Approches à noyau pour l'analyse et l'intégration de données omiques en biologie des systèmes

Jérôme Mariette et Nathalie Vialaneix

MIAT, Université de Toulouse, INRA, 31326 Castanet-Tolosan, France
e-mails: {prenom.nom@inra.fr}

Le développement des techniques de séquençage haut débit génère un volume de données en forte croissance à des coûts relativement faibles. Ces données sont souvent de très grande dimension, hétérogènes et mesurées de manières appariées sur plusieurs niveaux de l'échelle du vivant. Dans le cadre de la biologie des systèmes, de nombreuses méthodes ont été développées pour intégrer ces informations, c'est-à-dire, pour combiner les différentes vues obtenues sur les mêmes échantillons avec de l'information *a priori* et mieux comprendre les mécanismes sous-jacents ou mieux prédire une quantité (souvent un phénotype) d'intérêt. Parmi ces approches, les *noyaux* présentent de nombreux avantages qui en font une approche couramment utilisée pour ces applications. Dans ce chapitre, nous présentons le cadre général des approches à noyau et leur utilité pour l'analyse de divers types de données biologiques. En particulier, nous nous focaliserons sur les approches exploratoires (non supervisées) et l'intégration de données. Nous illustrerons les approches présentées par leur mise en œuvre sur une partie des données du projet *TARA Oceans* (Karsenti et al., 2011; Bork et al., 2015) à l'aide du package R **mixKernel** (Mariette and Villa-Vialaneix, 2018).

Table des matières

1. Introduction

Les avancées des nouvelles techniques de séquençage et la diversification des protocoles permettent aujourd'hui d'étudier un organisme à différentes échelles biologiques. Celles-ci permettent d'étudier le génome, ensemble des gènes d'un organisme, le transcriptome, ensemble des ARNs transcrits, mais aussi l'épigénome, ensemble des mécanismes moléculaires qui modulent l'expression du patrimoine génétique. Les données ainsi produites font partie de la famille dite des omiques dans laquelle on retrouve aussi par exemple les données décrivant le protéome, ensemble des protéines exprimées, ou encore le métabolome, ensemble des métabolites. Les données ainsi produites sont hétérogènes, volumineuse, de grande dimension et sont souvent obtenues sur un petit nombre d'individus ou d'expériences en comparaison avec le nombre de variables mesurées. Souvent les caractéristiques de ces données les rendent mal adaptées aux outils classiques de la statistique. Typiquement, ces données sont des données de comptage pour lesquelles la distance euclidienne usuelle est peu informative, les rendant mal adaptées aux approches utilisant des hypothèses de distribution gaussienne. En outre, ces données sont souvent collectées selon plusieurs types d'expériences de manière appariée, avec l'objectif de combiner l'information apportée par chacun des niveaux de l'échelle du vivant qui est capturé par ces expériences. Il est alors fréquent que cette intégration nécessite de considérer la combinaison de données de types différents (données continues, données de comptage, spectres – ou fonction, réseaux. . .) en particulier lorsqu'il est pertinent d'intégrer dans l'analyse une information biologique additionnelle fournie *a priori* (fonction d'un gène, annotation, structure spatiale d'une protéine, information de régulation entre gènes. . .).

Dans un tel contexte, de nombreuses méthodes intégratives ont été développées afin de combiner ces informations et ainsi considérer un système biologique dans son ensemble. Pour aborder ces questions, les approches à *noyaux* sont couramment utilisées (Schölkopf et al., 2004) car elles offrent un cadre naturel à l'analyse de ces données. En travaillant sur des mesures de similarité entre échantillons (dont le nombre est généralement faible), ces approches sont plus efficaces en calcul et mémoire que les approches fondées sur la représentation classique des données en tableaux individus × variables car elles compressent l'information contenue dans l'ensemble des très nombreuses variables mesurées. En outre, les *noyaux* offrent un cadre mathématique justifié permettant d'étendre beaucoup d'approches statistiques standards de manière naturelle. Enfin, ils sont adaptés à des données de types très variés et fournissent un cadre commun pour la combinaison de ces données, que ce soit à des fins exploratoires (Rappoport and Shamir, 2018) ou prédictives (Lanckriet et al., 2004; Borgwardt et al., 2005).

Dans ce chapitre, nous présentons ces approches et leur utilité pour divers types de données biologiques. En particulier, la section 2 définit ce que l'on appelle noyau et positionne la notion dans le contexte plus large des *données relationnelles*. La section 3 décrit des extensions de méthodes classiques au cadre des noyaux en se focalisant sur les approches exploratoires, non supervisées. On renvoit le lecteur à Ben-Hur et al. (2008) et Vert (2007) pour une description des approches à noyau supervisées en biologie. Ensuite, la section 4 décrit les approches utilisées pour l'intégration de données avec des noyaux dans le cadre de l'analyse exploratoire en décrivant plus spécifiquement sur l'approche proposée dans Mariette and Villa-Vialaneix (2018). Enfin, la section 5 illustre la mise en œuvre de ces méthodes pour l'analyse sur une partie des données du projet *TARA Oceans* (Karsenti et al., 2011; Bork et al., 2015) à l'aide du package R **mixKernel** disponible sur le CRAN [1].

1. https://cran.r-project.org/package=mixKernel

2. Données relationnelles

Dans ce chapitre, on considère que les échantillons d'intérêt (aussi appelés observations) sont notés $(x_i)_{i=1,\dots,n}$ et qu'ils prennent leurs valeurs dans un espace arbitraire, $\mathcal{X}$, qui recouvre tous les exemples discutés en introduction.

2.1. Données décrites par un noyau

On appelle *noyau* une fonction $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ qui peut être évaluée pour toute paire d'observations : $k_{ii'} := K(x_i, x_{i'})$. Cette fonction mesure une similarité entre paire d'observations et elle doit être symétrique ($\forall\, x, x' \in \mathcal{X},\ K(x, x') = K(x', x)$) et positive ($\forall\, N \in \mathbb{N},\ \forall\, \{\alpha_i\}_{i=1,\dots,N} \subset \mathbb{R}$, et $\forall\, \{x_i\}_{i=1,\dots,N} \subset \mathcal{X},\ \sum_{i,i'=1}^{N} \alpha_i \alpha_{i'} K(x_i, x_{i'}) \geq 0$). Elle permet de définir une *matrice noyau*, $\mathbf{K} := (k_{ii'})_{i,i'=1,\dots,n}$, de dimensions $n \times n$ qui est donc symétrique et définie positive.

Cet encart vise à illustrer la diversité des noyaux utilisés, avec un angle particulier sur les données issues de la biologie, en proposant un panel d'exemples utilisant des données de formes diverses (numériques, séquences, graphes...) et d'origines diverses (données transcriptomiques, protéines, métabolites...). Il ne prétend pas à l'exhaustivité.

Exemple 1 (Noyau pour des données numériques) *Un des noyaux les plus fréquemment utilisés, lorsque les données sont des éléments de $\mathcal{X} = \mathbb{R}^p$, est le* noyau gaussien *: $K(x, x') = \exp\left(-\gamma \|x - x'\|^2_{\mathbb{R}^p}\right)$, pour $\gamma > 0$, fixé. Toutefois, celui-ci est mieux adapté à la description de données continues dont l'asymétrie de distribution reste modérée. Pour l'utiliser avec les données de comptage issues des séquençages haut-débit (données RNA-seq, par exemple), il est courant de faire subir aux données un pré-traitement pour transformer les données initiales en données continues avec une variabilité réduite. Les transformations les plus fréquentes sont la transformation logarithmique ou bien le centrage et réduction par gène (Gönen and Margolin, 2014).*

Les noyaux gaussiens (ou d'autres noyaux pour données numériques comme le noyau polynomial ou même le noyau linéaire qui correspond au produit scalaire ordinaire) sont fréquemment utilisés pour traiter des échantillons décrits par des données non numériques. Les applications sont nombreuses comme dans Meher et al. (2016) pour le calcul préalable de descripteurs numériques pour l'identification de sites d'épissage à partir de séquences d'ADN ou bien Qiu et al. (2007) pour la classification de protéines à partir de divers descripteurs numériques.

Exemple 2 (Noyau pour des séquences) *De nombreux noyaux ont été proposés pour calculer des similarités entre des séquences biologiques, c'est-à-dire, des séquences $x_i = (x_{i1}, ..., x_{ip})$ qui prennent leurs valeurs dans $\mathcal{X} = \mathcal{A}^{\otimes p}$ où $\mathcal{A}$ est un alphabet fini. De telles séquences peuvent, par exemple, être une molécule d'ADN (avec $\mathcal{A} = \{A, C, T, G\}$) ou bien des protéines (avec $\mathcal{A}$ l'ensemble des acides aminés). Pour des séquences de protéines, Jaakkola et al. (2000) définissent un noyau basé sur un modèle de chaîne de Markov cachée qui correspond à la famille de protéines d'intérêt et utilisent ensuite le noyau gaussien du score de Fisher entre deux protéines basé sur ce modèle. Leslie et al. (2002, 2004) ont proposé des noyaux alternatifs rapide à calculer qui sont fondés sur les occurrences de k-mers dans la séquence : cette approche est appelée* noyau spectral *(*spectrum kernel*).*
Des approches alternatives pour la définition de noyaux sur les séquences utilisent des méthodes d'alignement ou d'alignement local, comme l'approche générale du noyau de convolution *présentée dans Haussler (1999) qui est étendue pour la définition d'un noyau entre protéines dans Saigo et al. (2004). Une alternative fondée sur le principe de l'alignement est aussi décrite dans Qiu et al. (2007). Ici, l'algorithme MAMMOTH, pour l'alignement structurel de séquences de protéines Oritz et al. (2002), produit un score asymétrique entre paires de séquences. Ce score est ensuite transformé en noyau par l'approche décrite dans Tsuda (1999).*

Exemple 3 (Noyaux pour des données structurées : arbres, graphes) *De nom-*

breuses applications en biologie utilisent des données représentées par des objets structurés comme les arbres ou les graphes : arbres phylogénétiques, arbres de fragmentation issus des spectres obtenues en métabolomique par spectrométrie de masse, graphes de co-expression entre gènes, graphe d'interactions protéine/protéine, représentation de la structure 3D d'une protéine sous la forme de graphe...Ces objets sont également bien adaptés au traitement par approches à noyau.

Les noyaux de comparaison d'arbres utilisent soit des approches fondées sur des comparaisons entre sommets, soit des approches qui tirent partie de la structure de l'arbre en comparant les sous-arbres ou les chemins communs aux deux arbres (voir Shen et al. (2014); Dührkop et al. (2015); Brouard et al. (2016) pour de multiples exemples correspondant à des arbres construits sur des arbres de fragmentation de spectres de masse en métabolomique, voir aussi Vert (2002) qui propose un noyau pour comparer des arbres phylogénétiques fondé sur la combinaison d'un modèle d'évolution probabiliste et d'une similarité entre sous-arbres).

Concernant les graphes, il faut différencier les approches de comparaison des sommets d'un graphe donné, souvent fondées sur des régularisations du Laplacien du graphe (Kondor and Lafferty, 2002; Smola and Kondor, 2003), des approches permettant de comparer une famille de graphes, fondées sur des comparaisons de sous-structures (Ramon and Gärtner, 2003; Mahé and Vert, 2009) ou des comparaisons de chemins dans les graphes (Gärtner et al., 2003; Vishwanathan et al., 2010). Les noyaux de comparaison entre sommets d'un graphe sont utilisés, par exemple, dans Vert and Kanehisa (2003); Rapaport et al. (2007) pour lier expression des gènes et voies métaboliques. Les seconds sont utilisés dans Borgwardt et al. (2005) pour comparer des protéines sur la base de leur structure 3D et dans Mahé and Vert (2009) pour comparer des molécules quelconques sur la base de leur réseau de covalence entre atomes.

Ce formalisme permet de représenter l'espace arbitraire $\mathcal{X}$ comme un espace muni d'une distance et d'un produit scalaire standards. En effet, Aronszajn (1950) montre que, lorsque les conditions de définition du noyau sont remplies, celui-ci définit de manière unique un espace de Hilbert $\mathcal{H}$, appelé espace image et muni du produit scalaire $\langle .,. \rangle_{\mathcal{H}}$, et une fonction de plongement $\phi : \mathcal{X} \to \mathcal{H}$, liés au noyau par la relation :

$$\forall\, x, x' \in \mathcal{X}, \qquad K(x, x') = \langle \phi(x), \phi(x') \rangle_{\mathcal{H}}. \tag{1}$$

En pratique, $\mathcal{H}$ et ϕ ne sont pas explicités mais sont utilisés implicitement au travers du noyau. C'est ce que l'on appelle l'*astuce noyau*. Celle-ci consiste à utiliser le plongement de $\mathcal{X}$ dans $\mathcal{H}$ en exprimant les produits scalaires et distances dans $\mathcal{H}$ à partir des valeurs du noyau. Par exemple, une distance, $d_{\mathcal{H}}(x, x')$, est définie entre deux éléments x et x' dans $\mathcal{X}$, comme la distance entre leur image respective dans $\mathcal{H}$, $\|\phi(x) - \phi(x')\|_{\mathcal{H}}$, qui s'exprime, en utilisation la relation de l'équation (1), par

$$d_{\mathcal{H}}(x, x') = \sqrt{K(x, x) + K(x', x') - 2K(x, x')}. \tag{2}$$

L'utilisation de l'astuce noyau en fouille de données et apprentissage est décrite de manière plus précise dans la section 3.

2.2. Données décrites par une mesure de (dis)similarité générale

Comme décrit précédemment, le cas de données décrites par un noyau offre un cadre formel rigoureux pour l'analyse de données de types très variés. Il est également strictement équivalent au cas de données euclidiennes décrites par leurs distances (calculées dans l'espace image de manière implicite). Toutefois, il ne suffit pas à couvrir l'intégralité des applications des données relationnelles. En effet, comme décrit dans Schleif and Tino (2015), les données relationnelles peuvent être décrites par des mesures de ressemblance ou de dissemblance qui peuvent sortir du cadre euclidien. Nous parlerons alors de *similarité* ou *dissimilarité*, dont la définition formelle n'est pas complètement fixée dans la littérature, mais que nous formaliserons de la manière suivante :

une *dissimilarité* est une mesure de dissemblance, $\delta : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^+$, qui peut être évaluée pour toute paire d'observations : $\forall\, x_i, x_{i'} \in \mathcal{X}$, $\delta_{ii'} := \delta(x_i, x_{i'})$. On suppose, en outre, que

$\delta(x, x) = 0$ pour tout $x \in \mathcal{X}$ et que la fonction est symétrique ($\forall\, x, x' \in \mathcal{X}$, $\delta(x, x') = \delta(x', x)$). On peut aussi définir la matrice de dissimilarité des observations comme $\mathbf{\Delta} := (\delta_{ii'})_{i,i'=1,\dots,n}$, qui est une matrice de dimensions $n \times n$, symétrique, à diagonale nulle et à entrées positives ;

une *similarité* est une mesure de ressemblance, $S : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, qui peut être évaluée pour toute paire d'observations : $\forall\, x_i, x_{i'} \in \mathcal{X}$, $s_{ii'} := S(x_i, x_{i'})$. On suppose, en outre, que cette fonction est symétrique ($\forall\, x, x' \in \mathcal{X}$, $S(x, x') = S(x', x)$) et à diagonale positive ($\forall\, x \in \mathcal{X}$, $S(x, x) \geq 0$). On peut alors définir la matrice de similarité des observations comme $\mathbf{S} := (s_{ii'})_{i,i'=1,\dots,n}$, qui est une matrice de dimensions $n \times n$, symétrique mais pas nécessairement positive.

