

# Méthode de génération de score de compatibilité 1D-3D par un algorithme d'apprentissage automatique

Safia SAFA-TAHAR-HENNI




## Plan:

- Introduction
- Matériel & Méthodes
- Résultats
- Conclusion

# Introduction



- Connaissances sur le repliement des protéines => faciliter la conception de médicaments.
- Détermination du repliement d'une protéine => méthodes expérimentales moléculaires.
- Développement de techniques de séquençage de nouvelle génération = découverte de nouvelles séquences protéiques ↗↗

- 
- **Problème:** Utilisation de techniques expérimentales pour déterminer le repliement des protéines extrêmement difficile car techniques longues et coûteuses.
  - **Solution:** Développer des méthodes de prédiction par ordinateur capables de classer automatiquement, rapidement et avec précision des séquences de protéines inconnues dans des catégories de repliements spécifiques.



## Historique:

Méthodes actuelles de classification => basées sur un classificateur SVM.

Li et al (2013) méthode de reconnaissance des repliements PFP-RFSM:

- représentation des caractéristiques informations séquentielles et structurelles
- à partir des séquences primaires et des structures prédites

Lampros et al (2014) basé sur :

- chaîne de Markov formée avec la structure primaire des protéines
- HMM à espace réduit



## Les méthodes basées sur l'apprentissage automatique

Les méthodes basées sur l'apprentissage automatique peuvent être classées en deux classes :

- (1) Méthodes basées sur **un seul classificateur**: utilisent un seul algorithme d'apprentissage spécifique pour construire des modèles de prédiction,
- (2) Méthodes basées sur **un classificateur d'ensemble**: utilisent des algorithmes d'apprentissage multiples, similaires ou différents, pour construire des modèles de prédiction.

# Objectif:

- Déterminer et caractériser des ensembles ("cluster") à partir d'une base de données de vecteur d'environnements.
- Prédire l'environnement d'une protéine à partir de sa séquence à l'aide d'un arbre de décision.

