Exploration des méthodes d'analyse de données compositionnelles pour l'étude du microbiome

Clément Hardy
encadré par Mahendra Mariadassou (MaIAGE), Magali Berland (MetaGenoPolis)
dans le cadre du stage de M1 du Master "Mathématiques Appliquées"

1er mai 2018 - 31 août 2018

INRA, Domaine de Vilvert, Jouy-En-Josas

Table des matières

Table des figures

1 Présentation

J'ai réalisé mon stage à l'Inra de Jouy en Josas dans l'unité "Mathématiques et Informatique Appliquées du Génome à l'Environnement" (MaIAGE) [1].

1.1 Unité de recherche

L'unité de recherche MaIAGE regroupe des mathématiciens, des informaticiens, des bioinformaticiens et des biologistes autour de questions de biologie et agro-écologie, allant de l'échelle moléculaire à l'échelle du paysage en passant par l'étude de l'individu, de populations ou d'écosystèmes.

L'unité développe des méthodes mathématiques et informatiques originales de portée générique ou motivées par des problèmes biologiques précis. Elle s'implique aussi dans la mise à disposition de bases de données et de logiciels permettant aux biologistes d'utiliser les outils dans de bonnes conditions ou d'exploiter automatiquement la littérature scientifique.

L'inférence statistique et la modélisation dynamique sont des compétences fortes de l'unité, auxquelles s'ajoutent la bioinformatique, l'automatique et l'algorithmique. Les activités de recherche et d'ingénierie s'appuient également sur une forte implication dans les disciplines destinatrices : écologie, environnement, biologie moléculaire et biologie des systèmes.

L'unité MaIAGE est structurée en 4 équipes de recherche :

1. bioinformatique et statistique pour les données "omiques"(StatInfOmics)
2. biologie des Systèmes(BioSys)
3. modélisation dynamique et statistique pour les écosystèmes, l'épidémiologie et l'agronomie(Dynenvie)
4. acquisition et formalisation de connaissances à partir de texte(Bibliome)

J'ai été accueilli au sein de l'équipe **StatInfOmics**.

1.2 Equipe

L'équipe StatInfOmics vise à développer et mettre en oeuvre des méthodes statistiques et bioinformatiques dédiées à l'analyse de données "omiques". D'un point de vue biologique, les questions abordées concernent principalement l'annotation structurale et fonctionnelle des génomes, les régulations géniques, la dynamique évolutive des génomes, et la caractérisation d'écosystèmes microbiens en terme de diversité et de fonctions présentes ; une cible commune étant la relation entre génotype et phénotype. Une part de plus en plus importante de notre activité est relative à l'intégration de données "omiques" hétérogènes pour en extraire de l'information pertinente et aussi prédire des processus biologiques. D'un point de vue méthodologique, nos travaux sont essentiellement d'ordre statistique : estimation de distributions, inférence de modèles à variables latentes, prédiction de relations entre jeux de variables, segmentation, visualisation et classification, avec une attention particulière au cadre de la grande dimension qui caractérise la majorité des jeux de données "omiques" étudiés. Ces recherches s'appuient souvent sur une ingénierie bioinformatique très forte.
Mon travail de stage a porté sur les méthodes d'analyse statistique utilisées pour étudier les écosystèmes microbiens.

1. `http://maiage.jouy.inra.fr/`

2 Contexte biologique et enjeux

2.1 Métagénomique

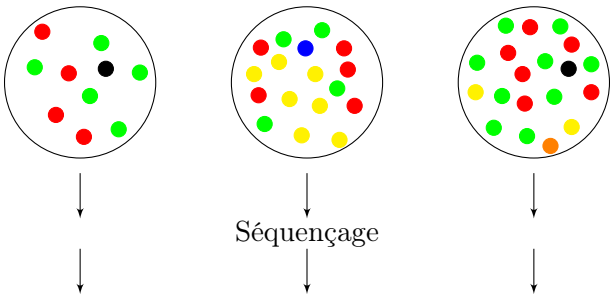
La métagénomique désigne l'étude du contenu génomique d'un échantillon issu d'un milieu complexe, c'est à dire qui abrite de nombreuses espèces, tel que le microbiote intestinal, l'eau de mer, le sol, etc. La métagénomique, à l'inverse de la génomique, analyse simultanément le génome de l'ensemble des espèces présentes dans un échantillon et ne nécessite donc pas d'isoler une à une les espèces présentes. Cette évolution de l'étude d'une espèce à l'étude d'un ensemble d'espèces s'est faite grâce aux nouvelles méthodes de séquencage, dites à haut-débit, et aux progrés de la bioinformatique pour le traitement des données de séquençage.

Deux approches complémentaires existent : la métagénomique "shotgun" et la métagéniomique "amplicon". L'approche "shotgun" consiste à séquencer la totalité du génome des espèces présentes. Pour cela, une première étape de préparation des échantillons consiste à briser la paroi des cellules bactériennes pour en extraire l'ADN avant de découper ces séquences d'ADN en petit fragments de longueur prédéfinie (typiquement quelques centaines de paires de base). Les fragments ainsi obtenus sont ensuite séquencés individuellement pour obtenir des "lectures" (ou "reads"). L'étape qui permet de repasser des séquences courtes aux génomes complets est appelée assemblage et est réalisée à l'aide de logiciels spécialisés. Il est également possible d'assigner directement une espèce bactérienne à chaque lecture, sur la base de sa similarité avec des séquences d'espèces connues.

L'approche "amplicon", également appelée "metabarcoding", est basée sur le séquençage un gène-marqueur, en général la sous unité 16S du ribosome (ou ARN 16S), qui va servir de "code-barre" pour identifier les espèces présentes. Le processus de séquençage n'est pas parfait et introduit des erreurs de lecture. Une première étape dans le traitement des données consiste donc à regrouper les lectures similaires en clusters pour construire des organismes hypothétiques, les OTU (Operational Taxonomic Units), qui sont considérées comme des pseudo-espèces bactériennes. Une assignation taxonomique est ensuite attribuée à chaque OTU en comparant sa séquence consensus à des séquences de références dont la taxonomie est connue.

Les deux approches varient dans leurs finalités : l'approche "shotgun" s'intéresse aux génomes complets et donc aux gènes et aux fontions présentes dans l'écosystème ("qui peut faire quoi ?") tandis que l'approche "amplicon" permet de faire un inventaire taxonomique des espèces en présence ("qui est là ?"). Les gènes de l'approche "shotgun" peuvent néanmoins être aggrégés sur la base de leur co-abondance pour construire des MetaGenomic Species (MGS)[6] qui s'apparente à des OTU. Dans la suite de ce rapport et par souci de simplicité, on utilisera le terme OTU dans tout les cas.

Le résultat typique d'une expérience de métagénomique est une table de comptage : un tableau contenant le nombre de séquences observées pour chaque OTU dans chaque échantillon (Fig. 1).



MGS/OTU	échantillon 1	échantillon 2	échantillon 3
●	4	5	6
●	0	7	2
●	5	4	9
●	1	0	1
●	0	1	0
●	0	0	1

FIGURE 1 – Table comptage des OTU

L'une des particularités de la table des OTU pour la métagénomique est qu'elle contient énormément de zéros. En effet, un OTU qui est présent dans un seul échantillon apparaîtra avec un comptage nul dans les autres échantillons. Les OTU présents dans chaque échantillon pouvant être très différents, il y a ainsi énormément de zéros (sur les jeux de données utilisés durant mon stage la proportion de zéros dans la table des OTU variait de 35% à 95%).

2.2 Problématique des données de comptage

L'une des caractéristiques clé des données issues de la métagénomique est la différence du nombre de lectures obtenues par échantillon. En effet, les séquenceurs hauts débits offrent la possibilité de séquencer un grand nombre d'échantillons en même temps, mais ne garantissent pas que le nombre de séquences (profondeur de séquençage) obtenues sera le même pour tous les échantillons. Le nombre de lectures obtenues suite au séquençage dans un échantillon peut être par exemple de 1000 tandis que dans un autre il ne peut être que de 500. De plus, les comptages obtenus ne sont pas directement proportionnels aux abondances réelles dans les échantillons. Ainsi, deux échantillons peuvent avoir la même profondeur de séquençage et le même vecteur de comptage, mais des biomasses et donc des nombres de cellules bactériennes, très différentes. La comparaison directe des comptages entre échantillons est donc impossible. Le passage aux données compositionnelles est une méthode de résolution possible, et numériquement peu coûteuse, de ces deux problèmes. Les données compositionnelles sont néanmoins contraintes (l'ensemble des proportions somme à 1) et nécessitent donc des méthodes d'analyse adaptées.

2.3 Objectif du stage

Le but de ce stage a été de transformer ces données de comptage en données compositionnelles avant de leur appliquer des méthodes d'analyses adéquates (notamment les transformations log-ratio). Ces transformations permettent de projeter les données dans un espace euclidien usuel et (i) d'y appliquer les méthodes d'analyse multivariée classique (réduction de dimension, clustering,...) pour identifier des structures biologiques d'intérêt et (ii) d'en apprendre la densité des données dans cet espace, à l'aide d'un modèle probabiliste, pour construire des simulateurs de données qui reproduisent les caractéristiques des données réelles.

3 Données Compositionelles et Géométrie d'Aitchison

Commençons par expliquer ce que sont les données compositionnelles.

3.1 Simplexe

Un vecteur $x = [x_1, x_2, ...x_D]$ est défini comme une D-composition lorsque toutes ses composantes sont strictement positives et ne contiennent que de *l'information relative*, par exemple un pourcentage, une proportion, ou des parties par million (notamment en géologie).

Le simplexe est l'espace de probabilité des données compositionnelles et est défini par :

$$S^D = \{x = [x_1, x_2, ..., x_D] \,|x_i > 0, i = 1, 2, ..., D; \sum_{i=1}^{D} x_i = \kappa\}$$

L'opération permettant de modifier la somme des composantes d'une composition (ex : passer d'une proportion à un pourcentage) se nomme la closure. En notant κ la somme des composantes souhaitée, la closure est définie par :

$$C(\kappa, x) = \left[\frac{\kappa * x_1}{\sum_{i=1}^{D} x_i}, \frac{\kappa \times x_2}{\sum_{i=1}^{D} x_i}, ..., \frac{\kappa \times x_D}{\sum_{i=1}^{D} x_i}\right]$$

Dans la suite du rapport, sauf mention explicite du contraire, la notation $C(x)$ correspondra à $C(1, x)$. Deux vecteurs $x, y \in \mathbb{R}_+^D$ tels que $x_i,\ y_i > 0,\ \forall i = 1, ..., D$ sont dits compositionnellement équivalents s'il existe un $\lambda \in \mathbb{R}^+$ tel que $x = \lambda y$ ce qui est équivalent à $C(x) = C(y)$.

La représentation graphique d'une 3-composition se fait grâce à un diagramme ternaire. Ce dernier est formé d'un triangle équilatéral. La représentation graphique de la 3-composition $X = [x_1, x_2, x_3]$ est telle que la distance entre X et le côté opposé au sommet i est exactement x_i (Fig. 2). De façon équivalent en, X est le barycentre des sommets du triangle, pondérés par les poids x_1, x_2, x_3.

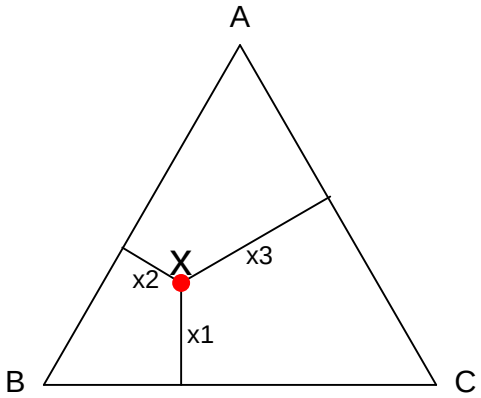


FIGURE 2 – Exemple de diagramme ternaire. X est le barycentre des sommets A, B, C pondérés par les poids x_1, x_2, x_3

3.2 Opérations dans le simplexe

Dans le simplexe, nous ne pouvons pas utiliser la géométrie euclidienne pour comparer des compositions. Remarquons cela en prenant en exemple les quatre 3-compositions suivantes :

$$x_1 = [0.1, 0.4, 0.5]\,,\ x_2 = [0.2, 0.3, 0.5]\,,\ x_3 = [0.4, 0.4, 0.2]\,,\ x_4 = [0.5, 0.3, 0.2]$$

La distance **euclidienne** entre les compositions x_1 et x_2 est la même que celle entre x_3 et x_4, pourtant la proportion de la première composante a doublé dans le premier cas tandis qu'elle n'a augmenté que de 25% dans le second. De même, la multiplication et l'addition usuelles ne sont pas appropriées dans le simplexe : la multiplication d'une composition par un scalaire ne permet pas de rester dans le simplexe (elle change la valeur de κ) et la somme de deux compositions multiplie elle aussi κ par deux. Pour ces raisons, nous avons besoin de définir une nouvelle géométrie dans le simplexe [2].

