



**Rapport de stage de 5.A :**

**Présenté par :**

**ABOUABDALLAH Mohamed Anwar**

**5A. Mathématiques appliquées et Modélisation 2018-2019**

## **Identification d'OTU par clustering et par modèle S.B.M.**

**Lieu de stage**



Institut Nationale de Recherche en Agronomie  
Unité MIAT

**Encadré par :**

Nathalie Peyrard, Unité **MIAT INRA Toulouse**, [nathalie.peyrard@inra.fr](mailto:nathalie.peyrard@inra.fr)

Alain Franc, Unité **BioGeCo INRA et Pleiade INRIA Bordeaux** [alain.franc@inra.fr](mailto:alain.franc@inra.fr)

Olivier Coulaud, équipe **Hiepacs, INRIA Bordeaux** [olivier.coulaud@inria.fr](mailto:olivier.coulaud@inria.fr) ,

**Unité Miat, INRA Centre de recherche Occitanie-Toulouse**

**Chemin de Borde-Rouge Auzeville CS 52627**

**31326 Castanet-Tolosan cedex**

**Tél. : 05 61 28 50 28**

<http://www.toulouse.inra.fr/>

**- "Time is very slow for those who wait, very fast for those who are scared, very long for those who lament, lery short for those who celebrate. But for those who love, time is eternal ." William Shakespeare.**

## Résumé

### Identification d'OTU par clustering et par modèle S.B.M.

Ce stage, a plusieurs finalités, voici une petite présentation :

- Tout d'abord, je dois m'approprier le cadre SBM et l'algorithme VEM utilisé classiquement pour l'estimation du modèle.
  - Ensuite, je mettrai en place un pipeline complet de traitement depuis la matrice des distances d'alignement entre séquences jusqu'à l'identification des groupes et de leurs liens.
  - Ensuite le pipeline sera mis en œuvre sur des données de biodiversité de la forêt guyanaise pour des données dont nous disposons de toute l'information taxonomique (ordre, espèces, genres, familles).
  - Enfin, je m'intéresserai à la comparaison la classification via SBM et une autre méthode d'apprentissage non supervisé qui est la classification hiérarchique. Pour cela, on prendra un même jeu de donnée, portant sur des arbres guyannais sur lequel, nous allons faire des classifications afin de retrouver les différent taxons appartenant aux différents niveaux taxonomiques. Puis on va calculer les tables de contingence afin de comparer entre nos deux méthodes et la réalité botanique.
- 

## Abstract

### OTU identification by clustering and by model S.B.M.

This internship, has several aims, here is a short presentation :

- First of all, I need to use the SBM framework and the VEM algorithm conventionally used for estimating the model.
- Next, I'll set up a complete processing pipeline from the Sequence Alignment Distance Matrix to identifying groups and their links.
- Then the pipeline will be implemented on biodiversity data of the Guyanese forest for data of which we have all the taxonomic information (order, species, genera, families).
- Finally, I will be interested in comparing the classification via SBM and another unsupervised learning method which is the hierarchical classification. For this, we will take the same set of data, on Guyanese trees on which, we will make classifications to find the different taxa belonging to different taxonomic levels. Then we will calculate the contingency tables in order to compare our two methods and the botanical reality.

## Remerciement

**M**on stage s'inscrit dans un sous domaine des mathématiques, les statistiques appliquées l'écologie, domaine que j'ai découvert durant mon premier stage. Ceci a fait que plusieurs membres de laboratoire de recherche m'ont aidé et permis de me familiariser avec le sujet et de réussir mon stage.

Ainsi, avant de commencer ce rapport, je tiens d'abord à rendre grâce et témoigner toute ma reconnaissance aux différentes personnes qui ont contribué au bon déroulement de mon stage.

Pour cela, je commencerais par adresser toute ma gratitude à mes trois responsables stage Nathalie Peyrard, Alain Franc et Olivier Coulaud, tout d'abord pour m'avoir accepté en stage, ensuite pour leur patience, leur disponibilité et surtout leurs judicieux conseils, qui ont contribué à améliorer mes compétences et surmonter les différents obstacles mathématiques et informatiques que j'ai pu rencontrer..

Ensuite, je tiens à remercier Sylvain Jasson, mon directeur d'unité ainsi que tous les membres de mon équipe et de mon unité, tout d'abord pour leur accueil, sympathie et les diverses discussions qu'on a eu durant les pauses.

Enfin je tiens aussi à remercier les membres de l'équipe BioGeCo de l'inra Bordeaux à la fois pour leur accueil et pour les balades après le déjeuner.

# Sommaire

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Présentation du laboratoire d'accueil . . . . .	3
1.1.1	Présentation l'INRA . . . . .	3
1.1.2	Présentation de l'unité . . . . .	3
1.2	Contexte du stage et problématique scientifique . . . . .	3
1.3	objectifs . . . . .	4
<b>2</b>	<b>Le problème d'identification d'espèces</b>	<b>6</b>
2.1	Calcul de la matrice des distances . . . . .	6
2.1.1	Calcul de la matrice des distances . . . . .	6
2.2	Apprentissage non supervisé . . . . .	6
2.3	Classification hiérarchique . . . . .	7
2.3.1	Éléments de vocabulaire : . . . . .	7
2.3.2	Stratégies d'agrégation sur dissimilarité . . . . .	8
2.3.3	Fonctionnement de la classification hiérarchique . . . . .	9
2.4	Jeu de données . . . . .	9
<b>3</b>	<b>Modèles à blocs stochastiques</b>	<b>10</b>
3.1	Modèles de mélanges . . . . .	10
3.2	Intuition du modèle . . . . .	10
3.3	Modèles à blocs stochastiques . . . . .	11
3.3.1	Notion de graphe . . . . .	11
3.3.2	Approche présence/absence . . . . .	12
3.3.3	Approche des poids . . . . .	13
<b>4</b>	<b>Méthode d'estimation des paramètres d'un S.B.M.</b>	<b>14</b>
4.1	Complexité du calcul exact . . . . .	14
4.1.1	Calcul de la vraisemblance . . . . .	14
4.2	Algorithme EM . . . . .	15
4.3	Algorithme V.E.M. . . . .	17
4.3.1	Reformulation E.M. . . . .	17
4.3.2	Itération d'un V.E.M. . . . .	18
4.3.3	Pseudo-code : . . . . .	18
4.4	Critère de selection du nombre de classes : . . . . .	18
4.4.1	Objective : . . . . .	19
4.4.2	Critère I.C.L. : . . . . .	19
<b>5</b>	<b>Librairie développée</b>	<b>20</b>
5.1	Dépendances . . . . .	20
5.2	Logique d'implémentation . . . . .	20
5.2.1	Entrées . . . . .	20
5.2.2	Présentation des principales fonctions . . . . .	21

<b>6</b>	<b>Résultats</b>	<b>22</b>
6.1	L'apport du modèle de distance par rapport au binaire . . . . .	22
6.1.1	Résultats obtenus : . . . . .	22
6.1.2	Explication de notre choix . . . . .	23
6.2	Phase d'exploration . . . . .	23
6.3	Exemple concret d'utilisation de la librairie . . . . .	27
6.4	La similarité et dissimilarité entre la CHA et SBM distance pour 2 ou 3 classes . . . . .	27
6.4.1	Similarité entre la CHA et SBM distance pour 2 ou 3 classes . . . . .	27
6.4.2	L'apport du modèle de distance par rapport au binaire pour plus de 3 classes . . .	30
	Myristiceae : . . . . .	31
	Annonaceae : . . . . .	32
<b>7</b>	<b>Perspectives et conclusion</b>	<b>34</b>
7.1	Perspectives : . . . . .	34
7.2	Conclusion . . . . .	34
<b>8</b>	<b>Références.</b>	<b>36</b>

# 1 Introduction

## 1.1 Présentation du laboratoire d'accueil

### 1.1.1 Présentation l'INRA

J'ai effectué mon stage à Toulouse au sein de l'unité mathématiques et informatique appliquées de Toulouse abrégée Mia du centre de recherche de l'I.N.R.A., un organisme français de recherche agronomique fondé en 1946. Cet institut a le statut d'Établissement public à caractère scientifique et technologique et est sous la double tutelle du ministère chargé de la Recherche et du ministère chargé de l'Agriculture.

Cet organisme est considéré comme le premier institut de recherche agronomique en Europe et le second au monde en nombre de publications en sciences agricoles et en sciences de la plante et de l'animal. L'I.N.R.A. mène des recherches afin d'améliorer la qualité de l'alimentation, ainsi que pour une agriculture durable et pour un environnement préservé et valorisé [7] [9].

Enfin, cet institut est composé de vingt centres régionaux comme le centre de Toulouse, dont les recherches s'organisent autour de trois pôles de compétences : le pôle production horticole intégrée, le pôle adaptation au changement global et le pôle santé des plantes.

De plus par ces chiffres, l'I.N.R.A. s'impose comme un acteur majeur de la recherche agronomique dans le monde par la qualité de ses productions scientifiques. L'Institut est présent dans le top 1% des institutions les plus citées au monde dans 15 des 22 champs disciplinaires.

### 1.1.2 Présentation de l'unité

L'Unité de Mathématiques et Informatique Appliquées de Toulouse MIAT est une unité propre du département MIA de l'INRA. Comme toutes les unités de MIA, elle a pour mission scientifique de développer et mettre en œuvre des méthodes mathématiques et/ou informatiques pertinentes pour résoudre des problèmes identifiés avec des collaborateurs qui sont issus principalement d'autres départements de l'INRA.

Enfin, cette unité se divise en cinq équipes : deux équipes de recherche MAD et SaAB, et trois équipes de service GENOTOU, RECORD et SIGENAE.

En ce qui me concerne, j'ai effectué mon stage dans l'équipe MAD qui développe entre autre, des modèles et des méthodes d'apprentissage pour des applications en écologie.

## 1.2 Contexte du stage et problématique scientifique

La biodiversité, un mot composé des mots "bio", du grec « vie » et diversité qui désigne la diversité de la vie sur la Terre. Il existe plusieurs types de biodiversité biologique [8], nous nous intéresserons à la caractérisation de la diversité taxonomique via la diversité moléculaire des organismes.

Bien qu'elle soit classiquement caractérisée par un ensemble d'indices (comme celui de Shannon [28] [10]) qui sont des informations globales qui ne décrivent pas l'organisation des espèces entre elles. Ces indices, conçus en un temps où la diversité était principalement décrite à partir d'observations des traits phénotypiques<sup>1</sup> des espèces durant une période où le frein était la production des données peu nombreuses, mais longues à collectionner.

Et parallèlement à cela, avec les phylogénies moléculaires, c'est à dire l'utilisation de séquences d'ADN

---

1. variations d'un caractère par exemple la couleur des yeux.

afin d'obtenir des informations sur l'histoire évolutive des organismes vivants, il est possible d'envisager une caractérisation plus riche de la biodiversité.

En effet, la notion de biodiversité peut se refléter dans un pattern de dissimilarités entre séquences, quantifiées par des distances d'alignement. L'un des enjeux actuels est l'identification à partir de ces données, les groupes de séquences correspondant à une même espèce, puis le pattern d'organisation des espèces entre elles.

L'élément de base est l'OTU (Operationnal Taxonomic Unit) qui est une classe regroupant des séquences proches, et candidates à regrouper des séquences. Un enjeu majeur est de construire un dictionnaire entre les OTUs et les espèces. Par ce fait, la construction d'OTUs [5] [27] se fait par clustering de séquences adn selon les similarité.

Des premiers travaux de mes encadrants [12] ont permis de représenter les séquences sous la forme d'un nuage de points en petite dimension et d'en extraire des zones de fortes densités de points. L'étape suivante consiste à revenir à l'espace de grande dimension initial et à analyser ces zones dans cet espace. Cela pose plusieurs questions :

Une zone correspond-elle vraiment à une seule espèce, ou découvre-t-on des sous-groupes ? Les "espèces" sont-elles connectées ou isolées ? Comment se positionnent les zones les unes par rapport aux autres ?

Afin de répondre à ces questions, nous commencerons par dire qu'il est envisageable de se ramener à une représentation du nuage de points par un graph en se fixant un seuil, et où un sommet représente une séquence, et un lien existe si la distance est inférieure au seuil. Ensuite nous analyserons la structure de ce graphe pour en déduire les OTUs et une description de la biodiversité associée. Nous travaillerons aussi sur le tableau de distance pour simplifier ce graphe.

Mon stage porte sur l'analyse de ces graphes basée sur l'utilisation des modèles à blocs stochastiques (Stochastic Block Model, SBM) qui permettent non seulement d'identifier des groupes de sommets connectés mais aussi de caractériser les interactions entre ces groupes.

On se ramène ainsi à un problème d'apprentissage non supervisé qui est forme de Clustering. Elle consiste à diviser un groupe hétérogène de données en sous-groupes, de manière à ce que les données considérées comme les plus similaires soient associées au sein d'un même groupe (homogène) et qu'au contraire les données considérées comme différentes se retrouvent dans d'autres groupes distincts. Ceci sera effectué sans qu'au préalable la nature des classes ait été choisie ; il s'agit donc d'une classification non supervisée (en aveugle) des données.

Dans le cadre de cette classification "aveugle", il en existe une infinité de méthodes. Pour les modèles SBM, la plus utilisé est la variante VEM de l'algorithme E.M.

### 1.3 objectifs

Ainsi, après avoir défini le contexte de ce stage et son intérêt scientifique, je vais maintenant présenter les finalités de mon stage :

- Tout d'abord, je dois m'approprier le cadre SBM et l'algorithme VEM utilisé classiquement pour l'estimation du modèle.
- Ensuite, je mettrai en place un pipeline complet de traitement depuis la matrice des distances d'alignement entre séquences jusqu'à l'identification des groupes et de leurs liens.
- Ensuite le pipeline sera mis en œuvre sur des données de biodiversité de la forêt guyanaise pour des données dont nous disposons de toute l'information taxonomique (ordre, espèces, genres, familles).



- Enfin, je m'intéresserai à la comparaison la classification via SBM et une autre méthode d'apprentissage non supervisé qui est la classification hiérarchique. Pour cela, on prendra un même jeu de donnée, portant sur des arbres guyannais sur lequel, nous allons faire des classifications afin de retrouver les différent taxons appartenant aux différents niveaux taxonomiques. Puis on va calculer les tables de contingence afin de comparer entre nos deux méthodes et la réalité botanique.

