PROJET LONG

MÉTHODE DE GÉNÉRATION DE SCORE DE COMPATIBILITÉ 1D-3D PAR UN ALGORITHME D´ APPRENTISSAGE AUTOMATIQUE

February 9, 2018

Safia SAFA-TAHAR-HENNI
Master II Bio-informatique
Université Paris Diderot

Contents

INTRODUCTION

0.0.1 Etat de l'art et problématique

La compréhension des mécanismes de repliement des protéines est souvent considérée comme un objectif important qui permettra aux biologistes structuraux de découvrir la relation mystérieuse entre la séquence, la structure et la fonction des protéines. Les connaissances sur le repliement des protéines pourraient faciliter davantage la conception de médicaments. Prédire la structure 3D (repliement) d'une protéine est un problème clé en biologie moléculaire. La détermination du repliement d'une protéine repose principalement sur des méthodes expérimentales moléculaires. Avec le développement de techniques de séquençage de nouvelle génération, la découverte de nouvelles séquences protéiques a rapidement augmenté. Avec un tel nombre de protéines, l'utilisation de techniques expérimentales pour déterminer le repliement des protéines est extrêmement difficile parce que ces techniques sont longues et coûteuses. La capacité de prédire les taux de repliement des protéines sans avoir besoin de travaux expérimentaux aiderait le travail de recherche des biologistes structuraux. Ainsi, il est urgent de développer des méthodes de prédiction par ordinateur capables de classer automatiquement, rapidement et avec précision des séquences de protéines inconnues dans des catégories de repliement spécifiques.

Les méthodes de calcul récentes, en particulier des méthodes basées sur l'apprentissage automatique, pour la reconnaissance des repliements protéiques ont été développées (Wei and Zou, 2016). Les méthodes basées sur l'apprentissage automatique peuvent être classées en deux classes selon les algorithmes d'apprentissage utilisés dans les modèles de prédiction : (1) méthodes basées sur un seul classificateur ; et (2) des méthodes basées sur un classificateur d'ensemble. Les méthodes basées sur un seul classificateur utilisent un seul algorithme d'apprentissage spécifique pour construire des modèles de prédiction, tandis que les méthodes basées sur un classificateur d'ensemble utilisent des algorithmes d'apprentissage multiples, similaires ou différents, pour construire des modèles de prédiction. La plupart des méthodes actuelles de classification unique utilisées dans la reconnaissance des repliements protéiques sont basées sur un classificateur SVM. Li, Wu, and Chen, 2013 ont récemment proposé une méthode de reconnaissance des repliements protéiques à base de RF appelée PFP-RFSM. Le cadre de PFP-RFSM implique un algorithme complet de représentation des caractéristiques qui peut capturer des informations séquentielles et structurelles distinctes à partir des séquences primaires et des structures prédites, respectivement. Lampros et al., 2014 proposent une nouvelle méthode d'optimisation pour la classification des repliements

protéiques; le modèle de prédiction de cette méthode est construit sur la base d'une chaîne de Markov formée avec la structure primaire des protéines et sur un HMM à espace réduit.

De nombreux outils bio-informatiques ont vu le jour au cours de la dernière décennie et chacun a présenté des caractéristiques différentes. L'article de Chang et al., 2015, présente une revue et une comparaison d'outils (SFoldRate FOLD-RATE Pred-PFR FoldRate K-Fold) de prédiction qui sont actuellement disponibles sur le Web et prédisent principalement le taux de repliement des protéines.

Ce projet porte sur l'implémentation d'une méthode de génération de score de compatibilité 1D-3D grâce un algorithme d´ apprentissage automatique.

Pour cela, la base de données utilisées est composée d'une structure de chaque famille de la base de donnée Homstrad. À chaque acide aminé (de chaque séquence de chaque structure) est associé un vecteur d'environnement. Ce vecteur est composé d'information sur la structure secondaire, l'accessibilité au solvant, la déformabilité de la protéines (grâce aux alphabets structuraux), et le nombre et types d'interaction créés. L'objectif de ce projet est double ; premièrement déterminer et caractériser des ensembles ("*cluster*") à partir d'une base de données de vecteur d'environnements. Et deuxièmement prédire l'environnement d'une protéine à partir de sa séquence à l'aide d'un arbre de décision (méthodes basées sur un seul classificateur).

MATÉRIEL & MÉTHODES

Le programme est disponible à l'adresse suivante : `https://github.com/SafiaSafa/projet_long`

0.0.2 Données

0.0.2.1 Homstrad

Les données utilisées sont issues de la base de données Homstrad. HOMSTRAD (HOMologous STRucture Alignment Database) est une base de données organisée d'alignements basés sur la structure pour les familles de protéines homologues. Toutes les structures protéiques connues sont regroupées en familles homologues (c'est-à-dire, une ascendance commune) et les séquences des membres représentatifs de chaque famille sont alignées sur la base de leurs structures 3D en utilisant les programmes MNYFIT, STAMP et COMPARER. Ces alignements

basés sur la structure sont annotés avec JOY et examinés individuellement (Mizuguchi et al., 1998). Dans le cadre de ce projet, la base de données est composée d'un représentant(une structure) par famille (qui possède sur Homstrad au moins 2 PDB).
Ce programme est testé sur 10 représentants de cette base.

0.0.2.2 PBxplore

L'utilisation d'un alphabet structural, les Blocs Protéiques (PBs), est un très bon outil pour donner une idée de la déformabilité de la protéine et donc son repliement. Les alphabets structuraux sont des librairies de fragment 3D qui permettent d'approximer une structure protéique. Les PBs sont des petits prototypes structurels qui vont permettre d'approximer localement la structure protéique. L'API PBxplore (Disponible sur `https://github.com/pierrepo/PBxplore`) est composé d'un ensemble d'outils qui permette d'analyser les Dynamique Moléculiare(MD) et la déformabilité structurale des protéines en utilisant les PBs. Il permet de retranscrire chaque snapshots (structures obtenues au cours de la MD) en séquences de Blocs Protéiques, et propose une analyse statistique des variations conformationnelles (Barnoud et al., 2017).

0.0.2.3 DSSP

Afin d'avoir accès aux informations concernant la structure secondaire et l'accessibilité au solvant, j'ai utilisé le programme DSSP, conçu par Joosten et al., 2011 et Kabsch and Sander, 1983. DSSP est une base de données d'affectations de structures secondaires (et beaucoup plus) pour toutes les entrées de protéines dans la PDB.

0.0.2.4 Arpeggio

Arpeggio, un serveur Web qui permet de calculer les interactions entre protéines et protéines, ADN ou de petites molécules ligands, comprenant les interactions de van der Waals, ioniques, carbonyles, métaux, hydrophobes, liaisons halogènes, les liaisons hydrogène et les interactions spécifiques aux atomes des cycles aromatiques et anneau aromatique-anneau aromatique (Jubb et al., 2017). Je me suis plus particulièrement intéressait aux interactions polaires, ioniques, hydrophobes, de van der Waals et aux liaisons hydrogènes.

0.0.3 Méthodes de réduction de dimension

0.0.3.1 Analyse en Composantes Principales

L'analyse en composantes principales (ACP) , (ou principal component analysis (PCA)) permet d'analyser et de visualiser un jeu de données contenant des individus décrits par plusieurs variables quantitatives. C'est une méthode statistique qui permet d'explorer des données dites multivariées (données avec plusieurs variables). L'analyse en composantes principales est utilisée pour extraire et de visualiser les informations importantes contenues dans une table de données multivariées. L'ACP synthétise cette information en seulement quelques nouvelles variables appelées composantes principales. Ces nouvelles variables correspondent à une combinaison linéaire des variables originels. L'information contenue dans un jeu de données correspond à la variance ou l'inertie totale qu'il contient. L'objectif de l'ACP est d'identifier les directions (i.e., axes principaux ou composantes principales) le long desquelles la variation des données est maximale (Kassambara, 2017).

0.0.3.2 Multidimensional scaling

La mise à l'échelle multidimensionnelle (MDS) est un moyen de visualiser le niveau de similarité des cas individuels d'un ensemble de données. Elle fait référence à un ensemble de techniques d'ordination associées utilisées dans la visualisation d'informations, en particulier pour afficher les informations contenues dans une matrice de distance. C'est une forme de réduction de dimensionnalité non-linéaire. Un algorithme MDS vise à placer chaque objet dans un espace à N dimensions de telle sorte que les distances entre les objets soient préservées aussi bien que possible. Chaque objet reçoit ensuite des coordonnées dans chacune des N dimensions. (Wikipedia, 2018). Les résultats de cette analyse sont disponibles en annexe (Figure 16).

0.0.3.3 Sammon mapping

Sammon mapping ou projection de Sammon est un algorithme qui mappe un espace de grande dimension vers un espace de dimensionnalité inférieure (voir Multidimensional scaling) en essayant de préserver la structure des distances inter-points dans l'espace de dimension inférieure. Il est particulièrement adapté pour une utilisation dans l'analyse de données exploratoire. La méthode a été proposée par Sammon, 1969. Elle est considérée comme une approche non-linéaire, car la cartographie ne peut être représentée comme une combinaison linéaire des variables d'origine comme il est possible dans des techniques telles

que l'analyse en composantes principales, ce qui la rend également plus difficile à utiliser pour les applications de classification.

