Rapport du Projet d'Analyse de Données et de Statistiques

Rémi Colin, Talel Taieb, Karima Ghamnia, Evan Rabineau

Contents

1 Introduction

Ce projet vise à étudier un jeu de données pour G = 1615 gènes d'une plante dont le modèle est :

$$Y_{gtsr} = log_2(X_{gtsr} + 1) - log_2(X_{gt_0} + 1)$$

où :

- X_{gtsr} est la mesure d'expression du gène $g \in G1, ..., G1615$ npour le traitement t $\in T1, T2, T3$ pour le réplicat $r \in R1, R2$ et au temps $s \in 1h, 2h, 3h, 4h, 5h, 6h$
- X_{gt_0} est l'expression du gène g pour un traitement de référence t_0

Toutes les sorties R présentes dans ce rapport que nous avons pu réaliser en Python sont présentes dans le Rmarkdown.

Cependant, nous avons rencontré des difficultés à superposer chaque figure exactement avec l'interprétation adéquate.

2 Statistiques Descriptives

2.1 Visualisation de la DataBase

Table 1: Data summary

Name	Data
Number of rows	1615
Number of columns	36
Column type frequency:	
numeric	36
Group variables	None

On a 1615 observations et 36 variables. Chaque réponse est liée à un traitement utilisé sur une réplicat pendant une certaine durée. On a 3 traitements (T1,T2,T3) effectués sur 2 réplicats (R1,R2) et on réalise des observations à chaque heure passée dans une durée de 6 heures.

2.2 Analyse des données

Dans cette partie, on va réaliser une analyse sur les réponses des gènes.

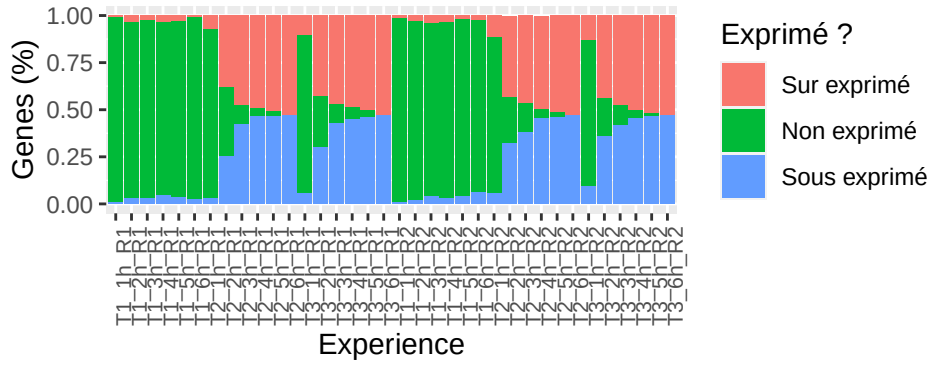


Figure 1: Barplot de pourcentage des gènes sur/non/sous exprimés pour chaque variable

Ce graphique montre le pourcentage des gènes non/sous/sur exprimés pour chaque variable. On remarque que la majorité des gènes pour le traitement T1 sont non exprimés. Tandis que, pour T2 et T3, ils sont soit sur exprimés soit sous exprimés.

On fait une étude sur le comportement moyen des traitements sur les deux réplicats. On remarque qu'il y a une grande corrélation entre T2 et T3. On remarque également qu'il y a une symétrie entre R1 et R2. Pour les prochaines analyses, on va se concentrer sur un seul réplicat car il y a une forte corrélation entre R1 et R2 donc on n'aura pas plus d'informations si on utilise les deux réplicats dans nos analyses.

Analyse sur R1 :

On remarque une faible variance pour la réponse liée au traitement T1, même en centrant et en réduisant les données. L'effet de T1 est trop faible. Les réponses sous T2 et T3 varient énormément surtout dans les dernières heures. On remarque également que quelques réponses dépassent 1 ou sont bien plus faibles que

