un travail collaboratif sur l'outil Github. Pour effectuer l'implémentation, nous avons utilisé le module Biopython<sup>3</sup> basé sur le langage de programmation python. Ainsi, nous avons utilisé le module Bio.PDB qui permet de travailler avec les structures des macromolécules biologiques, et plus particulièrement, la classe PDBParser permettant de produire un objet Structure utilisé pour accéder aux données atomiques dans un fichier PDB, par exemple.



**Figure I : Diagramme de l'architecture Structure/Model/Chain/Residue/Atom (SMCRA) servant à représenter une structure macromoléculaire<sup>3</sup>**

*Les lignes pleines avec des losanges indiquent l'agrégation*

*Les lignes pleines avec des flèches indiquent le référencement*

*Les lignes pleines avec des triangles indiquent l'héritage*

*Les lignes en pointillés avec des triangles indiquent la réalisation de l'interface*

Grâce à ce module un objet "enfant" (par exemple Atome ou Residue) peut être extrait de son "parent" en utilisant un identifiant comme clé. L'objet Structure se trouve en haut de la hiérarchie et peut ainsi contenir un ou plusieurs modèles. L'objet Model stocke une liste d'enfants Chain définie par l'identificateur de chaîne dans le fichier PDB (généralement une lettre).

L'objet Chain stocke une liste d'enfants Residue. Un identifiant de résidu est un tuple avec trois éléments. Pour les acides aminés seule la position peut être renseignée. L'objet Atom stocke les données associées à un atome et ne possède pas d'attribut "enfant". L'identifiant de l'atome est simplement le nom de l'atome.

Afin d'étudier les différents types d'interactions et de liaisons inter et intraprotéines, nous avons séquencé notre travail en plusieurs scripts. Les distances définissant les interactions ont été définies par le site PIC.

Pour les **interactions ioniques**, nous avons réalisé une sélection des acides aminés compatibles (possédant des atomes avec une différence d'électronégativité d'au moins 1.7). Ceci concerne les résidus arginine, histidine, lysine (chaîne latérale chargée positivement) et les acides aspartique et glutamique (chaîne latérale chargée négativement).

Afin de distinguer les résidus positifs des négatifs, nous avons rédigé la fonction nommée ionic\_fun(atom1, atom2, resid1, resid2, dist = 6). Les arguments : resid1 et resid2 étant les deux acides aminés étudiés, atom1 et atom2 étant les atomes potentiellement impliqués dans l'interaction. Pour savoir s'il s'agissait d'une interaction ionique, nous avons fait le choix d'appliquer une distance maximale entre atom1 et atom2 de 6 Å (ångström).

Les **interactions aromatiques-aromatiques** sont caractérisées par deux cycles phényl séparés de 4.7 à 7 Å. Les résidus concernés sont donc la phénylalanine, le tryptophane et la tyrosine. Les fonctions impliquées sont dist\_cal() et dist\_center\_coord() permettant respectivement de calculer la distance euclidienne entre deux points (de coordonnées x, y et z), et de déterminer les coordonnées du centre de deux cycles aromatiques. Ensuite, la fonction aro\_aro() permet de vérifier que la distance entre les centres des deux cycles aromatiques est bien comprise entre les marges définies (4.5 et 7 Å par défaut).

La démarche concernant les **interactions aromatiques-soufre** est proche de celle pour les aromatiques-aromatiques. En effet, une version modifiée de dist\_center\_coord() est utilisée afin de retourner les coordonnées du centre d'un seul phényl. Les résidus d'intérêt sont ici les acides aminés aromatiques (PHE, TRP et TYR) ainsi que les soufrés (CYS et MET). L'idée est donc de déterminer la distance entre l'atome de soufre de la cystéine ou de la méthionine et le centre du phényl. Toute distance inférieure à 5.3 Å est considérée comme interaction aromatiques-soufre.

La **liaison hydrogène** est une force intermoléculaire ou intramoléculaire impliquant un atome d'hydrogène (lié à un donneur) et un atome électronégatif comme l'oxygène, l'azote et le fluor (accepteur). Les distances maximales choisies

sont 3.5 Å pour les liaisons entre hydrogène et oxygène ou azote, et 4 Å entre hydrogène et soufre.

