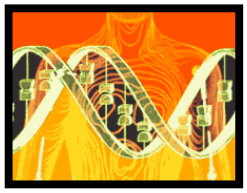
MASTER IBM 2012

# INTRODUCTION À LA FOUILLE DE DONNÉES BIOMÉDICALES

## COURS MASTER IBM 2012



JEAN-DANIEL ZUCKER
NUTRIOMIQUE ET UMMISCO

BLAISE HANCZAR
UNIVERSITÉ PARIS 5

Inserm
Institut national
de la santé et de la recherche médicale





## MODULE FOUILLE DE DONNÉES : DIDACTIQUE

- **Objectifs:**
  - COMPRENDRE LE DOMAINE DE LA FOUILLE DE DONNÉES
  - LE RÔLE DES ANALYSES PRÉDICTIVES DANS L'INFORMATIQUE MÉDICALE ET LA BIOINFO.
  - COMPRENDRE LES ALGORITHMES DE BASE DE LA FOUILLE DE DONNÉES BIOMÉDICALES
  - UTILISER UN ENVIRONNEMENT DE FOUILLE DE DONNÉES : R (WEKA / CLEMENTINE/ ETC.)

- **Examen: Mini-projet en R (Blaise Hanczar)**
- **Lecture: articles clefs à lire (JDZucker**

# Administratif: Module IF-FD Master IBM

- **Mercredi 17 Octobre 2011 –** Intro Générale / Clustering
  - **La fouille de données**
  - **Les données biomédicales**
  - **Le clustering**
  - **Les modèles graphiques (réseaux Bayésiens)**
- **Jeudi 18 Octobre 2011–** fouille de données (Blaise Hanczar)
  - Algorithmes pour la classification supervisée : Arbre de décision, SVM, Réseaux de Neurones, etc.
  - Evaluation
  - Exemples





# Site du cours

http://ouebe.org

PLAN

I. LA FOUILLE DE DONNÉES

II. LES DONNÉES BIOMÉDICALES

III. LE CLUSTERING





## Data rich but information poor! : Besoins d'

Explorer, analyser, compacter, réduire, extraire, utiliser, ces données :

... la fouille de données

the extraction of interesting (non-trivial, implicit, previously unknown and potentially useful) information or patterns from data in large databases

***Définition: "L'exploration et l'analyse de grandes quantité de données afin de découvrir des formes et des règles significatives en utilisant des moyens automatique ou semi-automatique."***

SUPERVISE

## TÂCHES DE LA FOUILLE DE DONNÉES (TYPOLOGIE 1/2)

- **Classification (valeurs discrètes): Oui/Non, 1/2/3, VND/US$/€**
  réponse qualitative à un médicament, classification de demandeurs de crédits, détermination des numéros de fax, dépistage de demandes d'assurances frauduleuses, etc.
- **L'estimation (valeurs continues): [1-10], [-1,1],[0,1000000]**
  réponse quantitative à un médicament, du nombre d'enfants d'une famille, revenu total par ménage, probabilité de réponse à une demande, etc.
- **La prédiction (pour vérifier il faut attendre):** «Dans 2 jours l'action d'apple doublera», demain il fera beau, ...
  durée de vie d'un patient, des clients qui vont disparaître, des abonnés qui vont prendre un service, etc..

**Succès de la tâche**: critère de performances sur nouvelles données





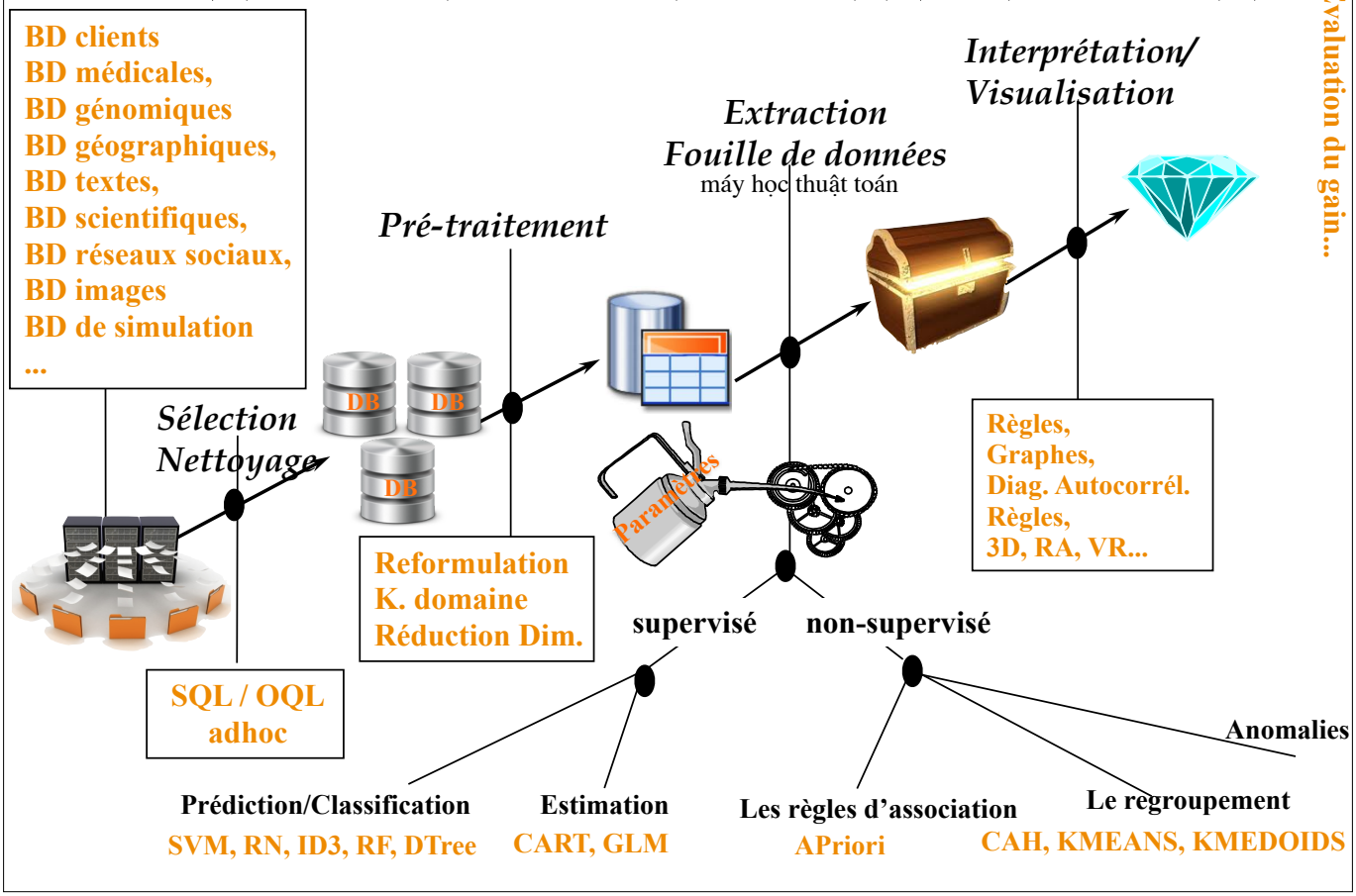
NON SUPERVISE

## TÂCHES DE LA FOUILLE DE DONNÉES (TYPOLOGIE 2/2)

- **Le regroupement par similitudes (Clustering):** des patients qui ont telles mutations génétiques développent tel type d'obésité, etc.
- **La recherche d'association :** «95% des parents qui vont au supermarché acheter des couches (3% des achats) achètent aussi des bierres». 95% est la confiance et 3% le support (Association Rules).
- **La recherche d'anomalie :** «*Il y a une concentration de véhicule «anormale» tous les dimanche matin à 10h près de Nga The*». «L'utilisateur Hung s'est connecté depuis Singapoore alors qu'il ne l'a jamais fait avant».(Anomaly analysis)

**Succès de la tâche**: critère d'intérêt des «connaissances découvertes»

# Le processus de Fouille de données






# Prédiction et Biologie : An evolution

EMBO reports

viewpoint

## The evolution of biology

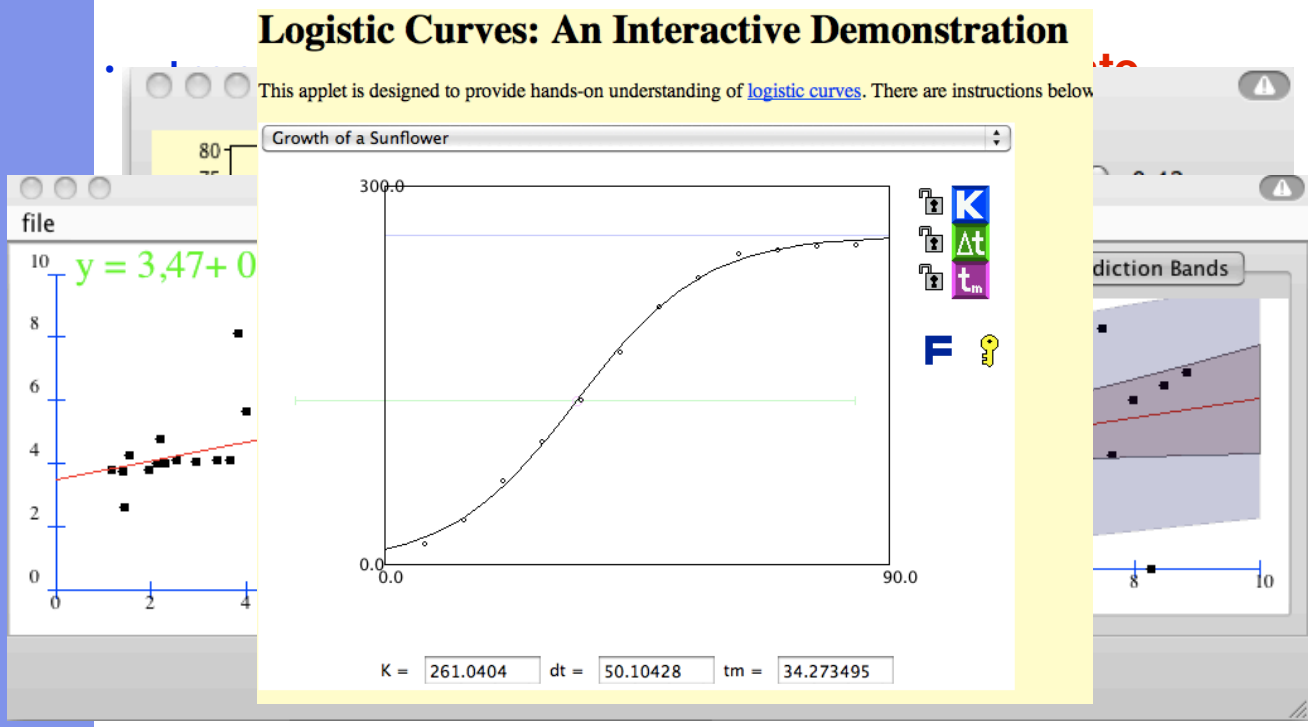
A shift towards the engineering of prediction-generating tools and away from traditional research practice

*Lawrence Kelley & Michael Scott*

L. Kelley and M. Scott, "The evolution of biology. A shift towards the engineering of prediction-generating tools and away from traditional research practice.," *EMBO reports*, vol. 9, no. 12, p. 1163, 2008.

# Lien avec les analyses statistiques connues ?

- *Oui !*

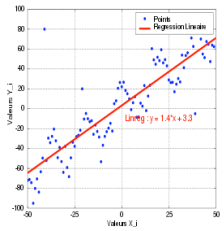


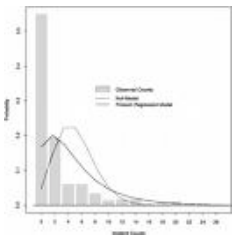





# Exemple 2 : Comparaison, entre les communautés « riche » et « pauvre ». Régression

- Tension artérielle moyennes :
  - Régression
- Proportion d'adultes hypertendus :
  - Régression LOGISTIQUE
- Nombre d'œufs de parasites dans les selles
  - Régression de POISSON





# Exemple 1 (suite) : Expression des résultats. Régression

- **Tension artérielle moyennes : Régression LINEAIRE : la tension artérielle systolique des pauvres des environ 30% plus élevée que celle des riches***
- **Proportion d'adultes hypertendus : Régression LOGISTIQUE : la proportion d'hypertendu est 1,5 plus grande chez les pauvres que chez les riches**
- **Nombre d'Œufs de parasites dans les selles : Régression de POISSON : Le nombre d'œufs de parasites dans les selles est en moyenne 12 fois plus grande chez les riches que chez les pauvres**

* Toute choses étant « égales par ailleurs »





# Dans la fouille: aspect «prédictif»

**Repose sur l'induction: Proposer des lois générales à partir de l'observation de cas particuliers**

**Problème**

**Quel est le nombre *a* qui prolonge la séquence :**

**1 2 3 5 … *a* ?**

## …

- **Solution(s).** Quelques réponses valides :
  - $a = 6$. Argument : c'est la suite des entiers sauf 4.
  - $a = 7$. Argument : c'est la suite des nombres premiers.
  - $a = 8$. Argument : c'est la suite de Fibonacci
  - $a = 2\pi$. ($a$ peut être n'importe quel nombre réel supérieur ou égal à 5)
    Argument : la séquence présentée est la liste ordonnée des racines du polynôme :

    $$P = x^5 - (11 + a)x^4 + (41 + 11a)x^3 - (61 - 41a)x^2 + (30 + 61a)x - 30a$$

    qui est le développement de : $(x - 1) . (x - 2) . (x - 3) . (x - 5) . (x - a)$

- **Généralisation**
  Il est facile de démontrer ainsi que n'importe quel nombre est une prolongation correcte de n'importe quelle suite de nombre

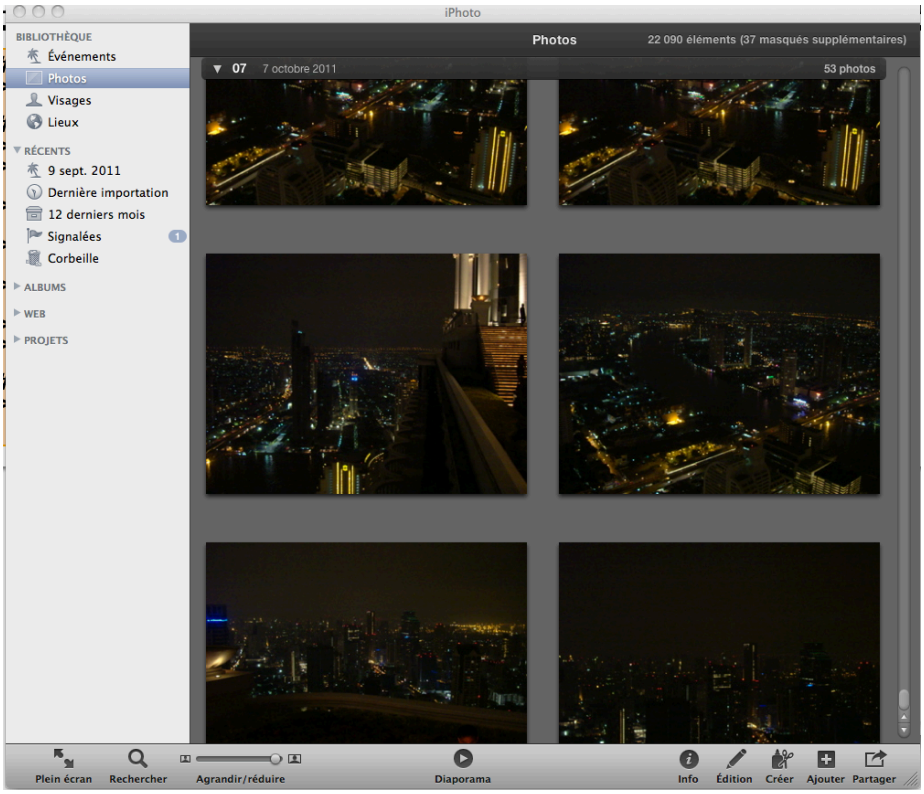
> ***Mais alors …*** ***comment faire de l'induction ?***
>
> ***et que peut-être une science de l'induction ?***