0.0.4 Algorithme

0.0.4.1 Clustering

1. Kmeans

   Le regroupement par K-means est l'algorithme d'apprentissage automatique non supervisé le plus couramment utilisé pour partitionner un ensemble de données donné en un ensemble de k groupes (c'est-à-dire k clusters), où k représente le nombre de groupes pré-spécifiés par l'analyste. Il classe les objets dans plusieurs groupes (clusters), de sorte que les objets d'un même cluster soient aussi similaires que possible (ie, haute similarité intra-classe), alors que les objets des différents clusters sont aussi différents que possibles (ie faible similarité inter-classe). En regroupement par K-means, chaque cluster est représenté par son centre (ie, centroïde) qui correspond à la moyenne des points assignés au cluster. L'idée de base de la classification k-means consiste à définir des clusters afin de minimiser la variation totale intra-cluster. Il existe plusieurs algorithmes k-means disponibles, l'algorithme standard est l'algorithme de Hartigan-Wong (1979).

2. Kmedoids

   Le regroupement par k-Medoids se différencie des k-means par le fait qu'un cluster est représenté avec son centre dans l'algorithme k-means, mais avec l'objet le plus proche au centre du cluster dans l'algorithme k-medoids. Il est plus robuste que k-means en présence de valeurs aberrantes. PAM (Partitioning Around Medoids) est un algorithme classique pour le regroupement par k-medoids.
   Dans ce programme, le package R pam() a été utilisé.

3. Carte de Kohonen

   Les cartes de Kohonen (en anglais, SOM : self organizing maps) sont des réseaux de neurones orientés à deux couches : l'entrée correspond à la description des données, la sortie est organisée sous forme de grille (le plus souvent) et symbolise une organisation des données. Les cartes servent à la fois pour la réduction de dimensionnalité (d'un espace à p dimensions, nous nous projetons dans un espace 2D), pour la visualisation (les proximités sur la grille correspondent à une proximité dans l'espace initial), et la

classification automatique (on peut procéder à des regroupements des neurones de la couche de sortie) (Rakotomalala, 2016).
Dans ce programme, le package R kohonen() a été utilisé.

0.0.4.2 Arbre de décision

Il y a beaucoup de paquets dans R pour modéliser les arbres de décision : rpart, parti, RWeka, ipred, randomForest, gbm, C50 (Malouche, 2016). Dans ce programme, le package R rpart, qui implémente le partitionnement récursif, a été utilisé. Les algorithmes d'apprentissage basés sur un arbre sont considérés comme l'une des méthodes d'apprentissage supervisé les plus utilisées. Contrairement à d'autres algorithmes basés sur des techniques statistiques, l'arbre de décision est un modèle non-paramétrique, n'ayant aucune hypothèse sous-jacente pour le modèle. Les arbres de décision sont de puissants classificateurs non-linéaires, qui utilisent une structure arborescente pour modéliser les relations entre les entités et les résultats potentiels. . Cette approche d'apprentissage machine est utilisée pour classer les données dans des classes et pour représenter les résultats dans un organigramme, tel qu'une structure arborescente.

RESULTATS

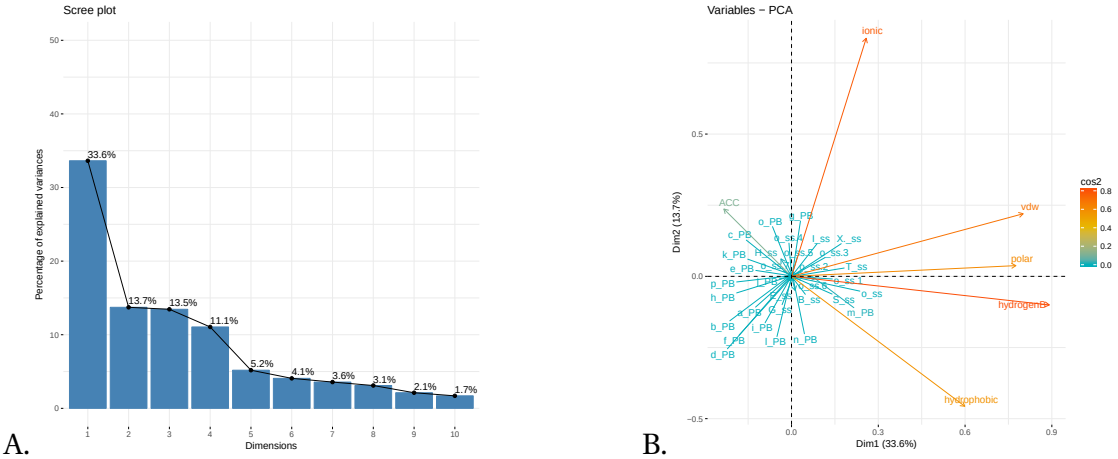
0.0.5 Méthodes de réduction de dimension

Avant d'effectuer la regroupement, une analyse en composante principale a été réalisé(Figure 1). On peut voir que les composante 1 et 2 (~ 47% de la variance totale) ne suffisent pas pour représenter la majorité de la variance (Figure 1.A). Le graphique de corrélation des variables(Figure 1.B) montre les relations entre toutes les variables. Il peut être interprété comme suit:

- Les variables positivement corrélées sont regroupées, par exemple : liaisons hydrogènes, Van der Waals et polaire.
- Les variables négativement corrélées sont positionnées sur les côtés opposés de l'origine du graphique (quadrants opposés), par exemple: accessibilité au solvant et hydrophobicité.
- La distance entre les variables et l'origine mesure la qualité de représentation des variables. Les variables qui sont loin de l'origine sont bien représentées par l'ACP, par exemple: interaction ionique.

De plus un cos2 élevé indique une bonne représentation de la variable sur les axes principaux en considération. C'est le cas ici des variables représentants les liaisons hydrogènes, Van der Waals, hydrophobe, ionic et polaire. Un faible cos2 indique que la variable n'est pas parfaitement représentée par les axes principaux. C'est le cas ici des variables représentants la structure secondaire ou la déformabilité de la protéine (Protein Block).

Figure 1: Analyse en Composantes Principales : A : Graphique de la variance porté par Composantes Principale. B : Graphique de corrélation des variables. La qualité de représentation des variables sur la carte de l'ACP s'appelle cos2 (Cosinus carré).