## 2 Le problème d'identification d'espèces

L'exploration de la biodiversité est centrée autour des collections et des descriptions naturalistes d'organismes essentiellement animaux, végétaux et fongiques [?, 6]. Cette exploration est construite à partir de la variabilité de l'empreinte moléculaire de l'évolution que l'on peut lire dans l'A.D.N. [3] notamment via les phylogénies moléculaires [1].

### 2.1 Calcul de la matrice des distances

#### 2.1.1 Calcul de la matrice des distances

Avant de présenter les résultats, nous allons commencer par expliquer comment sont obtenus les données dont on dispose.

##### Échantillon de metabarcoding

```

AGCG 0001 1000
AGCG 0002 1000
AGCG 0003 1000
AGCG 0004 1000
AGCG 0005 1000
AGCG 0006 1000
AGCG 0007 1000
AGCG 0008 1000
AGCG 0009 1000
AGCG 0010 1000
AGCG 0011 1000
AGCG 0012 1000
AGCG 0013 1000
AGCG 0014 1000
AGCG 0015 1000
AGCG 0016 1000
AGCG 0017 1000
AGCG 0018 1000
AGCG 0019 1000
AGCG 0020 1000
AGCG 0021 1000
AGCG 0022 1000
AGCG 0023 1000
AGCG 0024 1000
AGCG 0025 1000
AGCG 0026 1000
AGCG 0027 1000
AGCG 0028 1000
AGCG 0029 1000
AGCG 0030 1000
AGCG 0031 1000
AGCG 0032 1000
AGCG 0033 1000
AGCG 0034 1000
AGCG 0035 1000
AGCG 0036 1000
AGCG 0037 1000
AGCG 0038 1000
AGCG 0039 1000
AGCG 0040 1000
AGCG 0041 1000
AGCG 0042 1000
AGCG 0043 1000
AGCG 0044 1000
AGCG 0045 1000
AGCG 0046 1000
AGCG 0047 1000
AGCG 0048 1000
AGCG 0049 1000
AGCG 0050 1000
AGCG 0051 1000
AGCG 0052 1000
AGCG 0053 1000
AGCG 0054 1000
AGCG 0055 1000
AGCG 0056 1000
AGCG 0057 1000
AGCG 0058 1000
AGCG 0059 1000
AGCG 0060 1000
AGCG 0061 1000
AGCG 0062 1000
AGCG 0063 1000
AGCG 0064 1000
AGCG 0065 1000
AGCG 0066 1000
AGCG 0067 1000
AGCG 0068 1000
AGCG 0069 1000
AGCG 0070 1000
AGCG 0071 1000
AGCG 0072 1000
AGCG 0073 1000
AGCG 0074 1000
AGCG 0075 1000
AGCG 0076 1000
AGCG 0077 1000
AGCG 0078 1000
AGCG 0079 1000
AGCG 0080 1000
AGCG 0081 1000
AGCG 0082 1000
AGCG 0083 1000
AGCG 0084 1000
AGCG 0085 1000
AGCG 0086 1000
AGCG 0087 1000
AGCG 0088 1000
AGCG 0089 1000
AGCG 0090 1000
AGCG 0091 1000
AGCG 0092 1000
AGCG 0093 1000
AGCG 0094 1000
AGCG 0095 1000
AGCG 0096 1000
AGCG 0097 1000
AGCG 0098 1000
AGCG 0099 1000
AGCG 0100 1000

```



##### Matrice de distance

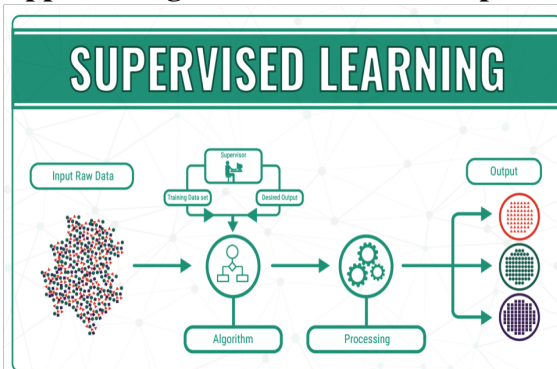
$$\begin{pmatrix} 0 & 2 & \dots & 3 \\ \vdots & 0 & \ddots & 11 \\ \vdots & \ddots & \ddots & 0 \\ 3 & \dots & \dots & 0 \end{pmatrix}$$

Cette matrice est calculée de cette manière [4] : On mesure le nombre de différences entre les séquences alléliques des individus deux à deux et on met le nombre de différence dans cette matrice [23].

### 2.2 Apprentissage non supervisé

Pour classer ces espèces on procède par identification via apprentissage.

#### • Apprentissage supervisé

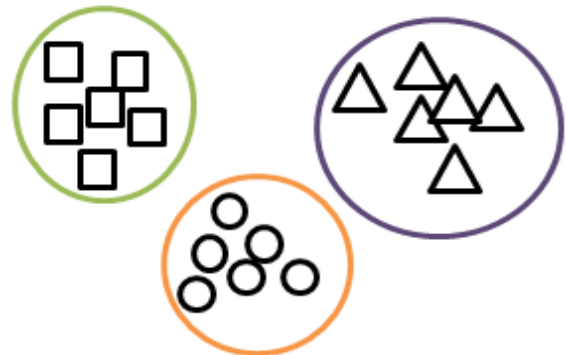


source : <http://bigdata-madesimple.com>

L'apprentissage supervisé consiste à "mapper" (de se positionner dans le génome) ces séquences sur un génome de référence<sup>a</sup>. Malheureusement ces bases ne sont pas toujours complètes.

<sup>a</sup>. Un génome ou base de référence est une base de données numérique de séquences d'acides nucléiques, constituant un exemple représentatif de l'ensemble des gènes d'une espèce.

#### • Apprentissage non supervisé



source : <https://fr.wikipedia.org/wiki/>

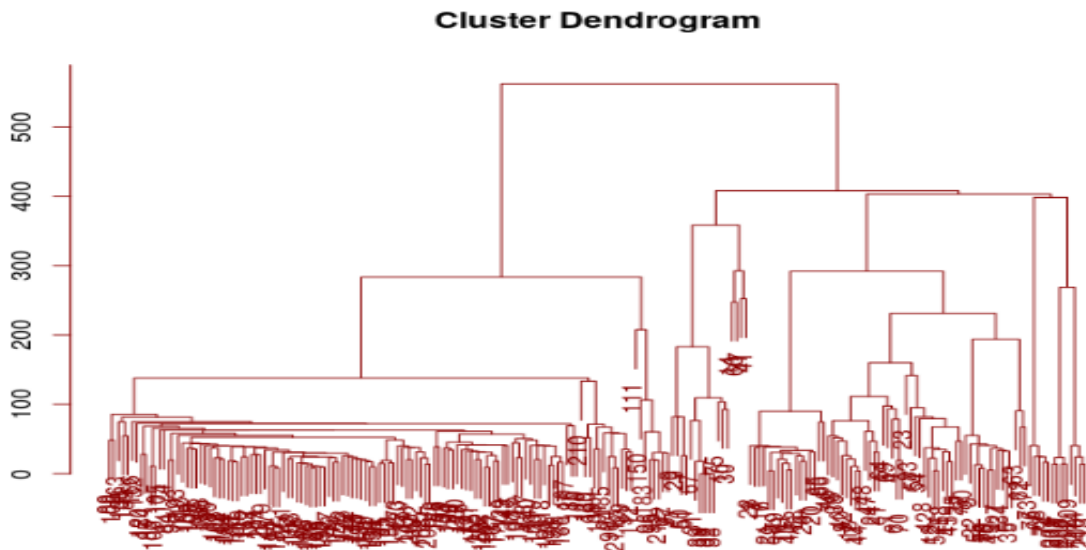
Dans le cas où les bases ne sont pas complètes, on est amené à utiliser des méthodes plus intuitives ce qui nous amène à mobiliser un apprentissage non supervisé (les plus utilisées sont le clustering bayésien et hiérarchique). Et dans ce cas, on souhaite faire une construction d'OTUs, qui sont considérés en biologie et en écologie tels des présomptions d'espèces. [23]

Et donc, dans ce stage, je serais amené à procéder par un apprentissage non supervisé pour la construction d'OTUs.

## 2.3 Classification hiérarchique

Parmi les méthodes d'identification d'OTUs existantes, on trouve la classification hiérarchique ascendante [2] (*abrégiée C.A.H.*) est très employée quand on souhaite faire un regroupement d'une manière automatique.

En effet, en dépit du fait que la C.A.H. soit couteuse en temps de calculs, son plus grand atout est qu'elle permet de donner un arbre de hiérarchie des classes.



### 2.3.1 Éléments de vocabulaire :

Soit  $E$  un ensemble de  $n$  éléments qu'on souhaite subdiviser en  $k$  classes,

#### Similarité et dissimilarité :

##### Similarité [17] :

**Définition 1** Soit  $s$  la fonction de similarité :

- $\forall i, j = 1, \dots, ns(i, j) = s(j, i)$
- $s(i, j) \geq s(i, j) \geq 0$

Dans le cadre de ce travail, nous allons utiliser la dissimilarité qui représente la distances génétique entre les séquences de deux individus.

##### Dissimilarité [17] :

**Définition 2** Soit  $d$  la fonction de dissimilarité :

- $\forall i, j = 1, \dots, nd(i, j) = d(j, i)$
- $0 \leq s(i, i) \leq s(i, j)$

#### Inertie intra et inter classes :

Pour  $n$  points dans un espace euclidien  $G = \{e_i : i = \{1 : n\}\}$ , on appel distance euclidienne entre deux points  $X_i$  et  $X_{i'}$  :

$$d^2(i, i') = d^2(X_i, X_{i'})$$

Pour une partition à  $k$  classes d'effectifs  $P_i$  :

- $G_1, G_2 \dots G_k$  leurs centres de gravité.
- $I_1, I_2 \dots I_k$  leurs inerties.

On appelle l'inertie intra-classes :

$$I_w = \frac{1}{n} \sum_{i=1}^k \sum_{j=1}^{n_j} d_{e_j, G_i}^2$$

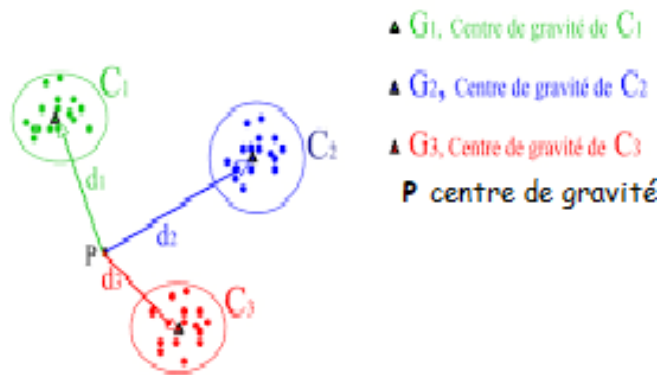
On appelle l'inertie inter-classes :

$$I_B = \frac{1}{n} \sum_{i=1}^k d_{G_i, G}^2$$

Enfin, on appelle l'inertie totale :

$$I = I_B + I_w$$

**Exemple pour 3 classes :**



Ainsi, pour comparer entre deux partitions à  $k$  classes, on choisit celle qui minimise  $I_w$  (ou maximise  $I_B$ ).

### 2.3.2 Stratégies d'agrégation sur dissimilarité

#### Single-Linkage et complete-Linkage

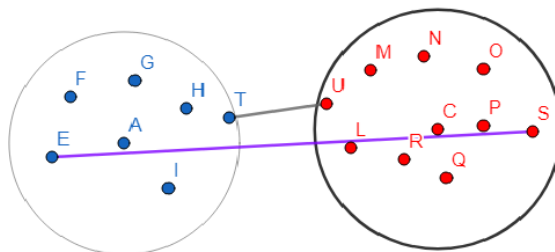
La stratégie du Single-Linkage ou saut minimum consiste à faire en sorte que la dissimilarité entre deux classes soit la plus petite distance entre les éléments des deux classes c'est à dire :

$$d(a-b, c) = \inf\{d(a, c); d(b, c)\}$$

La stratégie Complete-Linkage ou diamètre consiste à faire en sorte que la dissimilarité entre parties soit la plus petite distance entre les éléments des deux parties c'est à dire :

$$d(a-b, c) = \sup\{d(a, c); d(b, c)\}$$

Ci dessous, on montre un exemple de plusieurs points au sein de deux clusters.



Dans cet exemple le single-linkage consisterait à dire que  $d=[JK]$  et le complete-linkage  $d=[ES]$

### Algorithme de Ward

La méthode de ward [29] consiste à regrouper les classes tout en veillant à maximiser l'inertie inter-classes. C'est à dire que pour un groupe de  $n$  individus, on initialise le nombre de classes à  $n$ . Ensuite, à chaque itération, on calcul l'inertie interclasse puis on réduit le nombre de classes de 1 en regroupant les classes de sorte que l'inertie interclasses soit maximale.

### 2.3.3 Fonctionnement de la classification hiérarchique

## 2.4 Jeu de données

Les données sur lesquels nous avons travaillé concernent 1502 arbres d'une parcelle expérimentale en Guyane Française à la foche de Saint-Flic. Nous disposons de la nomenclature botanique de chacun des individus (ordre, espèce, genre, famille) et du tableau de distances 1502 x 1502 entre les marqueurs aléatoires de ces individus.



### 3 Modèles à blocs stochastiques

#### 3.1 Modèles de mélanges

Avant de présenter les modèles S.B.M. [14], nous allons commencer par présenter les modèles de mélanges, un cadre important pour les problèmes de classification non supervisée surtout quand on est amené à vouloir maximiser la vraisemblance d'un modèle.

Ainsi, contrairement aux autres méthodes connues (Kmeans, Classification hiérarchiques ...), il s'agit d'une approche probabiliste de la classification dont l'un des principaux atouts est que la partition résultante peut être interprétée de manière statistique.

Une distribution mélange (ou loi) est une fonction de densité (*cas continu*) ou distribution de probabilité (*cas discret*) issue de la combinaison convexe de plusieurs fonctions de distributions.

