CONSTRUCTION ET ANALYSE DE SIGNATURES DYNAMIQUES EN SIMULATION DE DYNAMIQUE MOLÉCULAIRE

Cazalis Jean, Debavelaere Vianney

2015

Table des matières

Remerciements

Nous tenons à remercier nos encadrants Alain TROUVÉ et Luba TCHERTANOV, toute l'équipe de BiMoDyn qui nous a suivi tout au long de ce stage, notamment Florent LANGENFELD et Nolan CHATRON, ainsi que Pierre ROUSSILLON du CMLA.

Abstract

Dans ce mémoire nous allons vous présenter de nouveaux outils améliorant la compréhension des dynamiques moléculaires. À l'aide d'un algorithme de clustering spectral et d'une distance quantifiant la désynchronisation entre deux carbones α, nous rechercherons dans un premier temps les domaines fonctionnels des protéines étudiées, STAT5a et STAT5b. Par la suite, dans le but de discriminer les protéines, nous construirons une signature dynamique des simulations de dynamique moléculaire. Cela nous amènera à l'introduction de plusieurs invariances sur les dynamiques des carbones α. Ces résultats mettent en avant l'importance de la dynamique sur le comportement des protéines.

Présentation du problème

Les protéines sont les éléments centraux autant qu'essentiels du vivant, c'est pourquoi étudier leur comportement est primordial. Une protéine est formée d'une chaîne d'acides aminés qui adopte un repliement tridimensionnel spécifique à chaque protéine. La composition en acide aminés (structure primaire) et le repliement adopté par la protéine influence la dynamique de la protéine, qui est elle-même fortement liée aux domaines fonctionnels.

L'équipe de bio-informatique de Luba Tchertanov [1] s'intéresse tout particulièrement à l'analyse de ces éléments dynamiques. À partir de la structure primaire des protéines, l'équipe BiMoDyn simule leurs trajectoires par intégration successive de la seconde loi de Newton. Pour analyser ces mouvements, l'équipe décrit alors les IDS (Independent Dynamical Segment), qui ont été l'objet d'étude des stages précédemment proposés par Alain Trouvé [2]. Cette approche, utilisée au sein du logiciel MONETA, regroupe les résidus d'acides aminés dont la dynamique est très corrélée en éléments (les IDS) dont la dynamique est indépendante. La caractérisation précise des IDS et de la dynamique des protéines porte un intérêt à la fois fondamental (compréhension des mécanismes biologiques à l'échelle atomique) et appliqué (développement de nouvelles stratégies d'inhibition).

Dans le cadre de ce stage interdisciplinaire, nous souhaitons explorer une nouvelle approche dans l'analyse de la dynamique moléculaire de systèmes protéiques, utilisant un algorithme de clustering spectral. Nous nous intéresserons à la classification des protéines et dans ce but nous allons rechercher des signatures dynamiques reposant sur la création de distances. Pour cela, nous utiliserons les trajectoires de STAT5a et STAT5b, phosphorylées ou non, générées par l'équipe BiMoDyn sur des échelles de temps allant de $15ns$ à $200ns$. STAT5a et STAT5b sont des protéines transmettant une information d'origine extracellulaire à l'intérieur de la cellule après activation (phosphorylation) et qui permet la transcription de gènes cibles et ainsi une réponse à l'échelle de la cellule. La dérégulation de l'activation de STAT5 est à l'origine de différents types de cancers (leucémies, cancer de la prostate,...). Pour limiter la taille des données, nous nous sommes restreint à l'étude des carbones α associés aux acides aminés.

1 Clustering

1.1 Introduction et notations

Étant donné la trajectoire d'une protéine, on cherche à regrouper les carbones α par similarité (par exemple la proximité géométrique ou la synchronisation dynamique).

On se fixe un entier l qui correspond au nombre de clusters voulu. Nous allons munir l'ensemble des carbones α d'une distance et chercher une partition $(A_1, \dots, A_l)$ (chaque A_i est un cluster) de $\mathcal{V}$ qui maximise les connexions intra-clusters et minimise les connexions inter-clusters [3].

Pour pouvoir utiliser des outils de clustering spectral [4], on représente nos données sous la forme d'un graphe $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ pondéré et non orienté. On construit le graphe $\mathcal{G}$ à partir de la distance entre les carbones α et de telle sorte que le poids d'une arête entre deux carbones α représente leur similarité. Par la suite, on notera $\mathcal{V} = [\![1, \dots, n]\!]$ ($n = |\mathcal{V}|$) et $w_{i,j} = w_{j,i}$ le poids de l'arête entre les sommets i et j . Il existe de nombreuses façons de construire $\mathcal{G}$, les plus usuelles étant : le graphe des k-plus-proches voisins et le graphe complet.

On introduit maintenant certaines quantités qui permettront de décrire le fonctionnement de l'algorithme.

Pour tout $i \in \mathcal{V}$, on définit le degré du sommet i :

$$d_i := \sum_{j \in \mathcal{V}} w_{i,j} \, .$$

On note $W = (w_{i,j})_{i,j \in \mathcal{V}}$ la matrice des poids et $D = diag(d_1, \dots, d_n)$ la matrice des degrés. On définit alors le laplacien non normalisé L du graphe $\mathcal{G}$ par $L := D - W$.

Le laplacien est une matrice symétrique positive qui satisfait la propriété suivante :

$$\forall x = (x_1, \dots, x_n) \in \mathbb{R}^n, \; x^T L x = \frac{1}{2} \sum_{i,j=1}^{n} w_{i,j} (x_i - x_j)^2 \, .$$

On peut aussi définir le laplacien normalisé L_{sym} du graphe $\mathcal{G}$ par $L_{sym} := I_n - D^{-1/2} W D^{-1/2}$ matrice symétrique positive qui vérifie la propriété suivante :

$$\forall x = (x_1, \dots, x_n) \in \mathbb{R}^n, \; x^T L_{sym} x = \frac{1}{2} \sum_{i,j=1}^{n} w_{i,j} (\frac{x_i}{\sqrt{d_i}} - \frac{x_j}{\sqrt{d_j}})^2 \, .$$

On le verra, ces deux laplaciens jouent un rôle fondamental dans les algorithmes de clustering décrits dans la partie 1.2. Pour $A, B \subset \mathcal{V}$, on définit :

$$W(A, B) := \sum_{i \in A, j \in B} w_{i,j} \, .$$

$W(A, B)$ mesure l'intensité des connexions entre A et B . On remarque aussi que $W(A, A)$ mesure l'intensité des connexions à l'intérieur de A. On définit pour $A \subset \mathcal{V}$:

$$\mathrm{vol}(A) := \sum_{i \in A} d_i = W(A, A) + W(A, \overline{A}) \, .$$

Le volume $\mathrm{vol}(A)$ d'une partie A mesure les connexions de A . On définit aussi pour toute partition $(A_1, \dots, A_n)$ de $\mathcal{V}$:

$$\mathrm{RatioCut}(A_1, \dots, A_l) := \sum_{i=1}^{l} \frac{W(A_i, \overline{A}_i)}{|A_i|}$$

$$\mathrm{NCut}(A_1, \dots, A_l) := \sum_{i=1}^{l} \frac{W(A_i, \overline{A}_i)}{\mathrm{vol}(A_i)} \, .$$

On veut minimiser ces quantités puisque cela revient à minimiser les connexions entre chaque clusters (minimiser $W(A_i, \overline{A_i})$) et à maximiser la taille des clusters (maximiser $|A_i|$ ou vol(A)). Cela amène à deux algorithmes très similaires qu'on définit dans la partie suivante.

1.2 Algorithmes de clustering spectral

1.2.1 RatioCut

On veut se ramener à un problème d'algèbre linéaire dont on connaît la solution exacte. Soit $(A_1, \dots, A_l)$ une partition de $\mathcal{V}$. On définit la matrice $H = (h_{i,j})$ de taille $n \times l$ de la façon suivante :

$$\forall i = 1, \dots, n,\ \forall j = 1, \dots, l,\ h_{i,j} := \begin{cases} \frac{1}{\sqrt{|A_j|}} & \text{si } v_i \in A_j \\ 0 & \text{sinon} \end{cases} . \tag{1}$$

En notant h_j la j^e colonne de H, on a pour tout $j = 1, \dots, l$:

$$\begin{aligned} h_j^T L h_j &= \frac{1}{2} \sum_{p,q \in \mathcal{V}} w_{p,q}(h_{p,j} - h_{q,j})^2 \\ &= \frac{1}{2}\Big[\sum_{p \in A_j, q \notin A_j} w_{p,q}(h_{p,j} - h_{q,j})^2 + \sum_{p \notin A_j, q \in A_j} w_{p,q}(h_{p,j} - h_{q,j})^2 \\ &\quad + \sum_{p \notin A_j, q \notin A_j} w_{p,q}(h_{p,j} - h_{q,j})^2 + \sum_{p \in A_j, q \in A_j} w_{p,q}(h_{p,j} - h_{q,j})^2 \Big] \\ &= \sum_{p \in A_j, q \notin A_j} w_{p,q}(h_{p,j} - h_{q,j})^2 = \sum_{p \in A_j, q \notin A_j} \frac{w_{p,q}}{|A_j|} = \frac{W(A_j, \overline{A}_j)}{|A_j|} . \end{aligned}$$

On en déduit que :

$$\text{RatioCut}(A_1, \dots, A_l) = \sum_{i=1}^{l} h_i^T L h_i = \sum_{i=1}^{l} (H^T L H)_{ii} = tr(H^T L H) .$$

De plus les colonnes de H formant un système orthonormal, on a : $H^T H = I_l$.
Ainsi, minimiser la quantité RatioCut$(A_1, \dots, A_l)$ revient à résoudre :

$$\min_{A_1, \dots, A_l} tr(H^T L H) \text{ tel que } H^T H = I_l \text{ et } H \text{ définit comme en 1} .$$

On relaxe le problème en autorisant les coefficients de H à prendre ses valeurs dans $\mathbb{R}$:

$$\min_{H \in \mathbb{R}^{n \times l}} tr(H^T L H) \text{ tel que } H^T H = I_l .$$

On sait que la solution $\tilde{H}$ de ce problème est donnée par la matrice dont les colonnes sont constituées des l premiers vecteurs propres de L. On trouve alors une solution approchée du problème initial.