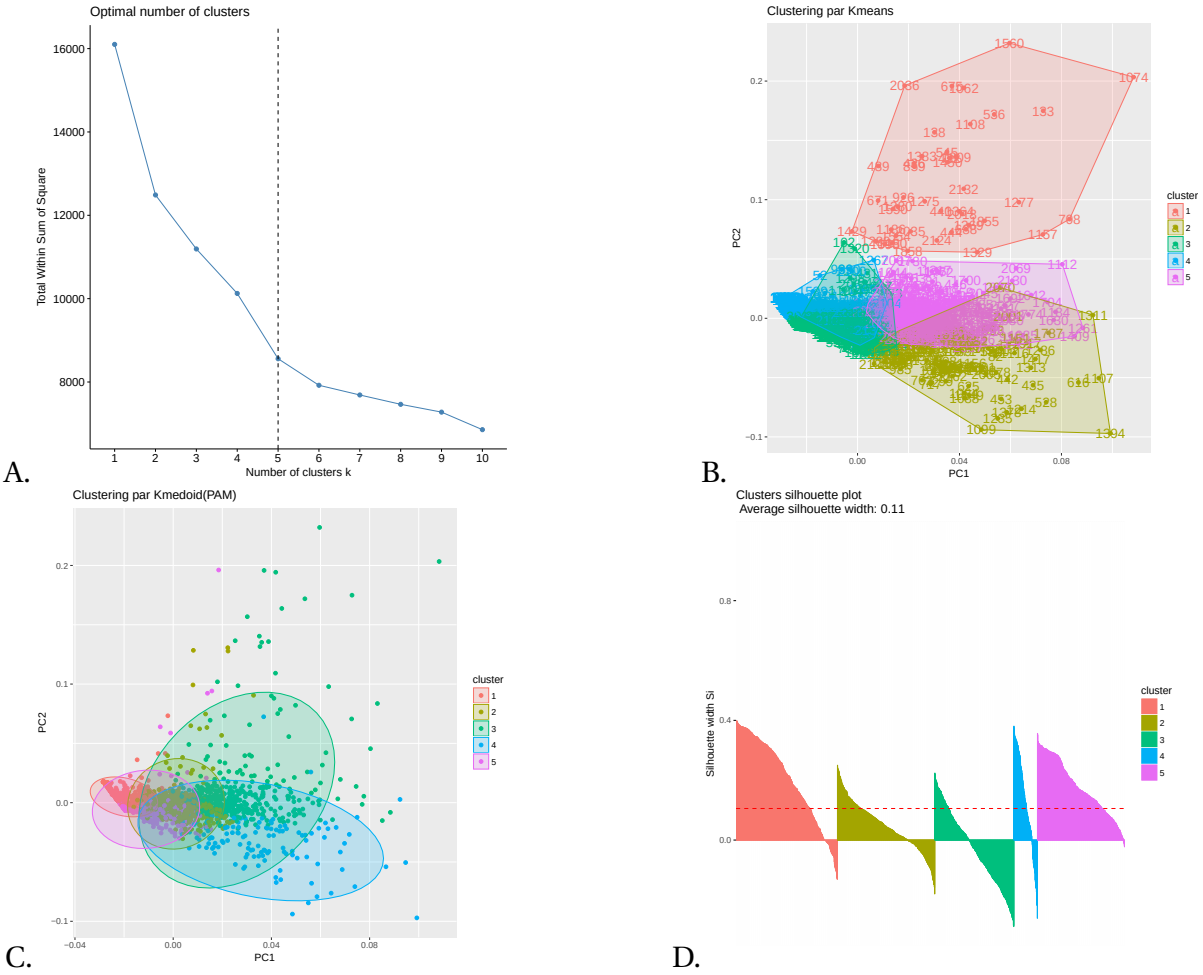
0.0.6 Clustering

0.0.6.1 Kmeans

Pour le regroupement par Kmeans ou K-medoid, il est nécessaire de spécifier le nombre de clusters à utiliser a priori. La figure 2.A présente la méthode du "coude" utilisé pour déterminer le nombre de groupes optimal. L'emplacement d'une courbe (coude) dans le graphique est généralement considéré comme un indicateur du nombre approprié de groupes. On peut voir la présence de coude pour 2 groupes ou 5 groupes. Pour la suite de ce projet, j'ai fait l'hypothèse que le nombre de groupes idéal est de 5. La figure 2.B présente le graphique de regroupement par les Kmeans, on peut voir que le groupe 1 se distingue biens des autres groupes en revanche les quatre autres groupes se confonde sur le graphique. Au vu de la répartitions des points 2 groupes sembles être plus approprié.

La figure 3 présente la distribution des scores dans chaque cluster déterminé par la

Figure 2: Regroupement par Kmeans. A : Détermination du nombre de clusters optimal par la méthode du "coude". B : Résultats de la Regroupement par les Kmeans.
Regroupement par K-medoids. C : Graphique de Regroupement par les K-medoids. D : Graphique des silhouettes des clusters



méthode des Kmeans. Le score est calculé de la manière suivante :

$$\log_2 \frac{freq_{obs}}{freq_{att}}$$

La fréquence observée ($freq_{obs}$) correspond à la fréquence de l'acide aminé dans le cluster. La fréquence attendue ($freq_{att}$) correspond à la fréquence de l'acide aminée dans la base de données. On peut voir que le cluster 1 (Figure 3.A) est enrichie en acide aminé hydrophobe (lysine, valine, triptophane). Lorsque l'on regarde la distribution (en %) des autres caractéristiques qui on servi à définir les vecteurs d'environnement (Figure 4), on remarque que le pourcentage d'accessibilité au solvant (Figure 4.A) dans ce cluster est très

faible en comparaison des autres clusters. Ce qui est logique, un acide aminé hydrophobe à peu de chance de se retrouver proche du solvant. La distribution des acides aminés dans les cluster 2 (Figure 3.B) et 5 (Figure 3.E) est assez uniforme. Dans le cluster 5, on retrouve tous les types d'interaction, majoritairement polaire (dû à la présence de Lysine) et hydrophobe (dû à la présence de Tryptophane). Le cluster 3 est enrichie en acide aminé chargé (glutamate, lysine et arginine), le pourcentage d'interaction hydrophobe dans ce cluster est très faible a contrario des autres type d'interaction. La présence majoritaire d'acide aminée chargé concorde avec le pourcentage d'accessibilité au solvant élevé. Le cluster 4 est enrichie en acide aminé chargé, polaire et hydrophobe, ce qui explique que les différents types d'interaction soient présents dans ce cluster. La distinction entre le cluster 1 est les autres que l'on a constaté sur le graphique de la Figure 2.B semble être du a sont caractère hydrophobe.

0.0.6.2 K-medoids

La figure 2.**C** présente le graphique de regroupement par les K-medoid, on peut voir que les groupes se confondent sur le graphique. Contrairement à la méthode par Kmeans, il ne semble pas y avoir un nombre de groupes plus approprié a posteriori. Pour mieux observé la qualité des groupes obtenus par K-medoid, on peut s'interesser au graphique de la Fig.2.**D** qui montre les silhouettes associés à chaque cluster. Une grand silhouettes *si* (proche de 1) suggère que les observations correspondantes sont très bien groupées, un petit *si* (autour de 0) signifie que l'observation se situe entre deux groupes, et les observations avec un *si* négatif sont probablement placées dans le mauvais groupe. Puisque la moyenne des *si* est respectivement de 0.11 , les groupes ne sont pas correctement identifiés. On remarque que les cluster 1 et 5 sont le seul à ne pas avoir (ou très peu) de *si* négatif, donc les observations dans ces clusters sont très bien groupées.

Pour déterminer ce qui caractérise ces clusters, on peut s'intéresser à la figure 5 qui présente la distribution des scores dans chaque cluster déterminé par la méthode des Kmedoid. On peut voir que le cluster 1 est enrichie en acide aminé polaire(cystéine) et hydrophobe (isoleucine). Lorsque l'on regarde la distribution (en %) des autres caractéristiques qui on servi à définir les vecteurs d'environnement (Figure 6), on remarque que le pourcentage d'interaction hydrophobe est élevé dans ce cluster. Le cluster 5 est enrichie en acide aminé chargé (aspartate, Glutamate et Arginine) ce qui explique le fort pourcentage d'interaction ionique.

0.0.6.3 Carte de Kohonen

- Classique

Dans le cadre de ce projet, j'ai produit 4 carte de Kohonen classique. La taille des grilles hexagonale est de : 2x2 (Figure 7), 3x3 (Figure 10), 4x4 (Figure 11) et 5x5 (Figure 12), avec une structure de voisinage circulaire

La Figure 7.**A** est un Count plot, il représente l'effectif (= nombre d'observation par neurone) dans chaque neurone et permet ainsi d'identifier les zones de fortes densité. En effet plus la couleur du neurone est foncée plus les effectifs sont élevés. Dans l'idéal, la répartition est homogène. S'il y a beaucoup de vide alors la taille de la carte doit être réduite, a contrario on augmentera la taille de la carte s'il y a beaucoup de zone à effectif élevé. Le nombre idéal de neurone semble être au minimum de 25 (Figure 12.**A**).

La Figure 7.**B** représente la distance au voisinage, il est aussi appelé "Graphique U-matrice". Il correspond à la somme au voisin immédiat pour chaque neurone. Les neurones d'une même classe seront proches (rouge foncé). Les neurones les plus éloignés (claire) vont délimiter une zone frontière. Une bonne séparation des classes lors de la typologie va correspondre à la présence de neurones foncés concentré aux extrémités de la carte. On constate que même avec 25 neurones ((Figure 12.**B**), on n'obtient pas une bonne séparation des classes.

Le Codes Book vector (Figure 7.**C**) qui correspond aux vecteurs des poids (profil) de chaque neurone dans un diagramme circulaire permet d'établir le rôle des variables dans la définition des différentes zones qui compose la carte topologique. Ceci permet de distinguer en un coup d'œil les différentes zones de la carte au regard des variables "actives". Sur la Figure 12.**C** on peut voir que les neurones situés à nord-est semble être caractérisé par les structures secondaires. L'ouest semble être caractérisé par les Protein Block. Le sud semble être caractérisé par les différentes types d'interaction.

La carte Heatmap (Figure 7.**D**), va permettre de préciser les variables associé à chaque zone. En effet, cette représentation va permettre de mettre en contraste les zones selon l'intensité de l'activité de la variable. Une variable fortement activée est représentée en rouge. La Figure 12.**D** confirme ce que l'on a vu avec la Codes Map. De plus, on peut voir les que les zones pour lesquels accessibilité au solvant (Sud Est) et hydrophobicité (Nord Ouest) sont active sont diamétralement opposé.

- Toroidal

  Une variante architecturale de la carte de Kohonen possède une structure toroïdal. C'est a dire que le bord gauche sera voisin du bord droit et le bord haut celui du bord bas. La taille des grilles hexagonale est, comme précédemment de : 2x2 (Figure 8), 3x3 (Figure 13), 4x4 (Figure 14) et 5x5 (Figure 15), avec une structure de voisinage circulaire

  Le nombre idéal de neurone pour cette carte toroïdal semble se situer entre 16 neurones (Count plot 14.**A**) et 25 neurones (Count plot 15.**A**).

