Apprentissage de réseaux géniques, de l'inférence au clustering

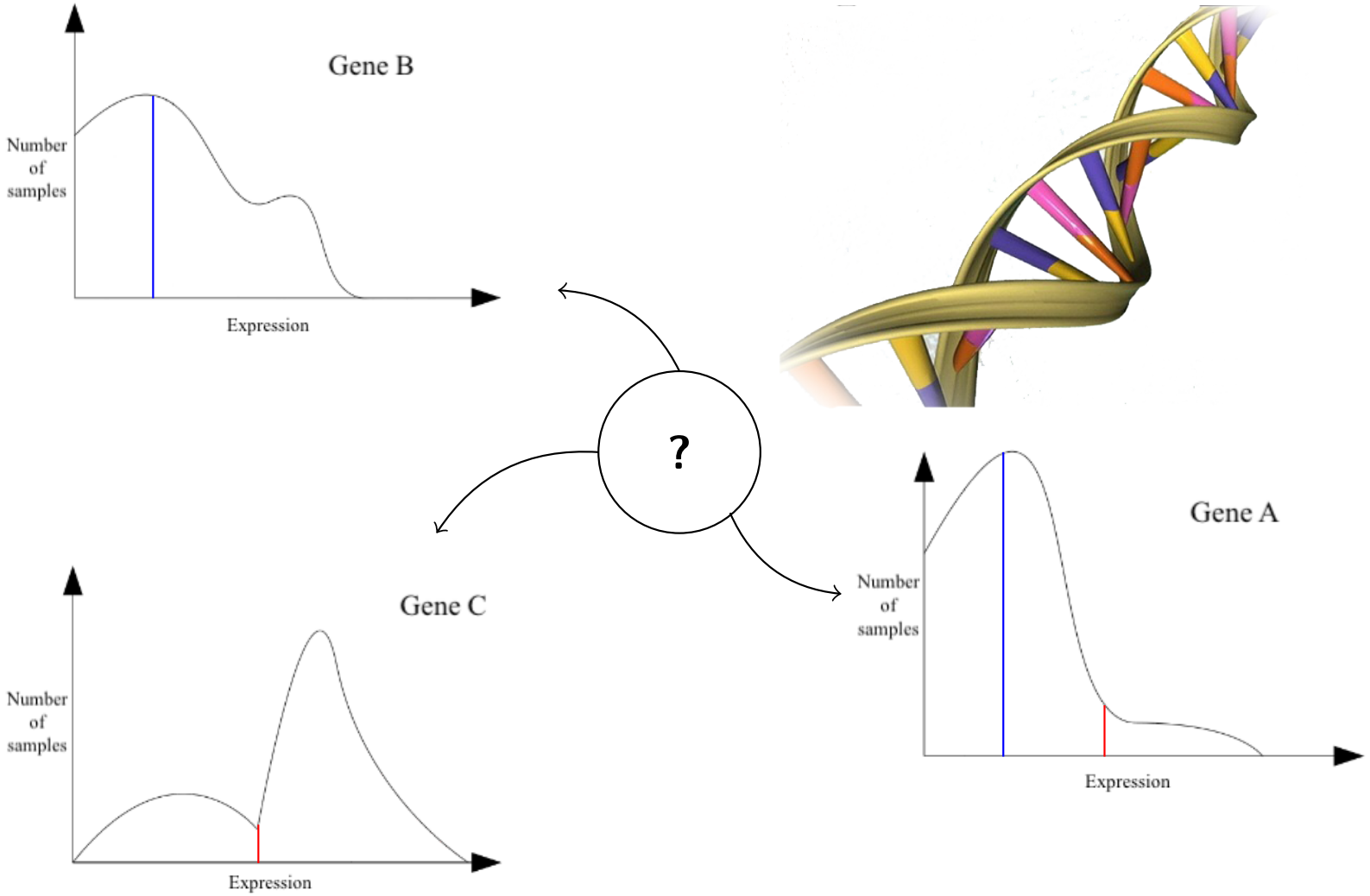
Magali Champion

Université Paris Cité

27-29/11/2023

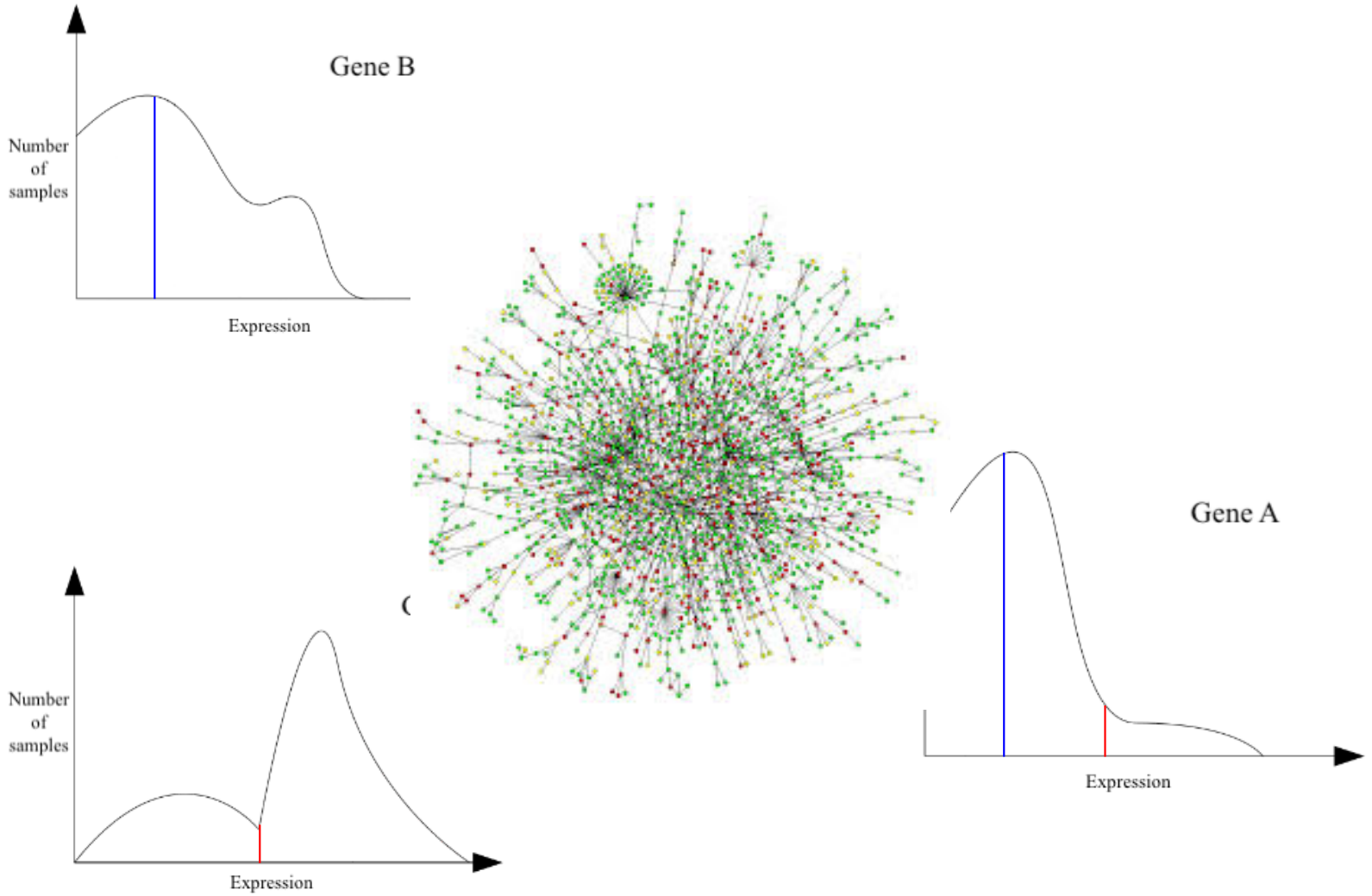
Introduction

Réseaux de régulation de gènes



Introduction

Réseaux de régulation de gènes



Introduction

Réseaux de régulation de gènes

Motivations :

- Retrouver le réseau de régulation de gènes $\mathcal{G}$ qui modélise les interactions entre un ensemble de gènes donnés, Inférence de réseaux
- Explorer le réseau de régulation de gènes $\mathcal{G}$ pour découvrir des groupes de gènes connectés et comprendre le développement de maladies.

Clustering de réseaux

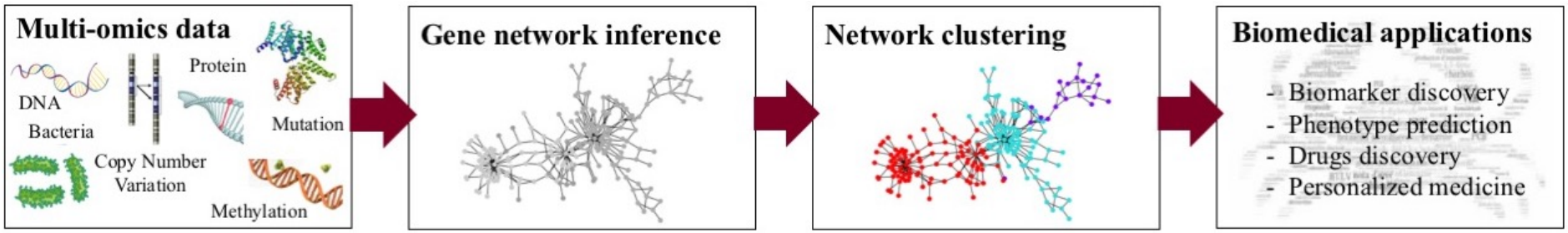
Introduction

Réseaux de régulation de gènes

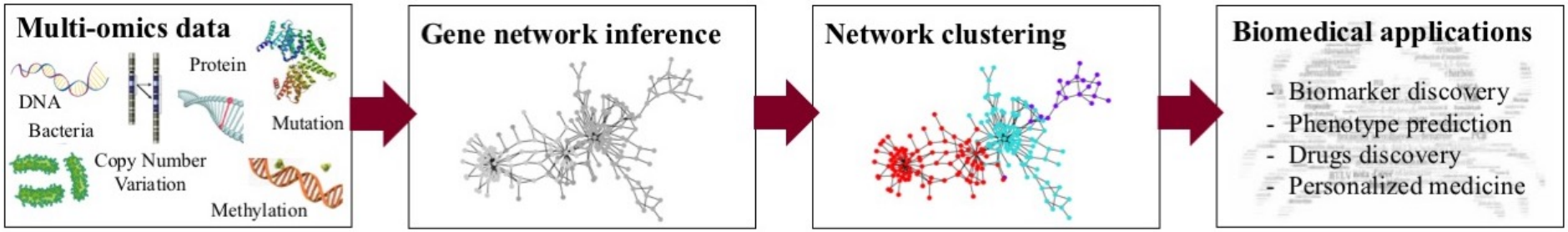
Motivations :

- Retrouver le réseau de régulation de gènes $\mathcal{G}$ qui modélise les interactions entre un ensemble de gènes donnés, Inférence de réseaux
- Explorer le réseau de régulation de gènes $\mathcal{G}$ pour découvrir des groupes de gènes connectés et comprendre le développement de maladies.

Clustering de réseaux



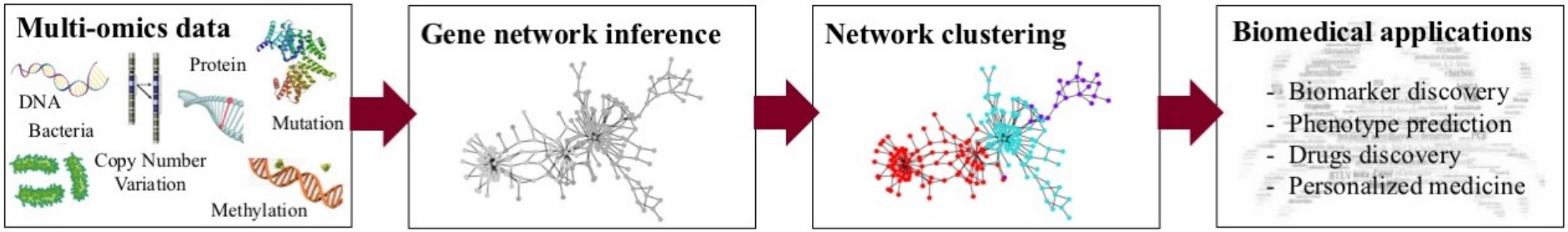
Plan du cours



1. Les réseaux de régulation de gènes
2. Inférence de réseaux
3. Clustering de réseaux
4. Applications

I. Les réseaux de régulation de gènes

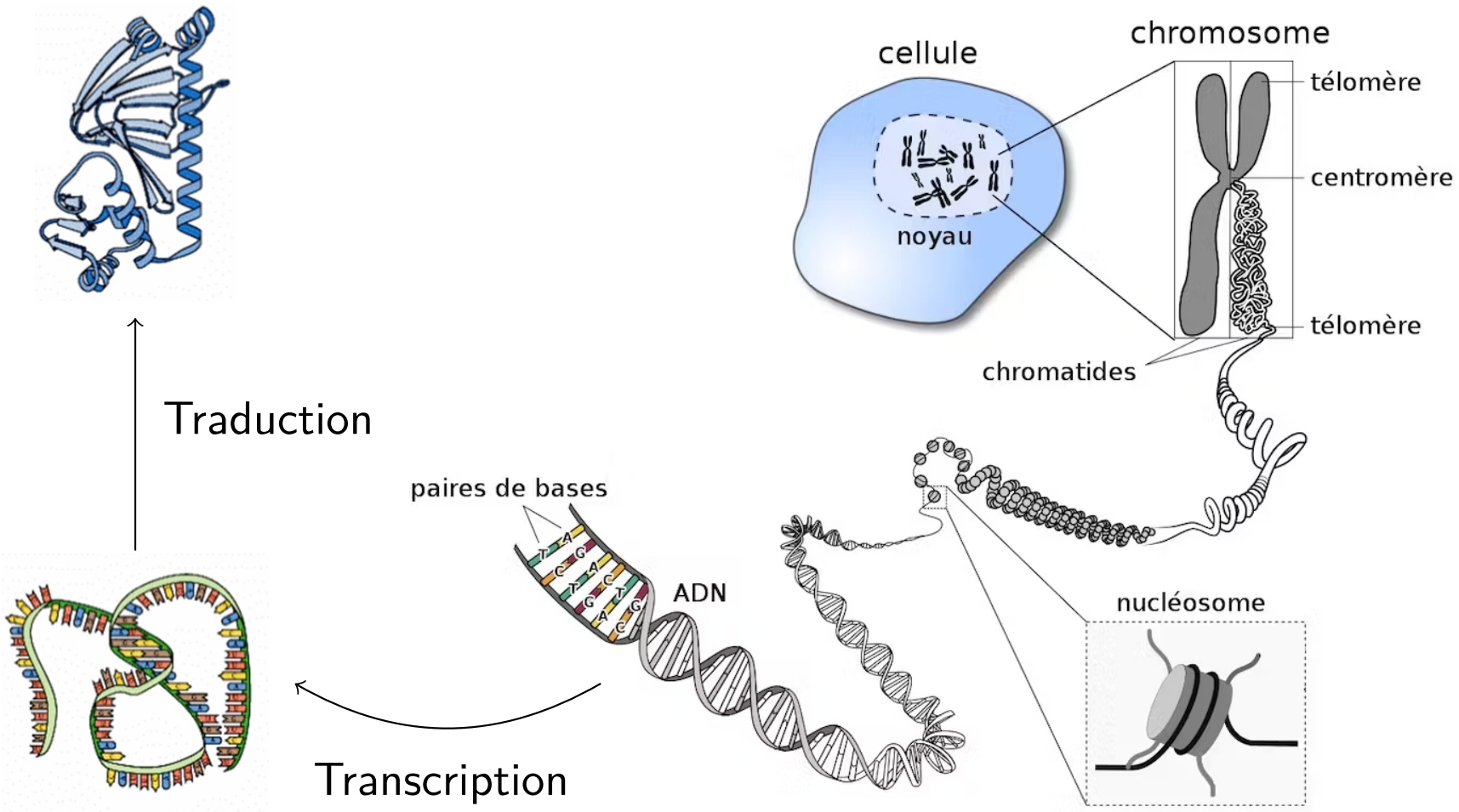
Plan du cours



1. Les réseaux de régulation de gènes
   - Origine des données
   - Mécanismes de régulation
   - Modélisation des réseaux
   - Enjeux
2. Inférence de réseaux
3. Clustering de réseaux
4. Applications

I. Les réseaux de régulation de gènes

1. Origine des données



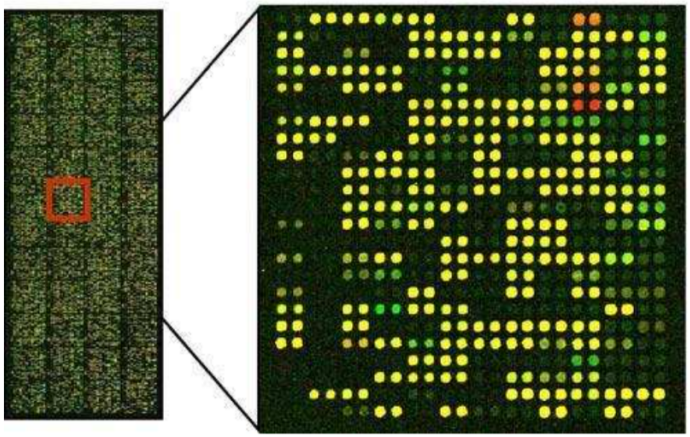
L'**expression d'un gène** correspond à son niveau de transcrits.