Pour définir les clusters, on utilise l'algorithme des k-means sur les lignes de $\tilde{H}$. L'algorithme des k-means (voir partie 2. pour sa description détaillée) crée une partition de $[1, \dots, n] = \mathcal{V}$ en minimisant la somme des distances euclidiennes intra-clusters. Pour H (définit en 1) solution du problème non relaxé, la partition $(A_1, \dots, A_l)$ de $\mathcal{V}$ minimisant la somme des distances intra-clusters est trivialement donnée par :

$$\forall j = 1, \dots, l,\ \forall i = 1, \dots, n,\ i \in A_j \iff h_{i,j} \neq 0 .$$

Ainsi la partition donnée par l'algorithme des k-means correspond aux clusters recherchés dans le cas de la solution exacte. Pour le problème relaxé, l'algorithme donnera alors une solution approchée.

1.2.2 NCut

On va mettre en œuvre une démarche analogue mais qui va faire appel au laplacien normalisé symétrique. Soit $(A_1, \dots, A_l)$ une partition de $\mathcal{V}$.

Comme tout à l'heure, on définit une matrice $H = (h_{i,j})$ de taille $n \times l$ de la façon suivante :

$$\forall i = 1, \dots, n,\ \forall j = 1, \dots, l,\ h_{i,j} := \begin{cases} \frac{1}{\sqrt{\text{vol}(A_j)}} & \text{si } v_i \in A_j \\ 0 & \text{sinon} \end{cases} . \tag{2}$$

Cependant les colonnes de H ne forment pas un système orthonormal, mais on peut vérifier que celles de $G := D^{1/2}H$ en forment un. On peut montrer de la même façon que précédemment que :

$$\text{NCut}(A_1, \dots, A_l) = \sum_{i=1}^{l} h_i^T L h_i = \sum_{i=1}^{l} (H^T L H)_{ii} = tr(H^T L H)\,.$$

Or, en remarquant que $H^T L H = G^T L_{sym} G$, le problème de minimiser $\text{NCut}(A_1, \dots, A_l)$ revient à résoudre :

$$\min_{A_1,\dots,A_l} tr(G^T L_{sym} G) \text{ tel que } G^T G = I_l,\ G = D^{1/2}H \text{ et } H \text{ définit comme en 2}\,.$$

On relaxe le problème en autorisant G à prendre ses valeurs dans $\mathbb{R}$:

$$\min_{G \in \mathbb{R}^{n \times l}} tr(G^T L_{sym} G) \text{ tel que } G^T G = I_l\,.$$

On sait que la solution $\tilde{G}$ de ce problème est donnée par la matrice dont les colonnes sont constituées des k premiers vecteurs propres de L_{sym}.

On définit $\tilde{H} := D^{-1/2}\tilde{G}$. C'est la matrice dont les colonnes sont les l premiers vecteurs propres du laplacien normalisé de la marche aléatoire L_{rw} définit par : $L_{rw} := I_n - D^{-1}W$. On peut facilement calculer $\tilde{H}$ à partir de L sans passer par L_{rw}. En effet, même si L_{rw} n'est en général, pas symétrique, $\tilde{H}$ représente les l premiers vecteurs propres du problème généralisé aux valeurs propres : $Lu = \lambda Du$. On applique ensuite l'algorithme des k-means aux lignes de $\tilde{H}$ et on définit les clusters de la même façon que dans l'algorithme du RatioCut.

Voici le pseudo-code correspondant à cette démarche :

Algorithm 1 Normalized Spectral Clustering

Inputs : matrice des similarités $A \in \mathbb{R}^{n \times n}$, nombre l de clusters
$D \leftarrow diag(d_1, \dots, d_n)$
$L_{rw} \leftarrow I_n - D^{-1}A$ {calcul du laplacien de la marche aléatoire}
$V \leftarrow l$ premier vecteurs propres de L_{rw} en colonne
$P \leftarrow k$-means(V, l)
Outputs : partition P

2 Algorithme des k-means

Soient $(x_1, \dots, x_n)$ des observations dans $\mathbb{R}^d$. Soit k un entier. L'algorithme des k-means permet de partitionner les n observations dans k ensembles $(S_1, \dots, S_k)$ afin de minimiser les distances entre les points à l'intérieur de chaque ensemble. Le but de l'algorithme des k-means est donc de résoudre le problème suivant :

$$\min_{S_1,\dots,S_k} \sum_{i=1}^{k} \sum_{x_j \in S_i} \|x_j - \mu_i\|^2 \text{ avec } \mu_i \text{ la moyenne des points dans } S_i\,.$$

L'algorithme se décompose en deux étapes.

1. On initialise les moyennes $\mu_1, \ldots, \mu_k$. Il existe plusieurs manières de faire, dans l'algorithme standard, on choisit k points au hasard parmi les n observations.
2. Répéter jusqu'à convergence :
   - on assigne à chaque observation la partition la plus proche :

$$\forall i = 1, \ldots, k,\ S_i = \{x_j \mid \forall l = 1, \ldots, k,\ \|x_j - \mu_i\| \leq \|x_j - \mu_l\|\}\,,$$

   - on met à jour les moyennes des clusters :

$$\forall i = 1, \ldots, k,\ \mu_i = \frac{1}{|S_i|} \sum_{x_j \in S_i} x_j\,.$$

Il y a convergence car il n'y a qu'un nombre fini de partitions possibles et à chaque fois qu'on réitère la deuxième étape, on fait diminuer la fonction de coût. On atteint nécessairement un minimum local.

Cependant, puisque ce minimum n'est pas global, l'algorithme peut présenter des instabilités. Pour y pallier, on peut le lancer plusieurs fois avec des initialisations aléatoires ou utiliser des distributions non uniformes bien choisies. Dans ce second cas, on commence par choisir une observation uniformément qui sera notre première moyenne. Pour choisir la deuxième moyenne, on munit chaque observation restante d'un poids correspondant à la distance entre cette observation et la première moyenne et on choisit la deuxième moyenne avec des probabilités proportionnelles à ces poids. On réitère ce procédé avec un poids égal au minimum des distances entre l'observation et les moyennes précédemment calculées. Cette méthode fait en sorte qu'il soit peu probable que deux moyennes soient proches l'une de l'autre.

3 Graphes de similarités

Pour appliquer les résultats présentés ci-dessus, il faut tout d'abord créer un graphe fini, pondéré et non orienté à l'aide des données fournies par les bio-informaticiens. Chaque sommet représente un carbone α et le poids d'une arête (i, j) la similarité entre les sommets i et j . Il faut cependant choisir la manière de les relier. Pour cela, deux choix s'offrent à nous.

On munit l'ensemble des sommets d'une distance d (les distances que l'on utilisera seront définies dans le paragraphe 5).

3.1 Graphe complet

La première possibilité consiste à relier tous les sommets avec un poids dépendant de la distance entre ceux-ci. Plus précisément, on choisit comme fonction de similarité $\exp(-\frac{d(i,j)^2}{2\sigma^2})$ où $\sigma \in \mathbb{R}_*^+$.

Le choix du paramètre σ est important. En effet, si σ est trop petit, la matrice d'adjacence sera proche de l'identité, les similarités entre atomes étant pratiquement toutes égales et les clusters ne seront pas stables. Nous verrons cependant dans le paragraphe 9 que choisir σ grand devant les distances permet d'obtenir une interprétation géométrique du clustering.

3.2 Graphe k-plus-proches

Soit k un entier. Ce graphe relie deux sommets i et j avec un poids 1 si i se trouve parmi les k plus proches voisins (au sens de d) de j ou si j se trouve parmi les k plus proches voisins de i . Un sommet est donc relié à au moins k autres sommets.

Ici, le paramètre est k. On considère que le graphe est bien relié si k est de l'ordre de $\ln(n)$ où n est le nombre de sommets dans le graphe [3].

Cette manière de construire le graphe est "brutale" pour plusieurs raisons. La première est que le poids d'une arête est soit 0 soit 1. On pourrait prendre un poids dépendant de la distance entre les sommets (comme pour le graphe complet) mais cela rajouterait un autre paramètre. La seconde est que deux sommets pourtant proches au sens de d pourraient ne pas être reliés (c'est le cas si par exemple les deux sommets sont dans une zone dense au sens de d).

Par la suite, nous nous sommes rendu compte que le graphe complet nous permettait d'obtenir de biens meilleurs résultats. C'est pourquoi tous les résultats présentés dans la suite de ce rapport utiliserons le graphe complet.

4 Interprétations du NCut et du RatioCut

4.1 Interprétation probabiliste du NCut

On peut remarquer que le coefficient (i, j) de $Q = D^{-1}W = I - L_{rw}$ est égal à :

$$Q(i,j) = \frac{w_{i,j}}{\sum_{l=1}^{n} w_{i,l}}\,.$$

Il s'agit exactement de la probabilité pour une marche aléatoire de passer de l'atome i à l'atome j. $I - L_{rw}$ est donc la matrice de transition d'une marche aléatoire sur le graphe. La chaîne associée à Q est réversible de mesure d'équilibre $\mu(i) = \frac{d_i}{\sum_j d_j}$. Par suite, si $(X_n)_{n\geq 0}$ est le processus canonique et P_μ est l'unique loi sur $[\![1, \cdots, n]\!]$ qui vérifie

$$P_\mu(X_0 = i_0, \cdots, X_p = i_p) = \mu(i_0) \prod_{k=0}^{p-1} Q(i_k, i_{k+1})$$

alors X est une chaîne de Markov de loi initiale μ et de matrice de transition Q.

De plus, utiliser l'algorithme de clustering revient à minimiser le NCut, c'est à dire la somme sur k des

$$P_\mu(X_1 \in \overline{A}_k | X_0 \in A_k) = \frac{\sum_{i\in A_k, j\in\overline{A}_k} w_{i,j}}{\sum_{i\in A_k, j\in\mathcal{V}} w_{i,j}}\,.$$

Chaque terme de cette somme est la probabilité que la chaîne de Markov X sorte du cluster A_k. Minimiser la somme revient donc à minimiser les probabilités de sortir des clusters. En pratique, pour quatre clusters et en utilisant le graphe complet, on trouve des probabilités de sortie allant de 45% à 70%, selon la valeur de σ. Ces valeurs sont inférieures à la probabilité de 75% correspondant à une répartition uniforme.

4.2 Comparaison du NCut et du RatioCut

Pour comparer l'algorithme du NCut et l'algorithme du RatioCut, on compare les quantités qu'ils minimisent à savoir :

$$\text{RatioCut}(A_1, \ldots, A_l) = \sum_{i=1}^{l} \frac{W(A_i, \overline{A}_i)}{|A_i|}\,,$$

$$\text{NCut}(A_1, \ldots, A_l) = \sum_{i=1}^{l} \frac{W(A_i, \overline{A}_i)}{\text{vol}(A_i)}\,.$$

C'est pour cela qu'on étudie la signification du volume d'une partie A. On remarque que : $\forall A \in \mathcal{V},\ \text{vol}(A) = W(A, A) + W(A, \overline{A})$. Ainsi, minimiser le NCut revient à trouver une partition $(A_1, \ldots, A_l)$ qui rend $W(A_i, \overline{A}_i)$ petit et $\text{vol}(A_i)$ grand c'est-à-dire qui rend $W(A_i, \overline{A}_i)$ petit et

$W(A_i, A_i)$ grand. Ce sont exactement les critères que doit remplir un algorithme de clustering à savoir minimiser les connexions inter-clusters et maximiser les connexions intra-clusters.