  Les caractéristiques d'une carte de Kohonen toroïdal vont permettre une meilleure séparation des classe lors de la typologie en effet les neurones foncés vont ce concentré aux extrémités de la carte. On constate qu'avec 25 neurones (Figure 15.**B**), les neurones voisins semble de concentré sur les extrémités de la carte, on commence à avoir une meilleure séparation des classes, mais ce n'est pas encore idéal.

  La Codes Map (Figure 15.**C**) et la Heatmap (Figure 15.**D**) sont assez difficiles à caractérisé en effet la majorité des caractéristique associé au vecteur d'environnement se répartissent uniformément sur l'ensemble des neurones.

0.0.7 Prédiction

Pour réaliser le second objectif de ce projet, j'ai utilisé le package *rpart* de R afin de réaliser une prédiction à l'aide d'un arbre de décision. Pour cela, je suis partie de la structure PDB 1a4fb (une globine) . La Figure 9.A représente l'arbre de décision obtenue. La partie terminale de l'arbre (*feuille*) contient la fréquence prédite de chaque acide aminé données dans l'ordre alphabétique. L'acide aminé le plus fréquent est écrit en haut de chaque case. Les variables déterminantes sont : l'accessibilité au solvant, la structuration en hélice α, en feuillet β, et en coude ("bend").

Généralement, un arbre de décision est élager (on parle de *Pruning*) avant de pouvoir être interprété. Pour cela, j'ai "joué" sur le nombre d'observations minimum par nœud grâce à l'option *minsplit* de *rpart*. Les Figures 9.B (3 observations minimum par nœuds) et C (10 observations minimum par nœuds) sont similaires dues à la faible taille de la base de données de vecteurs d'environnements. Les acides aminés prédits sont les suivant : l'Alanine (32%), la Glycine (23%), la Valine (21%) et le Glutamate (21%). Les critères déterminants sont : l'accessibilité au solvant, la structuration en brin β et en hélice α. Ce que l'on observe sur ces arbres semble concorder avec une réalité structurale. En effet d'après l'arbre de décision de la Figure 9.B, la prédiction de l'alanine est conditionnée par une faible accessibilité au solvant, une faible probabilité de présence dans une structure en feulait β et un forte probabilité de

présence dans une structure en hélice α. Or, l'alanine est connue pour favoriser la formation d'hélices α. Et il est hydrophobe se qui explique une faible probabilité d'accessibilité au solvant. La prédiction de la glycine est conditionnée par une faible accessibilité au solvant, une faible probabilité de présence dans une structure en feuillet β et une faible probabilité de présence dans une structure en hélice α. En effet, on sait que la glycine a un rôle déstabilisant dans la création des hélices α (Wikipedia, 2017b)
La prédiction de la valine est conditionnée par une faible accessibilité au solvant, une forte probabilité de présence dans une structure en feulliet β. En effet les acides aminés dotés de chaînes latérales ramifiées (Thréonine, **valine** et isoleucine) sont principalement trouvés au cœur des feuillets β.(Wikipedia, 2017a) De plus, la valine est un acide aminé hydrophobe.
Le dernier acide aminé prédit est le Glutamate, qui est chargé négativement. Sa prédiction est conditionnée par une forte accessibilité au solvant, en effet cet acide aminé à une probabilité élevé de se retrouver à la surface d'une protéine. (`http://biochimiedesproteines.espaceweb.usherbrooke.ca/1a.html`). La visualisation de la protéine sur Pymol (Figure 9.**D** montre une protéine tout hélice (bleu), et confirme la présence de Glutamate (sphère rouge) à la surface de la protéine.

CONCLUSION

Pour conclure durant ce projet, j'ai pu déterminer et caractériser des environnements à partir d'une base de données issue d'informations sur la structure secondaire, l'accessibilité au solvant, la déformabilité de la protéines (grâce aux alphabets structuraux), et le nombre et types d'interaction créés. C'est une approche novatrice dans le sens où l'utilisation des Protein Block n'est pas intégrée dans ce genre d'analyse. L'approche non supervisée est un plus pour les méthodes de prédiction elle peut permettre de mettre en évidence, sans apriori, des caractéristiques qui vont aider à la prédiction. L'utilisation d'arbre de prédiction est très populaire de par son efficacité et sa simplicité d'utilisation. En effet, cette méthode a permis de rapidement et de manière efficace, prédire l'environnement d'une protéine à partir de sa séquence. Cependant, des améliorations à ce projet peuvent être apportées. Comme l'intégration de plus de structure dans la base de données. L'élagage de l'arbre de décision par d'autres méthodes comme la validation croisée. J'aurais souhaité développer plus avant la partie Sammon Mapping et Multidimensional Scalling. Sur le long terme, une prédiction par réseau de neurones pourrait être mise en place.

REFERENCES

References

Barnoud, Jonathan et al. (2017). "PBxplore: a tool to analyze local protein structure and deformability with Protein Blocks". en. In: *PeerJ* 5, e4013. ISSN: 2167-8359. DOI: 10.7717/peerj.4013. URL: https://peerj.com/articles/4013 (visited on 02/08/2018).

Chang, Catherine Ching Han et al. (2015). "Towards more accurate prediction of protein folding rates: a review of the existing web-based bioinformatics approaches". en. In: *Briefings in Bioinformatics* 16.2, pp. 314–324. ISSN: 1467-5463. DOI: 10.1093/bib/bbu007. URL: https://academic.oup.com/bib/article/16/2/314/246305 (visited on 02/08/2018).

Joosten, Robbie P. et al. (2011). "A series of PDB related databases for everyday needs". en. In: *Nucleic Acids Research* 39.suppl_1, pp. D411–D419. ISSN: 0305-1048. DOI: 10.1093/nar/gkq1105. URL: https://academic.oup.com/nar/article/39/suppl_1/D411/2506762 (visited on 02/08/2018).

Jubb, Harry C et al. (2017). "Arpeggio: A Web Server for Calculating and Visualising Interatomic Interactions in Protein Structures". In: *Journal of Molecular Biology* 429.3, pp. 365–371. ISSN: 0022-2836. DOI: 10.1016/j.jmb.2016.12.004. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5282402/ (visited on 02/08/2018).

Kabsch, Wolfgang and Christian Sander (1983). "Dictionary of protein secondary structure: Pattern recognition of hydrogen-bonded and geometrical features". en. In: *Biopolymers* 22.12, pp. 2577–2637. ISSN: 1097-0282. DOI: 10.1002/bip.360221211. URL: http://onlinelibrary.wiley.com/doi/10.1002/bip.360221211/abstract (visited on 02/08/2018).

Kassambara (2017). *ACP - Analyse en Composantes Principales avec R: L'Essentiel - Articles - STHDA*. fr. URL: http://www.sthda.com/french/articles/38-methodes-des-composantes-principales-dans-r-guide-pratique/73-acp-analyse-en-composantes-principales-avec-r-l-essentiel/ (visited on 02/08/2018).

Lampros, Christos et al. (2014). "Assessment of optimized Markov models in protein fold classification". In: *Journal of Bioinformatics and Computational Biology* 12.04, p. 1450016. ISSN: 0219-7200. DOI: 10.1142/S0219720014500164. URL: http://www.worldscientific.com/doi/abs/10.1142/S0219720014500164 (visited on 02/08/2018).

Li, Junfei, Jigang Wu, and Ke Chen (2013). "PFP-RFSM: Protein fold prediction by using random forests and sequence motifs". en. In: *J. Biomedical Science and Engineering*,

pp. 1161–1170. DOI: http://dx.doi.org/10.4236/jbise.2013.612145. URL: https://file.scirp.org/pdf/JBiSE_2013122017024148.pdf.

Malouche, Dhafer (2016). *Arbres de décisions sous R*. URL: https://rstudio-pubs-static.s3.amazonaws.com/133943_8dd7e91ae27e40a5ba4493f14e302ce6.html (visited on 02/08/2018).

Mizuguchi, Kenji et al. (1998). "HOMSTRAD: A database of protein structure alignments for homologous families". en. In: *Protein Science* 7.11, pp. 2469–2471. ISSN: 1469-896X. DOI: 10.1002/pro.5560071126. URL: http://onlinelibrary.wiley.com/doi/10.1002/pro.5560071126/abstract (visited on 02/08/2018).

Rakotomalala, Ricco (2016). *SVM avec R et Python*. fr. URL: https://eric.univ-lyon2.fr/~ricco/tanagra/fichiers/fr_Tanagra_Kohonen_SOM_R.pdf (visited on 08/08/2016).

Sammon, J. W. (1969). "A Nonlinear Mapping for Data Structure Analysis". In: *IEEE Transactions on Computers* C-18.5, pp. 401–409. ISSN: 0018-9340. DOI: 10.1109/T-C.1969.222678.

Wei, Leyi and Quan Zou (2016). "Recent Progress in Machine Learning-Based Methods for Protein Fold Recognition". en. In: *International Journal of Molecular Sciences* 17.12, p. 2118. DOI: 10.3390/ijms17122118. URL: http://www.mdpi.com/1422-0067/17/12/2118 (visited on 02/08/2018).