## Apprendre par coeur ? IMPOSSIBLE

# Exemple de systeme utilisation la classification





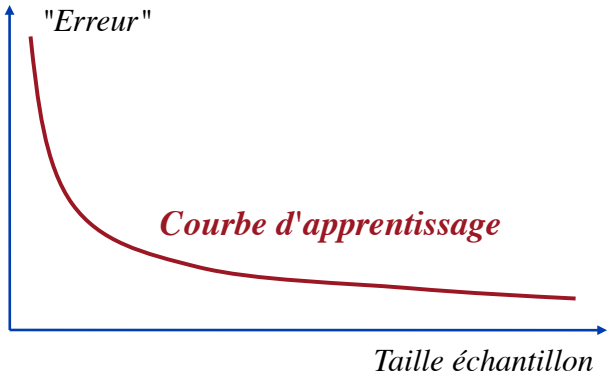



# Représenter

- **Extraction de caractéristiques (descripteurs, attributs)**
  - ***Eliminer*** **les descripteurs non pertinents**
  - ***Introduction*** **de nouveaux descripteurs**
    - **Utilisation de connaissances a priori**
      - **Invariance par translation**
      - **Invariance par changement d'échelle**
    - **Histogrammes**
    - **Combinaisons de descripteurs**
  - ***Ajouter*** **des descripteurs (beaucoup) !!**

# Valider les résultats

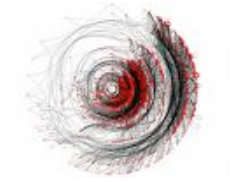
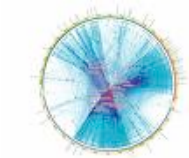
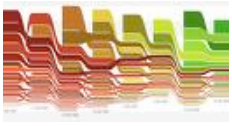
- Quel *critère de performance* (de succès) ?
  - Probabilité de misclassification
  - Risque
  - Nombre d'erreurs
- Apprentissage sur un *échantillon d'apprentissage*
- Test sur une *base de test*

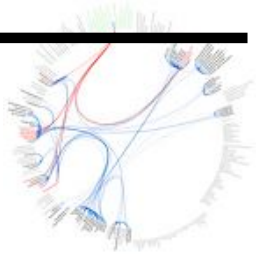
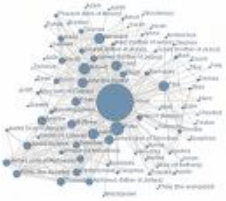


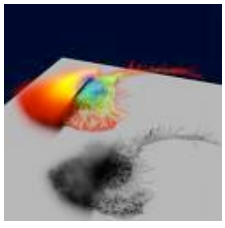





# Visualiser les résultats



http://www.google.org/flutrends/

google.org Suivi de la grippe

Langue : français

Page d'accueil de Google.org (en anglais)

Suivi de la grippe

Sélectionnez un pays/territoire

Accueil

Comment ça marche ?

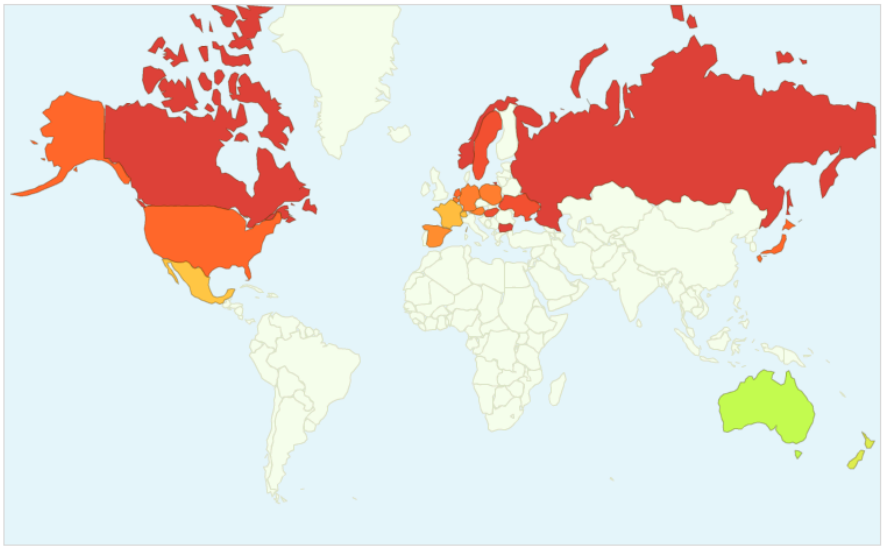
FAQ

Propagation du virus

- Très élevée
- Élevée
- Modérée
- Basse
- Minimale

## Suivez l'évolution de la grippe dans le monde entier

Certains termes de recherche semblent être de bons indicateurs de la propagation de la grippe. Afin de vous fournir une estimation de la propagation du virus, ce site rassemble donc des données relatives aux recherches lancées sur Google. En savoir plus »



Télécharger les données de propagation du virus dans le monde





# LE PROCESSUS DE FOUILLE DE DONNÉES

BD clients
BD médicales,
BD géographiques,
BD textes,
BD scientifiques,
BD cinéphiles,
...

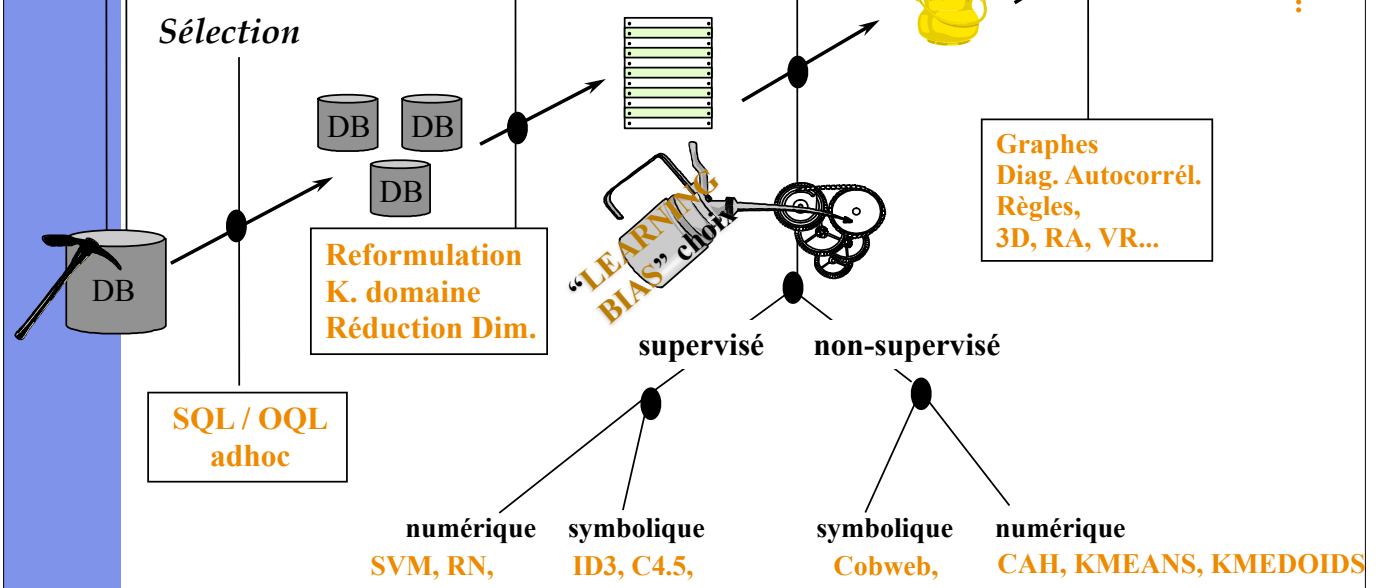
*Sélection*

*Pré-traitement*

*Extraction*
*Fouille de données*

*Interprétation/*
*Visualisation*

*Evaluation du gain...*



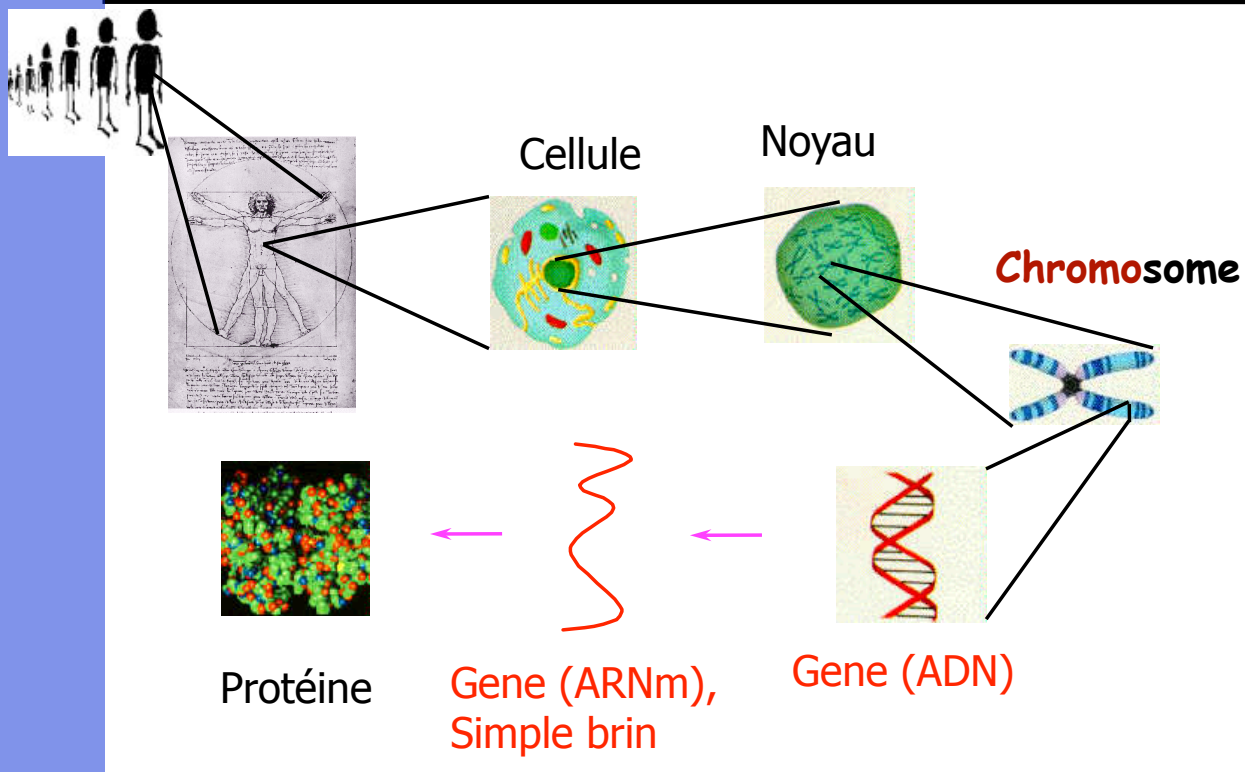
PLAN

I. La fouille de données
II. Les données biomédicales
I. Le clustering





# Niveaux DE l'information biologique




National Human Genome Research Institute

l'**étymologie** de **chromosome**: *chromo* = couleur et *soma* = corps: des corps colorables

# Données Biomédicales

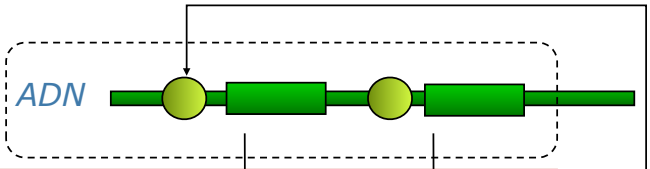
Données écologiques
Données épidémiologiques
Données démographique
Données d'analyses radiographiques
Données d'analyse clinique
Données d'analyse anthropomorphique
Données d'analyse sanguines
Données d'analyse psychologique
Données d'analyse d'effort
Données ...
....
Données 'omiques

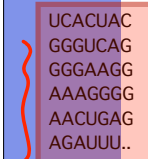




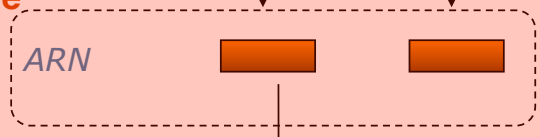
# LES TYPES D'INFORMATION BIOLOGIQUE : LES "OMES" ET OUTILS


**Génome**



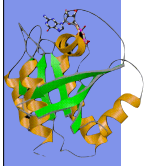



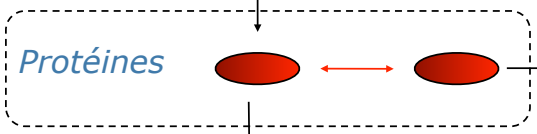

**Transcriptome**




*régulation*

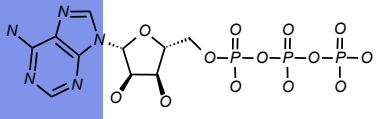
**Protéome**

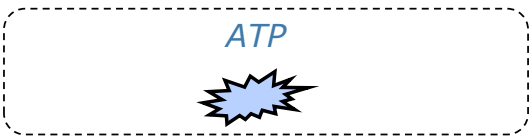




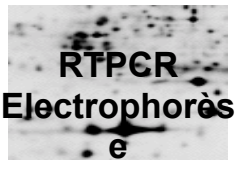

*enzymes*

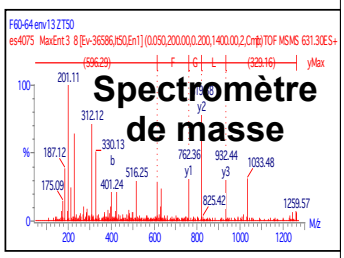
**Métabolome**







# Quelles Types d'informations: «omes» ?

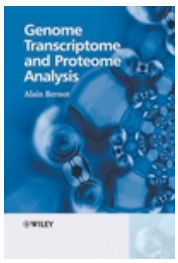
**Génome** (l'ensemble du matériel génétique d'un individu ou d'une espèce.)

