Apprentissage non supervisé
Partie 2

Analyse de données et Classification 2
ENSEEIHT - 3ème année Sciences du Numérique

Contact :
Sandrine.Mouysset@irit.fr
sandrine.mouysset@toulouse-inp.fr

Chaîne d'analyse des données

- Approches par partitionnement :
  - K-ppv
  - K-means
  - Classification hiérarchique
  - DBScan
  - Approche par noyau : Classification spectrale
  - Approche par graphe : Modularité

- Méthodes d'évaluation non supervisée de la partition :
  - Cohesion, Séparation
  - SSW, SSB et variantes

Méthodes à noyaux (kernel-based methods) : elles sont basées sur le principe de séparation entre la représentation des données à l'aide des noyaux (fonction de comparaison entre deux données) et les algorithmes utilisant uniquement des évaluations de noyaux.

Méthodes à noyaux libèrent la nature de la séparation entre les classes

On considère un ensemble de points $x = \{x_1 \dots x_n\} \in \mathbb{R}^p$. On peut représenter ces données dans une matrice de produit scalaire K définie par :

$$K_{ij} = k(x_i, x_j) \text{ où } k \text{ est une } \textit{fonction noyau}.$$

Avantages des méthodes à noyaux :

- La représentation est la même quel que soit le type de données (image, video, texte...).
- C'est également le même algorithme de classification qui sera utilisé quel que soit le type de données (algorithme modulaires et généraux)
- Les choix de la matrice de produit scalaire et celui de la méthode de classification sont découplés (principe de séparation) :

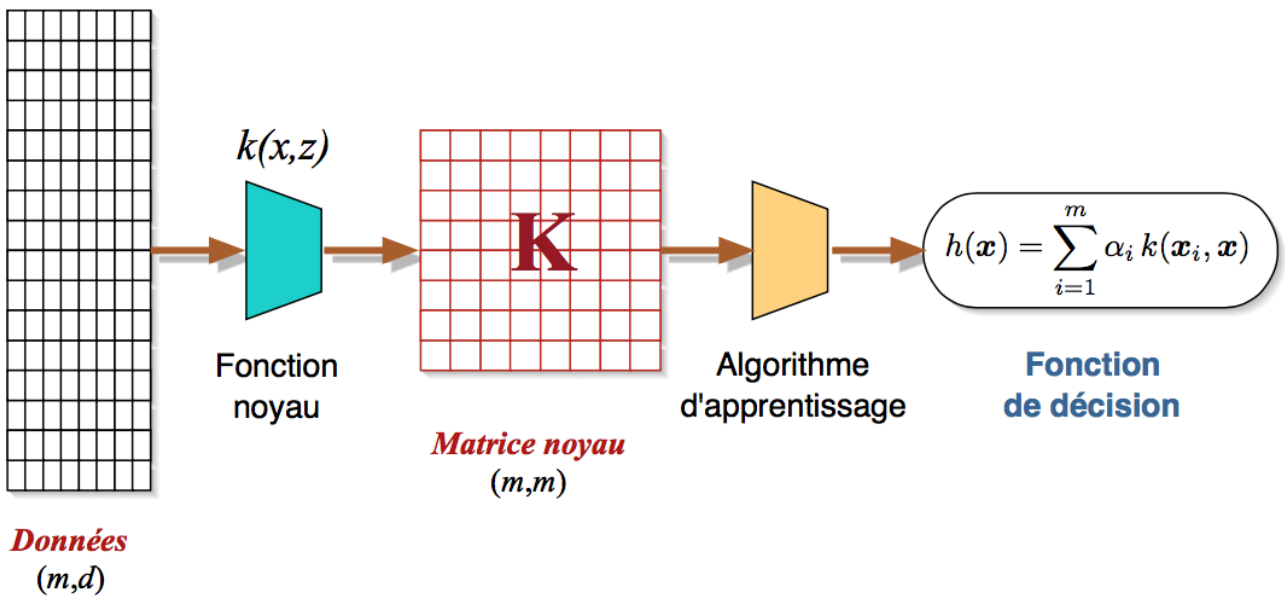


Figure: Chaîne de traitement générique des méthodes à noyaux

Avantages des méthodes à noyaux :