I. Les réseaux de régulation de gènes

1. Origine des données

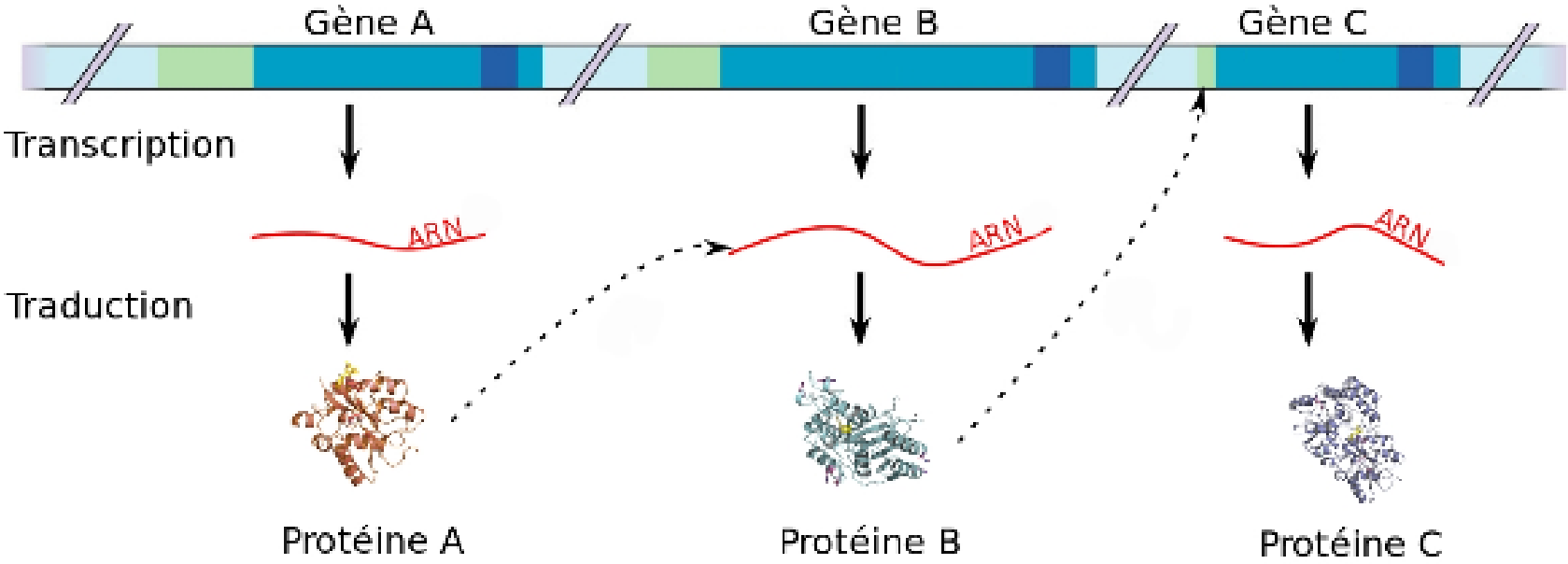
L'expression des gènes peut être mesuré par :

- puces à ADN (*microarray*),
- séquençage à haut débit (*RNA-seq*),
- séquençage de cellule unique (*single-cell*).



I. Les réseaux de régulation de gènes

2. Mécanismes de régulation

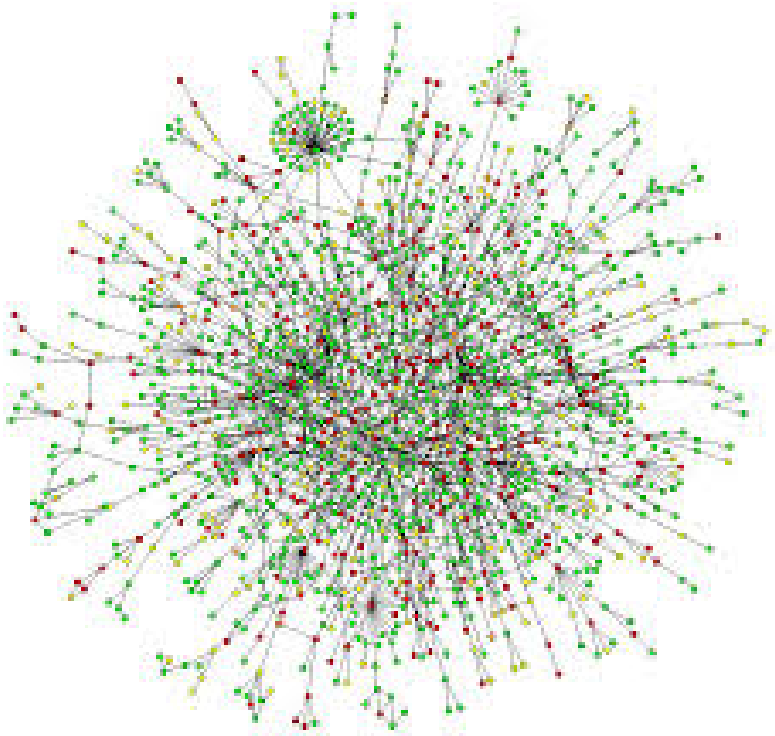


I. Les réseaux de régulation de gènes

3. Modélisation des réseaux

Les réseaux de régulations de gènes sont modélisés sous la forme de graphes pour lesquels :

- les nœuds représentent les entités biologiques étudiés, ici les **gènes**,
- les arêtes représentent les **régulations** entre les gènes.

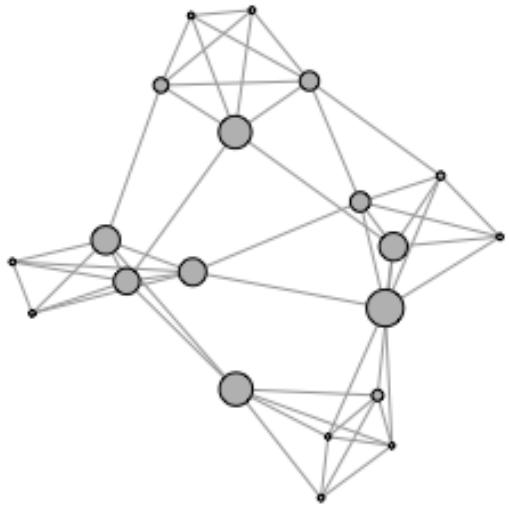


I. Les réseaux de régulation de gènes

3. Modélisation des réseaux

Il existe différents types de graphes permettant de répondre à différents types de problèmes :

- **les graphes non-dirigés**

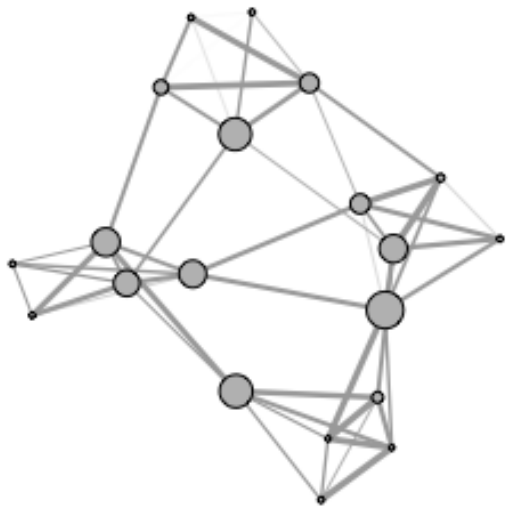


I. Les réseaux de régulation de gènes

3. Modélisation des réseaux

Il existe différents types de graphes permettant de répondre à différents types de problèmes :

- graphes non-dirigés
- **les graphes pondérés**

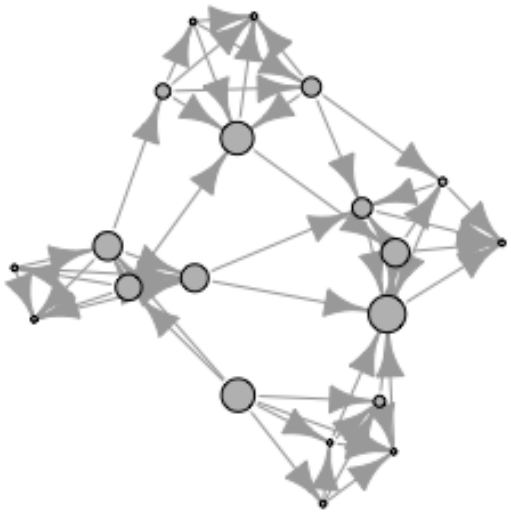


I. Les réseaux de régulation de gènes

3. Modélisation des réseaux

Il existe différents types de graphes permettant de répondre à différents types de problèmes :

- graphes non-dirigés
- les graphes pondérés
- **les graphes orientés**

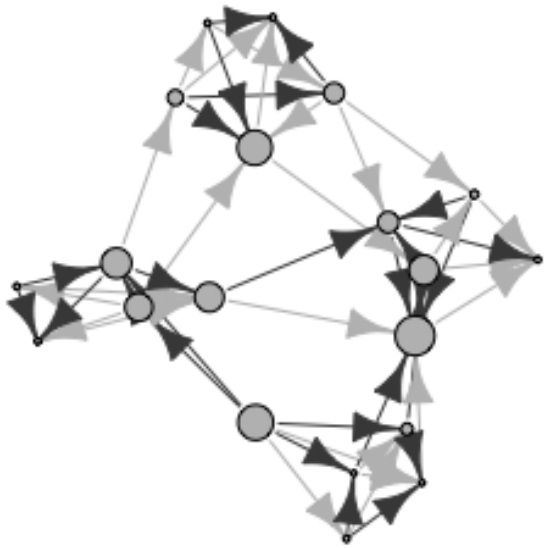


I. Les réseaux de régulation de gènes

3. Modélisation des réseaux

Il existe différents types de graphes permettant de répondre à différents types de problèmes :

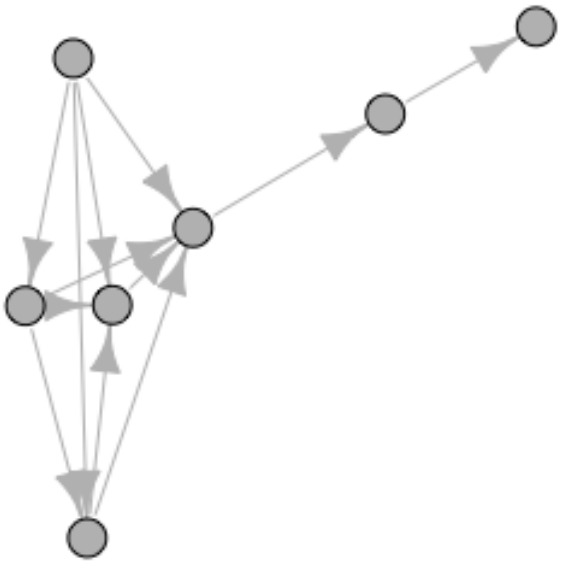
- graphes non-dirigés
- les graphes pondérés
- les graphes orientés
- les **graphes pondérés orientés**

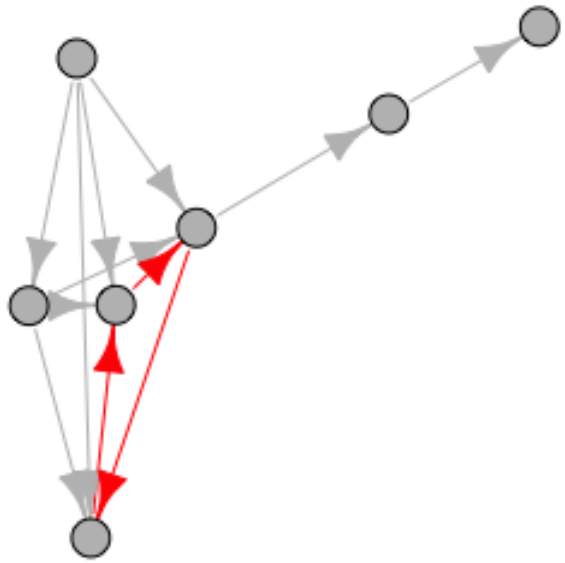


I. Les réseaux de régulation de gènes

3. Modélisation des réseaux

Les **Graphes Acycliques Dirigés** (DAG) sont des graphes qui ne contiennent pas de boucles et sont dirigés. Ce sont des outils particulièrement utilisés pour modéliser des réseaux de régulation de gènes parce qu'ils assurent l'**identifiabilité*** du modèle.





* *L'identifiabilité du modèle permet d'assurer que la régulation entre 2 gènes (effet causal) peut être estimé à partir des observations, ici les données d'expression des gènes.*

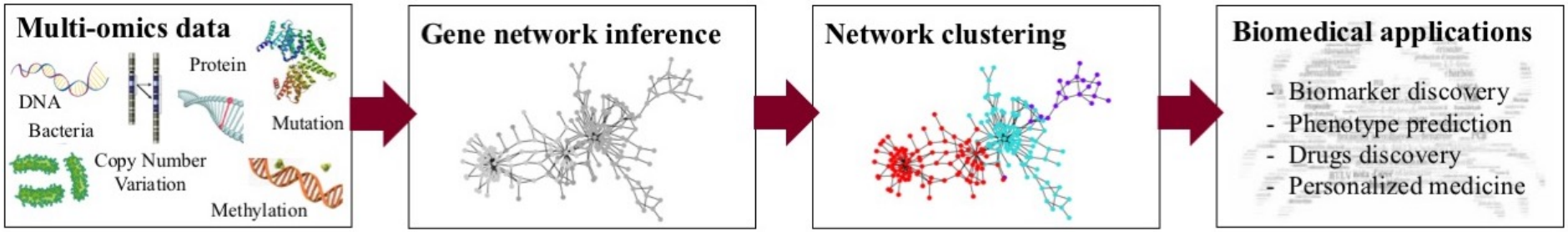
I. Les réseaux de régulation de gènes

4. Enjeux

- La **grande dimension** des données : le nombre de gènes est très supérieur au nombre d'échantillons.
- La **parcimonie** (sparsité) : les gènes d'un réseau de régulation ne sont que peu connectés les uns aux autres.
- La **causalité** : il faut différencier les effets de corrélation des effets causaux.