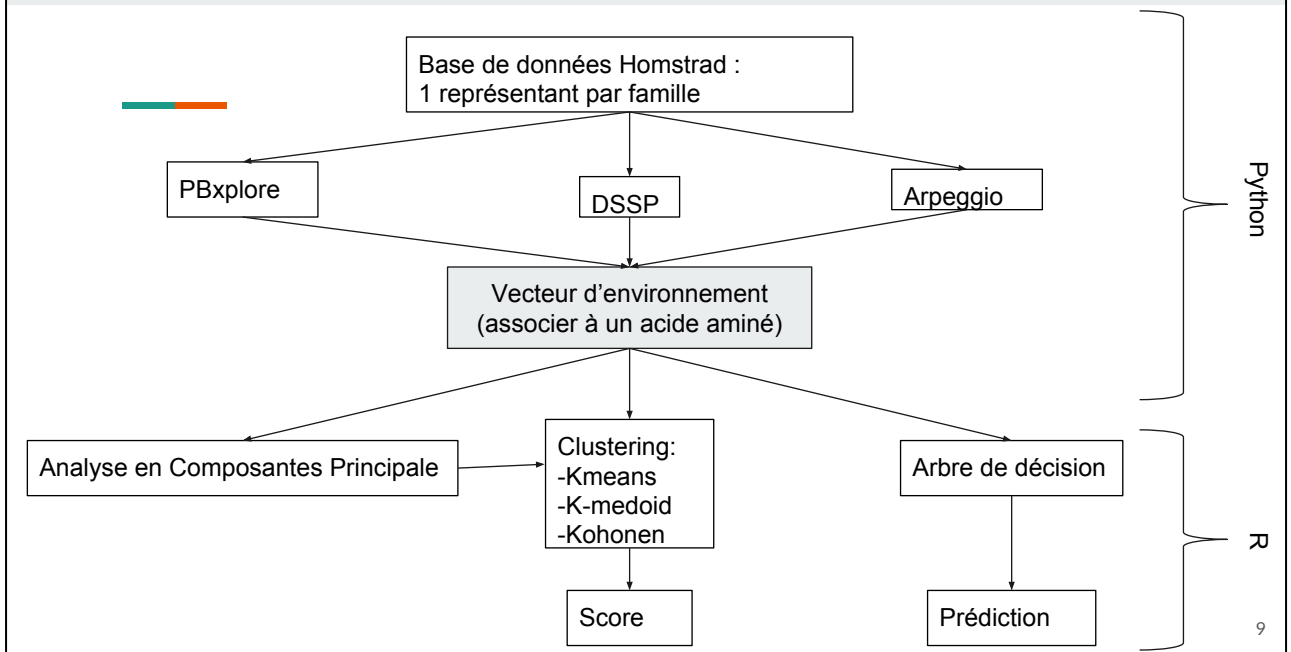
---

---

# Matériel & Méthodes



## Workflow:

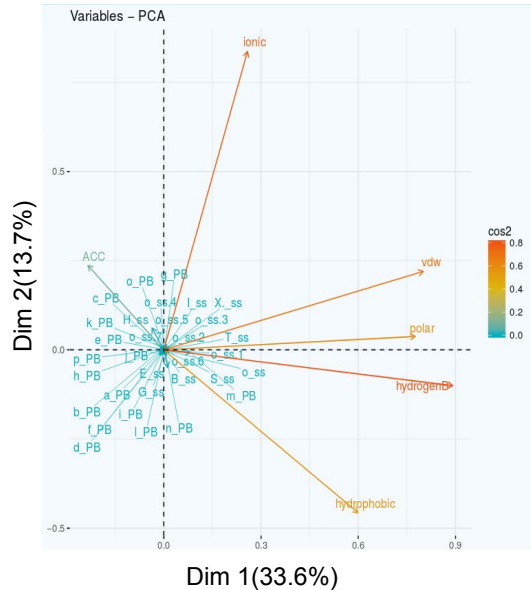
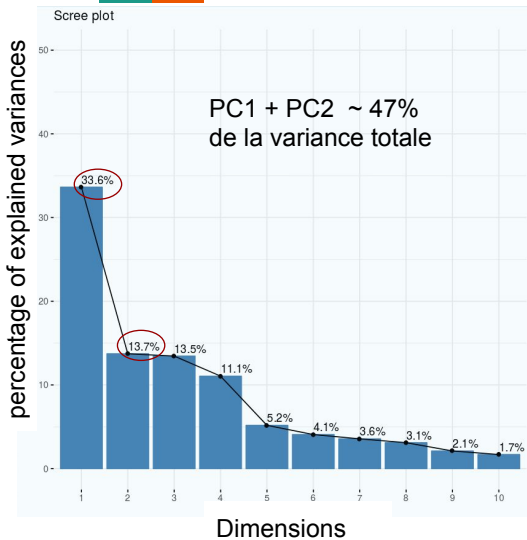


- Pour cela, la base de données utilisées est composée d'une structure de chaque famille de la base de donnée Homstrad.
  - À chaque acide aminé (de chaque séquence de chaque structure) est associé un vecteur d'environnement.
  - Ce vecteur est composé d'information sur:
    - la structure secondaire, l'accessibilité au solvant, => DSSP
    - la déformabilité de la protéines (grâce aux alphabets structuraux), => PBxplore
    - et le nombre et types d'interaction créés. => Arpeggio
- => développé en python
- Sur ce vecteur vont être appliqué les analyses suivantes:
    - Analyse en composantes principale (réduction dimension)
    - Regroupement par Kmeans (*kmeans*) /Kmedoid (*pam*) /Carte de Kohonen (*kohonen*) => caractérisation + score
    - Prédiction => arbre de décision (*rpart*)