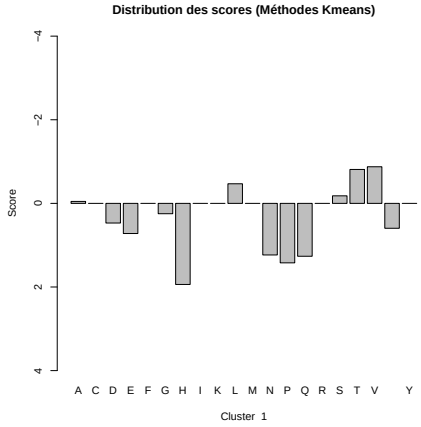
Wikipedia (2017a). *Beta sheet*. en. Page Version ID: 809146850. URL: https://en.wikipedia.org/w/index.php?title=Beta_sheet&oldid=809146850 (visited on 02/08/2018).

— (2017b). *Hélice alpha*. fr. Page Version ID: 141763841. URL: https://fr.wikipedia.org/w/index.php?title=H%C3%A9lice_alpha&oldid=141763841 (visited on 02/08/2018).

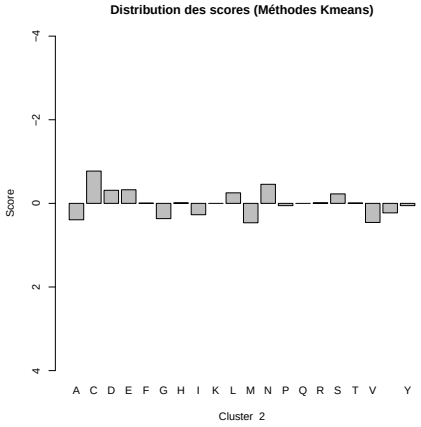
— (2018). *Multidimensional scaling*. en. Page Version ID: 824564931. URL: https://en.wikipedia.org/w/index.php?title=Multidimensional_scaling&oldid=824564931 (visited on 02/08/2018).

ANNEXE

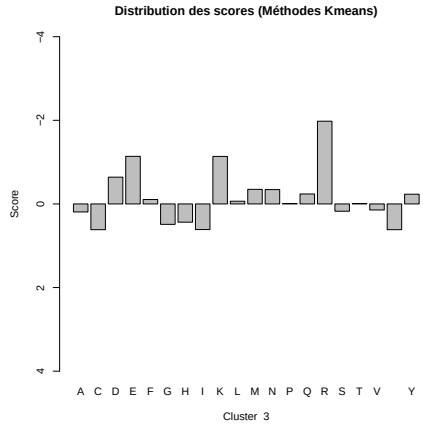
Figure 3: Caractérisation des clusters obtenus par K-means : scores de chaque acide aminé dans chaque cluster (1-**A**, 2-**B**, 3-**C**, 4-**D**, 5-**E**)



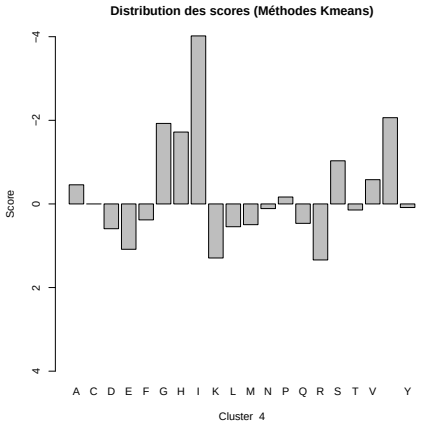
A.



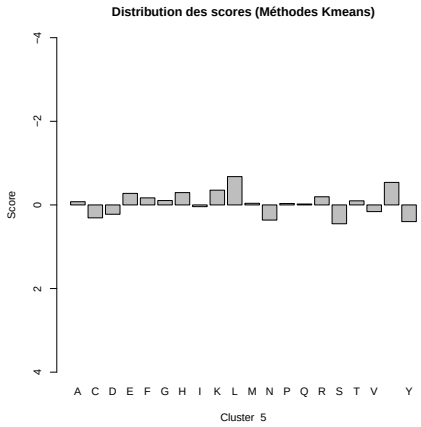
B.



C.

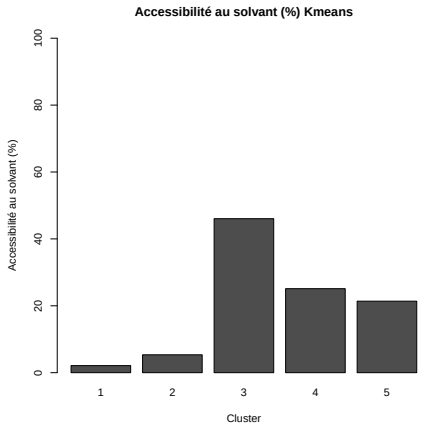


D.

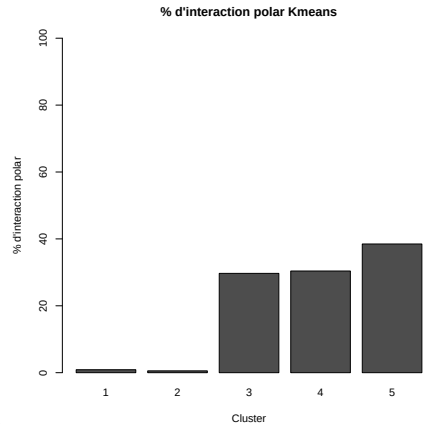


E.

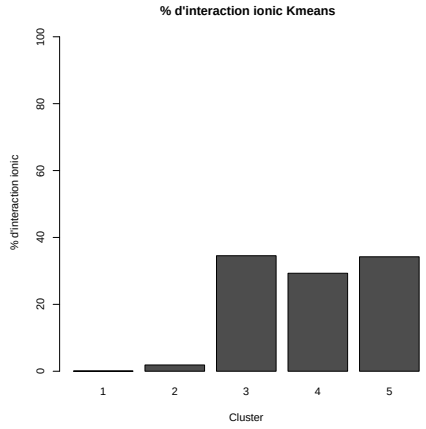
Figure 4: Caractérisation des clusters obtenus par K-means : Distribution (%) des autres caractéristiques (Accessibilité-**A**, Interaction Polaire-**B**, Interaction ionique-**C**, Interaction hydrophobe-**D**, Interaction de Van der Walls-**E**, Liaison hydrogènes-**F**) dans chaque cluster



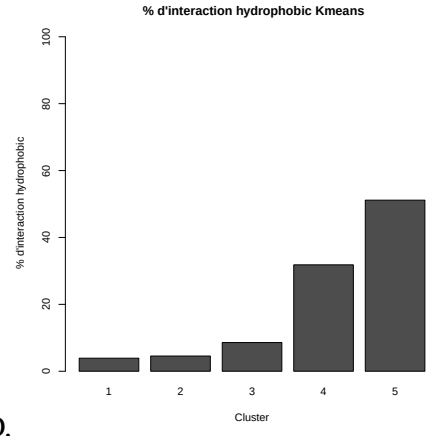
A.



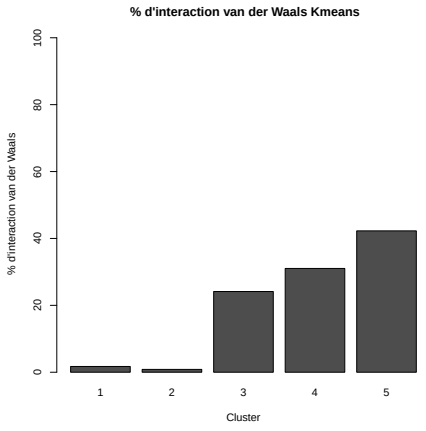
B.



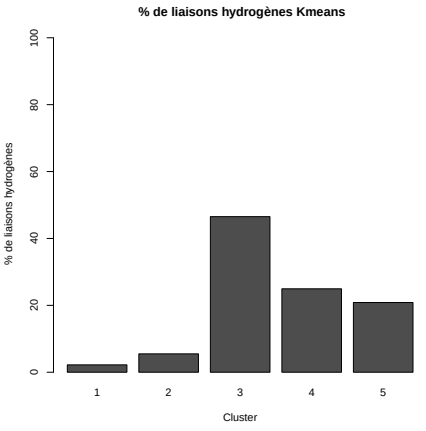
C.



D.

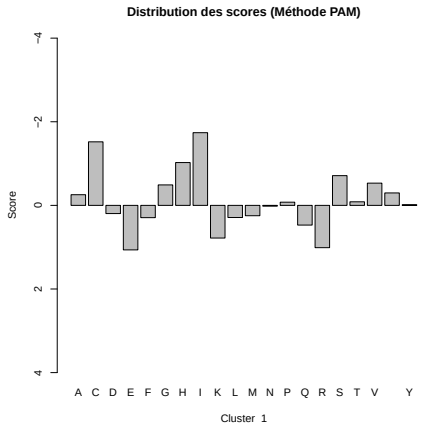


E.

