

# **INTRODUCTION à l'INTELLIGENCE ARTIFICIELLE**

## **MODELES D'APPRENTISSAGE**

**Pr Pascal Roy MD PhD  
21 Novembre 2022**

# 1. VARIABILITE

# Une Problématique (1)

Un patient n'est jamais identique à un autre !

sa maladie, sa réponse au traitement, son pronostic, tout varie ....

→ **Comment Analyser, Comprendre, Décider  
dans un monde où la variabilité est la règle ?**

# Une Problématique (2)

*Si les patients étaient identiques entre eux, il n'y aurait pas besoin de biostatistique – mais il n'y aurait pas de médecine non plus : un médecin disposant de la « notice de fonctionnement de l'homme » y suffirait.*

Valleron AJ *Préface*. Beuscart R, Benichou J, Roy P, Quantin C. *Biostatistique*, Omniscience<sup>®</sup>

## 2. MODELES

# MODELES

## Linéaire

$$Y = \beta_0 + \sum \beta_j X_j$$

en univarié  $Y = \beta_0 + \beta_1 X_1$

### 1. Diagnostique (ex logistique)

$$P(Y = 1|X) = \frac{e^{\beta_0 + \sum \beta_j X_j}}{1 + e^{\beta_0 + \sum \beta_j X_j}} = \frac{1}{1 + e^{-(\beta_0 + \sum \beta_j X_j)}}$$

### 2. Pronostique (ex modèle de survie de Cox)

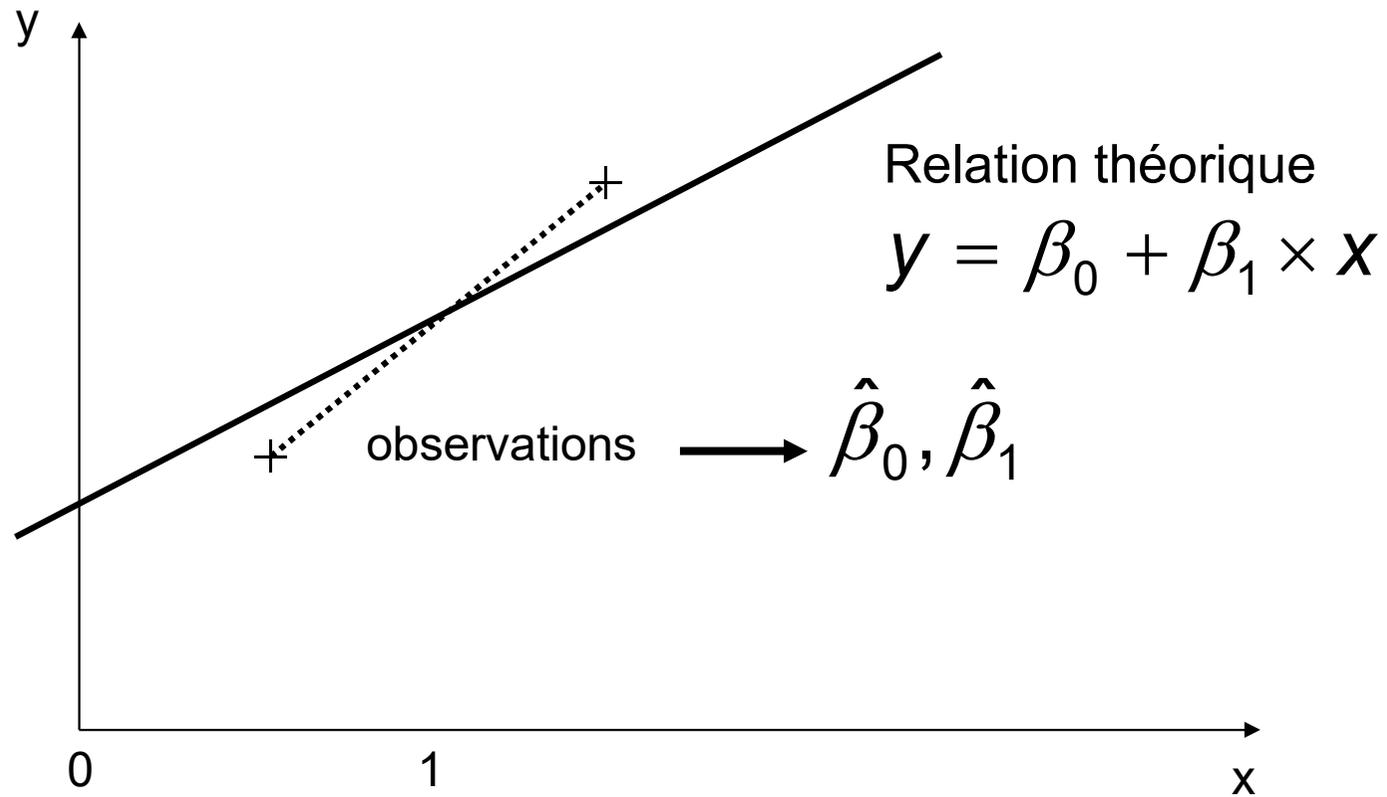
$$\lambda(t, X) = \lambda(t, \mathbf{0}) \exp\left(\sum \beta_j X_j\right)$$