Commençons par définir les opérations de bases :

$\oplus$: Perturbation d'une composition $x \in S^D$ par une composition $y \in S^D$,

$$x \oplus y = C\left([x_1 y_1, x_2 y_2, ..., x_D y_D]\right)$$

$\odot$: Puissance d'une composition $x \in S^D$ par une constante $\alpha \in \mathbb{R}$,

$$x \in S^D,\ \alpha \in \mathbb{R},\ \alpha \odot x = C\left([x_1^\alpha, x_2^\alpha, ..., x_D^\alpha]\right)$$

Les deux opérations ci-dessus permettent au simplexe de devenir un espace vectoriel (la perturbation $\oplus$ en tant que loi de composition interne, la puissance $\odot$ en tant que loi de composition externe). À ces deux opérations, on ajoute un produit scalaire ainsi que la distance et la norme associées à ce produit scalaire :

Produit scalaire de deux compositions $x, y \in S^D$,

$$x, y \in S^D,\ \langle x, y \rangle_a = \frac{1}{2D} \sum_{i=1}^{D} \sum_{j=1}^{D} ln\frac{x_i}{x_j} ln\frac{y_i}{y_j}$$

Distance entre x et $y \in S^D$,

$$x, y \in S^D,\ d_a(x, y) = \sqrt{\frac{1}{2D} \sum_{i=1}^{D} \sum_{j=1}^{D} \left(ln\frac{x_i}{x_j} - ln\frac{y_i}{y_j} \right)^2}$$

Norme d'une composition $x \in S^D$,

$$\|x\|_a = \sqrt{\frac{1}{2D} \sum_{i=1}^{D} \sum_{j=1}^{D} \left(ln\frac{x_i}{x_j} \right)^2}$$

Il devient ainsi possible de définir des objets géométriques telles que les lignes compositionnelles de direction x et passant par x_0 (droite dans l'espace euclidien) : $y = x_0 \oplus (\alpha \odot x)$, avec $x, x_0 \in S^D$ et $\alpha \in \mathbb{R}$. Ce qui donne graphiquement pour D=3,

2. Des raisons supplémentaires, telles que la construction de régions de confiance pour les compositions aléatoires, sont évoquées dans [7]

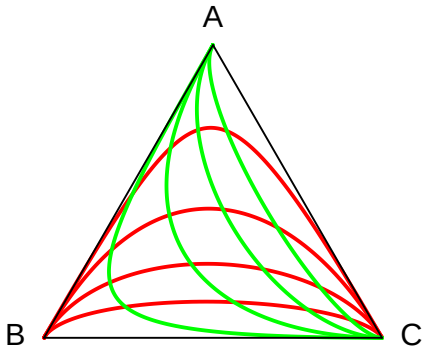


FIGURE 3 – Exemple de droites compositionnelles dans le simplexe

Les mesures statistiques telles que la moyenne et la variance doivent être elles aussi redéfinies dans ce simplexe pour prendre en compte sa géométrie. Pour un échantillon de taille n de D-compositions, la moyenne est définie comme :

$$\bar{g} = C\left([g_1, g_2, ..., g_D]\right)$$

avec $\bar{g}_i = \left(\prod_{j=1}^{n} x_{ij}\right)^{1/n}$, $i = 1, 2, .., D$
La matrice de covariance quant à elle est définie par

$$T = \begin{pmatrix} t_{11} & t_{12} & ... & t_{1D} \\ ... & ... & ... & ... \\ t_{D1} & t_{D2} & ... & t_{DD} \end{pmatrix}, \quad t_{ij} = var\left(ln\frac{x_i}{x_j}\right),$$

ce qui permet de définir la variance totale :

$$totvar\left[X\right] = \frac{1}{2D}\sum_{i=1}^{D}\sum_{j=1}^{D} t_{ij}$$

Pour centrer des données, il est alors nécessaire de leur retrancher $\bar{g}$, c'est à dire de considérer $(g_1 \oplus (-1) \odot \bar{g}, \ldots, g_n \oplus (-1) \odot \bar{g})$. Pour les réduire, il faut ensuite les diviser par $totvar\left[X\right]^{-1/2}$.

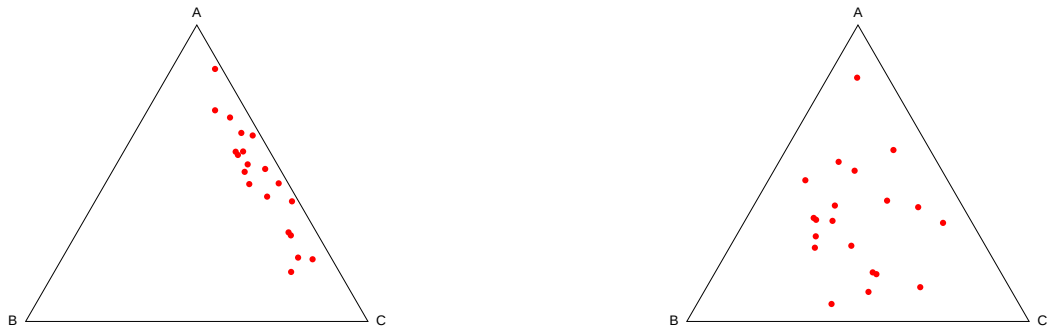


FIGURE 4 – Données compositionnelles avant (gauche) et après (droite) centrage et réduction.

3.3 Transformation

Comme nous avons pu le voir, l'espace du simplexe nécessite une redéfinition des opérations de base et impose sa propre géométrie, ce qui le rend difficile à manipuler. Nous allons voir ici des transformations permettant de repasser du simplexe à un espace euclidien en transformant la géométrie d'Aitchinson en la géométrie usuelle. Remarquons tout d'abord que la valeur de κ (somme des composantes) d'une composition n'a pas fondamentalement une grande importance. En effet, l'étude de

données ne change pas selon qu'elles soient exprimées en pourcentages ou en proportions. Les rapports relatifs des composantes les unes par rapport aux autres sont au contraire très informatifs. Il n'est donc pas étonnant que les transformations reposent sur des log-ratios de composantes.

Commençons par l'*additive log-ratio* transformation (alr).

$$alr : S^D \to \mathbb{R}^{D-1},\ alr\,(x) = \left[ln\left(\frac{x_1}{x_D}\right), ln\left(\frac{x_2}{x_D}\right), ..., ln\left(\frac{x_{D-1}}{x_D}\right)\right]$$

La dernière composante d'une composition pouvant s'exprimer comme :

$$x_D = \kappa - \sum_{i=1}^{D-1} x_i$$

Les D composantes d'une D-composition sont liées et vivent réellement dans un espace de dimension $D-1$. L'alr résout ce problème en prenant une composante comme composante de référence et en comparant toutes les composantes à cette référence. La D-ième cordonnée $\ln(x_D/x_D)$ étant identiquement nulle, elle n'est pas reprise dans la transformation alr. La transformation alr permet de se débarasser du lien linéaire entre les composantes mais est grandement dépendante de la composante choisie comme référence. De plus, cette transformation ne préserve pas les distances, au sens où

$$d_a\,(x, y) \neq \|alr\,(x) - alr\,(y)\,\|_2$$

Cet aspect est facilement observable en reprenant les quatre compositions précédentes : $x_1 = [0.1, 0.4, 0.5]\,,\ x_2 = [0.2, 0.3, 0.5]\,,\ x_3 = [0.4, 0.4, 0.2]\,,\ x_4 = [0.5, 0.3, 0.2]$ et en calculant les distances d'Aitchison entre elles dans le simplexe et les distances euclidiennes entre leurs images par la transformation alr :

	Simplexe				alr		
	x_1	x_2	x_3		x_1	x_2	x_3
x_2	0.71			x_2	0.75		
x_3	1.64	1.18		x_3	2.48	2.01	
x_4	1.86	1.3	0.36	x_4	2.6	2.05	0.36

La transformation *centered log-ratio* (clr) ne souffre pas de ces deux problèmes. Elle s'exprime comme suit :

$$clr : S^D \to \mathbb{R}^D,\ clr\,(x) = \left[ln\frac{x_1}{g(x)}, ..., ln\frac{x_D}{g(x)}\right] = \xi$$

avec g le centre de la composition définie par :

$$g(x) = \left(\prod_{i=1}^{D} x_i\right)^{1/D}$$

Comme vu précédemment, les D composantes sont liées par une relation linéaire et l'espace des D-compositions est de dimension $D-1$. Comme la transformation clr conserve le nombre de coordonnées, les coordonnées images ne sont pas indépendantes : on voit simplement que la somme des coordonnées images est égale à 0. En conséquence, la matrice de covariance des vecteurs $clr(x)$ est singulière.

Voyons enfin la transformation *isometric log-ratio* (ilr) définie par :

$$ilr : S^D \to \mathbb{R}^{D-1},\ ilr\,(x) = clr\,(x)\,\Psi'$$

avec Ψ l'image *clr* d'une base du simplexe S^D (cf. section suivante pour le choix de la base). La transformation ilr permet de résoudre les problématiques des autres transformations : elle préserve les distances et projette les compositions dans un espace de bonne dimension. Pour cela, elle utilise un

type de coordonnées particulier nommé "balance binaire".

Construction et interprétation de Ψ :
Prenons comme système générateur de $\mathbb{R}^D$, les vecteurs canoniques

$$e_i = (\underbrace{0, ..., 0}_{i-1}, 1, 0, ..., 0), \quad i = 1, ..., D$$

en appliquant clr^{-1} et la closure à ces vecteurs, on obtient un système générateur du simplexe $\S^D$,

$$W_i = C([\underbrace{1, ..., 1}_{i-1}, e, 1, ..., 1]), \quad i = 1, ..., D$$

En prenant l'image clr des vecteurs W_i (qui ne sont pas les vecteurs canoniques e_i étant donné que la closure a été appliquée entre les transformations clr^{-1} et clr), puis en utilisant le processus d'orthogonalisation Gram-Schmidt à la matrice $(clr(W_1)|\dots|clr(W_D))$, on obtient la matrice Ψ. Comme la famille des $(clr(W_i))_i$ est de rang $D-1$ (à l'instar de la famille $(W_i)_i$), le processus de Gram-Schmidt conduit à une ligne nulle qui est supprimée de Ψ.

$$\Psi = \begin{pmatrix} \frac{1}{\sqrt{20}} & \frac{1}{\sqrt{20}} & \frac{1}{\sqrt{20}} & \frac{1}{\sqrt{20}} & \frac{-2}{\sqrt{5}} \\ \frac{1}{\sqrt{12}} & \frac{1}{\sqrt{12}} & \frac{1}{\sqrt{12}} & -\frac{\sqrt{3}}{\sqrt{4}} & 0 \\ \frac{1}{\sqrt{6}} & \frac{1}{\sqrt{6}} & -\frac{\sqrt{2}}{\sqrt{3}} & 0 & 0 \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 & 0 \end{pmatrix}$$

TABLE 1 – Exemple de matrice Ψ en dimension 5

Les coordonnées obtenues par la transformation *ilr* ont une interprétation naturelle en terme de *balance binaire*. Considérons la matrice binaire (dans laquelle chaque coefficient est remplacé par son signe) associée à Ψ.

$$\begin{matrix} x1 & x2 & x3 & x4 & x5 \end{matrix}$$
$$\begin{pmatrix} 1 & 1 & 1 & 1 & -1 \\ 1 & 1 & 1 & -1 & 0 \\ 1 & 1 & -1 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 \end{pmatrix}$$

TABLE 2 – Matrice binaire associée à la matrice Ψ du tableau 1

La première coordonnée ilr représente ainsi le ratio entre la moyenne géométrique des quatres premières composantes et la dernière composante. Une composante x avec une grande valeur pour $ilr(x)_1$ met ainsi plus de poids sur la composante x_5 comparativement aux autres. De même, la dernière coordonnée *ilr* représente le rapport entre les composantes x_1 et x_2 : une composition x avec une grande valeur pour $ilr(x)_4$ met plus de poids sur la deuxième composante que sur la première. Toutes les coordonnées ilr peuvent se comprendre en terme de balance entre un premier ensemble de composantes (celles affectées du signe +1 sur cette cordonnées) et un deuxième ensemble (celles affectées du signe -1).

Il peut être intéressant d'utiliser des balances binaires faciles à interpréter plutôt que celles construites par la procédure par défaut. Par exemple, si des personnes interrogées dans le cadre d'un sondage ont eu le choix entre cinq réponses possible A,B,C,D,E et qu'on souhaite comparer le poids de A,B,C à celui de D,E, on peut considérer la balance binaire $(1, 1, 1, -1, -1)$ et la coordonnée ilr associée. Une coordonnée ilr inférieure (resp. supérieure) à 0 indique que la moyenne géométrique de D,E est supérieure (resp. inférieure) à celle de A,B,C.

Ces transformations permettent de passer dans un espace euclidien et de retrouver les formes géométriques habituelles (Fig. 5). Elles ont bien également chacune une transformation inverse définie par :

$$
\begin{aligned}
alr^{-1}(\xi) &= C\left(\frac{\exp(\xi_1)}{\sum_{i=1}^{d-1}\exp(\xi_i)+1},\ ...,\ \frac{1}{\sum_{i=1}^{d-1}\exp(\xi_i)+1}\right)\\
clr^{-1}(\xi) &= C\left(\exp(\xi_1),\ ...,\ \exp(\xi_D)\ \right)\\
ilr^{-1}(\xi) &= C\left(\exp \xi \Psi\right)
\end{aligned}
$$

Remarque La transformation clr est un morphisme de $\left(S^D, \oplus, \odot\right)$ dans $\left(\tilde{R}^D, +, \times\right)$ avec $\tilde{R} = \left\{x \in \mathbb{R}^D, \sum_{i=1}^{D} x_i = 0\right\}$. De même la transformation ilr est un morphisme de $\left(S^D, \oplus, \odot\right)$ dans $\left(R^{D-1}, +, \times\right)$.

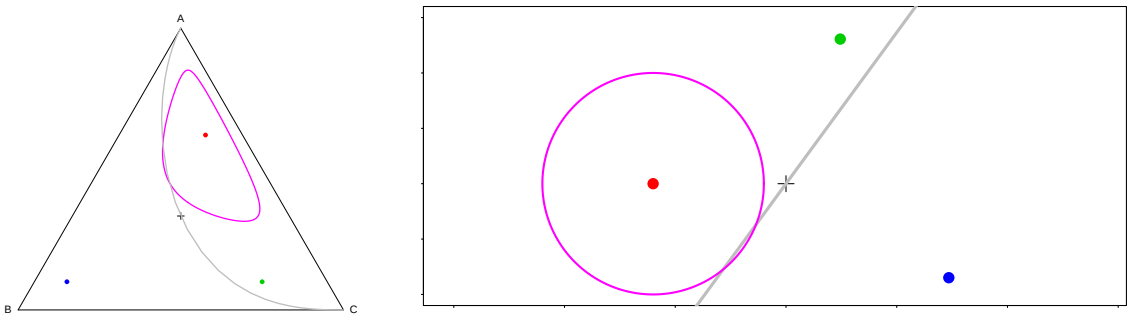


FIGURE 5 – Données et formes compositionnelles avant (gauche) et après (droite) transformation ilr.

Si nous reprenons l'exemple du début de cette partie (avec les quatre compositions), nous remarquons bien que la transformation ilr ne préserve pas la distance euclidienne (Fig. 6). L'intuition visuelle associée aux distances euclidiennes est fausse : une distance (euclidienne) à priori similaire entre deux points (panel de gauche) correspond à des distances très différentes dans l'espace ilr (panel de droite). On peut noter en particulier que les distances sont d'autant plus déformées qu'on se rapproche des bords du simplexe.