Les deux définitions précédentes tombent dans le cadre euclidien lorsque :

— la matrice de similarité, $\mathbf{S}$, est définie positive. $\mathbf{S}$ est alors simplement une matrice noyau et le formalisme de la section précédente s'applique ;

— la matrice de dissimilarité, $\mathbf{\Delta}$, est une matrice de distance euclidienne (Schoenberg, 1935; Young and Householder, 1938).

Schleif and Tino (2015) font une revue des approches permettant d'analyser des mesures de similarité non euclidiennes. Celles-ci se séparent, schématiquement, en deux grandes familles : l'une consiste à transformer les données d'une similarité non euclidienne en noyau (par correction du spectre, approches par troncature spectrale ou inversion spectrale (Chen et al., 2009), ou plongement dans un espace euclidien en minimisant la distorsion avec les mesures initiales (Kruskal, 1964)). L'autre utilise les mesures de similarité directement et s'appuie sur un cadre formel appelé *espace pseudo-euclidien* (Goldfarb, 1984). De manière plus précise, si $\mathbf{S}$ est une matrice de similarité quelconque, on peut montrer qu'il existe deux espaces euclidiens uniques, $\mathcal{E}_+$ et $\mathcal{E}_-$, et deux plongements $\phi_+ : \mathcal{X} \to \mathcal{E}_+$ et $\phi_- : \mathcal{X} \to \mathcal{E}_-$ tels que :

$$\forall\, x_i, x_{i'} \in \mathcal{X}, \qquad s_{ij} = \langle \phi_+(x_i), \phi_+(x_{i'}) \rangle_{\mathcal{E}^+} - \langle \phi_-(x_i), \phi_-(x_{i'}) \rangle_{\mathcal{E}^-}.$$

De manière schématique, les approches basées sur ce formalisme utilisent des extensions des méthodes d'analyse statistique qui sont similaires aux extensions s'appuyant sur l'astuce noyau, en travaillant dans les espaces de plongement $\mathcal{E}_+$ et $\mathcal{E}_-$ de manière implicite.

Le cadre de la dissimilarité peut paraître encore plus général. Si on suit l'analogie dissimilarité/distance et similarité/produit scalaire, on peut définir une matrice de dissimilarité à partir d'une matrice de similarité de manière unique en reproduisant l'égalité de l'équation (2) :

$$\delta^2_{ii'} := s_{ii} + s_{i'i'} - 2s_{ii'}, \tag{3}$$

à condition toutefois que le terme de droite de l'égalité soit positif (on dit alors parfois que la matrice de similarité est à diagonale dominante). Réciproquement, la donnée d'une matrice de dissimilarité ne définit pas de manière unique une similarité, même en s'appuyant sur l'analogie précédente, car les valeurs de s_{ii} restent à fixer de manière arbitraire. Toutefois, lorsqu'une dissimilarité δ est donnée, le cadre pseudo-euclidien reste valide avec une égalité du type :

$$\forall\, x_i, x_{i'} \in \mathcal{X}, \qquad \delta^2_{ij} = \|\phi_+(x_i) - \phi_+(x_{i'})\|^2_{\mathcal{E}^+} - \|\phi_-(x_i) - \phi_-(x_{i'})\|^2_{\mathcal{E}^-}.$$

En pratique, dans la suite, nous utiliserons la transformation suivante (Lee and Verleysen, 2007),

$$s_{ii'} := -\frac{1}{2}\left(\delta^2_{ii'} - \frac{1}{n}\sum_{k=1}^{n}\delta^2_{ik} - \frac{1}{n}\sum_{k=1}^{n}\delta^2_{ki'} + \frac{1}{n^2}\sum_{k,k'=1}^{n}\delta^2_{kk'}\right),$$

qui satisfait l'équation (3), pour passer d'une dissimilarité à une similarité. Lorsque la matrice de similarité, $\mathbf{S}$ est définie positive, cette similarité est un noyau qui présente l'avantage d'être centré (c'est-à-dire, que toutes les observations ont une moyenne nulle dans l'espace image), ce qui la rend directement utilisable pour une ACP à noyau par exemple (voir section 3.2).

Exemple 4 (Biodiversité) *L'analyse de la biodiversité conduit à étudier des échantillons qui sont caractérisés par l'abondance d'un ensemble d'espèces ou de taxons. Pour ce faire, différents indices ont été proposés, en particulier pour comparer des échantillons (diversité β). Les premiers indices utilisés, tels que les indices de Jaccard (Jaccard, 1912) et Sørensen (Sørensen, 1948), traitent les espèces rares et abondantes de façon équivalente en comparant uniquement le nombre d'espèces partagées et uniques entre les échantillons. Si $x_i = (x_{i1}, \dots, x_{ip})$ avec x_{ik} le nombre d'observations de l'espèce k dans l'échantillon i, l'indice de Jaccard est défini par*

$$\delta_J(x_i, x_{i'}) = \frac{\sum_{k=1}^{p} \left(\mathbf{1}_{\{x_{ik}>0,\, x_{i'k}=0\}} + \mathbf{1}_{\{x_{i'k}>0,\, x_{ik}=0\}} \right)}{\sum_{k=1}^{p} \mathbf{1}_{\{x_{ik}+x_{i'k}>0\}}}.$$

D'autres indices, comme la dissimilarité de Bray–Curtis, proposée par Bray and Curtis (1957), améliorent ces premiers indices en tenant compte de l'abondance elle-même :

$$\delta_{BC}(x_i, x_{i'}) = \frac{\sum_{k=1}^{p} |x_{ik} - x_{i'k}|}{\sum_{k=1}^{p} (x_{ik} + x_{i'k})}.$$

D'autres approches, comme la distance UniFrac (Lozupone and Knight, 2005; Lozupone et al., 2007), la distance UniFrac pondérée ou bien la distance UniFrac généralisée (Chen et al., 2012) proposent le calcul de distance entre échantillons en tenant compte de la phylogénie des espèces observées ainsi que, parfois, de la rareté de certaines espèces. Le principe global est de pondérer les quantités présentes dans le calcul de l'indice de Jaccard ou bien de la dissimilarité de Bray-Curtis par la longueur des branches d'un arbre phylogénétique entre espèces.

Exemple 5 (Données de séquençage (RNA-seq)) *Witten (2011) montre que la distance euclidienne est mal adaptée aux données issues du séquençage haut débit et développe une dissimilarité basée sur une modélisation par une loi de Poisson en utilisant la statistique du rapport de vraisemblance d'un test de différence entre les moyennes des échantillons comme mesure de l'écart entre les deux échantillons.*

3. Analyse exploratoire pour des données relationnelles

La plupart des méthodes d'analyse statistique, qu'elles soient exploratoires ou prédictives, font l'hypothèse que les données d'entrées sont des données numériques multivariées de $\mathbb{R}^p$ (pour un $p \in \mathbb{N}^*$). L'utilisation de ces outils classiques de la statistique à des données décrites par des relations, comme nous les avons présentées ci-dessous, nécessite donc une adaptation. Lorsque les données à analyser se présentent sous la forme d'un noyau, le principe général de ces adaptations repose sur deux principaux ingrédients :

1. les calculs de produits scalaires et de normes qui sont réalisés au cours de l'algorithme sont remplacés par leur équivalent dans l'espace image, $\mathcal{H}$. L'avantage de ce principe est que le produit scalaire ou la norme dans l'espace image sont connus à partir des seules valeurs du noyau $(k_{ii'})_{i,i'=1,\dots,n}$: ils ne nécessitent pas la définition explicite de l'espace image, qui peut être très complexe. Cette faculté à pouvoir travailler dans un espace image de manière implicite est l'astuce noyau dont nous avons parlé en section 2.1 ;
2. lorsque l'algorithme nécessite la définition de nouvelles observations, qui ne sont pas des éléments de l'ensemble initial des observations (comme par exemple le barycentre d'une classe), celles-ci sont exprimées sous la forme d'une combinaison linéaire ou convexe des images par le plongement ϕ associé au noyau des données initiales : $\sum_{i=1}^{n} \beta_i \phi(x_i)$. Les distances à ces nouveaux individus s'expriment eux aussi exclusivement à partir des valeurs du noyau initial et il n'est donc pas utile de les calculer explicitement (seuls les coefficients β_i sont calculés).

Nous détaillons ci-dessous des extensions particulières de cette approche générale dans le cadre de l'analyse exploratoire, en particulier pour la classification non supervisée, l'ACP et les cartes

auto-organisatrices. La présentation est limitée au cas des noyaux mais elle s'étend de manière similaire au cas de données décrites par des dissimilarités quelconques en utilisant une approche similaire à l'astuce noyau mais fondée sur le contexte pseudo-euclidien. Dans certains cas, nous expliciterons les différences entre les deux approches. La section se termine par une discussion sur les limites des approches à noyau et sur quelques extensions visant à résoudre ces limites.

3.1. Classification non supervisée à noyau

La classification non supervisée est une approche très commune pour l'exploration de données. Son objectif est de regrouper les observations similaires dans des groupes (ou classes) sans *a priori* (contrairement à la classification supervisée qui cherche à apprendre un modèle de prédiction à partir de classes connues pour certains échantillons). Les deux approches les plus simples de classification non supervisée sont la classification ascendante hiérarchique (CAH) et l'algorithme des k-moyennes.

CAH et CAH à noyau Le principe de l'algorithme CAH est fondé sur la création itérative d'une suite de partitions imbriquées. La méthode est initialisée par la définition d'une partition triviale, $\mathcal{P}_1$, où chaque classe de la partition est un singleton $\{x_i\}$ avec $i \in \{1, \dots, n\}$. Puis, par fusions successives de classes deux à deux, l'algorithme produit une hiérarchie de partitions qui se termine avec la partition triviale, $\mathcal{P}_n$, composée d'une unique classe dans laquelle sont regroupés tous les individus $\{x_i\}_{i=1,\dots,n}$. À chaque étape de l'algorithme, les fusions sont choisies de sorte à minimiser une certaine quantité déterminée à l'aide d'un critère de lien qui définit une dissimilarité entre classes (disjointes). Ce critère de lien est généralement fixé directement à partir de mesures de dissimilarité fournies entre les individus et l'algorithme est donc directement adapté à ce type de données. C'est le cas des critères de liens dit de *lien complet* (maximum des dissimilarités entre les observations des deux classes), de *lien simple* (minimum des dissimilarités entre les observations des deux classes) ou de *lien moyen* (moyenne des dissimilarités entre les observations des deux classes).Toutefois, un des critères de lien les plus utilisés, en raison de son interprétation naturelle et de son lien avec l'algorithme des k-moyennes, est défini initialement pour des données de $\mathbb{R}^p$: il s'agit du lien de Ward (Ward, 1963) qui mesure l'augmentation de l'inertie intra-classes induite par la fusion de deux classes :

$$\forall\, C,\, C' \subset \{x_i\}_{i=1,\dots,n}, \qquad L(C, C') = I(C \cup C') - I(C) - I(C'),$$

avec $I(C) = \frac{1}{|C|} \sum_{x_i \in C} \|x_i - \bar{x}_C\|^2_{\mathbb{R}^p}$ avec $\bar{x}_C = \frac{1}{|C|} \sum_{x_i \in C} x_i$. En utilisant les principes généraux décrits plus haut, la CAH s'adapte aux données décrites par un noyau

— en redéfinissant le critère de lien au travers de l'inertie intra-classe dans l'espace image :

$$I(C) = \frac{1}{|C|} \sum_{x_i \in C} \|\phi(x_i) - \bar{x}_C\|^2_{\mathcal{H}},$$

— et en redéfinissant le barycentre des observations dans l'espace image par :

$$\bar{x}_C = \frac{1}{|C|} \sum_{x_i \in C} \phi(x_i).$$

Cette approche, décrite par Qin et al. (2003), a été modifié dans Ambroise et al. (2018) pour fournir une formule explicite simplifiée de la définition du critère de lien $L(C, C')$ qui permet de définir la CAH à noyau comme dans l'algorithme 1.

k-moyennes à noyau Une alternative à l'algorithme CAH se base sur la définition d'un centroïde par classe : c'est l'approche des k-moyennes. Dans sa version initiale, formulée pour des données qui prennent leurs valeurs dans $\mathbb{R}^p$, l'objectif de cette méthode est la définition d'une partition

Algorithme 1 Classification Ascendante Hiérarchique (CAH) à noyau avec lien de Ward

1: **Initialisation** : $\mathcal{P}_1 = \{\{x_1\}, \{x_2\}, \ldots, \{x_n\}\}$
2: **Pour** $t = 1$ à $n-1$ **Faire**
3: Calculer les critères de lien entre tous les clusters de la partition courante $\mathcal{P}_t$:

$$L(C, C') = \frac{|C|\,|C'|}{|C| + |C'|} \left(\frac{1}{|C|^2} \Sigma_{CC} + \frac{1}{|C'|^2} \Sigma_{C'C'} - \frac{2}{|C|\,|C'|} \Sigma_{CC'} \right),$$

avec $\Sigma_{CC'} = \sum_{i \in C, i' \in C'} k_{ii'}$.
4: Fusionner les deux classes pour lesquelles le critère de lien est minimal pour obtenir la partition $\mathcal{P}_{t+1}$
5: **Fin Pour**
6: **Retourner** $\{\mathcal{P}_1, \ldots, \mathcal{P}_n\}$.

qui minimise l'inertie intra-classe pour un nombre P de classes, fixé à l'avance :

$$\underset{C_1, \ldots, C_P}{\operatorname{argmin}} \frac{1}{n} \sum_{j=1}^{P} \sum_{x_i \in C_j} \|x_i - \bar{x}_{C_j}\|^2_{\mathbb{R}^p}.$$

La méthode procède de manière itérative en alternant une phase de calcul des centres de classes, $\bar{x}_{C_j}$, avec une phase d'affectation d'une observation (version stochastique de l'algorithme) ou de toutes les observations (version *batch* de l'algorithme) au centre de classes le plus proche. La configuration initiale est généralement choisie au hasard ou bien dérivée des résultats d'un algorithme CAH (qui permet alors de s'appuyer sur la représentation graphique en *dendrogramme* pour le choix d'un nombre de classes P). L'extension de cette méthode aux données décrites par un noyau est très similaire à l'extension de la CAH. Elle est décrite dans l'algorithme 2.

