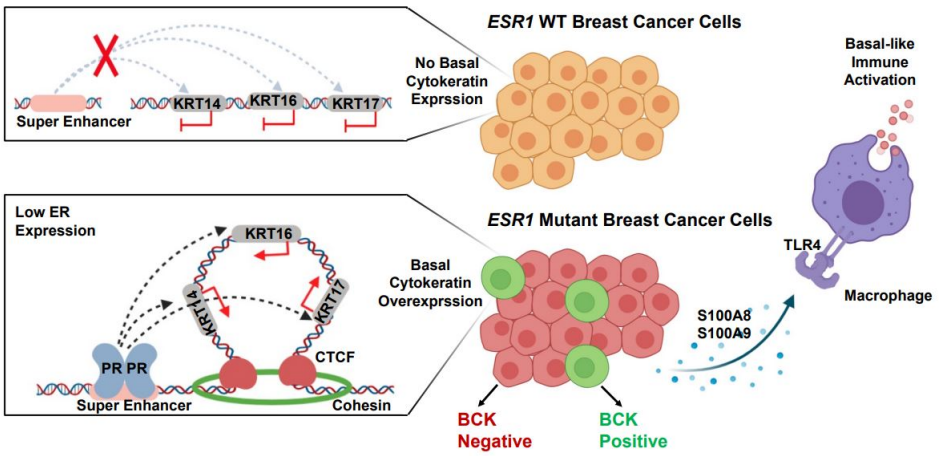
Cancer du sein, expression des gènes ESR1 et S100A8
- L2 Math, IMIA -

L'hormonorésistance constitue l'un des défis majeurs dans le traitement du cancer du sein avancé exprimant le récepteur aux estrogènes (RE) et sans surexpression de HER2. Des mutations dans le gène ESR1, qui altèrent la région où le ligand se lie (c'est-à-dire la partie du récepteur aux œstrogènes où les hormones ou autres molécules se fixent), ont été découvertes récemment comme étant l'un des principaux moyens par lesquels certaines cellules cancéreuses résistent aux inhibiteurs de l'aromatase (IA). Les IA sont des médicaments utilisés dans le traitement du cancer du sein hormono-dépendant en bloquant la production d'œstrogènes. Ces mutations induisent une activation constitutionnelle du RE conduisant à une résistance acquise aux IA.
ESR1 est actif dans $\pm$ 75 % des tumeurs du cancer du sein.

Les protéines de la famille S100 sont souvent dérégulées dans le cancer. La protéine S100A8, mésurée dans le sang des patients, a été identifié comme étant l'une deux protéines dont l'expression augmente le plus dans le cancer ESR1 mutant.



Source : Li, Z., McGinn, O., Wu, Y., Bahreini, A., Priedigkeit, N. M., Ding, K., ... , Oesterreich, S. (2022). ESR1 mutant breast cancers show elevated basal cytokeratins and immune activation. Nature Communications, 13(1), 2011.

Le jeu de données est issu d'une étude impliquant 32 patientes atteintes d'un cancer du sein avec une tumeur à récepteurs d'øestrogènes positifs et ayant reçu une chimiothérapie au tamoxifène.
Les variables enregistrées sont :

- `grade` : grade histologique de la tumeur (grade 1 vs 3),
- `ganglion` : état des ganglions lymphatiques (0 : non affecté, 1 : ganglions lymphatiques affectés et enlevés),
- `taille` : taille de la tumeur en cm,

— Expression des gènes `ESR1` et `S100A8` dans les biopsies de tumeurs (technologie des puces à ADN).

Ici on cherche à savoir si l'expression du gène ESR1 est liée à celle de le protéine S100A8. L'intérêt clinique est que l'expression de la protéine S100A8 est plus facile à mesurer que l'expression du gène ESR1. La connaissance précise du type de cancer permet de mieux cibler les traitements.

1 Statistiques descriptives

1. Importer les données

```
import pandas as pd
data = pd.read_csv("brca.csv",sep=",")
print(data) # pour voir ce que contient la table de donnees
ESR1 = data["ESR1"]
S100A8 = data["S100A8"]
```

2. Tracer un nuage de points de l'expression de la protéine S100A8 en fonction de l'expression du gène ESR1. Voyez vous un lien entre les deux quantités ?
3. Le lien linéaire n'est pas évident. On propose d'appliquer une transformation en log aux deux expressions. Tracer le nuage de points pour les quantités transformées. Que peut-on dire de la relation entre ces deux variables ?
4. Quel est l'intérêt selon vous de tracer ces expressions de gènes en échelle log-log avec la fonction `loglog` du module `matplotlib` ? Tracer la figure avant de répondre.

2 Estimation de la droite de régression

2.1 Régression linéaire simple

5. Utiliser les fonction développées au TP précédent pour estimer les paramètres de la droite de régression entre $\log(ESR1)$ et $\log(S100A8)$.
6. Tracer le nuage de points et superposer la droite de régression.
   Commenter le graphique obtenu.

2.2 Estimation par la méthode de la descente du gradient

La solution obtenue ci-dessus est bonne mais l'algorithme utilisé n'est pas général dans le sens où il ne peut être utilisé que pour la régression linéaire simple.
On propose alors d'implémenter l'algorithme de la descente de gradient pour approcher l'estimateur des moindres carrés.