---

# Résultats

# PCA



11

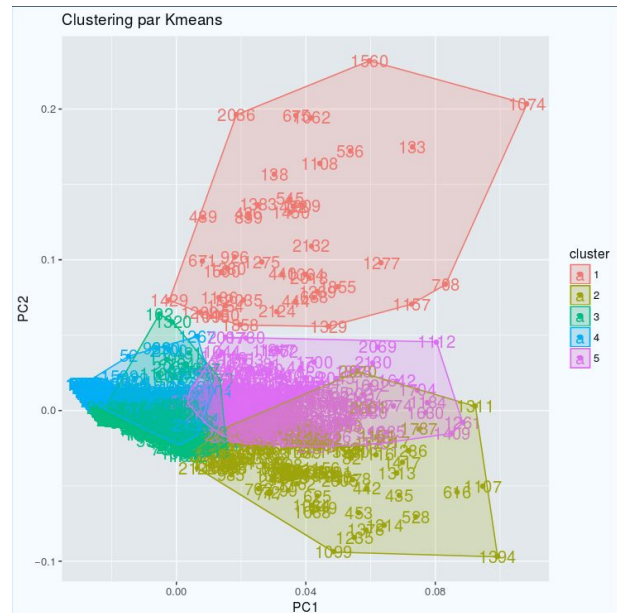
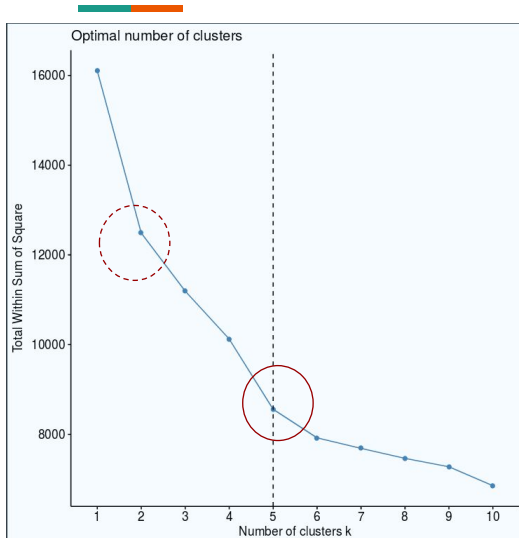
PC 1 et PC 2 ne suffisent pas pour représenter la majorité de la variance (~ 47\% de la variance totale).

Graphique de corrélation des variables :

- positivement corrélées = regroupées (ex: liaisons hydrogènes, Van der Waals et polaire).
- négativement corrélées = quadrants opposés (ex: accessibilité au solvant et hydrophobicité).
- variables loin de l'origine = bien représentées (ex: interaction ionique)
- $\cos^2$  élevé = bonne représentation variable sur PC1/PC2 (ex: liaisons hydrogènes, Van der Waals, hydrophobe, ionic et polaire).
- faible  $\cos^2$  = variable pas parfaitement représentée par PC1/PC2 (ex: structure secondaire et Protein Block).

# Clustering:

## • K-means



12

Kmeans: 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. Il classe les objets dans plusieurs groupes (clusters), de sorte que les objets d'un même cluster soient aussi similaires que possible, alors que les objets des différents clusters sont aussi différents que possibles. Chaque cluster est représenté par son centre (ou centroïde) qui correspond à la moyenne des points assignés au cluster => définir des clusters afin de minimiser la variation totale intra-cluster.

Méthode du "coude" -> déterminer le nombre de groupes optimaux:

- présence de coude pour 2 groupes ou 5 groupes.
- hypothèse nombre de groupes = 5.