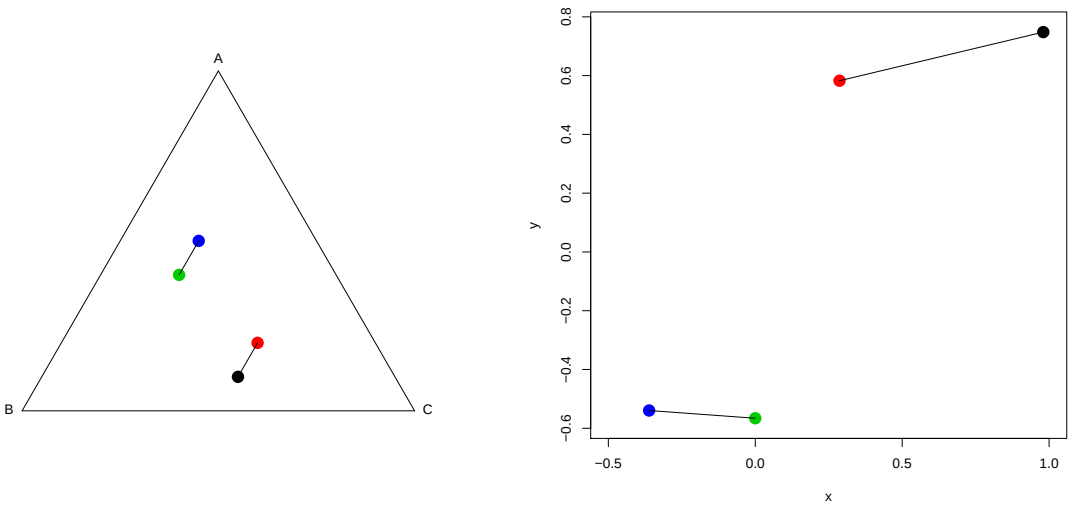


FIGURE 6 – Distances euclidiennes dans le simplexe (gauche) et dans l'espace euclidien après transformation ilr (droite).

4 Analyse Multivariée

4.1 Réduction de dimension

L'ACP est une méthode de réduction de dimension couramment utilisée pour synthétiser l'information contenue dans un ensemble d'échantillons. Aitchison redéfinit l'ACP pour données compositionnelles en centrant et réduisant les données dans le simplexe avant de calculer les vecteurs propres de la matrice de covariance totale.

Nous préférons ici tirer parti du fait que ilr est un morphisme en définissant l'ACP pour données compositionnelles comme l'ACP standard sur données ilr-transformées. Les deux définitions sont rigoureusement équivalentes.

4.2 Maximum a posteriori

Nos jeux de données étant des jeux de données de comptage, de nombreux zéros sont présents comme expliqué précédemment et les transformations logarithmiques (ilr, clr) ne peuvent donc pas être directement appliquées : il est nécessaire d'effectuer un lissage lors du passage des données de comptage aux proportions. Pour ce faire, nous n'utilisons pas l'estimateur du maximum de vraisemblance (sous un modèle multinomial) pour estimer la proportion de chaque OTU dans un échantillon mais lui préférons l'estimateur bayesien du Maximum A Posteriori (MAP) avec un priori uniforme sur les compositions. Formellement, en mettant une distribution a priori sur les proportions θ et en utilisant le théorème de Bayes, nous avons

$$\begin{aligned} P(\theta|X) &= \frac{P(X|\theta)\,P(\theta)}{P(X)} \\ &\propto P(X|\theta)\,P(\theta) \end{aligned}$$

Dans notre cas, nous allons prendre un a priori de Dirichlet, plus précisément $Dir(\alpha)$ avec $a = (1, ..., 1,)$ ce qui correspond à une loi uniforme sur le simplexe. Il en découle l'estimateur suivant :

$$MAP(\theta) = \underset{\therefore}{argmax}\,(\theta|X) = \left(\frac{X_i + 1}{\sum_{i=1}^{D}(X_i + 1)}\right)_{i=1...D}$$

Cela revient simplement à ajouter un pseudo-comptage de 1 à chaque OTU avant de considérer l'estimateur du maximum de vraisemblance correspond.

5 Transformation ilr et classification

5.1 Jeux de données

Cinq jeux de données, décrits ci-après, ont été mis à ma disposition. Ils se présentent sous la forme de table de comptage d'OTU accompagnés de quelques variables descriptives d'intérêt.

Chaillou[3]

Il s'agit d'un jeu de données issues de matrices alimentaires, qui s'intéresse aux flores liées à l'altération des aliments après la date de péremption. Pour cela, quatre types de viande : boeuf haché, veau hache, lardons, saucisses de volaille et quatre types de produits de la mer : saumon fumé, crevettes, filet de saumon et filet de cabillaud ont été échantillonnés. Pour chaque type de nourriture, huit échantillons de lots différents (réplicats biologiques) ont été étudiés et 508 OTU ont été conservés lors de l'étude. Durant mon stage, la variable qualitative d'intérêt était la matrice alimentaire d'origine.

Mach[5]

L'évolution du microbiote intestinal de porcelet au cours des premiers mois (avant et après le sevrage) ainsi que son impact sur le métabolisme était le sujet de l'étude contenant ce jeu de données. Les fèces de 31 porcelets ont été séquencés à cinq périodes de leur vie (14, 36, 48, 60 et 70 jours, le sevrage intervenant entre les jours 14 et 36). Le jeu de données contient ainsi 155 échantillons et le nombre d'OTU s'élève à 4031.

Liver[8]

L'objectif de cette étude était de caractériser le microbiote intestinal lié à la cirrhose du foie, pour cela le microbiote de 114 patients sains et 123 patients atteint de cirrhose ont été séquencés. Pour chaque patient, plusieurs caractéristiques cliniques ont également été relevées : indice de masse corporelle, indice de coagulation du sang, tests du bon fonctionnement des reins, etc. Le jeu de données contient ainsi 237 échantillons et le nombre de MGS s'élève à 1529.

Ravel[9]

Dans cette étude, les chercheurs s'intéressent au lien entre le microbiote vaginal et les maladies urogenitales. Pour étudier ce lien, ils ont séquencé le microbiote vaginal de 394 femmes provenant de quatre catégories ethniques différentes (asiatiques, blanches, noires et hispaniques). 247 OTU ont été gardés pour réaliser cette étude. Sur chaque échantillon (un pour chaque femme), les chercheurs ont également mesuré le *nugent score*, un indice de vaginose bactérienne, qui a été discrétisé en 3 catégories (low, intermediate, high). En parallèle de ce paramètre clinique, ils ont défini 5 archétypes de communautés : les Community State Type (CST) que j'ai étudiés durant mon stage.

Vacher[4]

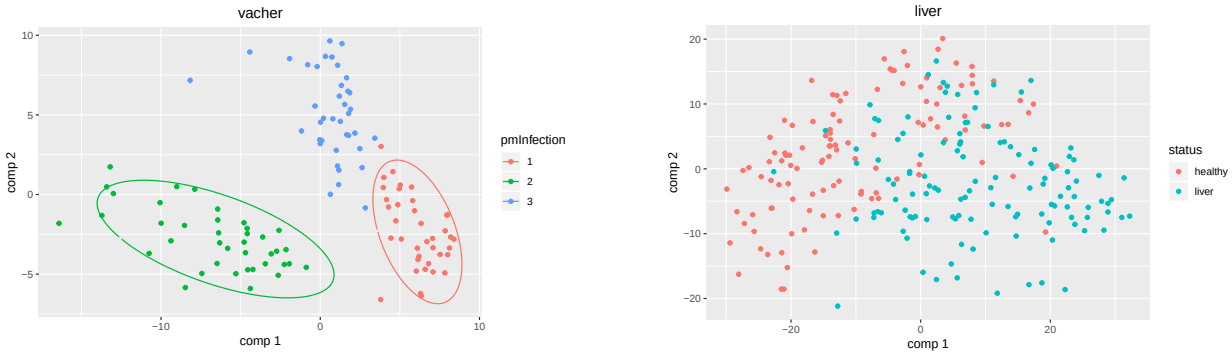
L'objectif de l'étude d'où provienne les données était de trouver les interactions les plus probables entre un champignon responsable de l'oidium du chêne (*Erysiphe alphitoides*) et les autres espèces de microorganismes présentes sur les feuilles de chêne. Pour ce faire, les feuilles de trois chênes ont été prélevées et séquencées (40 sur chaque arbre). Pour chaque arbre, les feuilles proviennent de quatre branches différentes (10 pour chaque branche) et plusieurs caractéristiques ont été relevées pour chaque feuille : sa distance à la base de l'arbre, sa distance au tronc de l'arbre et sa distance par rapport au sol ainsi que son orientation (South west, North East) et son niveau d'infection. Cependant dans le jeu de données, seuls 116 échantillons de feuilles sont présents (il en manque quatre) et 114 OTU ont été mesurés sur chaque feuille. L'objectif durant mon stage était d'étudier le lien entre l'arbre de prélèvement et la communauté microbienne de chaque feuille.

Pour Liver et Mach, le nombre d'OTU étant très grand et la matrice de comptage extrêmement creuse, j'ai choisi d'en enlever une partie (ceux présents dans moins de 5% des échantillons). Les

nouvelles dimensions des jeux de données sont ainsi : 155 échantillons sur 1084 OTU pour Mach, 237 échantillons et 533 MGS pour liver. Dans la suite du rapport, seuls les résultats obtenus sur les jeux de données Vacher et Liver seront présentés. Les résultats sur les autres jeux de données sont comparables et présentés en annexe.
J'ai choisi de présenter les résultats de ces deux jeux de données pour leurs caractéristiques, Liver est le jeu de donnée dans lequel la structure des données est la plus difficile a analyser tandis que Vacher a quant à lui des résultats similaires (les résultats visibles dans la suite du rapport) à Ravel et Mach sans pour autant avoir des caractéristiques trop différentes entre les différents groupes (type de nourriture) d'échantillons. Ces deux jeux de données proviennent de deux types de séquencage différents (shotgun pour Liver et amplicon pour Vacher).

5.2 Graphique ACP

Pour avoir un premier aperçu des jeux de données, faisons une ACP sur les données compositionnelles et regardons la projection des échantillons sur les deux premiers axes.



Pour Vacher la séparation des échantillons issus des différents arbres est très franche.
Dans le cas des données de Liver, la séparation est moins franche, mais la densité le long du premier axe montre une différence entre les individus sains et les individus malades.

5.3 Classification non supervisée

Dans chacun des jeux de données utilisés, les groupes semblent relativement bien séparés comme vu précédement (tout du moins sur les premiers axes d'une acp après transformation ilr). Une des questions que nous pouvons alors nous poser est : le passage aux données compositionnelles (et transformation ilr) permet-il de bien retrouver une structure de groupes connue ?

Une classification non supervisée répond bien à cette question et j'ai donc utilisé plusieurs algorithmes (k-means, mélange gaussien, classification hiérarchique) en fixant le nombre de groupes et regardé s'ils avaient tendance à retrouver les groupes connus. Pour montrer l'intérêt de passer par les données compositionnelles pour la classification, j'ai comparé les résultats d'une classification effectuée sur les données initiales (de comptage) et sur les données compositionnelles (après transformation ilr).

Vacher
K-means

comptage

	1(obs)	2(obs)	3(obs)
1(pred)	1	0	0
2(pred)	0	3	1
3(pred)	37	36	38

ilr

	1(obs)	2(obs)	3(obs)
1(pred)	38	0	0
2(pred)	0	0	39
3(pred)	0	39	0

Classification hiérarchique

comptage

	1(obs)	2(obs)	3(obs)
1(pred)	1	3	1
2(pred)	16	16	11
3(pred)	21	20	27

ilr

	1(obs)	2(obs)	3(obs)
1(pred)	2	1	39
2(pred)	36	0	0
3(pred)	0	38	0

Mélange gaussien

comptage

	1(obs)	2(obs)	3(obs)
1(pred)	38	38	38
2(pred)	0	1	0
3(pred)	0	0	1

ilr

	1(obs)	2(obs)	3(obs)
1(pred)	1	0	0
2(pred)	37	39	38
3(pred)	0	0	1

Liver
K-means

comptage

	healthy(obs)	liver(obs)
1(pred)	93	91
2(pred)	21	32

ilr

	healthy(obs)	liver(obs)
1(pred)	32	93
2(pred)	82	30

Classification hiérarchique

comptage

	healthy(obs)	liver(obs)
1(pred)	12	26
2(pred)	102	97

ilr

	healthy(obs)	liver(obs)
1(pred)	42	98
2(pred)	72	25

Mélange gaussien

comptage

	healthy(obs)	liver(obs)
1(pred)	73	100
2(pred)	41	23

ilr

	healthy(obs)	liver(obs)
1(pred)	42	94
2(pred)	72	29

La transformation *ilr* semble améliorer sensiblement la classification non supervisée. Les classificateurs ont tendance, sur les données de comptage, à mettre tous les échantillons dans le même groupe. La séparation entre les groupes n'a pas l'air toujours très franche comme dans le cas de Liver (un grand nombre d'échantillons est mal classé) aussi bien pour les données de comptage que pour les données compositionnelles mais même dans ce cas, le passage par les données compositionnelles améliore l'accord entre les classifications.

5.4 Classification supervisée

Regardons si des classificateurs supervisés permettent de mieux faire la différence entre les groupes (notamment sur les jeux de données où une classification non supervisée ne donne pas de bon résultats). Pour cela une classification supervisée a été réalisée par le biais de trois algorithmes qui sont, le randomForest (avec la matrice de confusion), le Support Vector Machine (SVM) et le k-Nearest Neighbors (k-NN), en faisant pour les deux derniers une 10-fold cross validation.

Vacher
random Forest

comptage

	1(obs)	2(obs)	3(obs)
1(pred)	38	0	0
2(pred)	0	39	0
3(pred)	0	0	39

ilr

	1(obs)	2(obs)	3(obs)
1(pred)	38	0	0
2(pred)	0	39	0
3(pred)	0	0	39

svm

comptage

	1(obs)	2(obs)	3(obs)
1(pred)	32	0	0
2(pred)	6	39	5
3(pred)	0	0	34

ilr

	1(obs)	2(obs)	3(obs)
1(pred)	37	0	0
2(pred)	1	39	1
3(pred)	0	0	38

kNN

comptage

	1(obs)	2(obs)	3(obs)
1(pred)	24	1	7
2(pred)	0	28	2
3(pred)	14	10	30

ilr

	1(obs)	2(obs)	3(obs)
1(pred)	38	0	0
2(pred)	0	38	0
3(pred)	0	1	39

Liver
random Forest

comptage

	healthy(obs)	liver(obs)
healthy(pred)	103	11
liver(pred)	19	104

ilr

	healthy(obs)	liver(obs)
healthy(pred)	97	17
liver(pred)	13	110

svm

comptage

	healthy(obs)	liver(obs)
healthy(pred)	77	16
liver(pred)	37	107

ilr

	healthy(obs)	liver(obs)
healthy(pred)	105	19
liver(pred)	9	104

kNN

comptage

	healthy(obs)	liver(obs)
healthy(pred)	90	55
liver(pred)	24	68

ilr

	healthy(obs)	liver(obs)
healthy(pred)	90	22
liver(pred)	24	101

Les meilleurs performances des classificateurs sur les données compositionnelles que sur les données de comptage sont confirmés, les classifications sur les données compositionnelles (après transformation ilr) ont, dans le pire des cas, les mêmes performances que celles sur données de comptage. Malheureusement, dans le cas de certains classificateurs tel que le svm, la classification supervisée sur certains jeux de données de comptage ne donne pas du tout des résultats concluants (voir Chaillou en Annexe .3).