**Transcriptome** (l'ensemble des ARN messagers transcrits à partir du génome)

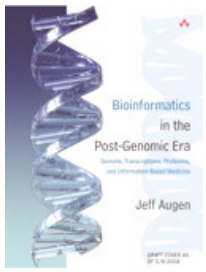
**Protéome** (l'ensemble des protéines exprimés à partir du génome)

**Métabolome** (l'ensemble des composés organiques (sucres, lipides, amino-acides, …)

**Intéractome** (l'ensemble des interactions protéine-protéine)…




*Bioinformatics in the Post-Genomic Era: Genome, Transcriptome, Proteome, and Information-Based Medicine*

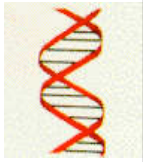




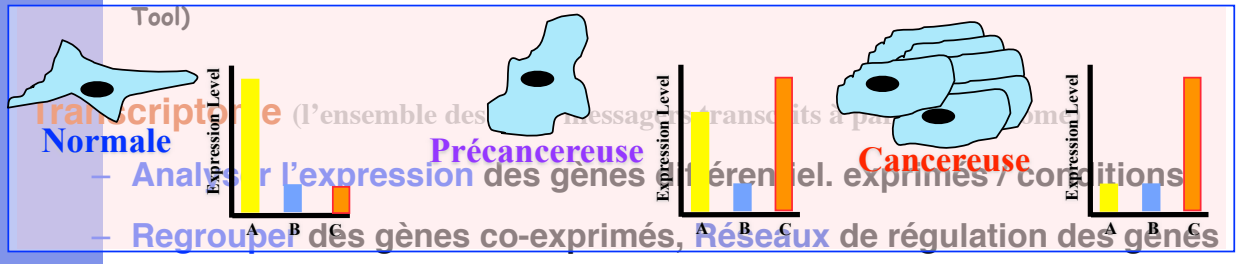



# Problemes algorithmiques et "omes" : le rôle de

**Génome** (l'ensemble du matériel génétique d'un individu ou d'une espèce.)

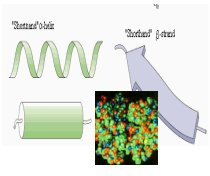


- **Identifier, prédire** les gènes dans une séquence (HMM)
- **Aligner** et comparer de **séquences** ex: BLAST (Basic Local Alignment Search Tool)




**Transcriptome** (l'ensemble des messagers transcrits à partir du génome)

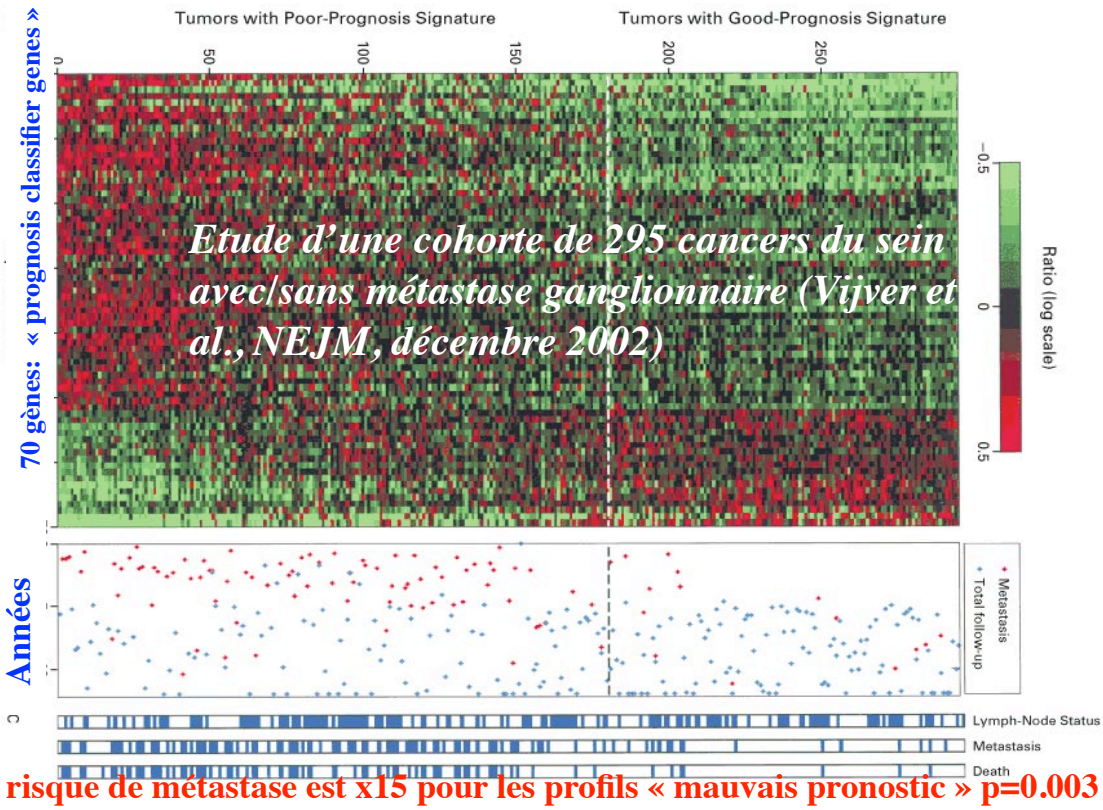
- **Analyser l'expression** des gènes différentiel. exprimés / conditions
- **Regrouper** des gènes co-exprimés, **Réseaux** de régulation des gènes
- **Identifier** la fonction de gènes.



**Protéome** (l'ensemble des protéines exprimés à partir du génome)

- **Prédire** de la **structure** secondaire, la **fonction** des protéines, …
- **Analyser, mesurer l'expression** en fonction des organes

# PREUVE DE CONCEPT




*Etude d'une cohorte de 295 cancers du sein avec/sans métastase ganglionnaire (Vijver et al., NEJM, décembre 2002)*

**Le risque de métastase est x15 pour les profils « mauvais pronostic » p=0.003**



## Niveau de l'information Biologique

DNA
mRNA
Proteins
Informational Pathways
Informational Networks
Cells
Organs
Individuals
Populations
Ecologies

Traditional Biology

'omics

Genomics
Functional Genomics
Proteomics
Metabolomics
Systems Biology
Cellular Biology
Medicine
Medicine
Genetics
Ecology

# Des BD et encore des BD…

- Base de Données ADN
  - GenBank, DDBJ, EMBL,…
- Base de Données Protéines
  - PIR, Swiss-Prot, PRF, GenPept, TrEMBL, PDB,…
- Base de Données EST
  - dbEST, DOTS, UniGene, Gls, STACK,…
- Base de Données Structure
  - MMDB, PDB, Swiss-3DIMAGE,…
- Base de Données voies métabol.
  - KEGG, BRITE, TRANSPATH,…
- Base de Données intégrées
  - SRS

Base de Données de Motifs
- Prosite, Pfam, BLOCKS, TransFac, PRINTS, URLs,…

Base de Données sur les maladies
- GeneCards, OMIM, OMIA,…

Base de Données taxonomique

Base de données littérature scient.
- PubMed, Medline,…

Base de données de brevets
- Apipa, CA-STN, IPN, USPTO, EPO, Beilstein,…

Autres…
- RNA databases, QTL…

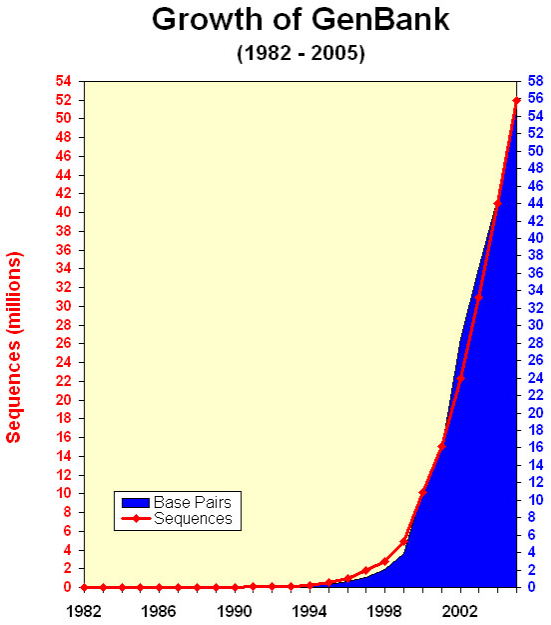


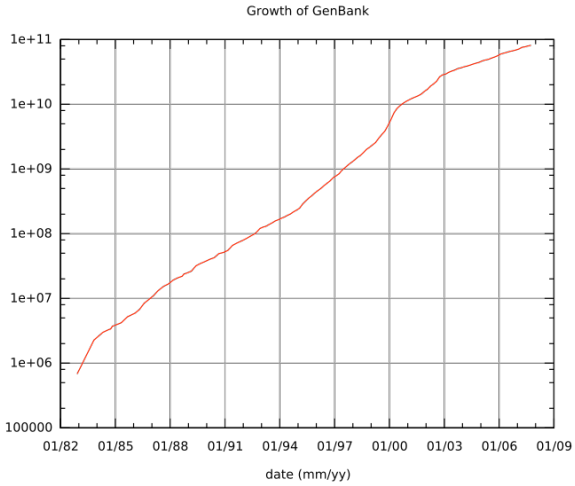


# Données en bioinformatique

- Explosion de la quantité de données (ADN 73 Gb, arrivée des données biopuces, voies métaboliques, …)
- Croissance exponentielle des données (11-15% tous les 3 mois), plus traitable localement
- Données hétérogènes dans leur structure et leur sémantique
- Systèmes d'information hétérogènes
- Beaucoup de connaissances cachées, privées ou inconnues.
- …

# Croissance des données dans GenBank



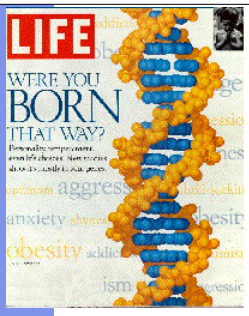

http://www.ncbi.nlm.nih.gov/Genbank/genbankstats.html

http://commons.wikimedia.org/wiki/Image:Growth_of_Genbank.svg





# LES DEFIS

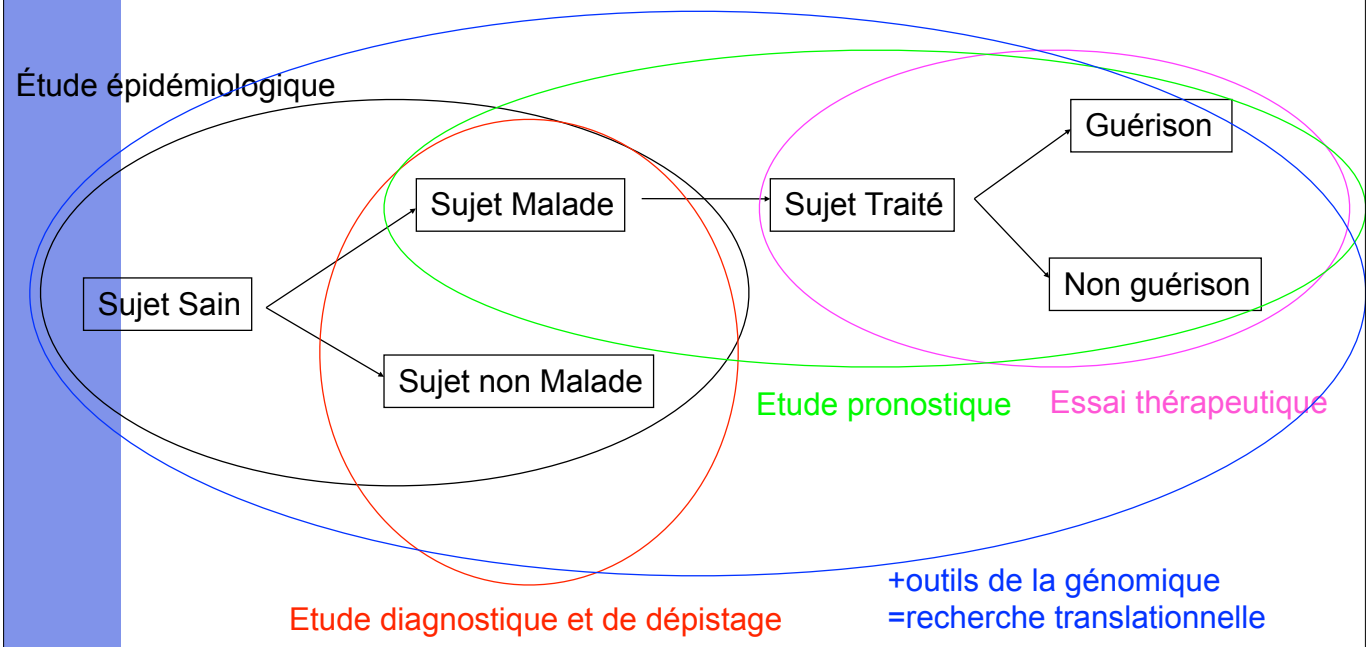






- **Décoder** l'information contenue dans les séquences d'ADN et de protéines
  - Trouver les gènes
  - Différencier entre introns et exons
  - Analyser les répétitions dans l'ADN
  - Identifier les sites des facteurs de transcription
  - Étudier l'évolution des génomes
- **Génomique Comparative**
  - Construire les relations de parenté entre organismes
- **Génomique fonctionnelle**
  - Étudier l'expression des gènes
  - Étudier la régulation des gènes
  - Déterminer les réseaux d'interaction entre les protéines
- **Génomique structurale**:
  - Modéliser les structures 3D des protéines et des ARN structurels
  - Déterminer la relation entre structure et fonction
- **Pharmacogénomique**

# Une révolution pour la recherche clinique








Plan

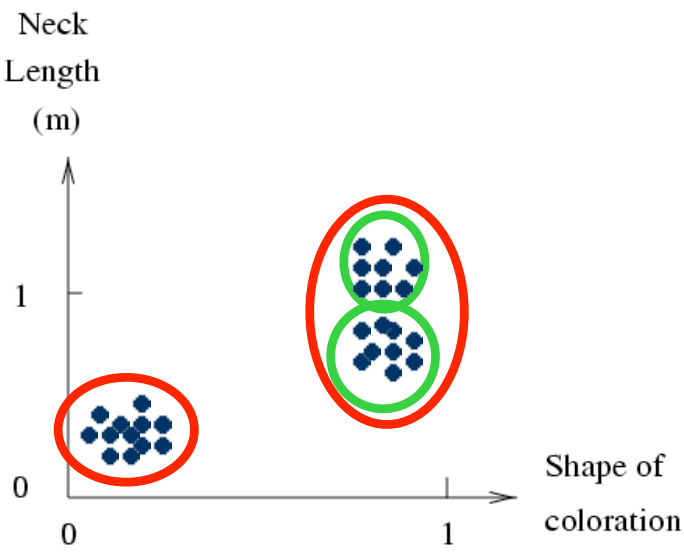
# Introduction

- **Utilisé dans de très nombreux domaines.**
- **But : rassembler les éléments en groupes :**
  - *Homogènes*
    - »Les éléments dans un groupe sont *aussi similaires que possible*
  - *Séparés*
    - »Les éléments de différents groupes sont *aussi différents que possible*