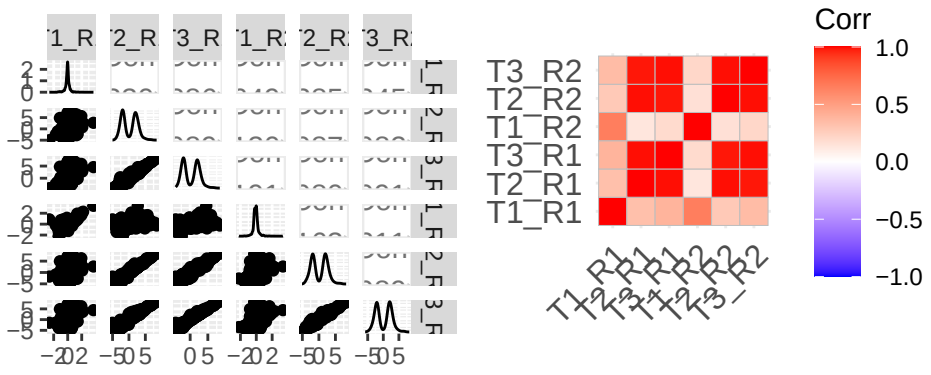


Figure 2: Etudes sur le comportement moyen des gènes pour T1, T2 et T3 pour chaque réplicat

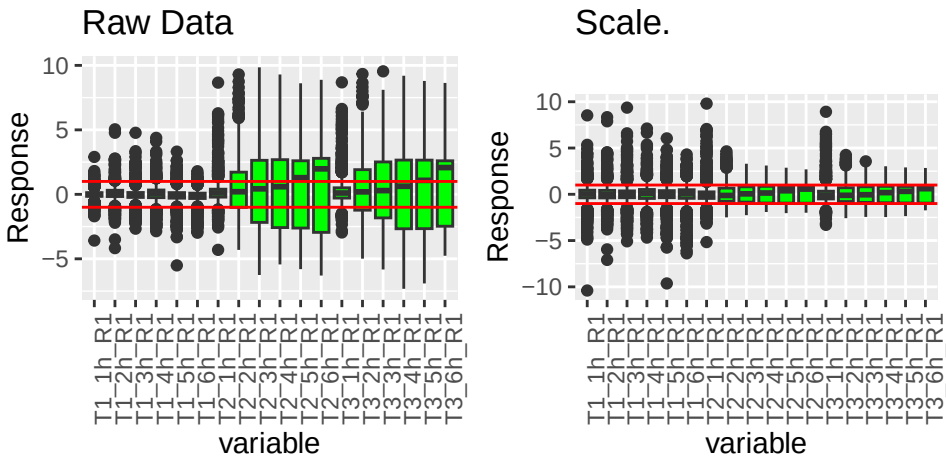


Figure 3: Boxplot sur les données brutes et données centrées réduit pour la réplicat R1

-1. A partir de 3h, la moyenne des réponses dépasse 1 ce qui traduit qu'il y a beaucoup de gènes qui sont sur exprimés.

Donc il y aura des données manquantes pour T2 et T3 surtout dans les dernières heures comme on peut le voir ci-dessous.

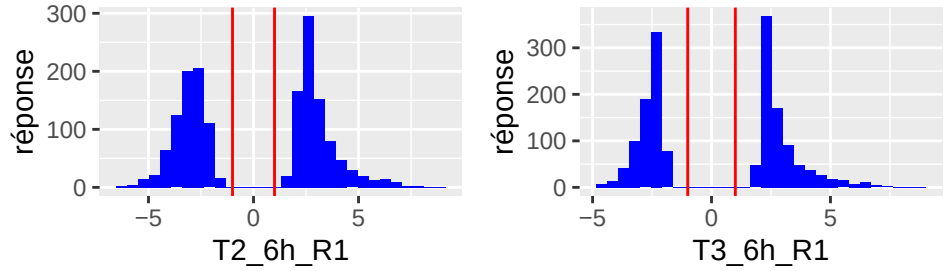


Figure 4: Repartition des réponses des gènes pour T2 et T3 a 6h pour R1

2.3 Analyse en Composantes Principale

Nous allons maintenant réaliser une Analyse en Composantes Principales (ACP) sur le jeu de donnée étudié afin de réduire son nombre de dimensions. De plus, au vu du boxplot des données brutes, on observe une variance très forte. Nous décidons donc de réduire et centrer les données au préalable.