Que ce soit pour la classification supervisée ou non supervisée, les résultats sont meilleurs sur les données compositionnelles (après transformation ilr) que sur les données initiales. Il semble donc y avoir un intérêt à passer par les données compositionnelles pour la classification.

6 Simulateur

Passons maintenant à la partie simulation de jeux de données. L'objectif est de développer un simulateur polyvalent, basé sur un modèle probabiliste et capable de générer des jeux de données présentant les mêmes caractéristiques que les données réelles. Un tel simulateur a vocation à générer des données réalistes, dans un cadre contrôlé, pour comparer entre elles des méthodes statistiques concurrentes (par exemple de détection d'abondance différentielle ou de clustering) sans en favoriser une par rapport aux autres.

Dans les grandes lignes, le simulateur apprend la densité des données réelles comme suit. Les données initiales (données de comptage) sont tout d'abord transformées en données compositionnelles (via le MAP), avant d'être envoyées dans $\mathbb{R}^{D-1}$ avec la transformation ilr. Une fois revenu dans un espace euclidien, le simulateur apprend la densité des données à l'aide d'une réduction de dimensions et d'un mélange gaussien dans l'espace réduit. La simulation de nouvelles données peut alors être réalisée en tirant des données suivant le mélange gaussien puis en appliquant les opérations inverses ("augmentation de dimension", ilr^{-1}, etc.) pour obtenir un nouveau jeu de données de comptage. Lors du calcul de l'estimateur MAP, un a priori de Dirichlet a été choisi. La transformation inverse, qui consiste à passer d'un point du simplexe à un vecteur de comptages, est un tirage suivant une loi multinomiale.

Les jeux données réelles présentant un grand nombre de 0, une gestion spécifique de cette caractéristique est nécessaire et une zéro-inflation est effectuée en tout fin de processus, après le tirage multinomial, afin d'obtenir le jeu de données simulées final. Le schéma récapitulatif du simulateur est présenté Fig. 7. Les étapes non décrites (réduction de dimension, estimation de densité), le sont en détails à la suite de ce schéma.

6.1 Schéma simulateur

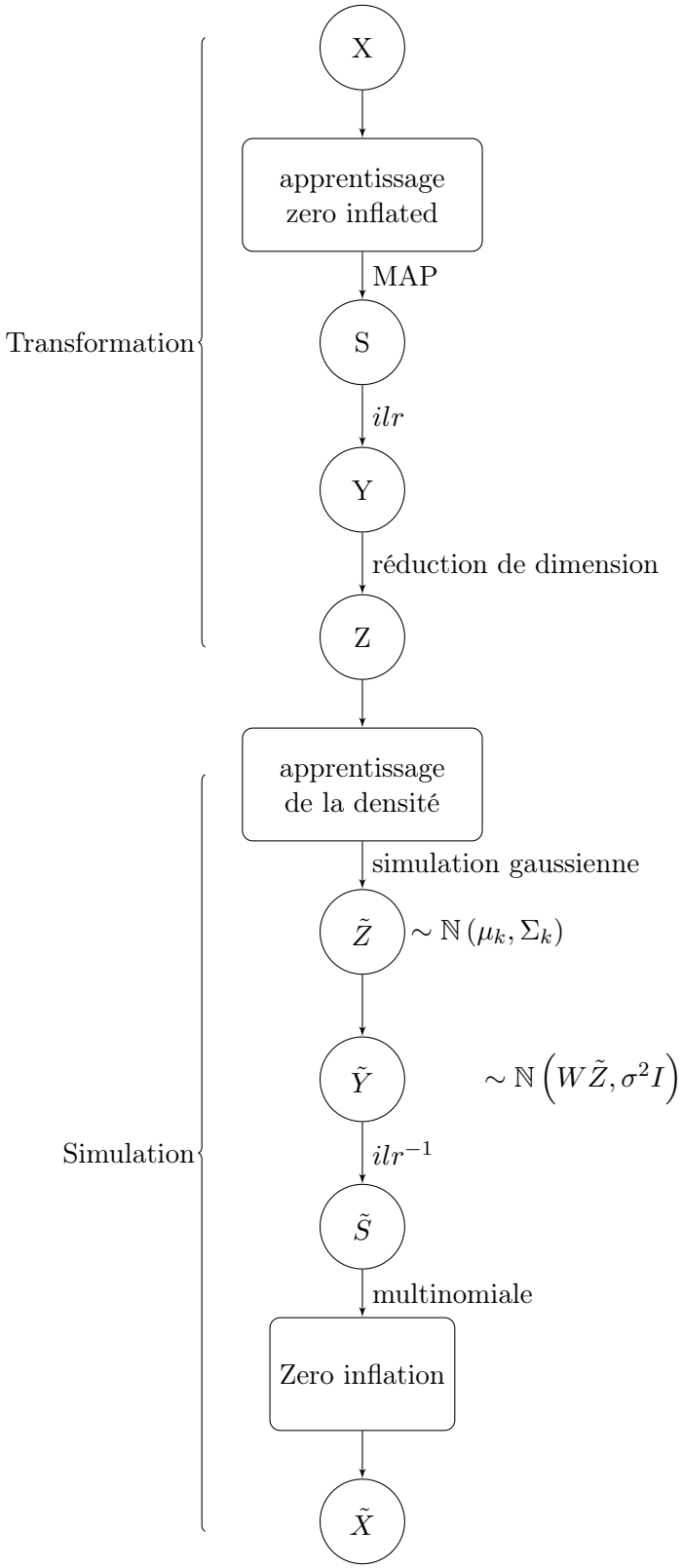


FIGURE 7 – Schéma du simulateur de données.

6.2 Réduction de dimension

Les jeux de données utilisés dans le cadre du simulateur ont un grand nombre de variables, largement supérieur au nombre d'échantillons disponibles (grande dimension) : une réduction de dimension s'im-

pose donc avant d'apprendre la densité des observations. Dans notre cas, nous cherchons à construire un modèle probabiliste pour simuler nos données. L'ACP dans sa définition la plus commune n'est pas associée à un modèle probabiliste et ne permet pas de construire un tel modèle. Nous allons donc lui substituer une variante de l'ACP, l'ACP probabiliste[11] qui est définie comme :

$$X_i = W y_i + \mu + \epsilon_i$$

où $y_i \sim \mathcal{N}(0, I_d)$, $\epsilon_i \sim \mathcal{N}\left(0,\ \sigma^2 I_p\right)$ est un bruit blanc gaussien et W une $p \times d$ matrice.
Si nous notons A la matrice des vecteurs propres de la matrice $X^T X$, Λ la matrice diagonale contenant les valeurs propres associées et R une matrice orthogonale arbitraire, on peut montrer que le maximum de vraisemblance de W s'exprime comme :

$$W_{ML} = A\left(\Lambda - \sigma^2 I_d\right)^{1/2} R$$

De même, l'estimateur du maximun de vraisemblance de σ^2

$$\sigma^2_{ML} = \frac{1}{d-q} \sum_{j=q+1}^{d} \lambda_j$$

Cet estimateur peut être perçu comme la variance perdue lors de la projection. Le choix de d, la dimension de l'espace latent, est toujours une question délicate dans les modèles à variables latentes. Nous avons utilisé ici l'heuristique de pente (du package capushe[2]) pour choisir d.

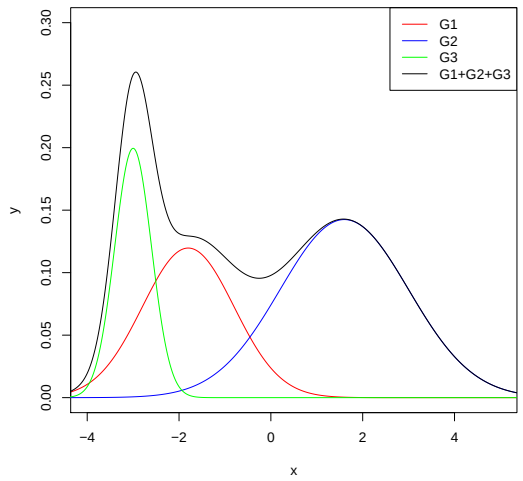
6.3 Mélange Gaussien

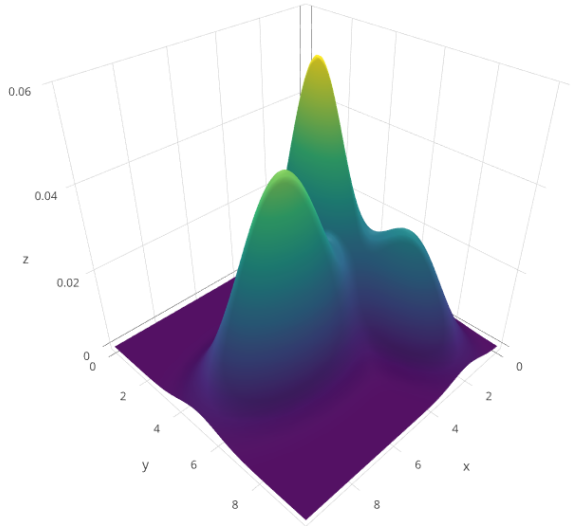
La densité des données dans le sous-espace de dimension d est apprise à l'aide d'un mélange gaussien, défini comme :

$$g(x; \pi, \theta) = \sum_{k=1}^{K} \pi_k f(x, \theta_k)$$

où $\theta = (\theta_1, ..., \theta_k)$ sont les paramètres des lois gaussiennes et $\pi = (\pi_1, \ldots, \pi_K)$ sont les proportions du mélange Gaussien, qui somment à 1.

Voici quelques exemples de mélange gaussien





Pour le simulateur, le nombre de gaussiennes nécessaires à l'estimation de la densité est choisi à l'aide du critère BIC.

6.4 Zero inflation

Les jeux de données contiennent un nombre important de zéros (comme vu précédemment), la modélisation faite par le mélange gaussien ne parvient pas à bien prendre en compte cette surreprésentation. Pour résoudre ce problème, nous allons modéliser cette particularité indépendamment comme suit.

Commençons par estimer la proportion de zéros pour chaque OTU j dans chaque composante gaussienne k, notée π_{jk}. Cette estimation est faite en attribuant chaque échantillon à sa composante gaussienne la plus probable puis en calculant au sein de chaque composante k, le nombre total N_k d'échantillons et le nombre N_{jk} d'échantillons dans lesquels l'OTU j est absent. π_{jk} est estimée par

$$\pi_{jk} = \frac{N_{jk} + \alpha}{N_k + \alpha + \beta}$$

avec $\alpha = 0.09$ et $\beta = 0.01$ pour lisser les probabilités et éviter d'avoir des valeurs à 1 ou 0. Les valeurs de α et β ont été choisies de sorte $E(\pi_{jk}) = 0.9$ la proportion moyenne de zéros dans les jeux de données. La régularisation adoptée revient à mettre un à priori $Beta(\alpha, \beta)$ sur la probabilité de 0-inflation. Par la suite, nous modélisons les zéros à l'aide d'une loi de Bernoulli,

$$P\left(Z_{jk} = x\right) = \begin{cases} \pi_{jk} & \text{si } x = 0 \\ 1 - \pi_{jk} & \text{si } x = 1 \end{cases}$$

Ce qui permet d'exprimer $\tilde{X}_i$ comme

$$\tilde{X}_i = (Z_{1k}, Z_{jk}, ..., Z_{Dk}) \times T_i$$

où $Z_{jk} \sim Bern\left(\pi_{jk}\right)$ et $T_i \sim \mathbb{M}\left(N; \tilde{S}_i\right)$ et $\tilde{S}_i$ qui est le résultat de $ilr^{-1}\left(\mathbb{N}\left(W\mu_k, W\Sigma_k W^T + \sigma^2 I\right)\right)$

Illustrons ce mécanisme à l'aide d'un jeu de données à 2 OTU. Lors de l'apprentissage, le simulateur a trouvé que le nombre de gaussiennes optimal est de 2 et il a calculé les estimateurs de zero inflation suivant :

	Gaussiennes	
OTU		
	1/3	1/2
	2/3	3/4

TABLE 3 – Exemple de matrice de zero-inflation.

Il simule un nouveau jeu de données (tableau de gauche), 3 données appartiennent à la gaussienne rouge et 5 à la gaussienne verte. Lors de la zero inflation, le simulateur va multiplier chaque comptage d'OTU du nouveau jeu de données (chaque case du tableau) par une variable de bernoulli de probabilité définie dans la case correspondante (OTU et gaussienne identique) du tableau Tab. 3. Le résultat final obtenu par le simulateur est alors le jeu de données de droite.

Avant zero inflation

1	2	18	5	9	7	0	21
32	2	3	6	2	1	2	14

Après zero inflation

1	0	18	0	9	0	0	21
0	0	3	0	0	1	0	14

6.5 Correction de la profondeur de séquençage

La zéro inflation rajoute des zéros dans les jeux de données simulées, la profondeur de séquencage (le nombre de lecture par échantillon) va en conséquence diminuer. Pour remédier à ce problème et éviter que la profondeur de séquencage finale ne soit trop inférieure à celle souhaitée, il est nécessaire de

l'augmenter au préalable pour corriger l'effet de la zéro inflation. Cela permet de contrôler la profondeur de séquençage *moyenne* produite par le simulateur.