### 3. Théranostique (avec interation)

*... plus tard dans le cursus ...*

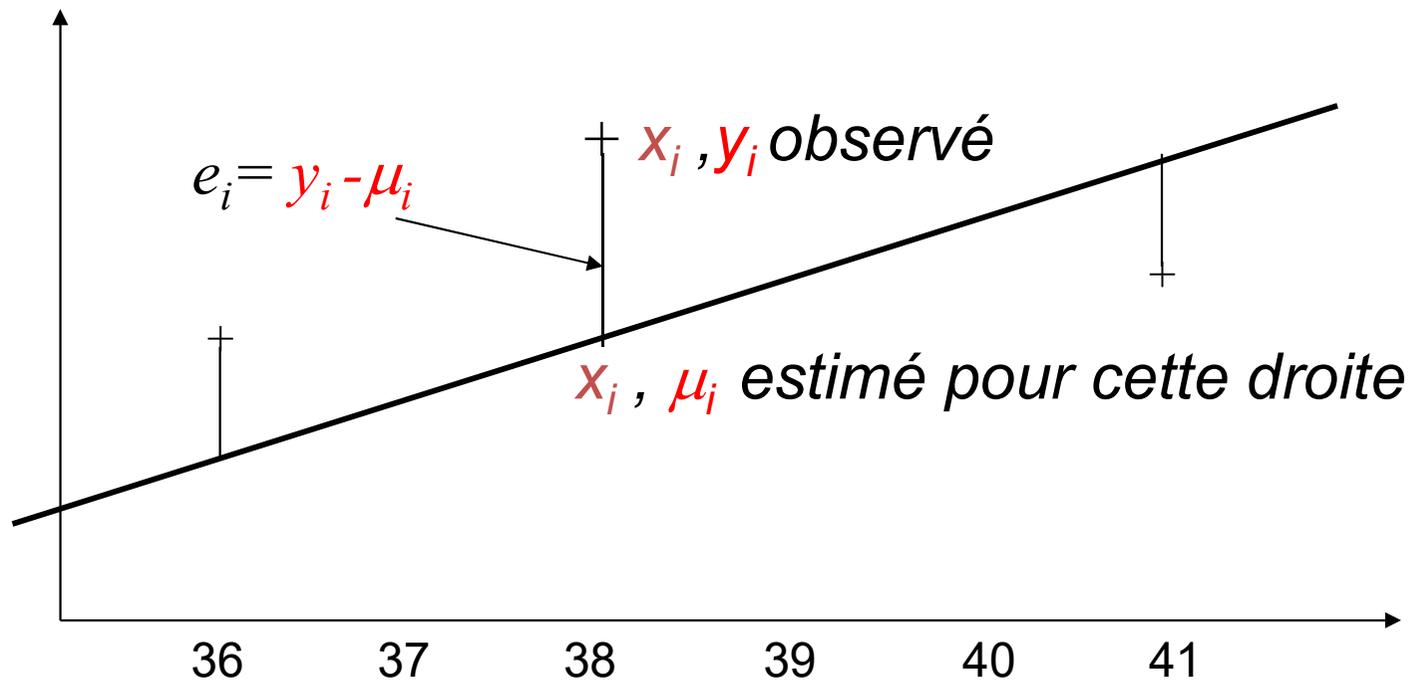
# 3. ESTIMATION

# Modèle Linéaire



# Modèle linéaire et Moindres carrés

Choisir la droite qui minimise la somme des carrés des écarts  $e_i$



# *Estimation* et *estimateur* des moindres carrés

Choisir les paramètres de sorte que

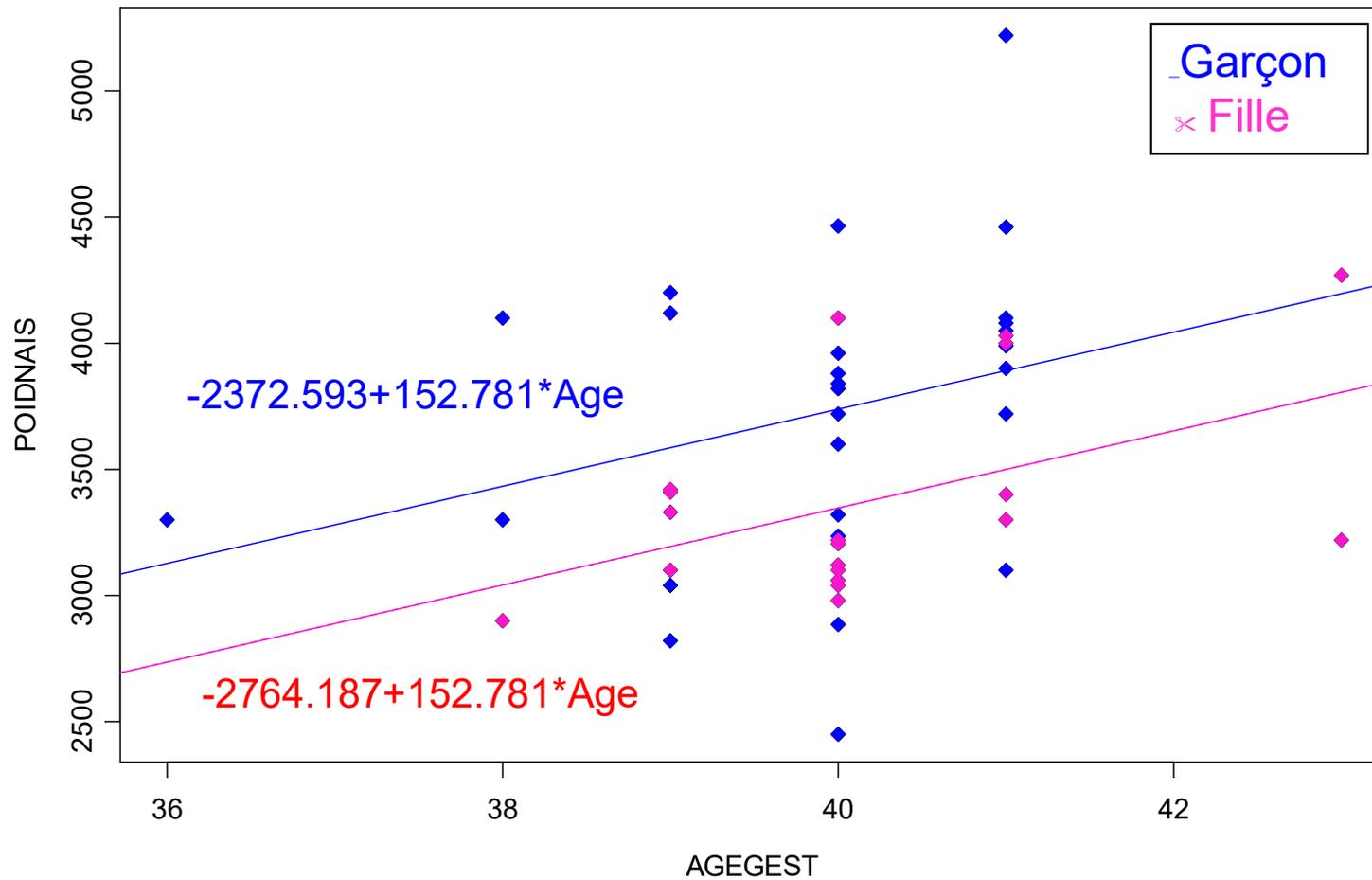
$$\sum_{i=1}^n e_i^2 = \sum_{i=1}^n (y_i - \mu_i)^2 = \sum_{i=1}^n [y_i - (\beta_0 + \beta_1 x_i)]^2 \quad \text{soit le plus petit possible}$$

Les valeurs  $\hat{\beta}_0, \hat{\beta}_1$  qui rendent minimale cette somme des carrés des écarts sont les *estimations* des paramètres.

La fonction qui associe *ces valeurs des paramètres* à un *échantillon* s'appelle un *estimateur*.