**Définition 3** Soit une variable aléatoire  $X$  et un ensemble de paramètres  $(\theta_1, \dots, \theta_K)$  et un vecteur de scalaires  $(\alpha_1, \dots, \alpha_K) \in [0, 1]^K$  vérifiant

$$\sum_{i=1}^K \alpha_i = 1$$

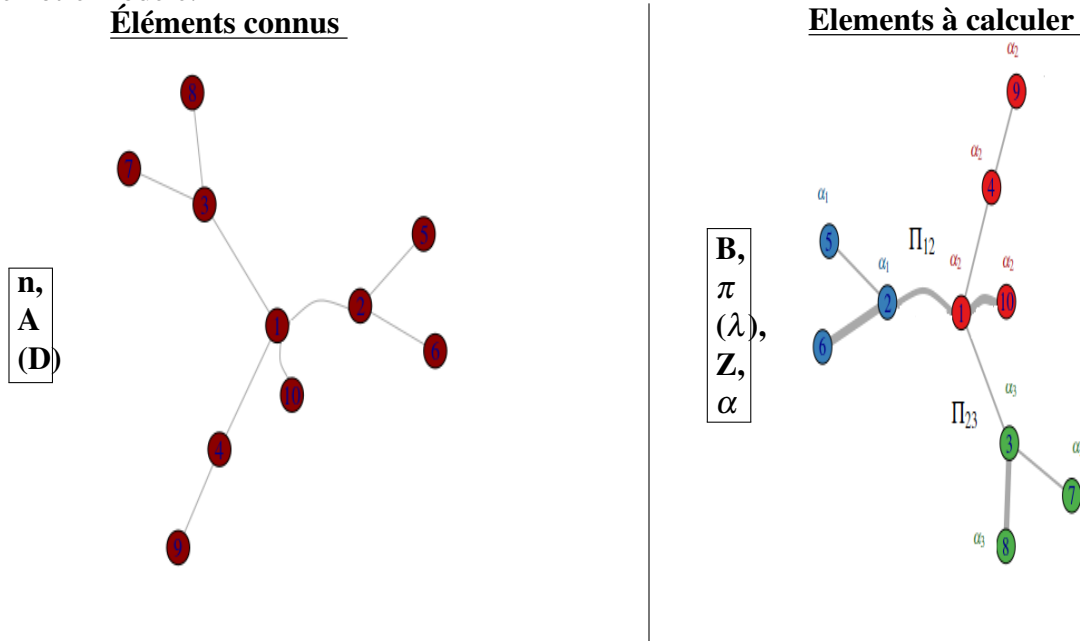
. Alors, une variable aléatoire  $X$  suit une loi de mélange à  $K$  composante si la loi de  $X$  s'écrit :

$$P(X = x) = \sum_{i=1}^k \pi_i P_{\theta_i}(X = x)$$

avec  $P_{\theta_i}$  est une loi de probabilité

#### 3.2 Intuition du modèle

On se donne un jeu de données à  $n$  séquences de brins d'ADN, on calcul les différences entre les  $n$  individus deux à deux pour trouver la matrice des distances. On trace le graphe puis on le met en entrée de notre modèle.



Ainsi, notre méthode prend un graphique et permet de donner les classes des sommets de ce graphe. Dans l'exemple ci-dessus, nous avons trois classes (rouge (4 éléments), vert et bleu (3 éléments chacun)).

### 3.3 Modèles à blocs stochastiques

#### 3.3.1 Notion de graphe


En mathématiques, on définit un graphe comme une structure composée d'objets : Dans cette structure certaines paires d'objets sont liées ou connectées.

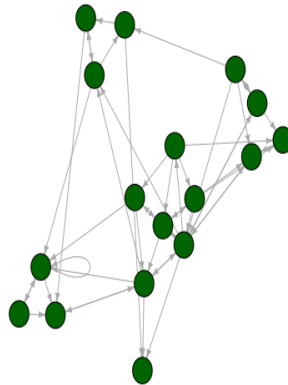
Les objets correspondent à des abstractions mathématiques sont appelés sommets(ou nœuds), et les liens entre sommets sont des arêtes.

Ci-dessous nous allons donner une définition plus formelle des graphes :

**Définition 4** Un graphe  $G = (V, E)$  fini est défini par :

- $V = (v_1, \dots, v_n)$  qui représente l'ensemble fini des sommets de  $G$ .
- $E \in V \times V$  représente l'ensemble fini des arêtes de  $G$ .

Voici un exemple simple de graphe simulé via de la librairie igraph de  :



**Définition 5** Soit  $G = (V, E)$  un graphe à  $n$  sommet, sa matrice d'adjacence  $A \in M_n(\{0, 1\})$  et une matrice symétrique, dont l'élément non diagonal  $a_{ij}$  indique la présence d'une arête liant le sommet  $i$  au sommet  $j$ .

$$a_{ij} = \begin{cases} 1 & \text{si } (i, j) \in E \\ 0 & \text{sinon} \end{cases}$$

### 3.3.2 Approche présence/absence

Présentation de notre modèle :

$$\boxed{A, n, B, Z, \pi, \alpha}^2$$

- $A \in \mathbb{M}_n(\{0, 1\})$  est une matrice représentant la connectivité des sommets c'est à dire la matrice d'adjacence. Ainsi  $A_{i,j} = \mathbb{1}(i \leftrightarrow j)$ .
- $n \in \mathbb{N}$ , représente le nombre de d'individus.
- $B \in \mathbb{N}$ , représente le nombre de blocs (classes).
- On définit aussi la variable cachée  $Z \in M_{n,B}([0, 1])$  telle que :

$$\begin{cases} Z_{i,q} = \mathbb{1}(i \in q) \\ \sum_{q=1}^B Z_{i,q} = 1 \end{cases}$$

- Ensuite nous définissons  $\alpha \in [0, 1]^B$  telle cela :

$$\alpha_q = P(Z_{i,q} = 1)$$

Elle représente la probabilité d'appartenance à une classe.

- Nous définissons ensuite  $\Pi = \{\pi_{b,l}\}_{1 \leq b,l \leq B}$  comme cela :  $\pi_{b,l}$  la probabilité qu'un sommet de classe  $i$  soit connecté avec un sommet de classe  $j$ . Elle est définie de cette manière :
- Dans notre problème, la probabilité  $P(A)$  est issue **des modèles de mélanges binomiaux** et donc sa distribution serait donnée par :

$$P(A, Z = z) = P_\pi(A|z)P_\alpha(z)$$

$$\Rightarrow$$

$$P(A, Z = z) = \prod_{i \in V} \prod_{j \neq i} P_\pi(A_{i,j}|z_i, z_j) \prod_{l=1}^n P_\alpha(z_l)$$

$$\Rightarrow$$

$$\boxed{P(A, Z = z) = \prod_{i \in V} \prod_{j \neq i} (\sum_{b=1}^K \sum_{l=1}^K z_{i,b} z_{j,l} \Pi_{b,l}^{A_{i,j}} (1 - \pi_{j,l})^{(1-A_{i,j})}) \prod_{b=1}^B \alpha_b^{z_{i,b}}}$$

### Calcul de la vraisemblance des paramètres

À partir de cela, nous pouvons calculer la vraisemblance du modèle :

$$P_\theta(A) = \sum_z P(A, z)$$

$$\Rightarrow$$

$$\boxed{P_\theta(A) = \sum_z (\prod_{i \in V} \prod_{j \neq i} (\sum_{b=1}^K \sum_{l=1}^K z_{i,b} z_{j,l} \Pi_{b,l}^{A_{i,j}} (1 - \pi_{j,l})^{(1-A_{i,j})}) \prod_{b=1}^B \alpha_b^{z_{i,b}})}$$

2. Les variables en rouges sont des variables cachées, c'est à dire qu'elle ne sont là que pour définir les classes des variables et ne sont pas connues.



### 3.3.3 Approche des poids

#### Présentation du modèle

$$D, n, B, Z, \lambda, \alpha$$

Le second modèle serait similaire au premier avec comme seule différence la matrice A qui serait une matrice de distance  $A_2 \in \mathbb{M}_{n,B}(\mathbb{N})$

- $D \in \mathbb{M}_n(\mathbb{N})$  est une matrice des distances. Ainsi  $D_{i,j}$  représente la distance d'éloignement entre les sommets.
- Nous définissons ensuite  $\lambda$  tel suit :  $\lambda_{l,b}$  la probabilité qu'il y ait une distance d entre un sommet de classe b et un sommet de classe b. Cette probabilité est issue **d'un modèle de mélanges de poisson** dont la distribution serait donnée par :

$$P(A, Z = z) = \prod_{i \in V} \prod_{j \neq i} \left( \sum_{b=1}^K \sum_{l=1}^K z_{i,b} z_{j,l} \prod_{b,l}^{A_{i,j}} \frac{\lambda_{b,b'}^k \exp^{-\lambda_{b,b'}}}{k!} \right) \prod_{i=1}^n \prod_{b=1}^B \alpha_b^{z_{i,b}}$$

- Les autres termes sont définis de manière analogue au premier modèle.

#### Calcul de la vraisemblance des paramètres

Ainsi, de manière analogue nous allons calculer la loi du graphe défini par A, donc vraisemblance des paramètres :

$$P_{\theta}(A) = \sum_z \left( \prod_{i \in V} \prod_{j \neq i} \left( \sum_{b=1}^K \sum_{l=1}^K z_{i,b} z_{j,l} \prod_{b,l}^{A_{i,j}} \frac{\lambda_{b,b'}^k \exp^{-\lambda_{b,b'}}}{k!} \right) \prod_{i=1}^n \prod_{b=1}^B \alpha_b^{z_{i,b}} \right)$$

## 4 Méthode d'estimation des paramètres d'un S.B.M.

### 4.1 Complexité du calcul exact

Pour l'instant, nous supposons B connu, la méthode classique pour estimer  $\theta$  est le calcul de l'estimateur du maximum de vraisemblance MLE.

#### 4.1.1 Calcul de la vraisemblance

##### Complexité du calcul

On peut remarquer que notre premier somme est une première somme de  $B^n$ , ensuite un produit de "autant de case non nulle" de la matrice d'adjacence puis  $\sum_{i=1}^B \sum_{j=1}^B A_{i,j} + 3$  et un  $\sum_1^n \sum_1^b z_{i,b}$   
Ceci nous donnera un nombre de calcul maximal de :

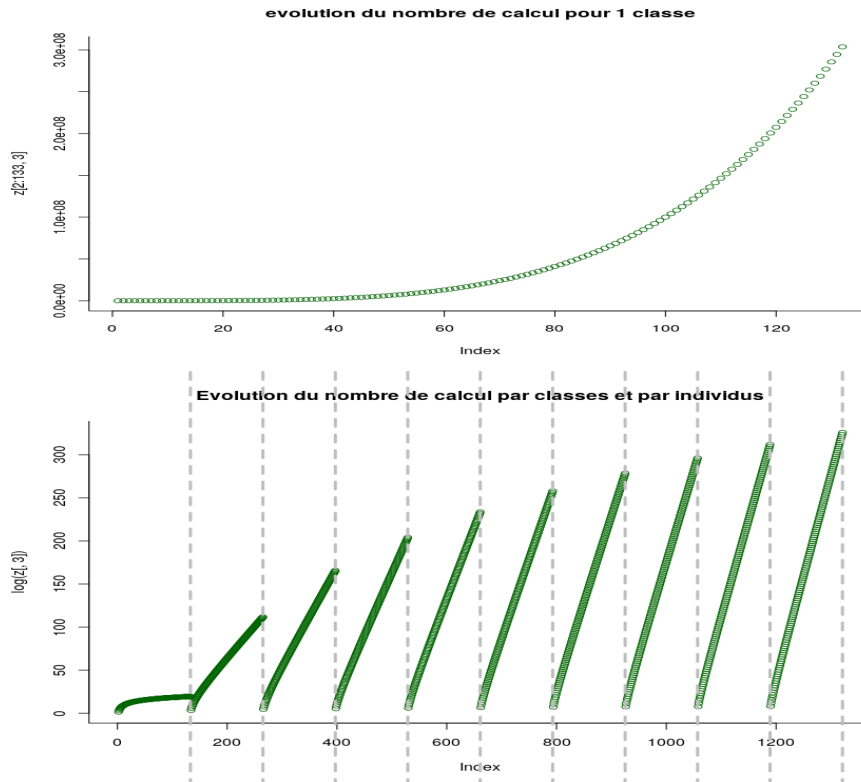
$$\sum_{l=1}^{B^n} \sum_{k=1}^{n^2} \left( \sum_{i=1}^B \sum_{j=1}^B (4) \right) + \sum_1^n \sum_1^n B$$

$$= \sum_{l=1}^{B^n} \sum_{k=1}^n (4(B)^2 + B^2 n) = (4 + n)n^2 B^{n+3}$$

$$\text{nombre de calculs} = (4 + n)n^2 B^{n+3} \simeq B^{n+3}$$

##### Exemples :

Pour n=132 et B=10 ce nombre vaut  $3 \times 10^{141}$



★ **Conséquence :** Face à cette difficulté, nous allons utiliser l'algorithme E.M. pour notre approximation.

Afin d'estimer les paramètres d'un modèle avec des données cachées ou manquantes, on utilise des méthodes itératives de résolution approchée.

Parmi ses membres on peut trouver :

- **Approche bayésien et simulation de Monte-Carlo [21]** : est une méthode qui conserve les valeurs minimales et maximales des probabilités à évaluer et qui évoluent de manière itératives. Après la convergence de cette méthode [13], il est possible d'estimer les probabilités recherchées en choisissant des valeurs dans les intervalles finaux de manière cohérente. Cette méthode est trop coûteuse en terme de temps de simulation.
- **Expectation-maximization** : Est un algorithme très célèbre créé par Dempster, Laird et Rubin en 1977. Cet algorithme constitue la brique de base dans ce stage car c'est grâce à l'une de ses variantes qu'on va effectuer toutes nos estimations.

## 4.2 Algorithme EM

Nous commençons par désigner l'ensemble des paramètres du modèle par  $\theta = (\alpha, \pi)$  ( $\theta_{\text{dist}} = (\alpha_{\text{dist}}, \pi_{\text{dist}})$  pour le modèle basé sur les distances). L'objectif de l'algorithme E.M. [15] est de déterminer les paramètres qui maximisent la vraisemblance de  $P_{\theta}(A)$  quand ni cette expression ni son logarithme ne peuvent être maximisés par les méthodes de type MLE classiques.

Soit notre jeu de données  $\{A, Z\}$  qu'on ne connaît pas totalement, on connaît seulement les valeurs des  $A_{i,j}$ , sa vraisemblance est donnée par  $p_{\theta}(A)$  calculée.