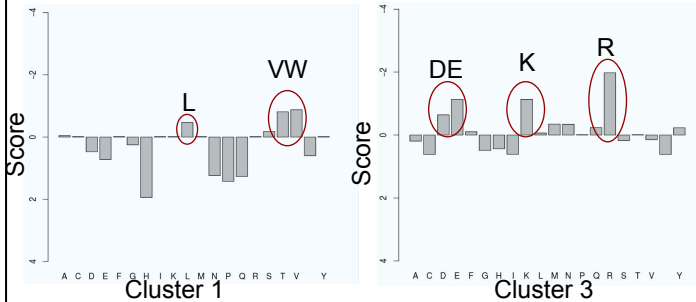
Graphique de regroupement par Kmeans :

- le groupe 1 se distingue bien des autres groupes
- les quatre autres groupes se confondent sur le graphique

=> 2 groupes semblent être plus appropriés.

# Scoring :

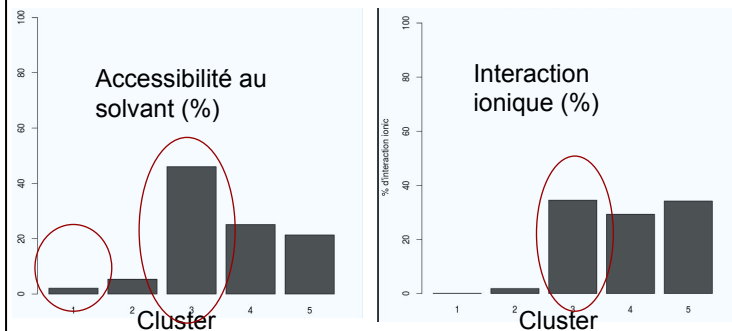
Distribution des scores



$$Score = \log_2 \left( \frac{\text{Fréquence observée}}{\text{Fréquence attendue}} \right)$$

**Fréquence observée** = Fréquence de l'acide aminé dans le cluster.

**Fréquences attendues** = Fréquence de l'acide aminé dans la base de données



## Cluster 1:

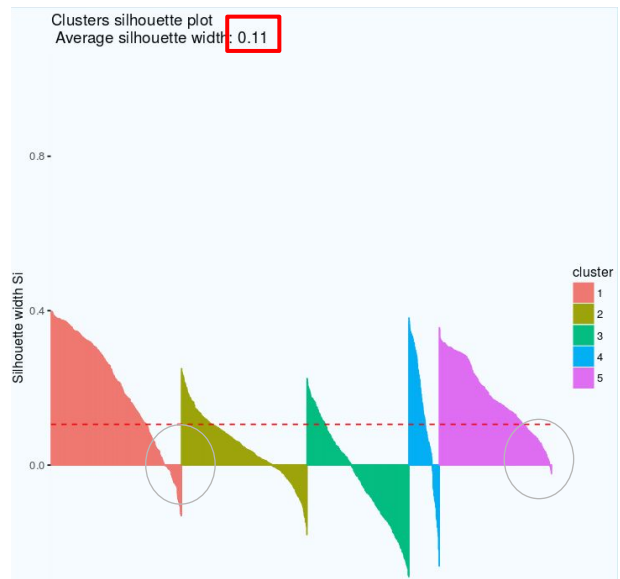
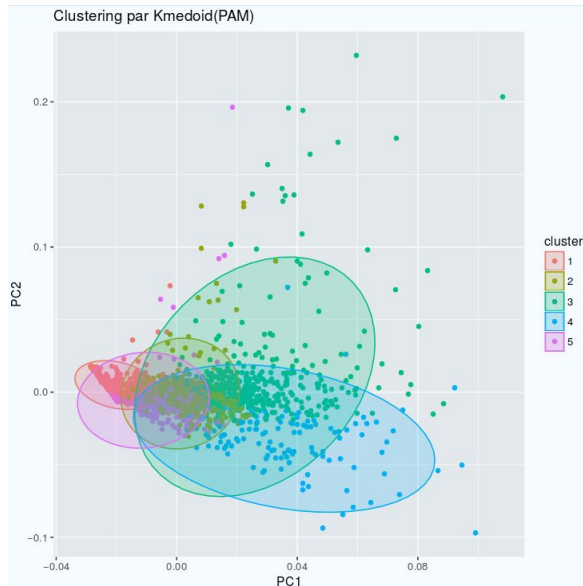
- acide aminé hydrophobe (lysine, valine, tryptophane).
- % accessibilité au solvant très faible

## Cluster 3:

- acide aminé chargé (glutamate, lysine et arginine),
- % interaction ionique très fort
- % accessibilité au solvant élevé.