Quant à lui, l'algorithme du RatioCut essaie de trouver des clusters de taille suffisante puisqu'il prend en compte leur cardinal. Même s'il minimise les connexions inter-clusters, c'est le NCut qui répond le mieux aux critères que nous nous sommes fixés dans notre travail. C'est pourquoi, par la suite, nous utiliseront exclusivement l'algorithme du NCut.

5 Distances

Le but de cette partie est de présenter les différentes distances utilisées par la suite dans ce rapport. On notera :

- N le nombre total d'atomes,
- K le nombre d'instants de simulation,
- $X \in \mathcal{M}_{K,3N}(\mathbb{R})$ la matrice des positions, fournie par l'équipe BiMoDyn, chaque triplet de colonnes de cette matrice correspond aux coordonnées (x, y, z) d'un carbone α toutes les $5ps$,
- $\Sigma \in \mathcal{M}_{3N}(\mathbb{R})$ la matrice de variance-covariance,
- $M \in \mathbb{R}^{3N}$ le vecteur ligne des moyennes.

5.1 Distance euclidienne

La première distance utilisée entre deux atomes est la distance euclidienne entre les moyennes temporelles de leur position. On a donc, pour deux atomes i et j ,

$$d(i, j) := \|M(i, i + 1, i + 2) - M(j, j + 1, j + 2)\|$$

où $M(i, i + 1, i + 2)$ représente les colonnes i, $i + 1$, $i + 2$ de M, c'est à dire la position au cours du temps du i^e carbone α.

Cette distance est uniquement géométrique et ne prend pas en compte la dynamique du système. C'est pour cela qu'on utilisera également une autre distance.

5.2 Distance stochastique

On choisit deux atomes i et j et on note $U \in \mathcal{M}_{K,3}(\mathbb{R})$ les 3 colonnes représentant l'atome i et V celles représentant j . On note p_U et p_V les projecteurs sur les sous-espaces vectoriels engendrés respectivement par les 3 colonnes de U et V . On utilise comme produit scalaire sur les matrices : $\langle M, N\rangle = tr(M^T N)$ pour lequel la norme associée est la norme de Frobenius ou de Hilbert-Schmidt sur les matrices.

On appelle alors distance stochastique :

$$d(i, j) := \|p_U - p_V\| .$$

Cette distance ne dépend en fait que de la matrice de covariance.

On note :

- $\Gamma_{1,1} = U^T U \in \mathcal{M}_{3,3}$ la matrice contenant l'intersection des lignes $i, i + 1, i + 2$ et des colonnes $i, i + 1, i + 2$ de la matrice Σ
- $\Gamma_{1,2} = U^T V \in \mathcal{M}_{3,3}$ la matrice contenant l'intersection des lignes $i, i + 1, i + 2$ et des colonnes $j, j + 1, j + 2$ de la matrice Σ
- $\Gamma_{2,1} = V^T U = \Gamma_{1,2}^T \in \mathcal{M}_{3,3}$ la matrice contenant l'intersection des lignes $j, j + 1, j + 2$ et des colonnes $i, i + 1, i + 2$ de la matrice Σ
- $\Gamma_{2,2} = V^T V \in \mathcal{M}_{3,3}$ la matrice contenant l'intersection des lignes $j, j + 1, j + 2$ et des colonnes $j, j + 1, j + 2$ de la matrice Σ .

Théorème 1. *La distance stochastique entre l'atome i et l'atome j vaut :*

$$d(i,j) = \sqrt{6 - 2tr(\Gamma_{1,1}^{-1}\Gamma_{1,2}\Gamma_{2,2}^{-1}\Gamma_{2,1})}\,.$$

Démonstration. Le projecteur sur le sous-espace engendré par U est de la forme $p_U = UT$ avec $T \in \mathcal{M}_{3,K}(\mathbb{R})$. De plus, pour tout $Y \in \mathbb{R}^K$ et $X \in \mathbb{R}^3$, on a :

$$\langle Y - P_U Y, UX\rangle = 0 \Rightarrow tr(Y^T UX) = tr(Y^T T^T U^T UX)\,.$$

Ce résultat étant vrai pour tout X et tout Y, on a donc :

$$U^T = U^T UT\,.$$

La matrice $U^T U$ étant supposée inversible, ce qui est vérifié expérimentalement, on a finalement :

$$P_U = UT = U(U^T U)^{-1}U^T = U\Gamma_{1,1}^{-1}U^T\,.$$

De même, $P_V = V\Gamma_{2,2}^{-1}V^T$. Ainsi, on a :$\|P_U\|^2 = tr(U\Gamma_{1,1}^{-1}U^T U\Gamma_{1,1}^{-1}U^T) = tr(I) = 3\,.$
Et : $\langle P_U, P_V\rangle = tr(U\Gamma_{1,1}^{-1}U^T V\Gamma_{2,2}^{-1}V^T) = tr(\Gamma_{1,1}^{-1}\Gamma_{1,2}\Gamma_{2,2}^{-1}\Gamma_{2,1})\,.$
On trouve finalement :

$$\|P_U - P_V\|^2 = 6 - 2tr(\Gamma_{1,1}^{-1}\Gamma_{1,2}\Gamma_{2,2}^{-1}\Gamma_{2,1})\,.$$

□

Finalement, disposant de la matrice de covariance, il est assez simple de calculer la distance stochastique.

Il est à noter que c'est cette distance qui prend réellement en compte la dynamique du système, tout en négligeant les distances euclidiennes entre atomes. En effet, la distance stochastique est nulle si et seulement si l'espace engendré par U est égal à l'espace engendré par V. En particulier, si deux atomes appartiennent à un même IDS et sont donc prédits linéairement l'un par l'autre, leur distance stochastique sera nulle.

À un triplet de colonnes, la distance stochastique associe l'espace vectoriel engendré. Elle est donc invariante par transformation linéaire sur les colonnes. Plus précisément, on munit $\mathcal{M} = \mathcal{M}_{K,3N}(\mathbb{R})$ d'une action de groupe (multiplication à droite). Soit $\mathcal{H}$ le groupe des transformations linéaires réversibles sur chaque triplet de colonnes associées à un carbone α :

$$\mathcal{H} := \left\{ \begin{pmatrix} A_1 & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & A_N \end{pmatrix} \in GL_{3N}(\mathbb{R}) \mid \forall i = 1,\dots,N,\ A_i \in GL_3(\mathbb{R}) \right\}$$

Utiliser la distance stochastique revient donc à travailler sur $\mathcal{M}/\mathcal{H}$.

La distance stochastique est également reliée à la notion d'angle entre espaces vectoriels. Elle est nulle si l'angle entre les espaces vectoriels engendrés par U et V est nul, maximale (égale à $\sqrt{6}$) si les deux sous-espaces sont orthogonaux.

5.3 Analyse des distances sur STAT5a

Soit $\mathcal{V} = [\![1,\dots,n]\!]$ l'ensemble des carbones α . Soit d une distance sur $\mathcal{V}$. Soient i et j dans $\mathcal{V}$. Notons $Q_{i,j}$ l'ensemble $\{(k,l) \mid d(k,l) \leq d(i,j)\}$.

On définit le quantile $q_d(i,j)$ associé à i et j par : $q_d(i,j) = \frac{|Q_{i,j}|}{n^2}$. C'est le pourcentage de couples de carbones α dont la distance est plus faible que celle entre les carbones i et j. Cette

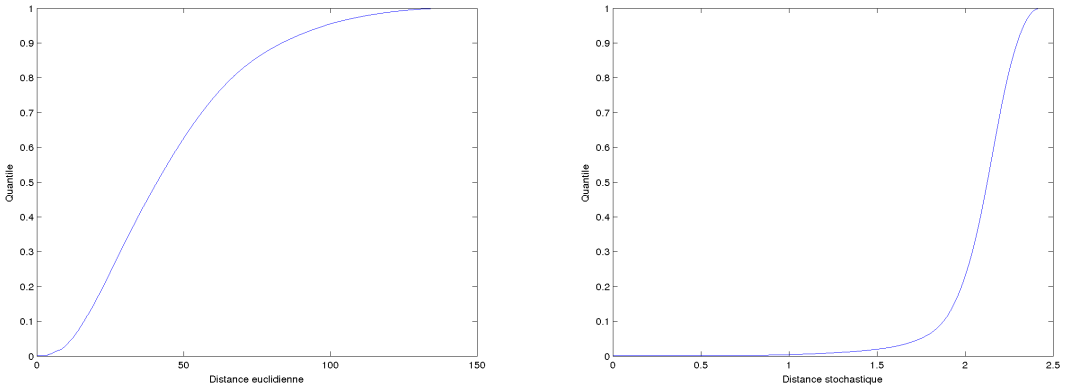
quantité va permettre de donner des représentations plus claires des distances en utilisant de manière uniforme la palette de couleurs disponible.

Notons d_e (resp. d_s) la distance euclidienne (resp. stochastique) définie dans la partie précédente. On va utiliser la distance euclidienne pour valider l'algorithme de clustering puisque c'est une distance "visuelle". La distance qu'on étudiera plus particulièrement sera la distance stochastique puisqu'elle prend en compte la dynamique des carbones α .

Jusqu'à la fin de cette partie les images présentées proviennent de calculs exécutés sur la trajectoire de STAT5a1 de $30ns$.

Dans la figure 1, on a représenté les courbes de transfert associées à d_e et d_s , c'est-à-dire q_{d_e} (resp. q_{d_s}) en fonction de d_e (resp. d_s).

Dans la figure 2, on a représenté les quantiles associés à la distance euclidienne et à la distance stochastique. Chaque points (i, j) de la figure est associé à une couleur d'autant plus proche du rouge que le $q_d(i, j)$ est proche de 1 et d'autant plus bleue que $q_d(i, j)$ est proche de 0.



(a) Courbe de transfert associée à d_e .

(b) Courbe de transfert associée à d_s.