Algorithme 2 k-moyennes à noyau

1: **Initialisation** : choisir (au hasard) une partition des données en P classes $\mathcal{P}_1 = \{C_1^1, \ldots, C_P^1\}$ $(t = 1)$
2: On note $f^1(x_i)$ la classe (dans $\{1, \ldots, P\}$) de l'observation x_i
3: **Pour** $t = 1$ à T **Faire**
4: Affecter toutes les observations aux classes de $\mathcal{P}_t$ de barycentre le plus proche :
5: ▷ Phase d'affectation

$$f^{t+1}(x_i) = \underset{j=1,\ldots,P}{\operatorname{argmin}} \|\phi(x_i) - \bar{x}_{C_j^t}\|^2_{\mathcal{H}},$$

avec $\bar{x}_{C_j^t} = \frac{1}{|C_j^t|} \sum_{x_l \in C_j^t} \phi(x_l)$ et l'astuce noyau qui permet d'écrire

$$\|\phi(x_i) - \bar{x}_{C_j^t}\|^2_{\mathcal{H}} = k_{ii} - \frac{2}{|C_j^t|} \sum_{x_l \in C_j^t} k_{il} + \frac{1}{|C_j^t|^2} \sum_{x_l,\, x_{l'} \in C_j^t} k_{ll'}.$$

6: Redéfinir les classes à partir de l'affectation précédente ▷ Phase de représentation

$$\mathcal{P}^{t+1} = \{C_1^{t+1}, \ldots, C_P^{t+1}\} \text{ avec } C_j^t = \{x_i : \ f^{t+1}(x_i) = j\}.$$

7: **Fin Pour** ▷ Convergence
8: **Retourner** $\{C_1^{T+1}, \ldots, C_n^{T+1}\}$.

Une autre extension pour des données décrites par des dissimilarités a été proposée par Kaufman and Rousseeuw (1987) sous la forme d'un algorithme des k-médoïdes dans lequel les centroïdes de chaque classe sont remplacés par une observation dans $\mathcal{X}$. Ce type de solution approchée peut être lourd à mettre en œuvre car il nécessite une phase d'optimisation discrète dans l'ensemble d'apprentissage. Une généralisation a été proposée dans Rossi et al. (2007). Celle-ci est fondée sur le cadre théorique d'espace pseudo-euclidien présenté en section 2.2, qui est très proche de la version de l'algorithme 2. En outre, Rossi et al. (2007) proposent une version parcimonieuse de la représentation des centres de classes et un algorithme efficace utilisant des calculs pré-stockés et une mise à jour astucieuse pour traiter de gros volumes de données.

3.2. Analyse en composantes principales à noyau

L'Analyse en Composantes Principales (ACP) est une autre approche très courante en analyse exploratoire des données, dont l'extension aux données décrites par un noyau a été décrite dans Schölkopf et al. (1998). Dans sa forme initiale, l'ACP est une méthode de réduction de dimension qui projette des données, $(x_i)_{i=1,\dots,n}$, de $\mathbb{R}^p$, dans un espace de dimension plus faible, $\mathbb{R}^q$, avec $q < p$ ou $q \ll p$. Pour cela, les variables observées sont combinées linéairement pour définir un nouvel ensemble de variables linéairement décorrélées les unes des autres et appelées composantes principales (ou axes principaux). Ces dernières sont obtenues par décomposition spectrale de la matrice de variance/covariance (empirique) des données initiales et correspondent aux axes permettant de reproduire au mieux l'inertie (c'est-à-dire la variance) de la projection des données en dimension q. La projection des données sur les composantes principales autorise, non seulement, une représentation graphique simplifiée des données, mais fournit aussi une simplification des données par leur représentation dans un espace de dimension plus faible. Dans le cadre de l'ACP, la projection des données est linéaire et le critère minimisé est de type moindre carrés, mais de nombreuses extensions existent, comme les approches de projections non linéaires présentées dans Lee and Verleysen (2007) ou l'utilisation de pertes non quadratiques pour l'ACP comme dans Collins et al. (2001).

L'extension de l'ACP au cadre des données décrites par un noyau est une de ces approches de réduction de dimension non linéaire. Elle correspond exactement à la réalisation d'une ACP dans l'espace image $\mathcal{H}$ associé au noyau et comporte les mêmes étapes :

1. **les données sont centrées dans l'espace image.** Ceci revient à modifier la fonction image associée au noyau, ϕ, en une nouvelle fonction image correspondant aux données centrées et définie par :

$$\tilde{\phi}(x_i) = \phi(x_i) - \frac{1}{n}\sum_{i'=1}^{n}\phi(x_{i'}).$$

   On montre que cette fonction image est associée au noyau centré $\tilde{K}$ de valeurs

$$\tilde{k}_{ii'} = \left\langle \tilde{\phi}(x_i), \tilde{\phi}(x_{i'}) \right\rangle_{\mathcal{H}} = k_{ii'} - \frac{1}{n}\sum_{l=1}^{n}(k_{il} + k_{i'l}) + \frac{1}{n^2}\sum_{l,l'=1}^{n} k_{ll'}.$$

   D'un point de vue matriciel, $\tilde{\mathbf{K}}$ est obtenu par $\tilde{\mathbf{K}} = \mathbf{K} - \frac{1}{n}\mathbf{1}_n\mathbf{K} - \frac{1}{n}\mathbf{K}\mathbf{1}_n + \frac{1}{n^2}\mathbf{1}_n\mathbf{K}\mathbf{1}_n$, dans lequel $\mathbf{1}_n$ est une matrice de dimension $n \times n$ dont les entrées sont toutes égales à 1 ;

2. **la décomposition spectrale du noyau centré, $\tilde{\mathbf{K}}$, est réalisée.** En effet, si l'ACP est généralement présentée sous la forme d'une décomposition spectrale de la matrice de variance / covariance, sa forme duale se base sur la décomposition spectrale de la matrice des produits scalaires entre paires d'échantillons (centrés), c'est-à-dire, dans l'espace image, la matrice dont les entrées sont $\langle\tilde{\phi}(x_i), \tilde{\phi}(x_{i'})\rangle_{\mathcal{H}} = \tilde{k}_{ii'}$. Notons alors $(\beta_j)_{j=1,\dots,n}$ les vecteurs propres de $\tilde{K}$ associés aux valeurs propres $(\lambda_j)_{j=1,\dots,n}$, rangées en ordre décroissant. On peut, sans perte de généralité, supposer que ces vecteurs sont orthogonaux et de normes respectives $1/\sqrt{\lambda_j}$. Ils permettent alors de définir les composantes principales, dans $\mathcal{H}$ sous

la forme

$$a_j = \sum_{i=1}^{n} \beta_{ji} \tilde{\phi}(x_i).$$

On peut alors montrer que ces composantes principales sont orthonormées dans $\mathcal{H}$;

3. **les données sont projetées sur les q premières composantes principales**. La projection de $\tilde{\phi}(x_i)$ sur la composante a_j a alors pour coordonnée $\langle \tilde{\phi}(x_i), a_j \rangle_{\tilde{\mathcal{H}}} = \sum_{i'=1}^{n} \beta_{ji'} \langle \tilde{\phi}(x_i), \tilde{\phi}(x_{i'}) \rangle_{\tilde{\mathcal{H}}} = [\tilde{\mathbf{K}} \beta_j]_i = \lambda_j \beta_{ji}$.

Ces coordonnées sont utiles pour obtenir une représentation des échantillons dans un espace de faible dimension, qui met en valeur leur structuration. Cependant, contrairement à l'ACP standard, l'ACP à noyau ne permet pas de représenter les variables, les échantillons étant décrits par leurs relations, à travers le noyau, et non pas par des valeurs numériques standards. Les composantes principales sont alors plus difficiles à interpréter car elles sont définies par leur similarité à tous les échantillons et pas par des corrélations individuelles avec des variables décrivant les échantillons. De plus, la complexité de la décomposition en valeurs singulières du noyau est de l'ordre de $\mathcal{O}(n^3)$, ce qui fait de l'ACP à noyau une analyse mal adaptée aux jeux de données dans lesquels le nombre d'observations, n, est grand (voir la section 3.4 pour une discussion plus approfondie sur les limites des méthodes à noyau).

L'extension de cette approche au cas de données décrites par des dissimilarités porte le nom de PCoA (*Principal Correspondance Analysis*) ou de MDS (*Multi-Dimensional Scaling*). Elle est basée sur la recherche de coordonnées principales $(a_i)_{i=1,\dots,n}$, qui sont des vecteurs de $\mathbb{R}^d$ $(d \leq n)$ et qui minimisent une fonction dite *de stress*, dont le but est de préserver les distances entre individus dans l'espace de projection :

$$\text{Stress}(a_1, \dots, a_n) = \sum_{i,i'=1}^{n} (\tilde{\delta}^2_{ii'} - a_i^\top a_{i'})^2,$$

où $\tilde{\delta}^2$ correspond au centrage de la dissimilarité $\boldsymbol{\Delta}^2$. Les coordonnées principales sont alors obtenues par décomposition spectrale de $\widetilde{\boldsymbol{\Delta}^2} = (\tilde{\delta}^2_{ii'})_{i,i'=1,\dots,n}$.

3.3. Cartes auto-organisatrices à noyau

Les cartes auto-organisatrices, aussi appelées *cartes de Kohonen* (Kohonen, 2001) ou SOM (*Self-Organizing Maps*), sont une méthode de classification non supervisée permettant d'allier projection des données dans un espace de faible dimension et classification. Initialement inspirées par des principes biologiques, elles font partie de la famille des réseaux de neurones artificiels.

L'algorithme SOM est proche de celui des k-moyennes mais, à la différence de ce dernier, il affecte les observations à des classes qui sont organisées sur une grille munie d'une distance (ou d'une structure topologique) comme illustrée dans la figure 1 (à droite). De manière plus précise, on appelle grille ou carte un ensemble de U classes, souvent appelées unités ou neurones. Celle-ci est munie d'une distance ou d'une relation de voisinage. Pour simplifier, on peut considérer que les neurones sont positionnés dans $\mathbb{R}^2$ selon une grille régulière et que la distance entre les neurones u et u' est la distance usuelle de $\mathbb{R}^2$ entre leur position respective. Dans sa version standard, définie pour des données à valeurs dans $\mathbb{R}^p$, l'algorithme SOM itère deux étapes très similaires aux étapes de l'algorithme des k-moyennes. Ces deux étapes sont basées sur la définition d'un *prototype* de l'unité u, p_u, qui prend ses valeurs dans l'espace des données, $\mathbb{R}^p$, et représente l'unité (donc est proche des observations classées dans cette unité). Les deux étapes de l'algorithme sont alors :

— l'*étape d'affectation* dans laquelle une observation choisie au hasard, x_i, (version stochastique) ou toutes les observations (version *batch*) sont affectées à l'unité dont le prototype est le plus proche de l'observation :

$$f(x_i) := \underset{u=1,\dots,U}{\text{argmin}} \|x_i - p_u\|^2_{\mathbb{R}^p} \ ;$$

— l'*étape de représentation* qui recalcule les valeurs des prototypes pour les mettre en accord avec la nouvelle classification. Dans sa version stochastique, cette étape prend la forme d'une pseudo-descente de gradient stochastique autour de l'observation x_i traitée dans l'étape d'affectation, pour les prototypes des unités voisines de $f(x_i)$:

$$p_u \leftarrow p_u + \mu H(f(x_i), u)(x_i - p_u),$$

où $\mu > 0$ est généralement choisi de manière à décroitre avec les itérations, t, de l'algortihme, par exemple en $1/t$, et H est une fonction décroissante en la distance sur la grille. Elle est généralement choisie constante par morceaux ou de forme gaussienne et son intensité diminue au cours de l'apprentissage (pour généralement être restreinte à la fonction qui vaut 1 si et seulement si $f(x_i) = u$ en fin d'apprentissage, ce qui correspond à des itérations finales semblables à celles d'un algorithme de k-moyennes).

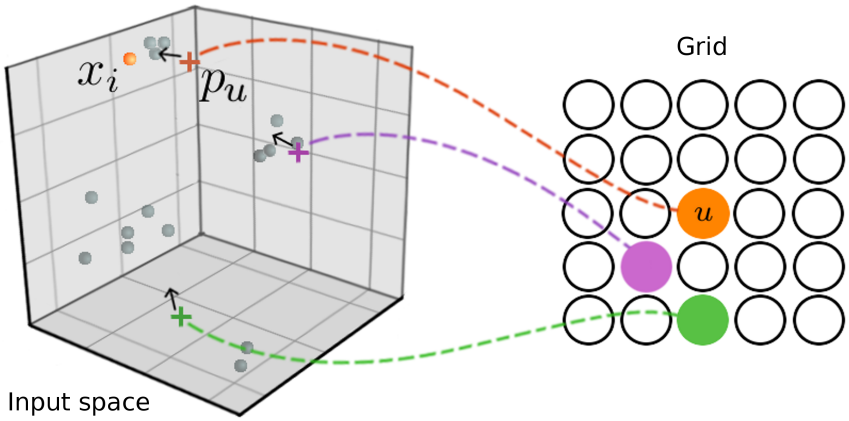

FIGURE 1 – **Algorithme des cartes auto-organisatrices.** Chaque neurone de la grille est associé à un prototype qui le représente dans l'espace d'origine. Une observation x_i est choisie aléatoirement. L'observation x_i est affectée au neurone vainqueur u, *i.e.*, le neurone orange, dont le prototype p_u est le plus proche de x_i dans l'espace de départ. Le prototype vainqueur, p_u, ainsi que les prototypes des neurones voisins sont mis à jour dans la direction de x_i lors de l'étape de représentation.

De nombreux travaux se sont intéressés aux propriétés théoriques de cet algorithme et notamment à sa convergence (voir Cottrell et al. (2016) pour une revue). Dans sa version standard, aucune garantie théorique de convergence n'est prouvée pour l'algorithme SOM, contrairement à celui des k-moyennes qui converge vers un minimum local du critère de variance intra-groupes. Toutefois, l'avantage de l'algorithme SOM est sa capacité à produire une typologie des données très facilement interprétable. La grille permet de visualiser la topologie des observations dans l'espace de départ par l'utilisation d'une projection non linéaire des données initiales. Son adaptation à des données non euclidiennes pose les questions de la définition des prototypes dans l'espace initial (non nécessairement vectoriel) ainsi que du calcul de la distance entre une observation et un prototype (pour l'étape d'affectation de l'algorithme). Pour cela, plusieurs extensions du SOM ont été proposées : le SOM médian (Kohohen and Somervuo, 1998; Kohonen and Somervuo, 2002) et ses variantes (Ambroise and Govaert, 1996; El Golli et al., 2006) choisissent, à l'instar de l'algorithme des k-médoïdes, les prototypes dans l'espace des données observées. Il a toutefois été montré par Rossi (2014) que cette approche pose des problèmes de représentation et d'organisation de la carte produite. D'autres extensions aux données décrites par des noyaux ont été proposées dans Graepel et al. (1998); Mac Donald and Fyfe (2000); Andras (2002); Villa and Rossi (2007) et aux données décrites par des dissimilarités (SOM relationnel) dans Hammer and Hasenfuss (2010); Olteanu and Villa-Vialaneix (2015) pour les versions *batch* et stochastiques de l'algorithme. Les deux principes généraux décrits en introduction de la section 3 s'appliquent ici pour :

— redéfinir la distance au travers de la distance induite par le noyau dans l'espace image ;
— redéfinir la notion de prototypes en la restreignant aux combinaisons linéaires (ou convexes) des images dans l'espace image des données initiales. Ceux-ci s'écrivent alors $p_u = \sum_{i=1}^{n} \beta_{ui}\phi(x_i)$ et l'algorithme consiste alors essentiellement à mettre à jour les $(\beta_{ui})_{i,u}$ sans calculer explicitement les prototypes.