- Les méthodes de classification disposent d'une version noyau : *kernel ACP* où la matrice noyau remplace Σ, *kernel k-means* où K remplace la distance euclidienne...
- De même, les problèmes de régression et de classification supervisée ont une version non linéaire : Kernel SVM, régression non linéaire...

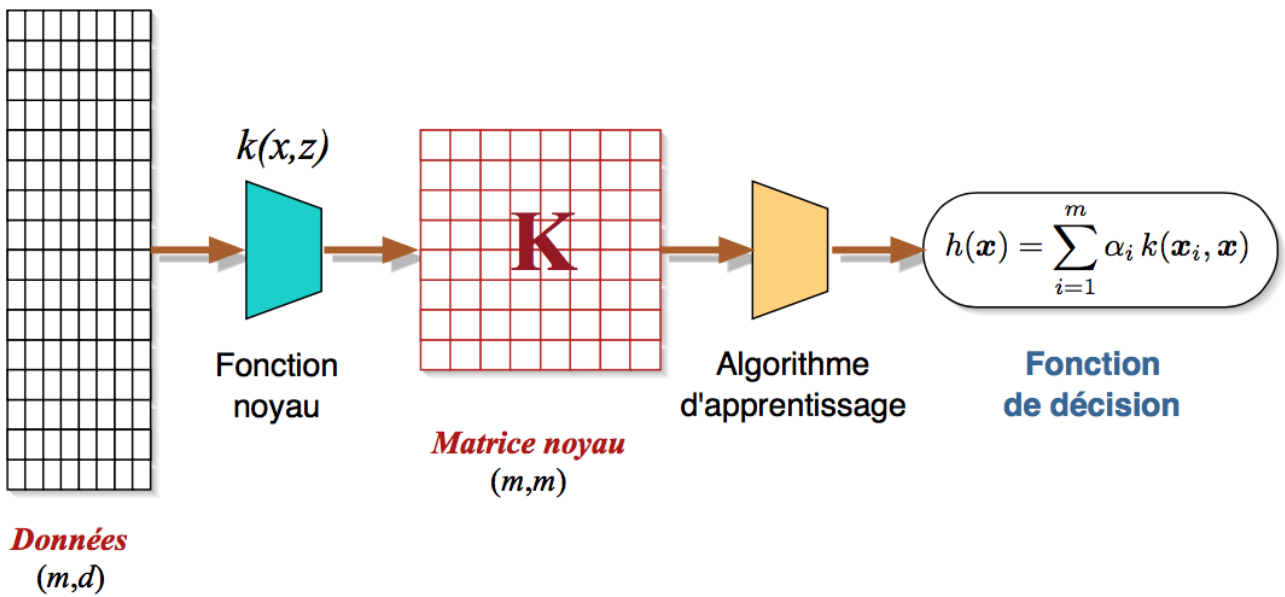


Figure: Chaîne de traitement générique des méthodes à noyaux

Fonction symétrique positive/Fonction noyau

On dit qu'une fonction $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ de deux variables est une **fonction noyau** si et seulement elle est symétrique positive :

- $\forall x, y \in \mathcal{X},\ k(x, y) = k(y, x)$
- $\forall a_1 \dots a_m \in \mathbb{R}$ et les vecteurs $x_1 \dots x_m \in \mathcal{X}$, $\sum_{i,j} a_i a_j k(x_i, x_j) \geq 0$

On dira que la **matrice noyau**, appelée aussi **matrice de Gram**, de m vecteurs est la matrice symétrique positive associée au noyau :

$$K = [k(x_i, x_j)]_{i,j} \in \mathcal{S}_+^d$$

où $\mathcal{S}_+^m = \left\{M | \forall a \in \mathbb{R}^m, a^T M a \geq 0\right\}$ est l'ensemble des matrices symétriques positives.