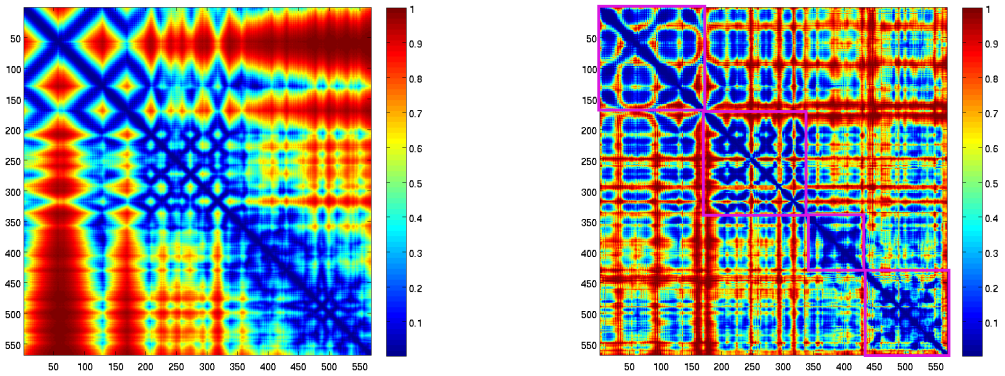
FIGURE 1 – La courbe de transfert associée à d_e se rapproche d'une répartition uniforme, ce qui est logique car STAT5a est une protéine relativement compacte. Cependant la présence de "bouts" dans la protéine va créer la saturation qu'on observe lorsque l'on se rapproche de la distance maximale. Quant à d_s, on remarque que la grande majorité des couples sont très largement décorrélés ce qui donne une courbe de transfert qui varie d'abord peu puis brusquement même si on observe aussi un phénomène de saturation.

6 Représentation et premiers résultats

6.1 Représentation des résultats

On note l le nombre de clusters et N le nombre d'atomes de la protéine. Nous avons deux moyens différents de présenter nos résultats.

Le premier consiste à créer un tableau de l lignes et N colonnes. Un 1 sera présent en (i, j) si l'atome j est dans le i^e cluster. Graphiquement, ce tableau est représenté de la manière suivante : le numéro de l'atome se trouve en abscisse, chaque unité de l'ordonnée correspond à un cluster. Par exemple, dans la figure 3, on représente le résultat du clustering spectral appliqué à la distance euclidienne sur la protéine STAT5a. Le 500^e atome est alors dans le 3^e cluster.
Cette représentation a pour avantage de mettre en avant les numéros des atomes présents dans chaque cluster.



(a) d_e sans ACP.

(b) d_s sans ACP.

FIGURE 2 – Dans le coin supérieur gauche de la figure (a), on observe un quadrillage qui correspond aux hélices α présentes sur la première partie de la structure secondaire de STAT5a, de plus on distingue trois domaines compacts qui semblant en accord avec la représentation 3D de la protéine. Dans la figure (b), on distingue cette fois 4 domaines le long de la diagonale qui correspondent aux domaines fonctionnels de la protéine, cependant, ces domaines sont trop liés les uns les autres ce qui pourrait nuire à l'algorithme de clustering dans leur recherche. On peut expliquer ces liaisons entre domaines par la présence d'un mouvement global de la protéine qui transporte plus d'énergie et donc qui prédomine sur les mouvements locaux des domaines fonctionnels.

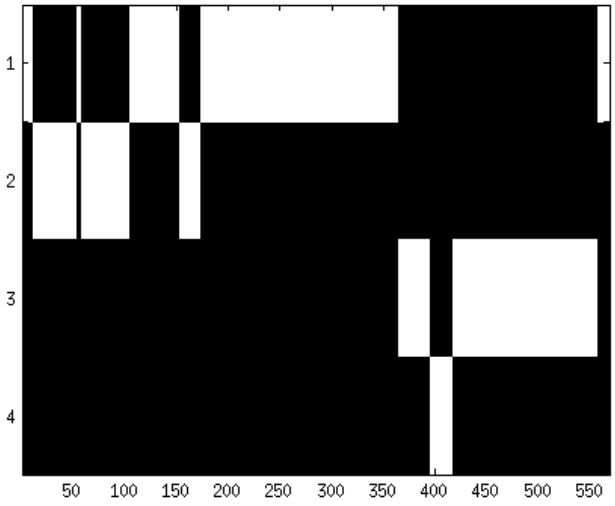


FIGURE 3 – STAT5a1, 4 clusters, distance euclidienne

La deuxième représentation consiste à utiliser le logiciel VMD (Visual Molecular Dynamics). Il représente les protéines dans l'espace, une couleur étant attribuée à un cluster. On peut alors se représenter la position des clusters au sein de la géométrie de la protéine.

6.2 Résultats à l'aide de la distance euclidienne

On applique l'algorithme du clustering à la protéine STAT5a1. On recherche, par exemple, 7 clusters pour le graphe complet, en utilisant le laplacien non normalisé. Le rendu sous VMD est présenté sur la figure 4.

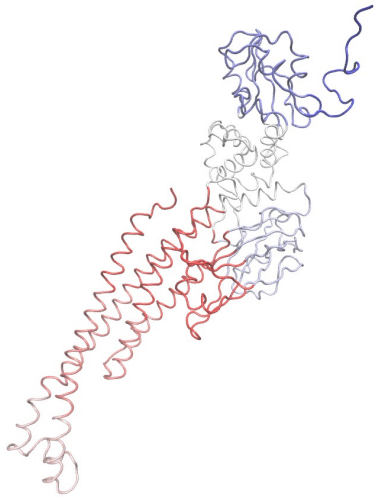


FIGURE 4 – STAT5a1, 7 clusters, distance euclidienne

Dans ce cas simple, le laplacien de la marche aléatoire nous donne un résultat semblable. On peut remarquer que les atomes se regroupent de façon à ce que les distances euclidiennes inter-atomes au sein d'un même cluster restent faibles.

Nous allons maintenant nous intéresser à la recherche de quatre clusters. En effet, nous savons que STAT5a et STAT5b possèdent quatre domaines fonctionnels, associées à quatre régions de la protéine. Nous aimerions les retrouver à l'aide de l'algorithme de clustering. Ces fonctions sont représentées sur la figure 5.

Cependant, la distance euclidienne ne permet pas de les retrouver (cf figure 6). Celle-ci se contente en effet de regrouper les atomes les plus proches en distance géométrique, condition que ne satisfont pas les domaines fonctionnels. Nous allons donc nous demander si la dynamique de la protéine contient ou non cette information, ce que semble suggérer l'inspection visuelle des distances stochastiques figure 2.

6.3 Premiers résultats à l'aide de la distance stochastique

Nos premières applications du clustering via la distance stochastique n'ont cependant pas fournies dans un premier temps de résultats satisfaisant à propos de la recherche des fonctions biologiques. On remarque figure 7 que ces fonctions ne sont pas retrouvées par l'algorithme de clustering de marche aléatoire utilisant le graphe complet. En effet, la protéine STAT5 possède

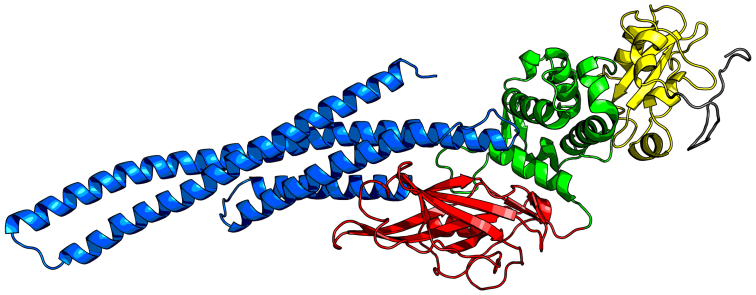


FIGURE 5 – Représentation des quatre domaines fonctionnels de Stat5

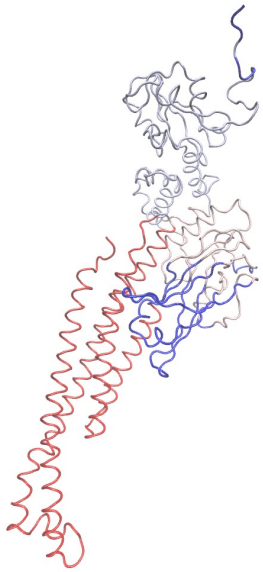


FIGURE 6 – STAT5a1, quatre clusters, distance euclidienne

un mouvement global important qui masque des phénomènes dynamiques plus locaux à l'algorithme de clustering. La très grande majorité des carbones α est regroupée dans un seul cluster. Il faut donc supprimer ce mouvement global et c'est dans ce but que nous allons utilisé l'analyse en composantes principales (ACP). L'idée sous-jacente est que les modifications de conformations globales de grandes amplitudes constituent des modes propres parmi les plus énergétiques car mettant en jeu le déplacement de beaucoup d'atomes. De tels modes devraient pouvoir être estimés par une ACP.

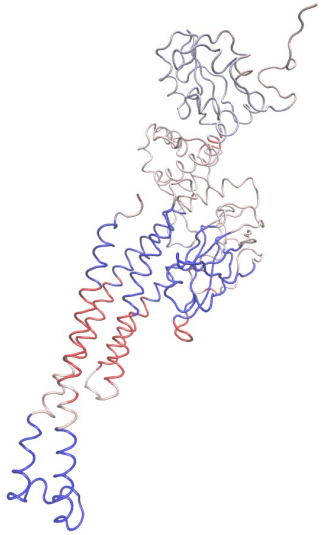


FIGURE 7 – STAT5a1, quatre clusters, distance stochastique, sans ACP

7 L'analyse en composantes principales

7.1 Introduction et notations

Soient n et p deux entiers. On considère un nuage de points X i.e. une matrice de $\mathcal{M}_{n,p}(\mathbb{R})$. Soit $q \leq p$ un paramètre fixé.

L'analyse en composantes principales (ACP) consiste à projeter ce nuage de points sur un espace de dimension plus petite (typiquement de dimension 2 ou 3) en minimisant l'erreur résiduelle. L'objectif de cette méthode est multiple :

— observer le nuage de points dans un espace de dimension q,

— compresser les données en approchant X par une matrice de rang q.

Chaque colonne j de X correspond à la réalisation d'une variable aléatoire X^j. L'idée est de conserver autant que possible les corrélations entre ces données. On note :

— pour tout $i = 1, \ldots, n$ et $j = 1, \ldots, p$, X_i^j la i^e composante de la j^e variable aléatoire,

— $\overline{X} = (\frac{1}{n}\sum_{i=1}^n X_1^i), \ldots, \frac{1}{n}\sum_{i=1}^n X_p^i)$ le vecteur des moyennes,

— $M = (X - \mathbb{1}_n \overline{X})$ la matrice centrée associée à X,

— $C = M^T M$ la matrice de covariance de M.