Pour cela, on construit le postérieur  $P_{\theta}(A|Z)$ , puis par son biais on peut calculer la vraisemblance du set complet :

$$P(A, Z|\theta) = P_{\theta}(A|Z)P_{\theta}(Z), \text{ car on ne sait pas calculer } P(A)$$

On suppose aussi qu'on connaît une estimation  $\theta_t$  pour  $\theta$  ce qui nous permet de calculer le postérieur  $p(Z|A, \theta_t)$ . À partir de notre estimation  $\theta_t$ , nous essayerons d'améliorer la valeur de  $\theta$  afin d'avoir  $p(A|\theta) > p(A|\theta_t)$ .

Voici un pseudo-code possible pour cet algorithme et expliciter le nombre de calcul derrière chaque itération c'est à dire une étape de maximisation et d'expectation :

---

### Algorithm 1 Algorithme EM

---

**Require:**  $X, \varepsilon$

$\theta^t \leftarrow \theta^0$ , ou  $\theta^0$  est une affectation aléatoire.

**E-step :**

$$Q(\theta|A, \theta^t) = E_{Z|A, \theta^t}[\log p(A, Z|\theta)]$$

**M-step :**

$$\theta^{(t+1)} \leftarrow \operatorname{argmax}_{\theta} Q(\theta|A, \theta^t)$$

**if**  $\|\theta^t - \theta^{(t+1)}\| > \varepsilon$  **then**

$$\theta^t \leftarrow \theta^{(t+1)}$$

**end if**

---

On commence par expliciter les termes de chaque itération :

**E-step :**  $Q(\theta|A, \theta^t) = E_{Z|A, \theta^t}[\log p(A, Z|\theta)]$

Nous expliciterons la fonction Q dans le cadre du modèle SBM :

$$\begin{aligned}
 Q(\alpha, \pi | \alpha^t, \pi^t) &= E_{Z,A | \alpha^t, \pi^t} [\log p(A, Z | \alpha, \pi)] \\
 &= E_{Z,A | \alpha^t, \pi^t} [\log(p(Z | A, \alpha, \pi) P(A | \alpha, \pi))] \\
 &= E_{Z,A | \alpha^t, \pi^t} [\log(p(Z | A, \alpha, \pi)) \log(P(A | \alpha, \pi))] \\
 &= E_{Z,A | \alpha^t, \pi^t} [\log(p(Z | A, \alpha, \pi))] E_{Z,A | \alpha^t, \pi^t} [\log(P(A | \alpha, \pi))] \\
 &= \log(P(A | \alpha, \pi)) + \sum_z \log(P(Z = z | \alpha, \pi)) \log(P(Z = z | \alpha^t, \pi^t))
 \end{aligned}$$

Ainsi, pour une itération données :

$$\text{E-détailée : } Q(\theta | A, \theta^t) = \log(P(X = x | \alpha, \pi)) + \sum_z \log(P(Z = z | \alpha, \pi)) \log(P(Z = z | A, \pi^t))$$

Maintenant on va calculer la complexité de cette étape. Pour cela, on remonte au calcul de la complexité de la vraisemblance fait auparavant :

$$\text{nombre de calculs/vraisemblance} = B^{n+1} (4B + n^2) * n^2$$

$\Rightarrow$

$$\text{nombre de calculs total} = \text{nombre de calculs/vraisemblance} + \sum_z 2 \text{nombre de calculs/vraisemblance}$$

<sup>3</sup> Puis

$$\text{nombre de calculs total} = \text{nombre de calculs/vraisemblance} + \sum_i^{B^n} 2 \text{nombre de calculs/vraisemblance}$$

$\Rightarrow$

$$\text{nombre de calculs total} = B^{n+1} (4B + n^2) * n^2 + 2B^n B^{n+1} (4B + n^2) * n^2$$

$\Rightarrow$

$$\text{Nombre de calculs E-step : } = B^{n+1} (4B + n^2) * n^2 (1 + 2B^n) = O(B)$$

En  $n = +\infty$ , on a  $B^{n+1} (4B + n^2) * n^2 (1 + 2B^n) \sim B^{n+1} x 4B x 2B^n = O(8B^{2(n+1)})$

Ainsi notre complexité du E-step est équivalente à cette formule :

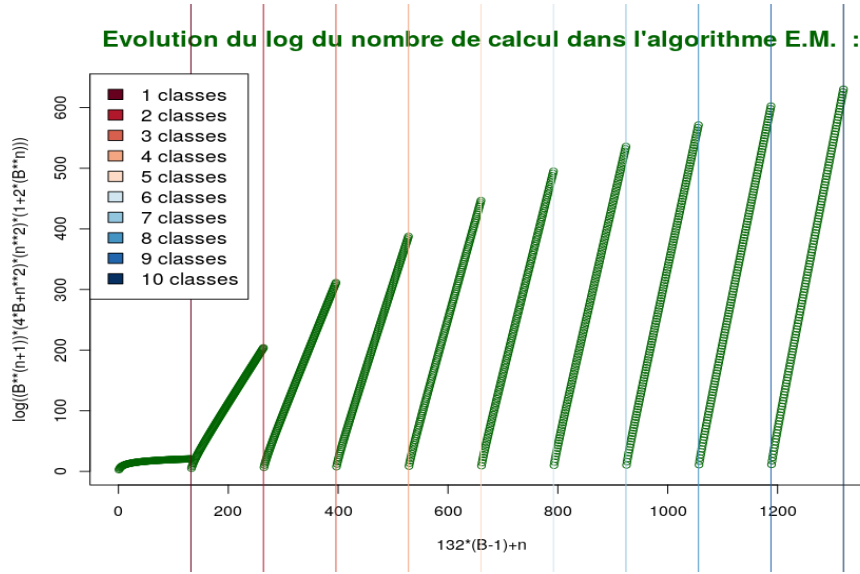
$$\text{Complexité } B^{2(n+1)}.$$

On peut remarquer qu'on est en présence d'un problème de classe EXPTIME.

Nous ferons varier le nombre de blocs et d'individu respectivement entre (1 :10) et (1 :132) puis nous allons le tracer nombre de calculs nécessaire :

---

3. ici on va ignorer le cout mais on se concentrera seulement sur le nombre de calcul



On peut remarquer que le nombre de calculs croît exponentiellement. À titre d'exemple, pour  $n=132$  et  $B=10$ , on doit réaliser  $10^{600}$  calculs. Cette complexité justifie amplement notre passage à l'algorithme V.E.M.

### 4.3 Algorithme V.E.M.

#### 4.3.1 Reformulation E.M.

Comme on l'a expliqué dans la section précédente, les estimations via l'algorithme E.M. restent très difficile ce qui nous conduit à utiliser des méthodes variationnelles pour approcher l'étape E du EM. Cela conduit à l'algorithme Variationnel EM [20].

Il repose sur une écriture particulière des deux étapes du EM comme deux étapes de maximisation d'une même fonction.

Commençons par la réécriture du E-step :

$Q(\theta|A, \theta^t) = E_{Z|A, \theta}[\log p(A, Z|\theta)]$  Ensuite on se donne la fonction suivante, où  $\alpha$  est une distribution sur les états de  $Z$

$$\begin{aligned}
 F(Q, \theta^t) &= E_Q[\log P_{\theta^t}(Z, A)] - \sum_z Q(z) \log Q(z) \\
 &= E_Q[\log P_{\theta^{t-1}}(A) P_{\theta^{t-1}}(Z|A)] - \sum_z Q(z) \log(Q(z)) \text{ Puis à partir de la propriété du log :} \\
 &\quad \log(ab) = \log(a) + \log(b) \text{ et la linéarité de l'espérance on obtient ;}
 \end{aligned}$$

$$F(Q, \theta^t) = E_Q[\log P_{\theta^{t-1}}(A)] + E_Q[P_{\theta^{t-1}}(Z|A)] - \sum_z Q(z) \log(Q(z))$$

**Définition 6 (Divergence Kullback Leilder. :)**

$$\begin{aligned}
 KL(Q|P) &= \sum_z Q(z) \log\left(\frac{Q(z)}{P(z)}\right) \\
 &= E_Q\left[\log\left(\frac{Q(z)}{P(z)}\right)\right] = E_Q[\log(Q(z))] - E_Q[\log(P(z))]
 \end{aligned}$$

On voit que :  $F(Q, \theta^t) = \log(P_{\theta^{t-1}}(A) - KL(Q|P_{\theta^{t-1}}(Z|A)))$  Maximisons  $F(Q, \theta^t)$  selon  $Q$  :

$$Q^t = \operatorname{argmax}_Q (F(Q, \theta^t)) = \operatorname{argmax}_Q (KL(Q|P_{\theta^t}(Z|A)))$$

$$\Leftrightarrow$$

$$Q^t = \operatorname{argmax}_Q (F(Q, \theta^t)) = [Q^t, \text{verifiant } Q^t = P_{\theta^t}(Z|A)]$$

Ainsi calculer  $\operatorname{argmax}_Q (f(Q, \theta^t))$  et calculer  $Q(\theta|\theta^t)$  se ramènent tous les deux à calculer  $P_{\theta^t}(Z|A)$ .

Ainsi, **VE-step :  $Q^t = \operatorname{argmax}_Q (KL(Q|P_{\theta^{t-1}}(Z|A)))$**

avec  $Q_{\text{ind}} = \{Q, \text{tel que}, Q(z) : \prod_{i=1}^n Q_i(z_i)\}$

Ensuite, on va faire de même pour la M-step :

$$\textbf{M-step : } \theta^t = \operatorname{argmax}_{\theta} F(Q^t|\theta)$$

#### 4.3.2 Itération d'un V.E.M.

On restreint  $Q$  à :  $Q(z) : \prod_{i=1}^n Q_i(z_i)$ , c'est à dire on impose l'indépendance des  $z_i$ . Puis, étant donné qu'il est difficile d'approximer le maximum de la fonction log-vraisemblance car on se retrouve avec plusieurs maxima locaux. On va donc lancer le VEM avec  $k^4$  initialisations  $((\alpha, \pi)^0, \dots, (\alpha, \pi)^k)$  ce qui nous donnera  $k$  résultats  $((\hat{\alpha}^0, \hat{\pi}^0)^0, \dots, ((\hat{\alpha}^0, \hat{\pi}^0)^k)$  puis on choisit le résultat qui maximise la log-vraisemblance approchée.

#### 4.3.3 Pseudo-code :

---

**Algorithm 2** Algorithme V.E.M. :

---

**Require:**  $X, p(Z|X, \theta), \varepsilon$

1:  $\theta_i \leftarrow \theta_0$ , ou  $\theta_0$  est une affectation aléatoire de valeurs.

2: **Variationnel E-step :**

3:  $Q^t = \operatorname{argmax}_Q (KL(Q|P_{\theta^{t-1}}(Z|A)))$

4: **M-step :**

5:  $(\alpha^{t+1}, \pi^{t+1}) \leftarrow \operatorname{argmax}_{\alpha, \theta} Q(\alpha, \theta|\alpha^t, \pi^t)$

---

#### 4.4 Critère de selection du nombre de classes :

Considérons maintenant le cas où  $B$  n'est pas connu, choisir  $B$  revient à choisir un modèle de SBM. Dans la littérature, plusieurs critères de sélection de modèle existent, si les critères AIC et BIC sont les plus utilisées en statistiques et notamment en régression. Si ces critères s'avèrent utiles pour la plus part des problèmes de classification, d'autres critères existent, certains consistent à optimiser un critère empirique pénalisé, tel que la log-vraisemblance pénalisée ce qui conduit à des problèmes de minimisation ou de contrôle optimale.

Dans le cadre de ce stage, je vais utiliser le critère ICL que je vais définir en dessous car il est plus adapté aux modèles de classification (ref à mettre Biernanacki, Govaert ...2000).

---

4. choisi arbitrairement

#### 4.4.1 Objective :

On commence cette section ar présenter la formule qui nous donne la vraisemblance complète.

$$\log(L(A, Z)) = \sum_i^n \sum_q^B Z_{i,q} \log(\alpha_q) + \frac{1}{2} \sum_{i \neq j} \sum_{q,l} Z_{i,q} Z_{j,l} \log(b(A_{j,j}, \pi_{q,l}))$$

Avec  $b(x, \pi) = \pi^x (1 - \pi)^{1-x}$

Ensuite, comme nos  $x_{i,j}$  ne sont pas marginalement indépendant des  $z_{i,j}$ , le calcul du maximum de vraisemblance devient difficile, ce qui nous oblige à utiliser des methodes variationnelle en plus de l'algorithme E.M. (Algo V.E.M.).

#### 4.4.2 Critère I.C.L. :

Enfin nous définissons notre critère de sélection du modèle qui découle d'une injection de la formule de Stirling dans le critère B.I.C. ce qui nous donne le critère Integration Classification likelihood abrégé ICL et qui est donné par la formule :

$$ICL(B) = \log(P(A, \tilde{Z}^B | \hat{\theta}^B, B)) - \frac{\vartheta_B}{2} \log(n)$$

- Avec  $B$  le nombre de classes,
- $\vartheta_B$  le nombre de paramètres d'un modèle à  $B$  classes,
- $\hat{\theta}^B$  l'estimateur obtenu par VEM pour un modèle à  $B$  classes
- et  $\tilde{Z}^B$  le mode de  $Q(Z)$  obtenu à la dernière itération du VEM pour un modèle à  $B$  classes.

Dans la suite on prendra le modèle avec le meilleur ICL.

## 5 Librairie développée

Durant mon stage, j'ai été amené à développer une librairie qui permet de prendre les distances entre séquences pour faire des classifications via un clustering hiérarchique et une classification sbm pour donner les différentes partitions et pouvoir les comparer à des taxons appartenant à un niveau taxonomique. Voici un lien pour le répertoire github de cette librairie : <https://github.com/mawro3301/explorationauto><sup>5</sup>


### 5.1 Dépendances

Concernant les dépendances de la librairie développées durant mon stage, on utilise plusieurs librairies :

♣ **Classification via des sbm** : La librairie blockmodels permet de faire notre classification SBM. Elle a des dépendances avec digest, Rcpp et parallel qui permettent de paralléliser les calculs.