La version stochastique du SOM à noyau est donnée dans l'algorithme 3 et la version relationnelle (adaptée aux dissimilarités quelconques) est très proche de cette version, s'appuyant sur les calculs équivalents dans l'espace pseudo-euclidien défini dans la section 2.

Algorithme 3 SOM à noyau, version stochastique

1: $\forall u = 1, \ldots, U$ et $\forall i = 1, \ldots, n$ initialiser β_{ui}^1 aléatoirement dans $[0, 1]$ tels que $\sum_{i=1}^{n} \beta_{ui}^1 = 1$
2: On note : $p_u^1 = \sum_{i=1}^{n} \beta_{ui}^1 \phi(x_i)$
3: **Pour** $t = 1$ à T **Faire**
4: Tirer un échantillon au hasard : $i \in \{1, \ldots, n\}$ ▷ Étape d'affectation

$$f^{t+1}(x_i) = \underset{u=1,\ldots,U}{\operatorname{argmin}} \|\phi(x_i) - p_u^t\|_{\mathcal{H}}^2$$
$$= \underset{u=1,\ldots,U}{\operatorname{argmin}} \left(k_{ii} - 2\sum_{l=1}^{n} \beta_{ul}^t k_{il} + \sum_{l,l'=1}^{n} \beta_{ul}^t \beta_{ul'}^t k_{ll'} \right)$$

5: Pour tout $u = 1, \ldots, U$, ▷ Étape de représentation

$$p_u^{t+1} = p_u^t + \mu_t H^t(f^{t+1}(x_i), u)(x_i - p_u^t) \quad \Leftrightarrow \quad \beta_u^{t+1} = \beta_u^t + \mu_t H^t(f^{t+1}(x_i), u)\left(\mathbf{1}_i^n - \beta_u^t\right),$$

où $\mathbf{1}_i^n$ est un vecteur de dimension n avec une seule entrée non nulle, l'entrée i, dont la valeur est 1.
6: **Fin Pour**
7: **Retourner** $(p_u^{T+1})_u$ (l'ensemble des prototypes) et $(f^{T+1}(x_i))_i$ (la classification finale des observations sur la carte).

3.4. Limites des approches relationnelles en apprentissage

Les approches relationnelles souffrent de deux désavantages principaux qui peuvent limiter leur utilisation pour l'apprentissage statistique. Nous décrivons ici deux limites fréquemment mises en avant : leur manque d'interprétabilité et leur complexité computationnelle. Des pistes de réponses à ces problèmes sont également décrites.

Manque d'interprétabilité. Si les données initiales étaient décrites par des variables, les valeurs de celles-ci ont été condensées dans le noyau et ne sont plus directement utilisables pour l'interprétation des résultats. En particulier, pour les méthodes basées sur des représentants de classes (centres de gravité pour l'algorithme des k-moyennes ou bien prototypes pour l'algorithme SOM), les représentants n'ont plus une forme explicite facilement interprétable dans l'espace initial mais sont représentés (symboliquement) par leurs proximités avec l'ensemble des observations de l'échantillon sous une forme du type $\sim \sum_{i=1}^{n} \beta_{ui} x_i$. Lorsque n est grand, ce type de représentation les rend très peu utilisables pour interpréter le sens des classes, contrairement au cadre euclidien standard. De même, pour les méthodes similaires à l'ACP à noyau, les axes factoriels n'ont plus de représentation explicite dans l'espace de départ et souffrent donc du même manque d'interprétabilité que les représentants de classes. Pour résoudre cette difficulté, diverses propositions ont été faites.

Description des prototypes et des axes sous des formats plus parcimonieux. Dans cette proposition, le représentant de chaque classe (centre de classe ou prototype) ou bien l'axe factoriel

en ACP est exprimé avec seulement un faible nombre de coefficients β_{ui} non nuls. Cette approche est notamment proposée par Hofmann et al. (2015) qui proposent diverses stratégies pour obtenir des représentations parcimonieuses (seuillage dur, pénalisation $\ell_1 \ldots$) dans un cadre supervisé mais qui peut s'étendre aisément au cadre non supervisé, par Rossi et al. (2007) pour l'algorithme des k-moyennes et par Mariette et al. (2017) pour l'algorithme SOM.

Permutation des valeurs initiales. Lorsque la forme des $\{x_i\}_{i=1,\ldots,n}$ le permet, c'est-à-dire lorsque les observations sont, par exemple, décrites par des variables numériques, des approches par permutation ou par permutation des valeurs initiales permettent de voir l'influence d'une variable sur le résultat final (classification, projection sur les axes d'une ACP, etc) et d'ordonner les variables par importance de leur impact sur le résultat de l'analyse. C'est l'approche développée dans Mariette and Villa-Vialaneix (2018) pour l'ACP à noyau.

Sélection de variables. Également pour le cas où les observations $\{x_i\}_{i=1,\ldots,n}$ sont décrites par p variables numériques, des approches de sélection de variables peuvent être incorporées dans le modèle. Elles consistent généralement à apprendre une pondération $w_j \geq 0$ pour chaque variable $j \in \{1, \ldots, p\}$ où une contrainte de parcimonie (par exemple par une pénalisation ℓ_1) est proposée sur le vecteur $w = (w_1, \ldots, w_p)$. Ce type d'approche a généralement été développé pour le cadre supervisé (voir, par exemple, Allen (2013)) et ne s'étend pas de manière simple au cas non supervisé, car il est fondé sur l'optimisation de la fonction de coût associée à la régression ou à la classification.

Complexité (computationnelle) importante lorsque le nombre d'échantillons, n, croît. En effet, la complexité de l'ACP à noyau est $\mathcal{O}(n^3)$ (complexité de la décomposition spectrale) et la complexité de l'algorithme SOM à noyau (dans sa version stochastique) est $\mathcal{O}(\gamma n^3 U)$ (Rossi, 2014) pour un nombre d'itérations de la forme $T \sim \gamma n$. Outre les approches de description parcimonieuse des prototypes et des axes, présentées plus haut, les principales méthodes permettant d'aborder ce problème sont :

Approximation de Nyström. L'approximation de Nyström (Williams and Seeger, 2000) permet d'obtenir une approximation de la décomposition spectrale d'un noyau K à faible coût. Plus précisément, la décomposition spectrale de K est approchée en sélectionnant (par exemple aléatoirement ou plus efficacement comme proposé dans Kumar et al. (2012)) m observations parmi $\{x_i\}_{i=1,\ldots,n}$, $\mathcal{T}_m$, et en utilisant la décomposition spectrale du noyau réduit $\mathbf{K}^{(m)} = (k_{ii'})_{i,i' \in \mathcal{T}_m}$. Le coût total de l'approche a une complexité dominée par $\mathcal{O}(nm^2)$ et, lorsque le noyau $\mathbf{K}$ est de rang supérieur à m, l'approximation est exacte.

Approches exactes. Rossi et al. (2007); Mariette et al. (2014) proposent des approches exactes pour réduire la complexité des approches à noyau. Ces approches sont basées sur le stockage de calculs intermédiaires et une mise à jour à plus faible coût des prototypes et du calcul des distances. De manière plus précise, pour un coût de stockage de $\mathcal{O}(nU)$, le coût de cette approche, pour l'algorithme SOM, est $\mathcal{O}(\gamma n^2 U)$ pour un nombre d'itérations de l'ordre de $T \sim \gamma n$.

4. Combiner les données relationnelles

4.1. Intégration de données en biologie des systèmes

Le développement des techniques d'acquisition de données, en particulier des approches de séquençage haut débit, ainsi que la mise à disposition publique, de plus en plus fréquente, des données omiques, ont créé des besoins importants pour le développement de méthodes d'*intégration de données* ou d'*approches multi-omiques*. Les articles qui font une revue de ces approches sont nombreux (Noble, 2004; Kristensen et al., 2014; Franzosa et al., 2015; Ritchie et al., 2015; Bersanelli et al., 2016) et ils proposent des typologies de ces approches selon les types d'outils mathématiques utilisés ou bien selon les objectifs de la méthode, par exemple.

L'objectif général est de combiner M types de données collectées sur les mêmes n individus, chacun de ces types de données pouvant éventuellement prendre ses valeurs dans un espace arbitraire non nécessairement numérique et fournissant une image spécifique des données. La combinaison peut avoir un objectif exploratoire, non supervisé (comprendre la structure des échantillons, réaliser une projection des données dans un espace de faible dimension pour les visualiser, effectuer une classification non supervisée ou bien une typologie, comme décrits dans la Section 3) ou bien prédictif, supervisé (apprendre une fonction de prédiction utilisant l'information apportée par les M jeux de données pour prédire une quantité d'intérêt sur les échantillons). Suivant la nomenclature de Noble (2004); Ritchie et al. (2015); Rappoport and Shamir (2018), illustrée dans la figure 2, on peut alors choisir de :

— *concaténer les M tableaux de données en un seul* sur lequel on effectue une analyse unique (exploratoire ou prédictive). Ces approches font généralement l'hypothèse que les différents tableaux de données sont numériques et utilisent cette propriété pour obtenir un résumé pertinent des M informations avant analyse, comme dans Singh et al. (2018) pour l'analyse canonique des corrélations ;

— *réaliser M analyses indépendantes sur chacun des tableaux de données et combiner les résultats.* Ces approches sont généralement regroupées sur le nom d'*approches d'ensembles* et sont particulièrement utilisées pour les analyses supervisées pour lesquelles la combinaison des résultats prend, par exemple, la forme d'une moyenne des prédictions. Dans le cadre non supervisé, la combinaison de plusieurs résultats est moins naturelle et les stratégies sont variées, comme décrit par Vega-Pons and Ruiz-Schulcloper (2011) dans le cadre de la classification non supervisée ;

— *plonger les M tableaux de données dans un espace de représentation commun dans lequel ils sont combinés avant une analyse unique.* Ces approches utilisent généralement des représentations des données sous la forme de graphes ou de noyaux et les combinent sous la forme d'un méta-graphe (Tang et al., 2009) ou d'un méta-noyau. L'avantage de ces approches, sur lesquelles nous allons nous focaliser dans la suite de cette section, est qu'elles permettent de combiner des données hétérogènes, c'est-à-dire, qui n'ont, par exemple, pas toutes une représentation numérique. Également, elles utilisent généralement une représentation des données sous la forme de relations entre échantillons : comme le nombre d'échantillons est souvent petit, en génomique, comparé au nombre de caractères ou de variables les décrivant, l'intégration à ce niveau de représentation est plus rapide et plus facile. Ce type d'intégration peut être considéré comme effectué à un niveau intermédiaire, entre les données et les résultats, car l'intégration est généralement effectuée au niveau des relations entre échantillons vues selon les différents types de données.

4.2. La place des approches à noyaux dans l'intégration de données

Un des avantages des approches à noyau est qu'elles permettent de combiner des sources de données hétérogènes obtenues sur les mêmes individus en proposant un cadre de représentation uniforme. Cette approche, connue sous le nom d'*apprentissage par noyaux multiples* (*multiple kernel learning*), propose de combiner des informations provenant de M noyaux différents, $(K^m)_{m=1,\dots,M}$, évalués sur les mêmes n observations, en un noyau unique

$$K^* = \sum_{m=1}^{M} \gamma_m K^m \qquad \text{tel que} \qquad \begin{cases} \gamma_m \geq 0, \ \forall\, m = 1, \dots, M \\ \sum_{m=1}^{M} \gamma_m = 1 \end{cases} . \tag{4}$$

Par construction, K^* est également symétrique et positif (et donc un noyau valide) et peut donc être utilisé dans des analyses exploratoires ou prédictives comme résumé de l'information intégrée provenant des M noyaux initiaux.

Un choix naturel pour les coefficients $(\gamma_m)_{m=1,\dots,M}$ est de les choisir tous égaux à $1/M$. Toutefois, ce choix considère de manière identique tous les noyaux et ne permet pas de prendre en compte le fait que certains noyaux peuvent être redondants ou bien atypiques, et que, selon le cas, un choix plus pertinent est de leur accorder un poids plus ou moins important. Cette question a été

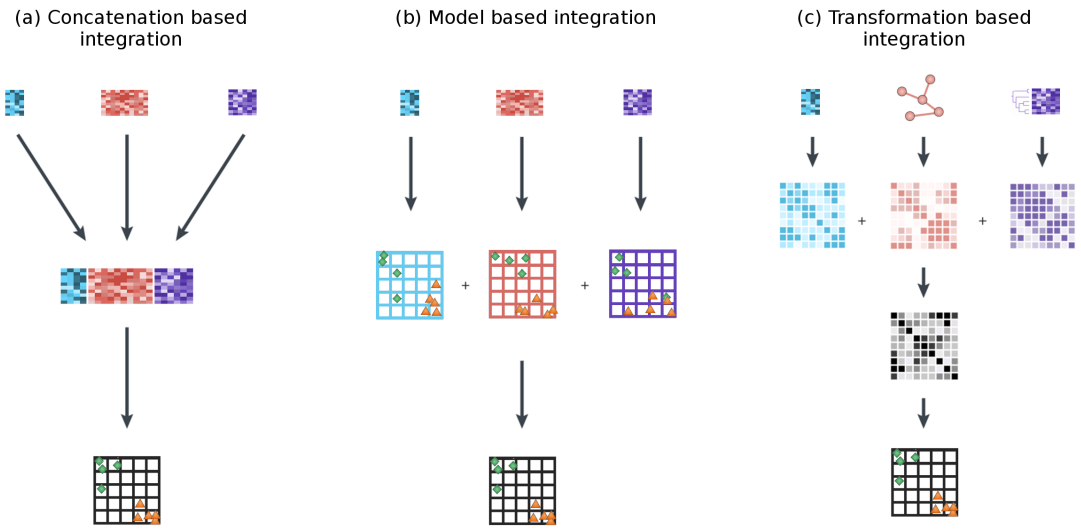

FIGURE 2 – (a) L'intégration par concaténation implique la combinaison des jeux de données au niveau des matrices de données, qui peuvent être les données d'origine ou une représentation numérique de celles-ci. (b) L'intégration par agrégation de résultats de modèles nécessite d'analyser chaque jeu de données indépendamment avant d'agréger les résultats des analyses. (c) L'intégration par transformation projette ou transforme les données d'origine pour pouvoir les combiner. L'objet combiné, pouvant par exemple prendre la forme d'un graphe ou d'un noyau, peut alors être analysé par n'importe quelle méthode conçue pour traiter ce type d'objet. *(Figure inspirée de Ritchie et al. (2015)).*

abordée de manière importante en apprentissage supervisé où une stratégie consiste à choisir les $(\gamma_m)_{m=1,\dots,M}$ de manière à minimiser l'erreur de prédiction (Gönen and Alpaydin, 2011). Dans un cadre non supervisé, un tel objectif n'existe pas. Pour la classification non supervisée, Zhao et al. (2009) proposent de choisir les $(\gamma_m)_{m=1,\dots,M}$ de manière à optimiser la marge entre les différentes classes et (Yu et al., 2012; Gönen and Margolin, 2014; Huang et al., 2012) de manière à minimiser la variance intra-classes entre les differentes classes dans des approches k-moyennes ou *fuzzy* c-moyennes. Toutefois, ces approches sont restreintes au cadre de la classification non supervisée et ne sont pas valables pour d'autres types d'analyses exploratoires comme l'ACP par exemple. Dans le cadre de l'ACP, Speicher and Pfeifer (2017) montrent que l'optimisation directe des valeurs de $(\gamma_m)_{m=1,\dots,M}$ dans le critère de l'ACP conduit à une solution triviale (la sélection d'un seul noyau, celui dont la variance reproduite sur les d axes d'intérêt est la plus grande). Ils proposent donc une approche pénalisant la perte d'inertie induite par la combinaison de noyaux de manière non linéaire sur chaque axe. Cette approche est également restreinte au cadre de l'ACP.