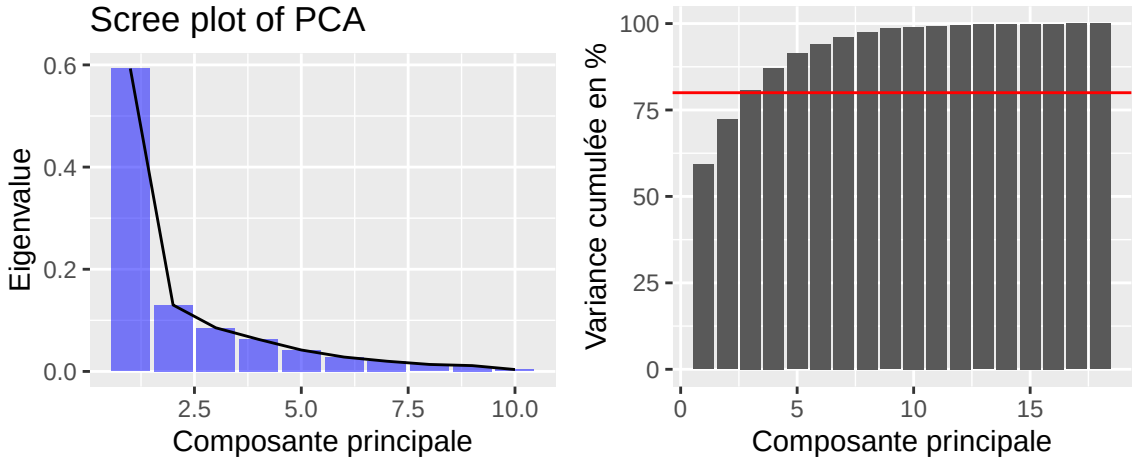


Figure 5: Inertie des composants en fonction des composantes principales

On remarque sur le graphique à droite que le pourcentage d'inertie expliquée est faible à partir de la 4-ème composante. Les quatre premières composantes contiennent plus de 95% de l'information d'après le graphique de gauche. On choisi donc de conserver uniquement les quatre premières composantes.

Sous le graphique ci-dessous est affiché la projection des coordonnées de l'ACP sur les deux premières dimensions.

On remarque qu'il y a un grand nombre de variables, ce qui peut rendre l'interprétation compliquée. Néanmoins, nous pouvons remarquer que la dimension 1 porte les variables liées au traitement T2 et T3, que la 2-ème dimension porte le traitement T1, que la 3-ème dimension porte sur les premières heures de chacun des traitements et que la 4-ème dimension porte sur l'évolution des différents traitements au cours du temps.