Un estimateur de la proportion de zéro rajoutés dans un échantillon i par la zéro inflation est :

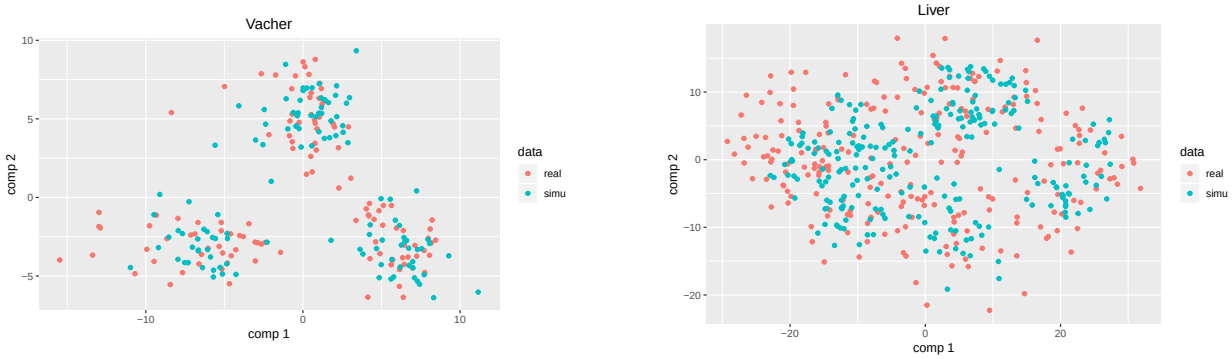
$$\hat{p_i} = \sum_{j=1}^{D} \pi_{jk} \tilde{S}_{ij}$$

La profondeur de séquencage (N de la multinomiale) à utiliser dans le simulateur est donc $\hat{N}_i = N_i / (1 - \hat{p_i})$.

7 Performances du simulateur

7.1 Méthode

La méthode d'évaluation la plus facile à mettre en œuvre est d'estimer visuellement si les données simulées ressemblent aux données réelles. Pour ce faire j'ai projecté les données (réelles et simulées) sur les premiers axes d'une ACP.



Globalement, les données simulées se mélangent bien avec les données réelles. La première impression sur les performances du simulateur est donc positive.

Cependant cette méthode, bien que très utilisée dans la littérature, n'est pas forcément rigoureuse. En effet, certains OTU peuvent être très mal simulés sans pour autant que nous ne puissions faire la différence visuellement. Par exemple, un OTU absent de tous les échantillons réels mais présent en très faible quantité dans les échantillons simulés n'aura quasiment aucune incidence sur une ACP (et n'induira donc aucune différence graphique). Cet OTU n'est pas pour autant bien simulé ; une bonne simulation donnerait des échantillons dans lesquels cet OTU est absent la plupart du temps.

Pour avoir des résultats plus fiables, l'étude de l'efficacité du simulateur est confiée à des classificateurs. L'objectif est de regarder si un classificateur parvient à faire la différence entre les données réelles et les données simulées. Une bonne simulation mènerait le classificateur à se tromper à de nombreuses reprises (l'objectif théorique est que le classificateur se trompe dans 50% des cas). Dans ce but, deux nouveaux jeux de données sont simulés, l'un d'entre eux servira de jeu d'entrainement (avec les données réelles) à un classificateur, l'autre jeu de données servira quant à lui de test. Une petite partie des données réelles n'est pas incluse dans le jeu d'entrainement, elle servira elle aussi de jeu de données test.

7.2 Résultats

J'ai utilisé deux classificateurs, les random Forest (RF) ainsi que le k Nearest Neighbors (K-NN). Le classificateur SVM a été ignoré en raison de sa faible efficacité sur les données de comptage, voir 5.4).

Vacher

Random forest

	real(obs)	simu(obs)
real(pred)	90	29.06
simu(pred)	10	70.94

kNN

	real(obs)	simu(obs)
real(pred)	73.64	54.68
simu(pred)	26.36	45.32

Liver

Random forest

	real(obs)	simu(obs)
real(pred)	93.04	5.84
simu(pred)	6.96	94.16

kNN

	real(obs)	simu(obs)
real(pred)	93.91	68.62
simu(pred)	6.09	31.38

Les performances du simulateur sont grandement variables selon le jeu de données. En effet, sur certains jeux de données, notamment Vacher, les classificateurs peinent à faire la différence entre les données simulées et réelles. À l'inverse sur Liver, le random Forest détecte aisément les données simulées

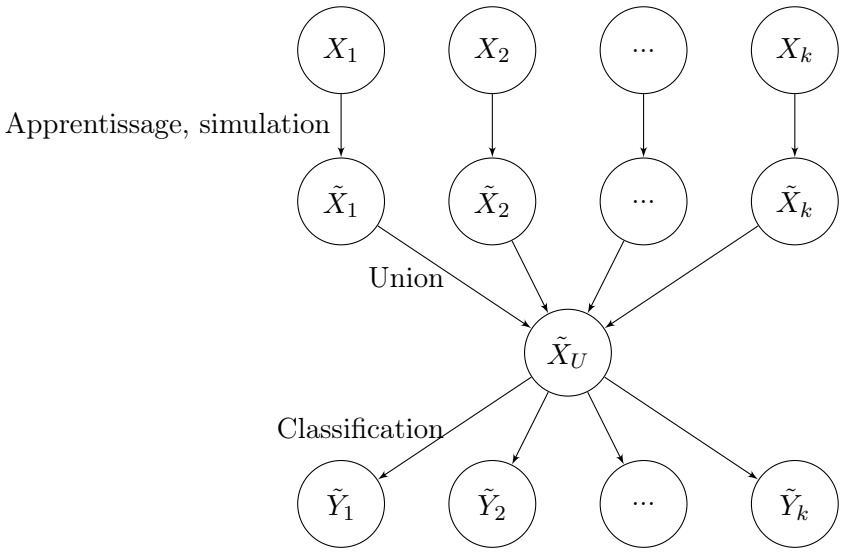


FIGURE 8 – Procédé utilisé pour évaluer la qualité du simulateur, notamment sa capacité à reproduire les caractéristiques de groupes différents.

tandis le k-NN y arrive moins facilement. Ainsi certaines caractéristiques des données Liver doivent être mal simulées (ce qui permet au random Forest de faire la différence), mais les données simulées doivent quand même relativement bien se mélanger aux données réelles (comme le montre le kNN).

7.3 Classification

Un autre aspect de la performance du simulateur est sa capacité à capturer et simuler les caractéristiques (les OTU présents et leurs abondances) de chaque groupe d'un jeu de données (exemple : les différents arbres dans le cas de Vacher ; le status malade ou sain des patients pour Liver) ; une donnée simulée ne devrait pas présenter les caractéristiques de plusieurs groupes. Pour étudier cet aspect, la simulation ne se fera plus sur l'ensemble du jeu de données mais séparément sur chaque groupe du jeu de données (par exemple sain puis malade). Les groupes des données simulées seront alors connus et nous pourrons vérifier si un classificateur *entrainé sur les données réelles* retrouve bien ces groupes.[3]

La figure Fig. 8 présente un schéma illustratif de la méthode : soit $1, 2, ..., k$ les groupes du jeu de données (ex : pour Liver k=2, malade, non malade)

Une bonne simulation devrait résulter en la présence des mêmes échantillons dans $\tilde{X}_i$ et dans $\tilde{Y}_i$ pour $i \in \{1, 2, ..., k\}$.

Vacher

Random forest

	1(obs)	2(obs)	3(obs)
1(pred)	100	0	0
2(pred)	0	100	0
3(pred)	0	0	100

kNN

	1(obs)	2(obs)	3(obs)
1(pred)	75.79	13.59	23.33
2(pred)	3.95	66.15	3.85
3(pred)	20.26	20.26	72.82

Liver

Random forest

	healthy(obs)	liver(obs)
healthy(pred)	92.54	3.98
liver(pred)	7.46	96.02

kNN

	healthy(obs)	liver(obs)
healthy(pred)	67.63	31.79
liver(pred)	32.37	68.21

La classification supervisée des données simulées est globalement bonne. Les résultats sont même quasiment comparables aux résultats de la validation croisée sur les données réelles (pour le random Forest).

3. Ou plus prosaïquement atteint les mêmes taux de bonne et mauvaise classification que sur les données réelles.

Ainsi même si une partie des échantillons simulés est malheureusement "mal simulée" (notamment dans le cas du jeu de données Liver) ; les caractéristiques des groupes restent quant à eux relativement bien simulés.

Une partie des erreurs de classification peut être dûe aux données "mal simulées" qui ne ressemblent aux données d'aucun groupe. Pour tester cette hypothèse et vérifier si les données considérées comme bien "simulées" sont faciles à classer dans un groupe (et ainsi relativement proches des données réelles de leur groupe respectif), nous allons procéder à une nouvelle expérience numérique et conserver uniquement les données "bien simulées". Nous allons ensuite utiliser le même classificateur que précédemment et vérifier la bonne classification de ces données (Fig. 9).

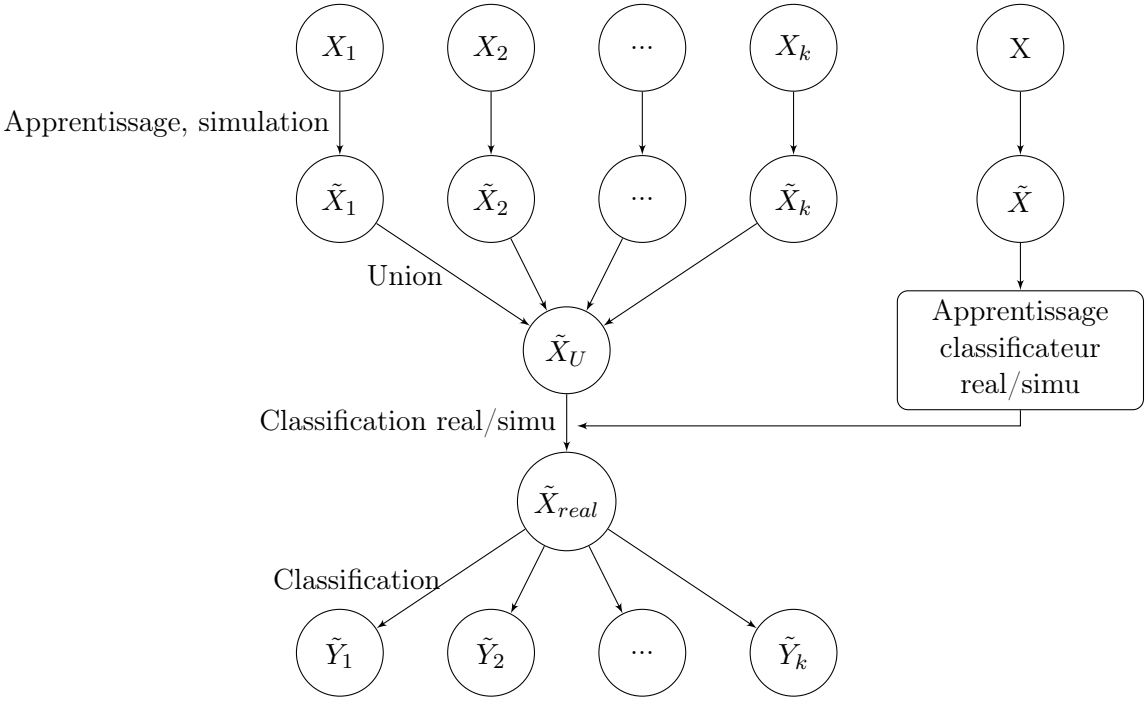


FIGURE 9 – Procédé utilisé pour évaluer la qualité du simulateur, notamment sa capacité à reproduire les caractéristiques de groupes différents, en se restreignant aux données bien simulées.

Vacher

Random forest

	1(obs)	2(obs)	3(obs)
1(pred)	100	0	0
2(pred)	0	100	0
3(pred)	0	0	100

kNN

	1(obs)	2(obs)	3(obs)
1(pred)	71.88	10.42	14.99
2(pred)	3.09	75.56	4.29
3(pred)	25.02	14.03	80.72

Liver

Random forest

	healthy(obs)	liver(obs)
healthy(pred)	99.44	1.27
liver(pred)	0.56	98.73

kNN

	healthy(obs)	liver(obs)
healthy(pred)	84.07	43.63
liver(pred)	15.93	56.37

Les résultats sont meilleurs même si comparables à ceux obtenus avec la méthode précédente. Les données considérées comme réelles (bien simulées) sont donc légèrement plus faciles que dans le cas précédent, sans pour autant atteindre les taux de bonne classification observée sur les données réelles. Les jeux de données où les résultats sur l'ensemble des données simulées étaient déjà très convaincants le sont toujours ; mais les jeux de données où la classification des données simulées était plus compliquée reste compliquée (le classificateur kNN pour liver). Cependant sur ces jeux de données, la validation croisée (Classification supervisée) sur les données réelles ne permet pas non plus d'obtenir de bons

résultats, la problématique ne semble donc pas venir des données simulées.

8 Conclusion

L'intérêt des données compositionnelles est de pouvoir s'affranchir de certaines contraintes techniques comme la profondeur de séquencage. De plus, les transformations log-ratio permettent de se ramener à l'espace euclidien usuel et ainsi d'accéder à l'ensemble des méthodes statistiques multivariées. L'utilisation des transformations des données compositionnelles pour repasser dans un espace euclidien s'est révélée pertinente sur les jeux de données utilisés durant mon stage, les résultats des différentes classifications étant au moins aussi bons que sur les données initiales.

De même, les résultats obtenus sur les données simulées sont encourageants. Les caractéristiques des différents groupes des jeux de données semblent relativement bien simulées et montrent que les grandes structures peuvent être apprises assez facilement. Cependant, les résultats sont très variables selon les jeux de données. Des tests sur un plus grand nombre de jeux de données seraient nécessaires pour étudier la robustesse du simulateur. L'intérêt d'un tel simulateur est qu'il permet d'obtenir des résultats convenables en terme de réalisme au prix d'une faible complexité computationnelle. Ce simulateur pourrait être amélioré en utilisant des méthodes de réduction de dimension et d'apprentissage de la densité plus sophistiquées (autres que des mélanges gaussiens).