II. Inférence de réseaux

Plan du cours



1. Les réseaux de régulation de gènes
2. Inférence de réseaux
   - Réseaux de co-expression
   - Méthodes de regréssion
   - Causalité et inférence
   - Métriques d'évaluation
3. Clustering de réseaux
4. Applications

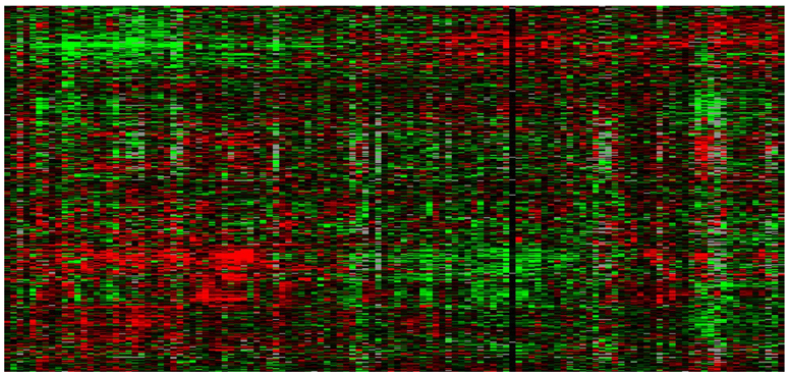
Introduction

Inférer un réseau, qu'est-ce que ça veut dire?

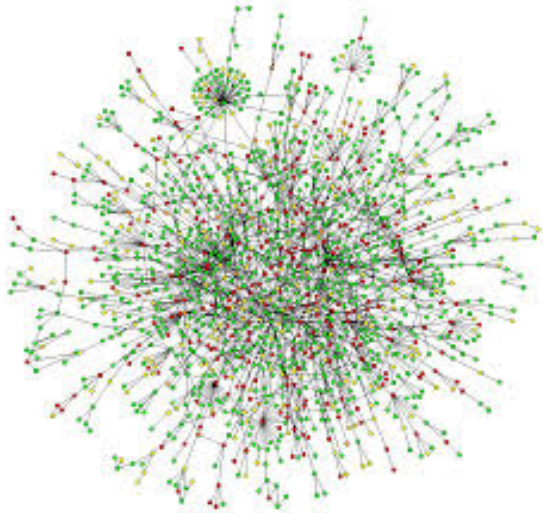
On suppose qu'on a un système biologique constitué de p gènes. On connaît les données d'expression de ces p gènes pour n échantillons, contenues dans la matrice $X \in \mathcal{M}_{n \times p}$.

L'objectif consiste alors à retrouver les interactions qui existent entre les p gènes à l'aide de ces données. Plus précisément, on cherche à reconstruire (*inférer*) le réseau/graphe représentant ces interactions.

Les méthodes d'inférence doivent tenir compte de la grande dimension des données ($n << p$), la parcimonie, la causalité.



Inférence $\longrightarrow$



II. Inférence de réseaux

1. Réseaux de co-expression

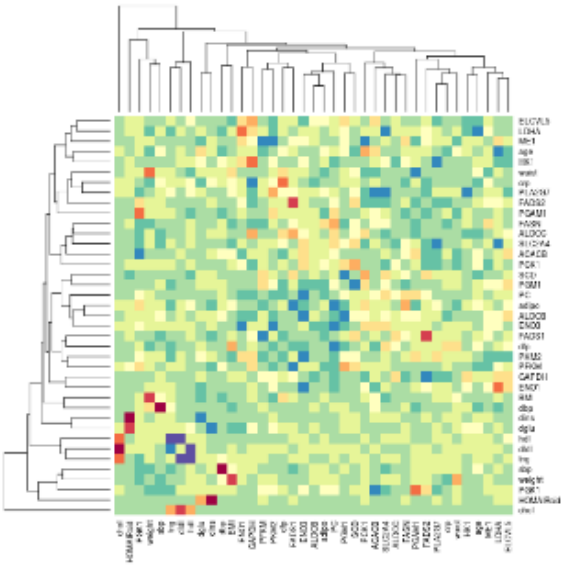
II.1. Réseaux de co-expression

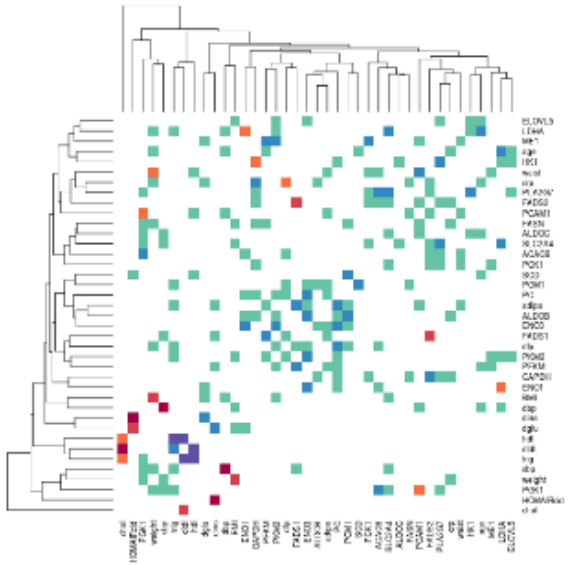
Une méthode naïve

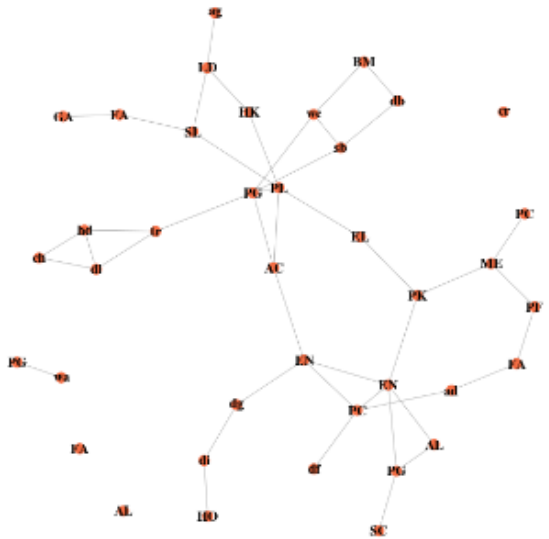
La méthode la plus naïve pour reconstruire un réseau consiste à :

1. calculer la corrélation entre chacun des gènes,
2. seuiller suivant un seuil pré-défini λ,
3. reconstruire le réseau associé.

$$i \leftrightarrow j \Leftrightarrow i \text{ connecté à } j \Leftrightarrow \text{Cor}(X^i, X^j) > \lambda.$$



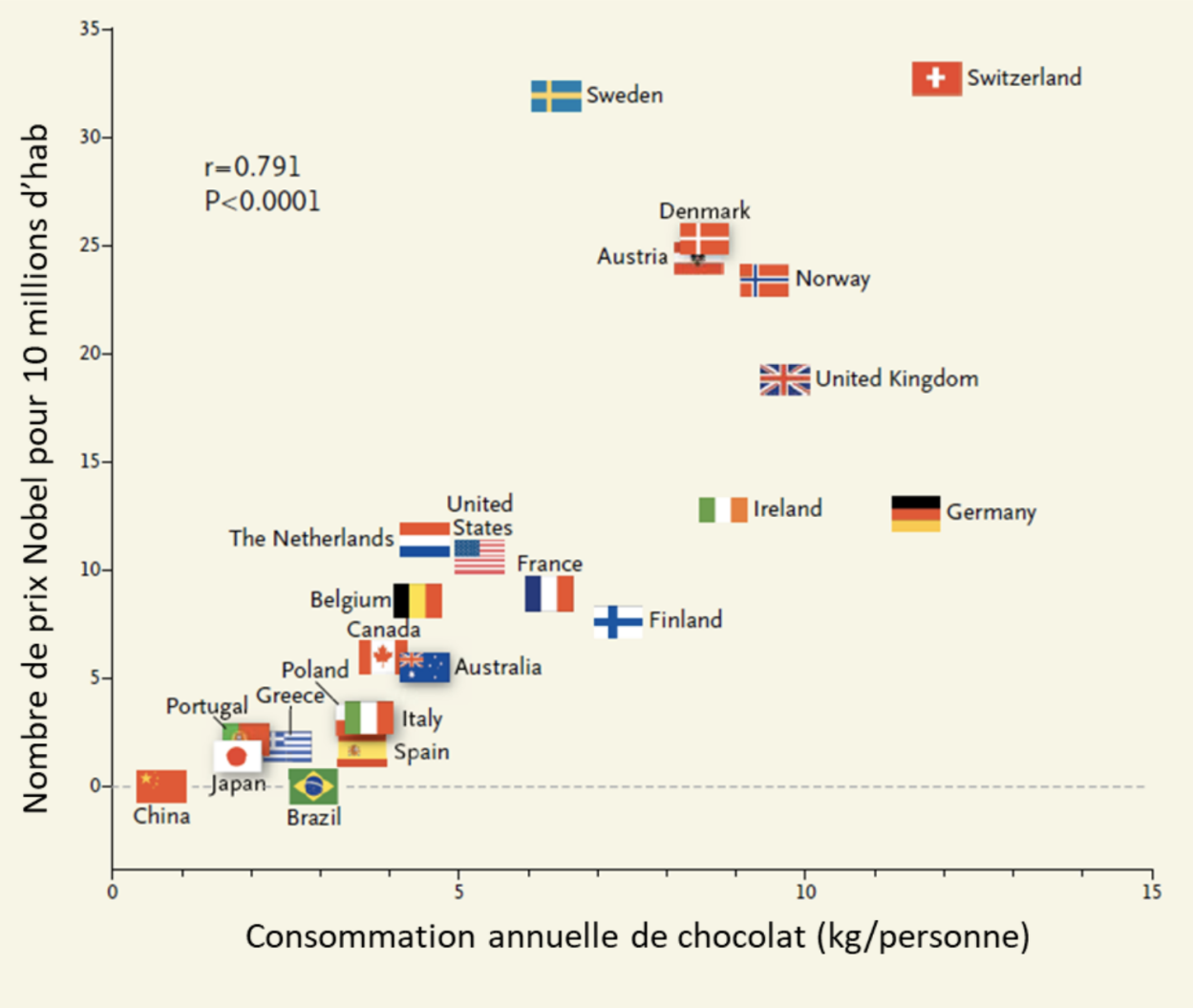




II.1. Réseaux de co-expression

Pourquoi c'est une mauvaise idée?

Corrélation n'est pas causalité!



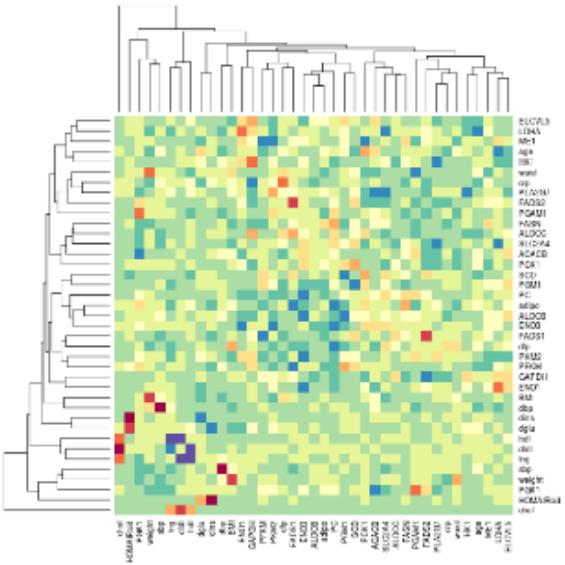
II.1. Réseaux de co-expression

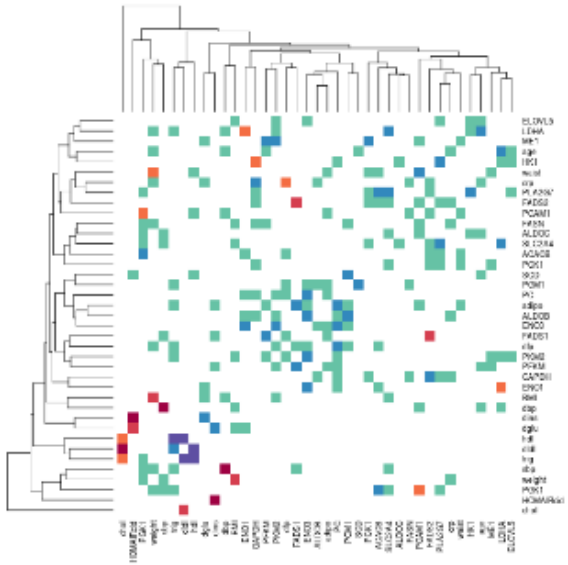
Corrélation partielle

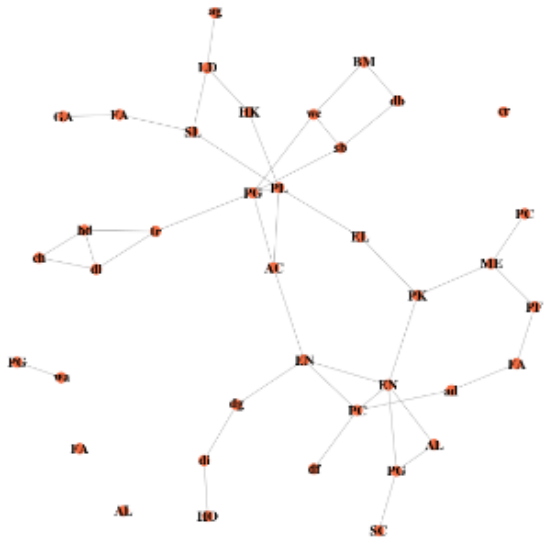
Pour éliminer une partie des relations indirectes, on peut :

1. calculer la corrélation partielle entre chacun des gènes,
2. seuiller suivant un seuil pré-défini λ,
3. reconstruire le réseau associé.

$$i \leftrightarrow j \Leftrightarrow i \text{ connecté à } j \Leftrightarrow \text{Cor}(X^i, X^j | X^k, k \neq i, j) > \lambda.$$







II.1. Réseaux de co-expression

Pour aller plus loin

Soit $\mathcal{G} = (V, E)$ un graphe où :

- $V = \{1, ..., p\}$ est l'ensemble des nœuds,
- E est l'ensemble d'arêtes.

On suppose que les données d'observation $X \in \mathcal{M}_{n \times p}$ vérifient :

$$X = (X^1, ..., X^p) \sim \mathcal{N}_p(0, \Sigma).$$

Définition

Le graphe $\mathcal{G}$ satisfait la ***propriété de Markov par paire*** *si :*

$$(i, j) \notin E \Rightarrow X^i \perp X^j | X^{V \setminus \{i,j\}}.$$

Tout couple de sommets non adjacents sont indépendants conditionnellement à tous les autres sommets.

II.1. Réseaux de co-expression

Modèles graphiques gaussien

On dit que $\mathcal{G}$ est un **modèle graphique gaussien** si $\mathcal{G}$ satisfait la propriété de Markov.

Proposition

Si $\mathcal{G}$ est un modèle graphique gaussien :

$$(i,j) \notin E \Leftrightarrow \Theta_{i,j} = 0,$$

où $\Theta = \Sigma^{-1}$ est la matrice de précision associée à X.

(Lauritzen, 1996)

La matrice de précision représente le motif de parcimonie du graphe.

II.1 Réseaux de co-expression

Lien avec la corrélation partielle

Proposition

Soit $\Theta = (\Theta_{i,j})_{1 \leq i,j \leq p}$ *la matrice de précision associée à* X*. La corrélation partielle entre* X^i *et* X^j *sachant tous les autres nœuds* X^k $(k \neq i, j)$ *est alors donnée par :*

$$\rho_{X^i, X^j | V \setminus \{i,j\}} = \frac{-\Theta_{i,j}}{\sqrt{\Theta_{i,i}\Theta_{j,j}}}.$$

II.1 Réseaux de co-expression

Estimer la matrice de précision

Le problème devient plus simple :

1. estimer la matrice de covariance,
2. l'inverser pour calculer la matrice de précision,
3. reconstruire le réseau associé.

$$i \leftrightarrow j \Leftrightarrow i \text{ connecté à } j \Leftrightarrow \Theta_{i,j} \neq 0.$$

II.1 Réseaux de co-expression

Estimer la matrice de précision

Le problème devient plus simple :

1. estimer la matrice de covariance,
2. l'inverser pour calculer la matrice de précision,
3. reconstruire le réseau associé.

$$i \leftrightarrow j \Leftrightarrow i \text{ connecté à } j \Leftrightarrow \Theta_{i,j} \neq 0.$$

Mais ce n'est pas toujours facile dans le cadre de la grande dimension!

