

# Compte rendu Projet Barstar

---

Team : Les Mugiwaras

Auteur : ARBES Hugo et HENRIQUES Adrien

M1 BIBS

## Introduction

La Barstar est une protéine synthétisée par une bactérie (*Bacillus amyloliquefaciens*) afin d'inhiber l'activité de la ribonucléase de la Barnase en formant un complexe avec celle-ci. En formant ce complexe, la Barstar couvre le site actif de la Barnase empêchant ainsi la Barnase de dégrader l'ARN dans le cytoplasme de la cellule.

Dans cette étude, nous allons nous intéresser à la protéine Barstar et notamment à ses changements conformationnel lors de l'interaction Barstar-Barnase. En effet, la majorité des algorithmes de docking considèrent que les deux protéines testées sont rigides et ne subissent donc pas de changement conformationnels. Le but est donc de créer un algorithme qui va nous permettre d'étudier la stabilité de la structure de la protéine Barstar au cours du temps ainsi que ses changements conformationnels. Pour cela, nous allons donc dans un premier temps s'intéresser aux changements conformationnels globaux, puis nous nous intéresserons aux changements conformationnels locaux.

**PEUT ETRE DIRE CE QU'ON POURRA VOIR DANS LES GLOBAUX ET LOCAUX**

## Manuel d'utilisation

Dans cette partie, nous allons aborder l'utilisation du programme et principalement des fichiers d'entrée ainsi que des pré-requis nécessaires pour pouvoir exécuter le code.

### Pré-requis

Le programme est composé de plusieurs fonctions pour calculer, afficher, récupérer des données. Ce programme est créé en langage Python (Python 2.7), pour que celui-ci puisse être compris par l'ordinateur, il est nécessaire d'avoir installé Python. Pour installer Python il faut ouvrir un terminal et selon votre debian/système d'exploitation, il faut alors taper :

- Unix : `$ sudo apt-get install python`

Si vous êtes sous Windows, il vous faudra alors télécharger la dernière version de python et l'installer (<https://www.python.org/downloads/>).

Une fois installé, certaines de ses fonctions utilisent des fonctions graphiques telles que Matplotlib. Pour pouvoir les utiliser il est donc nécessaire d'avoir installé Matplotlib. Pour installer Matplotlib, cela dépend une fois encore du système d'exploitation. Pour cela, ouvrez un terminal et tapez :

- Unix : `$ sudo apt-get install python-matplotlib`
- Fedora : `$ sudo yum install python-matplotlib`
- Windows : `python -m pip install --U steuptools`  
`python -m pip install matplotlib`

Ainsi pour exécuter le code, il nous suffira de taper la commande :

`$ Python BarStar.py`

qui exécutera le code en appelant les différentes fonctions.

### Fichiers d'entrée

Les fichiers qui vous seront demandés lors de l'exécution du programme doivent absolument suivre une structure spéciale. En effet, ceux-ci doivent être au format PDB suivant :

Colonne	Data Type	Field	Definition
1 - 6	Strings	"ATOM" ou "MODEL"	Type de données de la ligne
7 - 11	Integers	Atome	Identifiant de l'atome
13-16	Strings	Atome	Type d'atome
17 - 20	Strings	Type résidu	Type du résidu
21 - 26	Integers	Résidu	N° du résidu
32 - 38	Integers	Coordonnées	Position X
40 - 46	Integers	Coordonnées	Position Y
48 - 54	Integers	Coordonnées	Position Z
5 - 14	Integers	Model	N° du modèle

Si les fichiers d'entrée ne suivent pas ce format, le processus de parsing ne se fera pas correctement et le programme risque d'avoir des erreurs ou de ne pas donner les bonnes sorties.

De plus il est nécessaire de suivre les indications qui s'affichent dans le terminal et notamment lorsque l'on écrit les fichiers, il est nécessaire de respecter l'ordre d'entrée des fichiers. Tout d'abord on entre le fichier qui contiendra la Barstar ayant la conformation référent, puis dans un deuxième temps, le fichier contenant les différentes conformations de la Barstar dans le temps.

### Les tests

Le programme a été testé sous plusieurs angles. En effet, nous l'avons testé en entrant mal un des 2 fichiers pour tester si la reconnaissance d'erreur est détectée. De plus, nous avons testé les parties plus calculatoires (RMSD, Rayon de giration et calcul de distance) sur des jeux de données restreint pour pouvoir vérifier les résultats et ainsi la robustesse des calculs. Et enfin nous avons testé ce programme avec les données fournies afin de voir si celui-ci fonctionne.

**PEUT ETRE PARLER DES SORTIES**

### Calcule du rayon de giration et des distances

## Calcul de RMSD

Dans cette partie nous allons détailler le calcul de la RMSD afin de voir comment celui-ci est obtenu et à quoi peut-il servir.

La RMSD (Root Mean Square Deviation) correspond à la racine de la moyenne des distances élevées au carré. En effet, nous disposons de la Barstar de référence ainsi que d'un autre fichier avec les

différentes conformations de la Barstar, dans lesquels les séquences ont été alignées. Le principe consiste donc à comparer la Barstar de référence à ses conformations en les superposant et à calculer la distance entre toutes les paires d'atomes.

### Calcul de RMSD globale

La RMSD globale d'une protéine correspond à la distance spatiale globale entre les 2 protéines. Pour la calculer nous avons donc pour chaque pair d'atome calculé la RMSD pour ensuite additionner cette RMSD de chaque pair d'atomes pour avoir la RMSD de chaque résidu. Ensuite ces RMSD de chaque résidu sont additionnés pour avoir la RMSD de chaque conformation.

$$RMSD_{glob} = \sum RMSD_{residu} = \sum \sum RMSD_{pairatome}$$

$$\text{Avec } RMSD_{pairatome} = \sqrt{\frac{1}{N} \sum_{i=1}^N \delta^2 i}$$

Avec N le nombre de pair d'atome a chaque résidu.