D'une manière générale, les propositions pour combiner des noyaux dans un cadre non supervisé sont beaucoup moins nombreuses et peu d'entre elles abordent la question de la combinaison des noyaux de manière générique, sans recourir à un critère basé sur une méthode exploratoire spécifique (par exemple, classification non supervisée ou approche de réduction de dimension). Zhuang et al. (2011) identifient deux critères principaux permettant une combinaison générique non supervisée de noyaux tous calculés sur des données $(x_i)_{i=1,\dots,n}$ à valeurs dans $\mathbb{R}^p$:

— la minimisation de l'erreur de reconstruction entre le méta-noyau et les données initiales : $\|x_i - \sum_{i'=1}^{n} k^*_{ii'} x_{i'}\|^2_{\mathbb{R}^p}$;
— la préservation, par le méta-noyau, de la structure initiale de distances entre échantillons dans $\mathbb{R}^d$, qui se traduit par la minimisation du critère $\sum_{i'=1}^{n} k^*_{ii'} \|x_i - x_{i'}\|^2_{\mathbb{R}^p}$.

Les approches utilisant ces critères sont fondées sur le fait que les données permettant le calcul des noyaux sont des données de $\mathbb{R}^p$ et elles sont donc essentiellement liées à la représentation des échantillons dans cet espace : ce critère est donc contre-productif si on considère que les noyaux obtenus sur les données initiales sont plus informatifs que la distance euclidienne. Lin et al. (2010); Speicher and Pfeifer (2015) utilisent une approche permettant d'utiliser le critère de préservation de la structure locale des distances sans utiliser explicitement les valeurs $\|x_i - x_{i'}\|^2_{\mathbb{R}^p}$. Pour cela, ils effectuent une projection préalable des données (à partir de données dans $\mathbb{R}^p$ dans leurs exemples) dans un espace de faible dimension (en utilisant, en particulier, des approches de projection qui préservent le voisinage comme celle décrite dans He and Niyogi (2003)) et utilisent le graphe des plus proches voisins dans cet espace de projection pour remplacer les valeurs $\|x_i - x_{i'}\|^2_{\mathbb{R}^p}$ par des similarités basées sur ce graphe. Ainsi, même si le critère n'est pas fondé explicitement sur la distance euclidienne, il en dérive toutefois directement. Outre le fait que cette méthode ne s'applique par lorsque les données originales ne sont pas numériques ou lorsque les M données initiales ne prennent pas leurs valeurs dans un espace commun, le fait d'utiliser la distance euclidienne comme mesure d'une vérité sous-jacente, atténue, de nouveau, la portée du recours à un noyau (différent de la distance usuelle) sur ces données. Enfin, Wang et al. (2017) proposent une approche d'intégration multi-noyaux qui apprend simultanément, par itérations successives, des pondérations pour les différents noyaux et une matrice de similarité globale de faible rang. Cette approche est utilisée pour combiner des noyaux gaussiens de paramètres différents pour la visualisation de données simples cellules RNA-seq et fait l'hypothèse sous-jacente que les données sont structurées en P classes où P est un paramètre à fixer.

Dans Mariette and Villa-Vialaneix (2018), nous proposons trois approches permettant d'aborder la question de la combinaison de noyaux multiples dans un cadre non supervisé et sans hypothèse de structure *a priori*. La première permet d'obtenir un noyau consensuel, qui est le noyau le plus proche, en moyenne, de tous les autres noyaux. La deuxième et la troisième approches utilisent un point de vue différent, similaire à celui de Zhuang et al. (2011), et définissent un noyau qui minimise la distorsion avec l'ensemble des autres noyaux. Ces deux dernières approches diffèrent dans le fait d'utiliser une pénalité ℓ_2 ou ℓ_1 dans le critère optimisé, la pénalité ℓ_1 permettant, en outre, une sélection parmi les M noyaux d'entrée. Ces approches sont implémentées dans le package R **mixKernel** et illustrées dans la section 5.

4.3. Un noyau consensuel

La première proposition, STATIS-UMKL, utilise une approche de type STATIS (L'Hermier des Plantes, 1976; Lavit et al., 1994) pour déterminer un consensus. STATIS est une méthode d'analyse exploratoire des données qui est utilisée pour obtenir une analyse intégrée lorsque les données sont décomposées en blocs multiples. Une matrice consensus est obtenue comme la matrice ayant la similarité moyenne la plus grande avec l'ensemble des blocs, au sens de la norme de Fröbenius. Cette approche s'étend aisément au cas où les blocs correspondent à des noyaux différents.

De manière plus précise, une mesure de similarité entre noyaux est obtenue par calcul de leur cosinus au sens de la norme de Fröbenius : $\forall\, m, m' = 1, \ldots, M$,

$$C_{mm'} = \frac{\langle \mathbf{K}^m, \mathbf{K}^{m'} \rangle_F}{\|\mathbf{K}^m\|_F \|\mathbf{K}^{m'}\|_F} = \frac{\text{Trace}(\mathbf{K}^m \mathbf{K}^{m'})}{\sqrt{\text{Trace}((\mathbf{K}^m)^2)\text{Trace}((\mathbf{K}^{m'})^2)}}. \tag{5}$$

$C_{mm'}$ est une extension du coefficient RV (Robert and Escoufier, 1976) au cadre des noyaux et peut être utilisé pour une analyse exploratoire des relations entre noyaux (repérer les noyaux atypiques ou les groupes de noyaux redondants, identifiés, respectivement, par des valeurs $C_{mm'}$ proches de 0 et de 1).

Ainsi, la matrice $\mathbf{C} = (C_{mm'})_{m,m'=1,\ldots,M}$ contient les informations sur les similitudes entre noyaux et est utilisée pour déterminer un noyau consensus, $\mathbf{K}^*$, qui maximise la similarité moyenne

avec les autres noyaux :

$$\max_{\mathbf{v}} \sum_{m=1}^{M} \left\langle \mathbf{K}^{\mathbf{v}}, \frac{\mathbf{K}^m}{\|\mathbf{K}^m\|_F} \right\rangle_F \tag{6}$$

$$\text{avec} \sum_{m=1}^{M} \left\langle \mathbf{K}^{\mathbf{v}}, \frac{\mathbf{K}^m}{\|\mathbf{K}^m\|_F} \right\rangle_F = \mathbf{v}^\top \mathbf{C}\mathbf{v}, \tag{7}$$

$$\text{pour } \mathbf{K}^{\mathbf{v}} = \sum_{m=1}^{M} v_m \mathbf{K}^m,$$

$$\text{et } \mathbf{v} \in \mathbb{R}^M \text{ tel que } \|\mathbf{v}\|^2_{\mathbb{R}^M} = 1.$$

La solution du problème d'optimisation de l'équation (6) est donnée par la décomposition spectral de $\mathbf{C}$. De manière plus précise, si $\mathbf{v} = (v_m)_{m=1,\dots,M}$ est le premier vecteur propre, de norme 1, de cette décomposition, alors, ses coefficients sont tous positifs (car les matrices $\mathbf{K}^m$ sont définies positives) et il maximise la quantité $\mathbf{v}^\top \mathbf{C}\mathbf{v}$. En définissant $\gamma = \frac{\mathbf{v}}{\sum_{m=1}^{M} v_m}$, on obtient une solution satisfaisant les contraintes de l'équation (4) et qui correspond à un résumé consensuel des M noyaux initiaux.

Enfin, notons que cette méthode est équivalente à effectuer une analyse canonique des corrélations (CCA) multiples entre les M espaces images induits par les M noyaux, comme proposé dans Wang et al. (2008); Ren et al. (2013), respectivement pour un cadre supervisé et pour une ACP à noyau multiple. Toutefois, dans notre approche, seul le premier axe de la CCA est conservé et la contrainte sur la norme ℓ_2 ($\|.\|_{\mathbb{R}^M}$) est utilisée pour obtenir une solution à l'aide d'une simple décomposition spectrale. Cette solution est bien adaptée au cas où le nombre de noyaux est petit.

4.4. Un noyau parcimonieux qui préserve la topologie des données initiales

La proposition précédente vise à obtenir une information consensuelle. Elle a donc tendance à donner plus de poids à des noyaux redondants dans l'ensemble des noyaux et à ne pas tenir compte de l'information amenée par des noyaux atypiques. Il peut cependant être intéressant d'avoir une stratégie inverse, qui pondère de manière privilégiée des informations complémentaires plutôt que des informations redondantes. Dans notre seconde proposition, sparse-UMKL, nous établissons un critère permettant de préserver la géométrie locale des données, de manière à pondérer de manière plus équitable les diverses visions portées par les différents noyaux.

Pour mesurer la géométrie de l'espace initial de chaque noyau, et contrairement à Zhuang et al. (2011), nous utilisons uniquement l'information fournie par les noyaux que nous simplifions sous la forme d'un graphe des k plus proches voisins non orienté (pour une valeur donnée de $k \in \mathbb{N}^*$) au sens de la métrique induite par K^m. On note ce graphe $\mathcal{G}^m$. La matrice d'adjacence d'un graphe combiné, $\mathcal{G}$, est ensuite obtenue en sommant les matrices d'adjacence des graphes $\mathcal{G}^m$. Cette matrice, $\mathbf{W}$, est donc de dimensions $n \times n$ et comptabilise, pour chaque paire d'observations i et i', le nombre de fois où i' est dans les k plus proches voisins de i (et inversement) pour un des M noyaux.

Un critère est finalement défini, permettant de trouver des poids γ_m reproduisant au mieux la topologie telle que représentée par $\mathbf{W}$. Pour ce faire, si ϕ^γ est la fonction image associée au noyau $K^\gamma = \sum_{m=1}^{M} \gamma_m K^m$, on assure que lorsque $W_{ii'}$ est grand, $\phi^\gamma(x_i)$ et $\phi^\gamma(x_{i'})$ sont proches dans l'espace image (et réciproquement). Un critère naturel serait de trouver $\gamma \in \mathbb{R}^M$, tels que $\gamma_m \geq 0$ et $\sum_{m=1}^{M} \gamma_m = 1$, minimisant $\sum_{ii'} W_{ii'} \|\phi^\gamma(x_i) - \phi^\gamma(x_{i'})\|^2_{\mathcal{H}^\gamma}$. Toutefois, de manière similaire à ce qui est observé par Speicher and Pfeifer (2017) pour l'ACP, ce critère revient à minimiser

$$\sum_{m=1}^{M} \gamma_m \underbrace{\sum_{i,i'} W_{ii'}(k^m_{ii} + k^m_{i'i'} - 2k^m_{ii'})}_{=a_m}$$

qui a une solution triviale

$$\gamma_m = \begin{cases} 1 & \text{pour } m^* := \operatorname{argmin}_m a_m \\ 0 & \text{sinon} \end{cases} .$$

Suivant l'idée de Lin et al. (2010), nous proposons de nous restreindre à une représentation de $\phi^\gamma(x_i)$ qui correspond à la similarité de $\phi^\gamma(x_i)$ avec l'ensemble des $(\phi^\gamma(x_{i'}))_{i'=1,\dots,n}$,

$$C_i(\gamma) = \left\langle \phi^\gamma(x_i), \begin{pmatrix} \phi^\gamma(x_1) \\ \vdots \\ \phi^\gamma(x_n) \end{pmatrix} \right\rangle_{\mathcal{H}^\gamma} = \begin{pmatrix} K^\gamma(x_i, x_1) \\ \vdots \\ K^\gamma(x_i, x_n) \end{pmatrix}.$$

Le problème d'optimisation suivant est alors résolu :

$$\min_\gamma \sum_{i,i'=1}^{n} W_{ii} \left\| C_i(\gamma) - C_{i'}(\gamma) \right\|^2_{\mathbb{R}^n} ,$$

$$\text{pour } \gamma \in \mathbb{R}^M \text{ tels que } \gamma_m \geq 0 \text{ et } \sum_{m=1}^{M} \gamma_m = 1.$$

qui est équivalent à :

$$\min_\gamma \sum_{m,m'=1}^{M} \gamma_m \gamma_{m'} S_{mm'}, \tag{8}$$

$$\text{pour } \gamma \in \mathbb{R}^M \text{ tels que } \gamma_m \geq 0 \text{ et } \sum_{m=1}^{M} \gamma_m = 1,$$

avec $S_{mm'} = \sum_{i,i'=1}^{N} W_{ii'} \langle C_i^m - C_{i'}^m, C_i^{m'} - C_{i'}^{m'} \rangle$. La matrice $\mathbf{S} = (S_{mm'})_{m,m'=1,\dots,M}$ est positive et le problème d'optimisation est donc un problème standard de programmation quadratique (QP) avec des contraintes linéaires, qui peut être résolu avec le package R **quadprog**. De plus, comme $\gamma_m \geq 0$, la contrainte $\sum_{m=1}^{M} \gamma_m = 1$ est une contrainte en norme ℓ_1 dans un problème QP et elle réalise donc une sélection parcimonieuse des noyaux (la solution produite aura tendance à ne retenir que certains coefficients γ_m non nuls). Cette propriété, qui peut être vue comme une limitation si on souhaite utiliser l'ensemble des noyaux disponibles, peut être levée en modifiant le critère de l'équation (8) comme décrit dans la section suivante.