Polynomial Kernel	$\kappa(\mathbf{a}, \mathbf{b}) = (\mathbf{a} \cdot \mathbf{b} + c)^d$
Gaussian Kernel	$\kappa(\mathbf{a}, \mathbf{b}) = \exp(-\|\mathbf{a} - \mathbf{b}\|^2 / 2\sigma^2)$
Sigmoid Kernel	$\kappa(\mathbf{a}, \mathbf{b}) = \tanh(c(\mathbf{a} \cdot \mathbf{b}) + \theta)$

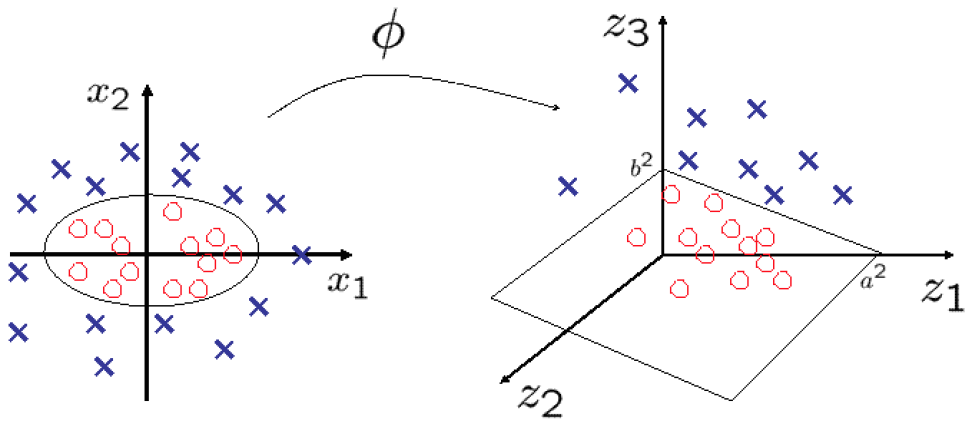
Figure: Exemples de fonctions noyaux k et leurs hyperparamètres respectifs

Exemple :
Soit $\mathcal{X} = \mathbb{R}^2$, on définit la fonction ϕ par :

$$\begin{array}{rcl} \phi : \mathbb{R}^2 & \rightarrow & \mathbb{R}^3 \\ (x_1, x_2) & \mapsto & (z_1, z_2, z_3) = (x_1^2, \sqrt{2}x_1x_2, x_2^2) \end{array}$$

Alors pour $x \in \mathbb{R}^2$ et $y \in \mathbb{R}^2$, $k(x, y) =< \phi(x), \phi(y) >= (x.y)^2$ est un noyau polynomial.
La séparation elliptique dans l'espace $\mathcal{X}$ devient un hyperplan dans $\mathbb{R}^3$:

$$\left(\frac{x_1}{a}\right)^2 + \left(\frac{x_2}{b}\right)^2 = 1 \Longrightarrow \frac{z_1}{a^2} + \frac{z_3}{b^2} = 1$$



Au lieu de chercher un hyperplan dans l'espace des entrées, on passe d'abord dans un espace de représentation intermédiaire (*feature space*) de plus grande dimension.

$$\begin{array}{rccc} \phi : & \mathcal{X} & \rightarrow & \mathcal{F} \\ & x & \mapsto & \phi(x). \end{array}$$

Une distance euclidienne s'exprime en fonction de produits scalaires :

$$d(x,y)^2 = < x,x > + < y,y > -2 < x,y >$$

Si on dispose d'un noyau on a donc une distance généralisée, qui s'exprime en fonction des évaluations des noyaux :

$$\begin{aligned} \widehat{d}(x,y)^2 &= < \phi(x),\phi(x) > + < \phi(y),\phi(y) > -2 < \phi(x),\phi(y) > \\ &= k(x,x) + k(y,y) - 2k(x,y) \end{aligned}$$

Algorithm 3 Algorithme Spectral Clustering

Input : $S = \{x_1 \dots x_n\}$, $x_i \in \mathbb{R}^p$, k nombre de classes.