- **Rcpp [16]** : Fournit des fonctions R ainsi que des classes C ++ offrant une intégration de R et C ++.
- **parallel [25]** : Permet de paralléliser l'exécution des gros morceaux de code.
- **digest [31]** : Nécessaire au fonctionnement de Rcpp et parallel
- **blockmodels [19]** : Estimation par modèle de bloc stochastique et latent par un algorithme EM variationnel.

♣ Tracé des graphes, histogrammes et plots :

- **ggplot2 [30]** : Permet de faire une belle visualisation de données en utilisant la grammaire des graphiques
- **ggthemes** : Procure à l'utilisateur plusieurs thèmes, géométries et échelles, elle nécessite l'utilisation de ggplot2
- **stats** : Librairie de base en 
- **ggraph** : Une implémentation de la grammaire des graphiques pour les graphes et les réseaux.

♣ Classification hiérarchique :

- **fastcluster [22]** : Ensembles de routines pour la classification hiérarchique.
- **Ape [24]** : Fonctions de lecture, d'écriture, de traçage et de manipulation d'arbres phylogénétiques. Elle permet aussi d'effectuer des analyses de données comparatives dans un cadre phylogénétique, des analyses de caractères ancestraux, des analyses de diversification et de macro-évolution. Nous l'utiliserons surtout pour les tracés des dendrogramme.

### 5.2 Logique d'implémentation

#### 5.2.1 Entrées

Afin d'utiliser cette librairie, il faudra disposer d'un jeu de données constitué de deux fichiers :

- **Fichier des caractères** Ce fichier devra faire une courte présentation des séquences étudiées c'est à dire présenter l'espèce, le genre, la famille et l'ordre à laquelle elles appartiennent.
- **Fichier des distances** : Un second fichier contenant la matrice des distances de dissimilarité entre les séquences.

5. Pour l'utiliser, il faudrait juste la cloner puis changer le nom du répertoire.



### 5.2.2 Présentation des principales fonctions

Dans la librairie que j'ai développée, il y a trois types de fonctions,

- **Fonctions d'exploration ;**

- **simplifnames()** Elle prend en entrée une data frame de caractères (avec des noms et des niveaux taxonomiques) et retourne une data frame de caractères simplifiés.
- **taxonames ()** Elle prend en entrée une data frame de caractères (avec des noms et des niveaux taxonomiques) et retourne liste de caractères contenant les niveaux taxonomiques avec les noms des taxons simplifiés
- **alltoxosfromtaxo()** Elle prend en entrée une data frame de caractères (avec des noms et des niveaux taxonomiques), une liste des noms taxonomiques et le nom d'un niveau taxonomique. Elle retourne une liste de caractères contenant les sous-taxons in ce taxon
- **viewmat()** Prend en entrée une matrice de distances et trace sa heatmap
- **histo()** Prend en argument une sous-matrice des distances et trace son histogramme.

- **Fonction de classification ;**

- **classifhier()** Elle prend en entrée une sous-matrice des distances et le nombre de classes. Elle retourne une liste d'éléments basé sur la classification hiérarchique des éléments de cette matrice.  
Cette fonction donne aussi le dendrogramme de notre classification.
- **classifsbm()** Elle prend en entrée une sous-matrice des distances et le nombre de classes. Elle retourne une liste d'éléments basé sur UNE classification via un modèle SBM distances des éléments de cette matrice.

- **Fonction de comparaison ;**

- **Contingence()** Elle prend en entrée deux partitions et retourne la table de contingence de ces deux partitions.  
Afin de comprendre les résultats issus de ces tables de contingence, je donnerais une petite explication ci dessous :  
Soit  $M \in \mathbb{M}_n(\mathbb{N})$ ,  $i$  l'indice de la ligne et  $j$  celui de la colonne :

- **Comparaison entre SBM distance et CAH :**  $M_{i,j} = \#(G_{sbm,i} \cap G_{cah,j})$
- **Comparaison entre SBM distance et la botanique :**  $M_{i,j} = \#(G_{sbm,i} \cap G_{bota,j})$
- **Comparaison entre CAH et la botanique :**  $M_{i,j} = \#(G_{cah,i} \cap G_{bota,j})$

## 6 Résultats

### 6.1 L'apport du modèle de distance par rapport au binaire

Durant ce stage, nous avons commencer par tester les modèles sbm binaire puis sbm distance sur des petits jeux de données portant sur des diatomées provenant d'Arcachon. Nous avons tester trois cas :

- **Cas<sub>1</sub>** : On compare les partitions résultantes des modèles avec les nombres de classes qui maximisent leurs critères I.C.L. Pour cela :
  1. On trouve le nombre de classe pour chaque méthode.  
Ces nombres seront notés  $n_1$  pour le nombre de classe issues de l'approche binaire et  $n_2$  pour l'approche des poids.
  2. On crée ensuite une table de contingence  $M \in M_{n_1, n_2}$  qu'on remplira avec le nombre d'éléments communs entres les classes.
- **Cas<sub>2</sub>** : On simule les deux modèles avec nombre de classes qui maximise le critère I.C.L.
  1. On refait notre classification sbm distance et on choisit le modèle avec  $n_1$  classes.
  2. On crée ensuite une table de contingence  $M \in \mathbb{M}_{n_1}$  qu'on remplira avec le nombre d'éléments communs entres les classes
- **Cas<sub>3</sub>** : On teste avec le nombre de classes qui maximisent le critère I.C.L. pour l'approche des poids.
  1. On refait notre classification sbm binaire et on choisit le modèle avec  $n_2$  classes.
  2. On crée ensuite une table de contingence  $M \in \mathbb{M}_{n_2}$  qu'on remplira avec le nombre d'éléments communs entres les classes.

#### 6.1.1 Résultats obtenus :

Dans le cadre de nos simulation le nombre de classe qui maximisent nos modèles sont :  $n_1 = 7$  et  $n_2=9$ .

Voici les résultats obtenus :

#### Tables de contingence :

Cas 1 :									Cas 2 :							Cas 3 :								
$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 12 & 0 & 3 & 0 \\ 0 & 14 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 7 & 17 & 0 & 0 & 0 & 0 & 4 \\ 0 & 0 & 0 & 0 & 6 & 0 & 0 & 0 & 0 \\ 0 & 0 & 16 & 3 & 0 & 0 & 6 & 0 & 12 \\ 0 & 0 & 0 & 0 & 10 & 0 & 0 & 0 & 0 \\ 5 & 0 & 0 & 0 & 0 & 0 & 0 & 15 & 0 \end{pmatrix}$									$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 12 & 0 \\ 0 & 14 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 7 & 17 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 6 & 0 & 0 \\ 0 & 0 & 16 & 3 & 0 & 0 & 6 \\ 0 & 0 & 0 & 0 & 10 & 0 & 0 \\ 5 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$							$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 11 & 0 & 3 & 0 \\ 0 & 14 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 7 & 17 & 0 & 0 & 0 & 0 & 4 \\ 0 & 0 & 0 & 0 & 5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 16 & 3 & 0 & 0 & 6 & 0 & 12 \\ 5 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 7 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 12 & 0 \\ 0 & 0 & 0 & 0 & 4 & 0 & 0 & 0 & 0 \end{pmatrix}$								

**Analyse :** On peut remarquer que les trois tables de contingences nous indiquent que les classes obtenues à l'aide d'un modèle à distances et celles obtenues via un modèle binaire sont très similaires.

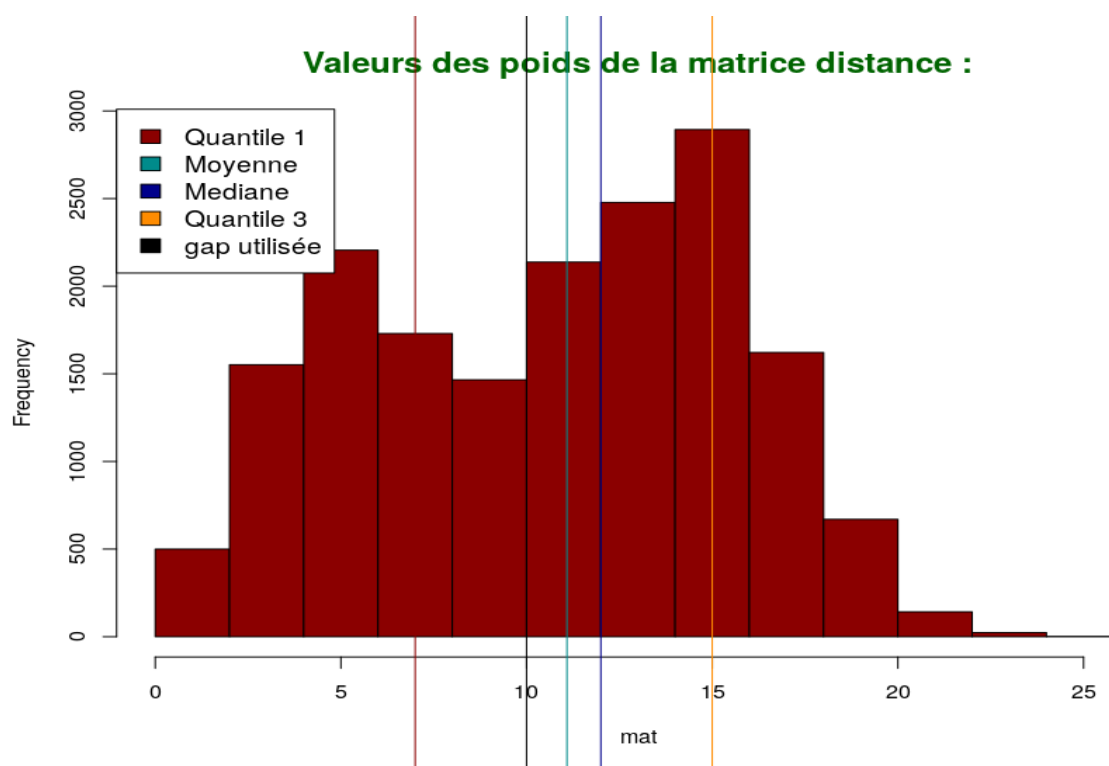
### 6.1.2 Explication de notre choix

Durant ce stage, la première partie consistait à explorer le modèle sbm binaire puis le modèle sbm à distances. Cela avait pour but de définir le modèle à utiliser jusqu'à la fin du stage.

Après avoir testé nos modèles sur plusieurs jeux de données, nous avons opté pour le modèle à distances pour différentes raisons :

- Le modèle à distance nous utilise plus d'information vue qu'il prend en entrée une matrice de distance et non une matrice d'adjacence indiquant s'il y a des connections entre les sommets. Et par ce fait, les grandes distances sont prises en compte.
- Le second avantage que nous procurent le modèle SBM distance est qu'il n'a pas besoin de seuil, un inconvénient majeur pour le modèle binaire car dans le cas où nous avons un très grand jeu de données, on se retrouve avec plusieurs possibilités de seuils.

Ci-dessous nous montrons un exemple concret :



#### **Interprétation :**

Ici, nous avons utilisé un gap de 10 pour l'approche Présence/absence, on peut faire varier ce gap entre le premier et le troisième quantile c'est-à-dire entre 7 et 15 afin de pouvoir visualiser les résultats obtenus.

## 6.2 Phase d'exploration

La classification classique du vivant classe les êtres vivants selon une hiérarchie de groupes de plus en plus vastes ces groupes s'appellent niveau taxonomique. Dans le cadre de ce travail on ne va s'intéresser qu'aux espèces, genres, familles et ordres et on aura comme objectifs de retrouver ces taxons via les deux méthodes de classification non supervisée présentées auparavant. Comme expliqué auparavant, le jeu de données étudié porte sur les données d'une communauté de plante en Guyane. Notre jeu contient 1502 individus, et est constitué de deux fichiers : une matrice des distances et un data frame de 1502 lignes et

4 colonnes, chacune correspond à un niveau taxonomique. Ces niveau taxonomiques sont disposés dans les proportions suivantes :

- 19 ordres.
- 49 familles.
- 185 genres.
- 379 espèces.

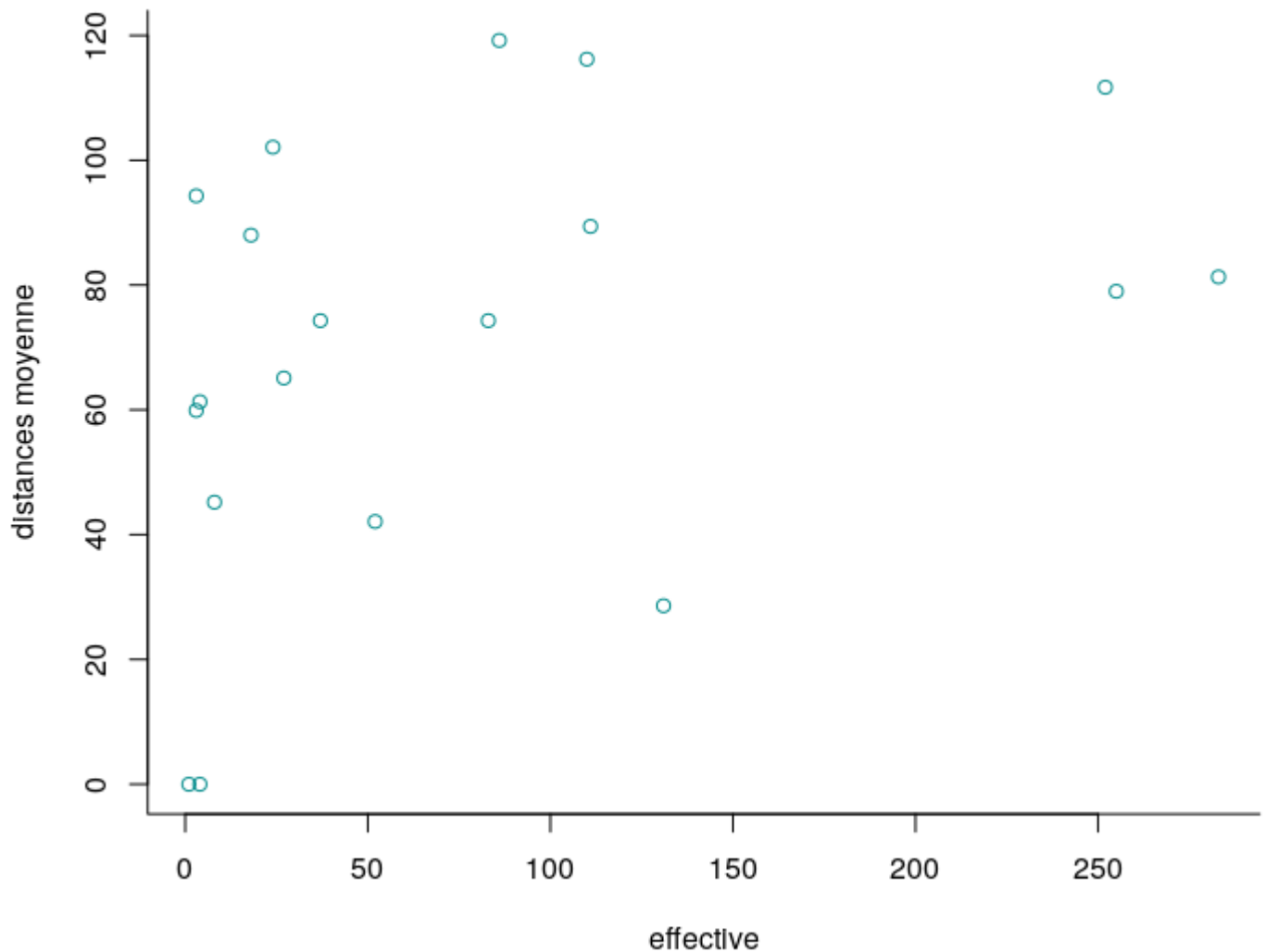
Ensuite, nous allons analyser les effectives des ordres :

Cette étape est importante car elle nous donne le classement des taxons dans leurs classes taxonomique selon leurs ordres dans notre jeu de données.