4.5. Un noyau complet préservant la topologie des données de départ

Une approche simple (que nous appelons *full*-UMKL) pour lever la propriété de parcimonie de la solution de l'équation (8) consiste à remplacer la contrainte sur la norme ℓ_1 par une contrainte sur la norme ℓ_2, de manière similaire à ce qui est réalisé dans l'équation (6) de STATIS-UMKL :

$$\min_{\mathbf{v}} \sum_{m,m'=1}^{M} v_m v_{m'} S_{mm'}, \tag{9}$$

$$\mathbf{v} \in \mathbb{R}^M \text{ tel que } v_m \geq 0 \text{ et } \|\mathbf{v}\|^2_{\mathbb{R}^M} = 1,$$

puis à définir $\gamma = \frac{\mathbf{v}}{\sum_m v_m}$. Ce problème est un problème de programmation quadratique avec contraintes quadratiques (QCQP) qui est réputé être plus difficile que les problèmes à contraintes linéaires. Nous proposons d'utiliser une résolution par ADMM (*Alterning Direction Method of Multipliers* ; (Boyd et al., 2011)) qui utilise la réécriture du problème initial en :

$$\min_{\mathbf{x} \text{ et } \mathbf{z}} \quad \mathbf{x}^T \mathbf{S} \mathbf{x} + \mathbb{I}_{\{\mathbf{x} \geq 0\}}(\mathbf{x}) + \mathbb{I}_{\{\mathbf{z} \geq 1\}},$$

$$\text{tels que } \mathbf{x} - \mathbf{z} = 0,$$

et permet d'obtenir la solution finale par $\gamma := \frac{\mathbf{z}}{\sum_m z_m}$.

5. Application

L'objectif de ce chapitre est de décrire les différentes étapes permettant d'intégrer et d'analyser les données de Sunagawa et al. (2015) à l'aide des méthodes à noyau non supervisées disponibles dans le package R **mixKernel** (version 0.3). Les informations de session correspondant aux résultats présentés dans cette section sont disponibles en Annexe A.

Les données utilisées ont été collectées dans le cadre de l'expédition *TARA Oceans* (Karsenti et al., 2011; Bork et al., 2015) qui a facilité l'étude des communautés planctoniques en mettant à disposition un ensemble de données de méta-génétique océanique couplé à des mesures environnementales. L'analyse présentée se concentre sur l'étude des 139 échantillons enrichis en procaryotes collectés à partir de 68 stations de prélèvements et sur trois couches océaniques : la surface (SRF), la couche du maximum de chlorophylle (DCM) et la zone mésopélagique (MES). Les 68 stations choisies sont localisées sur 8 océans et mers différents : l'océan Indien (IO), la mer Méditerranée (MS), l'océan Atlantique nord (NAO), l'océan Pacifique nord (NPO), la mer Rouge (RS), l'océan Atlantique sud (SAO), l'océan Pacifique sud (SPO) et l'océan Austral (SO).

Pour simplifier l'analyse et avoir des temps de calcul raisonnables à des fins illustratives, nous utiliserons uniquement un sous-ensemble des données analysées dans Mariette and Villa-Vialaneix (2018). Ce sous-ensemble, disponible dans le package **mixKernel**, inclut 1% des 35 650 unités taxonomiques opérationnelles en procaryotes et des 39 246 gènes bactériens. Ce sous-ensemble a été sélectionné aléatoirement.

Le package **mixKernel** est installé et chargé sous R avec les commandes :

```
install.package(mixKernel)
library(mixKernel)
```

5.1. Chargement des données TARA Ocean

Les jeux de données, préalablement normalisés, sont fournis sous la forme de matrices dont les noms des lignes, représentant les différents échantillons, sont identiques :

```
data(TARAoceans)
# more details with: ?TARAoceans
# we check the dimension of the data:
lapply(list("phychem" = TARAoceans$phychem,
            "pro.phylo" = TARAoceans$pro.phylo,
            "pro.NOGs" = TARAoceans$pro.NOGs),
       dim)
```

```
## $phychem
## [1] 139  22
##
## $pro.phylo
## [1] 139 356
##
## $pro.NOGs
## [1] 139 638
```

5.2. Intégration des données par approches à noyaux

Pour chaque jeu de données, un noyau est calculé en utilisant la fonction `compute.kernel` qui permet de choisir parmi le noyau linéaire, gaussien, Poisson (Witten, 2011), phylogénétique (Lozupone and Knight, 2005; Lozupone et al., 2007) ou d'abondance (Bray and Curtis, 1957; Sørensen, 1948; Bray and Curtis, 1957). L'utilisateur a aussi la possibilité de définir sa propre fonction à l'aide du paramètre `kernel.func` (pour plus d'information `?compute.kernel`).

Les résultats sont retournés sous la forme d'une liste dont l'élément `kernel` stocke la matrice noyau. Les noyaux ainsi obtenus sont des matrices symétriques dont la dimension est égale au nombre d'échantillons, *i.e.*, le nombre de lignes présentes dans le jeu de données d'origine.

```
phychem.kernel <- compute.kernel(TARAoceans$phychem,
                                 kernel.func = "linear")
pro.phylo.kernel <- compute.kernel(TARAoceans$pro.phylo,
                                   kernel.func = "abundance")
pro.NOGs.kernel <- compute.kernel(TARAoceans$pro.NOGs,
                                  kernel.func = "abundance")

# check dimensions
dim(pro.NOGs.kernel$kernel)
```

```
## [1] 139 139
```

Une fois l'ensemble des noyaux calculés, la fonction `cim.kernel` permet d'obtenir une vision globale de la structure des corrélations entre ceux-ci :

```
cim.kernel(phychem = phychem.kernel,
           pro.phylo = pro.phylo.kernel,
           pro.NOGs = pro.NOGs.kernel,
           method = "square")
```

La figure 3 montre que **pro.phylo** et **pro.NOGs** forment la paire de noyaux la plus corrélée. Ce résultat est attendu car ces noyaux fournissent tous deux une vue de la même communauté : la communauté bactérienne.

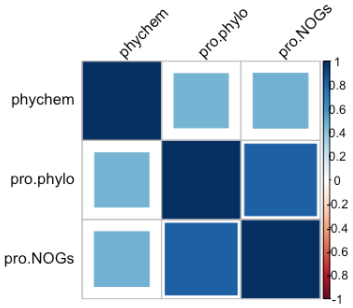

FIGURE 3 – Similarités entre noyaux calculées à l'aide de l'approche STATIS-UMKL. La couleur ainsi que l'aire du carré représentent le niveau de similarité entre deux noyaux.

La fonction `combine.kernels` propose trois méthodes différentes pour combiner plusieurs noyaux : **STATIS-UMKL**, **sparse-UMKL** et **full-UMKL**, toutes trois décrites dans les sections 4.3, 4.4 et 4.5. Cette fonction retourne un méta-noyau qui peut être utilisé en entrée de la fonction `kernel.pca`. Les trois méthodes apportent des informations complémentaires et le choix d'une approche devra se faire en fonction de la question de recherche posée. L'approche **STATIS-UMKL** donne une vue d'ensemble sur l'information commune entre les différents jeux de données. La méthode **full-UMKL** retourne un méta-noyau qui minimise la distorsion entre les différents noyaux fournis en entrée et **sparse-UMKL** est une version parcimonieuse de **full-UMKL** qui sélectionne les noyaux les plus pertinents.

```
meta.kernel <- combine.kernels(phychem = phychem.kernel,
                               pro.phylo = pro.phylo.kernel,
                               pro.NOGs = pro.NOGs.kernel,
                               method = "full-UMKL")
```

5.3. Analyse exploratoire : ACP à noyau

Une ACP à noyau peut alors être obtenue à partir du noyau combiné à l'aide de la fonction `kernel.pca`. Le paramètre `ncomp` permet de choisir le nombre de composantes à extraire de l'ACP à noyau.

```
kpc.res <- kernel.pca(meta.kernel, ncomp = 10)
```

La projection des échantillons sur les axes de l'ACP à noyau peut être affichée à l'aide de la fonction `plotIndiv` (résultat à gauche de la figure 4) :

```
all.depths <- levels(factor(TARAoceans$sample$depth))
depth.pch <- c(20, 17, 4, 3)[match(TARAoceans$sample$depth, all.depths)]
plotIndiv(kpc.res,
          comp = c(1, 2),
          ind.names = FALSE,
          legend = TRUE,
          group = as.vector(TARAoceans$sample$ocean),
          col.per.group = c("#f99943", "#44a7c4", "#05b052", "#2f6395",
                            "#bb5352", "#87c242", "#07080a", "#92bbdb"),
          pch = depth.pch,
          pch.levels = TARAoceans$sample$depth,
          legend.title = "Ocean / Sea",
          title = "Projection of TARA Oceans stations",
          size.title = 10,
          legend.title.pch = "Depth")
```

Ces résultats sont similaires à ceux présentés dans Sunagawa et al. (2015) : les échantillons sont séparés par leur couche océanique d'origine, *i.e.*, SRF, DCM ou MES, avec une forte atypicité des échantillons MES. La variance expliquée par chaque axe de l'ACP à noyau peut être obtenue avec la fonction `plot` (résultat à droite de la figure 4) et peut aider l'utilisateur dans le choix du nombre de composantes à extraire. Ici, le premier axe explique 20% de la variance totale.

```
plot(kpc.res)
```

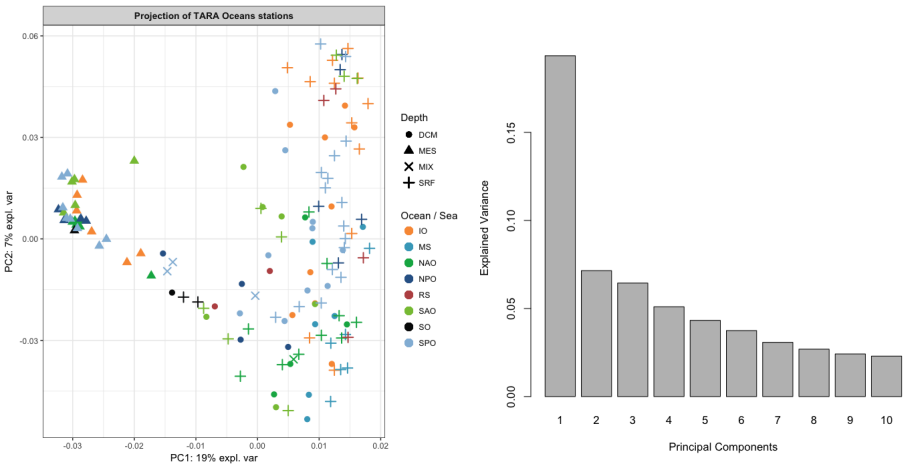

FIGURE 4 – À gauche : Projection des échantillons sur les deux premiers axes de l'ACP à noyau. Les couleurs et les formes représentent respectivement les régions et les couches océaniques. À droite : Variance expliquée par les 10 premiers axes de l'ACP à noyau réalisée sur le méta-noyau obtenu par l'approche **full-UMKL**.

Dans la suite de ce chapitre, nous nous intéressons uniquement à l'information portée par la première composante. Afin d'étudier l'influence des variables des différents jeux de données, leurs

valeurs sont permutées aléatoirement à l'aide de la fonction `permute.kernel.pca`. La figure 5 présente le résultat de la permutation des variables physico-chimiques au niveau des variables elles-mêmes (noyau `phychem`), des abondances en unité taxonomique opérationnelle (OTU) du noyau `pro.phylo` au niveau du phylum (les phyla des OTUs sont stockés dans la seconde colonne, nommée `Phylum`, de l'annotation taxonomique disponible par l'entrée `taxonomy` de l'objet `TARAoceans`) et des abondances en gènes du noyau `pro.NOGs` au niveau des GOs (les GOs sont disponibles à partir de l'entrée `GO` du jeu de données) :

```
head(TARAoceans$taxonomy[ ,"Phylum"], 10)
```

```
##  [1] Actinobacteria   Proteobacteria   Proteobacteria
##  [4] Gemmatimonadetes Actinobacteria   Actinobacteria
##  [7] Proteobacteria   Proteobacteria   Proteobacteria
##  ...
## 56 Levels: Acidobacteria Actinobacteria aquifer1 ... WCHB1-60
```

```
head(TARAoceans$GO, 10)
```

```
##  [1] NA  NA  "K" NA  NA  "S" "S" "S" NA  "S"
```

```
# here we set a seed for reproducible results with this tutorial
set.seed(17051753)
kpc.res <- kernel.pca.permute(kpc.res, ncomp = 1,
                              phychem = colnames(TARAoceans$phychem),
                              pro.phylo = TARAoceans$taxonomy[ ,"Phylum"],
                              pro.NOGs = TARAoceans$GO)
```

Les résultats, présentés dans la figure 5, peuvent être visualisés à l'aide de la fonction `plotVar.kernel.pca`. Le paramètre `ndisplay` permet de définir le nombre de variables à afficher pour chaque noyau :

```
plotVar.kernel.pca(kpc.res, ndisplay = 10, ncol = 3)
```

Proteobacteria est la variable la plus importante du noyau `pro.phylo`. Pour visualiser les valeurs d'abondance relative des *Proteobacteria*, celles-ci sont extraites pour colorer chacun des 139 échantillons, comme présenté à gauche de la figure 6 :

```
selected <- which(TARAoceans$taxonomy[ ,"Phylum"] == "Proteobacteria")
proteobac.sample <- apply(TARAoceans$pro.phylo[ ,selected], 1, sum)
proteobac.sample <- proteobac.sample / apply(TARAoceans$pro.phylo, 1, sum)
colfunc <- colorRampPalette(c("royalblue", "red"))
col.proteo <- colfunc(length(proteobac.sample))
col.proteo <- col.proteo[rank(proteobac.sample, ties = "first")]
plotIndiv(kpc.res,
          comp = c(1, 2),
          ind.names = FALSE,
          legend = FALSE,
          group = c(1:139),
          col = col.proteo,
          pch = depth.pch,
          pch.levels = TARAoceans$sample$depth,
          legend.title = "Ocean / Sea",
          title = "Representation of Proteobacteria abundance",
          legend.title.pch = "Depth")
```

De la même façon, la température est la variable la plus importante de noyau **phychem**. Les valeurs de température peuvent alors être affichées sur la figure représentant la projection de l'ACP à noyau (à droite de la figure 6) :