1. *Construction de la matrice affinité/similarité*

$$A_{ij} = \begin{cases} exp(-\frac{\|x_i - x_j\|^2}{2\sigma^2}) \text{ si } i \neq j \\ 0 \text{ sinon} \end{cases}$$

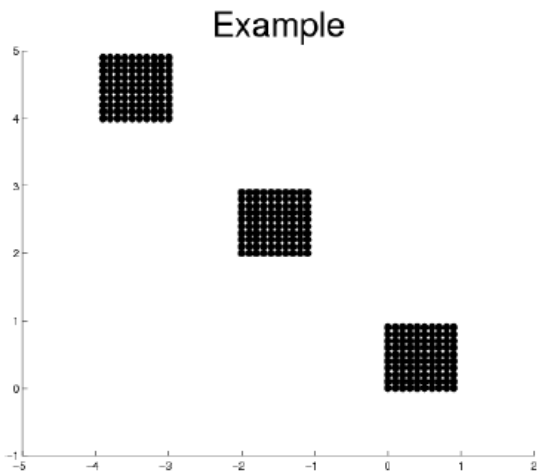
2. *Normalisation :* $L = D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$
   où D est la matrice diagonale d'éléments $D_{ii} = \sum_{j=1}^{n} A_{ij}$.
   L est donc stochastique (sa plus grande valeur propre est 1).

3. *Extraction des k plus grands vecteurs propres (associés aux k plus grandes valeurs propres)*
   $X = [X_1 \dots X_k] \in \mathbb{R}^{n \times k}$

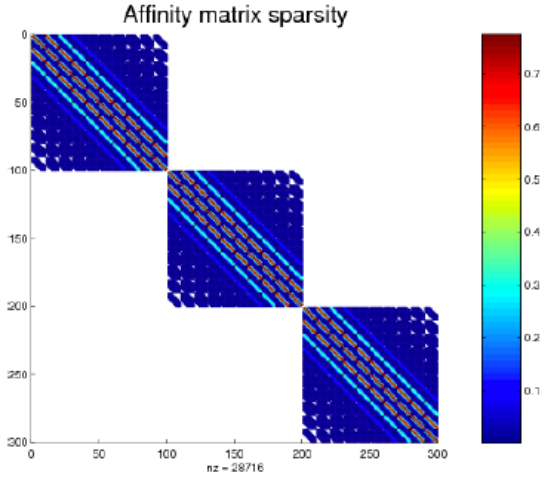
4. *Normalisation des lignes de* X *:* $\forall i \in \{1..n\}, Y_i = \frac{X_i}{\sum_{j=1}^{k} X_{ij}}$

5. *Classification par k-means* dans l'espace de projection spectrale.
   $Y = [Y_1 \dots Y_k]$

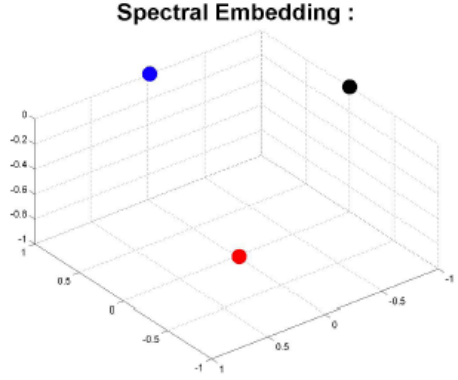
6. *Classification des données d'origine* via la relation d'équivalence suivante :
   La ligne i de Y est assignée à $\mathcal{C}^i$ si et seulement si x_i est assignée à la classe $\mathcal{C}^i$.