2.3 Exercice préparatoire en dimension 1

On choisit la fonction de perte de l'erreur aux moindres carrés

$$L(w) = \frac{1}{2}\sum_{i=1}^{n}(y_i - w_0 - w_1 x_i)^2$$

2.4 Implémentation du gradient

7. Calculer le gradient de L
8. Implémenter une fonction qui calcule le gradient de L pour la régression linéaire.

```
def grad_reglin(w,X,y) :
# Gradient du coût aux moindres carrés pour la régression linéaire
# Entrées
# - w : vecteur des poids (tableau numpy (d+1)x1)
# - x : prédicteurs (tableau numpy nxd)
# - y : matrice des réponses (tableau numpy nx1)
#Sortie :
# - grad : gradient en w
```

Si vous avez traité les questions 11 et 12 du TD, vous pouvez utiliser les expressions matricielles (voir questions (a) et (b) ci dessous, optionnel).

(a) Pour préparer les données, pour la formulation matricielle, suivre les indications ci-dessous.
Créer un tableau numpy `X`, tel que

$$X = \begin{bmatrix} 1 & x_{11} \\ \vdots & \vdots \\ 1 & x_{n1} \end{bmatrix}$$

Vous pourrez utiliser la fonction repeat du module numpy pour créer un vecteur de 1 et concatenate pour concaténer les deux vecteurs.

(b) On rappelle que (voir TD), si L est la fonction du coût aux moindres carrés,

$$\nabla L(w) = -\mathbf{X}^T(\mathbf{y} - \mathbf{X}\mathbf{w})$$

Attention, `w` et `y` doivent être des tableaux numpy. La fonction dot du module numpy permet de multiplier des vecteurs et/ou des matrices entre eux. La fonction transpose permet de transposer.

9. En choisissant, $w = \begin{pmatrix} 8 \\ -1 \end{pmatrix}$, vérifier que la fonction renvoie $gL = \begin{pmatrix} -118.4 \\ -857.0 \end{pmatrix}$

2.5 Algorithme de descente du gradient

10. Implémenter l'algorithme de descente du gradient

```
def grad_desc(w0,grad,x,y,alpha=.001,tol=1e-3,maxiter=1e4) :
# Algorithme de descente du gradient
# Entrées
# - w0 : point initial (tableau numpy (d+1)x1)
# - grad : gradient de la fonction de coût des moindres carrés
# - x : prédicteurs (tableau numpy nxd)
# - y : matrice des réponses (tableau numpy nx1)
# - alpha : taux d'apprentissage (defaut 0.1)
# - tol :  tolerance (defaut 1e-3)
# - maxiter :  nombre maximum d'itérations (defaut 1e4)
# Sortie :liste contenant
# - sol : solution
# - grad_w: valeur du gradient de la fonction de perte à la solution
# - iter : nombre d'itérations effectuées
# - cvge : booleen (True si l'algorithme a convergé)
```

11. Appliquer l'algorithme de descente du gradient avec les paramètres par défaut, aux données du cancer du sein pour estimer la droite de régression entre $\log(S100A8)$ et $\log(ESR1)$.
    On choisira comme point de départ $w = \begin{pmatrix} 8 \\ -1 \end{pmatrix}$.
    Quel résultat obtient-on ? L'algorithme a t'il convergé ?
12. Essayer d'autres valeurs du taux d'apprentissage.Vous pouvez aussi faire varier la tolérance et le nombre d'iterations maximum.
    Quel résultat obtient-on ? Pour quelle valeur du taux d'apprentissage ? En combien d'itérations ?

2.6 Algorithme de descente du gradient avec back-tracking

13. Implémenter l'algorithme du gradient avec back-tracking, dans une fonction dont l'entête sera la suivante

```
def grad_desc_bt(start,L,grad,x,y,alpha=.1,a=.5,b=.5,tol=1e-3,maxiter=1e4) :
# Algorithme de descente du gradient
# Entrées
# - start : point initial (tableau numpy (d+1)x1)
# - L : fonction de perte
# - grad : gradient de la fonction de perte
# - x : prédicteurs (tableau numpy nxd)
# - y : matrice des réponses (tableau numpy nx1)
# - alpha : taux d'apprentissage (defaut 0.1)
# - a, b : paramètres du back-tracking
# - tol :  tolerance (defaut 1e-4)
```

```
# - maxiter :  nombre maximum d'itérations (defaut 10)
# Sortie :liste contenant
# - sol : solution
# - grad_w: valeur du gradient de la fonction de perte à la solution
# - iter : nombre d'itérations effectuées
# - cvge : booleen (True si l'algorithme a convergé)
```

14. Appliquer l'algorithme de descente du gradient avec back-tracking avec les paramètres par défaut, aux données du cancer du sein pour estimer la droite de régression entre $\log(S100A8)$ et $\log(ESR1)$. On choisira comme point de départ $w = \begin{pmatrix} 8 \\ -1 \end{pmatrix}$.
    Quel résultat obtient-on ? L'algorithme a t'il convergé ? En combien d'itérations ?

3 Conclusion

L'expression du gène S100A8 est-elle liée à celle de la protéine ESR1 ? Quel est le lien (écrire l'équation) ?