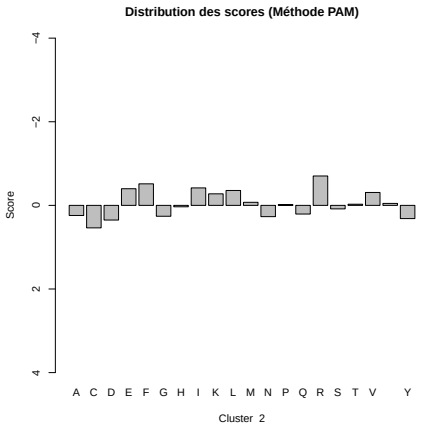


F.

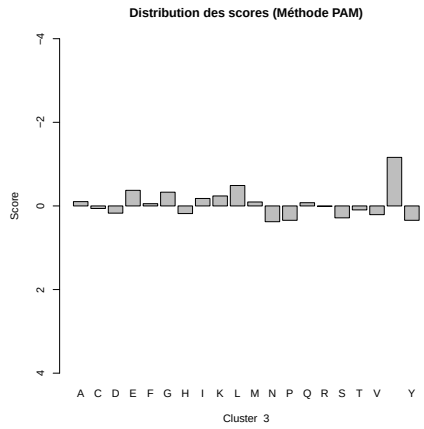
Figure 5: Caractérisation des clusters obtenus par K-medoids : scores de chaque acide aminé dans chaque cluster (1-**A**, 2-**B**, 3-**C**, 4-**D**, 5-**E**)



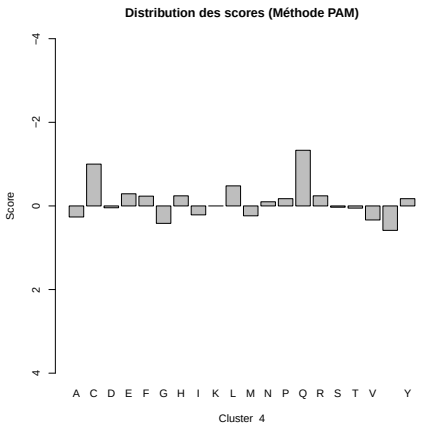
A.



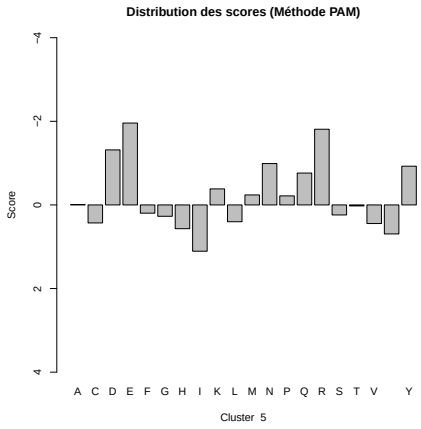
B.



C.

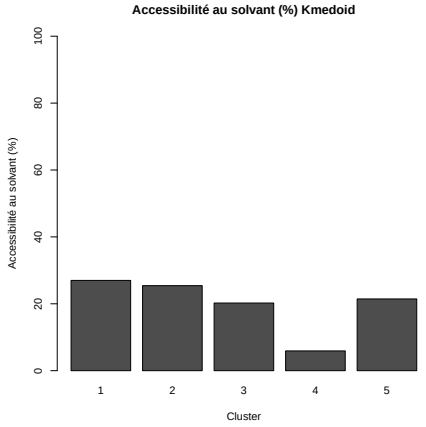


D.

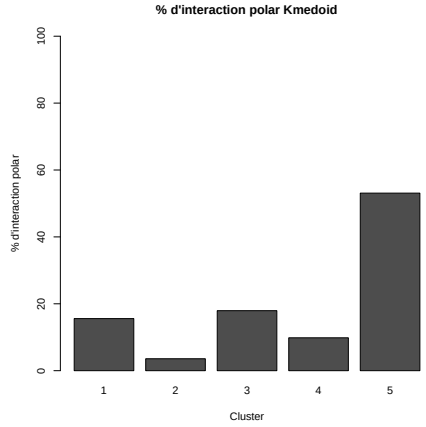


E.

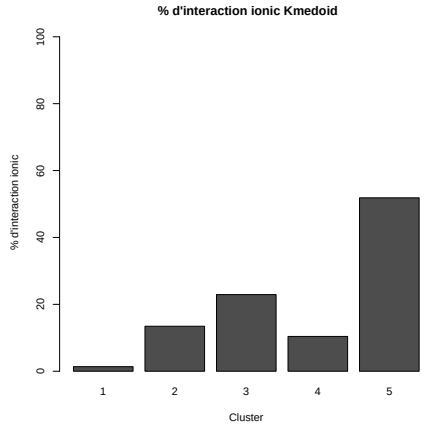
Figure 6: Caractérisation des clusters obtenus par K-medoids : Distribution (%) des autres caractéristiques (Accessibilité-**A**, Interaction Polaire-**B**, Interaction ionique-**C**, Interaction hydrophobe-**D**, Interaction de Van der Walls-**E**, Liaison hydrogènes-**F**) dans chaque cluster



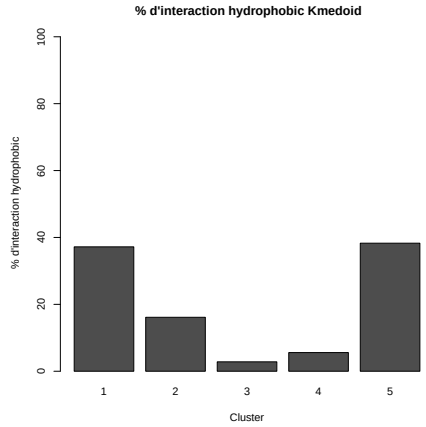
A.



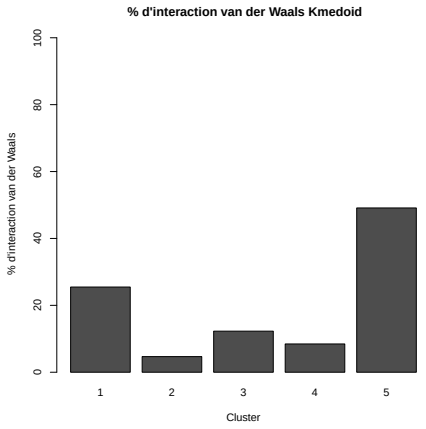
B.



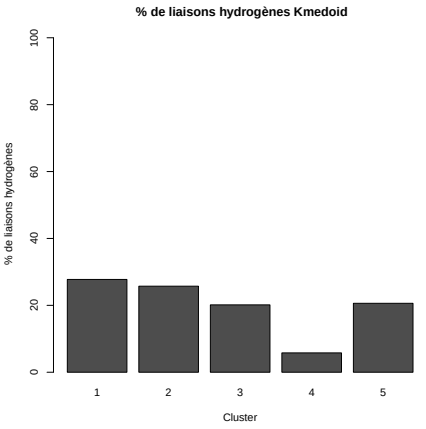
C.



D.

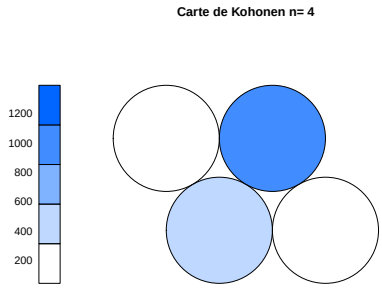


E.

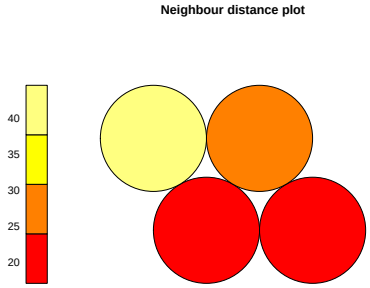


F.

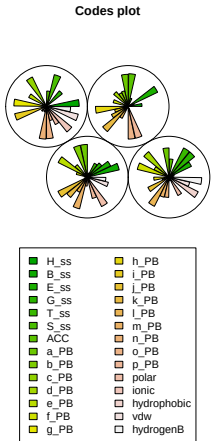
Figure 7: Carte de Kohonen avec 4 neurones : A = Count plot représente l'effectif dans chaque neurone, plus la couleur est foncé plus les effectifs sont élevé. B = Graphique U-matrice représente la distance au voisinage, plus les neurones sont proche plus la couleur est foncé. C = Codebook Vector correspond aux vecteurs des poids (profil) de chaque neurone dans un diagramme circulaire. D = Carte Heatmap (graphique par variables), en rouge les variables fortes et en bleu les variables faibles.



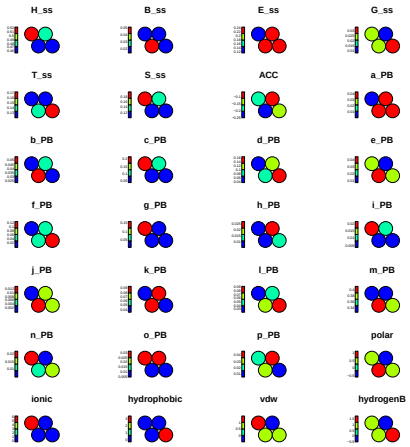
A.



