



---

# **CLASSIFICATION NON SUPERVISÉE DE SPECTRES GALAXIES**

*NOM : Zenati*

*PRÉNOM : Racim*

*FORMATION : Licence 2 informatique et MIAGE*

*TUTEUR DE STAGE : M. Didier Fraix-Burnet*

*ANNÉE UNIVERSITAIRE : 2024/2025*

*DATES : du 27 Mai au 28 Juillet 2025*

## **Remerciements**

*Je tiens à exprimer ma profonde gratitude à l'ensemble de l'équipe **SHERPAS** du laboratoire **IPAG**, pour leur accueil chaleureux et leur soutien tout au long de mon stage.*

*Je remercie tout particulièrement mon tuteur de stage, **M. Didier Fraix-Burnet**, chercheur au CNRS, pour son encadrement attentif, ses précieux conseils et la qualité de son accompagnement scientifique. J'adresse également mes sincères remerciements à **M. Hugo Chambon**, futur doctorant en astrophysique, pour sa disponibilité et son aide constante.*

*Tous deux m'ont guidé avec bienveillance dans mon acclimatation à l'environnement de travail et m'ont transmis la maîtrise des outils essentiels à la réalisation de ce stage. Leur accompagnement m'a permis de progresser rapidement, de gagner en autonomie et de me sentir pleinement à l'aise pour mener à bien ce stage d'excellence.*



Remerciements.....	2
<b>1. Introduction.....</b>	<b>5</b>
1.1. Contexte scientifique : astrophysique et spectres de galaxies.....	5
1.2. Problématique de classification en grande dimension.....	5
1.3. Objectifs du stage.....	6
<b>2. Contexte et état de l'art.....</b>	<b>6</b>
2.1. Données spectrales issues du SDSS.....	6
2.2. Classification non supervisée : enjeux et méthodes classiques.....	7
2.3. Présentation de Fisher-EM : principes et applications.....	7
2.4. Nouvelles approches par deep learning (GEMINI / GEMCLUS).....	7
2.5. Motivation pour la réécriture en C++.....	8
<b>3. Données et environnement de travail.....</b>	<b>8</b>
3.1. Jeu de données SDSS utilisé ( $\approx$ 302 248 spectres).....	8
3.2. Données spécifiques de la galaxie NGC 1068.....	8
3.3. Prétraitements des spectres.....	9
3.4. Environnement de calcul.....	10
<b>4. Première étape : Classification des spectres SDSS avec Fisher-EM.....</b>	<b>10</b>
4.1. Contexte et références.....	10
4.2. Objectif.....	11
4.3. Pipeline de classification (scripts R développés).....	11
4.4. Résultats principaux.....	13
4.5. Discussion : importance de la standardisation et limites de l'approche.....	13
→ Importance de la standardisation.....	13
→ Limites de l'approche.....	14
→ Conclusion de la discussion.....	14
<b>5. Deuxième étape : Exploration de GEMCLUS (GEMINI, deep learning).....</b>	<b>15</b>
5.1. Présentation du package GEMCLUS et de GEMINI.....	15
5.2. Méthodologie d'expérimentation.....	15
5.3. Tests sur les données de NGC 1068.....	16
5.4. Résultats obtenus : performances et qualité des regroupements.....	17
5.5. Comparaison avec Fisher-EM.....	18
5.6. Discussion critique : apports et limites du deep learning.....	19
5.7. Lien entre interprétation physique et continuité du projet.....	20
<b>6. Troisième étape : Réécriture et optimisation de Fisher-EM en C++.....</b>	<b>21</b>
6.1. Objectifs de la réécriture.....	21
6.2. Étapes majeures réécrites en C++.....	21
6.3. Benchmarks indépendants.....	22

6.4. Validation partielle des résultats.....	22
6.5. Tentative d'intégration complète.....	23
6.6. Complications rencontrées.....	23
<b>7. Discussion générale.....</b>	<b>23</b>
7.1. Synthèse des résultats obtenus.....	23
7.2. Comparaison des approches : statistique vs deep learning.....	24
7.3. Apports méthodologiques du stage.....	24
7.4. Limites rencontrées et solutions envisagées.....	25
<b>8. Conclusion et perspectives.....</b>	<b>25</b>
8.1. Bilan scientifique et technique du stage.....	25
8.2. Importance de la standardisation en haute dimension.....	25
8.3. Perspectives de recherche futures.....	26

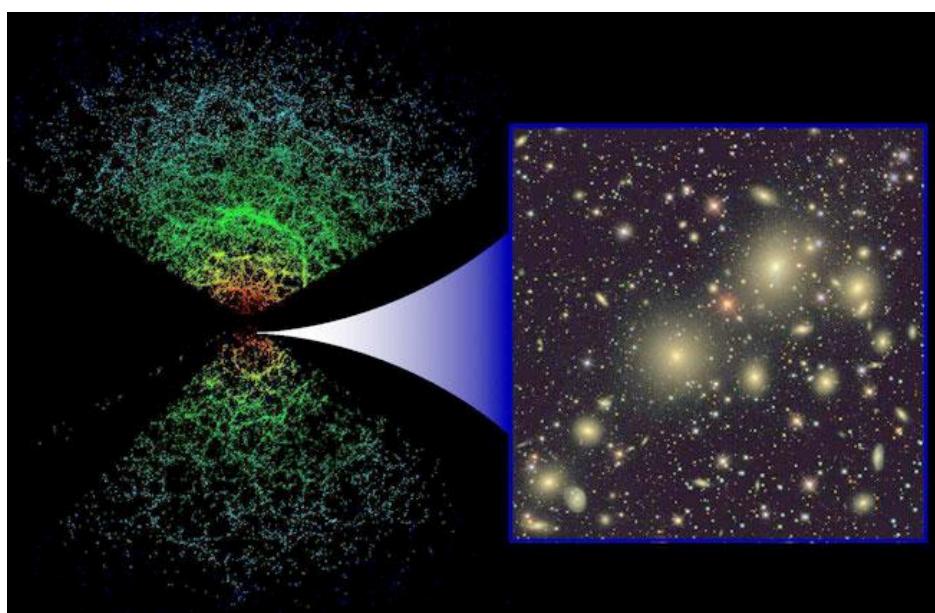
# 1. Introduction

## 1.1. Contexte scientifique : astrophysique et spectres de galaxies

L'astrophysique moderne s'appuie sur des bases de données massives issues de relevés spectroscopiques, tels que le **Sloan Digital Sky Survey (SDSS)**. Ces relevés permettent d'obtenir des spectres de galaxies, qui contiennent une information riche sur leurs propriétés physiques (composition chimique, âge des populations stellaires, activité nucléaire, etc.).

L'analyse de ces spectres constitue un enjeu majeur pour mieux comprendre l'évolution des galaxies et les phénomènes astrophysiques sous-jacents.

*Figure 1.a : La carte de l'univers en 3D du SDSS*



## 1.2. Problématique de classification en grande dimension

Les spectres de galaxies sont des **signaux de haute dimension** (souvent plusieurs milliers de longueurs d'onde par spectre). Travailler dans un espace aussi grand pose plusieurs difficultés :

- **bruit et redondance** dans les données,
- **complexité algorithmique** pour les méthodes classiques,
- nécessité de **réduction de dimension** et de **standardisation** pour rendre les données exploitables.

Un enjeu important est donc de développer des méthodes de **classification non supervisée**, capables d'identifier des structures dans les données sans connaissance préalable des classes.