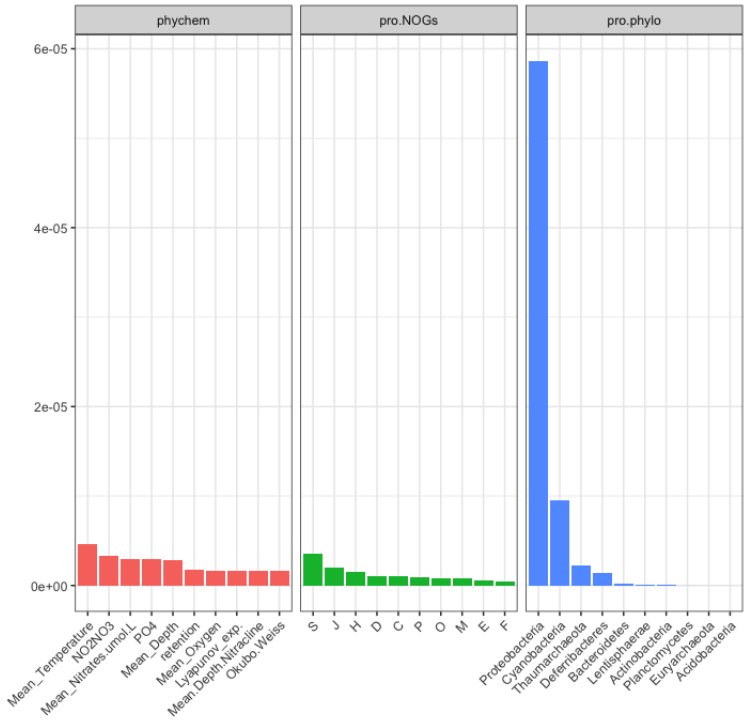

FIGURE 5 – Les cinq variables les plus importantes pour chacun des trois jeux de données, rangées par importance décroissante.

```
col.temp <- colfunc(length(TARAoceans$phychem[ ,4]))
col.temp <- col.temp[rank(TARAoceans$phychem[ ,4], ties = "first")]
plotIndiv(kpc.res,
          comp = c(1, 2),
          ind.names = FALSE,
          legend = FALSE,
          group = c(1:139),
          col = col.temp,
          pch = depth.pch,
          pch.levels = TARAoceans$sample$depth,
          legend.title = "Ocean / Sea",
          title = "Representation of mean temperature",
          legend.title.pch = "Depth")
```

Pour les deux graphiques de la figure 6, les gradients de couleurs observés sur le premier axe de l'ACP à noyau entre la gauche (faibles abondances en *Proteobacteria* et faibles températures) et la droite (fortes abondances en *Proteobacteria* et fortes températures) confirment la contribution des variables *Proteobacteria* et température à la définition du premier axe. Ces variables structurent de manière principale ce sous-échantillon de données (ce qui était déjà observé dans l'étude préliminaire de Mariette and Villa-Vialaneix (2018) sur l'ensemble des échantillons et est concordant avec les résultats discutés dans Sunagawa et al. (2015)). Ces variables séparent les échantillons selon leur profondeur, particulièrement les échantillons de la zone mésopélagique des autres échantillons (la plus profonde et la plus froide, en triangle sur la figure). Le deuxième axe montre une association inversée entre échantillons à température forte (en haut) qui correspondent à des échantillons dans lesquels l'abondance des *Proteobacteria* est plus modérée. D'une manière intégrée, l'analyse a permis l'extraction des variables les plus structurantes de la variabi-

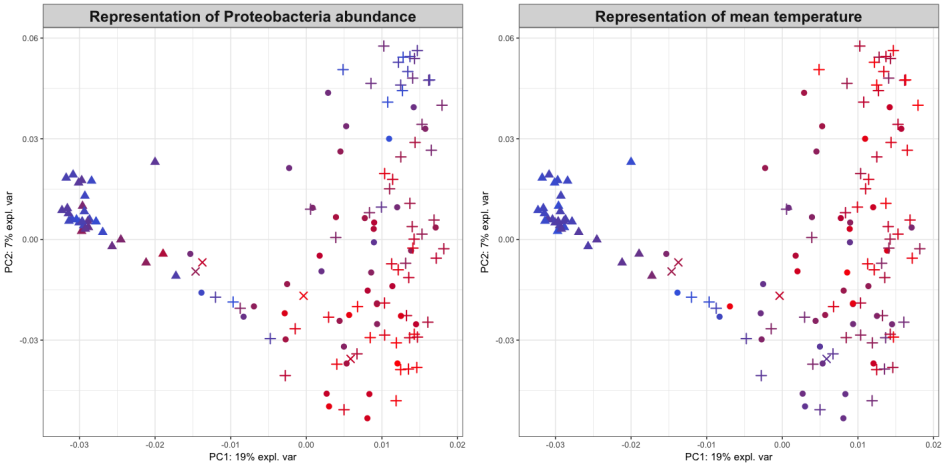

FIGURE 6 – Projection des échantillons sur les deux premiers axes de l'ACP à noyau. À gauche : Les couleurs représentent l'abondance relative en *Proteobacteria*. À droite : Les couleurs représentent la température (bleu pour les eaux froides et rouge pour les eaux chaudes).

lité entre les échantillons et une représentation permettant l'association entre ces variables et les caractéristiques des échantillons.

Références

Allen, G. I. (2013). Automatic feature selection via weighted kernels and regularization. *Journal of Computational and Graphical Statistics*, 22(2) :284–299.

Ambroise, C., Dehman, A., Koskas, M., Neuvial, P., Rigaill, G., and Vialaneix, N. (2018). Adjacency-constrained hierarchical clustering of a band similarity matrix with application to genomics. Preprint.

Ambroise, C. and Govaert, G. (1996). Analyzing dissimilarity matrices via Kohonen maps. In *Proceedings of 5th Conference of the International Federation of Classification Societies (IFCS 1996)*, volume 2, pages 96–99, Kobe (Japan).

Andras, P. (2002). Kernel-Kohonen networks. *International Journal of Neural Systems*, 12 :117–135.

Aronszajn, N. (1950). Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68(3) :337–404.

Ben-Hur, A., Ong, C. S., Sonnenburg, S., Schölkopf, B., and Rätsch, G. (2008). Support vector machines and kernels for computational biology. *PLoS Computational Biology*, 4(10) :e1000173.

Bersanelli, M., Mosca, E., Remondini, D., Giampieri, E., Sala, C., Castellani, G., and Milanesi, L. (2016). Methods for the integration of multi-omics data : mathematical aspects. *BMC Bioinformatics*, 17(Suppl 2) :S15.

Borgwardt, K. M., Ong, C. S., Schönauer, S., Vishwanathan, S., Smola, A. J., and Kriegel, H.-P. (2005). Protein function prediction via graph kernels. *Bioinformatics*, 2005(21) :i47–i56.

Bork, P., Bowler, C., de Vargas, C., Gorsky, G., Karsenti, E., and Wincker, P. (2015). Tara oceans studies plankton at planetary scale. *Science*, 348(6237) :873–873.

Boyd, S., Parikh, N., Chu, E., Peleato, B., and Eckstein, J. (2011). Distributed optimization and statistical learning via the alterning direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1) :1–122.

Bray, R. J. and Curtis, J. (1957). An ordination of the upland forest communities of Southern Wisconsin. *Ecological Monographs*, 27(4) :325–349.

Brouard, C., Shen, H., Dürkop, K., d'Alché Buc, F., Böcker, S., and Rousu, J. (2016). Fast metabolite identification with input output kernel regression. *Bioinformatics*, 32(12) :i28–i36.

Chen, J., Bittinger, K., Charlson, E., Hoffmann, C., Lewis, J., Wu, G., Collman, R., Bushman, F., and Li, H. (2012). Associating microbiome composition with environmental covariates using generalized unifrac distances. *Bioinformatics*, 28(16) :2106–2113.

Chen, Y., Garcia, E., Gupta, M., Rahimi, A., and Cazzanti, L. (2009). Similarity-based classification : concepts and algorithm. *Journal of Machine Learning Research*, 10 :747–776.

Collins, M., Dasgupta, S., and Schapire, R. E. (2001). A generalization of principal component analysis to the exponential family. In Dietterich, T., Becker, S., and Ghahramani, Z., editors, *Advances in Neural Information Processing Systems (NIPS)*, volume 13, pages 617–624, Cambridge, MA. MIT Press.

Cottrell, M., Olteanu, M., Rossi, F., and Villa-Vialaneix, N. (2016). Theoretical and applied aspects of the self-organizing maps. In Merényi, E., Mendenhall, M., and P., O., editors, *Advances in Self-Organizing Maps and Learning Vector Quantization (Proceedings of WSOM 2016)*, volume 428 of *Advances in Intelligent Systems and Computing*, pages 3–26, Houston, TX, USA. Springer International Publishing Switzerland.

Dührkop, K., Shen, H., Meusel, M., Rousu, J., and Böcker, S. (2015). Searching molecular structure databases with tandem mass spectra using CSI :FingerID. *Proceedings of the National Academy of Sciences of the United States of America*, 112(41) :12580–12585.

El Golli, A., Rossi, F., Conan-Guez, B., and Lechevallier, Y. (2006). Une adaptation des cartes auto-organisatrices pour des données décrites par un tableau de dissimilarités. *Revue de Statistique Appliquée*, LIV(3) :33–64.

Franzosa, E. A., Hsu, T., Sirota-madi, A., Shafquat, A., Abu-Ali, G., Morgan, X. C., and Huttenhower, C. (2015). Sequencing and beyond : integrating molecular 'omics' for microbial community profiling. *Nature Reviews Microbiology*, 13(6) :360–372.

Gärtner, T., Flach, A., and Wrobel, S. (2003). On graph kernels : Hardness a results and efficient alternatives. In *Proceedings of the Annual Conference on Computational Learning Theory*, pages 129–143.

Goldfarb, L. (1984). A unified approach to pattern recognition. *Pattern Recognition*, 17(5) :575–582.

Gönen, M. and Alpaydin, E. (2011). Multiple kernel learning algorithms. *Journal of Machine Learning Research*, 12 :2211–2268.

Gönen, M. and Margolin, A. A. (2014). Localized data fusion for kernel k-means clustering with application to cancer biology. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., and Weinberger, K., editors, *Proceedings of Advances in Neural Information Processing Systems 27 (NIPS 2014)*, volume 27, pages 1305–1313. Curran Associates, Inc.

Graepel, T., Burger, M., and Obermayer, K. (1998). Self-organizing maps : generalizations and new optimization techniques. *Neurocomputing*, 21 :173–190.

Hammer, B. and Hasenfuss, A. (2010). Topographic mapping of large dissimilarity data sets. *Neural Computation*, 22(9) :2229–2284.

Haussler, D. (1999). Convolution kernels on discrete structures. Technical Report, UCS-CRL-99-10.

He, X. and Niyogi, P. (2003). Locality preserving projections. *Proceedings of Advances in Neural Information Processing Systems*, 4.

Hofmann, D., Gisbrecht, A., and Hammer, B. (2015). Efficient approximations of robust soft learning vector quantization for non-vectorial data. *Neurocomputing*, 147 :96–106.

Huang, H.-C., Chuang, Y.-Y., and Chen, C.-S. (2012). Multiple kernel fuzzy clustering. *IEEE Transactions on Fuzzy Systems*, 20(1) :120–134.

Jaakkola, T., Diekhans, M., and Haussler, D. (2000). A discriminative framework for detecting remote protein homologies. *Journal of Computational Biology*, 7(1-2) :95–114.

Jaccard, P. (1912). The distribution of the flora in the alpine zone. *New Phytologist*, 11(2) :37–50.

Karsenti, E., Acinas, S., Bork, P., Bowler, C., de Vargas, C., Raes, J., Sullivan, M., Arendt, D., Benzoni, F., Claverie, J., Follows, M., Gorsky, G., Hingamp, P., Iudicone, D., Jaillon, O., Kandels-Lewis, S., Krzic, U., Not, F., Ogata, H., Pesant, S., Reynaud, E., Sardet, C., Sieracki, M., Speich, S., Velayoudon, D., Weissenbach, J., Wincker, P., and Tara Oceans Consortium (2011). A holistic approach to marine eco-systems biology. *PLoS Biology*, 9(10) :e1001177.

Kaufman, L. and Rousseeuw, P. (1987). Clustering by means of medoids. In Dodge, Y., editor, *Statistical Data Analysis Based on the L1-Norm and Related Methods*, pages 405–416. North-Holland.

Kohohen, T. and Somervuo, P. (1998). Self-organizing maps of symbol strings. *Neurocomputing*, 21 :19–30.

Kohonen, T. (2001). *Self-Organizing Maps, 3rd Edition*, volume 30. Springer, Berlin, Heidelberg, New York.

Kohonen, T. and Somervuo, P. (2002). How to make large self-organizing maps for nonvectorial data. *Neural Networks*, 15(8) :945–952.

Kondor, R. and Lafferty, J. (2002). Diffusion kernels on graphs and other discrete structures. In Sammut, C. and Hoffmann, A., editors, *Proceedings of the 19th International Conference on Machine Learning*, pages 315–322, Sydney, Australia. Morgan Kaufmann Publishers Inc. San Francisco, CA, USA.

Kristensen, V. N., Lingjærde, O. C., Russnes, H. G., Vollan, H. K. M., Frigessi, A., and Børresen-Dale, A.-L. (2014). Principles and methods of integrative genomic analyses in cancer. *Nature Reviews Cancer*, 14(5) :299–313.

Kruskal, Joseph, B. (1964). Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29(1) :1–27.

Kumar, S., Mohri, M., and Talwalkar, A. (2012). Sampling techniques for the Nyström method. *Journal of Machine Learning Research*, 13 :981–1006.

Lanckriet, G., Cristianini, N., Bartlett, P., El Ghaoui, L., and Jordan, M. (2004). Learning the kernel matrix with semidefinite programming. *Journal of Machine Learning Research*, 5 :27–72.

Lavit, C., Escoufier, Y., Sabatier, R., and Traissac, P. (1994). The ACT (STATIS method). *Computational Statistics and Data Analysis*, 18(1) :97–119.

Lee, J. and Verleysen, M. (2007). *Nonlinear Dimensionality Reduction*. Information Science and Statistics. Springer, New York ; London.

Leslie, C., Eskin, E., and Noble, W. S. (2002). The spectrum kernel : a string kernel for SVM protein classification. In Altman, R. B., Dunker, A. K., Hunter, L., Lauderdale, K., and Klein, T. E., editors, *Proceedings of the Pacific Symposium on Biocomputing (Biocomputing 2002)*. World Scientific, Connecting Great Minds.

Leslie, C. S., Eskin, E., Cohen, A., and Noble, W. S. (2004). Mismatch string kernels for discriminative protein classification. *Bioinformatics*, 20(4) :467–476.

L'Hermier des Plantes, H. (1976). *Structuration des tableaux à trois indices de la statistique*. PhD thesis, Université de Montpellier. Thèse de troisième cycle.

Lin, Y., Liu, T., and CS., F. (2010). Multiple kernel learning for dimensionality reduction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33 :1147–1160.

Lozupone, C. and Knight, R. (2005). UniFrac : a new phylogenetic method for comparing microbial communities. *Applied and Environmental Microbiology*, 71(12) :8228–8235.