II. Inférence de réseaux

2. Méthodes de régression

II.2. Méthodes de régression

Pourquoi parler de régression?

Résultats de Friedman (2009).

II.2. Méthodes de régression

Pourquoi parler de régression?

Résultats de Friedman (2009).

$$i \leftrightarrow j \Leftrightarrow i \text{ connecté à } j \Leftrightarrow \beta_i^j \neq 0 \text{ ou } \beta_j^i \neq 0,$$

où β_i^j est le i-ième coefficient issu de la régression linéaire entre X^j et toutes les autres variables :

$$X^j = \sum_{i=1, i\neq j}^{p} \beta_i^j X^i + \varepsilon.$$

II.2. Méthodes de régression

Pourquoi parler de régression?

Résultats de Friedman (2009).

$$i \leftrightarrow j \Leftrightarrow i \text{ connecté à } j \Leftrightarrow \beta_i^j \neq 0 \text{ ou } \beta_j^i \neq 0,$$

où β_i^j est le i-ième coefficient issu de la régression linéaire entre X^j et toutes les autres variables :

$$X^j = \sum_{i=1, i\neq j}^{p} \beta_i^j X^i + \varepsilon.$$

Et ça, on sait faire!

II.2. Méthodes de régression

Modèle linéaire gaussien

Soit $\mathcal{G} = (V, E)$ un graphe où :

- $V = \{1, ..., p\}$ est l'ensemble des nœuds,
- E est l'ensemble d'arêtes.

On suppose que les données d'observation $X \in \mathcal{M}_{n \times p}$ vérifient :

$$X = (X^1, ..., X^p) \sim \mathcal{N}_p(0, \Sigma).$$

Pour inférer le graphe, on estime les paramètres du modèle linéaire :

$$\forall j \in \{1, ..., p\}, \quad X^j = \sum_{i=1, i \neq j}^{p} \beta_i^j X^i + \varepsilon.$$

II.2. Méthodes de régression

Modèle linéaire gaussien

Soit $\mathcal{G} = (V, E)$ un graphe où :

- $V = \{1, ..., p\}$ est l'ensemble des nœuds,
- E est l'ensemble d'arêtes.

On suppose que les données d'observation $X \in \mathcal{M}_{n \times p}$ vérifient :

$$X = (X^1, ..., X^p) \sim \mathcal{N}_p(0, \Sigma).$$

Pour inférer le graphe, on estime les paramètres du modèle linéaire :

$$\forall j \in \{1, ..., p\}, \quad X^j = \sum_{i=1, i \neq j}^{p} \beta_i^j X^i + \varepsilon.$$

$\rightarrow$ On privilégie les méthodes **sparses**!

(Meinshausen et Bühlmann, 2006)

II.2. Méthodes de régression

Méthodes pénalisées

Pour simplifier, on considère le modèle linéaire gaussien :

$$Y = X\beta + \varepsilon,$$

où $Y = X^j$ ($j \in [\![1, p]\!]$).

Afin de favoriser les solutions parcimonieuses, on peut remplacer l'estimateur des moindres carrés :

$$\hat{\beta} = \underset{\beta \in \mathbb{R}^p}{argmin} \ \|Y - X\beta\|_2^2$$

par

$$\hat{\beta} = \underset{\beta \in \mathbb{R}^p}{argmin} \ \|Y - X\beta\|_2^2 + \lambda pen(\beta),$$

où $pen(\beta)$ est une fonction de pénalité à choisir. La valeur de λ (*paramètre de pénalisation*) fixe le degré de pénalité que l'on veut considérer.

II.2. Méthodes de régression

Régression Lasso

L'estimateur **Lasso** est défini par :

$$\hat{\beta}_{lasso} = \underset{\beta \in \mathbb{R}^p}{argmin} \ \|Y - X\beta\|_2^2 + \lambda \|\beta\|_1,$$

où $\|\beta\|_1 = \sum_{j=1}^{p} |\beta_j|$.

(Tibshirani, 1996)

- La pénalité lasso est utilisée pour obtenir des solutions parcimonieuses, c'est-à-dire telles que beaucoup de coefficients soient nuls (plus λ est grand, plus les solutions sont parcimonieuses).
- L'estimateur lasso est en général un estimateur de grande variance avec des problèmes de stabilité, notamment en présence de variables corrélées.
- L'estimateur Lasso n'a pas d'écriture propre, il faut le déterminer par un algorithme d'optimisation (*Lars*).

II.2. Méthodes de régression

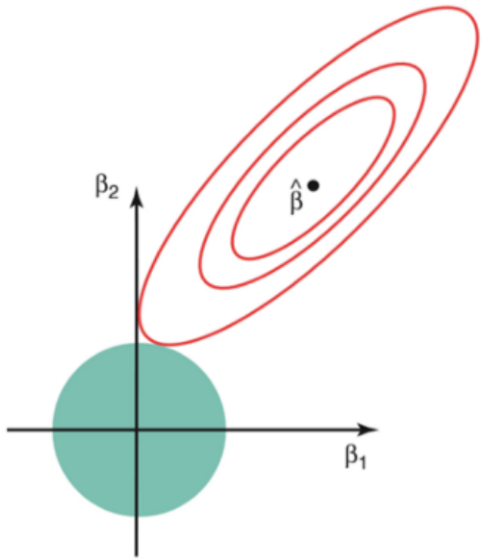
Régression Lasso et sparsité

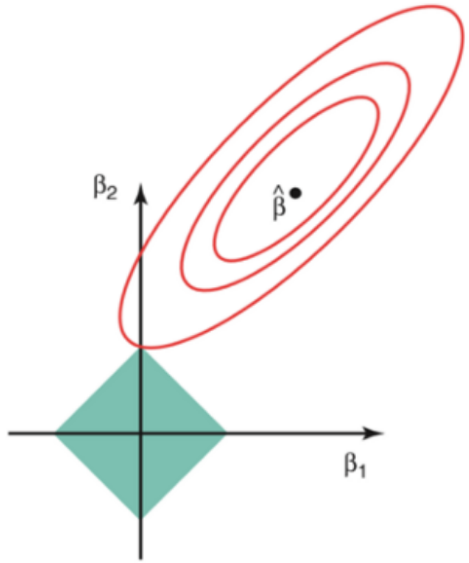
Le problème de minimisation

$$\hat{\beta}_{lasso} = \underset{\beta \in \mathbb{R}^p}{argmin} \ \|Y - X\beta\|_2^2 + \lambda \|\beta\|_1,$$

est équivalent à un problème dual de la forme :

$$\hat{\beta}_{lasso} = \underset{\beta \in \mathbb{R}^p}{argmin} \ \|Y - X\beta\|_2^2 \ \text{ sous la contrainte } \sum_i |\beta_i| \leq c(\lambda).$$





II.2. Méthodes de régression

Régression Lasso sous R

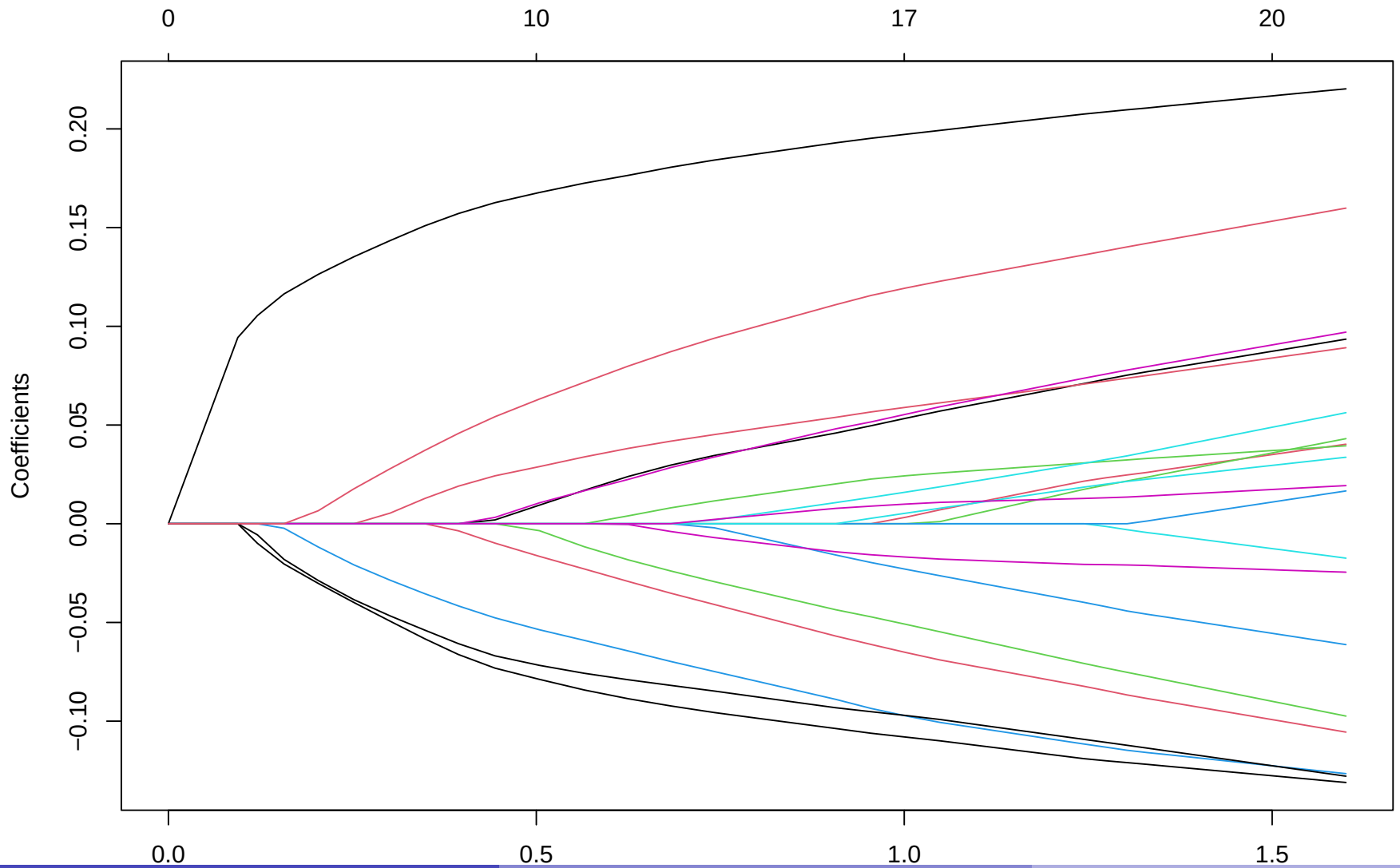
```
library(glmnet)
x = matrix(rnorm(100 * 20), 100, 20)
y = rnorm(100)

fit1 = glmnet(x, y)
print(fit1)
```

```
##
## Call:  glmnet(x = x, y = y)
##
##     Df  %Dev   Lambda
## 1    0  0.00 0.235600
## 2    1  0.83 0.214700
## 3    1  1.51 0.195600
## 4    1  2.08 0.178200
## 5    1  2.55 0.162400
## 6    1  2.95 0.148000
## 7    1  3.27 0.134800
```

II.2. Méthodes de régression

Régression Lasso sous R

```
plot(fit1)
```



II.2. Méthodes de régression

Le graphical lasso

Le **graphical lasso** propose une approche plus systématique pour estimer la matrice de précision et inférer le graphe. L'estimateur est défini de la manière suivante :

$$\hat{\Theta} = \underset{\Theta \in \mathcal{M}_{p \times p}}{argmax} \quad \log(\det \Theta) - \text{trace}(S\Theta) - \lambda \|\Theta\|_1,$$

(Friedman, Hastie et Tibshirani, 2008)

où S désigne la matrice de covariance empirique.

Le graphical lasso :

- maximise la log-vraisemblance,
- utilise une pénalité du type lasso pour obtenir des solutions parcimonieuses (plus λ est grand, plus les solutions sont parcimonieuses et plus le graphe est creux).

II. Inférence de réseaux

3. Causalité et inférence

II.3. Causalité et inférence

La d-séparation

Soit $\mathcal{G}(V, E)$ un DAG où :

- $V = \{1, ..., p\}$ est l'ensemble des nœuds,
- E est l'ensemble d'arêtes.

Définition

*Un chemin entre i et j est **bloqué** par un ensemble de nœuds $\mathcal{Z}$ (ne contenant ni i ni j) s'il existe un nœud k tel que l'une des deux conditions suivantes est satisfaite :*

- $k \in \mathcal{Z}$ *et* $i \to k \to j$ *ou* $i \leftarrow k \to j$ *ou* $i \leftarrow k \leftarrow j$,
- $i \to k \leftarrow j$ *et ni* k *ni aucun de ses descendants ne sont dans* $\mathcal{Z}$.

Si tous les chemins entre $i \in \mathcal{X}$ et $j \in \mathcal{Y}$ sont bloqués par $\mathcal{Z}$, on dit alors que $\mathcal{X}$ et $\mathcal{Y}$ sont d-séparés par $\mathcal{Z}$.

II.3. Causalité et inférence

Hypothèses importantes

On note $\mathcal{L}(X)$ la distribution jointe qui a servi à générer le DAG $\mathcal{G}$.

Définition

*La distribution $\mathcal{L}(X)$ satisfait la **propriété de Markov globale** si :*

$$(\mathcal{X}, \mathcal{Y}) \text{ } d\text{-séparés par } \mathcal{Z} \text{ dans } \mathcal{G} \Rightarrow \mathcal{X} \perp \mathcal{Y} | \mathcal{Z}.$$

Définition

*La distribution $\mathcal{L}(X)$ est **fidèle** au DAG $\mathcal{G}$ si :*

$$\mathcal{X} \perp \mathcal{Y} | \mathcal{Z} \Rightarrow (\mathcal{X}, \mathcal{Y}) \text{ } d\text{-séparés par } \mathcal{Z} \text{ dans } \mathcal{G}.$$

Sous ces deux conditions, on a un lien direct entre la notion de d-séparation (qui concerne les nœuds du graphe) et la dépendance conditionnelle (qui concerne les variables associées).

II.3. Causalité et inférence

Classe d'équivalence de Markov

Définition

Deux DAGs $\mathcal{G}_1$ et $\mathcal{G}_2$ sont ***Markov-équivalents*** *s'ils partagent les mêmes ensembles de d-séparation :*

$$(\mathcal{X}, \mathcal{Y}) \text{ d-séparés par } \mathcal{Z} \text{ dans } \mathcal{G}_1 \Leftrightarrow (\mathcal{X}, \mathcal{Y}) \text{ d-séparés par } \mathcal{Z} \text{ dans } \mathcal{G}_2.$$

II.3. Causalité et inférence

Classe d'équivalence de Markov

Définition

Deux DAGs $\mathcal{G}_1$ et $\mathcal{G}_2$ sont ***Markov-équivalents*** *s'ils partagent les mêmes ensembles de d-séparation :*

$$(\mathcal{X}, \mathcal{Y}) \text{ d-séparés par } \mathcal{Z} \text{ dans } \mathcal{G}_1 \Leftrightarrow (\mathcal{X}, \mathcal{Y}) \text{ d-séparés par } \mathcal{Z} \text{ dans } \mathcal{G}_2.$$

Deux résultats fondamentaux :

- en général, on ne peut pas identifier $\mathcal{G}$ à partir des données X mais on peut retrouver la classe d'équivalence de Markov de $\mathcal{G}$,

(Pearl, 2000)

- les DAGs qui sont dans la même classe d'équivalence de Markov partagent le même squelette et les mêmes *v*-structures.