Pour les scripts concernant les liaisons disulfures, les interactions hydrophobes et les interactions cation-pi, nous avons utilisé l'opérateur présent dans le module biopython pour calculer les distances séparant les atomes :

```
ca1 = residue1["CA"]
ca2 = residue2["CA"]
# On Soustrait les deux atomes pour obtenir leur distance
distance = ca1-ca2
```

Les interactions disulfures ont été sélectionnées en filtrant les résidus (acides aminés) de Cystéine et en comparant la distance entre les différents atomes de souffres. Selon les critères du site PIC, les ponts disulfures se forment pour des distances inférieures à 2.2 Angström.

Les interactions hydrophobes sont définies par une distance inférieure à 5 Angström pour les acides aminés : Alanine, Valine, Leucine, Isoleucine, Méthionine, Phénylalanine, Tryptophane, Proline et Tyrosine. Comme la chaîne principale de ces acides aminés ne participe pas aux interactions hydrophobes, nous avons retiré tous les atomes de la chaîne principale pour la détermination de ces interactions.

Enfin, les interactions cation-pi ont été déterminées par une distance inférieure à 6 Angström entre les atomes NH3 de la sidechain des acides aminés Arginine et Lysine avec les 6 atomes de carbones composant le cycle aromatique des acides aminés Phénylalanine, Tryptophane et Tyrosine.

#### 4) Results :

##### Comparaison ponts disulfures PIC/script python (fichier pdb : 1gai.pdb) :

	Position	Residue	Chain	Position	Residue	Chain	Distance Å
<b>PIC</b>	210	CYS	A	213	CYS	A	<b>2,03</b>
	222	CYS	A	449	CYS	A	<b>2,02</b>
	262	CYS	A	270	CYS	A	<b>2,04</b>
<b>Script python</b>	210	CYS	A	213	CYS	A	<b>2,03</b>
	222	CYS	A	449	CYS	A	<b>2,02</b>
	262	CYS	A	270	CYS	A	<b>2,04</b>

**Comparaison interactions hydrophobes PIC/script python (fichier pdb : 2yln.pdb) :**

Nbr interactions hydrophobes trouvées par PIC	191
Nbr interactions hydrophobes trouvées par le script python	330
Interactions trouvées par PIC mais pas par python	55
Interactions trouvées par python mais pas par PIC	194
% de correspondance PIC par rapport à python	71,2%

**Comparaison interactions cation-PI PIC/script python (fichier pdb : 2yln.pdb) :**

	Position	Residu e	Chain	Positio n	Residue	Chain	Distan ce	Angle
PIC	64	TYR	A	212	LYS	A	5.46	20.79
	92	PHE	A	80	ARG	A	5.45	52.83
	97	TRP	A	123	ARG	A	5.88	101.69
	97	TRP	A	123	ARG	A	5.98	102.04
Script python	97	TRP	A	123	ARG	A	/	/
	92	PHE	A	70	ARG	A	/	/
	109	PHE	A	105	ARG	A	/	/

**Comparaison interactions ioniques PIC/script python (fichier pdb : 5AGY.pdb.pdb) :**

	Position	Résidu	Chaîne	Position	Résidu	Chaîne	Distance Å
PIC	3	ASP	A	59	HIS	A	na
	3	ASP	A	59	HIS	B	na
	9	ASP	A	20	ARG	A	na
Script Python	3	ASP	A	59	HIS	A	4.637
	3	ASP	A	59	HIS	B	4.640
	9	ASP	A	20	ARG	A	4.175

**Comparaison interactions aromatique-aromatique PIC/script python (fichier pdb : 5AGY.pdb.pdb) :**

	Position	Résidu	Chaîne	Position	Résidu	Chaîne	Distance Å
<b>PIC</b>	11	TRP	A	208	PHE	A	5.34
	15	PHE	A	107	TYR	A	5.09
	30	TYR	A	32	TYR	A	6.88
<b>Script Python</b>	11	TRP	A	208	PHE	A	5.100
	15	PHE	A	107	TYR	A	6.992
	30	TYR	A	32	TYR	A	5.778