# Clustering:

## • K Medoid



14

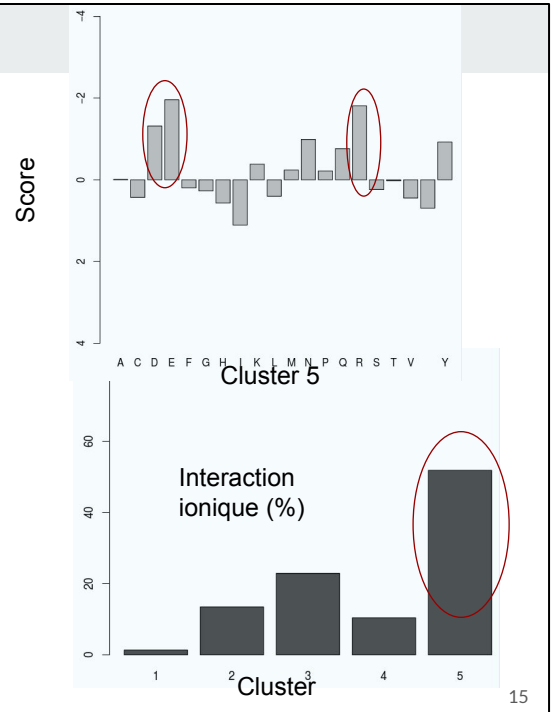
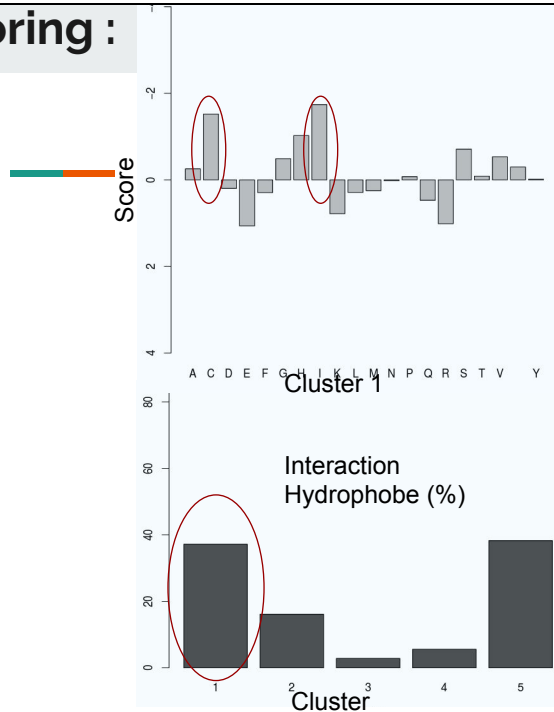
K-medoid: un cluster est représenté avec l'objet le plus proche au centre du cluster.  
Il est plus robuste que k-means en présence de valeurs aberrantes.

Les groupes se confondent sur le graphique.

- $si \approx 1$ : observations correspondantes sont très bien groupées,
- $si \approx 0$ : l'observation se situe entre deux groupes
- $si < 0$ : probablement placées dans le mauvais groupe.

Cluster 1 et 5 pas (ou très peu) de  $si < 0 \Rightarrow$  observations très bien groupées.