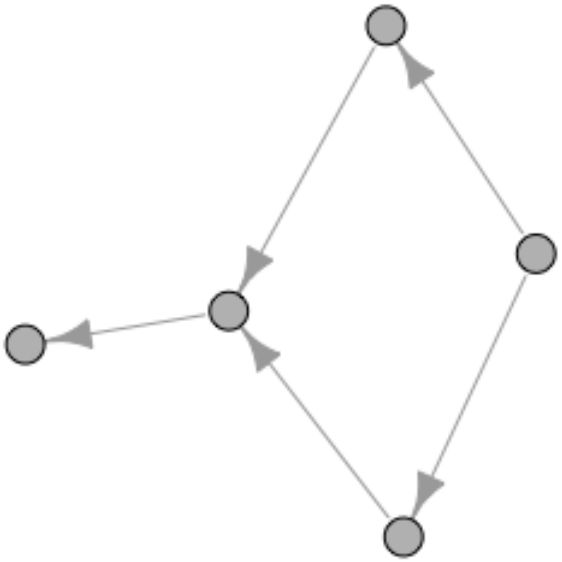
(Verma et Pearl, 2000)

II.3. Causalité et inférence

Classe d'équivalence de Markov

La classe d'équivalence de Markov peut être représentée sous la forme d'un CPDAG (*Completed Partially Directed Acyclic Graph*) :

- même squelette, graphe obtenu en enlevant toutes les orientations d'arêtes,
- mêmes v-structures, triplets de nœuds (i, j, k) tels que $i \rightarrow k \leftarrow j$ où i et j ne sont pas adjacents.



Classe d'équivalence de Markov?

II.3. Causalité et inférence

L'algorithme PC

L'**algorithme PC** (Kalisch et Bühlmann, 2007) permet de reconstruire la classe d'équivalence du DAG $\mathcal{G}$.

Initialisation : graphe complet pour lequel toutes les nœuds sont reliés par des arêtes non dirigés.

Puis, 3 étapes successives :

1. construction du squelette du graphe,
2. détermination des *v*-structures,
3. orientation du maximum d'arêtes restantes.

II.3. Causalité et inférence

L'algorithme PC

L'**algorithme PC** (Kalisch et Bühlmann, 2007) permet de reconstruire la classe d'équivalence du DAG $\mathcal{G}$.

Initialisation : graphe complet pour lequel toutes les nœuds sont reliés par des arêtes non dirigés.

Puis, 3 étapes successives :

1. construction du squelette du graphe,

Pour $k \in \{0, ..., p-2\}$, on considère à tour de rôle chaque paire (i, j) de sommets adjacents du graphe et tous les sous-ensembles de nœuds Z adjacents à i ou j de taille k. On enlève l'arête $i - j$ si $X^i \perp X^j | Z$.

2. détermination des *v*-structures,

3. orientation du maximum d'arêtes restantes.

II.3. Causalité et inférence

L'algorithme PC

L'**algorithme PC** (Kalisch et Bühlmann, 2007) permet de reconstruire la classe d'équivalence du DAG $\mathcal{G}$.

Initialisation : graphe complet pour lequel toutes les nœuds sont reliés par des arêtes non dirigés.

Puis, 3 étapes successives :

1. construction du squelette du graphe,
2. détermination des *v*-structures,

Les v-structures $i \rightarrow k \leftarrow j$ vérifient les deux conditions suivantes : pas d'arêtes entre i et j et k n'a pas servi pour montrer l'indépendance conditionnelle entre X^i et X^j.

3. orientation du maximum d'arêtes restantes.

II.3. Causalité et inférence

L'algorithme PC

L'**algorithme PC** (Kalisch et Bühlmann, 2007) permet de reconstruire la classe d'équivalence du DAG $\mathcal{G}$.

Initialisation : graphe complet pour lequel toutes les nœuds sont reliés par des arêtes non dirigés.

Puis, 3 étapes successives :

1. construction du squelette du graphe,
2. détermination des v-structures,
3. orientation du maximum d'arêtes restantes.

Par consistance avec les arêtes déjà orientés puisqu'on ne peut pas créer de nouvelles v-structures.

II. Inférence de réseaux

4. Métriques d'évaluation

II.4. Métriques d'évaluation

Définitions

On définit les quantités suivantes :

- **TP** (vrais positifs), **FP** (faux positifs), **TN** (vrais négatifs), **FN** (faux négatifs),

correspondant à la matrice de confusion :

		Estimation	
		Arête	Non-arête
Vérité	Arête	TP	FN
	Non-arête	FP	TN

- la **précision** $= \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}}$,
- le **recall** $= \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}$,
- le **taux de faux positifs** $= \frac{\mathrm{FP}}{\mathrm{FP}+\mathrm{TP}}$.

II.4. Métriques d'évaluation

Comment évaluer les méthodes?

Pour évaluer les méthodes, deux cas sont à distinguer :

1. aucun paramètre ne permet de calibrer la parcimonie du réseau reconstruit

II.4. Métriques d'évaluation

Comment évaluer les méthodes?

Pour évaluer les méthodes, deux cas sont à distinguer :

1. aucun paramètre ne permet de calibrer la parcimonie du réseau reconstruit

$\rightarrow$ on calcule des indicateurs de taux de faux positifs, précision, recall que l'on compare directement

II.4. Métriques d'évaluation

Comment évaluer les méthodes?

Pour évaluer les méthodes, deux cas sont à distinguer :

1. aucun paramètre ne permet de calibrer la parcimonie du réseau reconstruit

→ on calcule des indicateurs de taux de faux positifs, précision, recall que l'on compare directement

2. un paramètre permet de calibrer la parcimonie du réseau reconstruit

II.4. Métriques d'évaluation

Comment évaluer les méthodes?

Pour évaluer les méthodes, deux cas sont à distinguer :

1. aucun paramètre ne permet de calibrer la parcimonie du réseau reconstruit

→ on calcule des indicateurs de taux de faux positifs, précision, recall que l'on compare directement

2. un paramètre permet de calibrer la parcimonie du réseau reconstruit

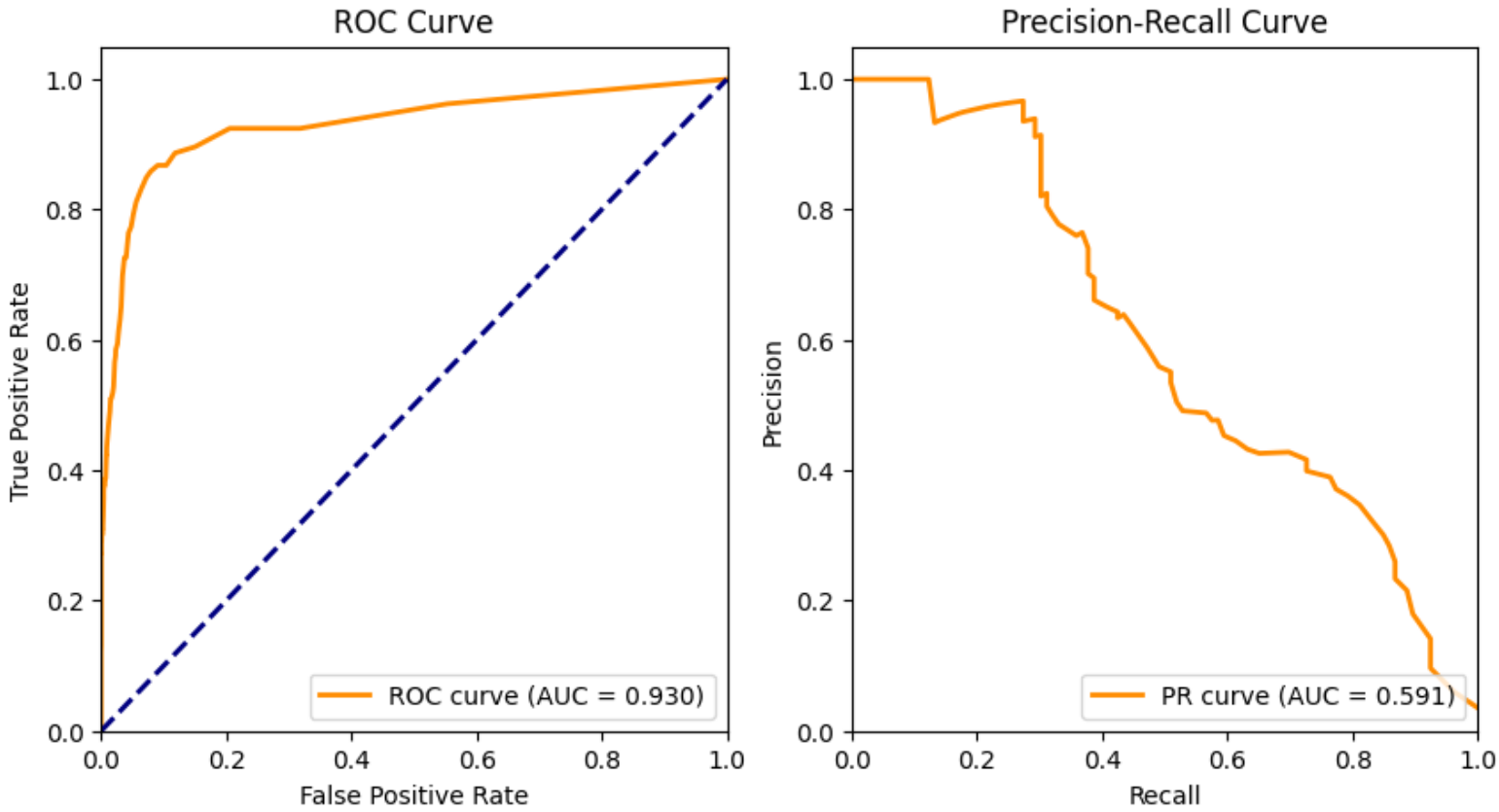
→ on calcule l'évolution du taux de faux positifs, la précision, le recall au cours de la reconstruction du réseau

⚠*En pratique, la vérité (le vrai réseau ayant servi à générer les données) n'étant pas connue, on ne peut pas comparer le réseau reconstruit au réseau réel.*

II.4. Métriques d'évaluation

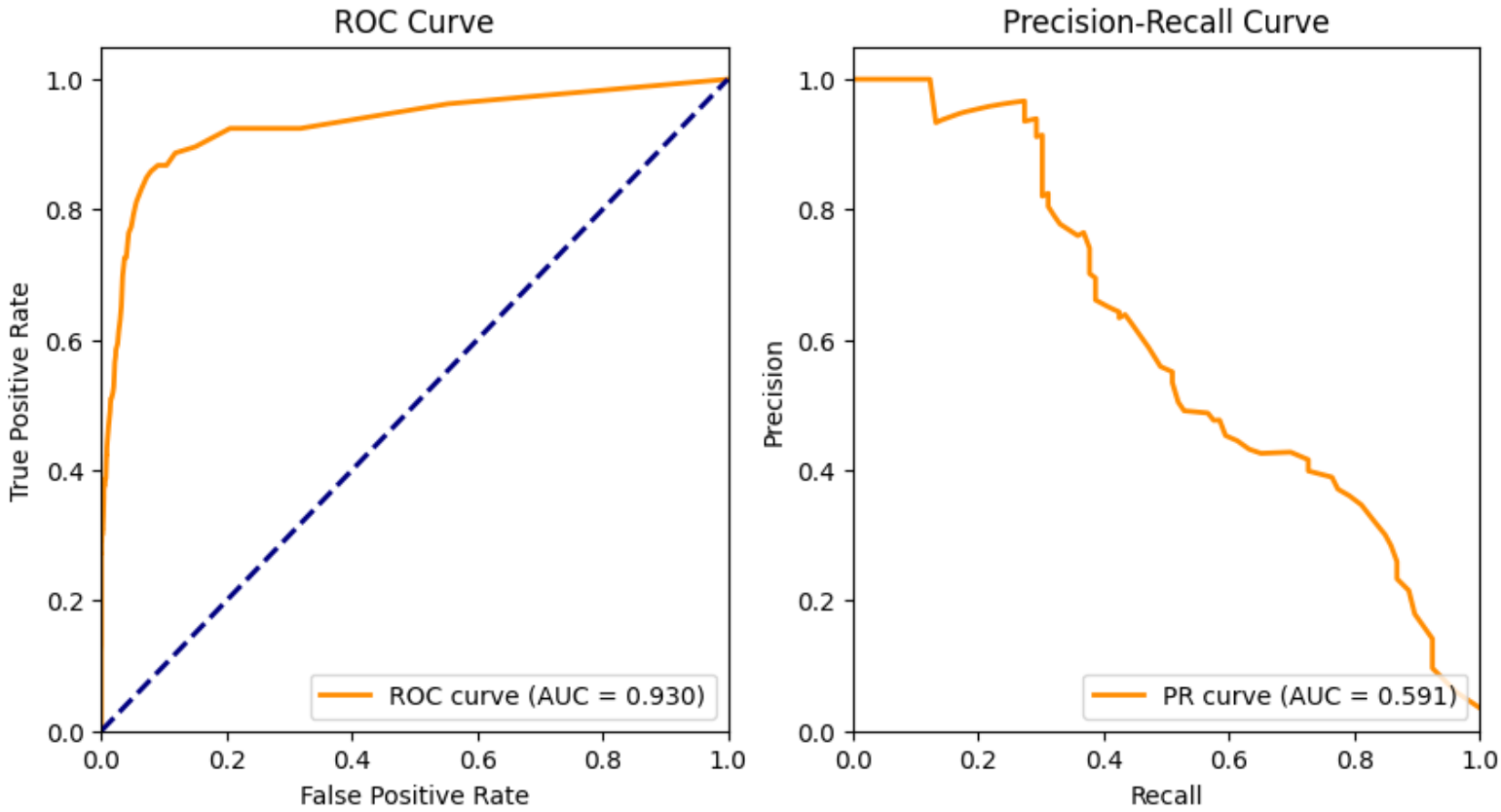
Courbes précision-recall et ROC

La courbe **précision-recall**, *resp.* **ROC**, est obtenue en traçant l'évolution de la précision en fonction du recall, *resp.* taux de vrais positifs en fonction du recall, pour différents réseaux reconstruits.



II.4. Métriques d'évaluation

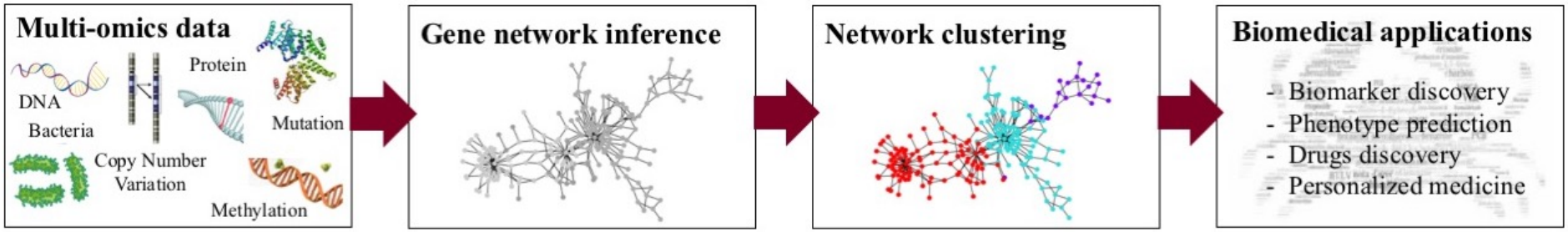
Aires sous les courbes



Les **aires sous les courbes** (AUC) donnent alors une idée de performance des méthodes.