### 1.3. Objectifs du stage

Le stage s'est déroulé en **trois étapes** :

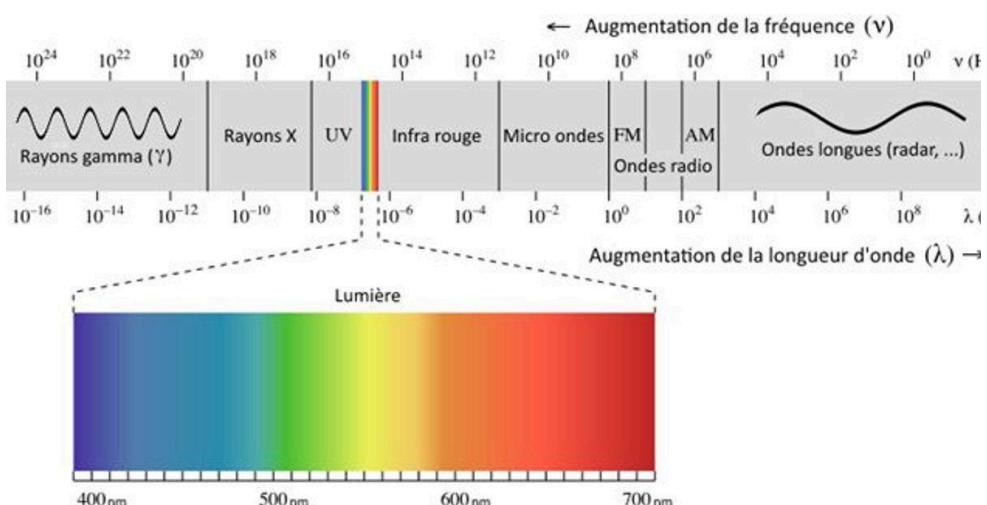
1. **Classification des spectres SDSS avec Fisher-EM (R)** – identification de sous-classes spectrales sur des données standardisées.
  2. **Exploration de GEMCLUS (GEMINI, deep learning)** – tests sur un jeu de données spécifique (NGC 1068) et comparaison avec Fisher-EM.
  3. **Réécriture de Fisher-EM en C++** – optimisation des performances et comparaison avec la version R.
- 

## 2. Contexte et état de l'art

### 2.1. Données spectrales issues du SDSS

Le **Sloan Digital Sky Survey (SDSS)** fournit des spectres de galaxies couvrant une large plage de longueurs d'onde avec une résolution suffisante pour analyser les raies spectrales. Ces données permettent d'extraire des informations physiques sur les galaxies, comme la composition chimique, l'âge des étoiles et l'activité nucléaire. Les spectres sont généralement représentés sous forme de matrices  $n \times p$  ( $n$  spectres,  $p$  longueurs d'onde).

Figure 2.a : Description d'une onde



## 2.2. Classification non supervisée : enjeux et méthodes classiques

La **classification non supervisée** vise à regrouper les objets similaires sans classes préalables. Les principaux défis sont :

- **Haute dimension des données** et bruit important.
- **Identification automatique de sous-classes** pertinentes.
- Méthodes classiques :
  - **K-means** : rapide mais sensible aux initialisations.
  - **Mélange gaussien (GMM)** : probabiliste, plus flexible.
  - **Analyse en composantes principales (PCA)** : utilisée pour réduire la dimension avant classification.

## 2.3. Présentation de Fisher-EM : principes et applications

Fisher-EM combine :

- **Modélisation par mélanges gaussiens** dans un **sous-espace discriminant**.
- **Réduction de dimension** optimisée selon le critère de **Fisher** (ratio interclasse/intraclasse).
- **Algorithme EM** pour estimer les probabilités d'appartenance.

Applications : classification automatique de spectres de galaxies, identification de sous-classes cohérentes et interprétation physique des regroupements.

## 2.4. Nouvelles approches par deep learning (GEMINI / GEMCLUS)

Les méthodes de **deep learning** comme **GEMINI** offrent une alternative aux méthodes statistiques classiques :

- Capacité à traiter de grandes bases de données et des relations complexes entre les variables.
- Flexibilité pour capturer des structures non linéaires.
- **GEMCLUS** est un package qui implémente GEMINI pour la classification non supervisée de données spectrales.

Comparaison avec Fisher-EM : robustesse, capacité à extraire des sous-classes plus fines, mais avec un coût computationnel plus élevé.

## 2.5. Motivation pour la réécriture en C++

L'implémentation R de Fisher-EM est **limité par les performances sur de gros volumes de données.**  
La réécriture en **C++** permet :

- Accélération du calcul.
  - Meilleure gestion de la mémoire.
  - Possibilité de traiter des ensembles de données beaucoup plus grands et d'intégrer Fisher-EM dans des pipelines automatisés.
- 

# 3. Données et environnement de travail

## 3.1. Jeu de données SDSS utilisé ( $\approx 302\,248$ spectres)

Le jeu de données principal provient du **Sloan Digital Sky Survey (SDSS DR14)** et contient environ **302 248 spectres de galaxies**.

Chaque spectre est représenté par **1 437 longueurs d'onde**, formant une matrice  $n \times p$  où  $n \approx 302\,248$  et  $p \approx 1437$ .

Ces spectres ont servi de base pour la classification non supervisée avec **Fisher-EM**.

## 3.2. Données spécifiques de la galaxie NGC 1068

Pour l'exploration de **GEMINI (GEMCLUS)**, un second jeu de données a été utilisé : les spectres de la **galaxie active NGC 1068**, bien connue pour son noyau de type Seyfert II.

Ces données ont permis une **comparaison ciblée** entre GEMINI et Fisher-EM sur un cas concret.

*Figure 3.a : La galaxie active NGC 1068*



### 3.3. Prétraitements des spectres

Avant toute classification, les spectres subissent plusieurs traitements :

- **Interpolation et normalisation** : sur une plage définie (par ex. 6400–6800 Å).

#### Interpolation / Rééchantillonnage et normalisation

Soit un spectre  $S(\lambda)$  défini sur

$$\lambda \in [3000, 9000] \text{ Å}.$$

1. Rééchantillonnage sur une grille régulière  $\{\lambda_j\}_{j=1}^M$  :

$$S_{\text{resample}}(\lambda_j) \approx S(\lambda).$$

2. Normalisation (centrage) :

$$S_{\text{norm}}(\lambda_j) = S_{\text{resample}}(\lambda_j) - \mu, \quad \mu = \frac{1}{M} \sum_{j=1}^M S_{\text{resample}}(\lambda_j).$$

---

#### Standardisation

Soit  $X = (x_1, \dots, x_n)$  l'ensemble des intensités pour une longueur d'onde donnée (sur  $n$  spectres).

$$z_i = \frac{x_i - \bar{x}}{s},$$

où  $\bar{x}$  est la moyenne et  $s$  l'écart-type sur ces  $n$  valeurs.

- **Standardisation** : centrage et réduction, stockés dans `sppart4567scaled.RData`.

#### Standardisation (centrage-réduction) des données spectrales :

On considère une seule longueur d'onde du spectre.

À cette longueur d'onde, on dispose des valeurs mesurées pour les  $n$  spectres de l'échantillon :

$$X = (x_1, x_2, \dots, x_n)$$

La transformation standardisée consiste à calculer :

$$z_i = \frac{x_i - \bar{x}}{s}$$

où :

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i \quad \text{et} \quad s = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2}$$

Ainsi, pour chaque longueur d'onde du spectre, on obtient un vecteur standardisé :

$$\text{moyenne}(Z) = 0, \quad \text{écart-type}(Z) = 1$$

- **Réduction de dimension** : intégrée directement dans Fisher-EM (sous-espace discriminant) et dans GEMINI (espace latent appris).