7.2 Recherche des axes principaux d'inertie

On recherche le vecteur unitaire u tel que la projection de M sur u possède la variance maximale i.e. on veut résoudre :

$$\max_{u\in\mathbb{R}^p,\|u\|=1} \pi_u(M)^T\pi_u(M) = \max_{u\in\mathbb{R}^p,\|u\|=1} u^TCu\,.$$

où π_u désigne le projecteur orthogonal sur la droite engendrée par u.

On remarque que : $u^TCu = \sum_{i=j}^{p}\langle u, X^j\rangle^2 = \mathbf{I}_u$ l'inertie du nuage de points par rapport à l'axe engendré par u. Ainsi l'ACP revient à trouver les axes principaux d'inertie d'un nuage de points.

La matrice C est diagonalisable car symétrique réelle. On note $(\lambda_1,\dots,\lambda_p)$ son spectre avec $\lambda_1 \geq \cdots \geq \lambda_p$. On note aussi pour tout i, v_i un vecteur propre associé à λ_i de telle sorte de $(v_1,\dots,v_p)$ soit une base orthonormée de $\mathbb{R}^p$.

On sait que les q premiers axes principaux d'inertie sont donnés par les q premières valeurs propres. L'ACP consiste à projeter la matrice M sur l'espace engendré par $V = (v_1,\dots,v_q)$ c'est-à-dire à calculer :

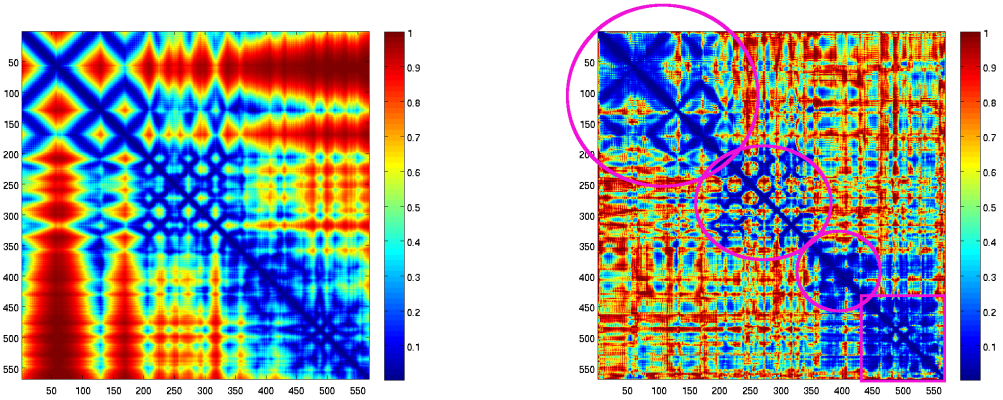
$$(MV)V^T\,.$$

7.3 Utilisation de l'ACP

Dans notre cas, nous voulons supprimer les mouvements globaux de grandes amplitudes. On fait l'hypothèse que l'ACP renvoie la matrice correspondant à ces mouvements. Notre algorithme d'ACP calculera donc $X - (MV)V^T$. Cette opération qui consiste à ôter les espaces engendrés par les plus grandes valeurs propres sera appelée filtrage.

Le problème sera alors de trouver le nombre optimal de valeurs propres à supprimer, de manière à supprimer les changements globaux de conformations sans perdre d'informations sur le mouvement local.

Pour se rendre compte des effets d'un filtrage par ACP, on peut comparer les figures 2 et 8.



(a) d_e avec ACP = 6.

(b) d_s avec ACP = 6.

FIGURE 8 – Les quantiles associés à la distance euclidienne ne changent pas, ce qui est normal puisque l'ACP ne modifie pas les moyennes temporelles. Les quantiles associés à la distance stochastique présentent toujours les quatre domaines fonctionnels mais cette fois bien plus séparés les uns des autres ce qui laisse penser que l'algorithme de clustering sera plus performant dans la recherche de ces fonctions après filtrage par ACP.

8 Résultats après ACP

On relance l'algorithme précédent sur les données transformées par ACP (5 valeurs propres retirées). Les mouvements globaux sont en partie filtrés. Les résultats sont présentés sur la figure 9. On retrouve les quatre domaines fonctionnels ce qui montre que les dynamiques contiennent une information de type fonctionnel qui peut être en partie décryptée par l'algorithme de clustering utilisant la distance stochastique. Ce résultat, très prometteur, montrerait que les domaines fonctionnels peuvent être associés à des domaines dynamiquement cohérents alors qu'ils sont habituellement identifiés par des méthodes très différentes.

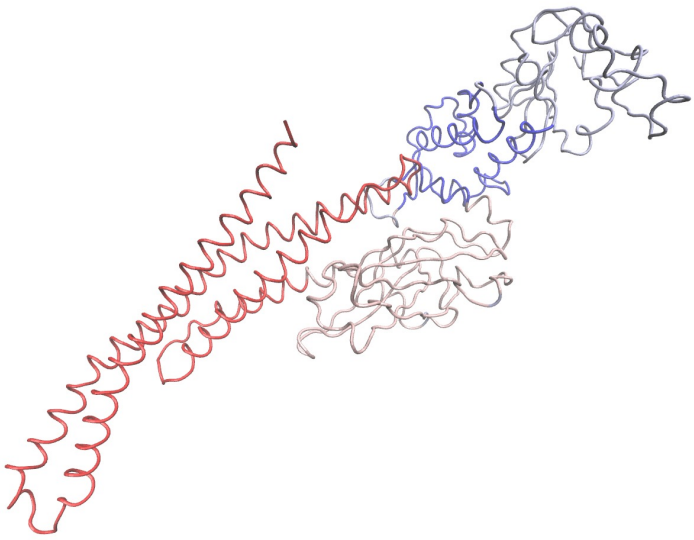


FIGURE 9 – STAT5b1, quatre clusters, distance stochastique, 5 valeurs propres supprimées

L'ACP permet de supprimer le mouvement lent, les domaines fonctionnels sont alors reconnaissables à leur mouvement. Ainsi, un domaine fonctionnel semble correspondre à un mouvement local particulier.

Nous avons utilisé ici le laplacien de la marche aléatoire. On remarque expérimentalement que pour $\sigma \in [0.5, +\infty]$, les clusters obtenus sont sensiblement les mêmes. Contrairement au laplacien non normalisé qui nécessite d'abord d'enlever un trop grand nombre de valeurs propres (souvent plusieurs dizaines) et le choix du σ fait énormément varier les clusters. Retrouver les domaines fonctionnels est alors bien plus difficile. Cela justifie notre choix dans la partie 4.2.

Pour la grande majorité des dynamiques que l'équipe BiMoDyn a mise à notre disposition on retrouve les mêmes clusters. Cependant, pour la réplique STAT5b2, nous n'avons pas réussi à retrouver les fonctions biologiques (cf figure 10). Ce n'est cependant pas la première fois que cette trajectoire pose problème, celle-ci présente en effet une dynamique différente des autres trajectoires.

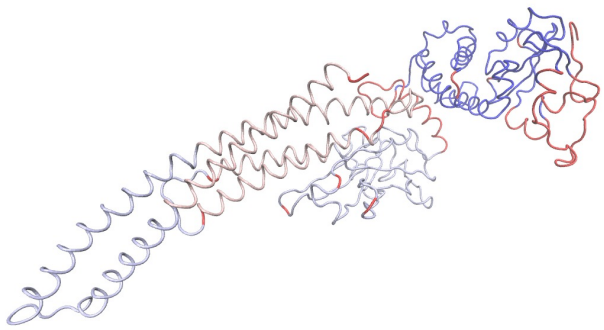


FIGURE 10 – Stat5b2, quatre clusters, distance stochastique, 10 valeurs propres supprimées

9 Étude asymptotique (ou comment le clustering spectral peut cacher une ACP)

Dans l'algorithme de clustering spectral, un facteur d'échelle σ apparaît. Après simulation, on se rend compte que l'algorithme du NCut fonctionne pour σ assez grand. Pour comprendre ce phénomène, on va mener une étude asymptotique et donc réaliser un développement limité de la matrice de transition. Alors que nous nous intéressions à cette étude asymptotique, nous nous sommes rendu compte qu'il était possible de rapprocher les vecteurs propres calculés par l'algorithme de clustering à une ACP sur les projecteurs centrés. C'est ce que nous prouvons ci-dessous. L'ACP a pour intérêt d'avoir une représentation géométrique que le clustering spectral ne possède pas à première vue.

On fixe $\varepsilon = \frac{1}{2\sigma^2}$.

On rappelle les notations : N désigne le nombre d'atomes, T le nombre de conformations, A la matrice des poids et D la matrice des degrés. On veut étudier la matrice de transition $M = D^{-1}A = (m_{ij})_{ij}$.

Théorème 2. *Lorsque ε tend vers 0, les vecteurs propres de la matrice de transition tendent vers celles de la matrice $\tilde{P}$ avec $\tilde{P}_{i,j} = -2\langle p_i - \overline{p}, p_j - \overline{p}\rangle$.*

Démonstration. Lorsque ε tend vers 0, on a :

$$\begin{aligned}
\forall i,j = 1,\dots,N,\ m_{ij} &= \frac{e^{-\varepsilon\|p_i-p_j\|^2}}{\sum_l e^{-\varepsilon\|p_i-p_l\|^2}} \\
&= \frac{1-\varepsilon\|p_i-p_j\|^2+o(\varepsilon)}{N-\varepsilon\sum_l\|p_i-p_l\|^2+o(\varepsilon)} \\
&= (\frac{1-\varepsilon\|p_i-p_j\|^2}{N}+o(\varepsilon))(1+\frac{\varepsilon}{N}\sum_l\|p_i-p_l\|^2+o(\varepsilon)) \\
&= \frac{1}{N}(1-\varepsilon\|p_i-p_j\|^2+\frac{\varepsilon}{N}\sum_l\|p_i-p_l\|^2+o(\varepsilon))\,.
\end{aligned}$$

On pose : $M_\varepsilon = \frac{1}{N}(\mathbb{1}\mathbb{1}^T - \varepsilon P)$ avec $P = (\|p_i - p_j\|^2 - \frac{1}{N}\sum_l \|p_i - p_l\|^2)_{ij}$. Comme $\|p_i\| = \|p_j\|$ pour tous i, j, on a, en posant $\overline{p} = \frac{1}{N}\sum_l p_l$ la moyenne des projecteurs :

$$\forall i, j = 1, \dots, N,\ P_{ij} = -2\langle p_i, p_j\rangle + \frac{2}{N}\sum_l \langle p_i, p_l\rangle = -2\langle p_i, p_j\rangle + 2\langle p_i, \overline{p}\rangle = -2\langle p_i, p_j - \overline{p}\rangle\,.$$

On a $\tilde{P}_{i,j} = -2\langle p_i - \overline{p}, p_j - \overline{p}\rangle$. On va essayer de relier les vecteurs propres de $\tilde{P}$ à ceux de M_ε. Pour cela, on cherche des relations entre P et $\tilde{P}$.