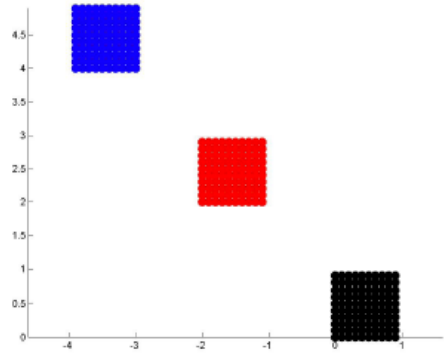
(a) Exemple pour $N = 300$ points



(b) Matrice affinité quasi bloc-diagonale (tronquée en dessous de 10^{-15})



(c) Lignes de Y

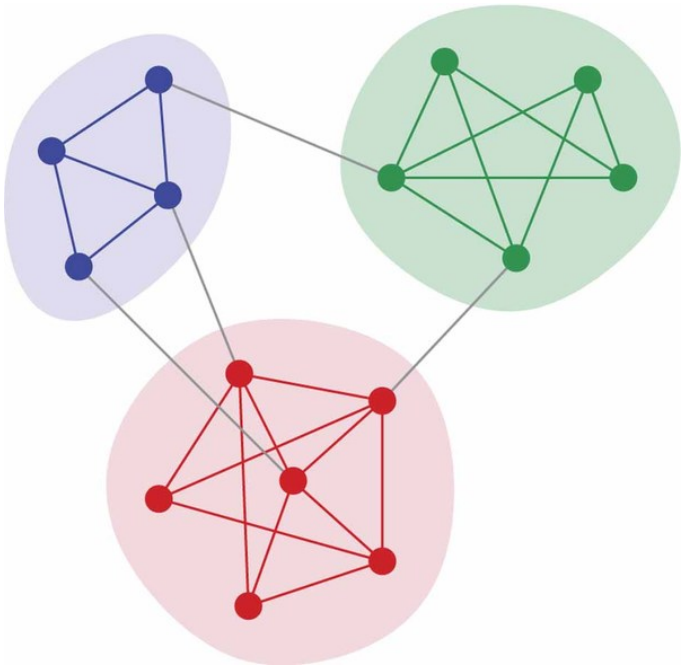


(d) 3 clusters séparés obtenus à la fin

FIGURE 1.1 – Illustration des étapes du clustering spectral

... vers un point de vue graphe où une classe représente une communauté :

Une **communauté** se définit par rapport à un graphe courant comme un groupe de noeuds qui sont particulièrement reliés entre eux et faiblement reliés au reste du réseau.