### 3.4. Environnement de calcul

- **R** : utilisé pour *Fisher-EM* et la visualisation (`FEMrecap()`, `mspsplit_paper()` ...).
  - **C++** : réécriture et optimisation de *Fisher-EM* pour améliorer les performances.
  - **Python** : utilisé pour *GEMINI* et pour la comparaisons des résultats obtenus avec les différents hyperparamètres.
  - **Clusters de calcul de l'IPAG** : exécutions massives (plusieurs centaines de runs *Fisher-EM* ou *GEMINI* pour explorer les paramètres).
- 

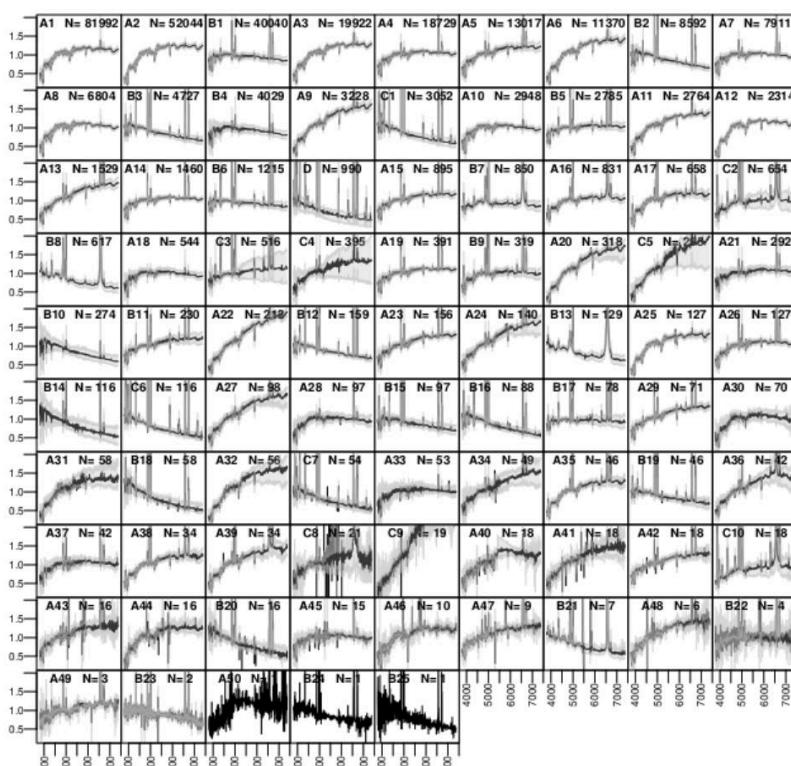
## 4. Première étape : Classification des spectres SDSS avec Fisher-EM

### 4.1. Contexte et références

Cette étape s'inspire directement de l'étude menée par [M. Fraix-Burnet et al., 2021](#), qui a appliqué **Fisher-EM** à la classification de spectres de galaxies normalisés en combinant :

- Des **modèles de mélanges gaussiens**.
- Une **réduction de dimension discriminante** basée sur le critère de Fisher.

*Figure 4.a : Classification obtenue Fraix-Burnet et Al.*



Notre travail propose une **ré-implémentation** sur des données **standardisées** issues du SDSS, afin de garantir une meilleure **robustesse et reproductibilité**.

- ❖ *Repository GitHub du projet : [FisherEM repository](#).*

## 4.2. Objectif

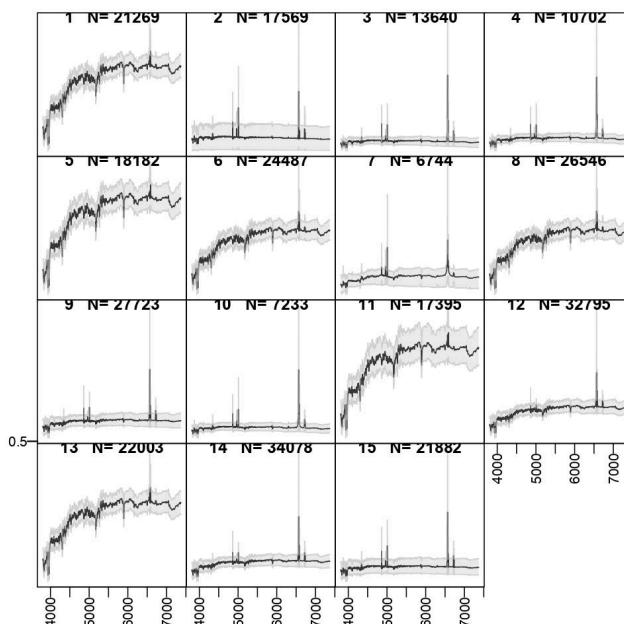
Mettre en place une **classification non supervisée** des spectres SDSS, afin d'identifier des **sous-classes spectrales** selon leurs propriétés physiques et structurelles, sans utiliser d'étiquettes préexistantes.

## 4.3. Pipeline de classification (scripts R développés)

Le pipeline complet que nous avons conçu et exécuté est structuré en **8 étapes principales** :

1. **Prétraitement des données**
  - Standardisation des spectres (*sppart4567scaled.RData*).
2. **Classification principale avec Fisher-EM**
  - Script : *run\_fem\_auto.R*
  - Exploration de plusieurs valeurs de *K* (10–40) et de modèles (*DkBk*, *DBk*, *AkB*, etc.).
  - Exécution sur cluster (SSH + batch).
3. **Choix de l'optimum via ICL**
  - Script : *plot\_FEMrecap.R*
  - Fonction : *FEMrecap()*
  - Optimum trouvé : *modèle DBk avec K = 15 (figure 4.b)*
4. **Visualisation des classes principales (K = 15)**
  - Script : *mspsplit\_sous\_classification.R*
  - Fonction : *mspsplit\_paper()*
  - Moyennes + quantiles des spectres → PDF.

*Figure 4.b : Optimum trouvé : DBK avec K=15*



## 5. Sous-classification des 15 classes principales

- Script : *run\_fem\_auto\_sous\_classification.R*
- Exploration locale (ex : K = 30–40).