# Poids à la naissance / Age gestationnel



# Vraisemblance

- La vraisemblance de **la valeur d'un paramètre** est la probabilité des données si le paramètre a cette valeur
  - C'est donc une quantité qui mesure « l'accord » des données avec la valeur de ce paramètre
- La méthode du **maximum de vraisemblance** consiste à choisir pour estimation du (*ou des*) paramètre(s) la valeur qui a la vraisemblance maximale
  - C'est donc une méthode générale de construction d'estimateurs
- La **Vraisemblance d'un modèle** est la vraisemblance des estimations du maximum de vraisemblance de ses paramètres

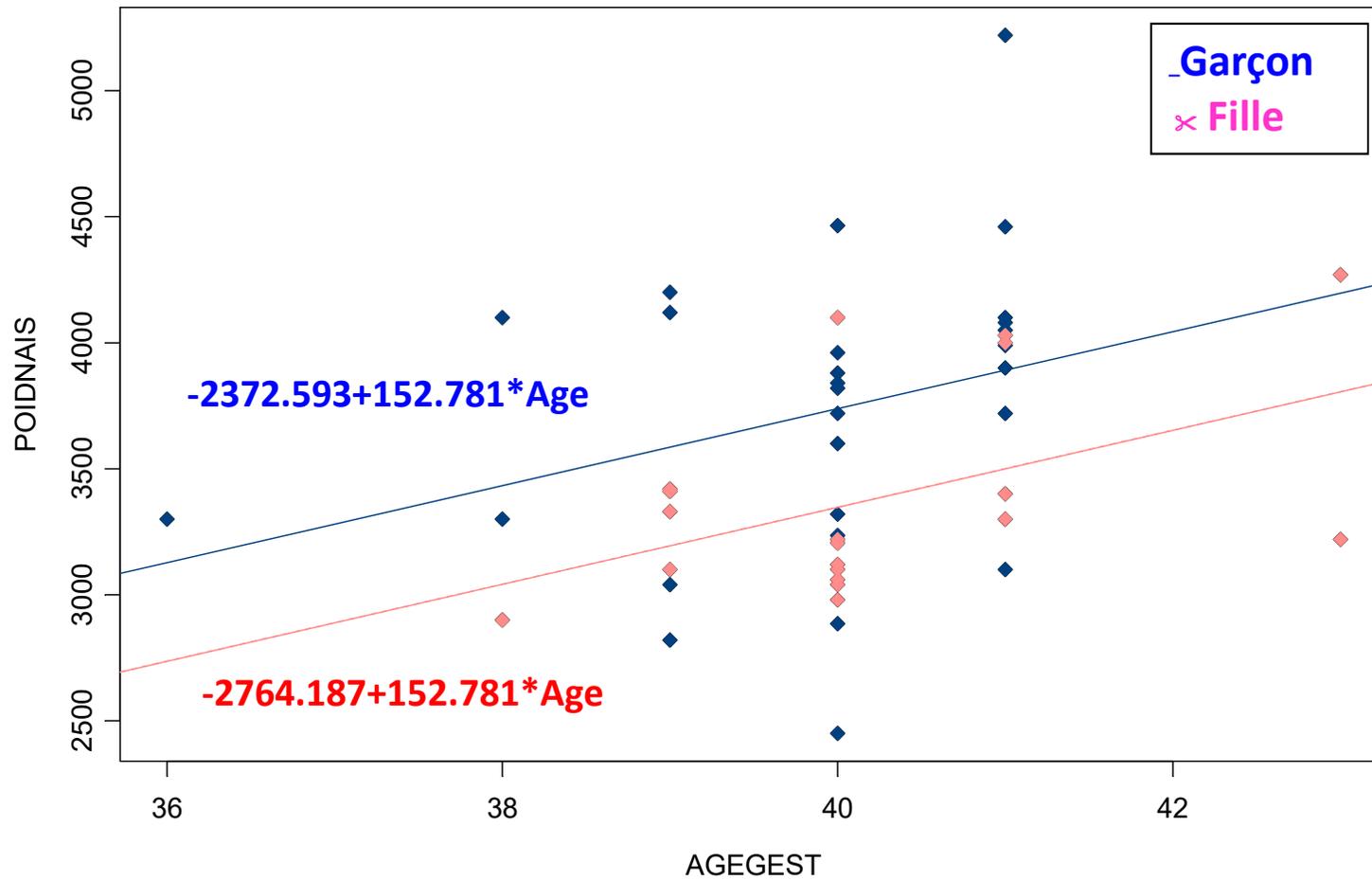
# Maximum de vraisemblance du modèle linéaire

Pour les modèles linéaires, les estimations du maximum de vraisemblance sont identiques à celles moindres carrés. Il est donc possible d'utiliser l'une ou l'autre des méthodes.

Lorsque le modèle n'est pas linéaire, il n'est plus possible d'utiliser la méthode des moindres carrés. Les valeurs des paramètres du modèle sont estimées en maximisant la vraisemblance.

# 4. PART DE VARIANCE EXPLIQUEE

# Variance totale et Variance expliquée



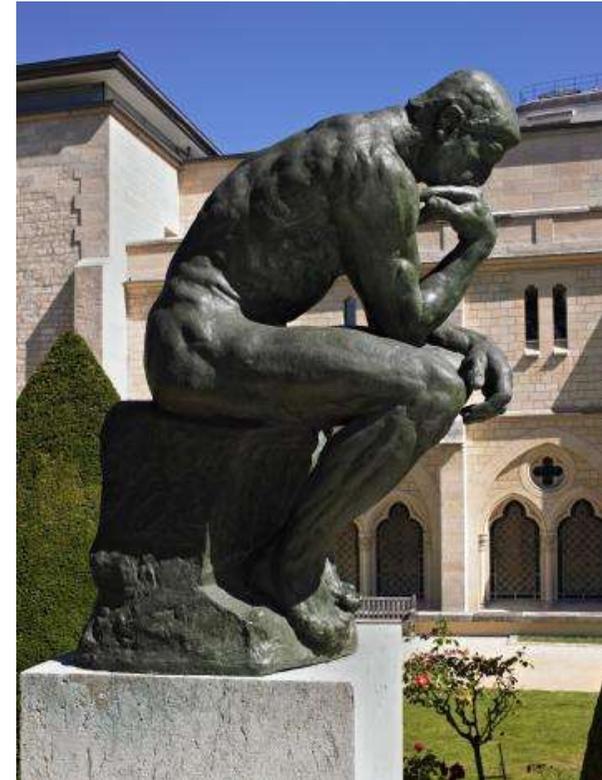
# 5. INTELLIGENCE ARTIFICIELLE

## Intelligence

faculté d'adaptation à l'environnement  
→ apprentissage

## Intelligence Artificielle

Résoudre des problématiques simples  
ou complexes à l'aide de modèles usuels  
et de réseaux de neurones

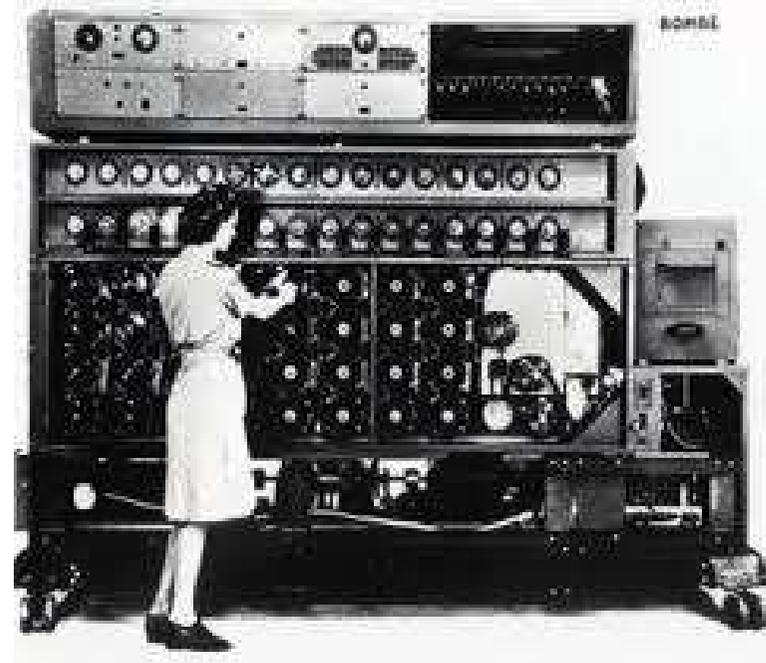


Musée Rodin, Jean de Calan ®

# Intelligence Artificielle

## Travaux de cryptanalyse

→ enigma     Jerzy Rozycki  
                  Marian Rejewski  
                  Alan Turing



Bombe Electromécanique de Turing (CNRS)



**Deep Blue IBM® (IBM.COM)**  
(Garry Kasparov en 1997)

## Test de Turing

Performances des calculateurs

## 6. Intelligence Artificielle

L'information vient des données !