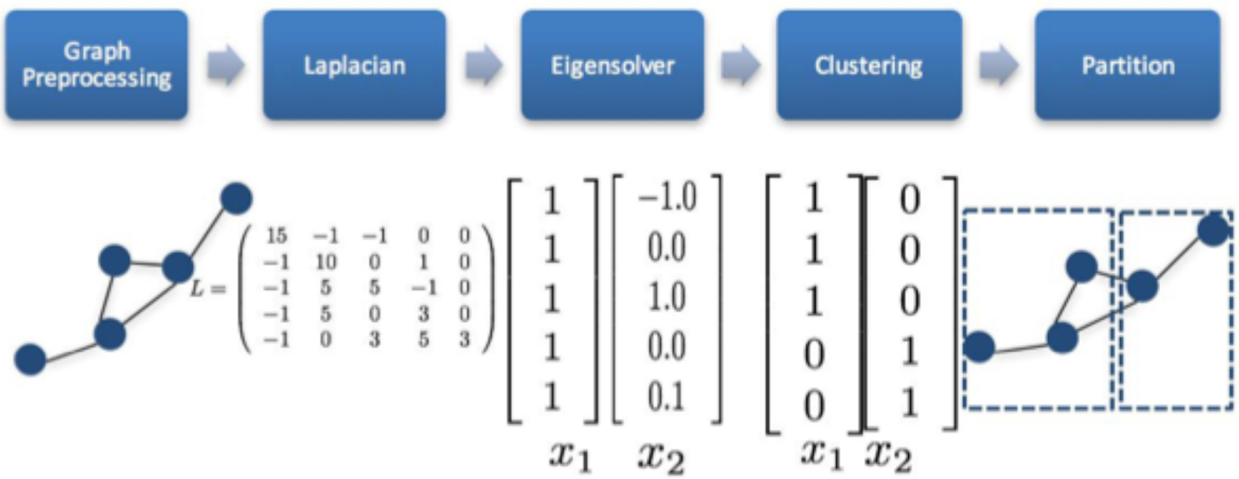


L'approche traditionnelle est de **minimiser le taux de coupe** (i.e le nombre d'arêtes inter-communautés) en réalisant un partitionnement, avec comme inconvénient principal de devoir choisir en amont le nombre de communautés.
⇒ *Approche alternative visant à trouver des sous-groupes densément connectés.*

On souhaite partitionner en se basant sur un critère de coupe, i.e en coupant les arêtes ayant un poids faible, avec un choix ex ante du nombre de communautés.

Algorithm 4 Algorithme Spectral Clustering - graphe

1. On calcule la matrice Laplacienne, qui se définit telle que : $L = D - A$ où D est la matrice des degrés, et A la matrice d'adjacence. Il convient de la normaliser en cas de forte hétérogénéité des degrés.
2. On calcule alors les k vecteurs propres de L correspondant aux k plus petites valeurs propres.
3. On effectue un k-means sur la matrice contenant en colonnes les k vecteurs propres.



Des variantes existent, notamment sur le type de normalisation, ou en prenant la **matrice de modularité du graphe** à la place du Laplacien définie par :

$$M = A - \frac{kk^T}{2m}$$

où

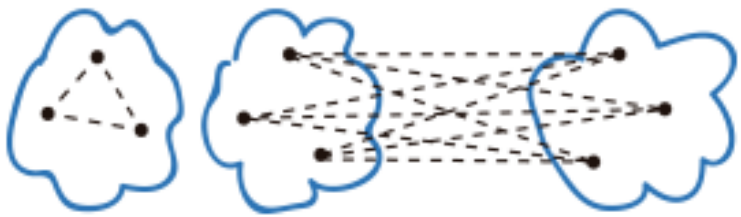
- A matrice d'adjacence symétrique $n \times n$
- k sont des vecteurs de n éléments, en l'occurence les degrés des noeuds (c-à-d le nombre de contacts directs)
- $2m = \sum_{i=1}^{n} k_i$.

$\Rightarrow$ issue de la **mesure de modularité** définie pour la qualité d'un partitionnement des noeuds d'un graphe, ou réseau, en communautés.

Comment évaluer les partitions d'une classification non supervisée ?

Cohésion et Séparation

- la cohésion d'une classe mesure comment les objets d'un cluster sont étroitement liés;
- la séparation d'une classe mesure comment une classe est distincte ou bien séparée de l'autre.



Pour une classe C_i,

$$Cohesion(C_i) = \sum_{x \in C_i, y \in C_i} prox(x, y)$$

$$Separation(C_i, C_j) = \sum_{x \in C_i, y \in C_j} prox(x, y), \ i \neq j$$

où *prox* est une fonction de proximité (similarité, dissimilarité, distances...)

- Sum of Squared Error (SSE) ou Sum of Squared Errors Within Cluster (SSW) correspond à la mesure de cohésion où *prox* est la distance euclidienne (à minimiser)

$$SSE(C_i) = \sum_{x \in C_i} d(c_i, x)^2 = \frac{1}{2|C_i|} \sum_{x \in C_i} \sum_{y \in C_i} d(x, y)^2$$

où x est un élement de C_i, c_i est le centre de C_i et $|C_i|$ le cardinal de la classe C_i.