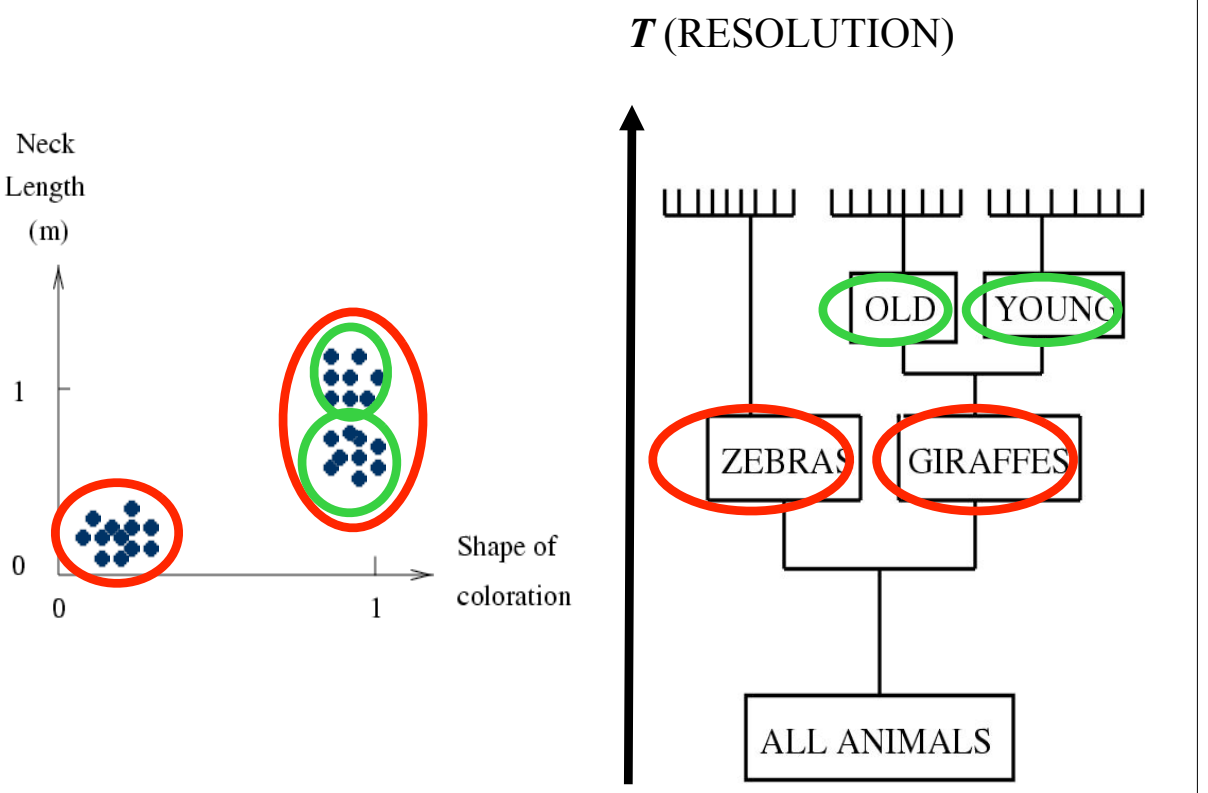




# Exemple de clustering



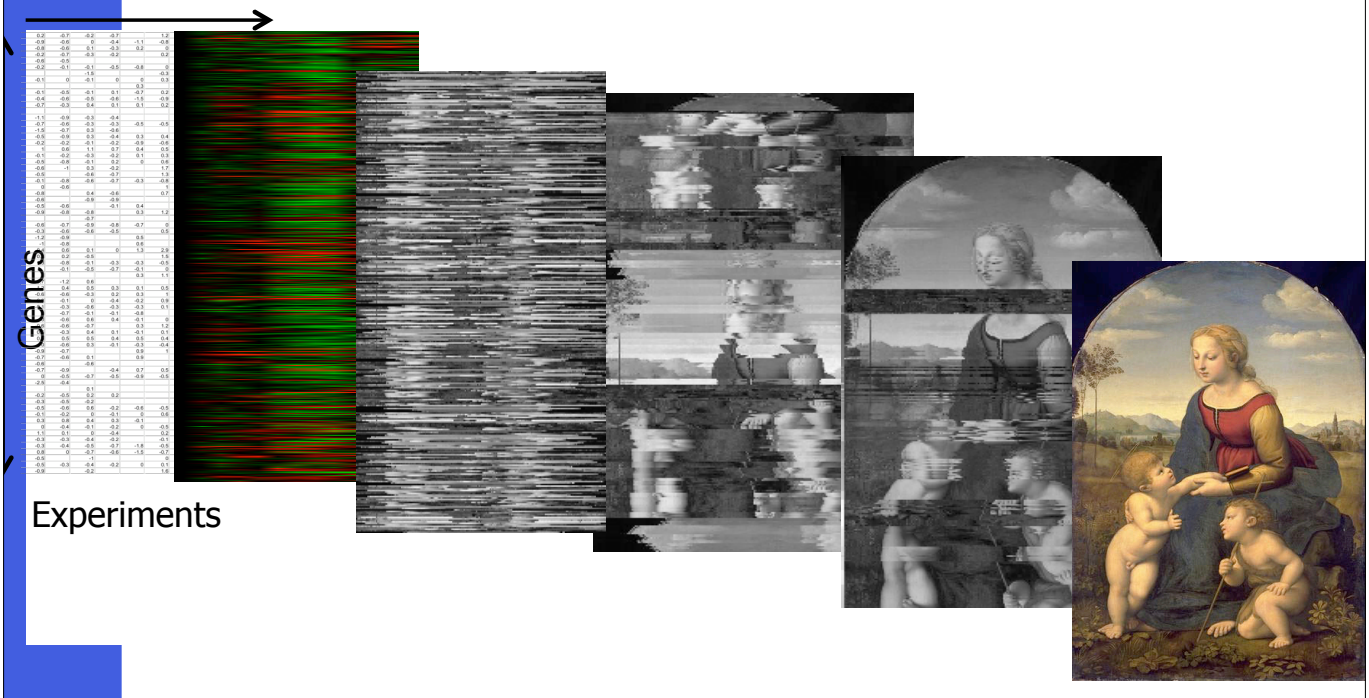
# Notion de dendogramme



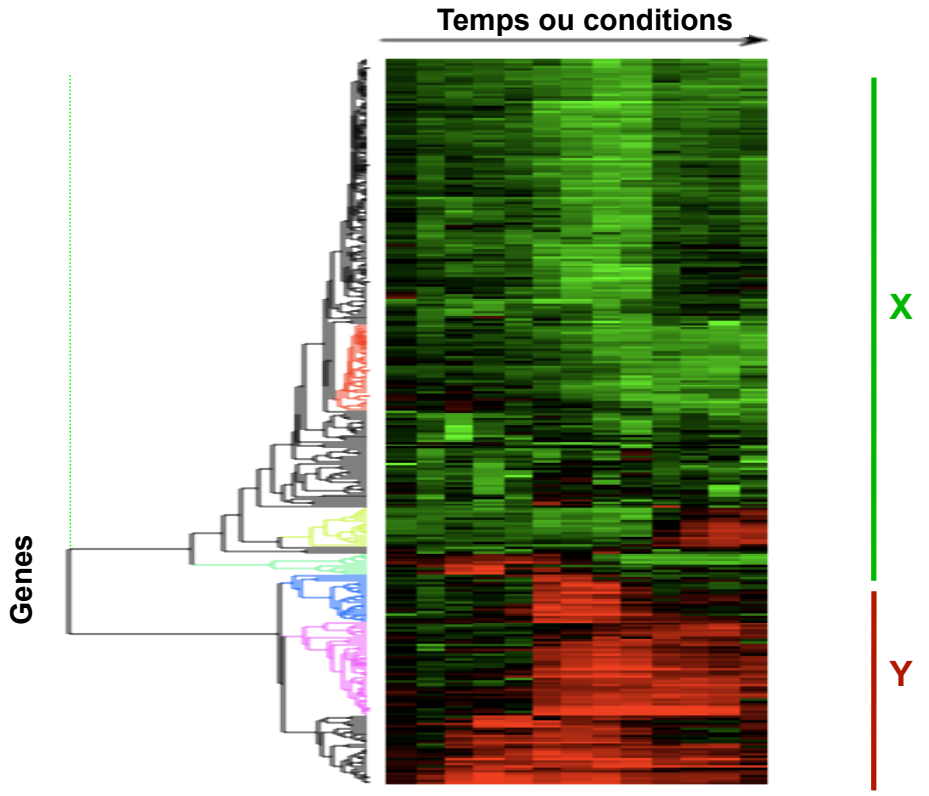





# Faire apparaître des Structures...



# Notion de dendogramme

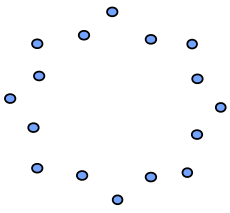




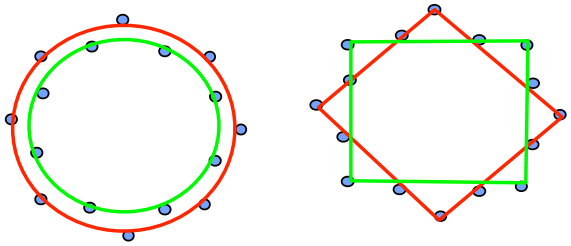



# Définition du problème

- Étant donnée une collection de $n$ "points" $\boldsymbol{X}_i$, $i$=1,2,...$n$, définis dans un espace de dimension $d$, **identifier des structures dans les données**.
- Exemple : **partitionner** les données en $\boldsymbol{K}$ **groupes** (clusters), tels que les points d'un cluster soient **les plus "similaires"**
  - **Identifier des groupes stables**
  - **Générer des dendrogrammes**



- Un **problème mal-posé** :
  - Notion de similarité ?
  - Valeur de $K$ ?



PLAN

A. NOTION DE DISTANCE

B. MÉTHODES ET ALGORITHMES

   A. K-MEANS

   B. SELF ORGANIZING MAPS

   C. CLASSIFICATION ASCENDANTE HIERARCHIQUE (CAH)

C. NOTION DE QUALITÉ D'UN CLUSTER (CHOIX DE K)





# MATRICE DE DISTANCE EN R

```
# Suppression des variables préalablement créées
rm(list=ls(all=TRUE))
# Generation des données
set.seed(160110)
data <- matrix(sample(1:20, 12, replace = FALSE),ncol = 2)
data <- cbind(data,as.matrix(seq(1:6)))
colnames(data) <- c("x","y","point")
rownames(data) <- paste("point",seq(1:6),sep="")
plot(data[,1:2], pch=21, col = "blue",xlim=c(-1, 25), ylim=c(-1, 25))
text(data[,1:2],labels=data[,3],pos=2, offset=0.5,cex=1.7)

# Calcul de la matrice de distances
matd=dist(data[,1:2],method="euclidean")

# Affichage de la matrice de distances au carré
matd^2

hv <- heatmap.2(as.matrix(matd),col = brewer.pal(11,"RdYlGn"),
scale="row", margins=c(10,16), xlab = "Points",main = "Zones de
chaleur")
```

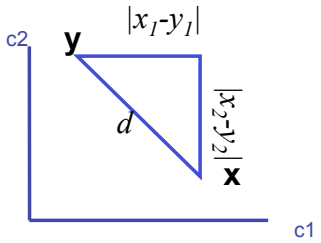
# Type de mesure de (di)similarité

- Un paramètre crucial est la mesure de similarité ou de dissimilarité entre objets
- Pour en citer quelques unes:
  - Euclidian distance
  - Manhattan distance
  - 1 - Pearson's coefficient of correlation
  - Mahalanobis distance
  - $\chi^2$ distance
- Ce choix dépend des données...





# Distance euclidienne




$$d_E = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2}$$

- Cela s'étend naturellement à des espaces de dimensions *p*

$$d_E = \sqrt{\sum_{i=1}^{p} (x_i - y_i)^2}$$

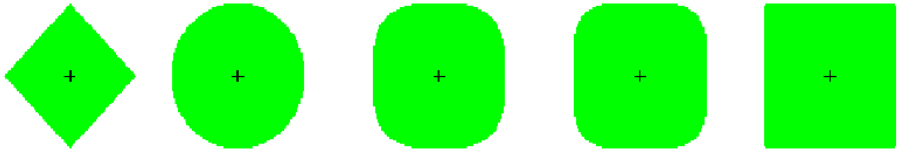
## La norme Lp (généralisation de la distance euclidienne)

# The $L_p$ norm

$$d(x, y) = \sqrt[p]{| x_1 - y_1 |^p + \ldots + | x_n - y_n |^p}$$

p = 2, Euclidean Dist.
p = 1 , Manhattan Dist.

Equidistant points from a center, for different norms



p=1 p=2 p=3 p=4 p=20





## Sous R : Fonction dist()

- t {stats}

**R Documentation**

Distance Matrix Computation

Description

**This function computes and returns the distance matrix computed by using the specified distance measure to compute the distances between the rows of a data matrix.**

Usage

```
dist(x, method = "euclidean", diag = FALSE, upper = FALSE, p = 2)
```

Arguments

**x**

**a numeric matrix, data frame or `"dist"` object.**

**method**

**the distance measure to be used. This must be one of `"euclidean"`, `"maximum"`, `"manhattan"`, `"canberra"`, `"binary"` or `"minkowski"`. Any unambiguous substring can be given....**

# Sous R : les distances

Available distance measures are (written for two vectors *x* and *y*):

`euclidean`:

Usual square distance between the two vectors (2 norm).

`maximum`:

Maximum distance between two components of *x* and *y* (supremum norm)

`manhattan`:

Absolute distance between the two vectors (1 norm).

`canberra`:
*sum(|x_i - y_i| / |x_i + y_i|)*

. Terms with zero numerator and denominator are omitted from the sum and treated as if the values were missing.

This is intended for non-negative values (e.g. counts): taking the absolute value of the denominator is a 1998 `R` modification to avoid negative distances.

`binary`:

(aka *asymmetric binary*): The vectors are regarded as binary bits, so non-zero elements are 'on' and zero elements are 'off'. The distance is the *proportion* of bits in which only one is on amongst those in which at least one is on.

`minkowski`:

The *p* norm, the *p*th root of the sum of the *p*th powers of the differences of the components.

Missing values are allowed, and are excluded from all computations involving the rows within which they occur. Further, when `Inf` values are





# Matrice de distance en R

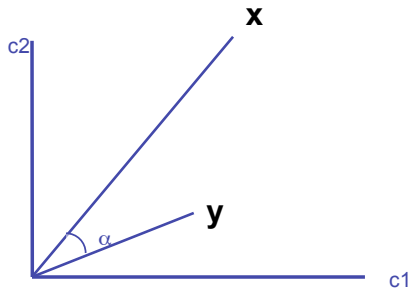
```
# Suppression des variables préalablement créées

rm(list=ls(all=TRUE))

# Generation des données

set.seed(160110)

data <- matrix(sample(1:20, 12, replace = FALSE),ncol = 2)

data <- cbind(data,as.matrix(seq(1:6)))

colnames(data) <- c("x","y","point")

rownames(data) <- paste("point",seq(1:6),sep="")

plot(data[,1:2], pch=21, col = "blue",xlim=c(-1, 25), ylim=c(-1, 25))

text(data[,1:2],labels=data[,3],pos=2, offset=0.5,cex=1.7)

# Calcul de la matrice de distances

matd=dist(data[,1:2],method="euclidean")
```

```
# Calcul de la matrice de distances
matdm=dist(data[,1:2],method="manhattan")
matdm
matdb=dist(data[,1:2],method="binary")
matdb
## example of binary and canberra distances.
x <- c(0, 0, 1, 1, 1, 1)
y <- c(1, 0, 1, 1, 0, 1)
#number of 1 within the ones where at least one is on.
dist(rbind(x,y), method= "binary")      ## answer 0.4
#sum | xi-yi | / | xi | + | yi |  http://people.revoledu.com/kardi/tutorial/Similarity/CanberraDistance.html
dist(rbind(x,y), method= "canberra")  ## answer 2
```

# De la corrélation à une distance

Soit deux objets :

$$r_p = \frac{\sum_{i=1}^{N}(x_i - \bar{x}) \cdot (y_i - \bar{y})}{\sqrt{\sum_{i=1}^{N}(x_i - \bar{x})^2} \cdot \sqrt{\sum_{i=1}^{N}(y_i - \bar{y})^2}}$$




Si on pose cos(α) = $r_p$

**Le coefficient de corrélation n'est autre que le cosinus entre les deux vecteurs centrés !**

Si r = 1, l'angle α = 0, les deux vecteurs sont colinéaires (parallèles).

Si r = 0, l'angle α = 90°, les deux vecteurs sont orthogonaux.