Analyse de données de grande dimension (BigData)

Puissance de calcul disponible

Quelle est l'Information réellement contenue dans les données

# 6. DEEP LEARNING

# 1. Apprentissage supervisé

(les modèles de régression le font)

## Entrainer les machines au lieu de les programmer

Reconnaître des images (objets, visages, voitures, panneaux routiers,...)

Exemple : Présenter une image

bonne réponse → faible modification des paramètres

sinon → ajustement important des paramètres

La fois suivante → image reconnue

**Applications** → reconnaissance d'images, conduite automatique, traduction, reconnaissance automatique de la parole, diagnostic des Tumeurs, mélanomes en imagerie médicale

**Nécessité** → milliers (dizaines de milliers) d'images

**Propriété de généralisation** → reconnaître des images jamais vues

## 2. Apprentissage profond (deep learning)

Réseaux de neurones multicouches

Système est entraîné sur des milliers (dizaines de milliers) d'exemples

Ajustement continu des coefficients

Convergence vers la combinaison qui minimise la différence entre la sortie attendue et celle produite par le système moyennée sur des milliers d'exemples

## 3. Apprentissage par renforcement

Bien pour les jeux (échecs)

Mauvaise réponse → punition

Bonne réponse → récompense → renforcement de la stratégie

Performances >> performances humaines

# 7. RESEAUX NEURONAUX

# 1. Réseau

Ensemble de neurones et de connexions entre neurones.

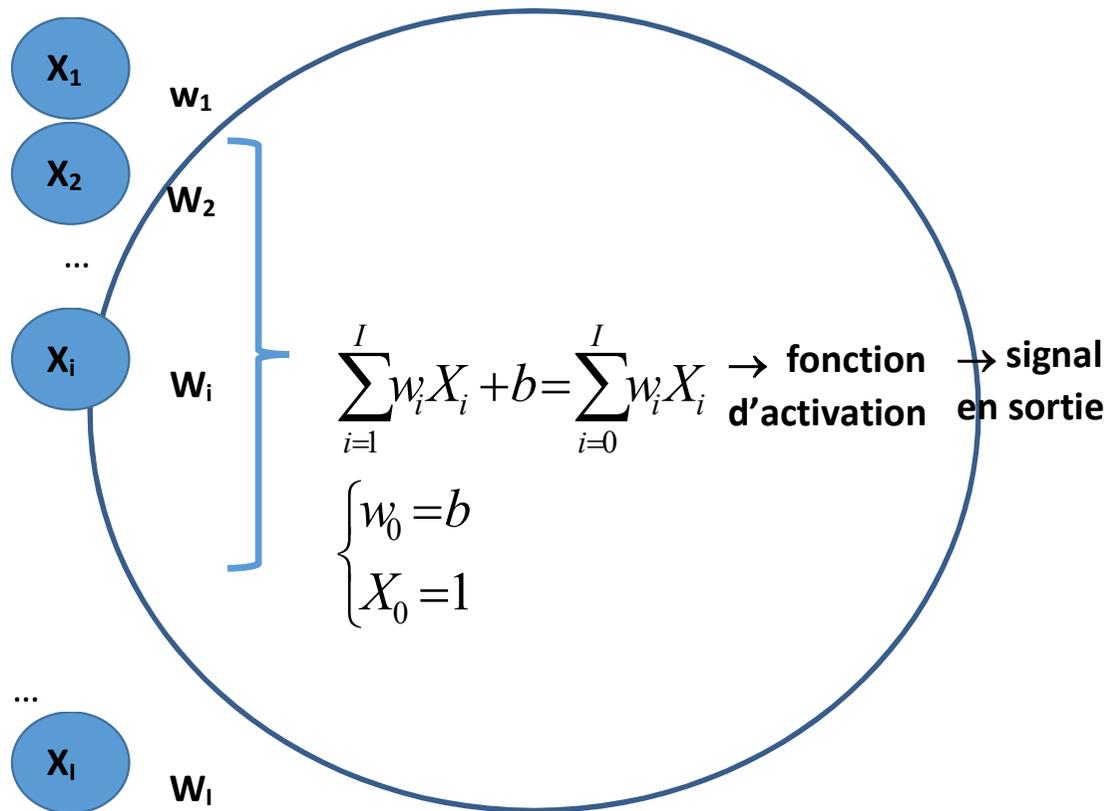
Les neurones reçoivent des signaux en entrée, les traitent et fournissent des signaux en sortie.

Les neurones sont des unités de calcul. Les traitements effectués peuvent être simples (e.g. sommation des signaux entrants) ou beaucoup plus complexes.

Les connexions caractérisent les flux entre neurones.

## 2. Perceptron monocouche

Schéma d'un ensemble de neurones recueillant  $I$  signaux en entrée, directement connectés à 1 ou plusieurs neurone(s) de sortie chacun avec sa fonction d'activation.  $I$  est la dimension du problème.



Pas de couche cachée,  
( $I$  Poids + 1 paramètre de biais + 1 fonction d'activation) par neurone de sortie.



## 2. Perceptron monocouche (exemple illustratif)

cas simple : 2 neurones en entrée

Fonction d'activation = fonction marche (Heaviside)

→ le modèle de classification conduit à séparer le plan d'axes  $X_1$  et  $X_2$  en 2 demi-plans séparés par une droite dont l'équation découle de l'annulation de  $z$