Soit u tel que $\mathbb{1}^T u = 0$. On a alors pour tout vecteur v : $\sum_{i,j}\langle \overline{p}, p_j - \overline{p}\rangle u_i v_j = 0$. On a ensuite :

$$\forall v \in \mathbb{R}^N,\ u^T P v = -2\sum_{i,j}\langle p_i, p_j - \overline{p}\rangle u_i v_j = u^T \tilde{P} v + 2\sum_{i,j}\langle \overline{p}, p_j - \overline{p}\rangle u_i v_j = u^T \tilde{P} v\,.$$

Ainsi : $p_{\mathbb{R}u_0^\perp} \circ (P - \tilde{P}) = 0$.

De plus : $\tilde{P}\mathbb{1} = 0$ et $p_{\mathbb{R}u_0^\perp} \circ \tilde{P} = \tilde{P}$. Ainsi, si on note $u_0 = \frac{1}{\sqrt{N}}\mathbb{1}$, $(u_k)_{k\geq 1}$ une base orthonormale de vecteurs propres de $\tilde{P}$, on a pour tout vecteur v :

$$Pv = p_{\mathbb{R}u_0^\perp}(Pv) + p_{\mathbb{R}u_0}(Pv) = \tilde{P}v + (u_0^T P v)u_0\,.$$

De plus : $(u_0^T P)_j = -\frac{2}{\sqrt{N}}\sum_i \langle p_i, , p_j - \overline{p}\rangle = -2\sqrt{N}\langle \overline{p}, p_j - \overline{p}\rangle$.

Notons σ_k les valeurs propres associées aux u_k et cherchons un vecteur propre de M_ε de la forme $u_k - \alpha u_0$ avec α réel. On a :

$$\begin{aligned} M_\varepsilon(u_k - \alpha u_0) = u_0 u_0^T u_k - \alpha u_0 u_0^T u_0 - \frac{\varepsilon}{N}\tilde{P}u_k + \frac{\alpha\varepsilon}{N}\tilde{P}u_0 \\ - \frac{\varepsilon}{N}(u_0^T P u_k)u_0 + \frac{\alpha\varepsilon}{N}(u_0^T P u_0)u_0\,. \end{aligned}$$

Sachant que : $u_0^T u_k = 0$, $u_0^T u_0 = 1$, $\tilde{P}u_k = \sigma_k u_k$, $\tilde{P}u_0 = 0$ et $u_0^T P u_0 = -\frac{2}{N}\sum_l \langle \overline{p}, p_l - \overline{p}\rangle = 0$, on déduit que :

$$M_\varepsilon(u_k - \alpha u_0) = -\frac{\varepsilon\sigma_k}{N}u_k - (\alpha + \frac{\varepsilon}{N}u_0^T P u_k)u_0\,.$$

On pose : $\lambda_k = -\frac{\varepsilon\sigma_k}{N}$ et on veut : $\alpha + \frac{\varepsilon}{N}u_0^T P u_k = \alpha\lambda_k$. On obtient :

$$\alpha = -\frac{\varepsilon u_0^T P u_k}{N + \varepsilon\sigma_k} = -\frac{\varepsilon u_0^T P u_k}{N} + o(\varepsilon^2)\,.$$

Finalement, le spectre de M_ε est donné par $\mu_0 = 1, (\mu_k = -\frac{\varepsilon\sigma_k}{N})_{k\geq 1}$, μ_0 est associée à $w_0 = \mathbb{1}$ et pour tout $k \geq 1$, un vecteur propre associé à μ_k est :

$$w_k = u_k - \frac{\varepsilon u_0^T P u_k}{N + \varepsilon\sigma_k}u_0 = u_k - \frac{\varepsilon u_0^T P u_k}{N}u_0 + o(\varepsilon^2)\,.$$

□

Ainsi, on peut approcher les vecteurs propres du laplacien de la marche aléatoire grâce aux vecteurs propres de $\tilde{P}$. Cette opération n'a pas juste pour effet d'améliorer les performances de l'algorithme, elle permet aussi de faire un lien entre les projecteurs associés aux carbones α et l'ACP.

Par abus de notations, p_i (resp. $\overline{p}$) désigne l'image du projecteur associé au i^e carbone α (resp. l'image de la moyenne des projecteurs) par l'isomorphisme canonique entre $\mathcal{M}_K(\mathbb{R})$ et $\mathbb{R}^{K^2}$.

Notons aussi : $X = \begin{pmatrix} p_1 - \overline{p} \\ \vdots \\ p_N - \overline{p} \end{pmatrix} \in \mathcal{M}_{N,K^2}(\mathbb{R})\,.$

On remarque que $X^TX = C$ désigne la matrice de covariance des projecteurs, $XX^T = G$ la matrice de Gram des projecteurs, proportionnelle à la matrice $\tilde{P}$ et que X est de rang N (il est réaliste de supposer qu'aucun carbone α n'a sa trajectoire proportionnelle à celle d'un autre carbone α, ce qui est confirmé expérimentalement).

Théorème 3. *L'ACP sur X et l'algorithme de clustering (lorsque σ tend vers ∞) calculent tous les deux les vecteurs propres de G. Si on désigne par $W = (w_1, \dots, w_N)$ les vecteurs calculés par l'ACP et $U = (u_1, \dots, u_N)$ ceux par l'algorithme de clustering, on a la relation :*

$$\forall i = 1, \dots, N,\ w_i = \sqrt{\lambda_i} u_i \,.$$

où λ_i est la valeur propre associé au vecteur propre u_i.

Démonstration. On va d'abord montrer que C est de rang N. Soit $a \in \mathbb{R}^{K^2}$ tel que : $Xa \neq 0$. On remarque que : $\langle X^TXa, a\rangle = \langle Xa, Xa\rangle = \|Xa\|^2 \neq 0$ et que : $\mathrm{Ker}(X) \subset \mathrm{Ker}(C)$. Ainsi le noyau de C et le noyau de X sont confondus. Par la suite, le théorème du rang montre que le rang de C est N.

Soient $v_1, \dots, v_N$ une base orthonormale du supplémentaire orthogonal du noyau de C telle que :

$$\forall i = 1, \dots, N, \exists \lambda_i \mid X^TXv_i = \lambda_i v_i \text{ avec } 0 < \lambda_1 \leq \cdots \leq \lambda_N \,.$$

Soient $u_1, \dots, u_N$ une base orthonormale de $\mathbb{R}^N$, calculée par l'algorithme de clustering, telle que :

$$\forall i = 1, \dots, N, \exists \mu_i \mid XX^Tu_i = \mu_i u_i \text{ avec } 0 < \mu_1 \leq \cdots \leq \mu_N \,.$$

On remarque que ces deux bases de vecteurs propres ainsi que ces deux spectres sont liées. En effet :

$$\forall i = 1, \dots, N,\ X^T(XX^Tu_i) = \mu_i(X^Tu_i) \,,$$

donc, quitte à réordonner les bases et à multiplier par -1 certains vecteurs propres, on a :

$$\forall i = 1, \dots, N,\ v_i = \frac{X^Tu_i}{\|X^Tu_i\|} \text{ et } \mu_i = \lambda_i \,.$$

Or : $\|X^Tu_i\|^2 = u_i^TXX^Tu_i = \lambda_i\|u_i\|^2 = \lambda_i$. Finalement :

$$\forall i = 1, \dots, N,\ v_i = \frac{X^Tu_i}{\sqrt{\lambda_i}} \,.$$

L'ACP projette X sur les axes de variance maximale i.e. calcule $w_i = Xv_i = \begin{pmatrix} \langle p_1 - \overline{p}, v_i\rangle \\ \vdots \\ \langle p_N - \overline{p}, v_i\rangle \end{pmatrix} = \sqrt{\lambda_i}u_i$.

□

Finalement, l'algorithme de clustering (pour σ grand) et l'ACP calculent les mêmes vecteurs à une dilatation des axes de rapport $(\sqrt{\lambda_1}, \dots, \sqrt{\lambda_N})$ près. Le clustering revient alors à appliquer le k-means sur la projection des projeteurs centrés sur les axes de variance maximale.

10 Classification des protéines

10.1 Visualisation en nuage de points

Après avoir exploité les premiers résultats de l'algorithme de clustering, nous nous sommes intéressés à l'étude des vecteurs propres sur lesquels l'algorithme des k-means est appliqué. Cela nous a tout d'abord conduit à l'étude asymptotique de l'algorithme et au rapprochement avec l'ACP. Dans le cas de 4 clusters, on peut projeter la matrice des vecteurs de propres sur $\mathbb{1}^{\perp}$, espace de dimension 3. Cette projection n'entraîne pas la perte d'information et permet de visualiser un nuage de points dans l'espace (figure 11). Cette visualisation est en fait un objet représentant la dynamique de la protéine. Nous nous sommes donc demandé s'il s'agissait d'une signature dynamique des protéines. Si tel était le cas, il serait alors possible, à l'aide de cette représentation, de distinguer deux répliques d'une même protéine voire un état de phosphorylation de l'autre.

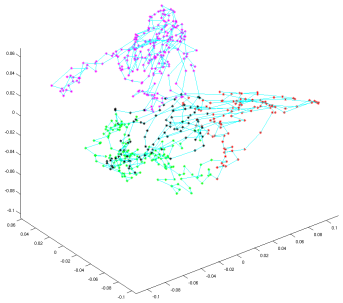


FIGURE 11 – Nuage de points associé à STAT5aP1 sur $30ns$, $ACP = 0$, $\sigma = 1000$.