## Scoring :



15

Cluster 1 :

- acide aminé polaire(cystéine) et hydrophobe (isoleucine).
- % d'interaction hydrophobe élevé

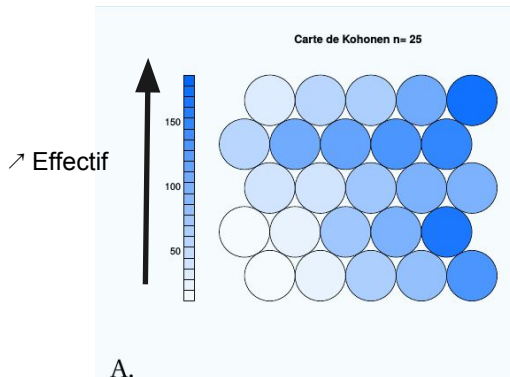
Cluster 5 :

- acide aminé chargé (aspartate, Glutamate et Arginine)
- fort % d'interaction ionique.

# Clustering:

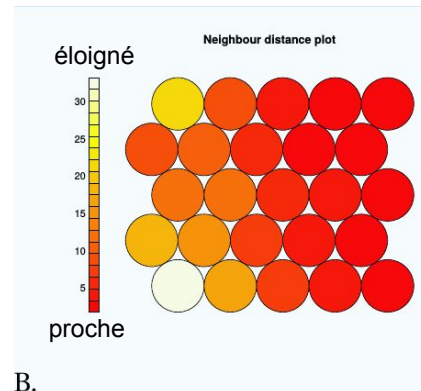
- **Carte de Kohonen** classique

Carte de Kohonen avec 25 neurones :



A.

A : Count plot (= effectif dans chaque neurone)



B.

B : Graphique U-matrice (= distance au voisinage).

16

Carte de kohonen: des réseaux de neurones orientés à deux couches : l'entrée correspond à la description des données, la sortie est organisée sous forme de grille et symbolise une organisation des données.

Taille des grilles hexagonale est de : 2x2, 3x3, 4x4 et 5x5

A: permet identifier zones de fortes densité:

- idéal: répartition homogène -> minimum 25 neurones

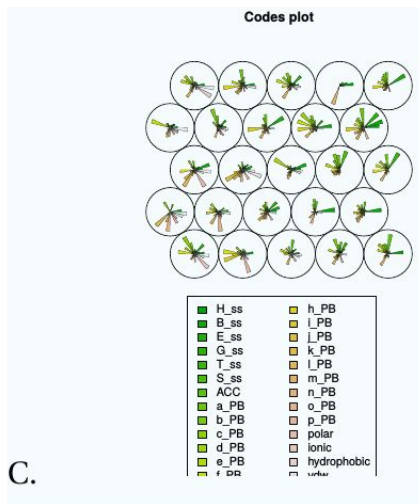
B: somme au voisin immédiat pour chaque neurone.

- neurones d'une même classe -> rouge foncé
- neurones claire -> délimiter zone frontière.

=> Avec 25 neurones pas une bonne séparation des classes.

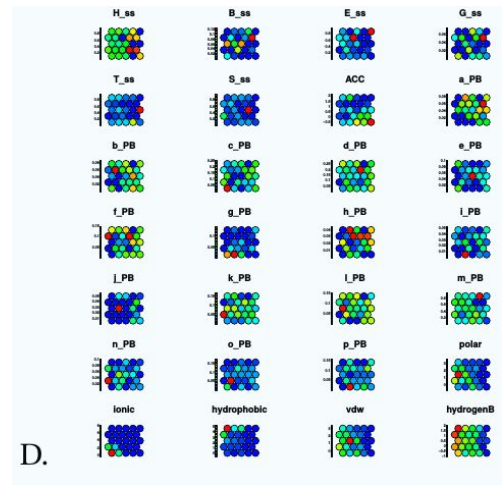


## ● Carte de Kohonen classique (2)



C.

C : Codebook Vector = aux vecteurs des poids de chaque neurone dans un diagramme circulaire.



D.