$$z = 0$$

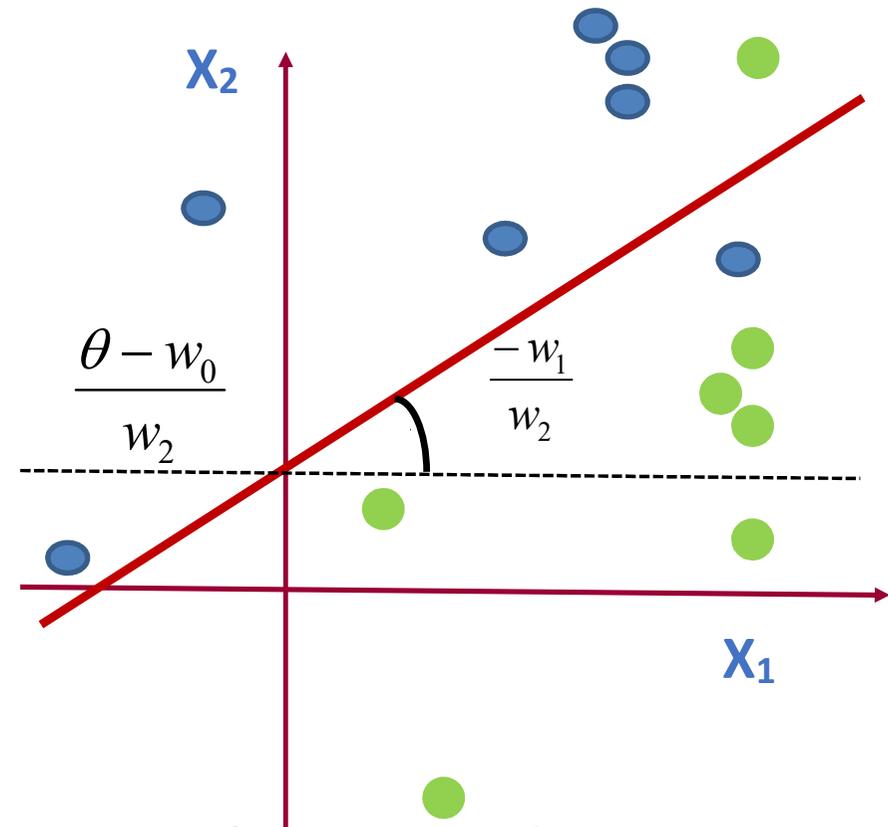
$$w_0 + w_1 X_1 + w_2 X_2 - \theta = 0$$

$$X_2 = \frac{1}{w_2} (\theta - w_0 - w_1 X_1)$$

Fonction marche :

3 neurones en entrée

>3 neurones en entrée



→ séparateur plan

→ séparateur hyperplan

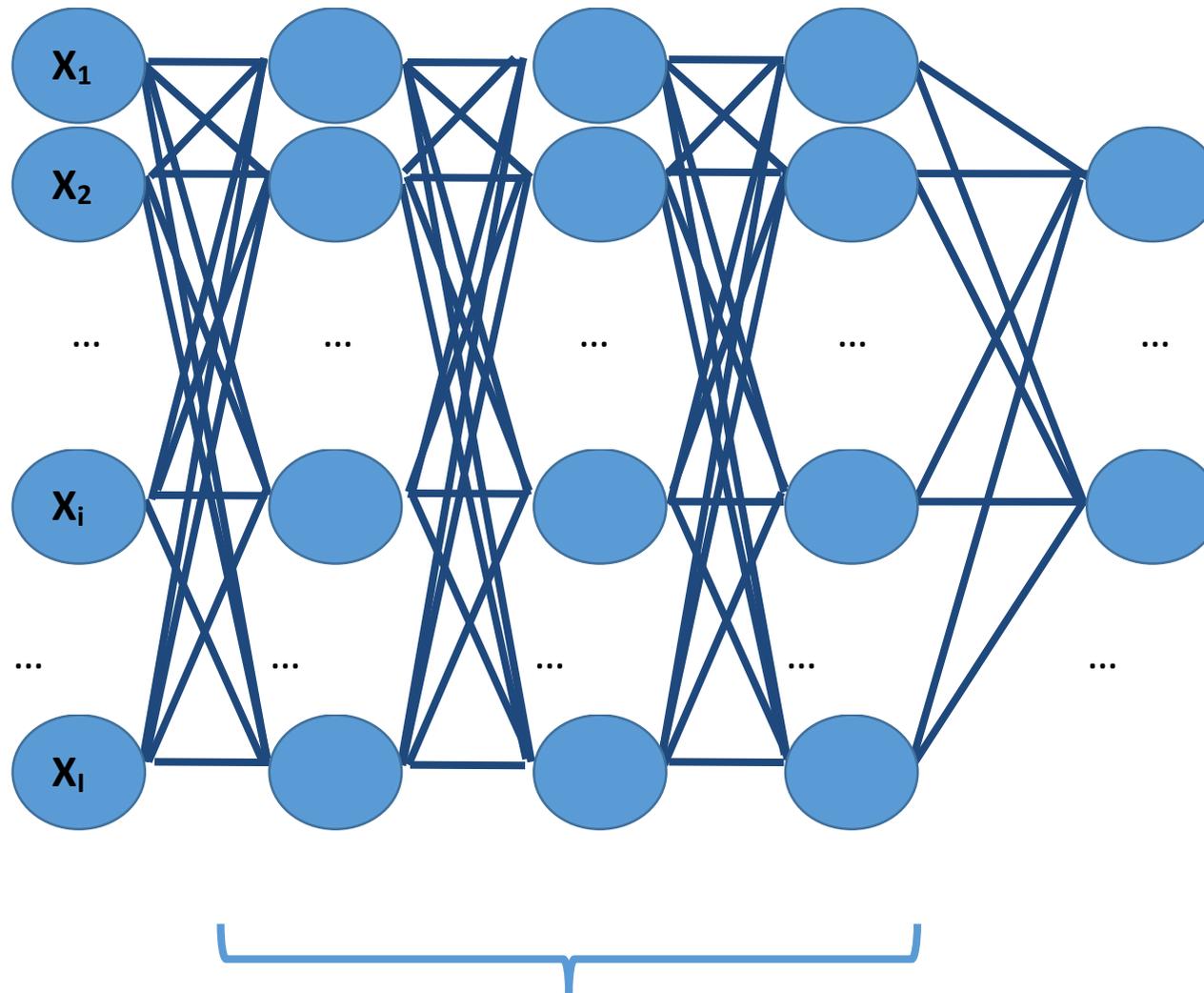
# 3. Réseau neuronal multicouche

Une couche d'entrée comprenant autant de neurones que de signaux à capter

Une ou plusieurs couches cachées

Une couche de sortie comprenant autant de neurones que de modalités d'intérêt

### 3. Réseau neuronal multicouche



Couche  
d'entrée

Couches  
cachées

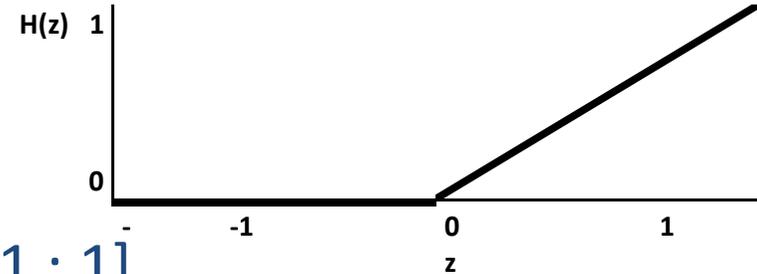
Couche  
de Sortie

# Fonctions d'activation (→ ensemble image)

fonction ReLU (Rectifier Linear Unit)  $\mathbb{R} \rightarrow [0 ; \infty[$

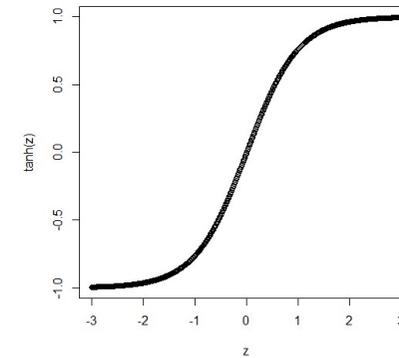
$$z = \sum w_i X_i - \theta$$

$$H(z) = \max(0, z)$$



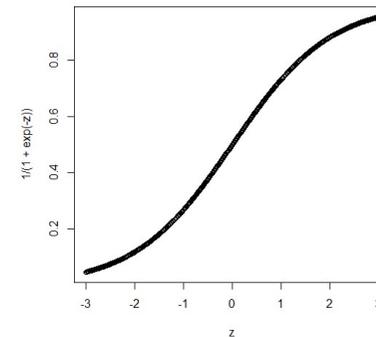
Fonction Tangente hyperbolique  $\mathbb{R} \rightarrow [-1 ; 1]$