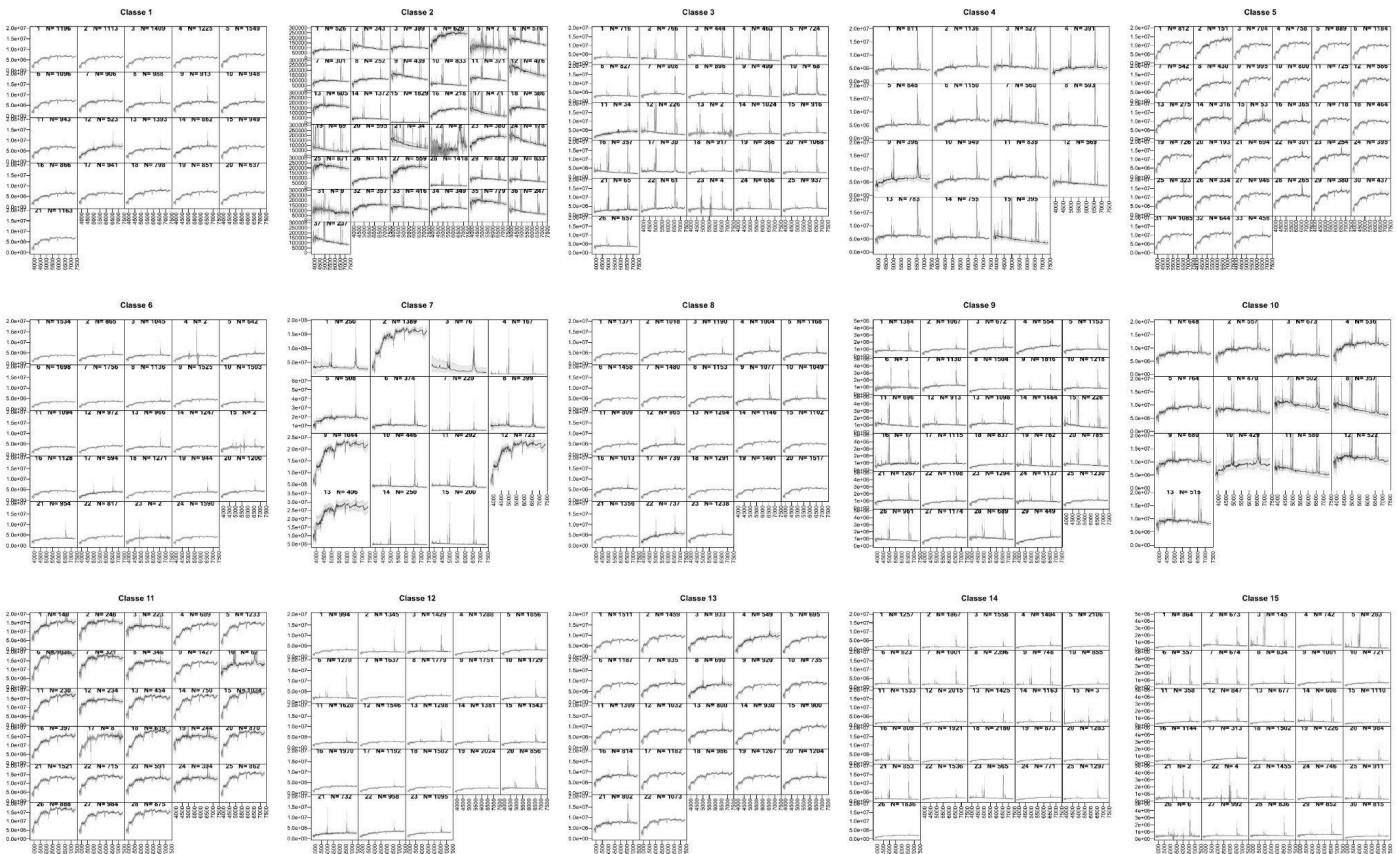
## 6. Choix des meilleures sous-classifications

- Script : *plot\_FEMrecap\_sous\_classification.R*
- ICL calculé pour chaque classe séparément.

## 7. Visualisation finale des sous-classes

- Scripts : *msspsplit\_sous\_classification.R* et *results\_msspsplit.R*
- Production de 15 PDFs fusionnés en une grille  $5 \times 3 \rightarrow$  *sous\_classification\_finale.pdf*.

Figure 4.c : Sous classification finale pour chaque classe de l'optimum 4.b



## 8. Attribution de noms standardisés aux sous-classes

- Script : *nameclass.R*
- Exemple : *A1, B3, M2...*
- Résultat : fichier *nameclass\_standardise.csv* contenant les noms des sous classes pour chaque spectre.

Pour cela, nous avons procédé de la manière suivante :

- Chaque classe principale issue de la première classification FisherEM (au nombre de 15) a été associée à une lettre de l'alphabet (de A à O).
- À l'intérieur de chaque classe, les sous-groupes obtenus par la sous-classification locale ont été numérotés (1, 2, 3, ...).

- Ainsi, chaque spectre reçoit un identifiant unique de type **A1, C5, M2**, etc., combinant **la lettre de la classe principale et le numéro de la sous-classe correspondante**.

## 4.4. Résultats principaux

- L'optimum global retenu est le modèle **DBk avec K = 15 classes principales**.
- Chaque classe a ensuite été **sous-classifiée** pour révéler des structures internes.
- Les résultats finaux incluent :
  - Les **moyennes spectrales** de chaque groupe.
  - Les **bandes de dispersion** (quantiles).
  - Une **fusion visuelle** de toutes les sous-classifications ([sous\\_classification\\_finale.pdf](#)).

## 4.5. Discussion : importance de la standardisation et limites de l'approche

### → Le résultat final

L'un des résultats majeurs de ce travail est l'identification de **362 classes finales** issues de la combinaison des deux étapes de classification :

- une **classification principale** avec Fisher-EM qui a permis de retenir, via le critère ICL, l'optimum **DBk avec K = 15 classes** ;
- une **sous-classification systématique** appliquée à chacune des 15 classes principales, aboutissant à un total de **362 sous-classes** après sélection des optima locaux.

Cette organisation hiérarchique des spectres permet de capturer à la fois la structure globale (15 grands groupes de galaxies) et la diversité interne (plusieurs dizaines de sous-groupes par classe). Elle illustre la puissance de l'approche Fisher-EM lorsqu'il s'agit de révéler des régularités statistiques dans des données spectrales de grande dimension.

### → Importance de la standardisation

La **standardisation préalable des spectres** s'est révélée essentielle. En effet, sans normalisation des intensités, les résultats obtenus sont dominés par des variations d'amplitude, ce qui biaise la séparation des classes.

Grâce à la standardisation :

- les spectres sont comparés sur la base de leur **forme spectrale** et non de leur intensité brute ;
- la classification devient plus robuste aux différences instrumentales ou aux conditions d'observation ;

- La reproductibilité des résultats est assurée, puisque les classes obtenues reflètent des propriétés astrophysiques plus intrinsèques.

Ce constat confirme les conclusions de travaux antérieurs (par ex. Fraix-Burnet et al.) et met en évidence l'importance de cette étape dans tout traitement de spectres astrophysiques à haute dimension.

### → **Limites de l'approche**

Malgré ces résultats encourageants, certaines limites doivent être soulignées :

#### **1. Complexité computationnelle :**

- L'algorithme Fisher-EM, appliqué à plus de 300 000 spectres, est coûteux en temps de calcul.
- La sous-classification, nécessitant l'exploration de multiples valeurs de K pour chaque classe, multiplie ce coût par un facteur important (d'où l'utilisation de clusters de calcul).

#### **2. Sur-segmentation possible :**

- L'obtention de **362 classes finales** peut poser la question d'un excès de granularité.
- Certaines sous-classes peuvent correspondre à des variations fines mais avec des interprétations peu pertinentes en astrophysique.
- Une étape ultérieure de regroupement et d'interprétation par des experts du domaine (**M. Fraix-Burnet & M. Chambon**) est donc nécessaire pour interpréter ces résultats.

#### **3. Limitation du modèle gaussien :**

- Fisher-EM repose sur une modélisation par mélanges gaussiens, ce qui peut être restrictif lorsque les distributions spectrales présentent des formes complexes.
- Les futures approches (par ex. GEMINI/GEMCLUS via réseaux de neurones) pourraient mieux capturer ces structures non linéaires.