Si r = -1, l'angle α vaut 180°, les deux vecteurs sont colinéaires de sens opposé.

Plus généralement : α = arccos(*r*), où arccos est la réciproque de la fonction cosinus.

Peut être convertit en une distance : $d_p = 1 - r_p$



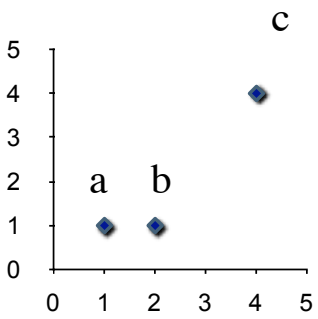


# De la corrélation à la distance sous R

```
## Use correlations between variables "as distance"
library(ellipse)
USJudgeRatings #
coco <- cor(USJudgeRatings)
ord <- order(coco[1,])
xc <- coco[ord, ord]
colors <-
   c("#A50F15","#DE2D26","#FB6A4A","#FCAE91","#FEE5D9","whi
   te","#EFF3FF","#BDD7E7","#6BAED6","#3182BD","#08519C")
plotcorr(xc, col=colors[5*xc + 6])

# Corelation matrix as distance
dd <- as.dist((1 -coco)/2)

# (prints more nicely)
round(1000 * dd)

# to see a heatmap
heatmap(dd)
```

# Impact de la distance

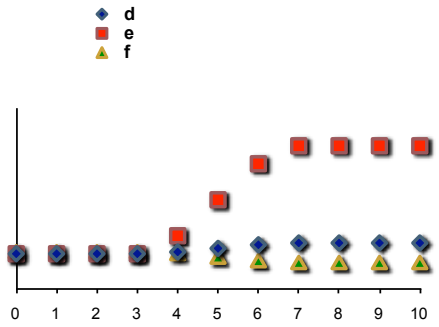
**A**




Distance euclidienne
- **a** proche de **b**

Coefficient de corrélation
- **a** proche de **c**

**B**




Distance euclidienne
- **d** proche de **f**

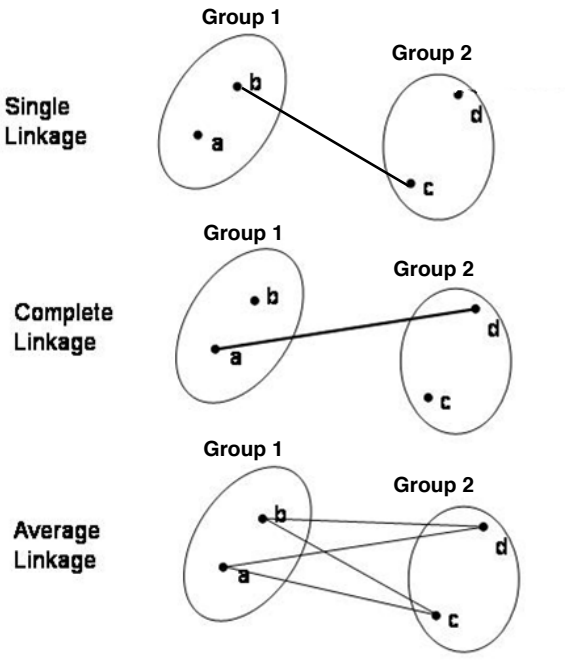
Coefficient de corrélation
- **d** proche de **e**

Classiquement, on utilise la distance euclidienne entre les patients, et une distance de corrélation entre les gènes





# Distance groupe à groupe (de gènes ou de patients)




$$d_{12} = \min_{(x,y)\in(1,2)} d(x,y)$$

$$d_{12} = \max_{(x,y)\in(1,2)} d(x,y)$$

$$d_{12} = \underset{(x,y)\in(1,2)}{moyenne}\, d(x,y)$$

# PLAN

A. NOTION DE DISTANCE

B. MÉTHODES ET ALGORITHMES

   A. K-MEANS

   B. SELF ORGANIZING MAPS

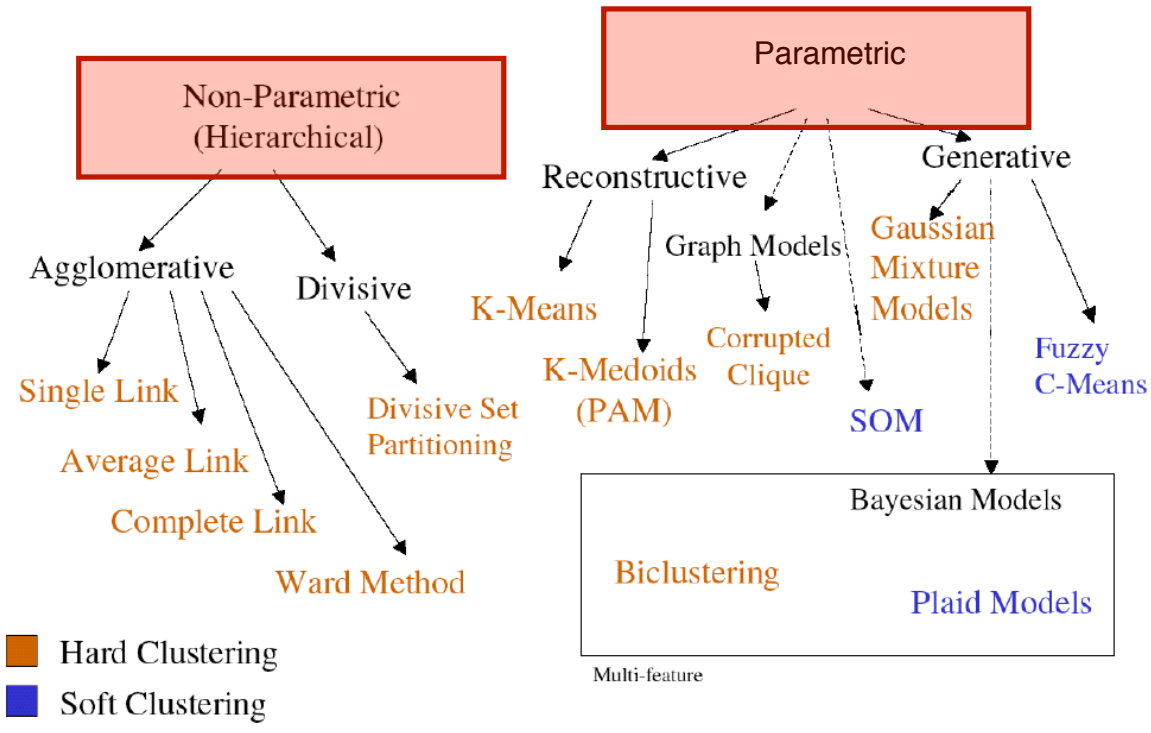
   C. CLASSIFICATION ASCENDANTE HIERARCHIQUE (CAH)

C. NOTION DE QUALITÉ D'UN CLUSTER (CHOIX DE K)





# TAXONOMIE DES MÉTHODES



## Méthodes Paramétrique de partitionnement

Partition des données en un nombre K préspécifié de groupes exhaustifs et mutuellement exclusifs.

Réallocation itérative des observations aux clusters jusqu'à attente d'un critère, par exemple minimiser la somme des carrés intra-cluster ou convergence.

Exemples :

- *k*-means, self-organizing maps (SOM), partitioning around medoids (PAM), etc.
- Fuzzy: besoin d'un modèle stochastique, par ex. Mélanges de gaussiennes.





## Méthodes non-paramétriques hierarchiques

Les méthodes de clustering hiérarchique produisent un arbre ou dendogramme.

Elles ne nécessitent pas de spécifier combien de clusters sont recherchés, puisqu'elles fournissent une partition pour chaque K en coupant l'arbre au niveau voulu.

L'arbre peut être construit de deux manières différentes

- bottom-up: clustering agglomeratif
- top-down: clustering divisif

# Méthodes hierarchiques

**Clustering agglomératif**

- **Commence avec *n* clusters**
- **A chaque étape, fusionne les deux clusters les plus proches en utilisant une mesure de dissimilarité entre clusters, qui reflète la forme des clusters obtenus**

**Clustering divisif**

- **Commence avec un seul cluster**
- **A chaque étape, divise les clusters en deux parties**
- ***Avantages.* Obtient la structure principale des données, ie se concentre sur les niveaux les plus hauts du dendogramme**
- ***Inconvénients.* Difficultés calculatoires quand il considère toutes les divisions possibles en deux groupes**





PLAN

# Méthode des K-means

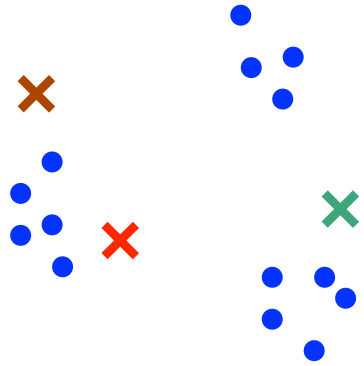
- K-means est une variante d'une méthode plus générale appelée "clustering around mobile centers"
- Les clusters sont définis par leur centre
- Après chaque assignation d'un élément à un centre, la position du centre est recalculée





# Méthode des K-means

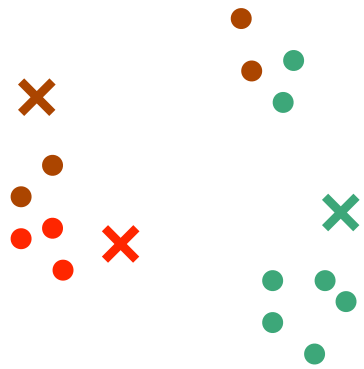
- Initialisation avec des position aléatoires des centres



Itération = 0

# Méthode des K-means

- Initialisation avec des position aléatoires des centres
- Assignation des points aux centres



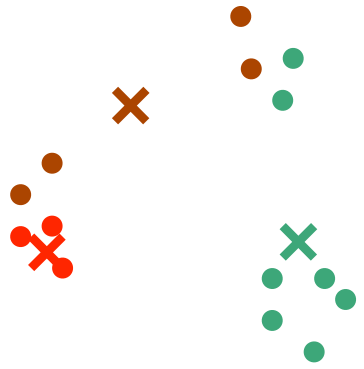
Itération = 1





# Méthode des K-means

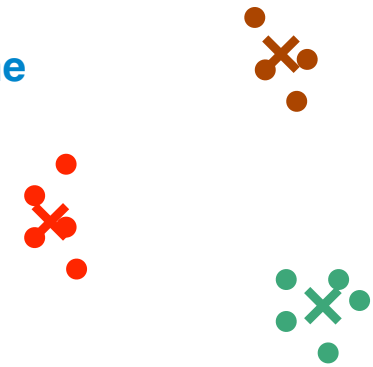
- Initialisation avec des position aléatoires des centres
- Assignation des points aux centres
- Repositionnement des centres



Itération = 2

# Méthode des K-means

- Initialisation avec des position aléatoires des centres
- Assignation des points aux centres
- Repositionnement des centres
- Itération jusqu'à obtention d'une fonction coût minimale

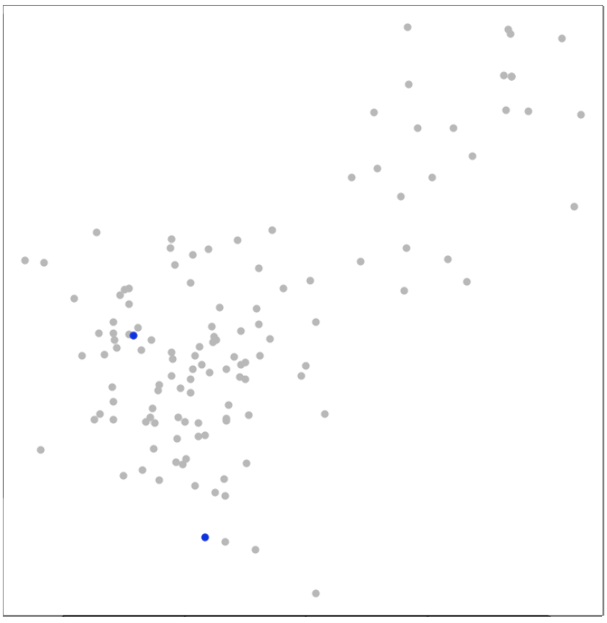


Itération = 3



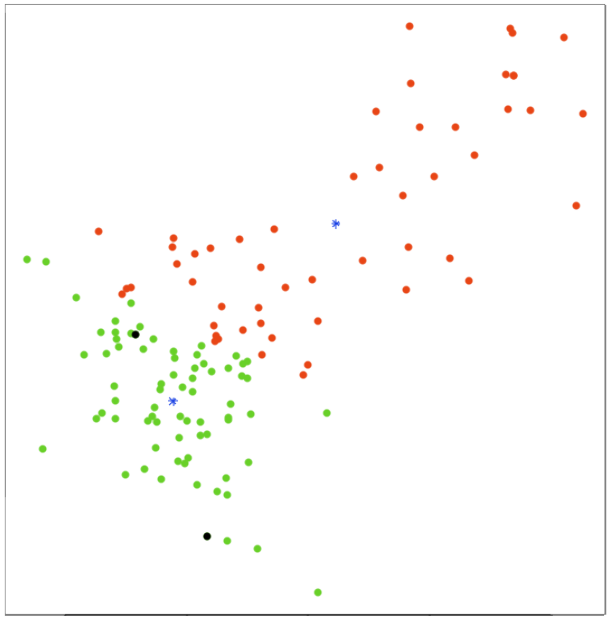


# Exemple – Conditions initiales



- Un ensemble de 125 points a été généré aléatoirement
- Deux points sont aléatoirement choisis comme "seeds" (bleu)

# Exemple – Après 3 itérations

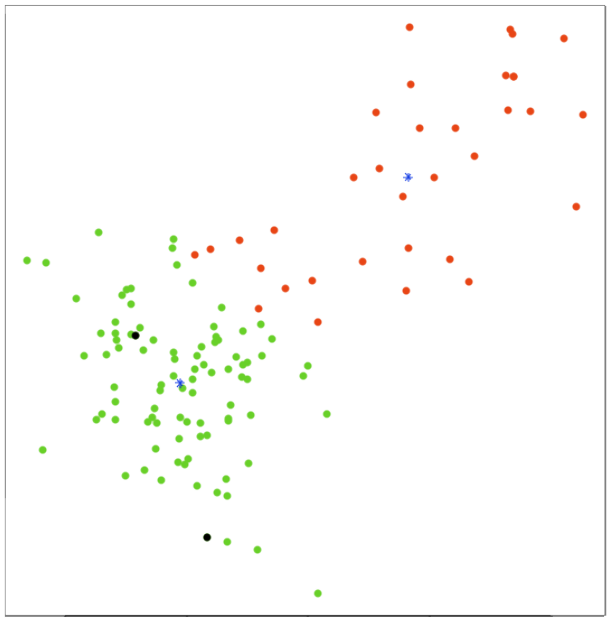


- A chaque étape,
  - les points sont réassignés aux clusters
  - les centres sont recalculés
- Les limites des clusters et les positions des centres évoluent à chaque itération