$$H(z) = \tanh(z)$$



Fonction sigmoïde (logistique)  $\mathbb{R} \rightarrow [0 ; 1]$

$$H(z) = \frac{1}{1 + \exp(-z)}$$



Fonction softmax (extension de logistique)

Intérêt : fonction continuellement

dérivable → entraînement des réseaux

# Exemple illustratif d'un perceptron monocouche particulièrement simple (interprétable)

$$X_2 = \beta_0 + \beta_1 X_1$$

$$\beta_0 = \frac{-w_0}{w_2}$$

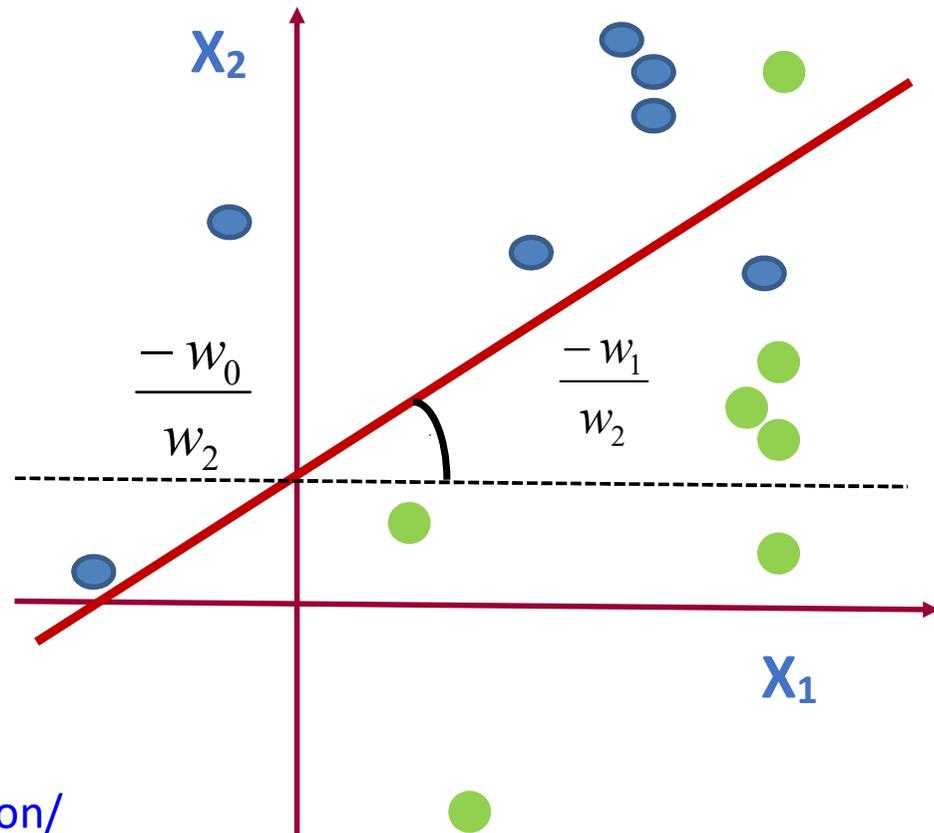
$$\beta_1 = \frac{-w_1}{w_2}$$

$$X_2 = .25 + 0.5 X_1$$

$$\beta_0 = \frac{-w_0}{w_2} = \frac{-(-0.5)}{(2)}$$

$$\beta_1 = \frac{-w_1}{w_2} = \frac{-(-1)}{(2)}$$

→ <https://lucleray.github.io/perceptron/>



## Perceptron monocouche (exemple illustratif)

$$X_2 = \beta_0 + \beta_1 X_1$$

$$\beta_0 = \frac{-w_0}{w_2}$$

$$\beta_1 = \frac{-w_1}{w_2}$$

$$X_2 = .25 + 0.5 X_1$$

$$\beta_0 = \frac{-w_0}{w_2} = \frac{-(-0.5)}{(2)}$$

$$\beta_1 = \frac{-w_1}{w_2} = \frac{-(-1)}{(2)}$$

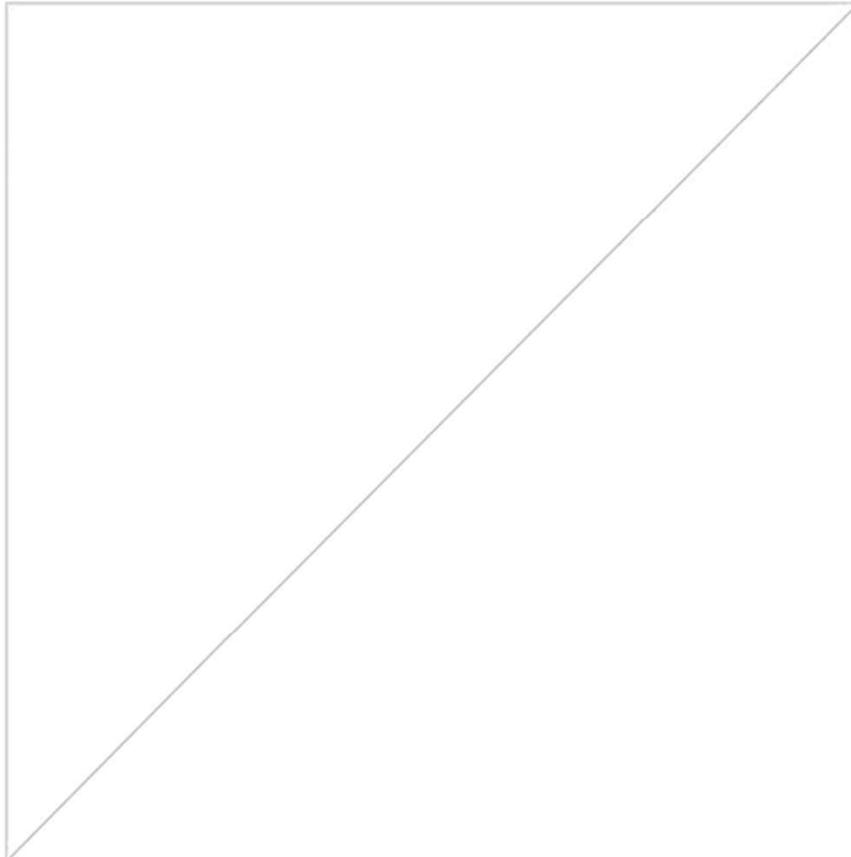
→ <https://lucleray.github.io/perceptron/>