### → **Conclusion de la discussion**

L'analyse non supervisée des spectres standardisés a fourni une classification plus fine que celle des spectres normalisés ([M. Fraix-Burnet et al., 2021](#)), avec 362 classes contre 87. On retrouve très bien la séparation en deux grandes catégories, séparant des spectres avec beaucoup de raies d'émission et une pente plutôt bleue (galaxies formant beaucoup d'étoiles), des autres spectres plus rouges et aux raies plus faibles (galaxies dites "passives"). Visuellement, les spectres de chacune des 362 classes sont très

homogènes, et correspondent grandement à ceux des 87 classes. Comme attendu, la différence principale résulte dans une ségrégation supplémentaire par rapport au niveau du continuum. Ceci explique en grande partie le nombre plus élevé de classes. En conclusion, la standardisation des spectres avant classification non-supervisée ne détruit pas la ségrégation dans la diversité des spectres de galaxies, et ajoute une discrimination associée à la luminosité des galaxies correspondant grossièrement à leur masse. Une étude plus approfondie va devoir être entreprise pour déterminer à quel point ce résultat change notre vision de l'évolution des galaxies.

---

## 5. Deuxième étape : Exploration de GEMCLUS (GEMINI, deep learning)

### 5.1. Présentation du package GEMCLUS et de GEMINI

Le package [GEMCLUS](#) implémente l'algorithme **GEMINI** (Generalised Mutual Information for Discriminative Clustering), une méthode de classification non supervisée reposant sur des techniques de deep learning. Contrairement aux approches classiques comme **K-means** ou **Fisher-EM**, qui reposent sur des hypothèses de distributions gaussiennes, GEMINI exploite des **distances de noyau** entre distributions, en particulier :

- **MMD (Maximum Mean Discrepancy)**, qui mesure la dissimilarité entre moyennes de distributions dans un espace de noyau ;
- **Wasserstein**, qui compare les distributions dans leur globalité, et permet de capturer des structures plus complexes.

Ces propriétés en font un outil prometteur pour analyser les spectres de galaxies, dont la variabilité dépasse souvent le cadre des modèles gaussiens.

### 5.2. Méthodologie d'expérimentation

Nous avons utilisé **GEMINI** sur les spectres de la galaxie **NGC 1068**, déjà bien étudiée avec Fisher-EM ([Chambon & Fraix-Burnet, 2023](#)).

La démarche a consisté à :

1. **Choisir les hyperparamètres** clés (dimension cachée, taux d'apprentissage, stratégie OvO/OvA, choix du noyau) et explorer leur influence sur les résultats.

Dans le cadre de l'exploration de l'algorithme GEMINI, plusieurs hyperparamètres ont été systématiquement testés afin d'évaluer leur influence sur la qualité du clustering des spectres.

- **Hidden dim (Dimension cachée)** :  
→ contrôle la capacité du réseau à représenter des structures complexes dans l'espace latent.

- **learning rate (Taux d'apprentissage) :**  
→ influence la vitesse et la stabilité de la convergence.
- **Stratégies de classification :**
  - **OvO (One-vs-One)** : comparaison par paires de classes.
  - **OvA (One-vs-All)** : séparation d'une classe contre toutes les autres.
- **Kernel (Noyaux)) :**  
→ fonction de pondération permettant d'estimer la distance entre distributions.

2. **Exécuter plusieurs classifications** en parallèle (grâce aux scripts Python du dossier `py/`), en sauvegardant systématiquement les résultats au format `.json(1)` (nombre de classes, temps d'exécution, silhouette score, valeurs des hyperparamètres).
3. **Comparer les performances** de GEMINI avec celles de Fisher-EM à l'aide de métriques comme le **silhouette score**, ainsi qu'à travers des visualisations dédiées (diagrammes, Sankey, moyennes de classes).

👉 Repository GitHub du projet : [GEMINI repository](#).

### 5.3. Tests sur les données de NGC 1068

La galaxie **NGC 1068**, souvent utilisée comme cas d'étude dans la littérature, a servi de terrain d'expérimentation. L'objectif était de vérifier si GEMINI pouvait retrouver des regroupements pertinents déjà identifiés par Fisher-EM, tout en évaluant :

- la **qualité de séparation des clusters** ;
- la **capacité à détecter des structures complexes** dans les spectres ;
- le **temps de calcul** en comparaison avec Fisher-EM.
- Comparer les résultats obtenus avec ceux issus de **FisherEM**, publiés par [Chambon & Fraix-Burnet, 2023](#).

Figure 5.a : La galaxie NGC 1068



## 5.4. Résultats obtenus : performances et qualité des regroupements

Pour déterminer les meilleures valeurs des hyperparamètres, il était nécessaire de disposer d'une **référence quantitative**. Nous avons donc utilisé des **scores d'évaluation du clustering** (notamment le *silhouette score*), qui servent de guide objectif pour comparer et sélectionner les configurations les plus pertinentes. Ce critère est détaillé dans la section suivante :

Soit un ensemble de points  $X = \{x_1, x_2, \dots, x_n\}$  partitionné en  $k$  clusters.

Pour un point  $x_i$  :

### 1. Cohésion intra-classe

$$a(i) = \frac{1}{|C_i| - 1} \sum_{\substack{x_j \in C_i \\ j \neq i}} d(x_i, x_j)$$

où  $C_i$  est le cluster auquel appartient  $x_i$ , et  $d(\cdot, \cdot)$  est la distance (souvent euclidienne).

👉 C'est la **distance moyenne** entre  $x_i$  et les autres points de son cluster.

### 2. Séparation inter-classe

$$b(i) = \min_{C \neq C_i} \frac{1}{|C|} \sum_{x_j \in C} d(x_i, x_j)$$

👉 C'est la **plus petite distance moyenne** entre  $x_i$  et les points d'un autre cluster.

### 3. Coefficient de silhouette pour $x_i$ :

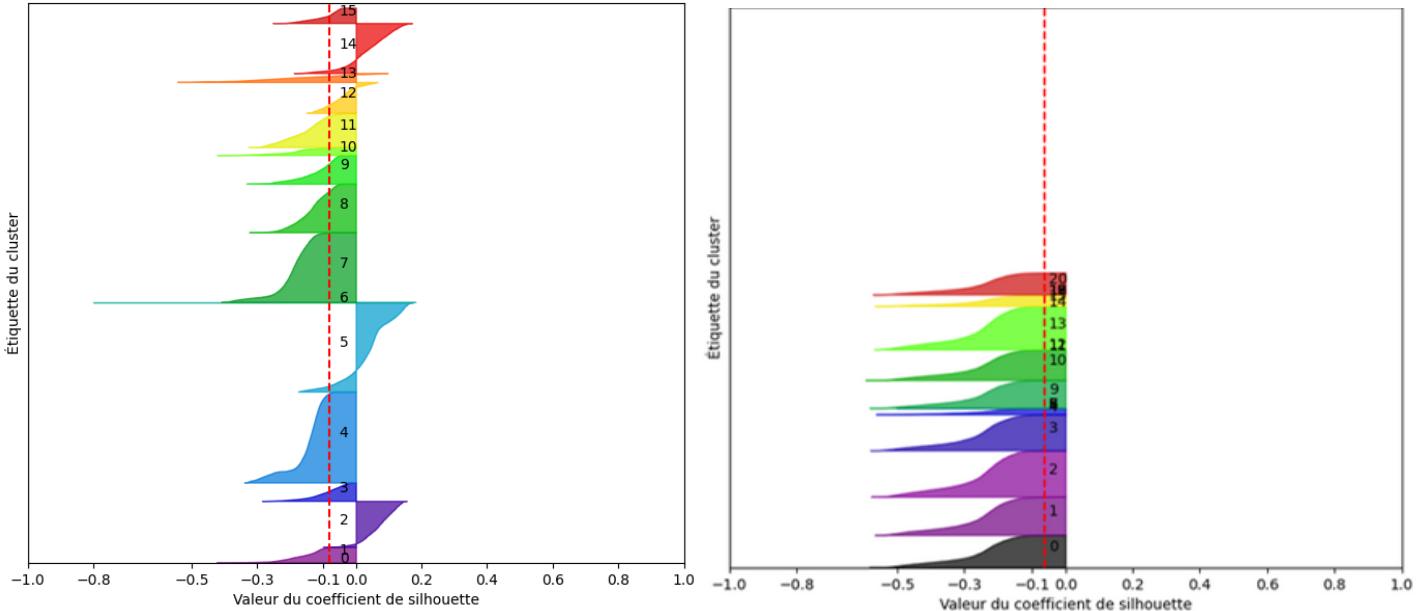
$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