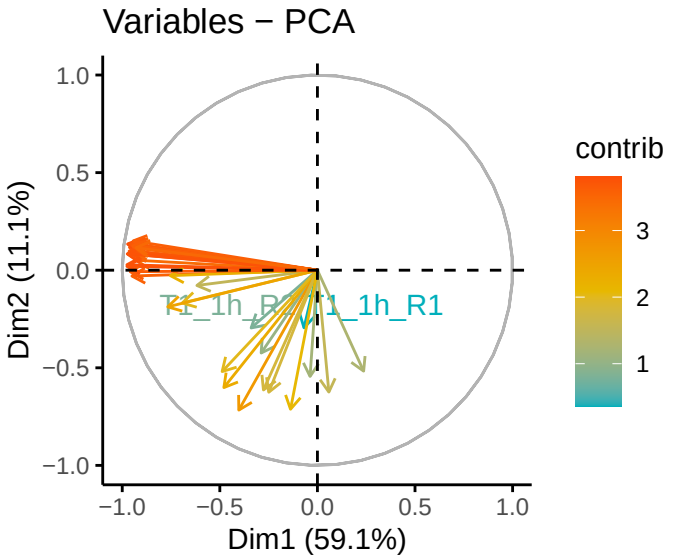


Figure 6: Cercle de corrélation ACP des données sur les 2 premières composantes

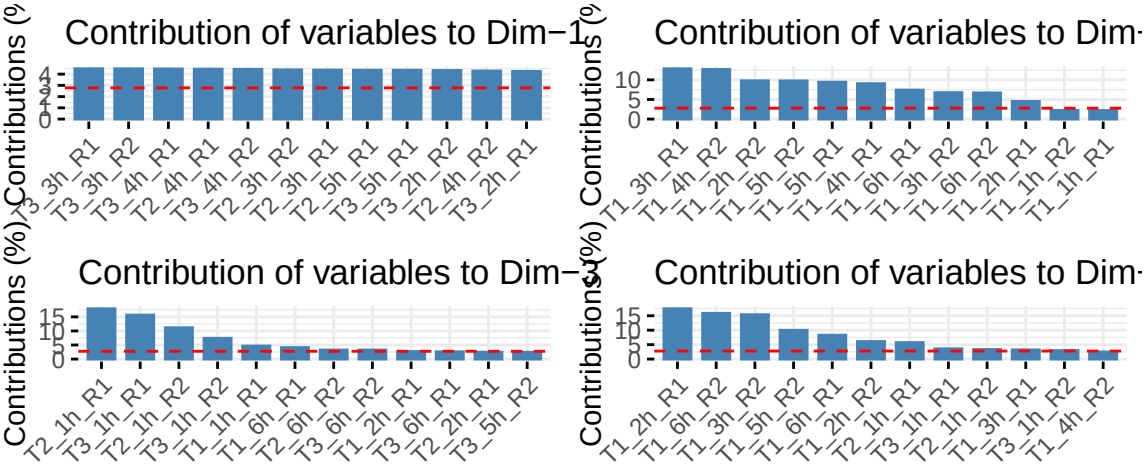


Figure 7: Participation des variables avec chaque dimension

Pour mieux appréhender les données, nous allons visualiser les variables qui contribuent le plus à chacune des quatre premières dimensions.

Ces résultats nous permettent d'affirmer que : - la dimension 1 porte essentiellement les traitements T2 et T3, - la dimension 2 porte essentiellement le traitement T1, - la dimension 3 porte les premières heures de chaque traitement, - la dimension 4 l'évolution des différents traitements au cours du temps.

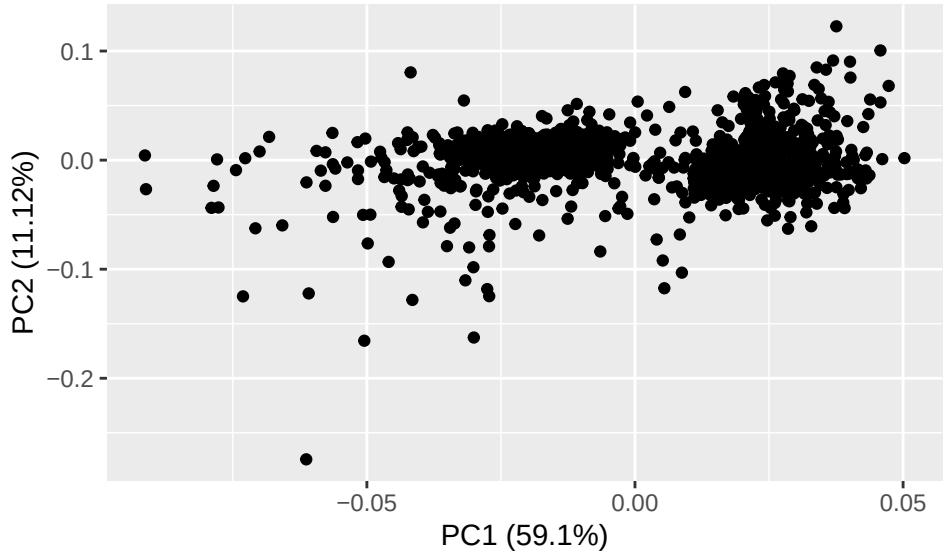


Figure 8: Acp sur les individus sur les 2 premières composantes

Ce nuage de points nous permet de nous donner une première idée sur des potentielles classes pour notre jeu de données. Notre hypothèse est que c'est deux classes présentes le groupes des gènes sous exprimés et les gène sur exprimés pour les traitements T2 et T3.

3 Classification

3.1 Classification des variables "Tx xh Rx"

Tout d'abord, nous transposons la matrice des données pour mettre les gènes en variables et les Tx_Hx_Rx en individus. Puis, nous réalisons une ACP pour observer les résultats projetés sur les deux premières dimensions de l'ACP.

On observe que les deux premières dimensions de l'ACP nous donnent 90 % de la variance expliquée. De plus, visuellement on observe 3 clusters :

- Un premier cluster contenant les traitements T1 et les traitements T2 et T3 mais sur les premières heures de l'experience. Au vu des analyses précédentes, on peut suggérer que ce cluster représente les temps et traitements pour lesquelle les gènes réagissent pas ou trés peu.
- Un deuxième cluster contenant les traitements T2 et T3 à des heures intermédiaires de l'expérience (de 2h à 4h).
- On peut observer un troisième clyster contenant les traitements T2 et T3 à des heures finales de l'expérience (de 4h à 6h).