Lozupone, C. A., Hamady, M., Kelley, S. T., and Knight, R. (2007). Quantitative and qualitative β eiversity measures lead to different insights into factors that structure microbial communities. *Applied and Environmental Microbiology*, 73(5) :1576–1585.

Mac Donald, D. and Fyfe, C. (2000). The kernel self organising map. In *Proceedings of 4th International Conference on knowledge-based Intelligence Engineering Systems and Applied Technologies*, pages 317–320.

Mahé, P. and Vert, J. (2009). Graph kernels based on tree patterns for molecules. *Machine Learning*, 75 :3–35.

Mariette, J., Olteanu, M., Boelaert, J., and Villa-Vialaneix, N. (2014). Bagged kernel SOM. In Villmann, T., Schleif, F., Kaden, M., and Lange, M., editors, *Advances in Self-Organizing Maps and Learning Vector Quantization (Proceedings of WSOM 2014)*, volume 295 of *Advances in Intelligent Systems and Computing*, pages 45–54, Mittweida, Germany. Springer Verlag, Berlin, Heidelberg.

Mariette, J., Olteanu, M., and Villa-Vialaneix, N. (2017). Efficient interpretable variants of online SOM for large dissimilarity data. *Neurocomputing*, 225 :31–48.

Mariette, J. and Villa-Vialaneix, N. (2018). Unsupervised multiple kernel learning for heterogeneous data integration. *Bioinformatics*, 34(6) :1009–1015.

Meher, P. K., Sahu, T. K., Rao, A., and Wahi, S. (2016). Identification of donor splice sites using support vector machine : a computational approach based on positional, compositional and dependency features. *Algorithms for Molecular Biology*, 11(16) :16.

Noble, W. S. (2004). *Kernel Methods in Computational Biology*, chapter Support vector machine applications in computational biology, pages 71–92. MIT Press.

Olteanu, M. and Villa-Vialaneix, N. (2015). On-line relational and multiple relational SOM. *Neurocomputing*, 147 :15–30.

Oritz, A. R., Strauss, C. E., and Olmea, O. (2002). MAMMOTH (matching molecular models obtained from theory) : an automated method for model comparison. *Protein Science*, 11(11) :2606–2621.

Qin, J., Lewis, D. P., and Noble, W. S. (2003). Kernel hierarchical gene clustering from microarray expression data. *Bioinformatics*, 19(16) :2097–2104.

Qiu, J., Hue, M., Ben-Hur, A., Vert, J.-P., and Stafford, N. W. (2007). A structural alignment kernel for protein structures. *Bioinformatics*, 23(9) :1090–1098.

Ramon, J. and Gärtner, T. (2003). Expressivity versus efficiency of graph kernels. In Washio, T. and de Raedt, L., editors, *Proceedings of First International Workshop on Mining Graphs, Trees and Sequences (held with ECML/PKDD'03)*, pages 65–74.

Rapaport, F., Zinovyev, A., Dutreix, M., Barillot, E., and Vert, J. (2007). Classification of microarray data using gene networks. *BMC Bioinformatics*, 8 :35.

Rappoport, N. and Shamir, R. (2018). Multi-omic and multi-view clustering algorithms : review and cancer benchmark. *Nucleic Acids Research*, 46(20) :10546–10562.

Ren, S., Ling, P., Yang, M., Ni, Y., and Zong, Z. (2013). Multi-kernel PCA with discriminant manifold for hoist monitoring. *Journal of Applied Sciences*, 13(20) :4195–4200.

Ritchie, M. D., Holzinger, E. R., Li, R., Pendergrass, S. A., and Kim, D. (2015). Methods of integrating data to uncover genotype-phenotype interactions. *Nature Reviews Genetics*, 16(2) :85–97.

Robert, P. and Escoufier, Y. (1976). A unifying tool for linear multivariate statistical methods : the rv-coefficient. *Applied Statistics*, 25(3) :257–265.

Rossi, F. (2014). How many dissimilarity/kernel self organizing map variants do we need ? In Villmann, T., Schleif, F., Kaden, M., and Lange, M., editors, *Advances in Self-Organizing Maps and Learning Vector Quantization (Proceedings of WSOM 2014)*, volume 295 of *Advances in Intelligent Systems and Computing*, pages 3–23, Mittweida, Germany. Springer Verlag, Berlin, Heidelberg.

Rossi, F., Hasenfuss, A., and Hammer, B. (2007). Accelerating relational clustering algorithms with sparse prototype representation. In *Proceedings of the 6th Workshop on Self-Organizing Maps (WSOM 07)*, Bielefield, Germany. Neuroinformatics Group, Bielefield University.

Saigo, H., Vert, J.-P., Ueda, N., and Akutsu, T. (2004). Protein homology detection using string alignment kernels. *Bioinformatics*, 20(11) :1682–1689.

Schleif, F.-M. and Tino, P. (2015). Indefinite proximity learning : a review. *Neural Computation*, 27(10) :2039–2096.

Schoenberg, I. (1935). Remarks to Maurice Fréchet's article "Sur la définition axiomatique d'une classe d'espace distanciés vectoriellement applicable sur l'espace de Hilbert". *Annals of Mathematics*, 36 :724–732.

Schölkopf, B., Smola, A., and Müller, K. (1998). Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5) :1299–1319.

Schölkopf, B., Tsuda, K., and Vert, J. (2004). *Kernel Methods in Computational Biology*. MIT Press, London, UK.

Shen, H., Dührkop, K., Böcher, S., and Rousu, J. (2014). Metabolite identification through multiple kernel learning on fragmentation trees. *Bioinformatics*, 30(12) :i157–i64.

Singh, A., Gautier, B., Shannon, C. P., Rohart, F., Vacher, M., Tebbut, S. J., and Lê Cao, K.-A. (2018). DIABLO : from multi-omics assays to biomarker discovery, an integrative approach. Preprint bioRxiv.

Smola, A. and Kondor, R. (2003). Kernels and regularization on graphs. In Warmuth, M. and Schölkopf, B., editors, *Proceedings of the Conference on Learning Theory (COLT) and Kernel Workshop*, Lecture Notes in Computer Science, pages 144–158, Washington, DC, USA. Springer-Verlag Berlin Heidelberg.

Sørensen, T. J. (1948). A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to analyses of the vegetation on Danish commons. *Biologiske Skrifter/Kongelige Danske Videnskabernes Selskab*, 5(4) :1–34.

Speicher, N. K. and Pfeifer, N. (2015). Integrating different data types by regularized unsupervised multiple kernel learning with application to cancer subtype discovery. *Bioinformatics*, 31(12) :i268–i275.

Speicher, N. K. and Pfeifer, N. (2017). Towards multiple kernel principal component analysis for integrative analysis of tumor samples. *Journal of Integrative Bioinformatics*, 14(2) :20170019.

Sunagawa, S., Coelho, L., Chaffron, S., Kultima, J., Labadie, K., Salazar, F., Djahanschiri, B., Zeller, G., Mende, D., Alberti, A., Cornejo-Castillo, F., Costea, P., Cruaud, C., d'Oviedo, F., Engelen, S., Ferrera, I., Gasol, J., Guidi, L., Hildebrand, F., Kokoszka, F., Lepoivre, C., Lima-Mendez, G., Poulain, J., Poulos, B., Royo-Llonch, M., Sarmento, H., Vieira-Silva, S., Dimier, C., Picheral, M., Searson, S., Kandels-Lewis, S., *Tara* Oceans coordinators, Bowler, C., de Vargas, C., Gorsky, G., Grimsley, N., Hingamp, P., Iudicone, D., Jaillon, O., Not, F., Ogata, H., Pesant, S., Speich, S., Stemmann, L., Sullivan, M., Weissenbach, J., Wincker, P., Karsenti, E., Raes, J., Acinas, S., and Bork, P. (2015). Structure and function of the global ocean microbiome. *Science*, 348(6237).

Tang, W., Lu, Z., and Dhillon, I. (2009). Clustering with multiple graphs. In Wang, W., Kargupta, H., Ranka, S., Yu, P. S., and Wu, X., editors, *Proceedings of IEEE International Conference on Data Mining (ICDM)*, pages 1016–1021, Miami, Florida, USA. IEEE Computer Society.

Tsuda, K. (1999). Support vector classifier with asymmetric kernel functions. In Verleysen, M., editor, *Proceedings of the 7th European Symposium on Artificial Neural Network (ESANN 1999)*, pages 183–188, Bruges, Belgium. D-Facto public.

Vega-Pons, S. and Ruiz-Schulcloper, J. (2011). A survey of clustering ensemble algorithms. *International Journal of Pattern Recognition and Artificial Intelligence*.

Vert, J. and Kanehisa, M. (2003). Extracting active pathways from gene expression data. *Bioinformatics*, 19(Suppl. 2) :ii238–ii244.

Vert, J.-P. (2002). A tree kernel to analyse phylogenetic profiles. *Bioinformatics*, 18(Suppl. 1) :S276–S284.

Vert, J.-P. (2007). *Kernel Methods in Bioengineering, Signal and Image Processing*, chapter Kernel methods in genomics and computational biology, pages 42–63. Idea Group.

Villa, N. and Rossi, F. (2007). A comparison between dissimilarity SOM and kernel SOM for clustering the vertices of a graph. In *6th International Workshop on Self-Organizing Maps (WSOM 2007)*, Bielefeld, Germany. Neuroinformatics Group, Bielefeld University.

Vishwanathan, S., Schraudolph, N. N., Kondor, R., and Borgwardt, Karsten, M. (2010). Graph kernels. *Journal of Machine Learning Research*, 11 :1201–1242.

Wang, B., Zhu, J., Pierson, E., Ramazzotti, D., and Batzoglou, S. (2017). Visualization and analysis of single-cell RNA-seq data by kernel-based similarity learning. *Nature Methods*, 14 :414–416.

Wang, Z., Chen, S., and Sun, T. (2008). MultiK-MHKS : a novel multiple kernel learning algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2) :348–353.

Ward, J. H. (1963). Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, 53(301) :236–244.

Williams, C. and Seeger, M. (2000). Using the Nyström method to speed up kernel machines. In Leen, T., Dietterich, T., and Tresp, V., editors, *Advances in Neural Information Processing Systems (Proceedings of NIPS 2000)*, volume 13, Denver, CO, USA. Neural Information Processing Systems Foundation.

Witten, D. (2011). Classification and clustering of sequencing data using a Poisson model. *The Annals of Applied Statistics*, 5(4) :2493–2518.

Young, G. and Householder, A. (1938). Discussion of a set of points in terms of their mutual distances. *Psychometrika*, 3 :19–22.

Yu, S., Tranchevent, L., Liu, X., Glanzel, W., Suykens, J. A., de Moor, B., and Moreau, Y. (2012). Optimized data fusion for kernel k-means clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(5) :1031–1039.

Zhao, B., Kwok, J., and Zhang, C. (2009). Multiple kernel clustering. In Apte, C., Park, H., Wang, K., and Zaki, M., editors, *Proceedings of the 2009 SIAM International Conference on Data Mining (SDM)*, pages 638–649, Philadelphia, PA. SIAM.

Zhuang, J., Wang, J., Hoi, S., and Lan, X. (2011). Unsupervised multiple kernel clustering. *Journal of Machine Learning Research : Workshop and Conference Proceedings*, 20 :129–144.

A. Information de session pour les résultats de la section 5

```
sessionInfo()
```

```
## R version 3.4.3 (2017-11-30)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 16.04.5 LTS
##
## Matrix products: default
## BLAS: /usr/local/lib/R/lib/libRblas.so
## LAPACK: /usr/local/lib/R/lib/libRlapack.so
##
## locale:
##  [1] LC_CTYPE=fr_FR.UTF-8       LC_NUMERIC=C
##  [3] LC_TIME=fr_FR.UTF-8        LC_COLLATE=fr_FR.UTF-8
##  [5] LC_MONETARY=fr_FR.UTF-8    LC_MESSAGES=fr_FR.UTF-8
##  [7] LC_PAPER=fr_FR.UTF-8       LC_NAME=C
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C
## [11] LC_MEASUREMENT=fr_FR.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] mixKernel_0.3   mixOmics_6.3.2  ggplot2_2.2.1   lattice_0.20-35
## [5] MASS_7.3-50     knitr_1.20
##
## loaded via a namespace (and not attached):
##  [1] Biobase_2.38.0         tidyr_0.8.1            splines_3.4.3
##  [4] jsonlite_1.5           foreach_1.4.4          ellipse_0.4.1
##  [7] shiny_1.1.0            assertthat_0.2.0       stats4_3.4.3
## [10] phyloseq_1.22.3        yaml_2.1.19            corrplot_0.84
## [13] pillar_1.2.3           backports_1.1.2        quadprog_1.5-5
## [16] glue_1.2.0             digest_0.6.15          manipulateWidget_
##      0.9.0
```

```
## [19] RColorBrewer_1.1-2     promises_1.0.1         XVector_0.18.0
## [22] colorspace_1.3-2       psych_1.8.4            htmltools_0.3.6
## [25] httpuv_1.4.3           Matrix_1.2-14          plyr_1.8.4
## [28] pkgconfig_2.0.2        zlibbioc_1.24.0        purrr_0.2.5
## [31] xtable_1.8-2           corpcor_1.6.9          scales_0.5.0
## [34] RSpectra_0.13-1        later_0.7.2            tibble_1.4.2
## [37] mgcv_1.8-23            IRanges_2.12.0         BiocGenerics_0.24.0
## [40] lazyeval_0.2.1         mnormt_1.5-5           survival_2.42-3
## [43] magrittr_1.5           mime_0.5               evaluate_0.10.1
## [46] nlme_3.1-137           foreign_0.8-70         vegan_2.5-2
## [49] data.table_1.11.4      tools_3.4.3            matrixStats_0.53.1
## [52] stringr_1.3.1          S4Vectors_0.16.0       munsell_0.4.3
## [55] cluster_2.0.7-1        bindrcpp_0.2.2         Biostrings_2.46.0
## [58] ade4_1.7-11            compiler_3.4.3         rlang_0.2.1
## [61] rhdf5_2.22.0           grid_3.4.3             iterators_1.0.9
## [64] biomformat_1.6.0       htmlwidgets_1.2        crosstalk_1.0.0
## [67] igraph_1.2.2           miniUI_0.1.1.1         labeling_0.3
## [70] rmarkdown_1.9          multtest_2.34.0        gtable_0.2.0
## [73] codetools_0.2-15       rARPACK_0.11-0         reshape2_1.4.3
## [76] R6_2.2.2               gridExtra_2.3          dplyr_0.7.6
## [79] bindr_0.1.1            rprojroot_1.3-2        LDRTools_0.2-1
## [82] permute_0.9-4          ape_5.2                stringi_1.2.2
## [85] parallel_3.4.3         Rcpp_1.0.0             rgl_0.99.16
## [88] tidyselect_0.2.4
```