Ensuite on va résumer les ordres dont nous disposons, le nombre d'éléments qu'ils contient et la moyenne des distances intra.

<b>Ordre</b>	<b>Nombre d'éléments</b>	<b>Distance intra-ordre moyenne</b>
Ericales	283	81.3
Fabales	255	79.0
Malpighiales	252	111.7
Laurales	131	28.6
Myrtales	111	89.4
Sapindales	110	116.2
Gentianales	86	119.2
Magnoliales	83	74.3
Oxalidales	52	42.1
Malvales	37	74.3
Rosales	27	65.1
Near	24	102.1
Santalales	18	88.0
Caryophyllales	13	94.3
Celastrales	8	45.2
Lamiales	4	0
Brassicales	4	61.3
Apiales	3	59.9
Inconnu	1	0

### Distances Intra ordres moyennes par rapport au nombre d'effectives



**Analyse :** Nous avons fait un tracé avec comme abscisses les effectives des ordres et comme ordonnées les distances moyennes intra-ordres.

Suite à cette courte présentation des ordres, nous allons tout d'abord ignorer les familles qui ne possèdent que très peu d'individus c'est à dire les Apiales, les Brassicales, les Celastrales, les Lamiales et l'individu dont on ne connaît pas l'ordre. Ensuite, on va se concentrer sur deux d'entre eux en particulier, les Magnoliales et Laurales.

Ceci qui nous amène à avoir un sous-jeu de données de 214 individus composés de cette manière :

- **Magnoliales**<sup>6</sup> : est constitué de plantes angiospermes primitives. En classification classique (1981), il comprend 10 familles.  
Notre sous jeu ne contenant que 83 individus et se décompose en familles qui se décomposent en 10 genres.
- **Laurales**<sup>7</sup> : L'ordre des Laurales se compose de plantes angiospermes de divergence ancienne. En

6. <https://fr.wikipedia.org/wiki/Magnoliales>

7. <https://fr.wikipedia.org/wiki/Laurales>

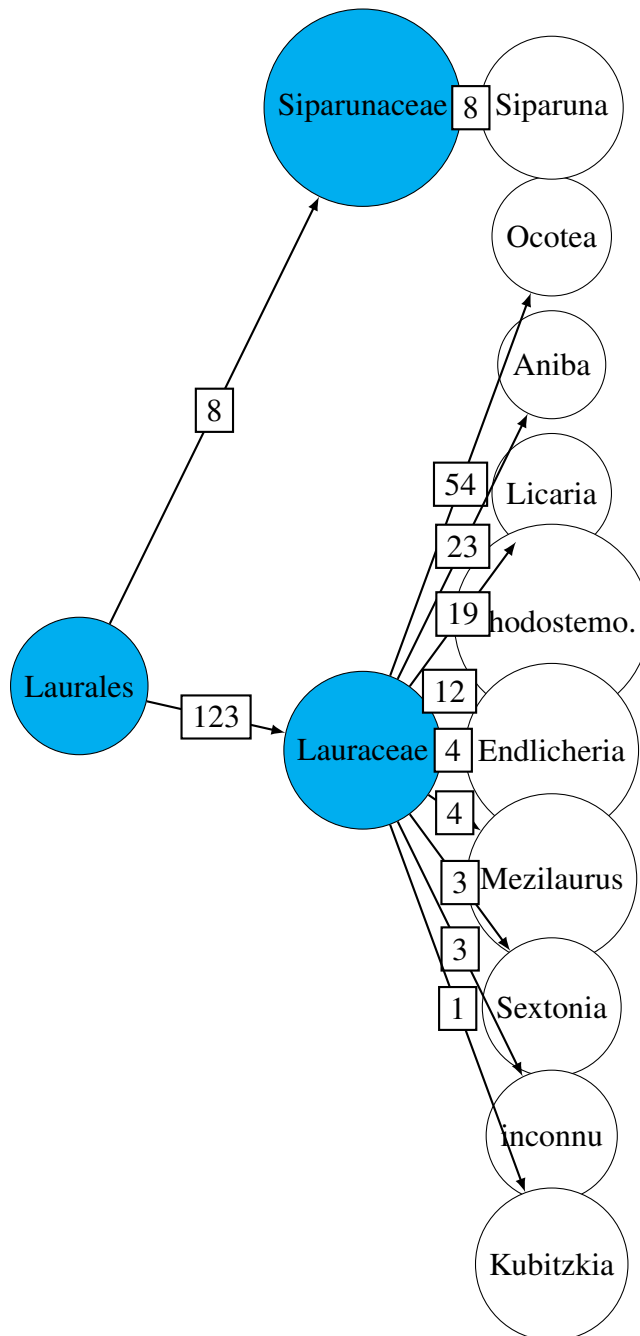
classification classique (1981), il comprend huit familles.

Comme pour notre sous-jeu des magnoliales, nous n'avons que très peu d'individus (131 individus) et donc notre ordre se décompose en 2 familles qui se décomposent en 10 genres.

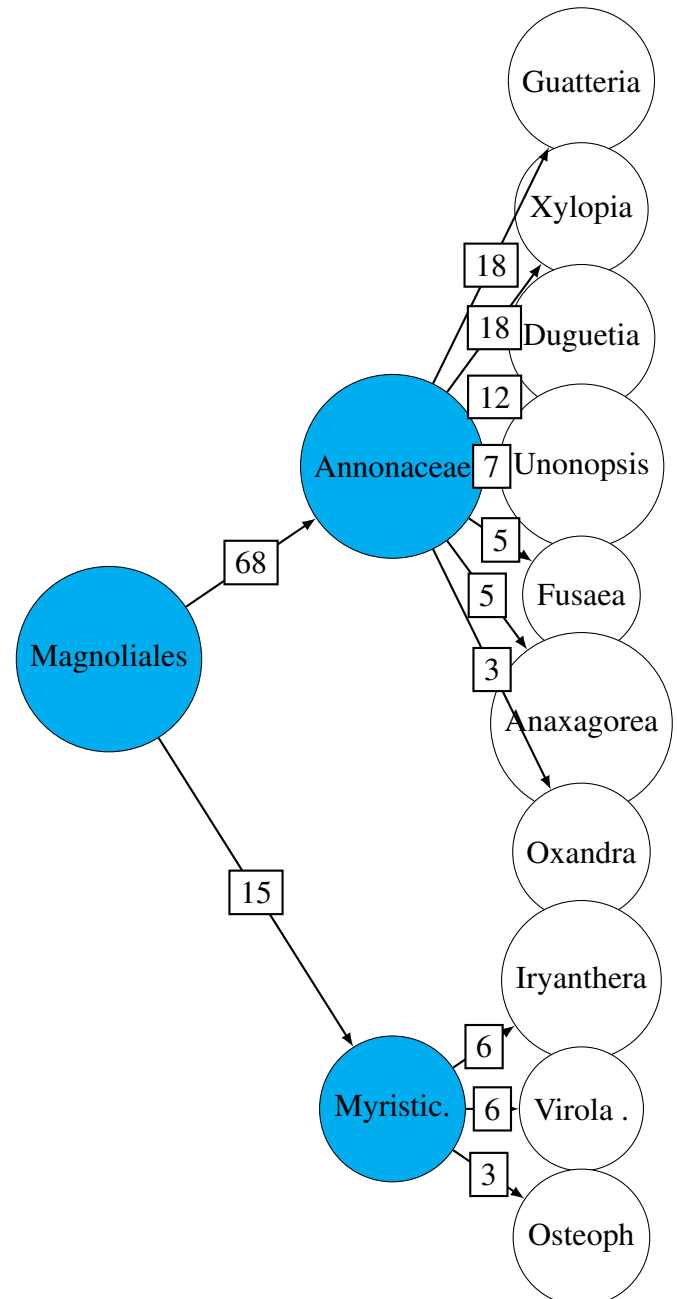
### Arbres taxonomiques :

Dans ce qui va suivre, on va montrer l'arbre taxonomiques de chacun de ces deux ordres.

#### Arbre Laurales :



#### Arbre magnoliales :



### 6.3 Exemple concret d'utilisation de la librairie

Dans cette section, nous allons montrer un exemple simple de l'utilisation de notre librairie. Notre exemple consistera à prendre un ordre, à extraire les nom des familles, à faire une première classification (selon une classification hiérarchique et via sbm distance) selon le nombre de familles. Ensuite on extrait les genres appartenant à chacune des familles puis on fait de même.

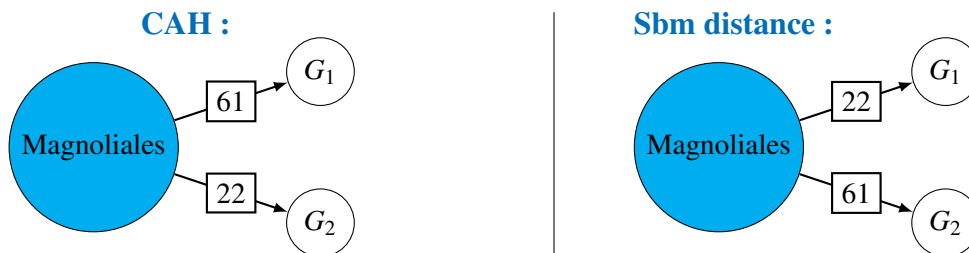
Enfin, on compare les résultats obtenus avec chacune des méthodes aux groupes à l'aide de Tables de contingence.

Pour cet exemple, nous travaillerons sur l'ordre des magnoliales, pour cela nous utiliserons la fonction `simplifynames()` pour simplifier les noms botaniques.

Ensuite, nous allons extraire les éléments appartenant à cette ordre via la fonction `extractfromlotname()`, cela nous permet de calculer la sous matrice des distances de cet ordre. Il ne reste plus qu'à classer les éléments de cette matrice. Ensuite, on va utiliser `alltoxosfromtaxo()` pour extraire les familles  $\in$  magnoliales.

Nous obtenons les "Annonaceae" et les "Myristicaceae", chose qui nous indique qu'il faudrait faire une classification à deux classes.

À l'aide des fonctions `classifhier` et `classifsbm`, nous faisons nos classifications ce qui nous donne :



#### Tables de contingence :

**Comparaison entre les données botaniques et CAH :**

$$\begin{pmatrix} 61 & 0 \\ 7 & 15 \end{pmatrix}$$

**Comparaison entre les données botaniques et SBM distance :**

$$\begin{pmatrix} 61 & 7 \\ 0 & 15 \end{pmatrix}$$

**Comparaison entre SBM distance et CAH :**

$$\begin{pmatrix} 61 & 0 \\ 0 & 22 \end{pmatrix}$$

**Interprétation :** Ici nous n'allons pas interpréter nos résultats car nous allons y revenir plus tard.

### 6.4 La similarité et dissimilarité entre la CHA et SBM distance pour 2 ou 3 classes

Ensuite, dans un soucis d'organisation, nous allons découper cette section en deux sous-sections.

#### 6.4.1 Similarité entre la CHA et SBM distance pour 2 ou 3 classes

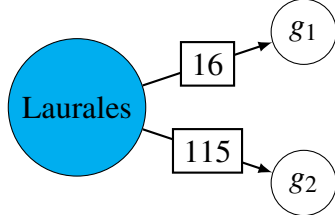
On commence par étudier les quatre ordres suivants : Les Ericales, les Gentianales, les Laurales et les Magnoliales. Pour cela, nous commencerons par analyser tout d'abord les données dont nous disposons pour voir en combien de familles chaque ordre se décompose.

## Ordres à deux familles

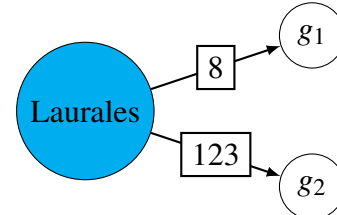
Concernant les ordre qui se décomposent en deux familles, il y a les Laurales et les Magnoliales (comme vous pouvez voir dans cette section suivante)

Ainsi, on commence avec les Laurales :

### Classification via SBM :



### Classification hiérarchique :



### Tables de contingence :

#### Comparaison entre les données botaniques et CAH :

$$\begin{pmatrix} 123 & 0 \\ 0 & 8 \end{pmatrix}$$

#### Comparaison entre les données botaniques et SBM distance :

$$\begin{pmatrix} 8 & 8 \\ 115 & 0 \end{pmatrix}$$

#### Comparaison entre SBM distance et CAH :

$$\begin{pmatrix} 8 & 8 \\ 115 & 0 \end{pmatrix}$$

**NB :** Pour la matrice à gauche : En lignes les classes CAH et en colonnes les familles botanique. Pour la matrice au milieu : En ligne SBM et en colonne les familles botanique.

Pour la matrice au milieu : En ligne SBM et en colonne CAH.