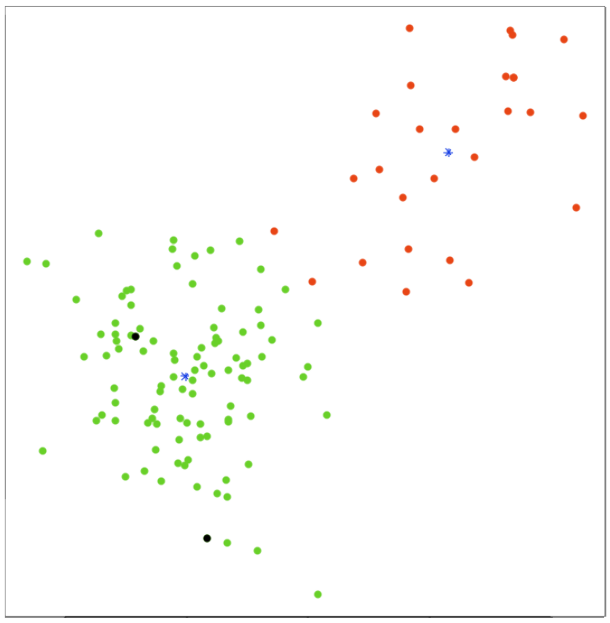


# Exemple – Après 4 itérations



- A chaque étape,
  - les points sont réassignés aux clusters
  - les centres sont recalculés
- Les limites des clusters et les positions des centres évoluent à chaque itération

# Exemple – Après 5 itérations

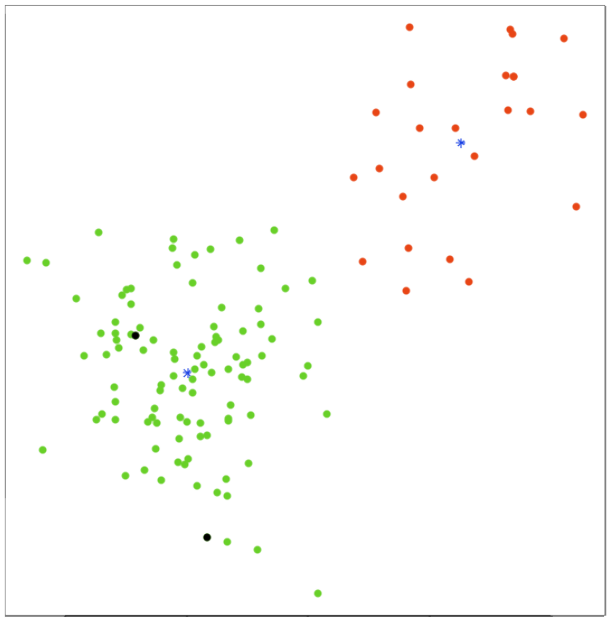


- A chaque étape,
  - les points sont réassignés aux clusters
  - les centres sont recalculés
- Les limites des clusters et les positions des centres évoluent à chaque itération



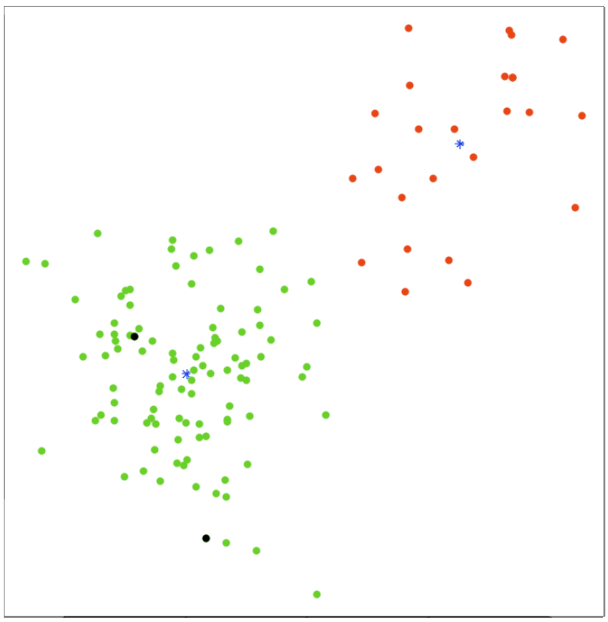


# Exemple – Après 6 itérations



- A chaque étape,
  - les points sont réassignés aux clusters
  - les centres sont recalculés
- Les limites des clusters et les positions des centres évoluent à chaque itération

# Exemple – Après 7 itérations



- Après 7 itérations, les clusters et les centres sont identiques au résultat de l'itération précédente



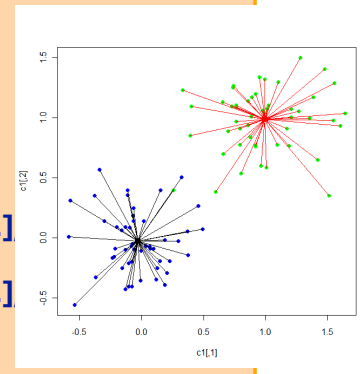


# La méthode des K-means

- Forces
  - Simplicité d'utilisation
  - Rapidité
  - Peut être utilisé avec de très grands jeux de données
- Faiblesses
  - Le choix du nombre de groupes est arbitraire
  - Les résultats dépendent de la position initiale des centres (optimum local)
  - L'implémentation R est basée sur la distance euclidienne, et les autres métriques ne sont pas proposées
- Solutions
  - Essayer différentes valeurs de K et comparer les résultats
  - Pour chaque valeur de K, lancer plusieurs fois l'algorithme avec des conditions initiales différentes
- Faiblesse de la solution
  - Au lieu d'un clustering, on obtient des centaines de clusterings différents parmi lesquels il faut en choisir un…

# K-means et R

```
# Suppression des variables préalablement créées
rm(list=ls(all=TRUE))
# Création d'une matrice de données artificielles contenant deux sous-
      populations
c1 <- matrix(rnorm(100, sd = 0.3), ncol = 2)
c2 <- matrix(rnorm(100, mean = 1, sd = 0.3), ncol = 2)
mat <- rbind(c1, c2)

# Affichage des points de c1 et c2
plot(c1, col="blue", pch=16, xlim=range(mat[,1]), ylim=range(mat[,2]))
points(c2, col="green", pch=16)

# Application de l'algorithme des kmeans
cl <- kmeans(mat, 2, 20)
# Affichage du résultat de kmeans
cl

# Visualisation du résultat des kmeans sur le graphique
points(cl$centers, col=1:2, pch = 7, lwd=3)
segments( mat[cl$cluster==1,][,1], mat[cl$cluster==1,][,2]
       cl$centers[1,1], cl$centers[1,2], col=1)
segments( mat[cl$cluster==2,][,1], mat[cl$cluster==2,][,2]
       cl$centers[2,1], cl$centers[2,2], col=2)
```

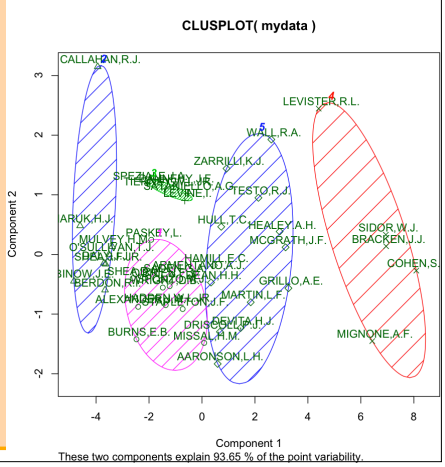







# K-means et Visualisation

```
mydata = USJudgeRatings
# Prepare Data
mydata <- na.omit(mydata) # listwise deletion of missing
mydata <- scale(mydata) # standardize variables

# K-Means Clustering with 5 clusters
fit <- kmeans(mydata, 5)

# Cluster Plot against 1st 2 principal components

# vary parameters for most readable graph
library(cluster)
clusplot(mydata, fit$cluster, color=TRUE,
shade=TRUE,labels=2, lines=0)

# Centroid Plot against 1st 2 discriminant
functions
library(fpc)
plotcluster(mydata, fit$cluster)
```



# K-means et Silhouette

```
library(cluster)
#Silhouette initiale
sil <- silhouette(c(rep(1,50),rep(2,50)),dist(mat))

#Silhouette pour 2 clusters
si2<- silhouette(cl$cluster,dist(mat))
plot(si2, nmax = 80, cex.names = 0.5)



sicah2 <-
silhouette( cutree(hclust(dist(mat)),k=2),dist(mat))
plot(sicah2, nmax = 80, cex.names = 0.5)
```





# Plan

# Self organizing map (SOM) [Kohonen 1982]

- Self-Organizing Maps ou SOM, souvent appelés "réseaux de Kohonen"
- Très proches des algorithmes des centres mobiles
  - représentent les données par un nombre réduit de prototypes (appelés neurones dans les SOM)
  - méthode d'apprentissage compétitive :
    - »le "gagnant" = le neurone le plus proche de l'objet donné
- Différence entre SOM & K-means :
  - préservation de la topologie (Introduit une notion de voisinage entre les classes)
  - les objets les plus semblables seront plus proches sur la carte => visualisation claire des groupements

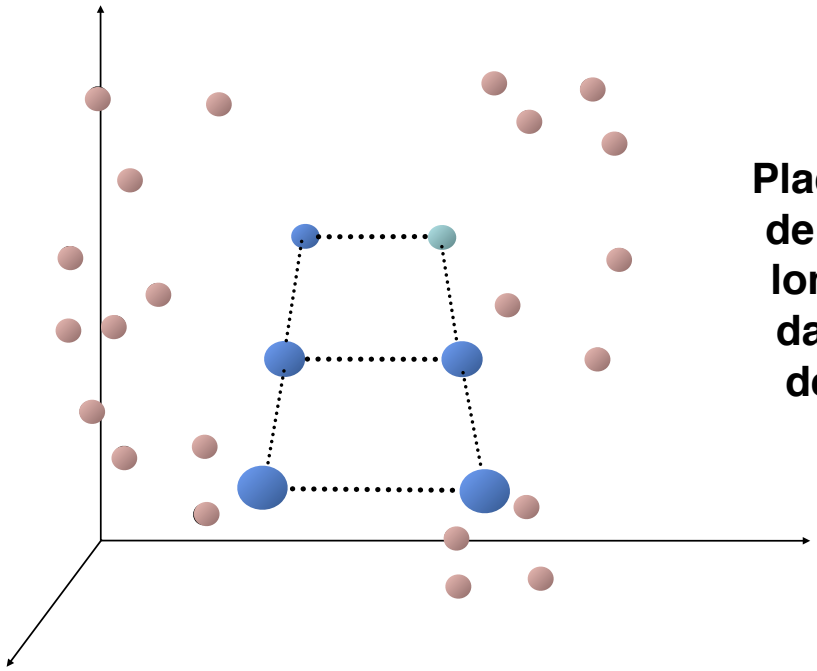




# SOM : Principe

- Les relations de voisinage entre les neurones définissent une topologie et donc un nouvel espace. On a donc 2 espaces :
  - Un espace des entrées dans lequel peuvent être représentés les données et les vecteurs poids des neurones
  - Un espace de sortie (ou carte à 1 ou 2 dimensions) qui contient l'ensembles des neurones et sur lequel une topologie a été définie.
- But de la cartographie associative :
  - Associer chaque vecteur d'entrée à un neurone de la carte
  - On espère que 2 vecteurs proches dans l'espace des entrées exiteront 2 neurones proches sur la carte.