D = Carte Heatmap (graphique par variables):

- rouge variables fortes
- bleu variables faibles

17

C: permet établir rôle variables , distinguer différentes zones grâce aux variables "actives".

- Neurones NE : structures secondaires.
- Neurones O: Protein Block.
- Neurones S: différents types d'interaction.

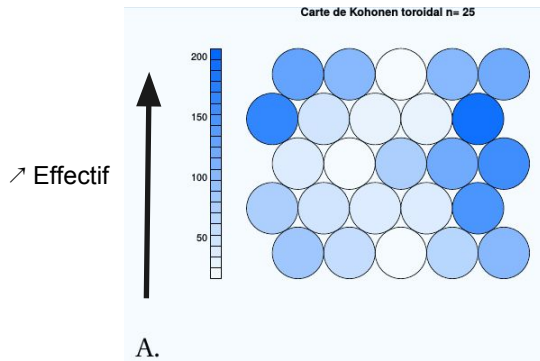
D: permet préciser variables associé à chaque zone:

- Accessibilité au solvant (SE) et Hydrophobicité (NO) actives -> opposé.

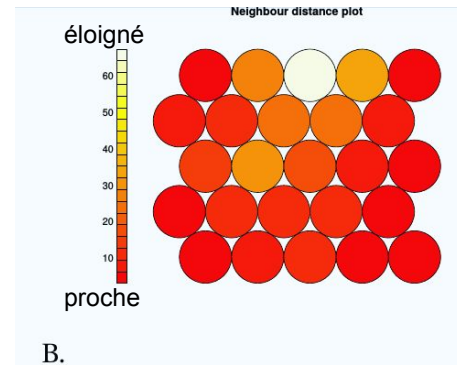
# Clustering:

- **Carte de Kohonen** toroïdale

Carte de Kohonen toroïdale avec 25 neurones :



A : Count plot ( = effectif dans chaque neurone)



B : Graphique U-matrice ( = distance au voisinage).

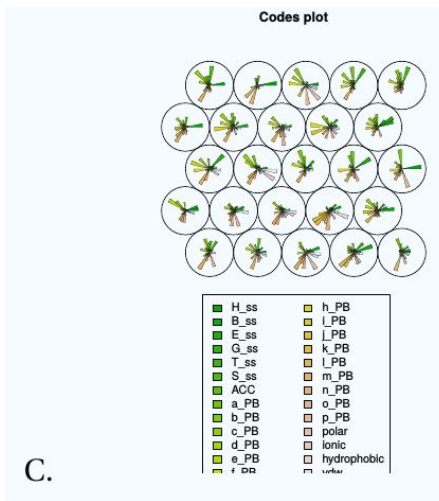
18

Variante architecturale de la carte de Kohonen => toroidal = le bord gauche sera voisin du bord droit et le bord haut celui du bord bas.

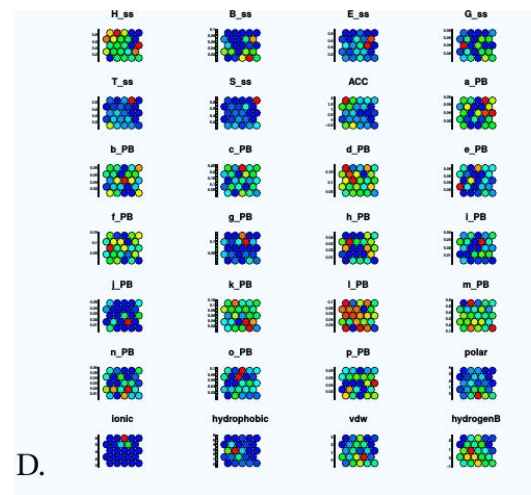
A : Le nombre idéal de neurone: entre 16-25 neurones .

B : Caractéristiques carte toroïdale : neurones foncés aux extrémités de la carte = meilleure séparation des classe .