III. Clustering de réseaux

Plan du cours



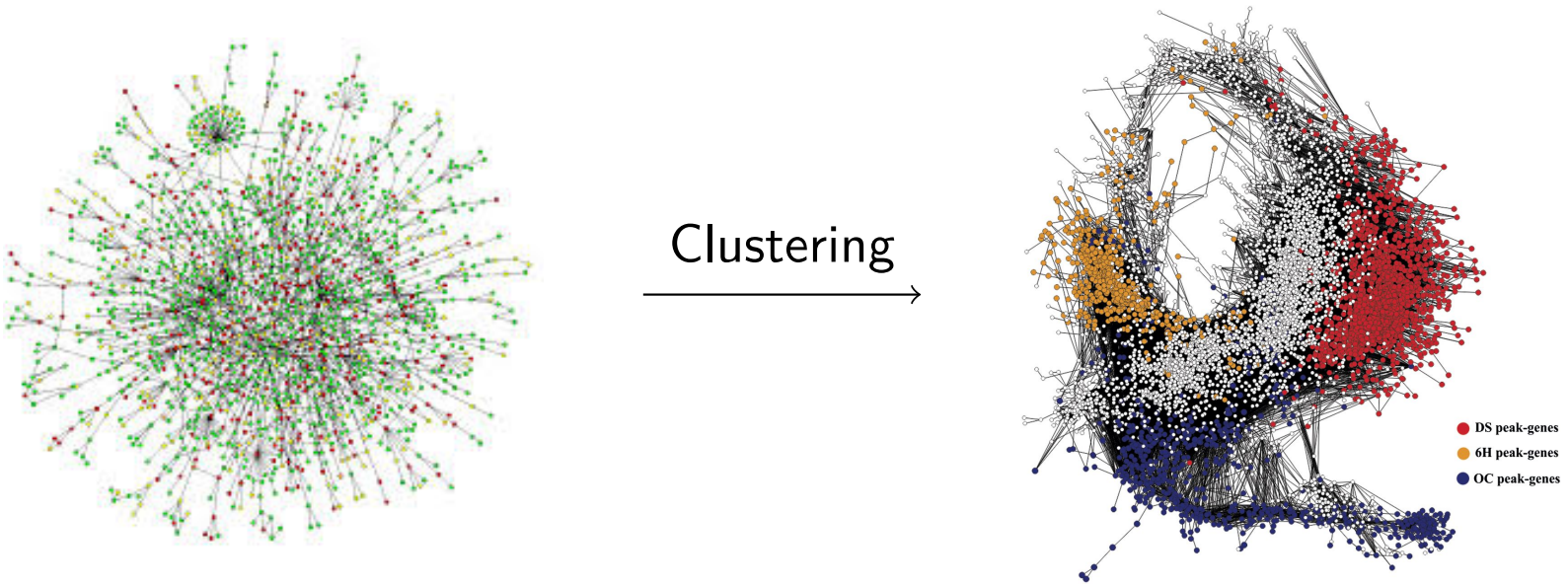
1. Les réseaux de régulation de gènes
2. Inférence de réseaux
3. Clustering de réseaux
   - Techniques de clustering
   - Clustering hiérarchique pour les graphes
   - Méthodes spectrales
   - Métriques d'évaluation
4. Applications

Introduction

Le clustering de graphes, c'est quoi?

On suppose qu'on a réussi à inférer un réseau de p gènes, modélisant les interactions qui existent entre ces gènes, en utilisant leurs données d'expression pour n échantillons, contenues dans la matrice $X \in \mathcal{M}_{n \times p}$.

L'objectif consiste alors à créer des groupes (*clusters*) de gènes partageant des caractéristiques communes afin d'améliorer notre compréhension du système biologique étudié.



III. Clustering de réseaux

1. Techniques de clustering

III. 1. Techniques de clustering

Revenons à la base

On considère :

- p variables explicatives $(X^1, ..., X^p)$,
- un n-échantillon $(x_i^1, ..., x_i^p)_{1 \leq i \leq n}$ de $(X^1, ..., X^p)$.

Une **partition** $\mathcal{C} = \{C_1, ..., C_k\}$ des données vérifie les propriétés suivantes :

- $C_i \cap C_j = \emptyset$ pour tout $i \neq j \in \{1, ..., k\}$,
- $\cup_{i \in \{1,...,k\}} C_i = n$.

L'objectif consiste à créer une partition $\mathcal{C} = \{C_1, ..., C_k\}$ des n individus (approche non-supervisée) telle que l'on ait :

- une petite variabilité intra-classe (une petite distance entre les individus d'un même groupe),
- une grande variabilité inter-classe (une grande distance entre les individus de groupes différents).

III.1. Techniques de clustering

Problématiques

Les **problématiques** associées au clustering sont de différents types :

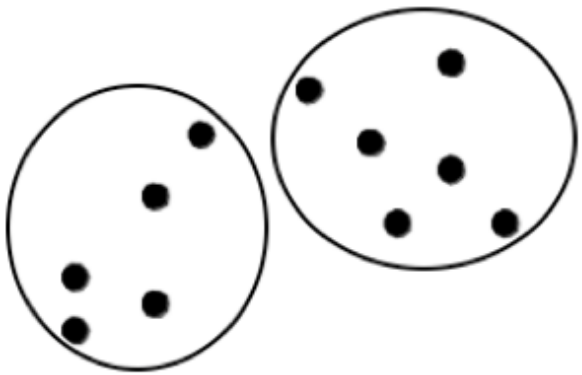
- nature des observations : les données sont-elles binaires, textuelles, numériques. . . ?
- notion de similarité : comment définir une similarité ou dissimilarité entre observations?
- interprétation d'un cluster : comment résumer un cluster?
- évaluation des performances d'un algorithme de clustering : comment évaluer les méthodes?
- définition du nombre de clusters : combien de clusters faut-il construire?

III.1. Techniques de clustering

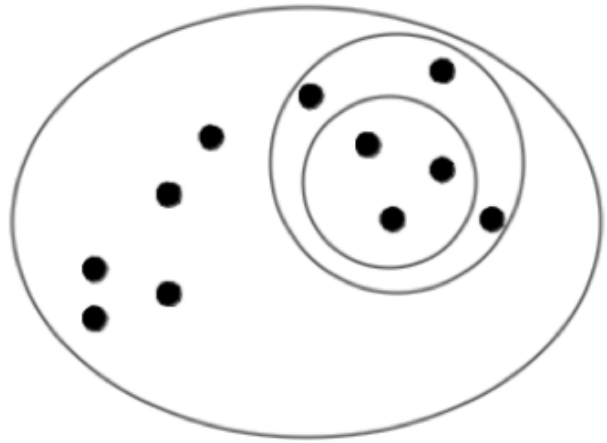
Différentes approches

Il existe 3 grandes familles de clustering :

- les approches **par partionnement**[1], pour lesquelles les classes construites sont toujours disjointes,
- les approches **hiérarchiques**[2], pour lesquelles les classes sont disjointes ou incluses les unes dans les autres,
- les approches **spectrales**.



1



2

III.1. Techniques de clustering

Approches par partitionnement

On considère :

- p variables explicatives $(X^1, ..., X^p)$,
- un n-échantillon $(x_i^1, ..., x_i^p)_{1 \leq i \leq n}$ de $(X^1, ..., X^p)$.

Objectif : Construire une partition des données en $k < n$ clusters $(C_j)_{1 \leq j \leq k}$.

III.1. Techniques de clustering

Approches par partitionnement

On considère :

- p variables explicatives $(X^1, ..., X^p)$,
- un n-échantillon $(x_i^1, ..., x_i^p)_{1 \leq i \leq n}$ de $(X^1, ..., X^p)$.

Objectif : Construire une partition des données en $k < n$ clusters $(C_j)_{1 \leq j \leq k}$.

Approche naïve : construire toutes les partitions possibles et en retenir la meilleure.

⚠Le nombre de partitions augmente de manière exponentielle : il s'agit d'un problème NP difficile.

III.1. Techniques de clustering

Approches par partitionnement : k-means

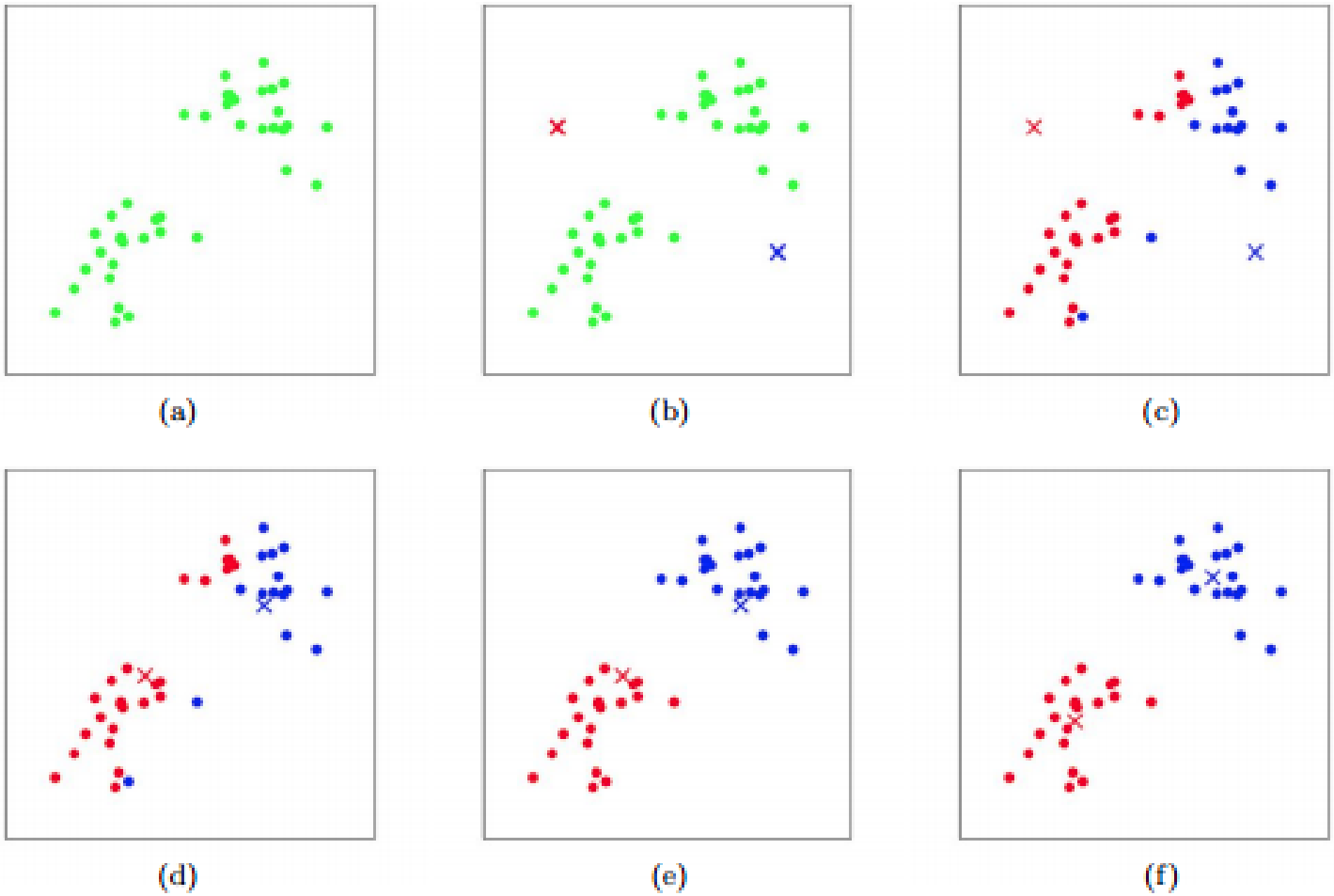
Le k-means (Forgy, 1965; MacQueen, 1967) est l'une des méthodes les plus anciennes et traditionnelles pour effectuer de le classification non-supervisée.

Principe :

1. choisir k éléments initiaux (centres de gravité) $(\mu_j)_{1 \leq j \leq k}$ pour les k clusters $(C_1, ..., C_k)$ à construire,
2. affecter chaque observation x_i à la classe C_j dont le centre μ_j est le plus proche,
3. recalculer le centre de gravité de chaque cluster $(C_1, ..., C_k)$,
4. itérer jusqu'à stabilité des clusters.

III.1. Techniques de clustering

Approches par partitionnement : k-means



III.1. Techniques de clustering

Approches par partitionnement : k-means

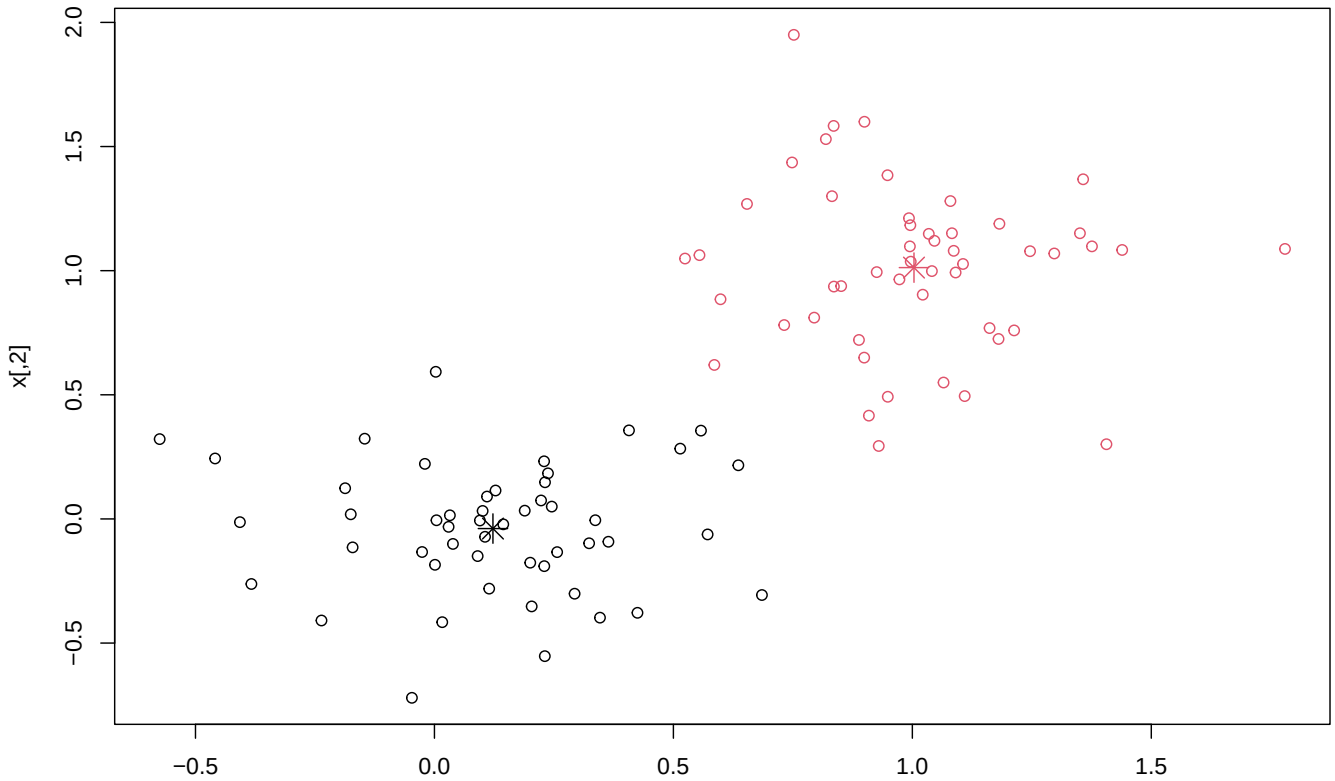
Remarques :

- les centres initiaux sont souvent choisis aléatoirement, des initialisations différentes peuvent donc mener à des clusters différents,
  - faire plusieurs essais,
  - utiliser du clustering hiérarchique pour déterminer les centres initiaux,
- l'utilisation du k-means nécessite de connaître le nombre de clusters,
  - fixer k à priori,
  - chercher la meilleure partition pour différentes valeurs de k,
- le k-means est sensible à la présence d'outliers et est en difficulté lorsque les clusters sont de différentes tailles ou densités.