# SOM : PRINCIPE

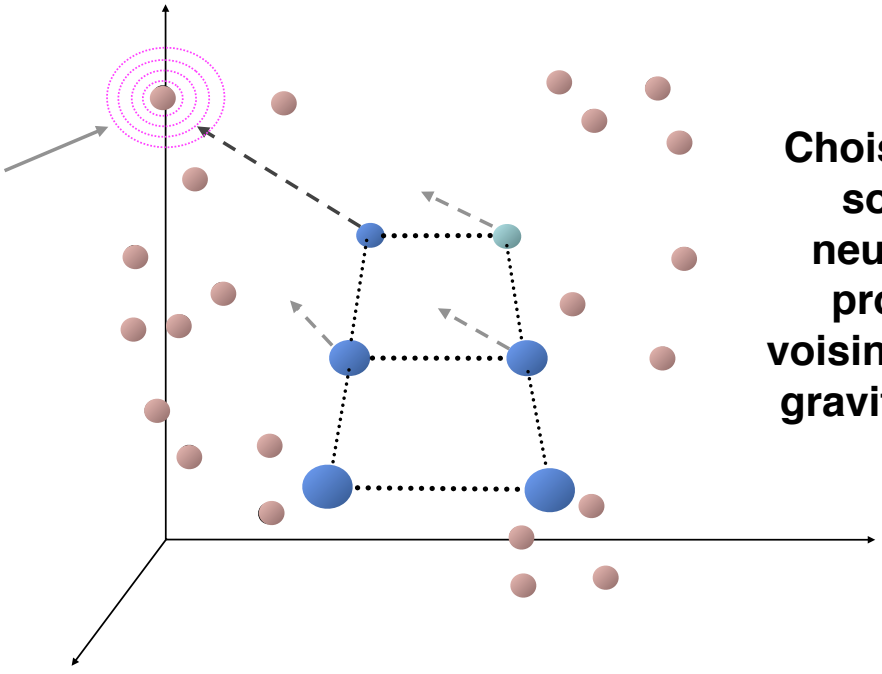


**Placer une grille de neurones le long d'un plan dans l'espace des données**



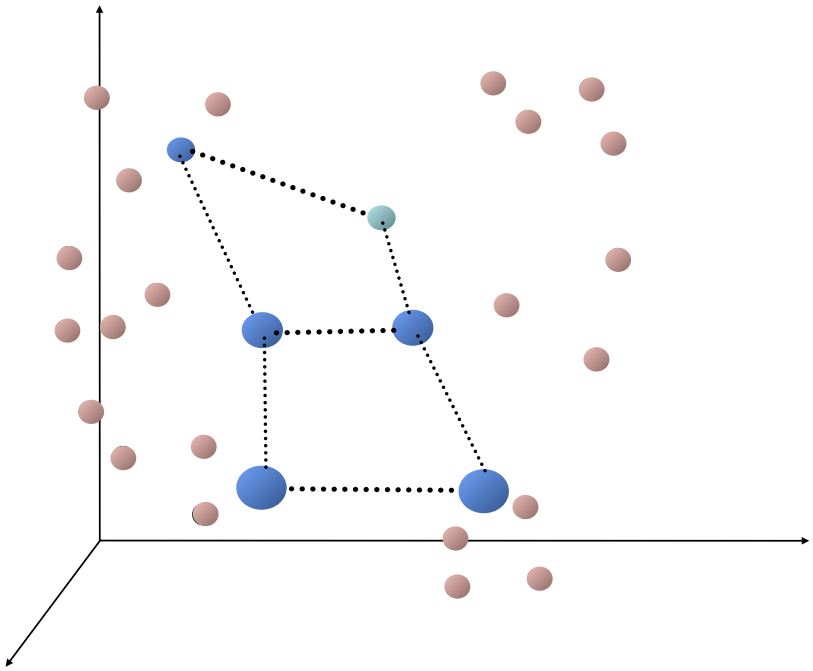


# SOM : PRINCIPE



**Choisir un gène et soumettre le neurone le plus proche et ses voisins à l'influence gravitationnelle du gène**

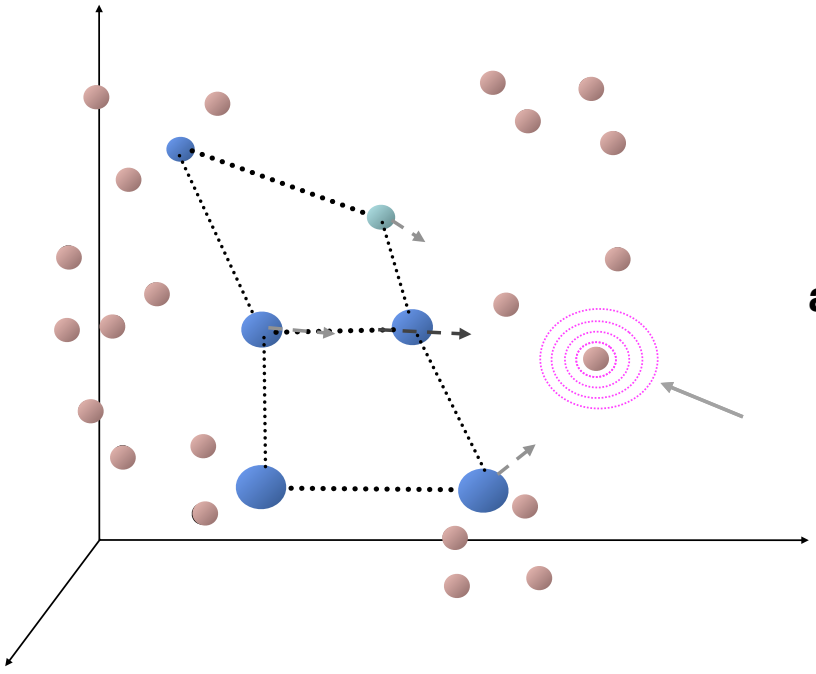
# SOM : PRINCIPE





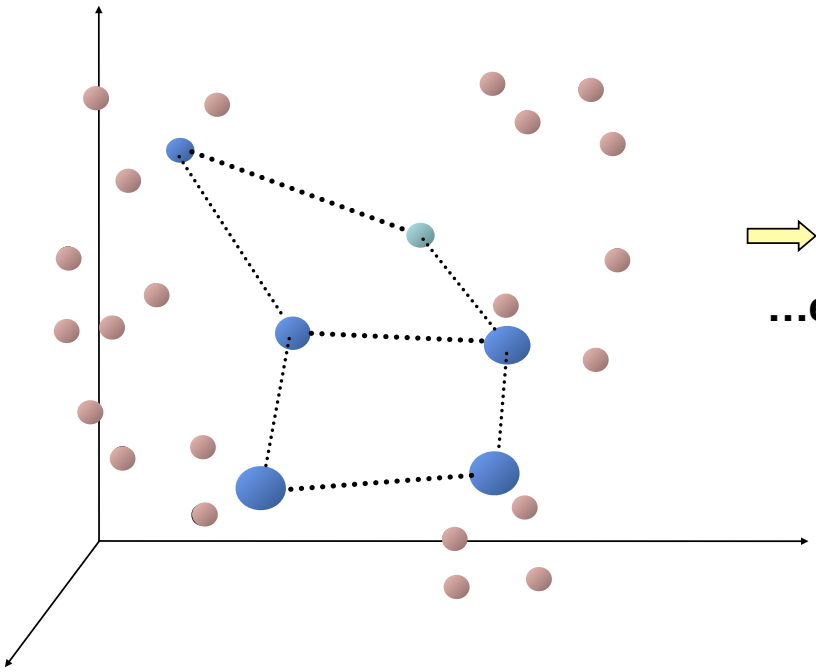


# SOM : PRINCIPE



**Choisir un autre gène…**

# SOM : PRINCIPE

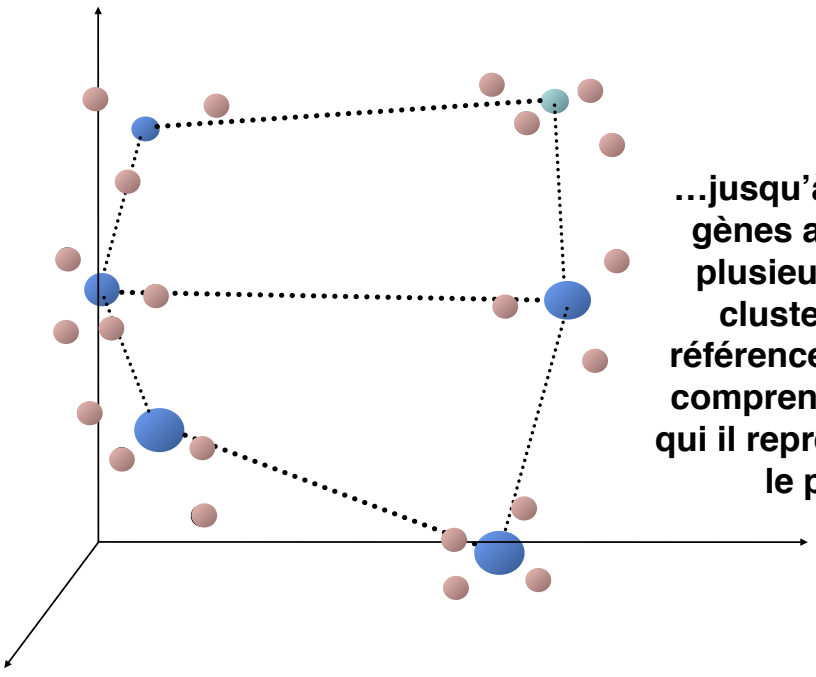


…et ainsi de suite…





# SOM : PRINCIPE



…jusqu'à ce que tous les gènes aient été choisis plusieurs fois. Chaque cluster est défini en référence à un neurone; il comprend les gènes pour qui il représente le neurone le plus proche

# SOM : ALGORITHME

- **Initialiser un ensemble de neurones ou prototypes suivant la topologie choisie**
  - **Architecture de la carte spécifiée => revient à choisir le nb de neurones et définir les relations de voisinage**
- **Boucle de base :**
  - **1. Lire un vecteur d'entrée x**
  - **2. Repérer le neurone k le plus proche de** x **(dist. euclidienne)**

$$\|x - w_k\| < \|x - w_i\| \text{ pour tout } i = k$$

  - **3. Mise à jour : tirer vers** x **les neurones i = {le gagnant k et les autres neurones du voisinage centré sur k}**

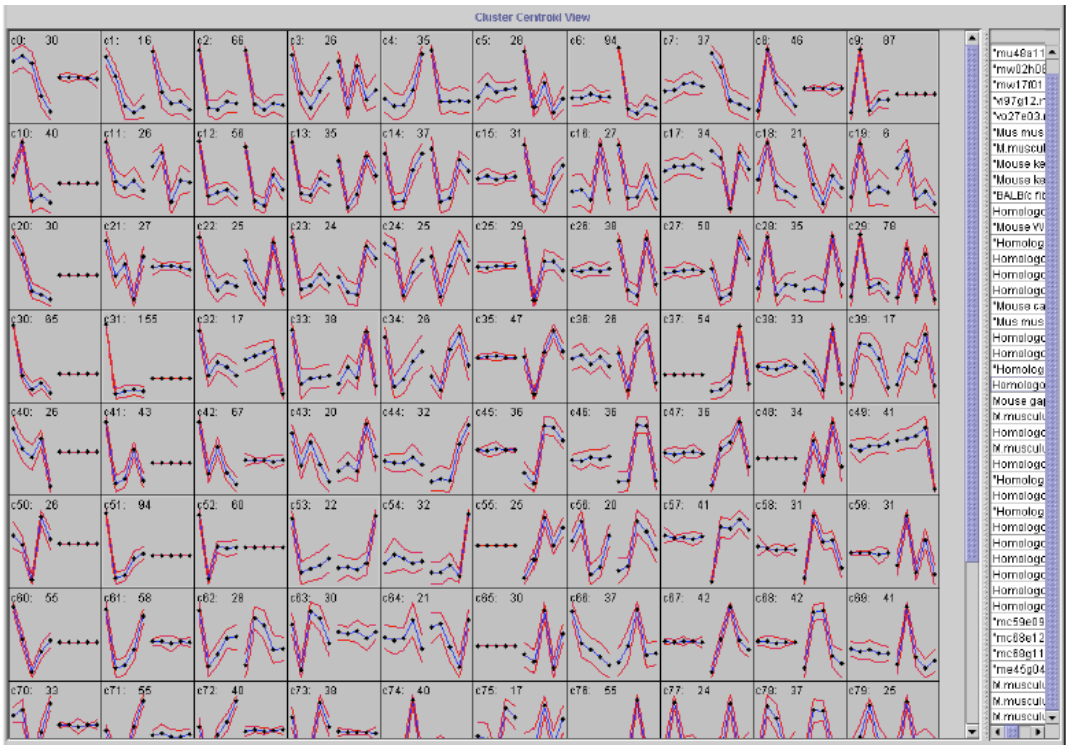
$$w'_i = w_i + h * L(w_i, w_k)(x - w_i)$$

- **L'importance du déplacement : régie par 2 paramètres:**
  - **le pas d'apprentissage h**
  - **la fonction de voisinage L(.) (souvent gaussienne)**





# EXEMPLE DE CLUSTERS OBTENUS PAR SOM



# SOM : Bilan

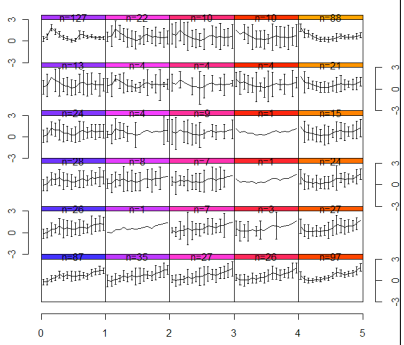
- Robuste aux données incomplètes
- Ajustable aux très grandes bases de données :
  - établissement d'une carte des 78 000 protéines de Swiss-Prot en 2000
  - regroupement d'environ 7 millions de documents (demandes de brevets)
- Essentiellement un outil d'analyse exploratoire et de visualisation.
- Tentatives pour utiliser les SOM à des fin prédictives non concluantes. [Michie et al. 94]





# SOM et R

```
# Suppression des variables préalablement créées
rm(list=ls(all=TRUE))
# Chargement de la librairie SOM
library(som)
# Chargement des données yeast
data(yeast)
# Nettoyage des données
yeast <- yeast[, -c(1, 11)]
yeast.f <- filtering(yeast)
yeast.f.n <- normalize(yeast.f)

# Application de SOM avec une carte 5x6
res <- som (yeast.f.n, xdim=5, ydim=6)
# Visualisation du résultat
plot(res)
```



PLAN

A. NOTION DE DISTANCE

B. MÉTHODES ET ALGORITHMES

A. K-MEANS

B. SELF ORGANIZING MAPS

C. CLASSIFICATION ASCENDANTE HIERARCHIQUE (CAH)

C. NOTION DE QUALITÉ D'UN CLUSTER (CHOIX DE K)





# MÉTHODES HIERARCHIQUES

**Clustering agglomératif**

- **Commence avec *n* clusters**
- **A chaque étape, fusionne les deux clusters les plus proches en utilisant une mesure de dissimilarité entre clusters, qui reflète la forme des clusters obtenus**

**Clustering divisif**

- **Commence avec un seul cluster**
- **A chaque étape, divise les clusters en deux parties**
- ***Avantages.* Obtient la structure principale des données, ie se concentre sur les niveaux les plus hauts du dendogramme**
- ***Inconvénients.* Difficultés calculatoires quand il considère toutes les divisions possibles en deux groupes**

# Classification ascendante hiérarchique : Principe

- **Single linkage method**

$\min(d_{ij})=d_{35}=2$

$$D=\{d_{ij}\}=$$

| | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 1 | 0 | | | | |
| 2 | 9 | 0 | | | |
| 3 | 3 | 7 | 0 | | |
| 4 | 6 | 5 | 9 | 0 | |
| 5 | 11 | 10 | 2 | 8 | 0 |





# Classification ascendante hiérarchique : Principe

- **Single linkage method**

$d_{(35)1}=\min(d_{31},d_{51})=\min(3,11)=3$

$d_{(35)2}=\min(d_{32},d_{52})=\min(7,10)=7$

$d_{(35)4}=\min(d_{34},d_{54})=\min(9,8)=8$

| | (35) | 1 | 2 | 4 |
|---|---|---|---|---|
| (35) | 0 | | | |
| 1 | 3 | 0 | | |
| 2 | 7 | 9 | 0 | |
| 4 | 8 | 6 | 5 | 0 |

$d_{(135)2}=\min[d_{(35)2},d_{12}]=\min(7,9)=7$

$d_{(135)4}=\min[d_{(35)4},d_{14}]=\min(8,6)=6$

| | (135) | 2 | 4 |
|---|---|---|---|
| (135) | 0 | | |
| 2 | 7 | 0 | |
| 4 | 6 | 5 | 0 |

# Classification ascendante hiérarchique : Principe

http://home.dei.polimi.it/matteucc/Clustering/tutorial_html/AppletH.html

- **Single linkage method**

$d_{(135)(24)}=\min[d_{(135)2},d_{(135)4}]=\min(7,6)=6$

| | (135) | (24) |
|---|---|---|
| (135) | 0 | |
| (24) | 6 | 0 |

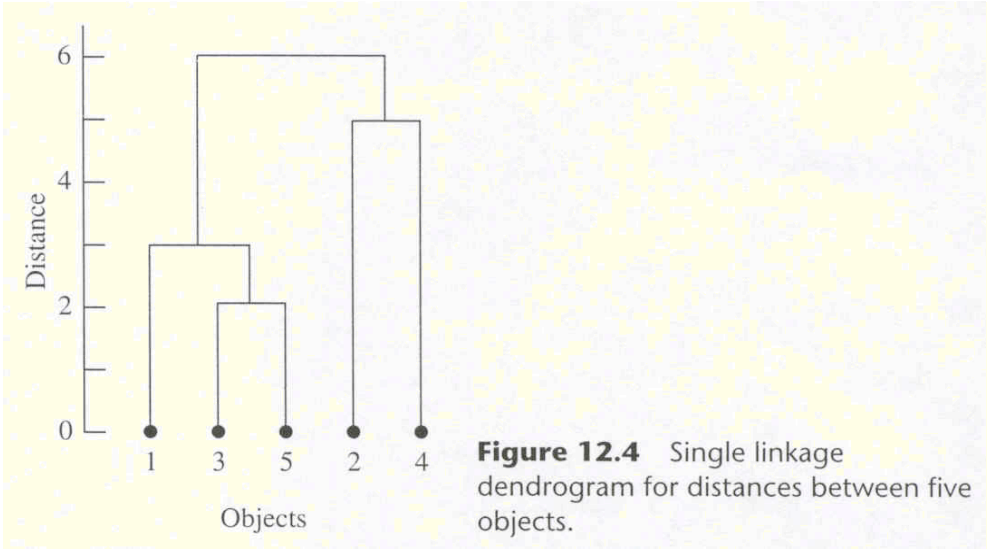


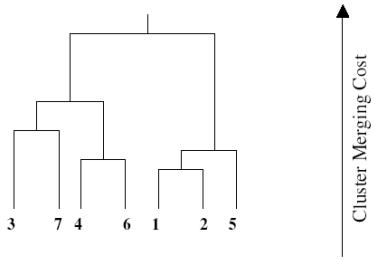

**Figure 12.4** Single linkage dendrogram for distances between five objects.





# Classification ascendante hiérarchique : Algorithme général

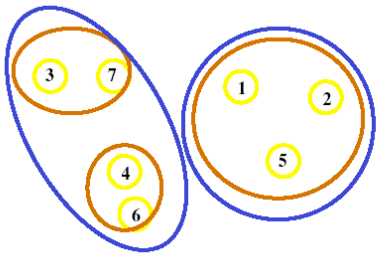
1. Place each element in its own cluster, $C_i=\{x_i\}$
2. Compute (update) the merging cost between every pair of elements in *the set of clusters* to find the two cheapest to merge clusters $C_i$, $C_j$,
3. Merge $C_i$ and $C_j$ in a new cluster $C_{ij}$ which will be the parent of $C_i$ and $C_j$ in the result tree.
4. Go to (2) until there is only one set remaining