La première difficulté réside dans l'exploitation de ces résultats. En effet, lorsque l'on affiche deux nuages de points, ceux-ci peuvent sembler très différents alors qu'ils sont pratiquement les mêmes à rotation et réflexion près. Il faut donc rechercher la transformation orthogonale optimale qui va permettre de comparer deux nuages de points. On a alors le résultat suivant [5] :

Théorème 4. *On note $X = (x_1, ..., x_p)$, $Y = (y_1, ..., y_p)$ deux familles de $\mathbb{R}^n$ centrées (ces familles représentent des nuages de n points dans $\mathbb{R}^p$) et $A = \underset{R \in \mathcal{O}_n(\mathbb{R})}{argmin}(\sum_{i=1}^{p} \|Rx_i - y_i\|^2)$. Notons $S = XY^T$. Soit $U\Sigma V^T$ la décomposition en valeur singulière de S, alors A est atteint si et seulement si $R = VU^T$.*

Démonstration. Simplifions le problème :

$$\begin{aligned}\sum_{i=1}^{p} \|Rx_i - y_i\|^2 &= \sum_{i=1}^{p} (Rx_i - y_i)^T (Rx_i - y_i) \\ &= \sum_{i=1}^{p} x_i^T R^T R x_i - y_i^T R x_i - R x_i^T y_i + y_i^T y_i.\end{aligned}$$

Or, $R^T R = I$ car R est orthogonale et $Rx_i^T y_i$ est un scalaire. Donc :

$$\sum_{i=1}^{p} \|Rx_i - y_i\|^2 = \sum_{i=1}^{p} x_i^T x_i - 2 y_i^T R x_i + y_i^T y_i \,.$$

Ainsi :

$$\begin{aligned}\operatorname{argmin}(\sum_{i=1}^{p}\|Rx_i - y_i\|^2) &= \operatorname{argmax}(\sum_{i=1}^{p} y_i^T R x_i)\\ &= \operatorname{argmax}(tr(Y^T RX))\\ &= \operatorname{argmax}(tr(RS))\end{aligned}$$

avec $S = XY^T$. Notons $S = U\Sigma V^T$ la décomposition en valeur singulière de S avec U et V orthogonales et Σ diagonale à coefficients positifs ou nul. Alors, $tr(RS) = tr(\Sigma V^T RU)$. Or, $M = V^T RU$ est une matrice orthogonale, tous ces coefficients sont donc inférieurs à 1. Ainsi, $tr(\Sigma M) = \sum_{i=1}^{p} \sigma_i m_{i,i} \leq \sum_{i=1}^{p} \sigma_i$. Finalement, le maximum de $tr(RS)$ est atteint pour $M = I$ c'est-à-dire pour $R = VU^T$. □

Il nous est alors possible de comparer visuellement deux protéines (figure 12). On peut également créer une distance entre deux nuages de points X et Y appelée interdistance entre X et Y.

Notons $\mathcal{C} = \mathcal{M}_{n,p}(\mathbb{R})$ et $\mathcal{G} = \mathcal{O}_n(\mathbb{R})$. On sait que $\mathcal{G}$ est un groupe qui agit par multiplication à gauche sur $\mathcal{C}$. On note aussi $[X]$ la classe de X dans $\mathcal{C}/\mathcal{G}$ et $d_{\mathcal{C}}$ la distance sur $\mathcal{C}$:

$$\forall X = (x_1, \ldots, x_p), Y = (y_1, \ldots, y_p) \in \mathcal{C},\ d_{\mathcal{C}}(X, Y) = \sqrt{\sum_{i=1}^{p}\|x_i - y_i\|^2}\,.$$

Définition . *Pour tout $[X], [Y] \in \mathcal{C}/\mathcal{G}$, on définit l'interdistance entre $[X]$ et $[Y]$ par :*

$$d_{\mathcal{G}}([X], [Y]) = \inf_{g\in\mathcal{G}} d_{\mathcal{C}}(gX, Y) = \sqrt{\sum_{i=1}^{p}\|Rx_i - y_i\|^2}\,.$$

avec $R = VU^T$ où $U\Sigma V^T$ est la SVD de XY^T.

Théorème 5. *$d_{\mathcal{G}}$ est une distance sur $\mathcal{C}/\mathcal{G}$.*

Démonstration. Ce résultat provient de l'équivariance de d sous l'action de $\mathcal{G}$: $\forall g \in \mathcal{G},\ d_{\mathcal{C}}(gX, gY) = d_{\mathcal{C}}(X, Y)$.

1. $d_{\mathcal{G}}$ est clairement symétrique
2. pour tout $X, Y \in \mathcal{C}$ il existe $R \in \mathcal{G}$ tel que : $d_{\mathcal{G}}([X], [Y]) = d_{\mathcal{C}}(RX, Y)$ donc si $d_{\mathcal{G}}([X], [Y]) = 0$ alors : $Y = RX$ i.e. $[X] = [Y]$.
3. Soient $X, Y, Z \in \mathcal{C}$ et $g, g' \in \mathcal{G}$. Par l'inégalité triangulaire, on a :

   $$d_{\mathcal{C}}(gX, Y) \leq d_{\mathcal{C}}(gX, g'Z) + d_{\mathcal{C}}(g'Z, Y) \Longrightarrow d_{\mathcal{G}}([X], [Y]) \leq d_{\mathcal{C}}(gX, g'Z) + d_{\mathcal{C}}(g'Z, Y)\,.$$

   Puisque l'inégalité précédente est vraie pour tout $g \in \mathcal{G}$, on en déduit que pour tout $g' \in \mathcal{G}$:

   $$\begin{aligned} d_{\mathcal{G}}([X], [Y]) &\leq \inf_{g'\in\mathcal{G}} (d_{\mathcal{C}}(gX, g'Z) + d_{\mathcal{C}}(g'Z, Y))\\ d_{\mathcal{G}}([X], [Y]) &\leq \inf_{g\in\mathcal{G}} d_{\mathcal{C}}(gX, g'Z) + d_{\mathcal{C}}(g'Z, Y)\\ d_{\mathcal{G}}([X], [Y]) &\leq d_{\mathcal{G}}([X], [Z]) + d_{\mathcal{C}}(g'Z, Y)\,,\end{aligned}$$

   et à nouveau puisque cette inégalité est vraie pour tout $g' \in \mathcal{G}$, on a de la même manière :

   $$d_{\mathcal{G}}([X], [Y]) \leq d_{\mathcal{G}}([X], [Z]) + d_{\mathcal{G}}([Z], [Y])\,.$$

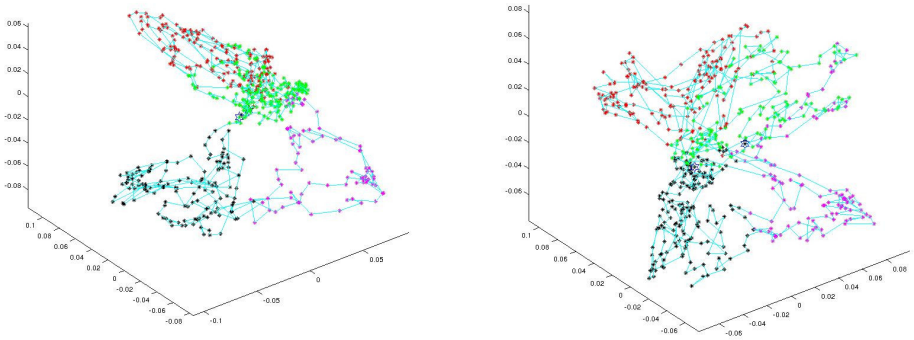


FIGURE 12 – Comparaison entre STAT5a et STAT5aP1

□

Les nuages de points manipulés sont maintenant définis à une transformation orthogonale près et on munit l'espace $\mathcal{C}/\mathcal{G}$ d'une métrique qui va nous permettre de comparer les trajectoires entre elles. L'introduction de cette deuxième invariance (après celle introduite dans la partie 5.2) définit la signature dynamique de la trajectoire, c'est-à-dire sa classe d'équivalence dans $\mathcal{C}/\mathcal{G}$.

Voici le pseudo code amenant au calcul de la distance entre les signatures de deux trajectoires :

Algorithm 2 Comparaison de deux protéines

Inputs : matrices des positions $M_1, M_2 \in \mathbb{R}^{K\times 3N}$, nombre k de composantes du mouvement à conserver, nombre l de vecteurs propres à filtrer par ACP
$X_i, i \in 1,2 \leftarrow ACP(M_i, l)$ {Filtrage de l vecteurs propres par ACP}
$d_i \leftarrow$ Calcul des distances stochastiques
$L^i_{rw} \leftarrow$ Calcul des laplaciens de la marche aléatoire
$W_i \leftarrow k$ premiers vecteurs propres de L^i_{rw} en ligne
$U\Sigma V \leftarrow$ SVD de $W_2 W_1^T$
$R \leftarrow UV^T$ {Calcul de la rotation optimale}
$d_{\mathcal{G}} \leftarrow d_{\mathcal{C}}(RU, V)$ {Calcul de la distance entre U et V}
Outputs : interdistance $d_{\mathcal{G}}$

10.2 Discrimination des trajectoires

Pour comparer les différentes dynamiques sur les mêmes durées, on va segmenter les simulations. Cela va permettre à la fois de comparer les répliques entre elles mais aussi d'étudier l'évolution des dynamiques longues (celles de $200ns$) au cours du temps. Par la suite, chaque simulation aura été découpée en sous-simulations de 15 ns. On calcule alors les distances entre chaque couple de protéines, puis leurs quantiles et on rassemble ces résultats dans une matrice.

Les protéines seront rangées dans l'ordre suivant : 1-2 : STAT5a2, 3-4 : STAT5aP1, 5-6 : STAT5aP2, 7-19 : STAT5a, 20-21 : STAT5b1, 22-23 : STAT5b2, 24-25 : STAT5bP1, 26-27 : STAT5bP2, 28-40 : STAT5b.

Deux questions se posent alors à nous. Faut-il filtrer le mouvement par ACP ? En quelle dimension devons-nous étudier la dynamique ? Cela revient à s'interroger sur les types de mouvement à comparer : les mouvements globaux ou les mouvements rapides ? Faut-il conserver l'ensemble des composantes du mouvement ?

Intéressons-nous tout d'abord à la suppression, ou non, des mouvements rapides. On prend en compte toutes les composantes du mouvement mais on filtre 60 vecteurs propres sur la figure 14 alors qu'on ne modifie pas la matrice du mouvement sur la figure 13.

Sur la figure 13, on ne filtre aucun mouvement et on peut différencier les trajectoires associées aux protéines STAT5a de celle associées aux protéines STAT5b, cependant, les protéines STAT5b1/2/P1/P2 sont assez éloignées de STAT5b. De plus, on ne sait pas différencier les protéines phosphorylées des protéines non phosphorylées.

Au contraire, sur la figure 14, les protéines STAT5a sont bien mieux séparées des protéines STAT5b. De plus, au sein d'un même bloc, les distances restent assez faibles. Enfin, les protéines phosphorylées se séparent d'avantage des non phosphorylées, sans que ce soit encore extrêmement clair.

Pour comparer ces deux images de manière mathématique, on va utiliser un algorithme de clustering hiérarchique. On commence par chercher deux clusters : on espère différencier les trajectoires associées à STAT5a et des trajectoires associées à STAT5b. On coupe ensuite chacun de ces deux clusters en deux sous clusters pour retrouver les protéines phosphorylées de celles non phosphorylées. Les résultats sans ACP sont présentés figure 15 et 16.