III.1.Techniques de clustering

Approches par partitionnement : k-means sous R

```
x <- rbind(matrix(rnorm(100, sd = 0.3), ncol = 2),
           matrix(rnorm(100, mean = 1, sd = 0.3), ncol = 2))
clusters <- kmeans(x, 2)
plot(x, col = clusters$cluster)
points(clusters$centers, col = 1:2, pch = 8, cex = 2)
```



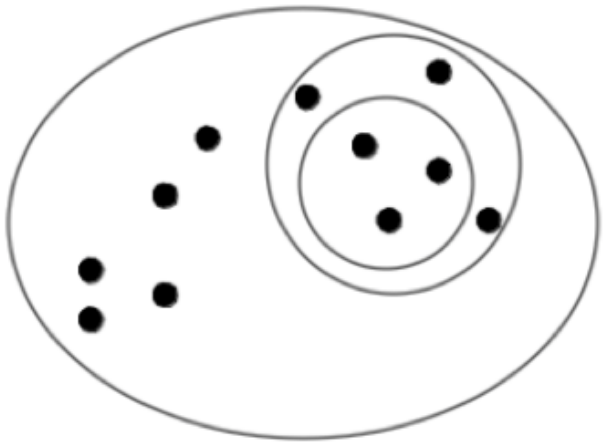
III.1. Techniques de clustering

Approches hiérarchiques

On considère :

- p variables explicatives $(X^1, ..., X^p)$,
- un n-échantillon $(x_i^1, ..., x_i^p)_{1 \leq i \leq n}$ de $(X^1, ..., X^p)$.

Objectif : Construire une partition **hiérarchique** des données en $k < n$ clusters $(C_j)_{1 \leq j \leq k}$.



III.1. Techniques de clustering

Approches hiérarchiques

Il existe **deux** types d'approches :

- clustering hiérarchique ascendant ou agglomératif (**CAH**) :
  - commencer en considérant que chaque individu est un cluster à lui seul,
  - à chaque étape, regrouper les clusters les plus proches jusqu'à obtenir 1 ou k clusters,

- clustering hiérarchique descendant ou divisif :
  - commencer en considérant un seul cluster contenant l'ensemble des individus,
  - à chaque étape, diviser un cluster jusqu'à obtenir des clusters ne contenant qu'un point ou k clusters.

III.1. Techniques de clustering

Approches hiérarchiques : CAH

Principe :

1. placer les n individus dans leur propre cluster (n clusters au total),
2. calculer la similarité entre chaque couple de clusters,
3. chercher les deux clusters les plus proches basé sur cette mesure de similarité,
4. fusionner ces deux clusters,
5. recalculer la similarité entre chaque couple de clusters,
6. itérer jusqu'à l'obtention de 1 ou k clusters.

III.1. Techniques de clustering

Approches hiérarchiques : CAH

Principe :

1. placer les n individus dans leur propre cluster (n clusters au total),
2. calculer la similarité entre chaque couple de clusters,
3. chercher les deux clusters les plus proches basé sur cette mesure de similarité,
4. fusionner ces deux clusters,
5. recalculer la similarité entre chaque couple de clusters,
6. itérer jusqu'à l'obtention de 1 ou k clusters.

Point-clé : définir le critère de choix de clusters à fusionner (critère d'agrégation)

III.1. Techniques de clustering

Approches hiérarchiques : CAH

Quelques critères d'agrégation :

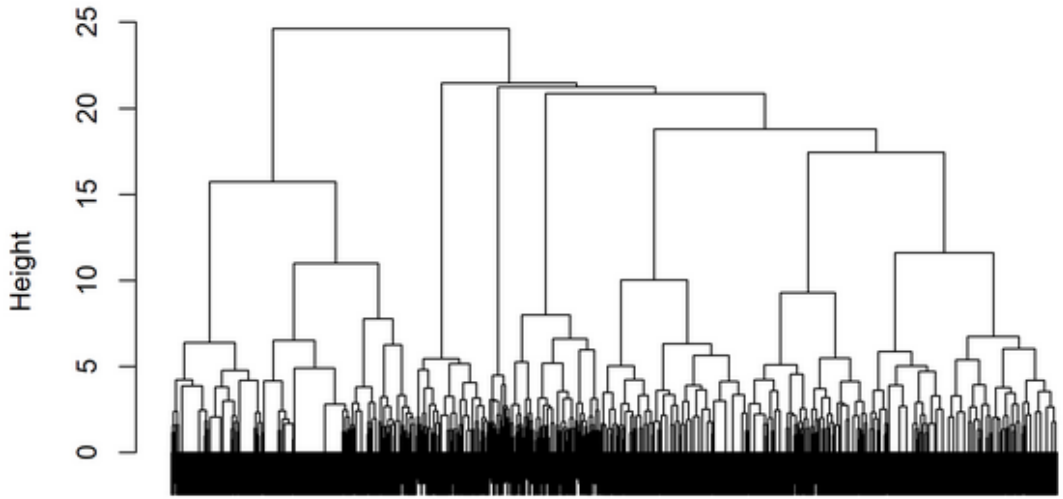
- saut minimal (single linkage) : basée sur la distance entre les deux points les plus proches de chaque cluster,
  - tendance à construire des clusters très généraux
- saut maximal (complete linkage) : basée sur la distance entre les deux points les plus éloignés de chaque cluster,
  - tendance à construire des clusters très spécifiques
- saut moyen : basée sur la distance moyenne entre les points des clusters,
  - tendance à construire des clusters de variance proche
- méthode de Ward
  - chaque cluster est représenté par son centre de gravité,
  - agglomération des clusters basée sur l'inertie intra-classe.

III.1. Techniques de clustering

Approches hiérarchiques : représentations graphiques

Les résultats des approches hiérarchiques sont représentés sous la forme d'un **dendogramme**, qui fournit une visualisation de la hiérarchie des partitions obtenues sous la forme d'un arbre dont :

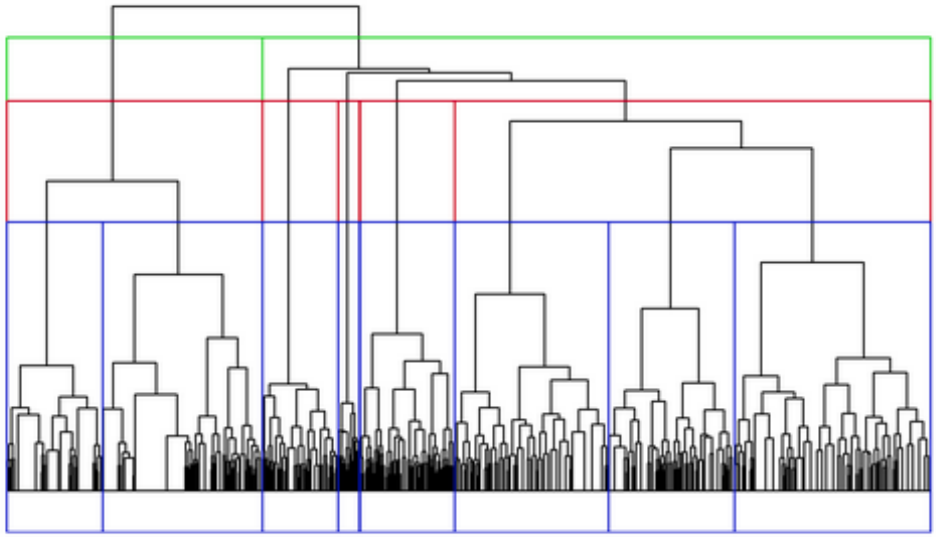
- les nœuds sont les différents clusters construits,
- les feuilles sont les individus,
- la racine est la partition finale en un unique cluster.



III.1. Techniques de clustering

Approches hiérarchiques : représentations graphiques

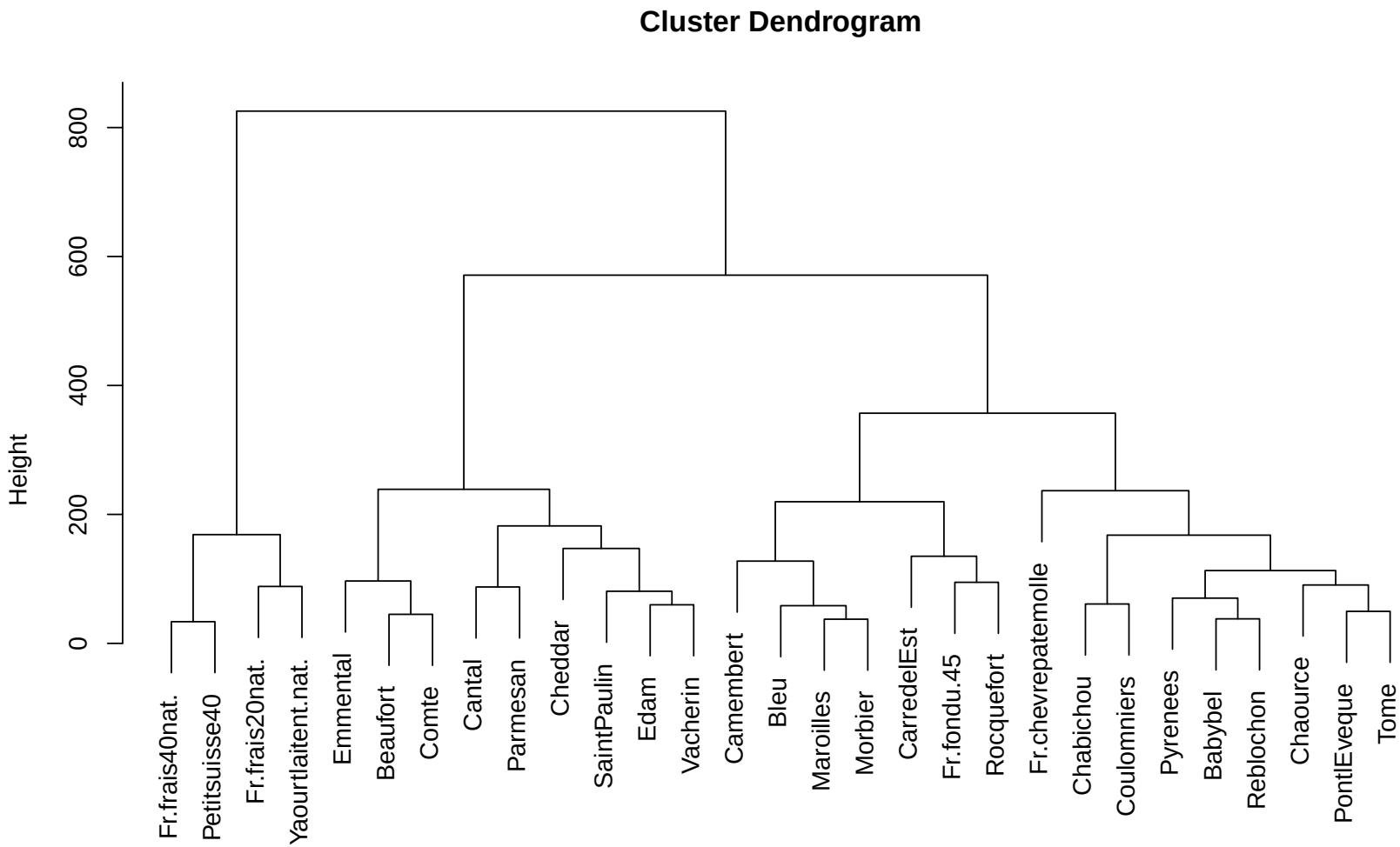
Choix du nombre de clusters : l'axe des ordonnées du dendrogramme indique la valeur du critère d'agrégation (avant fusion), on cherche donc des coupures nettes dans le dendrogramme.



III.1. Techniques de clustering

Approches hiérarchiques : CAH sous R

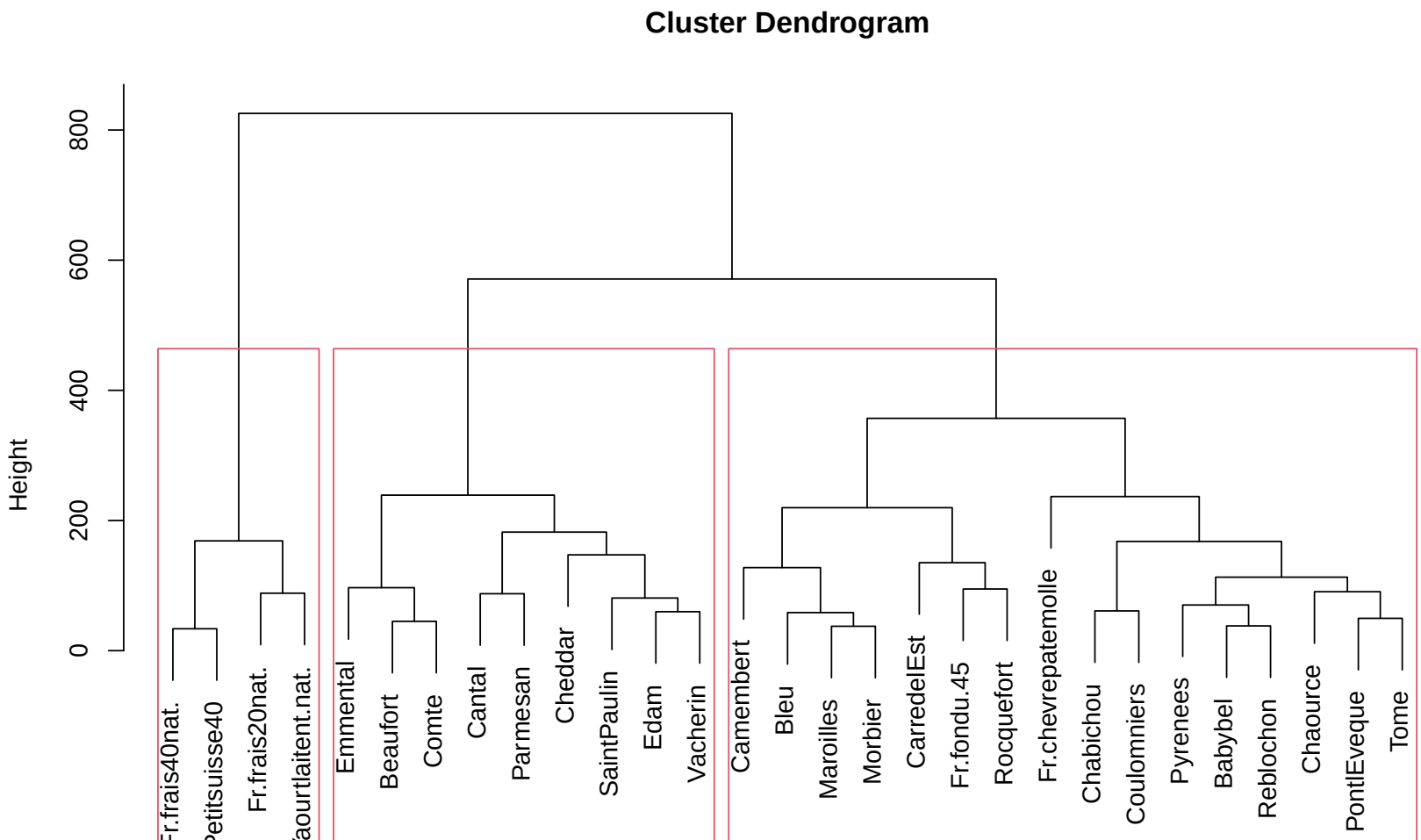
```
hc <- hclust(dist(fromages),method="ward.D2")
plot(hc)
```



III.1. Techniques de clustering

Approches hiérarchiques : CAH sous R

```
# dendrogramme avec matérialisation des groupes
plot(hc)
rect.hclust(hc,k=3)
```



III.1. Techniques de clustering

Classification mixte

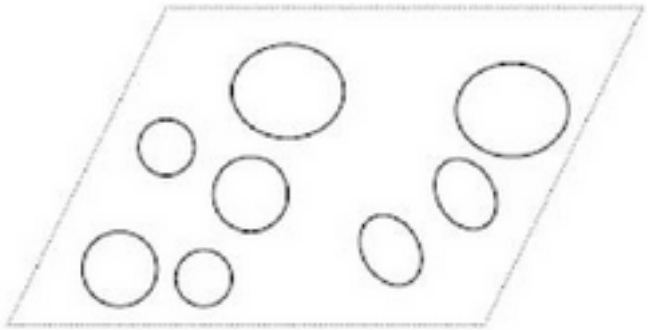
La **classification mixte** (Lebart, 1984) - combinaison du k-means et de la CAH - a été mise en place dans le but de pallier les difficultés de la CAH lors du passage à la grande dimension (calcul de la similarité de tous les individus 2 à 2).