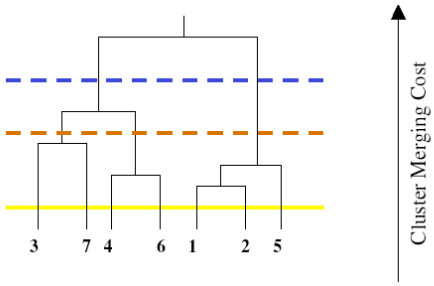



Maximum iterations: n-1

# Classification ascendante hierarchique : Algorithme général

- Input: Data Points, $x_1, x_2, \ldots, x_n$
- Output:Tree
  - the data points are leaves
  - Branching points indicate similarity between sub-trees
  - Horizontal cut in the tree produces data clusters

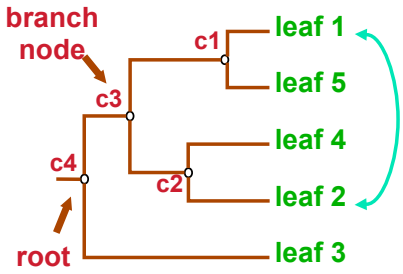


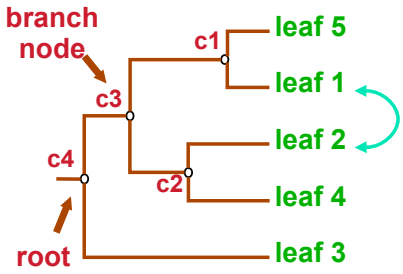








# Isomorphisme d'un arbre




- **Dans un arbre, la distance entre deux feuilles est la somme de la longueur des branches**
- **Les deux feuilles de chaque noeud peuvent être échangées**
- **Les deux arbres présentés ici sont équivalent, cependant**
  - **Arbre du haut: la feuille 1 est loin de la feuille 2**
  - **Arbre du bas: la feuille 1 est voisine de la feuille 2**
- **La distance verticale entre deux feuilles ne reflète pas leur distance réelle !**

# Classification hierarchique : Résumé

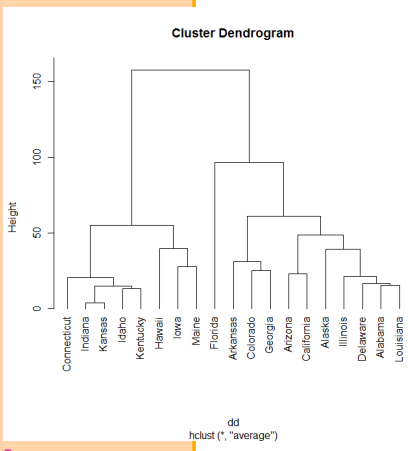
- Le choix de K peut être fait à posteriori
- Les résultats dépendent de la distance entre groupes utilisée
- Processus itératif glouton
- Pas robuste contre le bruit





# Classification hierarchique et R (1/2)

```
# Suppression des variables préalablement créées
rm(list=ls(all=TRUE))
# Chargement des données USArrests
data(USArrests)
# Sélection d'une partie de la base
mat <- USArrests[-c(20:50),]

# Calcul de la matrice de distances
dd <- dist(mat)
# Affichage de la matrice de distances
dd

# Application de l'algorithme de classification
    hiérarchique
hc <- hclust(dd, "average")
# Visualisation du résultat
plot(hc, hang=-1)
```



# CLASSIFICATION HIERARCHIQUE ET R (2/2)

```
cl2 <- cutree(hc,k=2)

# Visualisation du résultat
plot(hc, hang=-1)
rect.hclust(hc, k=3, border="red")

# Ward Hierarchical Clustering with Bootstrapped p values
library(pvclust)
fit <- pvclust(mydata, method.hclust="ward",
   method.dist="euclidean")
plot(fit) # dendogram with p values
# add rectangles around groups highly supported by
the data
pvrect(fit, alpha=.95)
```





# CLASSIFICATION HIERARCHIQUE PUCES ET R

```
# Suppression des variables préalablement créées
rm(list=ls(all=TRUE))

# Chargement de la librarie hybridClust
library(hybridHclust)
set.seed(101)
x <- matrix(rnorm(500),5,100)
x <- rbind(x,x[rep(1,4),]+matrix(rnorm(400),4,100))
x <- rbind(x,x[2:5,]+matrix(rnorm(400),4,100))
par(mfrow=c(1,2))
image(1-cor(t(x)),main='correlation
   distances',zlim=c(0,2),col=gray(1:100/101))
e1 <- eisenCluster(x,'correlation')
plot(e1)
```

PLAN

A. NOTION DE DISTANCE

B. MÉTHODES ET ALGORITHMES

A. K-MEANS

B. SELF ORGANIZING MAPS

C. CLASSIFICATION ASCENDANTE HIERARCHIQUE (CAH)

C. NOTION DE QUALITÉ D'UN CLUSTER (CHOIX DE K)





## INDICE SILHOUETTE

**La silhouette: représentation graphique de la qualité du clustering (Peter J. Rousseeuw,1986)**

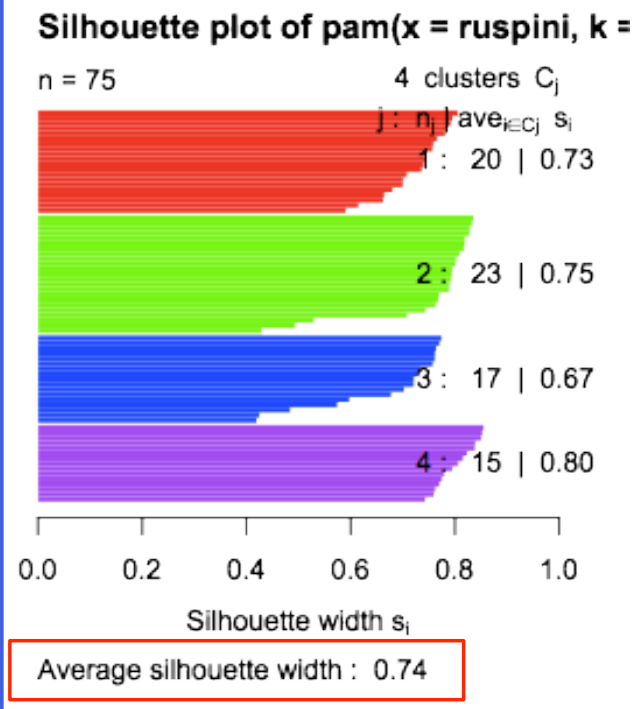
- La silhouette d'un gène *I* est définie comme:

$$s(i) = \frac{b_i - a_i}{\max(a_i, b_i)}$$

- $a_i$ = distance moyenne du gène *i* aux autres gènes du même cluster
- $b_i$ = distance moyenne du gène *i* aux gènes dans le cluster voisin le plus proche

**Représente à quel point un point a bien sa place dans son cluster**

# INDICE SILHOUETTE : EXEMPLE




```
> data(ruspini)
> pr4 <- pam(ruspini, 4)
> str(si <- silhouette(pr4))
 silhouette [1:75, 1:3] 1 1 1 1 1 1 1 1 1
1 ...
 - attr(*, "dimnames")=List of 2
  ..$ : chr [1:75] "10" "6" "9" "11" ...
  ..$ : chr [1:3] "cluster" "neighbor"
"sil_width"
 - attr(*, "Ordered")= logi TRUE
 - attr(*, "call")= language pam(x =
ruspini, k = 4)
>(ssi <- summary(si))
Silhouette of 75 units in 4 clusters from
pam(x = ruspini, k = 4) :
 Cluster sizes and average silhouette
widths:
       20        23        17        15
0.7262347 0.7548344 0.6691154 0.8042285
Individual silhouette widths:
   Min. 1st Qu.  Median    Mean 3rd Qu.
Max.
  0.4196  0.7145  0.7642  0.7377  0.7984
0.8549
> plot(si) # silhouette plot
> plot(si, col = c("red", "green", "blue",
"purple"))# with cluster-wise coloring
```

**Permet de sélectionner le K qui donne le clustering ayant le meilleur indice silhouette**





# SILHOUETTE DANS R

```
library(cluster)
data(ruspini)
#agnes: Computes agglomerative hierarchical clustering of the dataset.
#ar <- agnes(ruspini)
ar <- hclust(dist(ruspini))
#daisy: Compute all the pairwise dissimilarities (distances) between
    observations in the data set.
si3 <- silhouette(cutree(ar, k = 5),dist(ruspini))
# k = 4 gave the same as pam() above


plot(si3, nmax = 80, cex.names = 0.5)
## 2 groups: Agnes() wasn't too good:
si4 <- silhouette(cutree(ar, k = 2), daisy(ruspini))
plot(si4, nmax = 80, cex.names = 0.5)
```

# D'AUTRES MESURES

**Dunn index** [Dunn, 1974]
The Dunn index defines the ratio between the minimal intracluster distance to maximal intercluster distance. The index is given by:

$$D = \frac{d_{min}}{d_{max}},$$

where $d_{min}$ denote the smallest distance between two objects ¿from different clusters, and $d_{max}$ the largest distance of two objects from the same cluster. The Dunn index is limited to the interval [0, $\infty$] and should be maximized.

**Davies-Bouldin index** [Davies & Bouldin, 1979]
This index, DB, is defined as:

$$DB = \frac{1}{n} \sum_{i=1, i \neq j}^{n} max \left( \frac{\sigma_i + \sigma_j}{d(c_i, c_j)} \right),$$

where $n$ is the number of clusters, $\sigma_i$ is the average distance of all patterns in cluster $i$ to their cluster center $c_i$, $\sigma_j$ is the average distance of all patterns in cluster $j$ to their cluster center $c_j$, and $d(c_i, c_j)$ is the distance of cluster centers $c_i$ and $c_j$. Small values of DB correspond to clusters that are compact, and whose centers are far away from each other. Consequently, the number of clusters that minimizes DB is taken as the optimal number of clusters.

**BMC Bioinformatics. 2008 Feb 8;9:90.**
**Modularization of biochemical networks based on classification of Petri net t-invariants.**
Grafahrend-Belau E, Schreiber F, Heiner M, Sackmann A, Junker BH, Grunwald S, Speer A, Winder K, Koch I.





# D'AUTRES MESURES

**C-index** [Hubert, 1976]
The C-index is defined as:

$$C = \frac{S - S_{min}}{S_{max} - S_{min}},$$

where $S$ is the sum of distances over all pairs of objects form the same cluster, $n$ is the number of those pairs and $S_{min}$ is the sum of the $n$ smallest distances if all pairs of objects are considered. Likewise $S_{max}$ is the sum of the $n$ largest distances out of all pairs. The C-index is limited to the interval [0, 1] and should be minimized.

Based on an external cluster validation the validity measures were evaluated and compared on the basis of various sets of t-invariants of different types of Petri nets (i.e. metabolic, gene regulatory and signal transduction nets). With respect to the percentage of correct predictions best results were obtained using the Silhouette Width (75%) and the C-index (75%), followed by the Dunn-index (50%) and DaviesBouldin index (8%). Although offering good results, the C-index is hampered by the fact of showing optimal index values for different numbers of clusters, thus impeding a robust automatic determination of the optimal number of clusters. Given the noisy nature of biological data, robust measures like the Silhouette Width are preferable to noise-sensitive measures like the Dunn index, which is instable against outliers due to the consideration of only two distances. The Davies-Bouldin index requires the computation of the cluster center, which cannot achieved by average determination when dealing with binary data. An inappropriate choice of method for cluster center determination might have been one of the reasons for the insufficient clustering results obtained by this distance measure.

**BMC Bioinformatics. 2008 Feb 8;9:90.**
**Modularization of biochemical networks based on classification of Petri net t-invariants.**
Grafahrend-Belau E, Schreiber F, Heiner M, Sackmann A, Junker BH, Grunwald S, Speer A, Winder K, Koch I.

# Conclusion sur le Clustering

- Une étape clef dans l'analyse des données (en particulier en transcriptomique)
- Trois paramètres clefs: la distance, le nombre de clusters, la validation des clusters
- De très nombreux algorithmes: K-Means, K-medoïds, CAH, SOM, Clustering Spectral, etc.
- Pour les données puces très fréquent recours à la CAH
- Intérêt d'approches réseaux et nécessité d'annotation fonctionnelle pour interpréter les résultats
- Des packages multiples existent sous R pour traiter différents types de données (longitudinales, séries temporelles, etc.) .





# Complément Discretize

```
library(faraway)
rm(list=ls())
myData = iris
tresh = cut(myData$Petal.Length,breaks = 3, labels=
 c("short","medium","long"))
myData$Petal.Length.Symbol = as.factor(tresh)
myData[1:10,]

#http://www.sigmafield.org/2009/09/23/r-function-of-the-day-cut
Nbreaks=3
milieu = levels(cut(myData$Petal.Length, breaks = Nbreaks))
Allbreaks  = cbind(lower = as.numeric( sub("\\((.+),.*", "\\1", milieu) ),
                   upper = as.numeric( sub("[^,]*,([^]]*)\\]", "\\1",
 milieu) ))
BreaksPetal = Allbreaks[1:Nbreaks-1,2]
paste("Le Breakpoint est ",BreaksPetal[1])

myData$Petal.Length.Symbol <- ifelse(
Length                              myData$Petal.Length < BreaksPetal[1],
                           "Small", ifelse(myData$Petal.Length <
 BreaksPetal[2] , "Medium", "Big") )
myData$Petal.Length.Symbol
```

# Complément Discretize 2/3

```
##### Pour le Petal.Length avec un seuil et deux bins
Nbreaks=2
milieu = levels(cut(myData$Sepal.Length, breaks = Nbreaks))
Allbreaks  = cbind(lower = as.numeric( sub("\\((.+),.*", "\\1",
 milieu) ),
                   upper = as.numeric( sub("[^,]*,([^]]*)\\]",
"\\1", milieu) ))
BreaksSepal = Allbreaks[1:Nbreaks-1,2]
paste("Le Breakpoint est ",BreaksSepal[1])

myData$Sepal.Length.Symbol <- ifelse(
                       myData$Sepal.Length < BreaksSepal[1],
                       "Low", "High")
myData$Sepal.Length.Symbol
```





# Complément Discretize 3/3

```
##################
# Nice Magic Trick !!!!
##################
Bool1 = myData$Petal.Length < BreaksPetal[1]  # 1 = Small
Bool2 = myData$Petal.Length > BreaksPetal[2]  # 1 = Big
Bool3 = myData$Sepal.Length < BreaksSepal[1]  # 1 = low
num_code <- ( 1*as.numeric(as.logical(Bool1))
            + 2*as.numeric(as.logical(Bool2))
            + 4*as.numeric(as.logical(Bool3)) ) # values are 0,1,2,...,7
table(num_code)
# then
myData$Petal.Symbol <- c("Medium-High"  , # 0 = F,F,F
 "Small-High"  , # 1 = T,F,F
 "Big-High"  , # 2 = F,T,F
 "Invalid"  , # 3 = T,T,F
 "Medium-Low", # 4 = F,F,T
 "Small-Low" ,# 5 = T,F,T
 "Big-Low",# 6 = F,T,T
 "Invalid" # 7 = T,T,T
 )[num_code+1]
table(myData$Petal.Symbol)
```

```
 Big-High      Big-Low Medium-High  Medium-Low    Small-Low
       39            7           22           32           50
```