- Between group Sum of Squares (BSS) basé sur la séparation (à maximiser)

$$BSS = \sum_{i=1}^{K} |C_i| d(c_i, c)^2$$

où c est le centroid de l'ensemble de données.

- Coefficient Calinski-Harabasz (CH) est un ratio (à maximiser) entre la variance inter-classe et la variance intra-classe :

$$CH = \frac{\frac{BSS}{K-1}}{\frac{SSE}{K}}$$

- Indice de Hartigan basé sur le logarithme du ratio BSS/SSE :

$$H = \log\left(\frac{BSS}{SSE}\right)$$

- Indice de Dunn est le ratio de la plus petite distance entre les données de différents groupes et de la plus grande distance entre les groupes.
  Pour une classe C_i,

$$D_i = \frac{\min_{1<j<K, i\neq j} \delta(C_i, C_j)}{\max_{1<l<K} \Delta(C_l)}$$

avec $\Delta(C_i) = \max_{x,y\in C_i} d(x,y)$ et $\delta(C_i, C_j) = \min_{x\in C_i, y\in C_j} d(x,y)$.

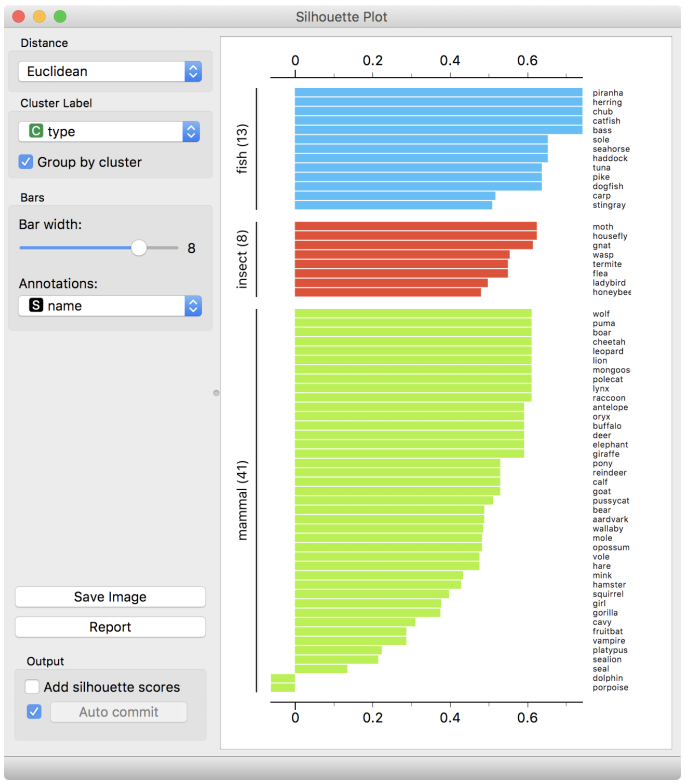
- Silhouette est le coefficient le plus connu combinant les mesures de cohésion et de séparation.

  Le **calcul du coefficient de silhouette** à un point donné comprend les trois étapes suivantes, pour chaque donnée $x \in C_i$,

  1. calcul de la distance moyenne entre x et tous les autres points au sein de la même classe :
  $$a(x) = \frac{1}{|C_i| - 1} \sum_{y \in C_i, x \neq y} d(x, y)$$
  2. calcul de la dissimilarité moyenne minimale entre le point x et une autre classe C_k $(i \neq k)$:
  $$b(x) = \min_{k \neq i} \frac{1}{|C_k|} \sum_{y \in C_k} d(x, y)$$
  3. calcul du coefficient silhouette $s(x) \in [-1, 1]$:
  $$s(x) = \begin{cases} \dfrac{b(x) - a(x)}{\max(a(x), b(x)} & \text{si } |C_i| > 1, \\ 0 \text{ si } |C_i| = 1. \end{cases}$$

Interprétation : le coefficient de silhouette permet de qualifier simplement le partitionnement:

- Des valeurs positives indiquent une séparation élevée entre les clusters.
- Les valeurs négatives indiquent que les classes sont mélangées entre elles (c'est-à-dire qu'il y a chevauchement de classes).
- Lorsque le coefficient de silhouette est nul, c'est une indication que les données sont uniformément réparties dans l'espace euclidien.