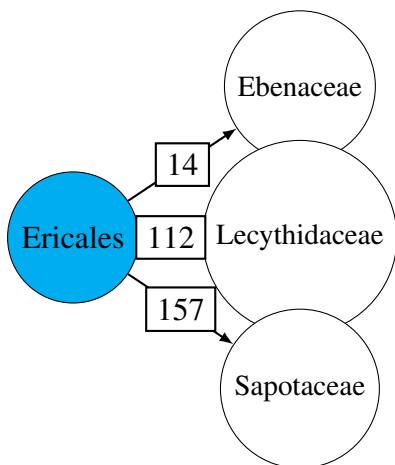
**Analyse :** Nous sommes en présence de trois matrice 2x2 qui représente le nombre d'éléments en commun entre les partitions résultantes entre les méthodes de classification et la réalité botanique.

On peut remarquer que les partition issues de la classification hiérarchique collent parfaitement aux familles botaniques. Sbm donnent 93.8% de bonne classifications.

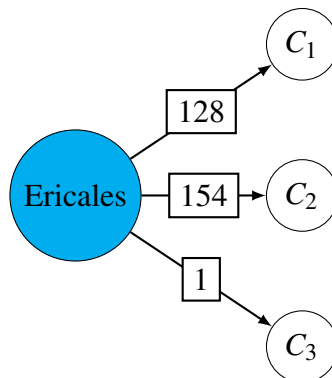
## Ordres à trois familles

Concernant les ordres qui se décomposent en deux familles, il y a les Ericales et les Gentianales. Pour ce dessein, nous commencerons par l'ordre des Ericales .

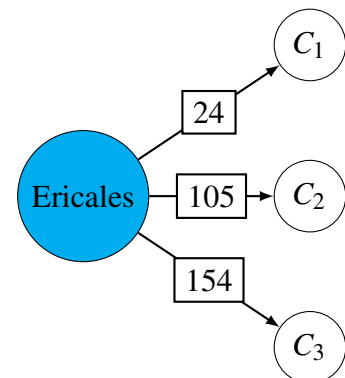
### Réalité botanique :



### Classification hiérarchique :



### Classification via SBM distance :



### Tables de contingence :



**Comparaison entre les données  
botaniques et CAH :**

$$\begin{pmatrix} 111 & 3 & 14 \\ 0 & 154 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

**Comparaison entre les données  
botaniques et SBM distance :**

$$\begin{pmatrix} 9 & 1 & 14 \\ 103 & 2 & 0 \\ 0 & 154 & 0 \end{pmatrix}$$

**Comparaison entre SBM  
distance et CAH :**

$$\begin{pmatrix} 23 & 105 & 0 \\ 0 & 0 & 154 \\ 1 & 0 & 0 \end{pmatrix}$$

**NB :** Pour la matrice à gauche : En lignes les classes CAH et en colonnes les familles botanique. Pour la matrice au milieu : En ligne SBM et en colonne les familles botanique.

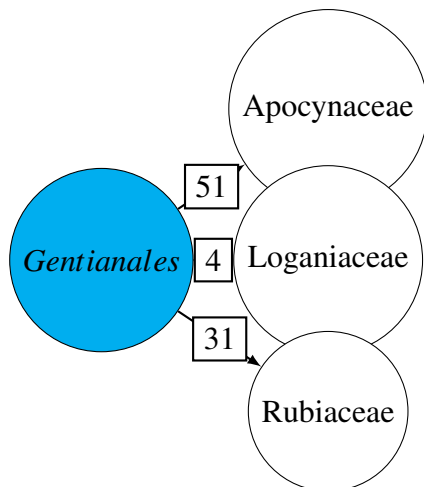
Pour la matrice au milieu : En ligne SBM et en colonne CAH.

**Analyse :** Nous sommes en présence de trois matrice 3x3 qui représente le nombre d'éléments en commun entre les partitions résultantes entre les méthodes de classification et la réalité botanique.

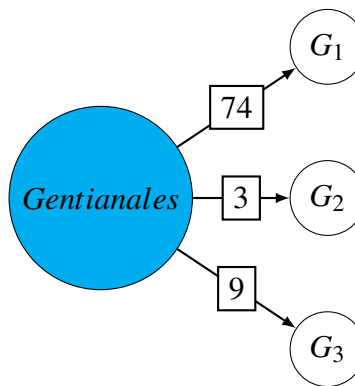
On peut remarquer que les partition issues de la classification hiérarchique ne représente pas la famille des Ebanaceae contrairement à la classification via des sbm distances.

**Gentianales :**

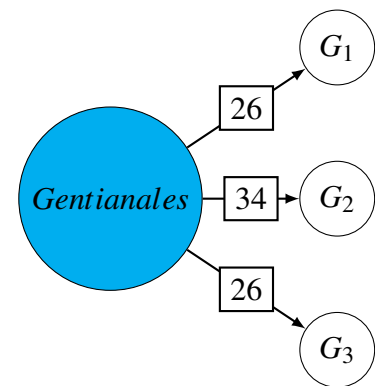
**Réalité botanique :**



**Classification hiérarchique :**



**Classification via SBM distance :**



**Tables de contingence :**

**Comparaison entre les données  
botaniques et CAH :**

$$\begin{pmatrix} 28 & 42 & 4 \\ 3 & 0 & 0 \\ 0 & 9 & 0 \end{pmatrix}$$

**Comparaison entre les données  
botaniques et SBM distance :**

$$\begin{pmatrix} 0 & 26 & 0 \\ 17 & 16 & 1 \\ 14 & 9 & 3 \end{pmatrix}$$

**Comparaison entre SBM  
distance et CAH :**

$$\begin{pmatrix} 17 & 34 & 23 \\ 0 & 0 & 3 \\ 9 & 0 & 0 \end{pmatrix}$$

**NB :** Pour la matrice à gauche : En lignes les classes CAH et en colonnes les familles botanique. Pour la matrice au milieu : En ligne SBM et en colonne les familles botanique.

Pour la matrice au milieu : En ligne SBM et en colonne CAH.

**Analyse :** Nous sommes en présence de trois matrice 3x3 qui représente le nombre d'éléments en commun entre les partitions résultantes entre les méthodes de classification et la réalité botanique.

On peut remarquer que les partitions issues de la classification hiérarchique ne collent presque pas du tout aux familles botaniques : En effet sur les trois partition qu'on obtient : On peut remarquer que  $G_1$ , la première, contient une grande partie des éléments des trois familles

**Bilan :**

En vue des résultats obtenus, on peut dire que pour des classifications à deux classe, la classification hiérarchique ascendante semble plus adaptée à la fois en terme de bonne attribution des classes et de temps de calcul.

Cependant, on peut remarquer que que dès que le nombre de classes passe à trois la classification via des modèles sbm devient de plus en plus intéressante.

Cela peut s'expliquer par le fait que quand essaye de classer des individus en deux classe on se retrouve dans une classification binaire. et donc la classification hiérarchique ou via des modèles sbm sont très proches.

Ensuite, quand on passe à une classification multinomiale (*multi-classes*), cela fait que les deux méthodes divergent complètement, en effet, la CAH résout le problème de classification multi-classes en divisant l'espace de sortie avec une approche **One-vs.plus proche voisin**. Contrairement à la classification via sbm distance qui se base sur une approche **One-vs.-all**.

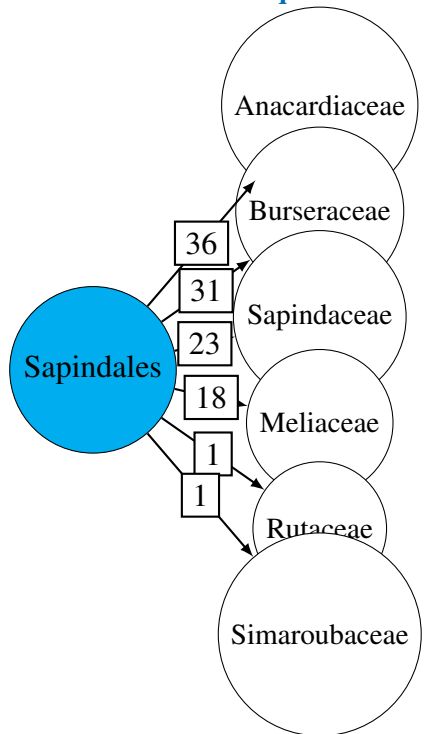
### 6.4.2 L'apport du modèle de distance par rapport au binaire pour plus de 3 classes

Cette sous-sous-section se décompose en deux partie, la première sera consacrée aux deux ordres qui se décomposent en plus de trois familles et la seconde sera destinée à exploiter nos deux manières de classification pour classer les genres.

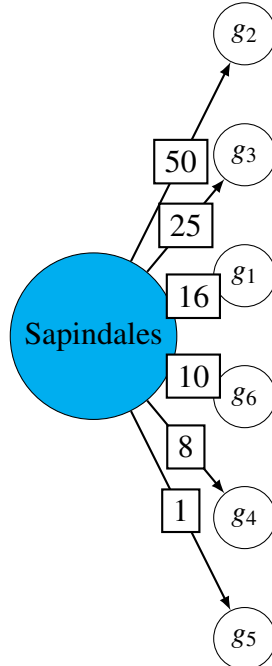
#### Ordre à plus de trois familles

Nous allons commencer par nous intéresser à l'ordre des Sapindales, cet ordre se décompose en 6 familles sous les proportions suivantes.

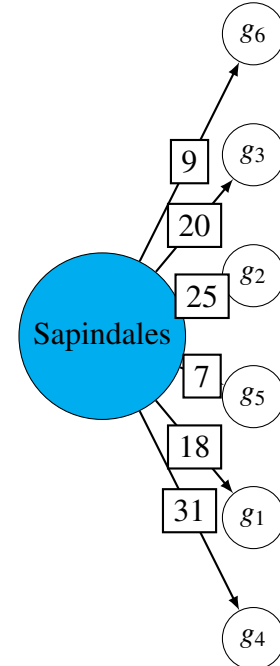
##### Réalité botanique :



##### SBM distance :



##### CAH :



Ensuite, afin d'interpréter nos résultats, nous utiliserons les tables de contingences. Étant donné que nous disposons de six classes, ces tables  $\in \mathbb{M}_6(\mathbb{N})$ , ce qui nous oblige à ne montrer que les comparaisons entre les partitions de nos méthodes de classification et la réalité botanique.

## Tables de contingence :

## Comparaison entre les données botaniques et

## Sbm distance :

$$\begin{pmatrix} 16 & 0 & 0 & 0 & 0 & 0 \\ 2 & 13 & 30 & 3 & 1 & 1 \\ 0 & 0 & 0 & 25 & 0 & 0 \\ 0 & 0 & 1 & 7 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 10 & 0 & 0 & 0 & 0 \end{pmatrix}$$

## Comparaison entre les données botaniques et

## CAH :

$$\begin{pmatrix} 9 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 19 & 1 & 0 & 0 \\ 0 & 0 & 0 & 25 & 0 & 0 \\ 7 & 0 & 0 & 0 & 0 & 0 \\ 0 & 9 & 6 & 2 & 0 & 1 \\ 2 & 14 & 6 & 8 & 1 & 0 \end{pmatrix}$$

**NB :** Pour la matrice à gauche : En ligne SBM et en colonne les familles botanique.

Pour la matrice à droite : En lignes les classes CAH et en colonnes les familles botanique.

**Analyse :** Nous sommes en présence de trois matrice 6x6 qui représente le nombre d'éléments en commun entre les partitions résultantes entre les méthodes de classification et la réalité botanique.

On peut remarquer que ni les partition issues de la classification hiérarchique ni sbm distance ne collent parfaitement avec la réalité botanique, bien que Sbm distance reste un peu plus proche de la réalité botanique.

## Classification des genres des magnoliales et laurales

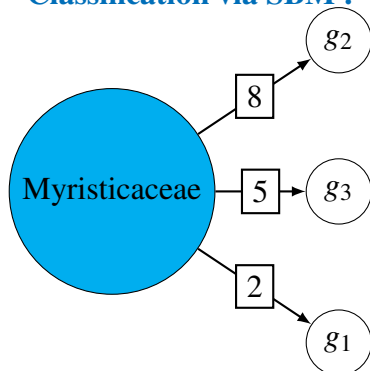
Comme on peut le voir dans l'arbre taxonomique des magnoliales et des laurales, quand on veut classer les genres que ce soit en passant par le niveau taxonomique précédant ou pas on se retrouve à vouloir faire une classification multi classe avec plus de trois classes.

## Magnoliales

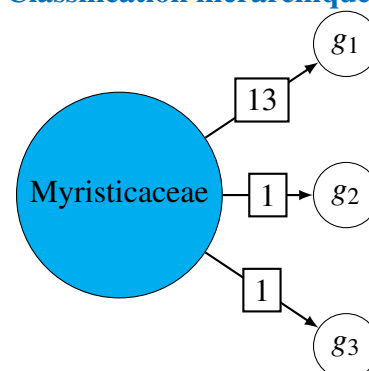
On commencera par faire une classification genres appartenant au magnoliales, pour cela, on va classer les éléments appartenant à chacune de ses familles. Pour cela nous allons procéder par famille.

## Myristiceae :

## Classification via SBM :



## Classification hiérarchique :



## Tables de contingence :

## Comparaison entre les données botaniques et CAH :

$$\begin{pmatrix} 5 & 3 & 5 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}$$

## Comparaison entre les données botaniques et SBM distance :

$$\begin{pmatrix} 1 & 0 & 1 \\ 5 & 3 & 0 \\ 0 & 0 & 5 \end{pmatrix}$$

## Comparaison entre SBM distance et CAH :

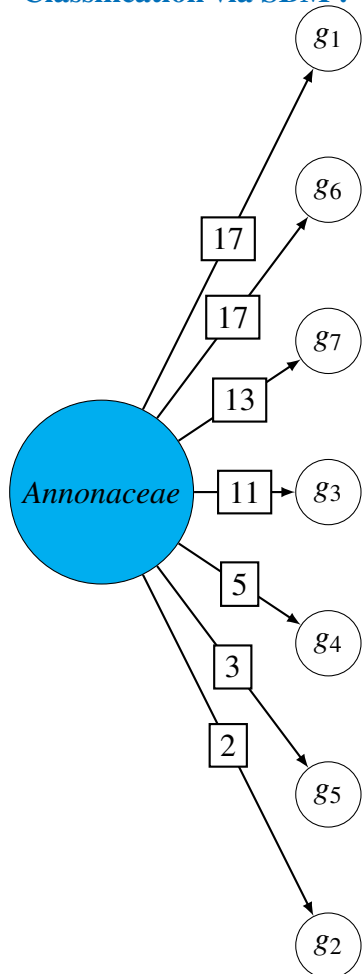
$$\begin{pmatrix} 0 & 8 & 5 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

**NB :** Pour la matrice à gauche : En lignes les classes CAH et en colonnes les familles botanique. Pour la matrice au milieu : En ligne SBM et en colonne les familles botanique.