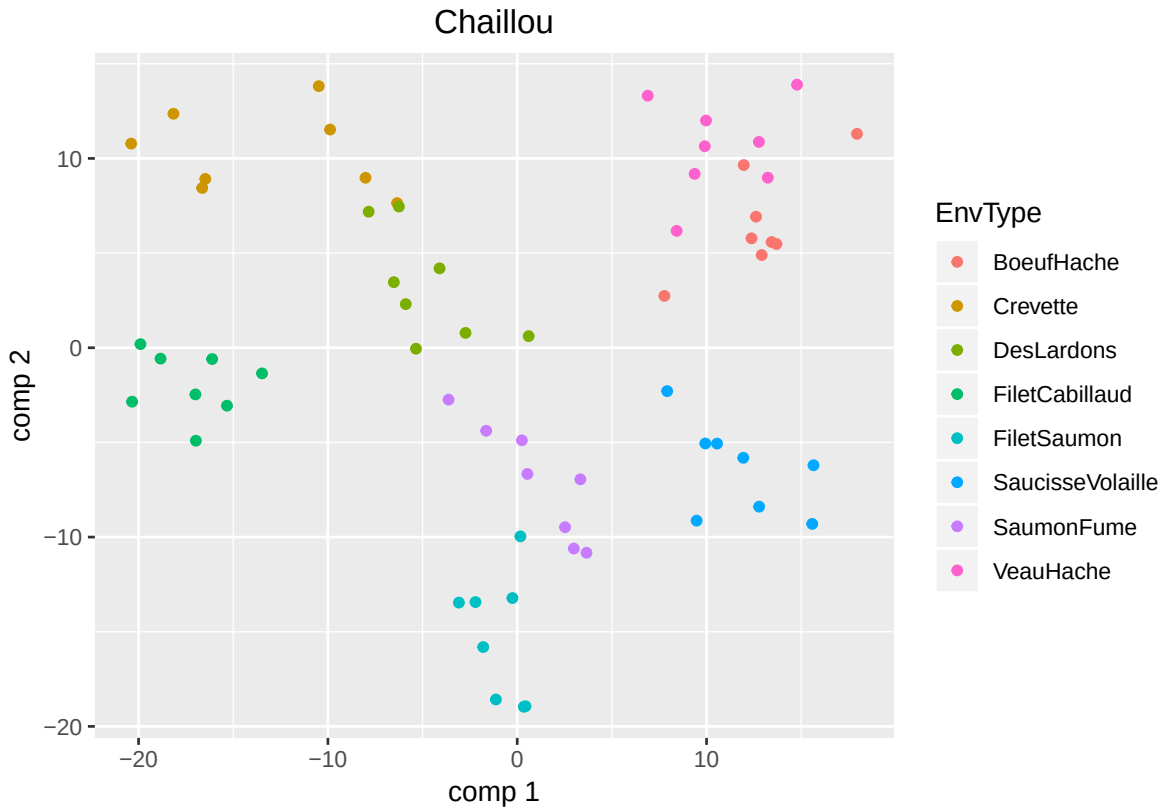
9 Remerciements

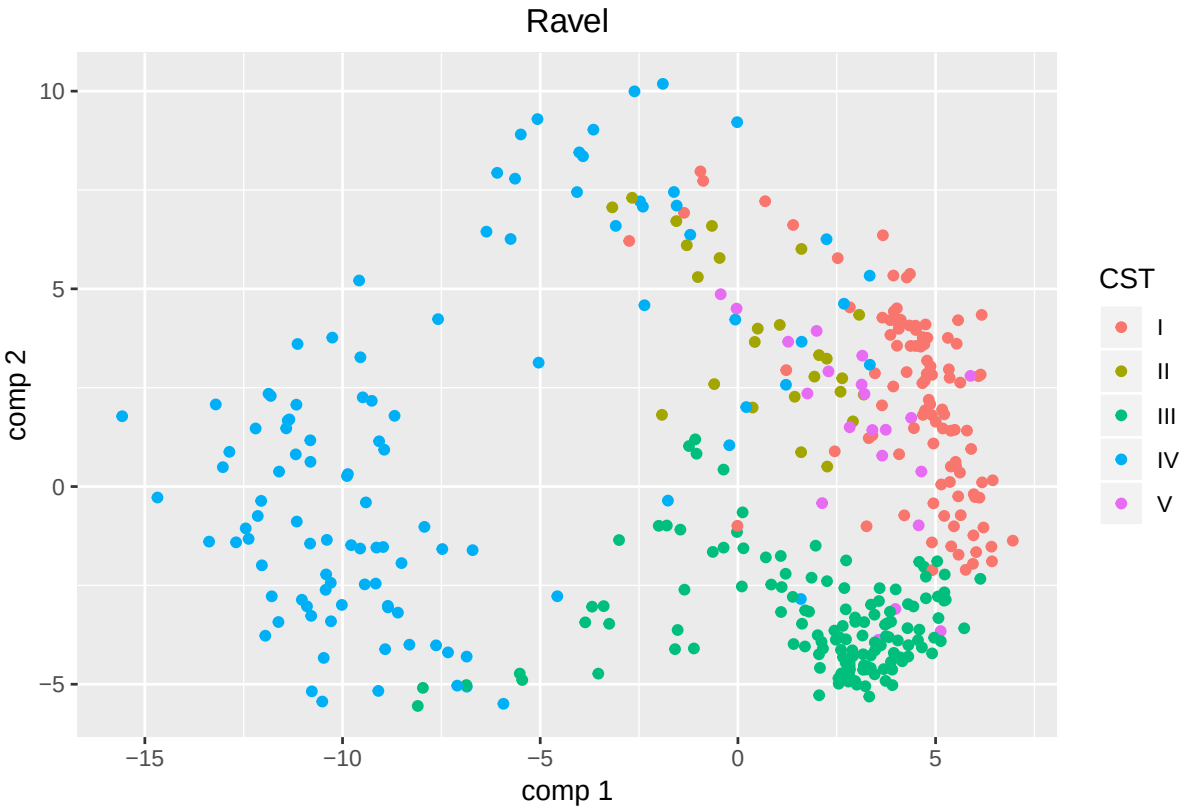
Merci infiniment à toute l'équipe de MaIAGE pour leur accueil. Cette ambiance chaleureuse m'a permis de m'intégrer facilement et de travailler sereinement. Merci à Mahendra et Magali pour leur hospitalité dès les premiers échanges, leur aide durant tout le stage ainsi que les compétences qu'ils ont eu à cœur de me transmettre.

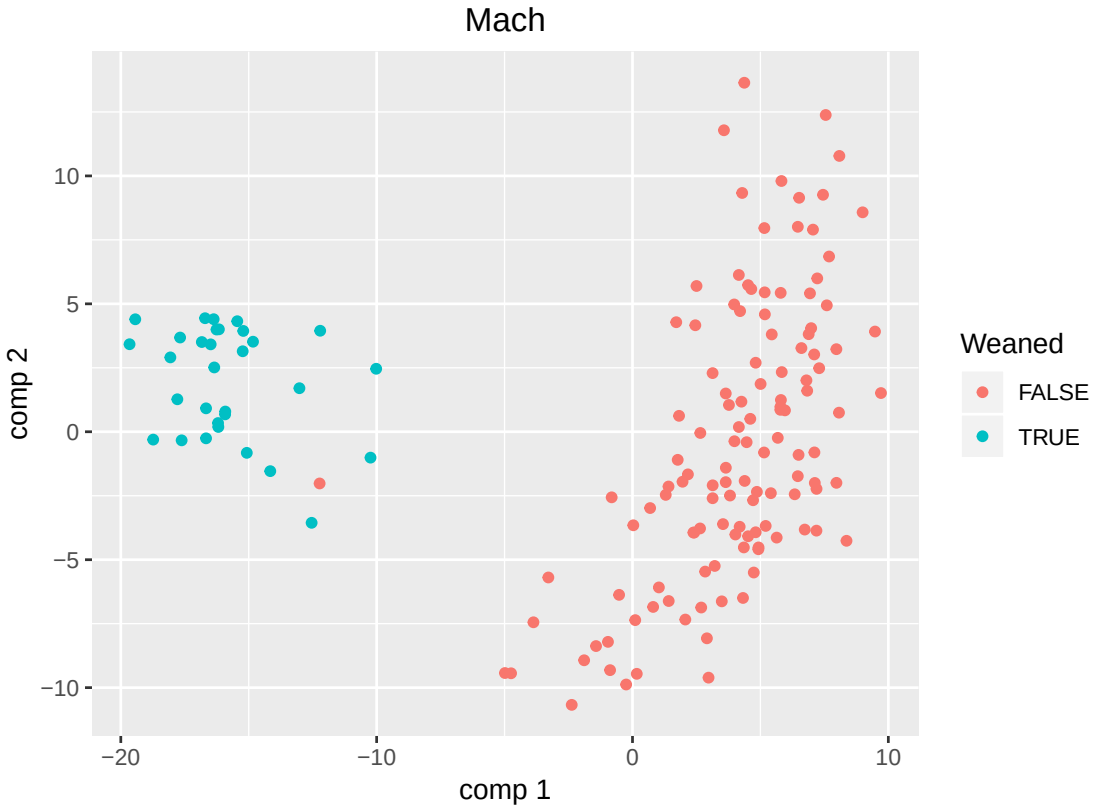
10 Annexe

Le code R programmé durant mon stage est disponible à l'adresse suivante : `https://github.com/ClemHard/metacoda`.

.1 Graphiques ACP







Dans le cas de Chaillou et de Mach, la séparation entre les différents groupes est franche. L'échantillon sevré de Mach mal situé provient d'un animal mal sevré, il n'est donc pas mal placé. Dans le cas de ravel, la séparation est moins franche.

.2 Classification non supervise

Chaillou

kmeans

comptage

	BoeufHache(obs)	Crevette(obs)	DesLardons(obs)	FiletCabillaud(obs)	FiletSaumon(obs)	SaucisseVolaille(obs)	SaumonFume(obs)	VeauHache(obs)
1(pred)	7	0	0	0	0	0	0	4
2(pred)	0	0	0	0	0	4	3	0
3(pred)	0	0	0	3	0	0	0	0
4(pred)	0	0	0	0	1	0	4	0
5(pred)	0	3	0	0	0	0	0	0
6(pred)	0	2	0	0	0	0	0	0
7(pred)	0	0	5	0	0	0	0	4
8(pred)	1	3	3	5	7	4	1	0

ilr

	BoeufHache(obs)	Crevette(obs)	DesLardons(obs)	FiletCabillaud(obs)	FiletSaumon(obs)	SaucisseVolaille(obs)	SaumonFume(obs)	VeauHache(obs)
1(pred)	0	0	0	8	0	0	0	0
2(pred)	8	0	0	0	0	0	0	0
3(pred)	0	0	0	0	0	0	0	8
4(pred)	0	0	0	0	0	8	0	0
5(pred)	0	0	0	0	8	0	0	0
6(pred)	0	8	0	0	0	0	0	0
7(pred)	0	0	8	0	0	0	0	0
8(pred)	0	0	0	0	0	0	8	0

hiérarchique

comptage

	BoeufHache(obs)	Crevette(obs)	DesLardons(obs)	FiletCabillaud(obs)	FiletSaumon(obs)	SaucisseVolaille(obs)	SaumonFume(obs)	VeauHache(obs)
1(pred)	7	0	2	0	0	0	0	5
2(pred)	1	3	1	5	8	4	1	0
3(pred)	0	3	0	0	0	0	0	0
4(pred)	0	2	0	0	0	0	0	0
5(pred)	0	0	5	0	0	0	0	3
6(pred)	0	0	0	3	0	0	0	0
7(pred)	0	0	0	0	0	4	3	0
8(pred)	0	0	0	0	0	0	4	0

ilr

	BoeufHache(obs)	Crevette(obs)	DesLardons(obs)	FiletCabillaud(obs)	FiletSaumon(obs)	SaucisseVolaille(obs)	SaumonFume(obs)	VeauHache(obs)
1(pred)	8	0	0	0	0	0	0	0
2(pred)	0	8	0	0	0	0	0	0
3(pred)	0	0	8	0	0	0	0	0
4(pred)	0	0	0	8	0	0	0	0
5(pred)	0	0	0	0	8	0	0	0
6(pred)	0	0	0	0	0	8	0	0
7(pred)	0	0	0	0	0	0	8	0
8(pred)	0	0	0	0	0	0	0	8

Mélange gaussien

comptage

	BoeufHache(obs)	Crevette(obs)	DesLardons(obs)	FiletCabillaud(obs)	FiletSaumon(obs)	SaucisseVolaille(obs)	SaumonFume(obs)	VeauHache(obs)
1(pred)	0	0	0	0	1	0	4	0
2(pred)	1	3	3	5	4	4	0	0
3(pred)	0	5	0	0	0	0	0	0
4(pred)	0	0	5	0	0	0	0	4
5(pred)	0	0	0	3	0	0	0	0
6(pred)	0	0	0	0	3	2	3	0
7(pred)	0	0	0	0	0	2	0	0
8(pred)	7	0	0	0	0	0	1	4

ilr

	BoeufHache(obs)	Crevette(obs)	DesLardons(obs)	FiletCabillaud(obs)	FiletSaumon(obs)	SaucisseVolaille(obs)	SaumonFume(obs)	VeauHache(obs)
1(pred)	8	0	0	0	0	0	0	0
2(pred)	0	8	0	0	0	0	0	0
3(pred)	0	0	8	0	0	0	0	0
4(pred)	0	0	0	8	0	0	0	0
5(pred)	0	0	0	0	8	0	0	0
6(pred)	0	0	0	0	0	8	0	0
7(pred)	0	0	0	0	0	0	8	0
8(pred)	0	0	0	0	0	0	0	8

Ravel

kmeans

comptage

	I(obs)	II(obs)	III(obs)	IV(obs)	V(obs)
1(pred)	0	0	0	0	19
2(pred)	86	0	0	0	0
3(pred)	0	20	0	0	0
4(pred)	17	5	9	108	2
5(pred)	2	0	126	0	0

ilr

	I(obs)	II(obs)	III(obs)	IV(obs)	V(obs)
1(pred)	100	0	8	2	1
2(pred)	0	23	0	8	15
3(pred)	0	0	5	74	0
4(pred)	0	0	119	3	5
5(pred)	5	2	3	21	0

hiérarchique

comptage

	I(obs)	II(obs)	III(obs)	IV(obs)	V(obs)
1(pred)	101	0	0	0	0
2(pred)	2	12	0	98	13
3(pred)	2	0	55	10	8
4(pred)	0	0	80	0	0
5(pred)	0	13	0	0	0

ilr

	I(obs)	II(obs)	III(obs)	IV(obs)	V(obs)
1(pred)	42	0	25	1	0
2(pred)	3	25	4	25	16
3(pred)	0	0	96	2	4
4(pred)	60	0	0	2	1
5(pred)	0	0	10	78	0

Mélange gaussien

comptage

	I(obs)	II(obs)	III(obs)	IV(obs)	V(obs)
1(pred)	83	0	0	0	0
2(pred)	2	2	1	58	0
3(pred)	1	0	1	0	16
4(pred)	12	23	130	50	4
5(pred)	7	0	3	0	1

ilr

	I(obs)	II(obs)	III(obs)	IV(obs)	V(obs)
1(pred)	21	2	4	0	3
2(pred)	32	20	47	107	11
3(pred)	0	0	17	0	1
4(pred)	25	3	32	0	3
5(pred)	27	0	35	1	3

Mach

kmeans

comptage

	FALSE(obs)	TRUE(obs)
1(pred)	88	0
2(pred)	36	31

ilr

	FALSE(obs)	TRUE(obs)
1(pred)	123	0
2(pred)	1	31

hiérarchique

comptage

	FALSE(obs)	TRUE(obs)
1(pred)	1	28
2(pred)	123	3

ilr

	FALSE(obs)	TRUE(obs)
1(pred)	1	31
2(pred)	123	0

Mélange gaussien

comptage

	FALSE(obs)	TRUE(obs)
1(pred)	18	25
2(pred)	106	6

ilr

	FALSE(obs)	TRUE(obs)
1(pred)	1	31
2(pred)	123	0

Sur les données ilr, les groupes sont plus distincts les uns des autres que pour les données de comptage (les classificateurs mettent ensemble les données des mêmes groupes). Excepté pour Ravel, aucune amélioration n'est visible en passant par les données ilr (voir moins bien que sur les données ilr).