Fonction d'activation sigmoïde – Algorithme de descente du gradient  
( $\alpha$  : Coefficient d'apprentissage = learning rate / epoch : nombre de cycle)

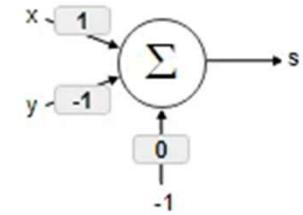
# Perceptron

RAZ

? Aide : survolez pour avoir plus d'informations



x : 1                      y : 0.3725  
[ajouter](#) [deplacer](#)                      [rouge](#) [vert](#)  
[supprimer](#)



apprentissage : simple / total

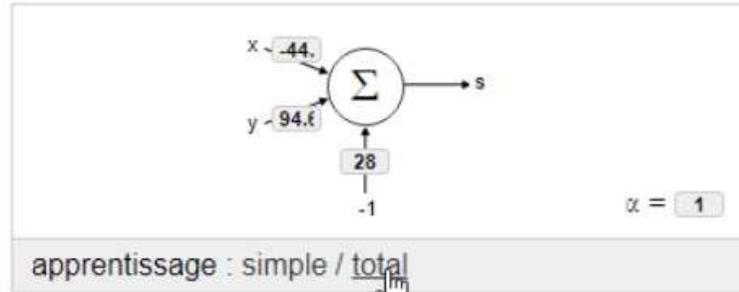
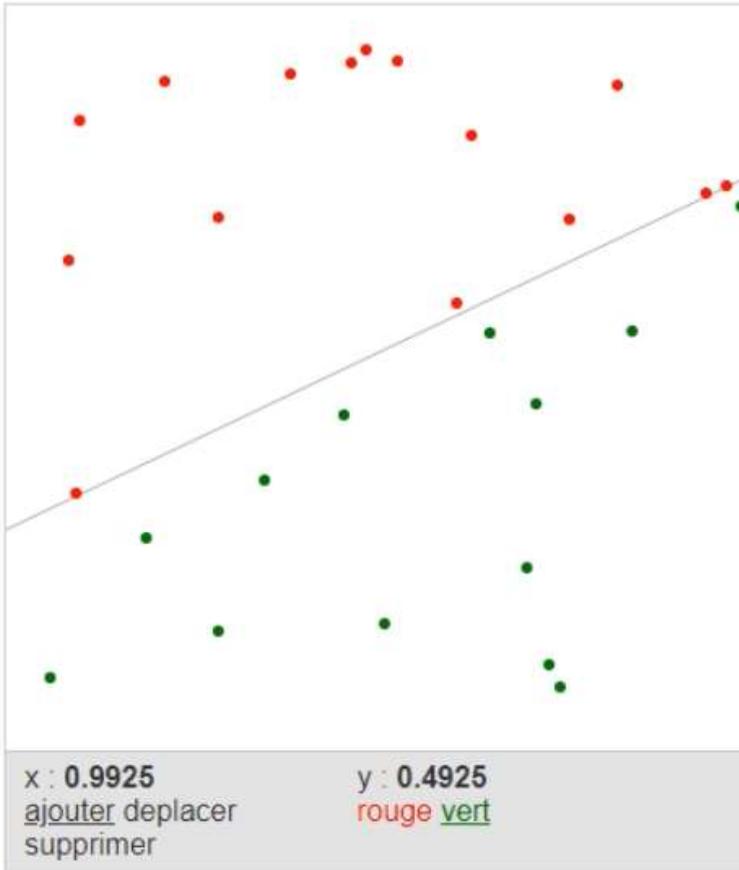
id	x	y
----	---	---

<https://lucleray.github.io/perceptron/>

# Perceptron

RAZ

? Effectuez l'apprentissage jusqu'à ce que tout les points soient correctement classés



id	x	y
0	0.2125	0.9
1	0.0825	0.66
2	0.0975	0.8475
3	0.485	0.9425
4	0.3825	0.91
5	0.0925	0.3475
6	0.6075	0.6025
7	0.285	0.7175
8	0.825	0.895
9	0.76	0.715
10	0.6275	0.8275
11	0.465	0.925
12	0.5275	0.9275
13	0.9725	0.76
14	0.345	0.75
15	0.0575	0.4

<https://lucleray.github.io/perceptron/>

$$X_2 = \frac{-(-28)}{94.6} + \frac{-(-44)}{94.6} X_1 \approx 0.30 + 0.47 X_1$$

## 4. Architectures

Réseau neuronal à propagation avant (feed-forward networks)  
(vus précédemment)

Réseaux récurrents (boucles d'activation)

## 5. Apprentissage supervisé

L'ajustement des poids se fait par apprentissage.  
Valeurs initiales aléatoires, mise à jour en continue.

L'échantillon d'apprentissage intègre évidemment la variable réponse.

Méthodes de rétropropagation du gradient ou assimilées (de la dernière couche → la première) pour les réseaux à propagation avant

Méthodes de rétropropagation à travers le temps pour les réseaux récurrents

→ maximiser la vraisemblance

## 6. Performances d'un réseau de neurones utilisé pour la classification (sur échantillon test)

Exactitude  $(VP + VN) / (VP+VN+FP+FN) = \text{Exactitude}$

Rappel  $VP / (VP + FN) = \text{sensibilité}$

Précision  $VP / (VP + FP) = \text{Valeur Prédicative Positive}$

Spécificité  $VN / (VN + FP) = \text{spécificité}$

AUC Aire sous la courbe ROC

# **7. BIG DATA**

## **Données de Grandes Dimensions**

## Les données – La data

Grand nombre d'observations (n)

Entrepôts de données

Assurance maladie

Grand nombre de variables (m)

Biologie moléculaire (omics)

Imagerie médicale (radiomique)

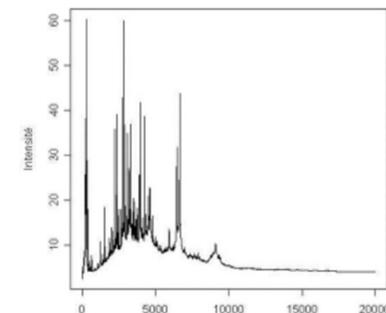
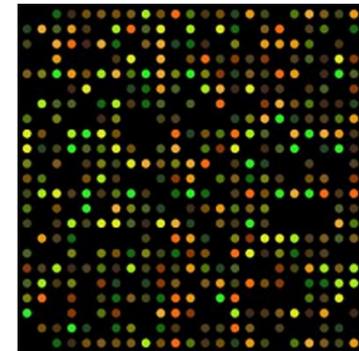
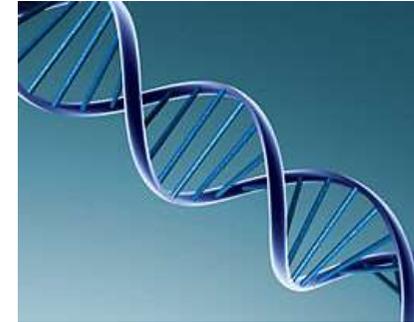
Grand nombre d'observations et de variables (n et m)