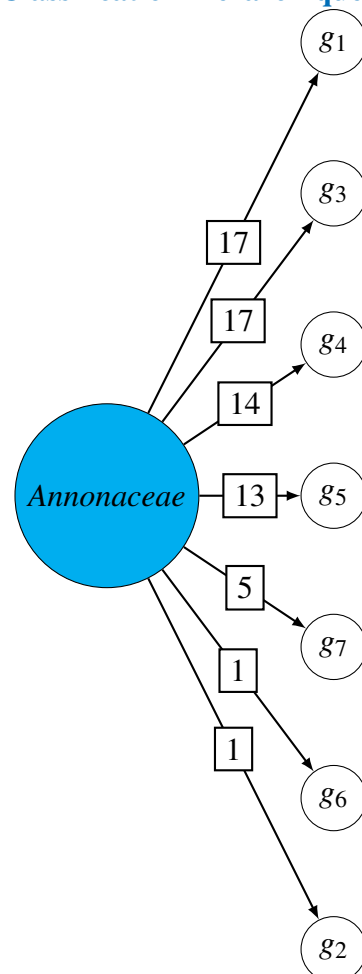
Pour la matrice au milieu : En ligne SBM et en colonne CAH.

**Annonaceae :** Pour les éléments. cette famille, nous allons faire dans un des classifications à 7 classes

**Classification via SBM :**



**Classification hiérarchique :**



**Tables de contingence :**

**Comparaison entre les données botaniques et CAH :**

$$\begin{pmatrix} 17 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 12 & 0 & 0 & 5 & 0 & 0 \\ 0 & 0 & 4 & 7 & 0 & 3 & 0 \\ 0 & 0 & 13 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 5 \end{pmatrix}$$

**Comparaison entre les données botaniques et SBM distance :**

$$\begin{pmatrix} 0 & 12 & 0 & 0 & 5 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 4 & 6 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 5 \\ 0 & 0 & 0 & 1 & 0 & 2 & 0 \\ 17 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 13 & 0 & 0 & 0 & 0 \end{pmatrix}$$

**Comparaison entre SBM distance et CAH :**

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 17 & 0 \\ 17 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 11 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 13 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 5 & 0 & 0 & 0 \end{pmatrix}$$

**NB :** Pour la matrice à gauche : En lignes les classes CAH et en colonnes les familles botanique. Pour la matrice au milieu : En ligne SBM et en colonne les familles botanique.

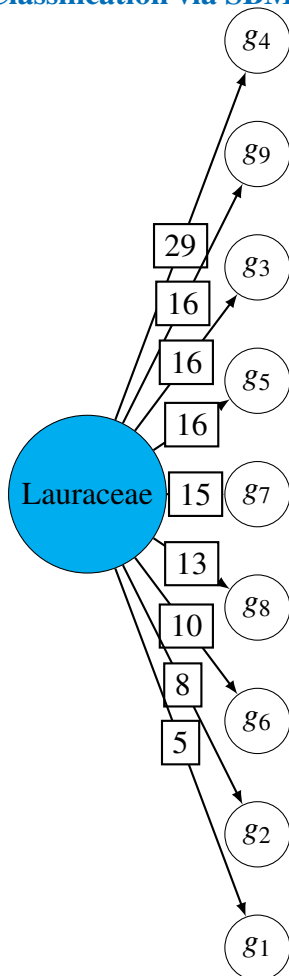
Pour la matrice au milieu : En ligne SBM et en colonne CAH.

## Laurales

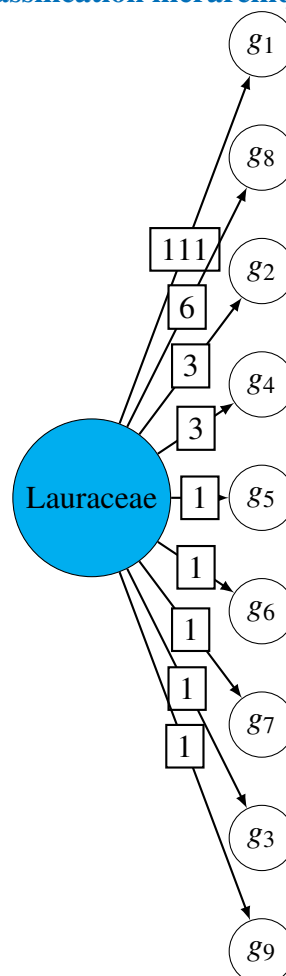
Ensuite on classera les genres appartenant au Laurales, pour cela, on va classer les éléments appartenant à chacune de ses familles.

On sait que cet ordre se décompose en deux familles, une première, les siparunacea qui donne un genre, tandis que la seconde, les Lauracea se décompose en 9 genres. Ceci fait qu'il aura besoin de faire une seule classification à neuf classes.

### Classification via SBM :



### Classification hiérarchique :



### Bilan :

En vue des résultats obtenus, on peut dire que pour un nombre de classes élevé, la classification SBM semble mieux coller avec la réalité botanique surtout pour des familles où la distance intra familles est très petite, comme les Lauraceae où on peut remarquer que dans le résultat CAH, on obtient une partition à 111 individus.

## 7 Perspectives et conclusion

### 7.1 Perspectives :

Pour parler des perspectives, je dirais que suite à ce stage je ferais une thèse portant sur le même sujet avec une approche basée sur des tenseurs pour approcher l'algorithme E.M.

**Définition 7** *Un tenseur est un tableau Multidimensionnel :*

$$A = [A(i_1, \dots, i_n)], i_k \in \{1, \dots, B\}$$

Avec,

- $n$  le nombre d'entrées
- $B$  : la dimension de chaque entrée

Après avoir défini les tenseurs<sup>26</sup>, nous allons maintenant parler de leur utilité. Par définition, un tenseur est un outil omniprésent en calcul numérique car ils formalisent des fonctions ou distributions à plusieurs variables [18] (y compris un très grand nombre de variables).

Les calculs sur les tenseurs exacts sont impossibles ( $d$  variables avec  $n$  valeurs pour chacune impliquent la manipulation de tableaux à  $n^d$  éléments). Aussi, des approches analogues aux approximations de rang faible en calcul matriciel ont-elles été développées depuis plusieurs décennies, avec plusieurs définitions du rang d'un tenseur [11]. On recherche un tenseur de rang faible le plus proche du tenseur exact, et on réalise les calculs sur le tenseur approché. Cette technique ne permet cependant pas de traiter des valeurs grandes de  $d$ . Il y a une dizaine d'années, ce domaine a "explosé", avec notamment un format économe en stockage, le format "tensor-train", dit TT, et surtout, avec des algorithmes d'approximation optimale ne nécessitant pas le stockage de tous les éléments du tenseur. L'idée à la base de ce choix pour la thèse est d'utiliser des approximations TT de rang faible du tenseur de la loi jointe d'un SBM ( $d$  est le nombre de séquences ...) pour passer à l'échelle via cette voie.

**Définition 8** *Définition d'un Tensor-Train (Oseledets, 2011) :*

*Un tensor train ou TT est un tenseur  $A = [A(i_1, \dots, i_n)], i_k \in \{1, \dots, B\}$  tel que :*

$$\begin{aligned} A[(i_1, \dots, i_n)] = & \sum_{a_1 \dots a_{n-1}} u[i_1, a_1] \times G[a_1, i_2, a_2] \times G[a_2, i_3, a_3] \\ & \times \dots \times G[a_{n-2}, i_{n-1}, a_{n-1}] \times v[a_{n-1}, i_n] \\ & \text{où } a_k \in \{1 \dots r\} \end{aligned}$$

Ainsi, l'utilisation de ces tenseurs est due à plusieurs raisons :

- Un tenseur dans une base est un tableau à plusieurs entrées, et il existe un calcul tensoriel pour les manipuler (opérations, notion de rang, d'approximation, ...)
- Un TT permet d'approcher au mieux la loi jointe ( $B^n$  termes) en seulement  $nBr^2$  termes  $\Rightarrow$  forte réduction de la taille (linéaire en  $n$  et  $B$ ).

### 7.2 Conclusion

Suite à cette présentation des perspectives, il est temps de conclure ce rapport, pour cela, je commencerais par effectuer en premier un bilan global du stage que j'ai effectué :

- Premièrement, sur un plan personnel ce stage m'a permis de reprendre contact avec le milieu de la recherche que j'ai pendant longtemps souhaité côtoyer.
- Ensuite, sur un point de vue pratique, ce stage m'a permis d'élargir mes connaissances sur le langage R, à approfondir mes connaissances en statistiques, en informatique et en machine learning. De plus, l'utilisation de machine dotée d'un système d'exploitation Linux m'a permis de me familiariser avec les lignes de commandes.
- Enfin, durant ce stage, j'ai aussi appris une notion très importante qui est la programmation structurée, beaucoup plus pratique et organisée que ma manière de programmer.

Enfin, concernant l'influence de ce stage sur mes perspectives d'avenir :

- Cette seconde expérience dans un laboratoire de recherche m'a permis de découvrir les différentes étapes de déroulement d'un projet de recherche scientifique, qui à mon opinion s'articule parfaitement avec la gestion de projet en entreprise ce qui constitue pour moi une bonne initiation dans le monde professionnel.
- Ensuite, durant mon stage, j'ai fait une première présentation de mon stage durant la journée réunion du GDR les 13 mai 2019 à Avignon ainsi que durant la journée des stagiaires de mon unité.

## Références

- [1] Barcoding et Metabarcoding.
- [2] Classification Ascendante Hiérarchique (CAH).
- [3] Code-barres ADN pour caractériser la biodiversité - Encyclopédie de l'environnement.
- [4] Présentation du metabarcoding.
- [5] *The Growth of Biological Thought; Diversity, Evolution, Inheritance*. 1985.
- [6] Classification, June 2019. Page Version ID : 160521779.
- [7] Institut national de la recherche agronomique, July 2019. Page Version ID : 161286495.
- [8] Taxonomie, August 2019. Page Version ID : 161503850.
- [9] Mohamed Anwar ABOUABDALLAH. Estimation des paramètres de diffusion d'une population dans un paysage hétérogène à partir d'équations aux dérivées partielles et de données génétiques spatialisées : . Technical report.
- [10] D. Jamet J.M. Ginoux B. Bandeira, J.L. Jamet. Mathematical convergences of biodiversity indices. *Ecological Indicators*, no 29, p. 522-528, juin 2013,.
- [11] T. G. Kolda & B. W. Bader. Tensor decompositions and applications. *SIAM Review*, 2009.
- [12] P Blanchard, JM Chaumeil, F Frigerio, F Rimet, S Salin, O Thèron, Coulaud, and A Franc. A geometric view of biodiversity : scaling to metagenomics. Technical report, INRIA, 2018.
- [13] F Forbes O François C Chen, E Durand. Bayesian clustering algorithms ascertaining spatial population structure : a new computer program and a comparison study. *Molecular Ecology Resources* j, 2007.
- [14] J.-J. Daudin, S. Robin, and F. Picard. A mixture model for random graphs. December 2007.
- [15] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. pages 1–38, 1977.
- [16] Dirk Eddelbuettel and James Joseph Balamuta. Extending extitR with extitC++ : A Brief Introduction to extitRcpp. *PeerJ Preprints*, 5 :e3188v1, aug 2017.
- [17] J. C. Gower. A General Coefficient of Similarity and Some of Its Properties. *Biometrics*, 27(4) :857, December 1971.
- [18] W. Hackbusch. Tensor spaces and numerical tensor calculus. *Springer*, pages 236–244, March , (2012).
- [19] INRA and Jean-Benoist Leger. *blockmodels : Latent and Stochastic Block Model Estimation by a 'V-EM' Algorithm*, 2015. R package version 1.1.1.
- [20] Beal M. Variational algorithm for approximate bayesian inference. *M.A., M.Sci., Physics, University of Cambridge, UK*, 2003.
- [21] Nicholas Metropolis. The beginning of the monte carlo method. *Los Alamos Science*, no 15,, 1987.
- [22] Daniel Müllner. fastcluster : Fast hierarchical, agglomerative clustering routines for R and Python. *Journal of Statistical Software*, 53(9) :1–18, 2013.
- [23] GUEDON OLIVIER and VERSHYNIN ROMAN. COMMUNITY DETECTION IN SPARSE NETWORKS VIA GROTHENDIECK'S INEQUALITY. page 28, 2015.
- [24] E. Paradis and K. Schliep. ape 5.0 : an environment for modern phylogenetics and evolutionary analyses in R. *Bioinformatics*, 35 :526–528, 2018.



- [25] R Core Team. *R : A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2019.
- [26] Anton Rodomanov. Introduction to the Tensor Train Decomposition and Its Applications in Machine Learning. In *HSE Seminar on Applied Linear Algebra, Moscow, Russia*, March 2016.
- [27] N. & al. Simon. Diversity and evolution of marine phytoplankton. *C. R. Biologies*, 2009.
- [28] I.F. Spellerberg and P.J. Fedor. A tribute to claude shannon (1916–2001) and a plea for more rigorous use of species richness, species diversity and the ‘shannon–wiener’ index. *Global Ecology Biogeography*, 2003.
- [29] Joe H. Ward. Hierarchical Grouping to Optimize an Objective Function. *Journal of the American Statistical Association*, 58(301) :236–244, March 1963.
- [30] Hadley Wickham. *ggplot2 : Elegant Graphics for Data Analysis*. Springer-Verlag New York, 2016.
- [31] Dirk Eddelbuettel with contributions by Antoine Lucas, Jarek Tuszynski, Henrik Bengtsson, Simon Urbanek, Mario Frasca, Bryan Lewis, Murray Stokely, Hannes Muehleisen, Duncan Murdoch, Jim Hester, Wush Wu, Qiang Kou, Thierry Onkelinx, Michel Lang, Viliam Simko, Kurt Hornik, Radford Neal, and Kendon Bell. *digest : Create Compact Hash Digests of R Objects*, 2019. R package version 0.6.20.