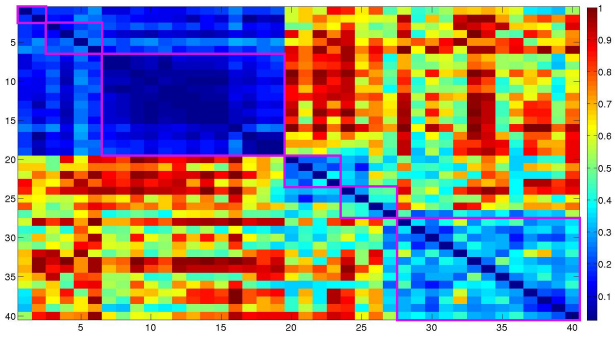


FIGURE 13 – Matrice des distances, toutes composantes, sans filtrage

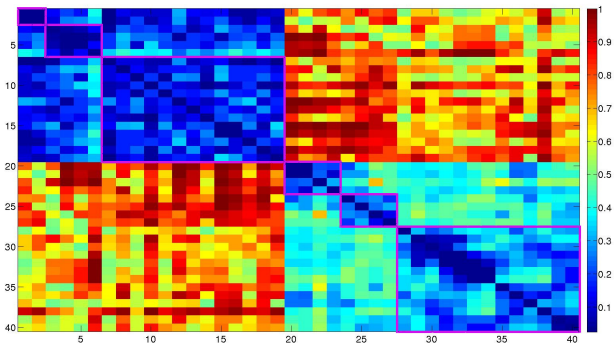
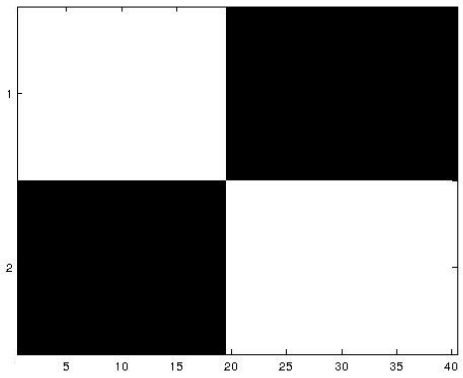
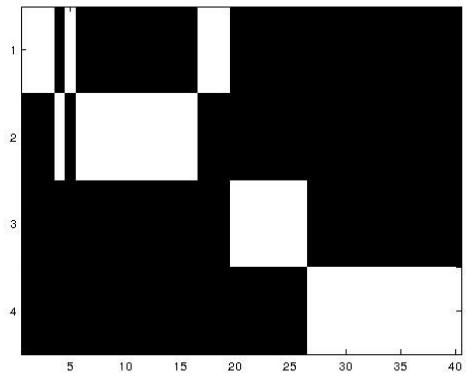


FIGURE 14 – Matrice des distances, toutes composantes, avec filtrage de 60 vecteurs propres

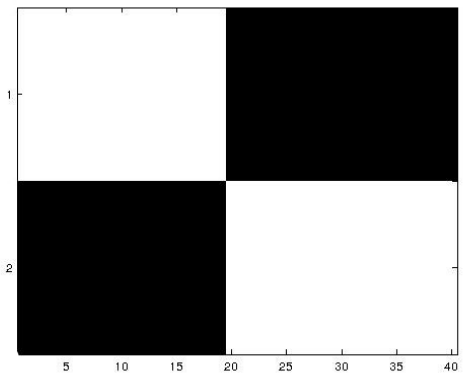


(a) Les protéines STAT5a sont bien séparées des protéines STAT5b

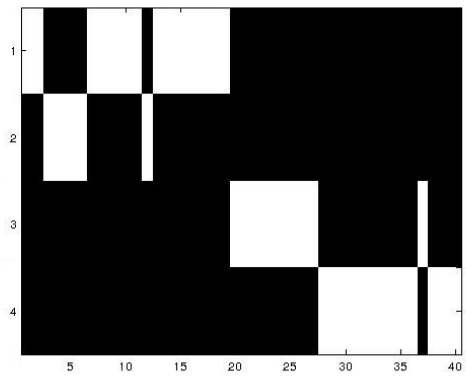


(b) On observe des erreurs au niveau de STAT5a. De plus les STAT5 phosphorylées sont rangées avec des STAT5 non phosphorylées

FIGURE 15 – Clustering hiérarchique avec bruit et sans filtrage



(a) Les protéines STAT5a sont bien séparées des protéines STAT5b



(b) STAT5a2 est bien séparé de STAT5aP1 et STAT5aP2. Il reste cependant des erreurs de séparation au niveau des longues simulations

FIGURE 16 – Clustering hiérarchique avec bruit et filtrage

En fait, on a ici pris toutes les composantes du mouvement. On a donc également pris en compte le bruit correspondant aux dernières composantes de ce mouvement. C'est pour enlever ce bruit qu'on réalise la figure 17 où on enlève 60 vecteurs propres par ACP et où l'on ne conserve que 500 dimensions du mouvement sur les 568, c'est à dire où l'on ne conserve que les 500 vecteurs propres les plus grands du laplacien. Les deux blocs sont maintenant parfaitement séparés. De plus, on peut distinguer les protéines phosphorylées de celles non phosphorylées et même les différentes répliques d'une même protéine. De plus, on remarque que la distance au sein de la simulation de 200ns de STAT5b augmente lorsqu'on compare des simulations éloignées temporellement.

On peut observer les résultats du clustering hiérarchique sur la figure 18. Le seul bémol est que les protéines phosphorylées STAT5bP1 et bP2 ne sont pas séparés de STAT5b1 et b2, contrairement à ce que l'on pourrait penser en regardant la matrice. Cependant, il ne faut pas oublier que l'algorithme de clustering a tendance à empêcher la création de clusters trop petits. La séparation entre formes phosphorylées ou non de STAT5b étant moins clair que pour l'isoforme STAT5a, l'algorithme crée plutôt des clusters de taille équivalente.

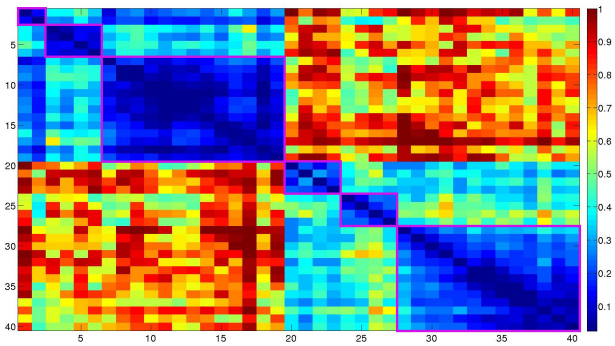
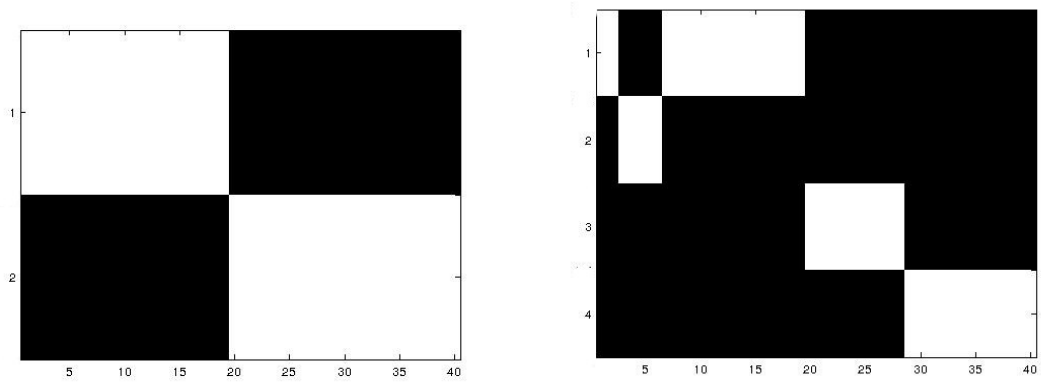


FIGURE 17 – Matrice des distances, sans bruit et avec filtrage



(a) Les protéines STAT5a sont bien séparées des protéines STAT5b

(b) Les STAT5a sont parfaitement séparés. La longue simulation de STAT5b est séparée des autres mais les trajectoires courtes ne sont pas différentiées par leur état de phosphorylation

FIGURE 18 – Clustering hiérarchique sans bruit, avec filtrage

Conclusion et perspectives

Étudier les dynamiques des protéines sous un nouvel angle, celui du clustering, nous a permis d'aboutir à de nombreux résultats. La création d'une distance ne prenant en compte que la dynamique de notre système a mis en avant le lien fort entre dynamique et domaines fonctionnels. Cette même distance a permis de créer une nouvelle méthode d'analyse de la dynamique du système. On peut alors distinguer une protéine de l'autre en analysant uniquement leurs dynamiques respectives. Il reste cependant quelques problèmes à résoudre. Il est notamment difficile de choisir le nombre de vecteurs propres à filtrer lors de l'étude de la dynamique et il n'existe pas encore de critère répondant à cette question. Se pose aussi le problème de la quantité de données. Nous n'avons travaillé qu'avec quelques dizaines de trajectoires, ce qui est insuffisant pour négliger l'influence des trajectoires anormales.

Ce stage ouvre également de nombreuses perspectives : le clustering spectral pourrait être appliqué à la recherche d'IDS. Les quelques tentatives que nous avons faites à ce sujet semblaient d'ailleurs encourageantes. De plus, la comparaison de la dynamique de couple de protéines pourrait aboutir à une compréhension plus fine de l'effet d'une perturbation (mutation, phosphorylation, liaison d'une molécule, etc.) sur la dynamique d'une protéine. En utilisant l'algorithme de discrimination des trajectoires, il serait possible de repérer la partie de la protéine dont la dynamique serait la plus impactée par cette perturbation. Pour trouver des réponses à ces problématiques, l'idée d'intégrer l'ensemble des atomes des résidus (et plus uniquement les carbones α) à l'analyse des dynamiques pourrait être une piste pour les stages à venir.

Références

[1] F. Langenfeld, Y. Guarracino, M. Arock, A. Trouvé, and L. Tchertanov. How intrinsic molecular dynamics controls intramolecular communication in signal transducers and activators of transcription factor stat5.

[2] A. Bengus-Lasnier and O. Bloede. La recherche d'ids dans le cadre de l'étude des dynamiques moléculaires. Technical report, ENS Cachan, 2014.

[3] U. Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 2007.

[4] D. Spielman. Spectral graph theory. *Lecture Notes, Yale University*, 2009.

[5] O. Sorkine. Least-squares rigid motion using svd.