Pour évaluer la pertinence de la structure issue du CH, on calculer la distance cophénétique entre deux objets. C'est la hauteur du dendrogramme où les deux branches qui comprennent les deux objets fusionnent en une seule branche.

On peut ainsi calculer le **coefficient de correlation cophénétique** qui mesure la fidélité avec laquelle un dendrogramme préserve les distances par paires entre les points de données originaux non modélisés.

$$c = \frac{\sum_{i<j}(d(x_i, x_j) - \bar{d})(T(x_i, x_j) - \bar{T})}{\sqrt{\sum_{i<j}(d(x_i, x_j) - \bar{d})^2 \sum_{i<j}(T(x_i, x_j) - \bar{T})^2}} \in [-1, 1]$$

avec

- $d(x_i, x_j)$ distace entre les données x_i et x_j
- $T(i, j)$ la distance cophénétique entre les données x_i et x_j
- $\bar{d}$ (resp. $\bar{T}$) est la moyenne des distances (resp. distances cophénétiques).

$\Rightarrow$ Très utilisé en biostatistique pour évaluer des modèles de séquences d'ADN.

Point de vue graphe

Un bon partitionnement d'un graphe implique un nombre d'arêtes intra-communautaires important et un nombre d'arêtes inter-communautaires faible.

La mesure de modularité est décrite comme la proportion des arêtes incidentes sur une classe donnée moins la valeur qu'aurait été cette même proportion si les arêtes étaient disposées au hasard entre les noeuds du graphe.

$\Rightarrow$ Toolbox *igraph* (Python)

La mesure de modularité est décrite comme la proportion des arêtes incidentes sur une classe donnée moins la valeur qu'aurait été cette même proportion si les arêtes étaient disposées au hasard entre les noeuds du graphe.

La modularité d'un partitionnement p (où p_i indique la classe attribuée au noeud i) est définie par :

$$Q = \frac{1}{2m} \sum_{i,j} \left[A_{ij} - \frac{k_i k_j}{2m} \right] \delta(p_i, p_j)$$

où

- A_{ij} est la valeur de la matrice d'adjacence entre les sommets i et j
- k_i est la somme des poids des arêtes adjacentes à i
- m est le nombre d'arêtes du graphe
- δ est le symbole de Kronecker ($\delta(p_i, p_i) = 1$ et $\delta(p_i, p_j) = 0$ si $i \neq j$).

Interprétation de la modularité d'un partitionnement p à maximiser définie par :

$$Q = \frac{1}{2m} \sum_{i,j} \left[A_{ij} - \frac{k_i k_j}{2m} \right] \delta(p_i, p_j) \in [-1, 1]$$

- La modularité est positive pour un graphe et une partition pour lesquels il n'y a aucune arête reliant 2 noeuds de classes différentes mais au moins une arête reliant 2 noeuds de la même classe.
- La modularité est négative pour un graphe et une partition pour lesquels il n'y a aucune arête reliant 2 noeuds de la même classe mais au moins une arête reliant 2 noeuds de classes différentes.
- La modularité tend vers 0 lorsque l'on partitionne aléatoirement un graphe dans lequel les arêtes sont distribuées aléatoirement.

- **Limite de résolution :** si l'on est confronté à des communautés de tailles différentes à l'intérieur d'un même graphe, certaines communautés, même bien définies, pourront ne pas être distinguées dans la partition de modularité optimale.

  ⇒ **Algorithme de Louvain**