.3 Classification supervise

Chaillou

random Forest

comptage

	BoeufHache(obs)	Crevette(obs)	DesLardons(obs)	FiletCabillaud(obs)	FiletSaumon(obs)	SaucisseVolaille(obs)	SaumonFume(obs)	VeauHache(obs)
BoeufHache(pred)	8	0	0	0	0	0	0	0
Crevette(pred)	0	8	0	0	0	0	0	0
DesLardons(pred)	0	0	8	0	0	0	0	0
FiletCabillaud(pred)	0	0	0	8	0	0	0	0
FiletSaumon(pred)	0	0	0	0	8	0	0	0
SaucisseVolaille(pred)	0	0	0	0	0	8	0	0
SaumonFume(pred)	0	0	0	0	0	0	8	0
VeauHache(pred)	0	0	0	0	0	0	0	8

ilr

	BoeufHache(obs)	Crevette(obs)	DesLardons(obs)	FiletCabillaud(obs)	FiletSaumon(obs)	SaucisseVolaille(obs)	SaumonFume(obs)	VeauHache(obs)
BoeufHache(pred)	6	0	0	0	0	0	1	1
Crevette(pred)	0	8	0	0	0	0	0	0
DesLardons(pred)	0	0	8	0	0	0	0	0
FiletCabillaud(pred)	0	0	0	8	0	0	0	0
FiletSaumon(pred)	0	0	0	0	8	0	0	0
SaucisseVolaille(pred)	0	0	0	0	0	8	0	0
SaumonFume(pred)	0	0	0	0	0	0	8	0
VeauHache(pred)	0	0	0	0	0	0	0	8

svm

comptage

	BoeufHache(obs)	Crevette(obs)	DesLardons(obs)	FiletCabillaud(obs)	FiletSaumon(obs)	SaucisseVolaille(obs)	SaumonFume(obs)	VeauHache(obs)
BoeufHache(pred)	0	0	0	0	0	0	0	0
Crevette(pred)	0	0	0	0	0	0	0	0
DesLardons(pred)	0	0	0	0	0	0	0	0
FiletCabillaud(pred)	0	0	0	0	0	0	0	0
FiletSaumon(pred)	0	0	0	0	0	0	0	0
SaucisseVolaille(pred)	0	0	0	0	0	0	0	0
SaumonFume(pred)	0	0	0	0	0	0	0	8
VeauHache(pred)	8	8	8	8	8	8	8	0

ilr

	BoeufHache(obs)	Crevette(obs)	DesLardons(obs)	FiletCabillaud(obs)	FiletSaumon(obs)	SaucisseVolaille(obs)	SaumonFume(obs)	VeauHache(obs)
BoeufHache(pred)	8	0	0	0	0	0	0	0
Crevette(pred)	0	8	0	0	0	0	0	0
DesLardons(pred)	0	0	8	0	0	0	0	0
FiletCabillaud(pred)	0	0	0	8	0	0	0	0
FiletSaumon(pred)	0	0	0	0	8	0	0	0
SaucisseVolaille(pred)	0	0	0	0	0	8	0	0
SaumonFume(pred)	0	0	0	0	0	0	8	0
VeauHache(pred)	0	0	0	0	0	0	0	8

kNN

comptage

	BoeufHache(obs)	Crevette(obs)	DesLardons(obs)	FiletCabillaud(obs)	FiletSaumon(obs)	SaucisseVolaille(obs)	SaumonFume(obs)	VeauHache(obs)
BoeufHache(pred)	6	0	1	0	0	0	1	2
Crevette(pred)	1	7	1	1	0	0	0	0
DesLardons(pred)	0	0	5	0	0	0	0	3
FiletCabillaud(pred)	0	1	0	7	0	0	0	0
FiletSaumon(pred)	0	0	0	0	7	1	0	0
SaucisseVolaille(pred)	0	0	0	0	0	4	0	0
SaumonFume(pred)	0	0	0	0	1	3	7	0
VeauHache(pred)	1	0	1	0	0	0	0	3

ilr

	BoeufHache(obs)	Crevette(obs)	DesLardons(obs)	FiletCabillaud(obs)	FiletSaumon(obs)	SaucisseVolaille(obs)	SaumonFume(obs)	VeauHache(obs)
BoeufHache(pred)	8	0	0	0	0	0	0	0
Crevette(pred)	0	8	0	0	0	0	0	0
DesLardons(pred)	0	0	8	0	0	0	0	0
FiletCabillaud(pred)	0	0	0	8	0	0	0	0
FiletSaumon(pred)	0	0	0	0	8	0	0	0
SaucisseVolaille(pred)	0	0	0	0	0	8	0	0
SaumonFume(pred)	0	0	0	0	0	0	8	0
VeauHache(pred)	0	0	0	0	0	0	0	8

Ravel

random Forest

comptage

	I(obs)	II(obs)	III(obs)	IV(obs)	V(obs)
I(pred)	103	0	0	2	0
II(pred)	0	22	0	3	0
III(pred)	1	0	129	5	0
IV(pred)	3	1	1	101	2
V(pred)	0	0	4	0	17

ilr

	I(obs)	II(obs)	III(obs)	IV(obs)	V(obs)
I(pred)	98	0	1	6	0
II(pred)	0	17	2	6	0
III(pred)	1	0	126	8	0
IV(pred)	2	1	2	101	2
V(pred)	0	1	6	1	13

svm

comptage

	I(obs)	II(obs)	III(obs)	IV(obs)	V(obs)
I(pred)	79	2	0	0	0
II(pred)	0	12	0	0	0
III(pred)	11	5	119	22	7
IV(pred)	15	6	16	86	3
V(pred)	0	0	0	0	11

ilr

	I(obs)	II(obs)	III(obs)	IV(obs)	V(obs)
I(pred)	89	1	4	3	1
II(pred)	0	16	0	0	0
III(pred)	1	1	112	1	10
IV(pred)	15	7	19	103	4
V(pred)	0	0	0	1	6

kNN

comptage

	I(obs)	II(obs)	III(obs)	IV(obs)	V(obs)
I(pred)	104	0	1	2	0
II(pred)	0	25	0	1	0
III(pred)	1	0	131	3	1
IV(pred)	0	0	2	102	0
V(pred)	0	0	1	0	20

ilr

	I(obs)	II(obs)	III(obs)	IV(obs)	V(obs)
I(pred)	101	0	3	5	1
II(pred)	0	24	0	4	1
III(pred)	3	0	125	8	5
IV(pred)	1	0	4	90	0
V(pred)	0	1	3	1	14

Mach

random Forest

comptage

	FALSE(obs)	TRUE(obs)
FALSE(pred)	123	1
TRUE(pred)	0	31

ilr

	FALSE(obs)	TRUE(obs)
FALSE(pred)	124	0
TRUE(pred)	0	31

svm

comptage

	FALSE(obs)	TRUE(obs)
FALSE(pred)	124	18
TRUE(pred)	0	13

ilr

	FALSE(obs)	TRUE(obs)
FALSE(pred)	123	0
TRUE(pred)	1	31

kNN

comptage

	FALSE(obs)	TRUE(obs)
FALSE(pred)	124	8
TRUE(pred)	0	23

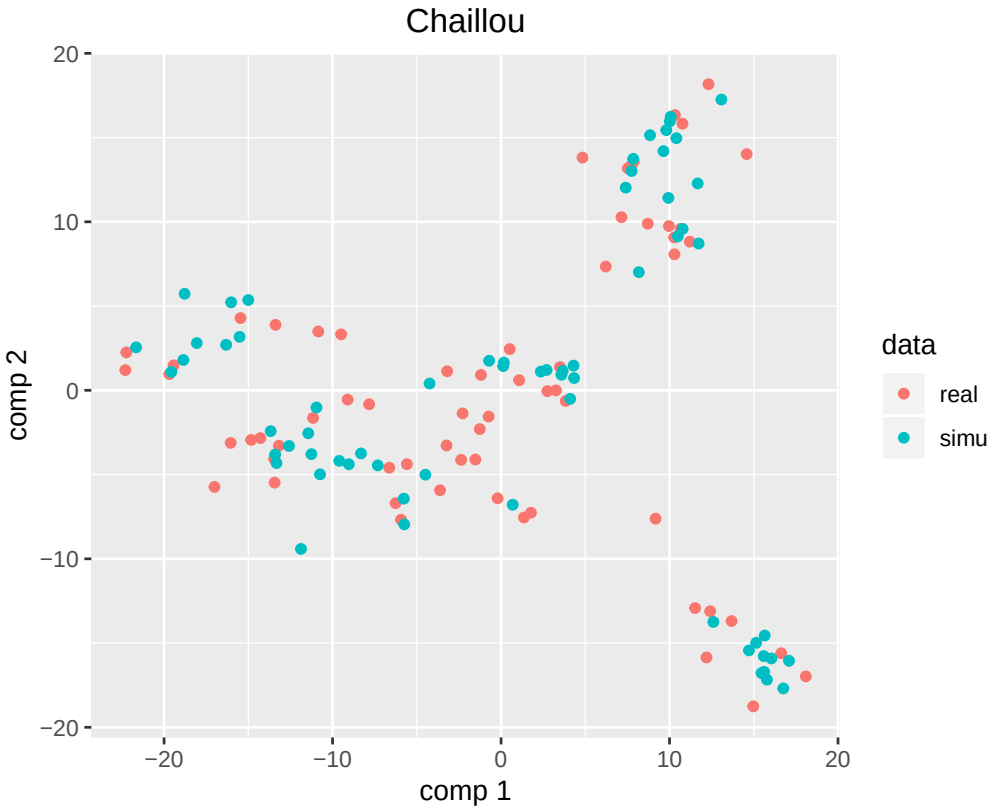
ilr

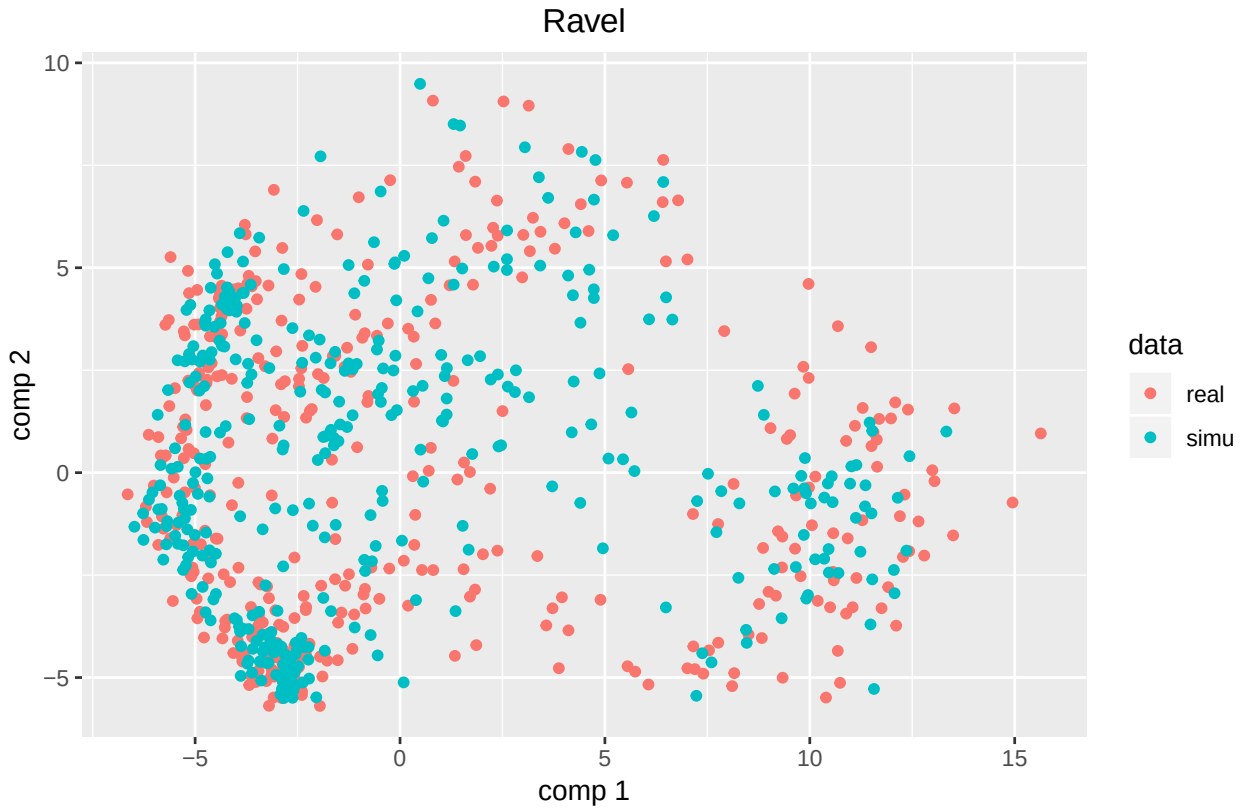
	FALSE(obs)	TRUE(obs)
FALSE(pred)	123	0
TRUE(pred)	1	31