**Comparaison interactions aromatique-soufre PIC/script python (fichier pdb : 5AGY.pdb.pdb) :**

	Position	Résidu	Chaîne	Position	Résidu	Chaîne	Distance Å
<b>PIC</b>	144	TYR	A	189	CYS	A	5.12
	144	TYR	B	189	CYS	B	5.08
	167	TYR	A	114	CYS	A	5.05
<b>Script Python</b>	144	TYR	A	189	CYS	A	5.110
	144	TYR	B	189	CYS	B	5.079
	167	TYR	A	114	CYS	A	5.051

**Comparaison liaisons hydrogène PIC/script python (fichier pdb :  
5AGY.pdb.pdb) :**

	Position	Chaîne	Résidu	Atome	Position	Chaîne	Résidu	Atome	Distance Å
<b>PIC</b>	6	A	V	N	58	A	I	O	2.92
	7	A	L	N	31	A	E	O	2.75
	8	A	L	N	56	A	V	O	2.87
<b>Script Python</b>	6	A	V	N	58	A	I	O	2.920
	7	A	L	N	31	A	E	O	2.750
	8	A	L	N	56	A	V	O	2.870

Les résultats obtenus avec nos scripts python pour les interactions aromatiques-aromatiques, aromatiques-soufre ainsi que pour les ponts disulfures semblent proches de ceux obtenus par PIC. Concernant les liaisons hydrogène ainsi que pour les interactions hydrophobes, nos résultats semblent concorder malgré un nombre supérieur de résultats obtenu par le programme python.

Lorsque l'on observe les interactions obtenues pour les liaisons cation-pi, seul un acide aminé semble concorder en utilisant le fichier 2yln.pdb.

De plus on peut observer que la forme des résultats de certains programmes n'est pas optimale. Notamment pour les programmes python sur les ponts disulfures ou les ponts sont présentés en doublons (Souffre 1 - Souffre 2 et Souffre 2 - Souffre 1). On peut remarquer aussi que les résultats du programme pour les interactions cations-pi, sortent en 6 exemplaires, correspondant aux 6 interactions avec les atomes de carbones définies dans le programme.

## **5) Discussion :**

La complexité de ce projet provenait principalement du manque d'information concernant les critères utilisés par le site PIC. N'ayant pas accès à ces critères, nous avons construit notre raisonnement principalement sur les distances séparant les atomes pour déterminer les différentes interactions.

Les disparités observées pour les interactions cation-pi peuvent être expliquées par les différentes méthodologies entre le script python et PIC. En effet, PIC semble prendre en compte les angles dièdres, que nous avons décidé de ne pas prendre en compte par manque de temps. De plus la distance conditionnant

l'interaction, dans le script python, a été définie par une distance inférieure à 6 Angström pour tous les carbones du cycle aromatique alors que nous supposons que la méthode utilisée par PIC pourrait être la distance par rapport au centre du cycle aromatique de l'acide aminé.

Concernant les autres interactions, nos résultats semblent concorder avec les résultats du site PIC, malgré quelques disparités au niveau des distances. À nouveau, nous pouvons supposer que cette différence est induite par notre choix de ne pas prendre en compte les angles dièdres ou par des méthodes de calcul des distances différentes.

Les objectifs de ce projet sont donc partiellement atteints concernant l'implémentation fidèle du site PIC en utilisant le langage de programmation python ainsi que concernant la forme des résultats de certains programmes. Il faudrait consulter plus précisément les données de littératures et tester un plus grand nombre de protéines pour savoir si les interactions supplémentaires observées sont des erreurs méthodologiques lors de la réalisation du script python ou de réelles interactions non prises en compte par le site PIC.

## 6) Bibliographie :

[1] K. G. Tina, R. Bhadra and N. Srinivasan, *PIC: Protein Interactions Calculator*, *Nucleic Acids Research*, 2007, Vol. 35, Web Server issue W473–W476.

[2] V. Srinivasa Rao, K. Srinivas, G. N. Sujini, and G. N. Sunand Kumar, *Protein-Protein Interaction Detection: Methods and Analysis*, *Int J Proteomics*. 2014.

[3] J. Chang, B. Chapman, I. Friedberg, T. Hamelryck, M. de Hoon, P. Cock, T. Antao, E. Talevich, B. Wilczyński, *Biopython Tutorial and Cookbook*, Last Update – September 4, 2020 (Biopython 1.78)