La RMSD obtenu ainsi nous permet de voir quelle conformation s'écarte beaucoup de la conformation de référence et aussi pouvoir déterminer si cette protéine est stable car si la RMSD change beaucoup, cela signifierait que la distance entre la Barstar de référence et la conformation change et donc que la protéine est instable.

Pour faciliter l'analyse, nous avons intégré une fonction graphique pour pouvoir observer la RMSD globale en fonction des conformations et donc en fonction du temps (car les conformations possèdent des identifiants et sont rangées dans l'ordre croissant) (voir figure 1).

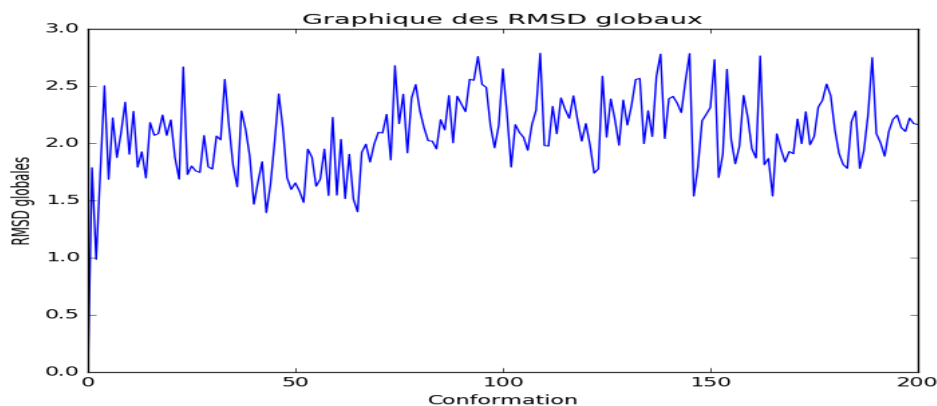


Figure1 : Exemple de graphique RMSD globaux

Nous pouvons ainsi avoir une vision globale des RMSD obtenue et pouvoir interpréter plus facilement. Nous avons aussi fait une fonction pour récupérer les données générées après l'analyse globale, dans un fichier texte pour pouvoir avoir une analyse plus approfondie.

### Calcul de RMSD locales.

La RMSD locale d'une protéine correspond quand a elle aux RMSD moyennes à chaque résidu. Pour la calculer, il nous faut donc calculer la RMSD pour chaque pairs d'atomes puis sommer la RMSD a

chaque pair d'atomes pour avoir la RMSD a chaque résidu. Ainsi nous obtenons la RMSD pour chaque conformation de chaque résidu et ensuite nous avons additionné ces RMSD à chaque conformation pour obtenir la somme de toutes les RMSD de chaque position. Enfin nous avons divisé chaque RMSD par le nombre de conformations pour obtenu une RMSD moyenne a chaque positions.

$$RMSD_{locale} = \frac{RMSD_{residu}}{nb \text{ de conformation}}$$

Avec  $RMSD_{residu}(j, i) = \sum_{j=0}^{Nbres} RMSD_{pairatome}(i)$

Avec Nbres correspondant au nombre de résidu dans les conformations.

#### FORMULE UN PEU BANCALE JE PENSE

La RMSD ainsi obtenu nous permet de voir la variation moyenne de RMSD dans la protéine. Ainsi nous pourrions déterminer quelle partie de la protéine est mieux conservé et quelles parties ne sont au contraire pas conservées. Pour faciliter l'analyse de cette partie, nous avons introduit une fonction graphique nous permettant d'observer la variation d'RMSD moyenne en fonction des positions et pouvoir ainsi voir visuellement cette variation (voir figure 2). Nous avons aussi créé une fonction nous permettant de récupérer les données générées de l'analyse locale dans un fichier texte pour ainsi avoir une analyse plus approfondie.

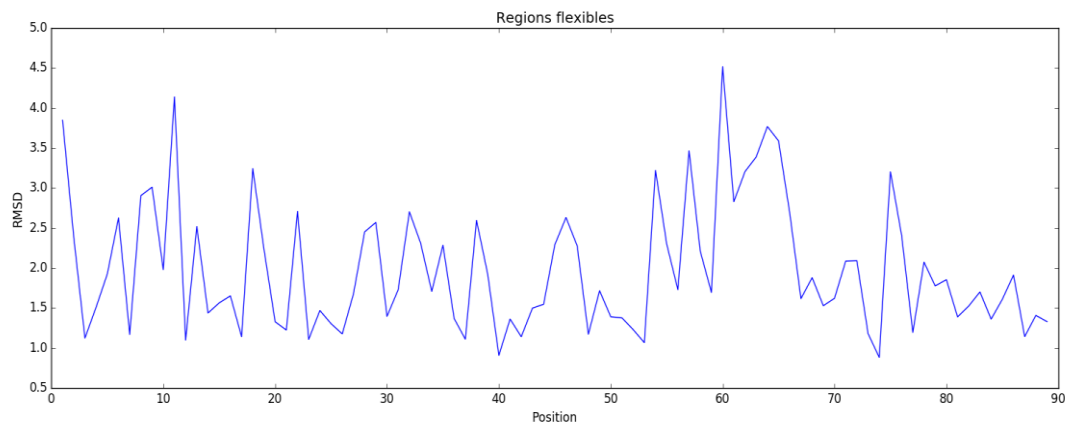


Figure 2 : Exemple de graphique RMSD local

## Résultats biologiques et interprétation

## Conclusion