- $s(i) \approx 1 \Rightarrow x_i$  est bien placé (proche de son cluster, loin des autres).
  - $s(i) \approx 0 \Rightarrow x_i$  est sur une frontière entre clusters.
  - $s(i) \approx -1 \Rightarrow x_i$  est probablement mal classé.
- 
- Avec la **distance MMD**, GEMINI produit des groupes assez **uniformes et sphériques**, rappelant le comportement de K-means. Cela permet une séparation nette, mais limite la détection de formes spectrales plus variées.
  - En comparaison, **Fisher-EM** génère des clusters aux **formes plus diversifiées**, mieux adaptées à la complexité astrophysique des spectres.
  - Les visualisations des **silhouette scores** confirment ce constat :
    - Fisher-EM obtient des scores plus contrastés, reflétant une segmentation fine ;

- GEMINI (MMD) reste plus homogène, mais parfois moins discriminant.

Un exemple visuel illustre bien cette différence : à gauche les silhouettes sous Fisher-EM, à droite celles de GEMINI.

*Figure 5.b : Visualisation des silhouettes scores pour chaque une des classes avec FisherEM (à gauche) et GEMINI à droite*

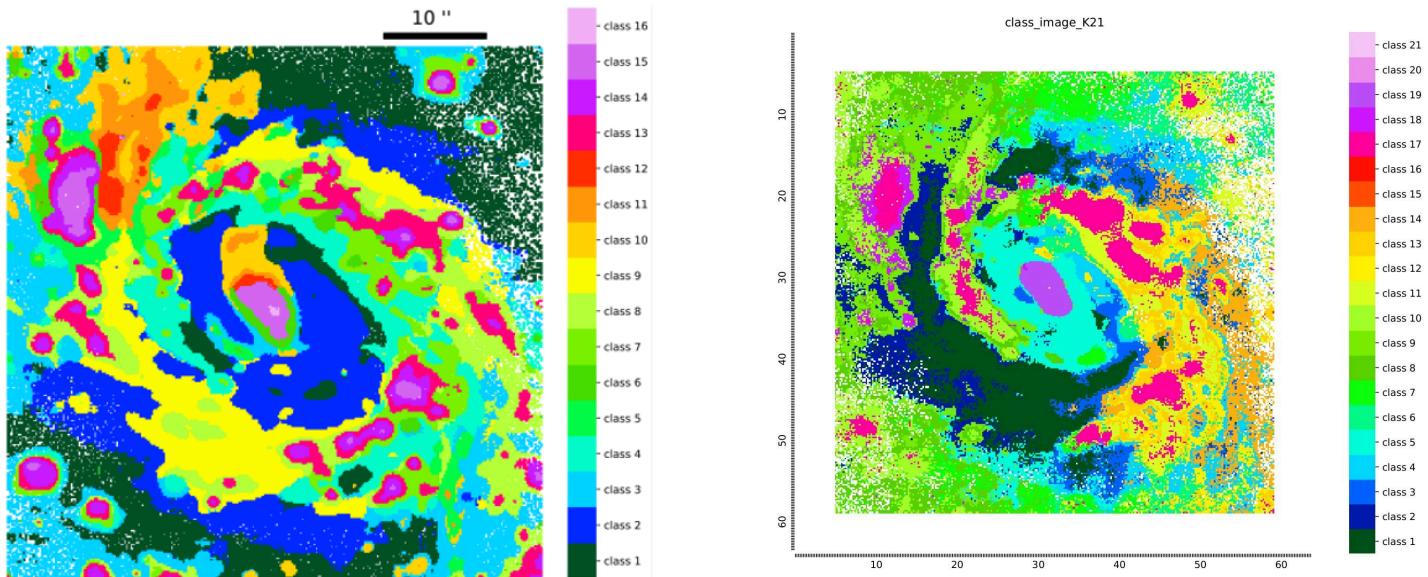


## 5.5. Comparaison avec Fisher-EM

La comparaison entre GEMINI et Fisher-EM a mis en évidence plusieurs points :

- **Robustesse** : Fisher-EM semble mieux capter les variations fines des spectres, tandis que GEMINI produit des partitions plus régulières.
- **Temps de calcul** : GEMINI repose sur des réseaux de neurones, ce qui le rend sensible aux hyperparamètres et parfois plus coûteux à entraîner. Cependant, il bénéficie d'implémentations optimisées sur GPU.
- **Interprétation scientifique** : Fisher-EM, basé sur des modèles statistiques explicites, offre une interprétabilité plus directe (classes gaussiennes discriminantes). À l'inverse, GEMINI est plus “boîte noire”, nécessitant des outils supplémentaires pour analyser les regroupements.

Figure 5.c : Cartes de classes obtenues : approche statistique (FisherEM) vs deep learning (GEMINI)



## 5.6. Discussion critique : apports et limites du deep learning

L'exploration de GEMINI a montré que les approches deep learning ouvrent de nouvelles perspectives pour la classification de spectres astrophysiques :

- **Apports :**

- Introduction de distances plus riches (MMD, Wasserstein), adaptées à la nature distributionnelle des spectres.
- Possibilité de capturer des structures complexes que les modèles gaussiens ne décrivent pas toujours bien.
- Intégration facilitée avec des pipelines modernes de deep learning (PyTorch).

- **Limites :**

- Résultats sensibles au **choix des hyperparamètres** (dimension cachée, taux d'apprentissage, type de noyau).
- Tendance à générer des clusters trop homogènes avec MMD.
- Interprétation moins immédiate que celle de Fisher-EM, ce qui complique la validation scientifique des classes.

En conclusion, GEMINI constitue un **complément** intéressant à Fisher-EM :

- Fisher-EM reste privilégié pour sa robustesse, son efficacité et son interprétabilité ;
- GEMINI, surtout avec la distance Wasserstein, ouvre la voie à une meilleure capture de la complexité des spectres, au prix d'une complexité computationnelle accrue.

## 5.7. Lien entre interprétation physique et continuité du projet

### ❖ *Interprétation MMD*

NGC1068 est une galaxie spirale possédant un noyau actif. Ses bras spiraux sont le lieu d'une intense formation d'étoiles dans des régions appelées HII. Le jet issu du noyau actif ionise une partie de la galaxie caractérisée par une forte intensité de la raie OIII à 5007 Angstroms. La classification des spectres optiques normalisés puis standardisés avec MMD permet de retrouver la morphologie spirale de cette galaxie ainsi que la dichotomie entre le noyau actif (*classe 19 de la figure 5.b de droite*) et le reste de la galaxie. Cependant, la zone d'ionisation du jet n'est pas caractérisée par une ou des classes. De plus, des régions de formation d'étoiles sont trouvées notamment avec la *classe 17* mais elles n'apparaissent qu'en région composite dans le diagramme BPT et non pas en région HII. Enfin, l'aspect visuel de la carte des classes montre des classes bruitées. La classification MMD possède donc des aspects physiques mais échoue à donner une vision précise et globale de **NGC1068**.