B.



C.



D.

Figure 8: Carte de Kohonen toroïdal avec 4 neurones : A = Count plot, B = Graphique U-matrice, C = Codebook Vector, D = Carte Heatmap

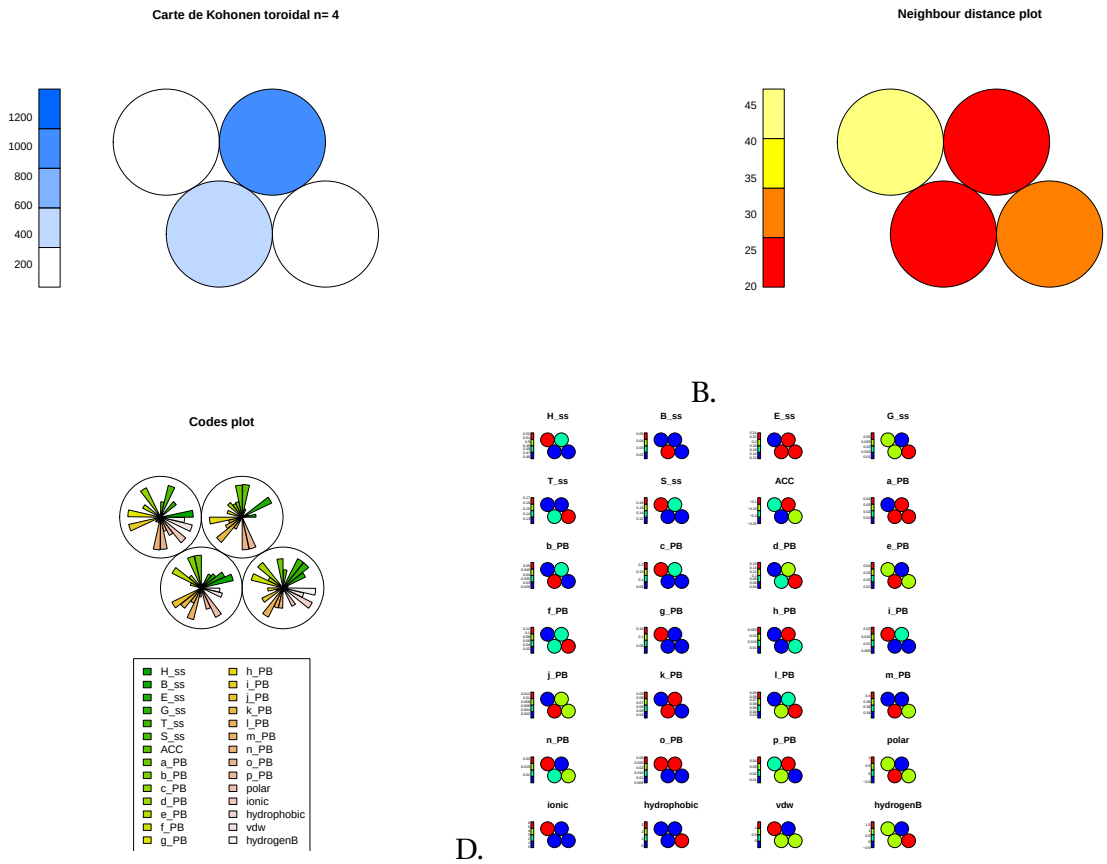
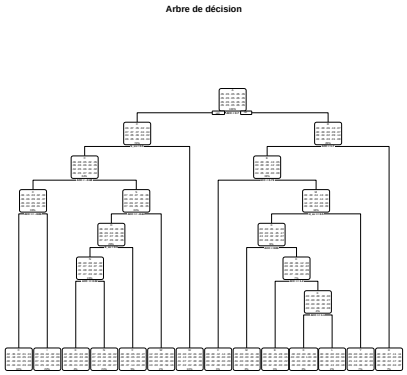
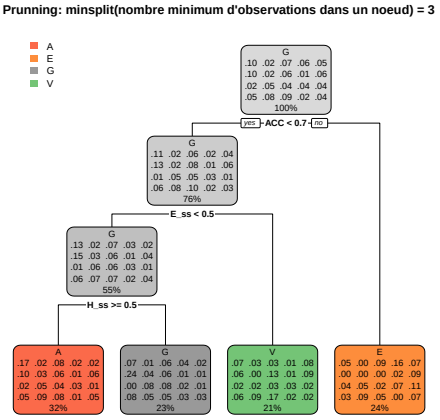


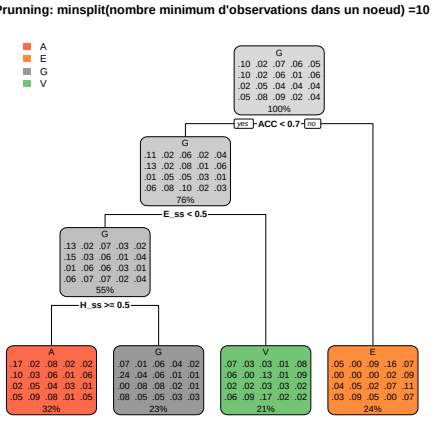
Figure 9: Prédiction de la séquence de la structure PDB 1a4fb (une globine) à l'aide d'un arbre de décision (*rpart*). A : Arbre de décision, la partie terminale de l'arbre (*feuille*) contient la fréquence prédite de chaque acide aminé dans l'ordre alphabétique, l'acide aminé le plus fréquent est écrit en haut de cette case. B : Arbre après élagage (Prunning), il y a 3 observations minimum par nœuds (option *minsplit*). C : Arbre après élagage, il y a 10 observations minimum par nœuds. D : Représentation en cartoon de 1a4fb. Protéine tout hélice (bleu), mise en évidence des α, Glutamate (sphère rouge) à la surface de la protéine.



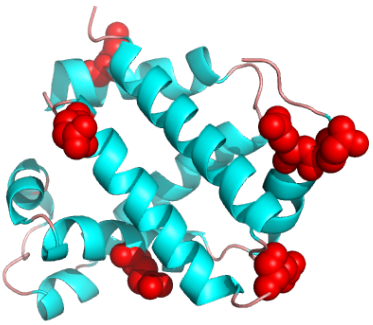
A.



B.

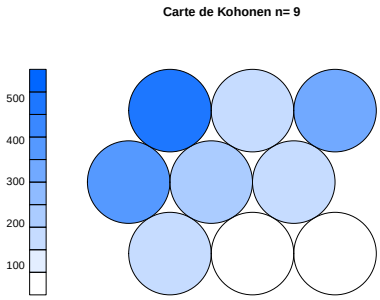


C.

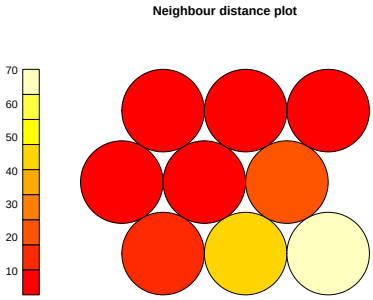


D.

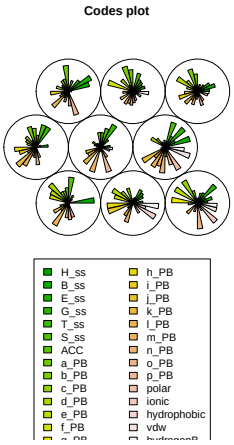
Figure 10: Carte de Kohonen avec 9 neurones : A = Count plot, B = Graphique U-matrice, C = Codebook Vector, D = Carte Heatmap



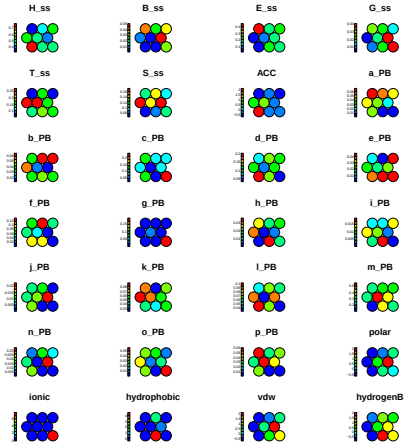
A.



B.

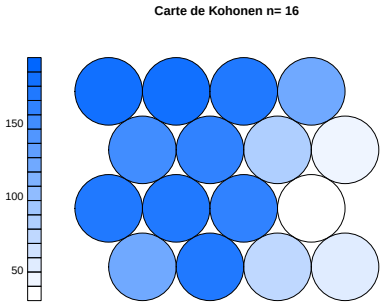


C.



D.