## ● Carte de Kohonen toroïdale (2)



C : Codebook Vector = aux vecteurs des poids de chaque neurone dans un diagramme circulaire.



D = Carte Heatmap (graphique par variables):

- rouge variables fortes
- bleu variables faibles

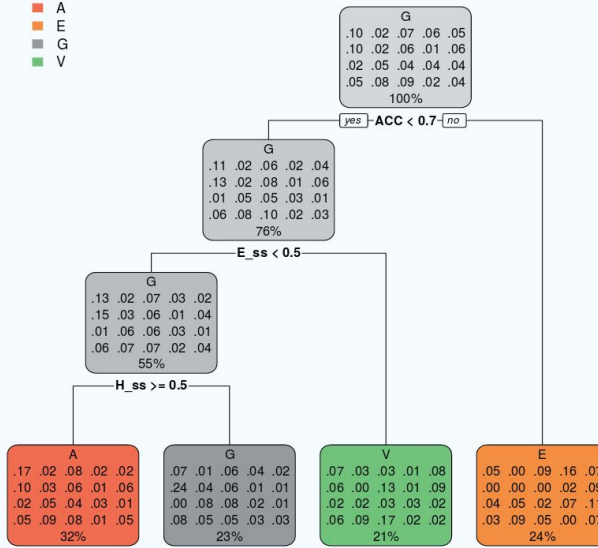
19

La Codes Map et la Heatmap sont assez difficiles à caractériser en effet la majorité des caractéristique associé au vecteur d'environnement se répartissent uniformément sur l'ensemble des neurones mis à part la structuration en "bend" (S\_ss), les interactions ionique ou le protein Block j

# Prédiction de la séquence de la structure PDB 1a4fb (une globine):

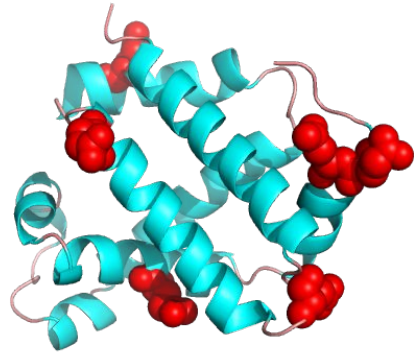
Prunning: minsplit(nombre minimum d'observations dans un noeud) = 3

■ A  
■ E  
■ G  
■ V



Arbre de décision (après élagage):

- feuille -> fréquence prédite de chaque AA (ordre alphabétique)
- AA le + fréquent écrit en haut



Représentation en cartoon de 1a4fb.

- hélice (bleu),
- Glutamate (sphère rouge)

20

Prédiction Alanine conditionnée par:

- faible accessibilité au solvant, => AA hydrophobe
- forte probabilité de présence dans hélice alpha => Alanine favorise formation hélices alpha.

Prédiction Glycine conditionnée par:

- faible probabilité de présence dans une structure en hélice alpha => Glycine déstabilise hélices alpha

Prédiction Valine conditionnée par:

- faible accessibilité au solvant => AA hydrophobe
- forte probabilité de présence dans feuillet beta => Chaînes latérales ramifiées cœur des feuillets beta.

Prédiction Glutamate conditionnée par:

- forte accessibilité au solvant => AA chargé + Probabilité élevé surface d'une protéine.

# Conclusion



**Objectifs:** déterminer et caractériser des environnements

=> approche novatrice : utilisation Protein Block nouveauté dans ce genre d'analyse.

*Le plus:* approche non supervisée -> mise en évidence, sans apriori, caractéristiques aides à la prédiction.

Arbre de prédiction: méthode **rapide et efficace** -> prédire l'environnement d'une protéine à partir de sa séquence.

## Améliorations:

- Intégration de plus de structure dans la base de données.
- Autres méthodes de Pruning (ex: validation croisée).
- Long terme: une prédiction par réseau de neurones

# Merci pour votre attention