### ❖ *Wasserstein*

*Les résultats présentés ci-dessous ont été obtenus ultérieurement par M. Hugo Chambon, après la fin de mon stage.*

#### → Constat :

Le calcul de la distance de Wasserstein en haute dimension avec un grand nombre d'échantillons est un challenge computationnel. En effet, les implémentations traditionnelles ont une complexité temporelle en  $O(n^3)$  et une forte complexité spatiale. La méthode implementée par défaut dans Gemini utilise l'approximation linéaire d'une distribution en somme de dirac mais résout le problème linéaire complet (**fonction ot.lp.emd2 de la librairie pot**). Cette fonction ne converge pas, Gemini Wasserstein n'est donc pas adaptée à la classification de l'échantillon de spectres de **NGC1068**.

#### → Solution :

La distance de Wasserstein peut être approchée à partir de la divergence de Sinkhorn avec une complexité temporelle et spatiale en  $O(n^2)$ . Parmi les différentes implémentations de la méthode de Sinkhorn, le package **GeomLoss** réussit avec son implémentation online et l'utilisation des librairies PyKeops et Pytorch à réduire la complexité spatiale et effectuer sous GPU l'estimation de la distance de Wasserstein. Cette transition permet de gagner un facteur 5 en temps de calcul (en convergeant) et de l'ordre de 10 pour la mémoire. Physiquement, les premiers résultats montrent une capacité à discriminer les zones à cinématique complexe proches du noyau galactique ce qui est une première.

# 6. Troisième étape : Réécriture et optimisation de Fisher-EM en C++

## 6.1. Objectifs de la réécriture

Cette troisième étape n'était pas initialement prévue dans le cahier des charges du stage.

En effet, après avoir terminé plus rapidement que prévu les deux premières parties du projet (classification des spectres SDSS avec Fisher-EM et exploration de GEMINI/GEMCLUS), une nouvelle mission m'a été proposée directement par **le créateur de l'algorithme Fisher-EM (Charles BOUVEYRON)**.

L'objectif était alors de **réimplémenter Fisher-EM en C++**, afin de :

- réduire les temps de calcul pour le traitement massif des spectres SDSS,
- exploiter des bibliothèques numériques performantes (**Armadillo, BLAS/LAPACK**),
- intégrer le code à l'environnement R via **Rcpp**,
- et préparer une version optimisée, autonome et portable du code.

## 6.2. Étapes majeures réécrites en C++

Trois composants essentiels de l'algorithme ont été recodés :

- **E-step (Expectation step)**
  - Calcul des responsabilités a posteriori de chaque observation vis-à-vis des classes.
  - Implémenté en C++ avec Armadillo pour les multiplications matricielles et l'exponentiation des densités.
- **M-step (Maximization step)**
  - Mise à jour des paramètres du modèle : moyennes, covariances, poids des classes.
  - Optimisation des inversions et factorisations avec Armadillo, en minimisant les copies mémoire.
- **F-step (Fisher step)**
  - Calcul de l'espace discriminant de Fisher (matrices inter-classes et intra-classes).
  - Résolution du problème généralisé aux valeurs propres pour obtenir le sous-espace discriminant.

Chaque étape a été encapsulée en fonctions C++ appelables indépendamment depuis R via **Rcpp**, permettant de tester et valider chacune séparément.

### 6.3. Benchmarks indépendants

Pour évaluer les performances, chaque étape (E-step, M-step, F-step) a été **benchmarquée individuellement** avec différents jeux de données.

- Les scripts R ont servi de référence pour comparer les temps d'exécution.
- Les tests ont été effectués avec **microbenchmark** côté R, en appelant les fonctions R et C++ correspondantes.
- Les résultats montrent systématiquement un **gain significatif en temps de calcul** pour la version C++.

Figure 6.a : Les résultats benchmarks de chaque étape

<u>e-step :</u> Unit: milliseconds							
expr	min	lq	mean	median	uq	max	neval
R	1.670602	1.986501	2.597026	2.500451	2.958301	4.440501	40
Cpp	1.067701	1.516901	1.861256	1.741601	2.131800	3.495901	40

<u>m-step :</u> Unit: microseconds							
expr	min	lq	mean	median	uq	max	neval
Mstep_cpp	61.601	72.5515	124.846	92.9515	110.2505	1264.301	40
Mstep_r	181.701	251.0010	440.076	290.6010	364.8510	1995.901	40

<u>f-step :</u> Unit: microseconds							
expr	min	lq	mean	median	uq	max	neval
R	1644.4	1737.2510	1959.388	1858.7510	2098.351	3157.001	40
Cpp	800.3	887.2015	1020.226	980.3005	1103.401	1798.201	40

### 6.4. Validation partielle des résultats

- Chaque étape prise séparément fournit des **résultats cohérents** avec l'implémentation R :
  - mêmes log-vraisemblances locales,
  - mêmes responsabilités et paramètres après itérations,
  - même espace discriminant dans la F-step.
- Cela confirme la **validité numérique** des trois blocs indépendants.

## 6.5. Tentative d'intégration complète

L'objectif final était d'assembler les trois étapes dans une **version complète de Fisher-EM en C++**.

- La structure de boucle EM a été codée (alternance E-step → M-step → F-step).
- L'appel intégré via Rcpp devait remplacer l'appel à la fonction `fstep.fisher` du package R.

Cependant, lors de l'exécution intégrale du package, une **erreur d'exécution critique** survient, empêchant le code de converger.

## 6.6. Complications rencontrées

Malgré la réussite des tests unitaires sur chaque étape séparée, l'assemblage complet du package a posé problème :

- **Erreur au moment de l'intégration complète** → plantage du programme lors des premières itérations.
- Hypothèses principales :
  - Problème de **dimensionnement des matrices** entre étapes,
  - **Incompatibilités de version** entre Armadillo et Rcpp,
  - ou encore gestion mémoire incorrecte lors de l'enchaînement des étapes.

Ces difficultés n'ont pas pu être totalement résolues dans le cadre du stage, mais les résultats obtenus sur les benchmarks individuels démontrent clairement la **faisabilité et l'intérêt de la réécriture en C++**.

# 7. Discussion générale

## 7.1. Synthèse des résultats obtenus

Ce stage a permis d'explorer de manière progressive et comparative plusieurs méthodes de **classification non supervisée de spectres galactiques**.

- Dans un premier temps, l'algorithme **Fisher-EM** a été appliqué à un jeu de données issu du **SDSS (~300 000 spectres)**. L'approche a permis d'identifier **362 classes distinctes**, confirmant la capacité du modèle à révéler une grande diversité de structures spectrales tout en mettant en évidence l'importance de la **standardisation des données** le comparant ainsi avec les 86 classes trouvées ([M. Fraix-Burnet et al., 2021](#)).

- Ensuite, l'exploration de **GEMINI/GEMCLUS** a introduit une perspective nouvelle en mobilisant des outils de **deep learning** et des **distances de distributions** (MMD, Wasserstein). L'application sur la galaxie **NGC 1068** a permis de comparer directement les résultats avec ceux obtenus par Fisher-EM ([Chambon & Fraix-Burnet, 2023](#)). Si GEMINI produit des regroupements plus uniformes, l'intégration de la distance Wasserstein ouvre des perspectives prometteuses pour la détection de structures complexes.
- Enfin, une étape complémentaire, proposée par l'auteur de Fisher-EM lui-même, a été consacrée à la **réécriture de l'algorithme en C++**. Les trois étapes majeures (E-step, M-step, F-step) ont été réimplémentées indépendamment, avec succès et des **gains de performances mesurables** via microbenchmarks. Toutefois, une difficulté subsiste concernant l'intégration finale du package R/C++, probablement liée à des incompatibilités de gestion mémoire ou de bibliothèques (Armadillo, Rcpp).