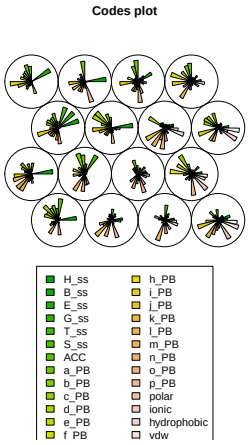
Figure 11: Carte de Kohonen avec 16 neurones: A = Count plot, B = Graphique U-matrice, C = Codebook Vector, D = Carte Heatmap



A.

B.

C.



D.

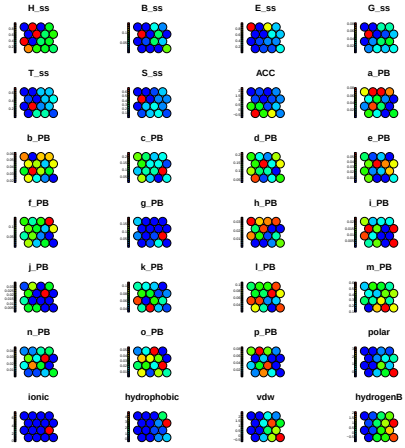
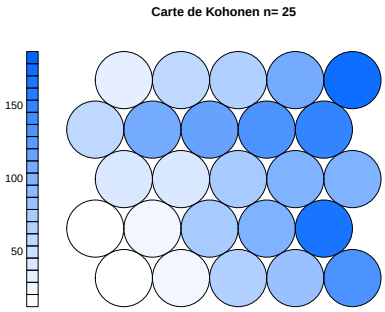
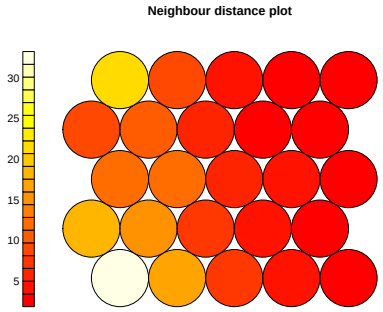


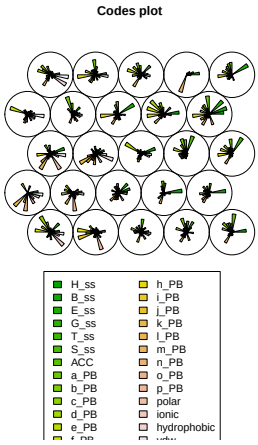
Figure 12: Carte de Kohonen avec 25 neurones: A = Count plot, B = Graphique U-matrice, C = Codebook Vector, D = Carte Heatmap



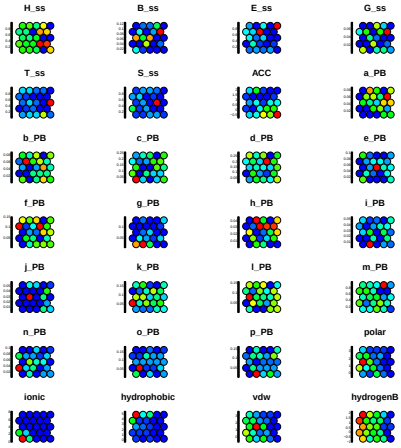
A.



B.

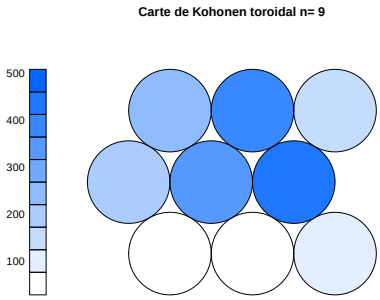


C.

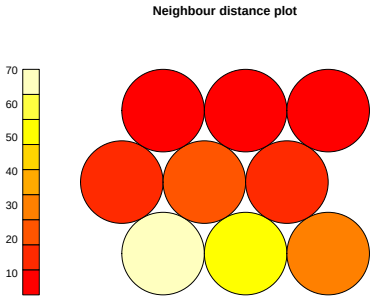


D.

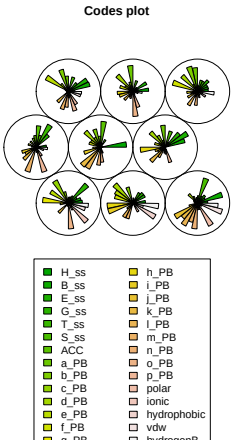
Figure 13: Carte de Kohonen toroidal avec 9 neurones: A = Count plot, B = Graphique U-matrice, C = Codebook Vector, D = Carte Heatmap



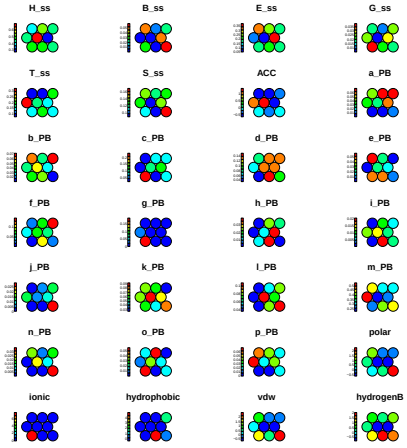
A.



B.

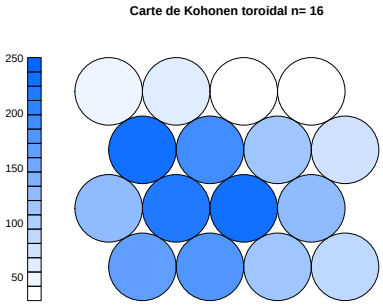


C.

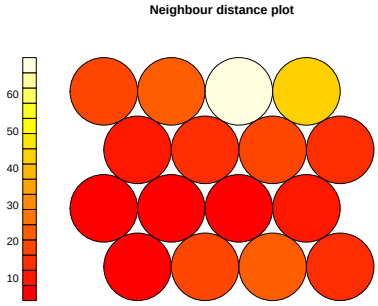


D.

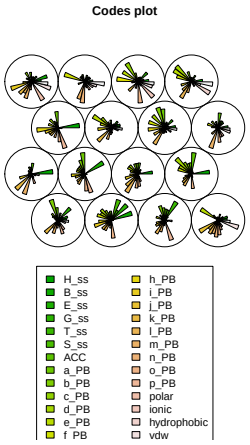
Figure 14: Carte de Kohonen toroidal avec 16 neurones: A = Count plot, B = Graphique U-matrice, C = Codebook Vector, D = Carte Heatmap



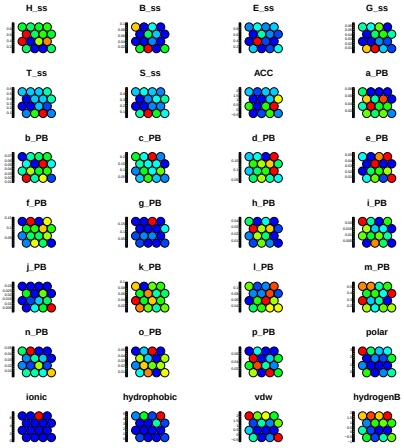
A.



B.

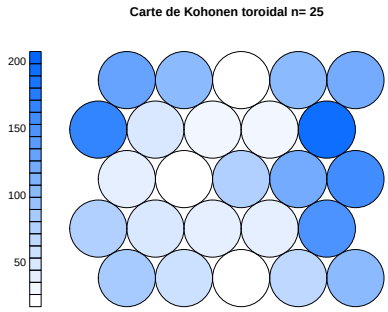


C.

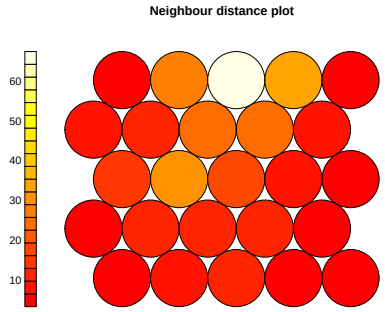


D.

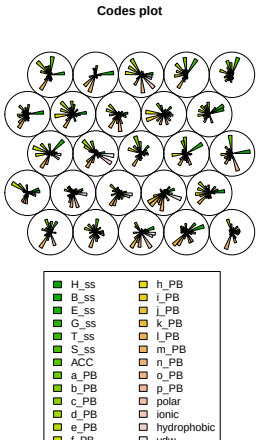
Figure 15: Carte de Kohonen toroidal avec 25 neurones: A = Count plot, B = Graphique U-matrice, C = Codebook Vector, D = Carte Heatmap



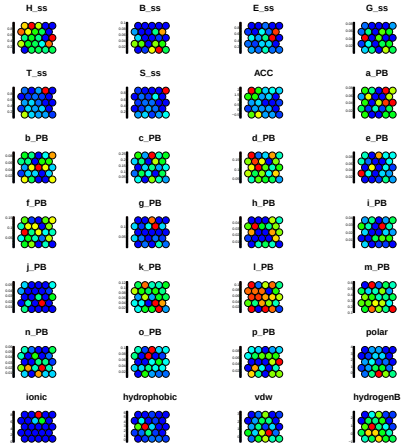
A.



B.



C.



D.

Figure 16: Multidimensional scaling: Diagramme de dispersion bidimensionnel.