# Omics ( $m \gg n$ )

$3 \times 10^9$  bases d'ADN (Génome entier)  
GWAS  $0.5-2 \times 10^6$  SNP

22 000 gènes (puces à ADN)

Quelques centaines de protéines



# ESTIMATION DES PARAMETRES ( $\beta_j$ ) DES MODELES

Linéaire

$$Y = \beta_0 + \sum \beta_j X_j$$

en univarié  $Y = \beta_0 + \beta_1 X_1$

## 1. Diagnostique (ex logistique)

$$P(Y = 1|X) = \frac{e^{\beta_0 + \sum \beta_j X_j}}{1 + e^{\beta_0 + \sum \beta_j X_j}} = \frac{1}{1 + e^{-(\beta_0 + \sum \beta_j X_j)}}$$

## 2. Pronostique (ex modèle de survie de Cox)

$$\lambda(t, X) = \lambda(t, \mathbf{0}) \exp\left(\sum \beta_j X_j\right)$$

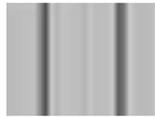
## 3. Théranostique (avec interation)

*... plus tard dans le cursus ...*

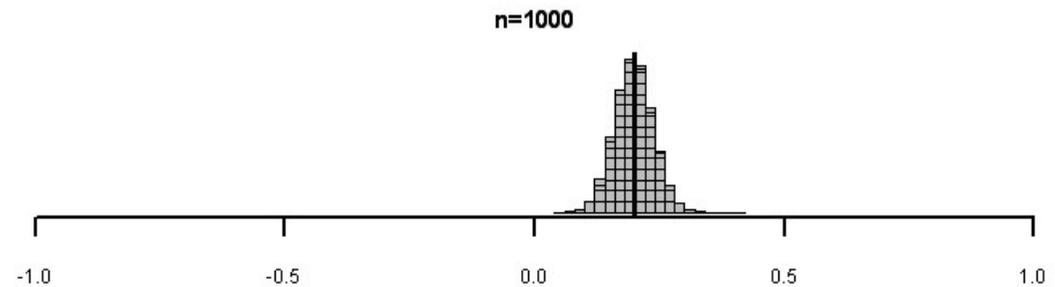
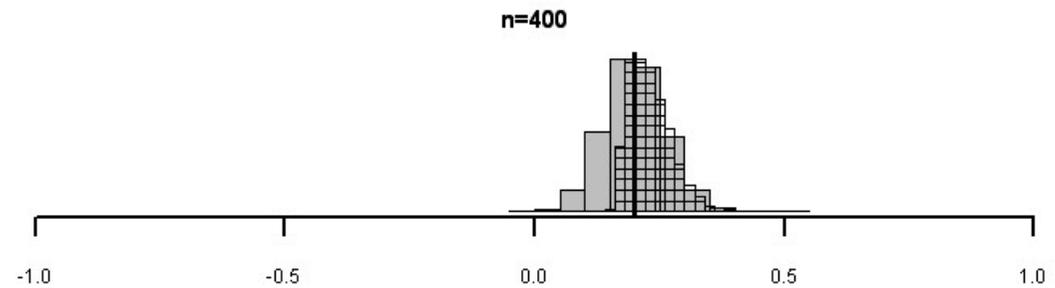
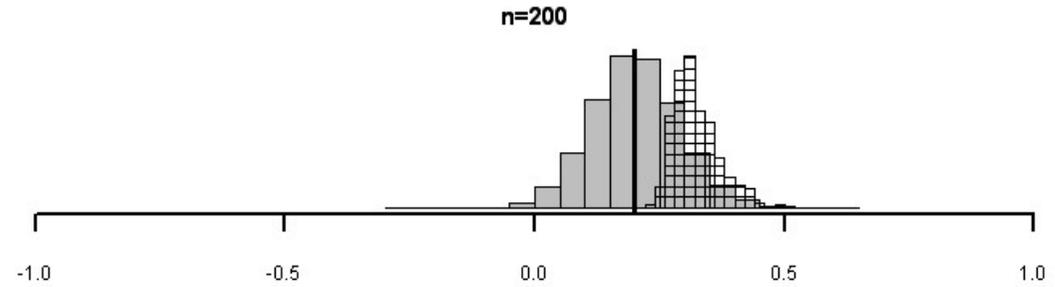
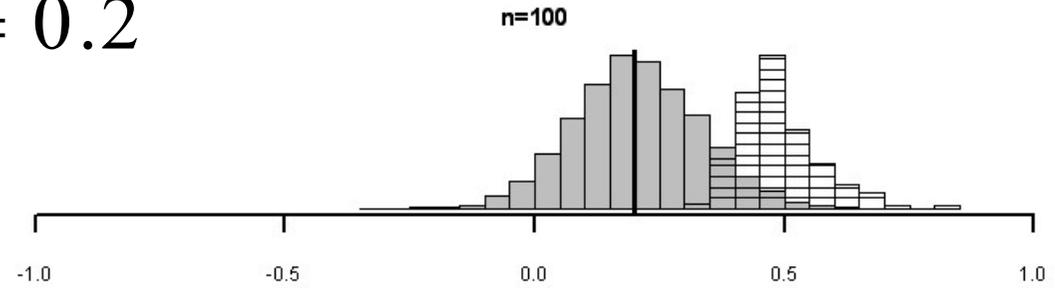
# Biais d'optimisme des Etudes d'identification

$$\beta = 0.2$$

$\Omega_{m1}$



$\Omega_S$



# Méthodes Pénalisées

Méthodes LASSO, RIDGE, etc ...

# CONCLUSION

Russel S, Norvig P. Artificial intelligence. A Modern Approach. Third Edition. *Pearson Education* 2016. Harlow Essex England.

Hastie T, Tibshirani R, Friedman J. The Elements of Statistical Learning. Data Mining, Inference and Prediction. Second Edition. 2008. Springer-Verlag.

Mitra S, Acharya T. Data mining? Multimedia, soft computing and bioinformatics. Wiley & Sons, Inc., Hoboken, New Jersey. 2003

Truntzer C, Maucourt-Boulch D, Roy, P. Comparative optimism in models involving both classical clinical and gene expression information. *BMC bioinformatics* 2008.

Roy P, Truntzer C, Maucourt-Boulch D, Jouve T, Molinari N. Protein mass spectra data analysis for clinical biomarker discovery: a global review, *Briefings in Bioinformatics* 2011.

Truntzer C, Maucourt-Boulch D, and Roy P. Impact of the selection mechanism in the identification and validation of new "omic" biomarkers. *Journal of Proteomics & Bioinformatics* 2013.

Tibshirani R. Regression shrinkage and selection via the lasso. *J. Roy. Statist. Soc. Ser. B* 1996.

Zhao Y, Dantony E, Roy P. Optimism Bias Correction in Omics Studies with Big Data: Assessment of Penalized Methods on Simulated Data. *Omics* 2019;23:297-213.

**Merci !**