## 7.2. Comparaison des approches : statistique vs deep learning

- **Fisher-EM (approche statistique) :**
  - Points forts : stabilité, bonne interprétation scientifique, capacité à détecter des structures variées et non sphériques, résultats cohérents avec l'état de l'art.
  - Limites : temps de calcul important, dépendance à la réduction de dimension, sensibilité au prétraitement des données.
- **GEMINI/GEMCLUS (approche deep learning) :**
  - Points forts : flexibilité, utilisation de distances puissantes (MMD, Wasserstein), efficacité sur de grands jeux de données grâce aux réseaux de neurones.
  - Limites : hyperparamètres nombreux et difficiles à régler, résultats parfois trop uniformes, interprétation scientifique moins directe.

La complémentarité des deux approches est manifeste : **Fisher-EM offre une base robuste et interprétable**, tandis que **GEMINI ouvre la voie à des modèles plus flexibles et scalables**.

## 7.3. Apports méthodologiques du stage

Ce stage a permis plusieurs avancées méthodologiques :

- Une **chaîne complète d'analyse** a été construite, depuis les **prétraitements (standardisation, configuration d'hyperparamètres)** jusqu'à la **classification et l'évaluation des résultats**.
- La **comparaison systématique** entre méthodes statistiques classiques et approches récentes de deep learning a apporté un éclairage précieux sur leurs avantages et leurs limites respectifs.

- La **réécriture en C++** illustre une démarche d'optimisation logicielle visant la **scalabilité** des méthodes appliquées à de grands jeux de données astronomiques.
- L'utilisation de **clusters de calcul** a permis de manipuler efficacement les spectres en haute dimension, renforçant la dimension pratique et opérationnelle du projet.

## 7.4. Limites rencontrées et solutions envisagées

Plusieurs limites et obstacles ont été identifiés au cours du stage :

- **Standardisation** : bien que nécessaire, elle peut masquer certaines informations astrophysiques pertinentes. Une piste serait d'explorer des méthodes de normalisation alternatives (par ex. robust scaling, quantile normalization).
- **Complexité computationnelle** : Fisher-EM reste coûteux sur de très grands jeux de données. La poursuite du développement C++ et l'intégration GPU (CUDA, OpenCL) pourraient fortement améliorer les performances.
- **Interprétation scientifique des clusters GEMINI** : les regroupements produits sont difficiles à relier directement aux classifications astrophysiques établies. Un travail futur consisterait à coupler GEMINI avec des bases de connaissances astrophysiques (par ex. classifications morphologiques de galaxies).
- **Problème technique de la réécriture en C++** : le plantage lors de l'intégration finale indique probablement un conflit entre bibliothèques. Une solution envisagée est de tester différentes versions d'Armadillo/Rcpp ou d'implémenter une gestion mémoire plus explicite (allocation/désallocation).

# 8. Conclusion et perspectives

## 8.1. Bilan scientifique et technique du stage

Au cours de ce stage, j'ai pu approfondir mes connaissances théoriques et pratiques sur le traitement de données en haute dimension, et plus particulièrement sur les méthodes de standardisation. J'ai mis en œuvre des techniques de prétraitement adaptées, évalué leur impact sur les performances des modèles, et développé des compétences en programmation, en analyse numérique ainsi qu'en interprétation de résultats expérimentaux. Ce travail m'a permis d'acquérir une vision plus globale des enjeux liés à la qualité des données et à la robustesse des algorithmes en contexte réel.

## 8.2. Importance de la standardisation en haute dimension

La standardisation constitue une étape cruciale dans l'analyse statistique et l'apprentissage automatique, en particulier lorsque les données sont de grande dimension. Elle permet d'éviter les biais liés aux échelles hétérogènes, de stabiliser les algorithmes d'optimisation, et de garantir une meilleure comparabilité entre les variables. Dans le cas étudié, les résultats obtenus confirment que la

qualité de la standardisation influence directement la performance des modèles, soulignant ainsi son rôle central dans tout processus de modélisation.

### 8.3. Perspectives de recherche futures

Plusieurs pistes de recherche peuvent être envisagées à la suite de ce travail :

- **Optimisation** : développer des méthodes de standardisation plus efficaces et adaptées aux contraintes computationnelles de la très grande dimension.
- **Extension à d'autres types de données** : appliquer et tester ces méthodes sur des données structurées complexes (textes, images, séries temporelles).
- **Évaluation comparative** : approfondir l'étude de l'impact des différentes techniques de normalisation et standardisation sur divers modèles prédictifs.

Ces perspectives ouvrent la voie à une amélioration continue des pratiques de prétraitement et à une meilleure intégration des méthodes dans des environnements appliqués.

---

### Références citées dans le rapport

- Chambon & Fraix-Burnet (2023) : [\*Spectral similarities in galaxies through an unsupervised classification of spaxels\*](#)
- Fraix-Burnet et al. (2021) : [\*Unsupervised classification of SDSS galaxy spectra\*](#)

### Lectures complémentaires

- Feroz, F., Hobson, M. P., Cameron, E., & Pettitt, A. N. (2021). *Importance Nested Sampling and the MultiNest Algorithm*. *Astronomy & Astrophysics*, 649, A53.  
→ [Le lien vers l'article](#)
  - Chavas, J.-P. (2010). *Entropie, information et complexité*. HAL archives-ouvertes.  
→ [Le lien vers l'article](#)
  - Ohl, L., Mattei, P.-A., Bouveyron, C., Harchaoui, W., Leclercq, M., Droit, A., & Precioso, F. (2022). *Generalised Mutual Information for Discriminative Clustering*. In *NeurIPS 2022*.  
→ [Le lien vers l'article](#)
  - Leinster, T. (2020). *Entropy and Diversity: The Axiomatic Approach*. Cambridge University Press.  
→ [Le lien vers l'article](#)
  - Ohl, L., Mattei, P.-A., Bouveyron, C., Leclercq, M., Droit, A., & Precioso, F. (2023). *Sparse and Geometry-aware Generalisation of the Mutual Information for Joint Discriminative Clustering and Feature Selection*.  
→ [Le lien vers l'article](#)
-

## Annexes techniques

- Package Gemclus : [GEMINI-Clustering](#)
  - Repository Github de la partie FisherEM : [FisherEM-Clustering](#)
  - Repository Github de la partie GEMINI : [GEMINI-Clustering](#)
  - FisherEM algorithm : [FisherEM: The FisherEM Algorithm to Simultaneously Cluster and Visualize High-Dimensional Data](#)
  - Archive FisherEM : [FisherEM package](#)
- 

### ***Message de fin***

*Ce stage m'a permis de développer des compétences à la fois scientifiques et techniques, tout en découvrant la richesse du travail de recherche au sein de l'IPAG. Je tiens à exprimer une nouvelle fois toute ma gratitude envers mes encadrants, M. Didier Fraix-Burnet et M. Hugo Chambon, pour leur disponibilité, leurs conseils précieux et leur accompagnement constant. Enfin, je remercie l'ensemble de l'équipe du laboratoire pour leur accueil chaleureux et leur soutien durant cette expérience enrichissante.*