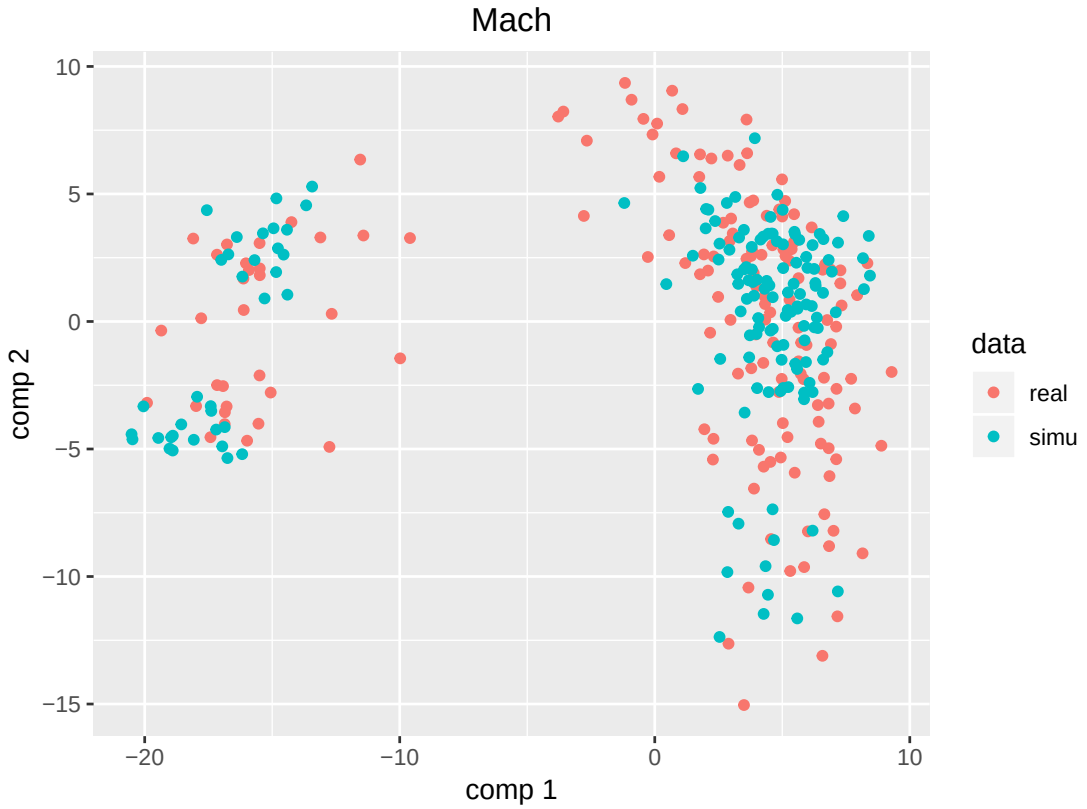
La classification supervisée améliore sensiblement les résultats sur le jeu de données Ravel, mais également sur les données de comptage (un classificateur semble mieux voir la différence entre les groupes qu'un classificateur non supervisé).

.4 Performance simulateur

Graphique







Pour Mach, les données simulées ne recouvrent pas l'ensemble des données réelles. Le simulateur semble pour ce jeu de données manquer d'un peu de variance.

Sur le jeu de données entier

Chaillou

Random forest

	real(obs)	simu(obs)
real(pred)	85	51.67
simu(pred)	15	48.33

kNN

	real(obs)	simu(obs)
real(pred)	45	43.34
simu(pred)	55	56.66

Ravel

Random forest

	real(obs)	simu(obs)
real(pred)	81.28	28.64
simu(pred)	18.72	71.36

kNN

	real(obs)	simu(obs)
real(pred)	54.87	43.68
simu(pred)	45.13	56.32

Mach

Random forest

	real(obs)	simu(obs)
real(pred)	96	24
simu(pred)	4	76

kNN

	real(obs)	simu(obs)
real(pred)	37.33	12.24
simu(pred)	62.67	87.76

Les résultats sont plutôt convaincants, pour les trois jeux de données une bonne partie des données simulées sont confondues avec des données réelles $(30 - 40\%)$.

Sur les groupes

Chaillou

Random forest

	BoeufHache(obs)	Crevette(obs)	DesLardons(obs)	FiletCabillaud(obs)	FiletSaumon(obs)	SaucisseVolaille(obs)	SaumonFume(obs)	VeauHache(obs)
BoeufHache(pred)	100	0	0	0.00	0	0.00	0	0
Crevette(pred)	0	100	0	0.00	0	0.00	0	0
DesLardons(pred)	0	0	100	0.00	0	0.00	0	0
FiletCabillaud(pred)	0	0	0	98.75	0	0.00	0	0
FiletSaumon(pred)	0	0	0	0.00	100	0.00	0	0
SaucisseVolaille(pred)	0	0	0	0.00	0	98.75	0	0
SaumonFume(pred)	0	0	0	1.25	0	1.25	100	0
VeauHache(pred)	0	0	0	0.00	0	0.00	0	100

kNN

	BoeufHache(obs)	Crevette(obs)	DesLardons(obs)	FiletCabillaud(obs)	FiletSaumon(obs)	SaucisseVolaille(obs)	SaumonFume(obs)	VeauHache(obs)
BoeufHache(pred)	92.50	1.25	0.00	0	0.00	1.25	0.00	7.50
Crevette(pred)	1.25	95.00	0.00	0	0.00	0.00	0.00	0.00
DesLardons(pred)	0.00	2.50	83.75	5	0.00	0.00	0.00	1.25
FiletCabillaud(pred)	0.00	0.00	0.00	90	0.00	0.00	0.00	0.00
FiletSaumon(pred)	0.00	0.00	0.00	0	98.75	0.00	0.00	0.00
SaucisseVolaille(pred)	0.00	0.00	0.00	0	0.00	68.75	3.75	0.00
SaumonFume(pred)	0.00	0.00	0.00	5	1.25	30.00	92.50	0.00
VeauHache(pred)	6.25	1.25	16.25	0	0.00	0.00	3.75	91.25

Ravel

Random forest

	I(obs)	II(obs)	III(obs)	IV(obs)	V(obs)
I(pred)	96.86	0.0	0.00	0.19	0.00
II(pred)	0.19	99.2	0.00	0.00	0.00
III(pred)	2.29	0.0	98.89	0.74	0.95
IV(pred)	0.00	0.8	0.30	99.07	0.00
V(pred)	0.67	0.0	0.81	0.00	99.05

kNN

	I(obs)	II(obs)	III(obs)	IV(obs)	V(obs)
I(pred)	95.33	0.0	0.15	1.11	0.00
II(pred)	0.29	98.8	0.07	0.00	0.00
III(pred)	2.48	0.0	97.56	3.89	0.95
IV(pred)	0.48	1.2	1.19	95.00	0.48
V(pred)	1.43	0.0	1.04	0.00	98.57

Mach

Random forest

	FALSE(obs)	TRUE(obs)
FALSE(pred)	100	0
TRUE(pred)	0	100

kNN

	FALSE(obs)	TRUE(obs)
FALSE(pred)	100	16.45
TRUE(pred)	0	83.55

Les bons résultats d'une classification supervisée se confirme sur ces jeux de données, le simulateur simule bien les différentes caractéristiques des groupes.

Sur les données simulées considérées comme réelles

Chaillou

Random forest

	BoeufHache(obs)	Crevette(obs)	DesLardons(obs)	FiletCabillaud(obs)	FiletSaumon(obs)	SaucisseVolaille(obs)	SaumonFume(obs)	VeauHache(obs)
BoeufHache(pred)	100	0.00	0.00	0.00	0	0.00	0	0
Crevette(pred)	0	98.22	0.00	0.00	0	0.00	0	0
DesLardons(pred)	0	0.00	99.84	0.00	0	0.00	0	0
FiletCabillaud(pred)	0	0.00	0.00	99.25	0	0.00	0	0
FiletSaumon(pred)	0	0.00	0.00	0.00	100	0.00	0	0
SaucisseVolaille(pred)	0	0.00	0.00	0.00	0	99.58	0	0
SaumonFume(pred)	0	1.78	0.16	0.75	0	0.42	100	0
VeauHache(pred)	0	0.00	0.00	0.00	0	0.00	0	100

kNN

	BoeufHache(obs)	Crevette(obs)	DesLardons(obs)	FiletCabillaud(obs)	FiletSaumon(obs)	SaucisseVolaille(obs)	SaumonFume(obs)	VeauHache(obs)
BoeufHache(pred)	91.79	0.25	0.00	0.00	0.00	0.00	0.00	5.19
Crevette(pred)	0.89	98.98	0.79	1.00	0.00	0.00	0.00	0.00
DesLardons(pred)	0.00	0.25	87.24	4.14	0.43	0.14	0.00	25.19
FiletCabillaud(pred)	0.00	0.25	0.00	92.10	0.00	0.00	0.00	0.00
FiletSaumon(pred)	0.89	0.00	0.00	0.00	90.75	4.80	3.33	0.00
SaucisseVolaille(pred)	0.36	0.25	0.79	0.50	6.24	80.65	0.00	0.00
SaumonFume(pred)	0.00	0.00	0.00	2.13	2.58	14.41	96.67	0.00
VeauHache(pred)	6.07	0.00	11.18	0.13	0.00	0.00	0.00	69.62

Ravel

Random forest

	I(obs)	II(obs)	III(obs)	IV(obs)	V(obs)
I(pred)	98.96	0.00	0.04	0.22	0.00
II(pred)	0.02	99.85	0.00	0.00	0.13
III(pred)	0.77	0.00	99.69	0.25	1.71
IV(pred)	0.02	0.15	0.01	99.53	0.00
V(pred)	0.23	0.00	0.26	0.00	98.16

kNN

	I(obs)	II(obs)	III(obs)	IV(obs)	V(obs)
I(pred)	97.57	0.0	0.08	1.99	0.00
II(pred)	0.30	99.9	0.00	0.00	0.39
III(pred)	1.23	0.0	99.12	2.98	1.18
IV(pred)	0.12	0.1	0.26	95.00	0.39
V(pred)	0.77	0.0	0.55	0.03	98.03

Mach

Random forest

	FALSE(obs)	TRUE(obs)
FALSE(pred)	100	0
TRUE(pred)	0	100

kNN

	FALSE(obs)	TRUE(obs)
FALSE(pred)	99.82	17.72
TRUE(pred)	0.18	82.28

Références

[1] J Aitchison. *The Statistical Analysis of Compositional Data.* Chapman & Hall, Ltd., London, UK, UK, 1986.

[2] Sylvain Arlot, Vincent Brault, Jean-Patrick Baudry, Cathy Maugis, and Bertrand Michel. *capushe : CAlibrating Penalities Using Slope HEuristics*, 2016. R package version 1.1.1.

[3] Stéphane Chaillou and Consortium ECOBIOPRO. Ecological inferences on the selection of core and variable components of bacterial communities associated with meat and seafood spoilage. 08 2014.

[4] Boris Jakuschkin, Virgil Fievet, Loïc SCHWALLER, Thomas Fort, Cécile Robin, and Corinne Vacher. Deciphering the pathobiome : intra- and interkingdom interactions involving the pathogen erysiphe alphitoides. *Microbial Ecology*, 72(4) :870–880, 2016.

[5] Nuria Mach, Mustapha Berri, Jordi Estellé, Levenez Florence, Gaëtan Lemonnier, Catherine Denis, Jean-Jacques Leplat, Claire Chevaleyre, Yvon Billon, Joel Dore, Claire Rogel-Gaillard, and Patricia Lepage. Early-life establishment of the swine gut microbiome and impact on host phenotypes : Role of early-life gut microbiome on pigs' health. 7, 03 2015.

[6] H Bjørn Nielsen, Mathieu Almeida, Agnieszka Sierakowska Juncker, Simon Rasmussen, Junhua Li, Shinichi Sunagawa, Damian R. Plichta, Laurent Gautier, Anders G. Pedersen, Emmanuelle Le Chatelier, Eric Pelletier, Ida Bonde, Trine Nielsen, Chaysavanh Manichanh, Manimozhiyan Arumugam, Jean-Michel Batto, Marcelo B. Quintanilha Dos Santos, Nikolaj Blom, Natalia Borruel, Kristoffer S. Burgdorf, Fouad Boumezbeur, Francesc Casellas, Joël Doré, Piotr Dworzynski, Francisco Guarner, Torben Hansen, Falk Hildebrand, Rolf S. Kaas, Sean Kennedy, Karsten Kristiansen, Jens Roat Kultima, Pierre Léonard, Florence Levenez, Ole Lund, Bouziane Moumen, Denis Le Paslier, Nicolas Pons, Oluf Pedersen, Edi Prifti, Junjie Qin, Jeroen Raes, Søren Sørensen, Julien Tap, Sebastian Tims, David W. Ussery, Takuji Yamada, MetaH. I. T Consortium , Pierre Renault, Thomas Sicheritz-Ponten, Peer Bork, Jun Wang, Søren Brunak, S Dusko Ehrlich, and MetaH. I. T Consortium . Identification and assembly of genomes and genetic elements in

complex metagenomic samples without using reference genomes. *Nat Biotechnol*, 32(8) :822–828, Aug 2014.

[7] Vera Pawlowsky-Glahn, Juan Jose Egozcue, and Raimon Tolosana-Delgado. Lecture notes on compositional data analysis. 01 2007.

[8] Nan Qin, Fengling Yang, Li Ang, Edi Prifti, Yanfei Chen, Li Shao, Jing Guo, Emmanuelle Le Chatelier, Jian Yao, Lingjiao Wu, Jiawei Zhou, Shujun Ni, Linda Liu, Nicolas Pons, Jean-Michel Batto, Sean P. Kennedy, Pierre Leonard, Chunhui Yuan, Wenchao Ding, and Lanjuan Li. Alterations of the human gut microbiome in liver cirrhosis. 07 2014.

[9] Jacques Ravel, Pawel Gajer, Zaid Abdo, G. Maria Schneider, Sara S. K. Koenig, Stacey L. McCulle, Shara Karlebach, Reshma Gorle, Jennifer Russell, Carol O. Tacket, Rebecca M. Brotman, Catherine C. Davis, Kevin Ault, Ligia Peralta, and Larry J. Forney. Vaginal microbiome of reproductive-age women. *Proceedings of the National Academy of Sciences*, 108(Supplement 1) :4680–4687, 2011.

[10] Sacha Schutz. Introduction à la métagénomique. http ://dridk.me/metagenomique.html, 2016. Accessed : 2018-08-10.

[11] M. E. Tipping and Christopher Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society, Series B*, 21/3 :611–622, January 1999.