Principe :

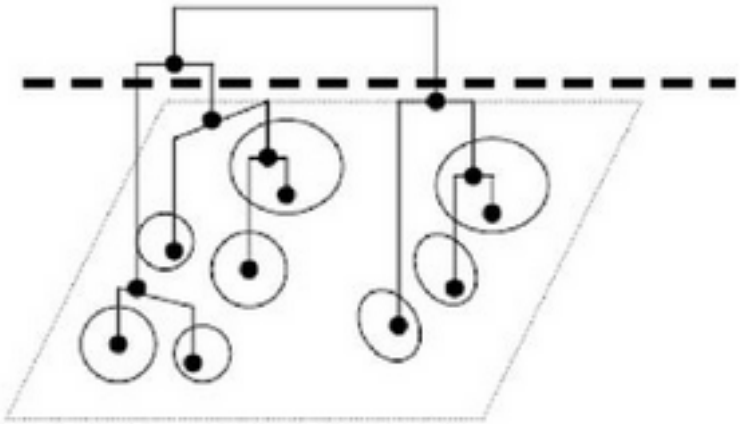
- Partitionnement préliminaire des individus en $K > k$ clusters par k-means
  - permet de diminuer la dimension du problème
- Classification hiérarchique CAH sur les K clusters initiaux
  - on utilise les barycentres des clusters obtenus lors de l'étape 1
  - le nombre de clusters final est déterminé en coupant l'arbre hiérarchique
- Consolidation de la partition par réaffectation des individus dans les clusters
  - permet d'augmenter l'inertie inter-classe et l'homégénéité des clusters

III.1. Techniques de clustering

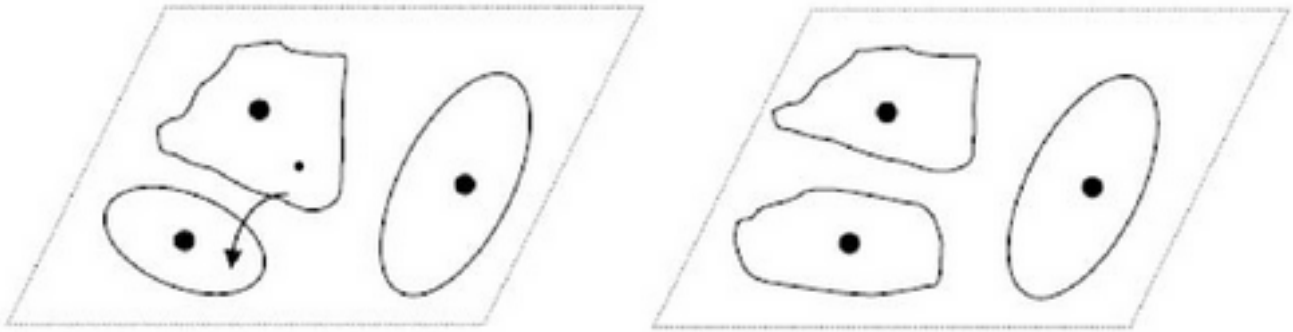
Classification mixte



Etape 1: Partitionnement préliminaire



Etape 2: CAH



Etape 3: Partition finale et consolidation de la partition par réallocations

III. Clustering de réseaux

2. Clustering hiérarchique pour les graphes

III.2. Clustering hiérarchique pour les graphes

Similarité/dissimiralité

On considère cette fois un graphe $\mathcal{G} = (V, E)$ constitué de p sommets reliés entre eux par les arêtes $(i, j) \in E$. On note A la matrice d'adjacence associée :

$$A = (A_{i,j}) = \begin{cases} 1 & \text{if } (i,j) \in E \\ 0 & \text{otherwise.} \end{cases}$$

Pour appliquer la méthode de clustering hiérarchique **CAH**, il nous faut définir une notion de **similarité/dissimilarité** qui pourra être basée sur :

- la distance euclidienne : $d_{i,j} = \sqrt{\sum_k (A_{i,k} - A_{j,k})^2}$,
- le plus court chemin : *la longueur du plus court chemin entre deux nœuds*.

III.2. Clustering hiérarchique pour les graphes

Similarité/dissimiralité

On considère cette fois un graphe $\mathcal{G} = (V, E)$ constitué de p sommets reliés entre eux par les arêtes $(i, j) \in E$. On note A la matrice d'adjacence associée :

$$A = (A_{i,j}) = \begin{cases} 1 & \text{if } (i,j) \in E \\ 0 & \text{otherwise.} \end{cases}$$

Pour appliquer la méthode de clustering hiérarchique **CAH**, il nous faut définir une notion de **similarité/dissimilarité** qui pourra être basée sur :

- la distance euclidienne : $d_{i,j} = \sqrt{\sum_k (A_{i,k} - A_{j,k})^2}$,
- le plus court chemin : *la longueur du plus court chemin entre deux nœuds*.

Mais on préfère les dissimilarités qui privilégient les groupes de nœuds densément connectés entre eux et peu connectés avec le reste!

III.2. Clustering hiérarchique pour les graphes

Pour être plus pertinent

D'autres méthodes hiérarchiques sont à privilégier dans ce contexte de clustering de graphes. Elles font appel aux notions de :

- **modularité** : soit $C = (C_1, ..., C_k)$ une partition candidate,

$$\text{modularité}(C) = \sum_{i=1}^{k} (e_{ii} - a_i^2),$$

où e_{ij} est la fraction des arêtes reliant les arêtes de C_i à C_j et $a_i = \sum_j e_{ij}$.

(Clauset, Newman et Moore, 2004)

III.2. Clustering hiérarchique pour les graphes

Pour être plus pertinent

D'autres méthodes hiérarchiques sont à privilégier dans ce contexte de clustering de graphes. Elles font appel aux notions de :

- **modularité** : soit $C = (C_1, ..., C_k)$ une partition candidate,

$$\text{modularité}(C) = \sum_{i=1}^{k} (e_{ii} - a_i^2),$$

où e_{ij} est la fraction des arêtes reliant les arêtes de C_i à C_j et $a_i = \sum_j e_{ij}$.

(Clauset, Newman et Moore, 2004)

- **centralité intermédiaire** : elle est définie pour chaque arête $(i, j) \in E$ du graphe

$$\text{centralité intermédiaire}_{i,j} = \sum_{(k,\ell) \neq (i,j)} \frac{\sigma_{k,\ell}(i,j)}{\sigma_{k,\ell}},$$

où $\sigma_{k,\ell}$ est le nombre total de plus courts chemins reliant k à ℓ et $\sigma_{k,\ell}(i,j)$ le nombre de ces chemins passant par (i,j).

(Newman et Girvan, 2004)

III. Clustering de réseaux

3. Méthodes spectrales

III.3. Méthodes spectrales

Définitions

On considère un graphe $\mathcal{G} = (V, E)$ constitué de p sommets reliés entre eux par les arêtes $(i, j) \in E$. On note A la matrice d'adjacence associée :

$$A = (A_{i,j}) = \begin{cases} 1 & \text{if } (i,j) \in E \\ 0 & \text{otherwise.} \end{cases}.$$

On définit le degré d_i d'un sommet $i \in E$ comme le nombre d'arêtes incidentes à i :

$$d_i = \sum_{j=1}^{p} A_{i,j} \text{ et } D \text{ la matrice diagonale des degrés associée.}$$

III.3. Méthodes spectrales

Définitions

On considère un graphe $\mathcal{G} = (V, E)$ constitué de p sommets reliés entre eux par les arêtes $(i, j) \in E$. On note A la matrice d'adjacence associée :

$$A = (A_{i,j}) = \begin{cases} 1 & \text{if } (i,j) \in E \\ 0 & \text{otherwise.} \end{cases}.$$

On définit le degré d_i d'un sommet $i \in E$ comme le nombre d'arêtes incidentes à i :

$$d_i = \sum_{j=1}^{p} A_{i,j} \text{ et } D \text{ la matrice diagonale des degrés associée.}$$

Définition

La matrice Laplacienne est définie par :

$$L = D - A.$$

III.3. Méthodes spectrales

Matrice Laplacienne

La **matrice Laplacienne** L a les propriétés suivantes :

- L est symétrique,
- la plus petite valeur propre de L est 0,
- L a p valeurs propres réelles positives $\lambda_1, ..., \lambda_p$.

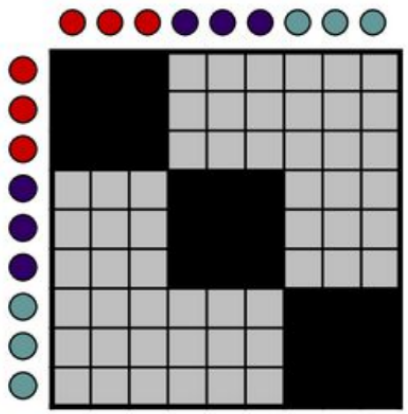
Proposition

Si on suppose que le graphe $\mathcal{G} = (V, E)$ est constitué de k composantes connexes $C_1, ...C_k$ alors :

- *la valeur propre 0 de L est de multiplicité k,*
- *les k vecteurs propres associés correspondent aux vecteurs indicateurs $(1_{C_i})_{1 \leq i \leq k}$ de ces k clusters.*

III.3. Méthodes spectrales

Comment ça marche?



Matrice d'ajacence

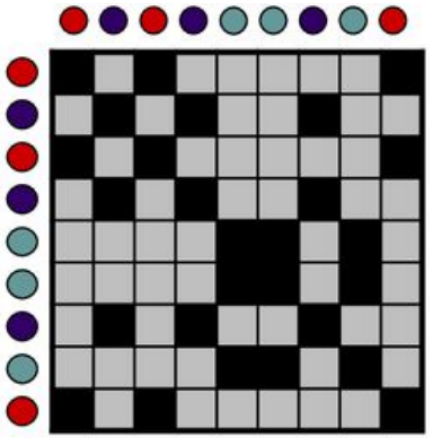
$\xrightarrow{\text{Calcul}}$ L $\xrightarrow[\text{Spectrale}]{\text{Décomposition}}$ $\lambda_1 = \cdots = \lambda_k = 0$

Vecteurs propres associés $\downarrow$

$$\begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \cdots, \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

III.3. Méthodes spectrales

Comment ça marche vraiment?



Matrice d'ajacence

$\xrightarrow{\text{Calcul}}$ L $\xrightarrow[\text{Spectrale}]{\text{Décomposition}}$ $\lambda_1, \ldots, \lambda_k \simeq 0$

Vecteurs propres associés

$$\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}, \ldots, \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \xleftarrow{k\text{-means}} \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}, \ldots, \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

III.3. Méthodes spectrales

Algorithme de spectral clustering

Algorithme de Spectral clustering *(Luxburg, 2007)*

Input: $\mathcal{G}$ un graphe, A sa matrice d'adjacence associée,
$\hat{k}$ nombre de clusters à construire.

1: Calculer la matrice Laplacienne $L = D - A$.

2: Faire la décomposition spectrale de L et conserver les $\hat{k}$ premiers vecteurs propres $U := (u_1, ..., u_{\hat{k}})$.

3: Utiliser le k-means sur U pour créer les $\hat{k}$ clusters $C_1, ..., C_{\hat{k}}$.

Output: Clusters $C_1, ..., C_{\hat{k}}$

III. Clustering de réseaux

4. Métriques d'évaluation

Métriques d'évaluation

Deux cas possibles

Il y a deux façons d'évaluer les différentes techniques de clustering :

- mesures externes : nécessitent de connaître la vraie composition des clusters (inaccessible en pratique avec des jeux de données réels),
  - précision/recall,
  - pureté,
  - information mutuelle,...
- mesures internes : ne nécessitent aucune donnée extérieure,
  - coefficient de silhouette.

III.4. Métriques d'évaluation

Précision/recall

On définit les quantités suivantes :

- **TP** (vrais positifs), **FP** (faux positifs), **TN** (vrais négatifs), **FN** (faux négatifs),

correspondant à la matrice de confusion :

		Estimation $\hat{C}$	
		Cluster A	Cluster B
Vérité C	Cluster A	TP	FN
	Cluster B	FP	TN

On peut alors évaluer les méthodes de clustering en calculant :

- la **précision** $= \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}}$,
- le **recall** $= \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}$.

III.4. Métriques d'évaluation

Pureté

La **pureté** mesure la précision des méthodes de clustering :

$$\text{pureté} = \frac{1}{p} \sum_{i=1}^{k} \max_{j} |C_i \cap \hat{C}_j|,$$

- Indice compris entre 0 et 1.
- Plus la pureté est proche de 1, meilleure est la méthode de clustering.
- L'indice de pureté dépend du nombre de clusters et atteint même 1 si tous les sommets sont isolés dans un cluster.

→ *Il vaut mieux privilégier les scores basés sur l'information mutuelle!*

III.4. Métriques d'évaluation

Information mutuelle

L'**information mutuelle** (MI) est définie de la manière suivante :

$$\text{MI}\ (\hat{C}, C) = H(\hat{C}) - H(\hat{C}|C),\ \text{où :}$$

- $H(\hat{C})$ est l'entropie des clusters estimés:

$$H(\hat{C}) = -\sum_i \mathbb{P}(\hat{C}_i) \log \mathbb{P}(\hat{C}_i) = \sum_i \frac{|\hat{C}_i|}{p} \log \frac{|\hat{C}_i|}{p}.$$

- $H(\hat{C}|C)$ est l'entropie conditionnelle des clusters estimés sachant les vrais clusters:

$$H(\hat{C}|C) = \sum_j H(\hat{C}|C_j) = -\sum_j \mathbb{P}(C_j) \sum_i \mathbb{P}(\hat{C}_i|C_j) \log \mathbb{P}(\hat{C}_i|C_j).$$

L'information mutuelle indique la réduction de l'entropie des clusters estimés que nous obtenons si nous connaissons les vrais clusters.

III.4. Métriques d'évaluation

Information mutuelle normalisée

L'**information mutuelle normalisée** (NMI) est la version normalisée de l'information mutuelle :

$$\mathrm{NMI}(\hat{C}, C) = \frac{2 \times \mathrm{MI}(\hat{C}, C)}{H(\hat{C}) + H(C)}.$$

- Bonne mesure de qualité de clustering.
- Plus grande est la NMI, meilleure est la méthode de clustering.
- Comprise entre 0 et 1, elle permet de comparer des méthodes de clustering qui ne construisent pas le même nombre de clusters.

→ *Une version ajustée par chance* (**AMI**) *permet d'assurer qu'une classification aléatoire soit associée à un score de 0.*

III.4. Métriques d'évaluation

Coefficient de silhouette

Le **coefficient de silhouette** est une mesure de qualité de clustering ne nécessitant pas de connaître les vrais clusters. Pour chaque point, on calcule son coefficient de silhouette :

$$\text{silhouette}_i = \frac{b_i - a_i}{max(a_i, b_i)}, \text{ où :}$$

- a_i est la distance moyenne du point i aux points du même cluster (mesure de **cohésion**),
- b_i est la distance moyenne du point i aux points des autres clusters (mesure de **séparation**).

Le coefficient de silhouette correspond à la moyenne du coefficient de silhouette pour tous les points.

IV. Applications

Applications

Jeu de données nutrimouse

On utilise les données `nutrimouse` : données microarrays donnant l'expression de 120 gènes préselectionnés pour leurs rôles potentiels dans des problèmes nutritionnels de 40 souris.

Martin et al. (2007)

```
data(nutrimouse)
Expr <- nutrimouse$gene
dim(Expr)
```

```
## [1]  40 120
```