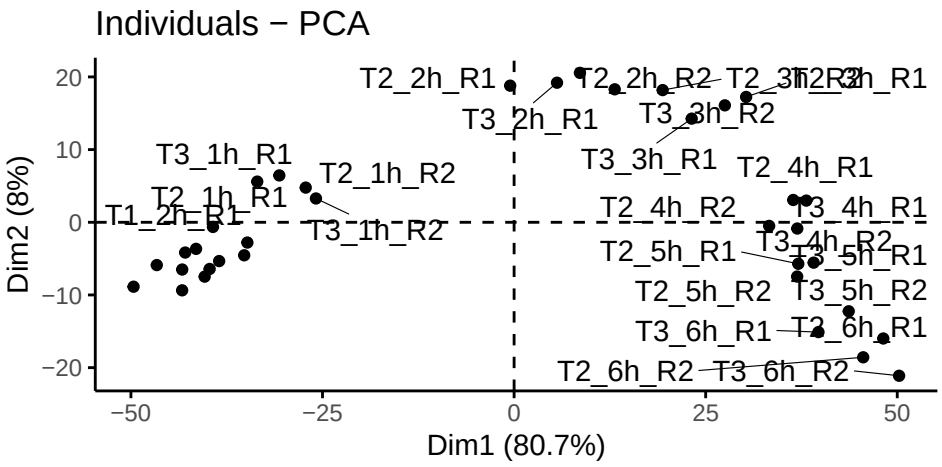


Figure 9: PCA sur les individus

Pour les deux derniers clusterings, on peut conjecturer que c'est l'efficacité du traitement qui les différencies.

Ce clustering n'est que visuel et n'utilise pas vraiment de méthode de clustering, il est juste là pour nous donner une idée de notre nouveau jeu de données. Maintenant, nous allons essayer de confirmer nos différentes hypothèses avec des méthodes de clustering.

3.1.1 Avec les Kmeans

Nous commençons avec un algorithme de Kmeans. Cet algorihtme est très sensible à l'initialisation qui est aléatoire. Pour réduire cet aléatoire, on choisit de répéter l'algorithme 15 fois (n_start=15). De plus, nous devons choisir le nombre de classes souhaité. Pour cela, nous allons utiliser deux critères de sélections : le critère de la variation de l'inertie Intra-Classe et le criteres Silhouette.

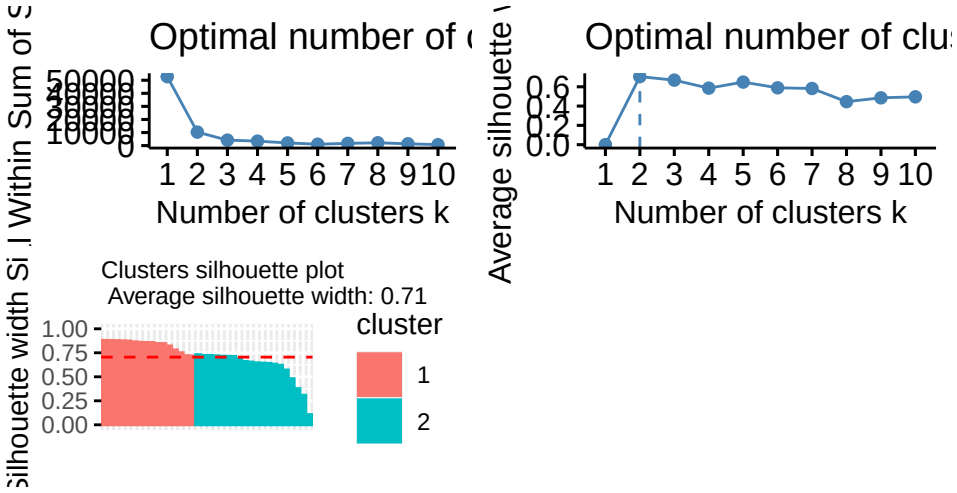


Figure 10: Des graphs pour faire le choix optimal de nombre de centre

On peut observer sur le premier graphique, l'évolution du total d'inertie intra-classe. Pour choisir le nombre de classes, on regarde l'endroit où la courbe forme un coude. Ici, on trouve K=2. De plus, pour le deuxième graphique, le critère Silhouette que l'on cherche à maximiser nous donne un K = 2 classes également. Enfin, grâce au troisième graphique, nous observons que l'on a un critère Silhouette moyen de 0.71 ce qui est assez élevé. On peut donc être assez confiant sur le fait de faire un clustering à 2 classes.

Observons le résultat du clustering sur les deux premiers plans de l'ACP.

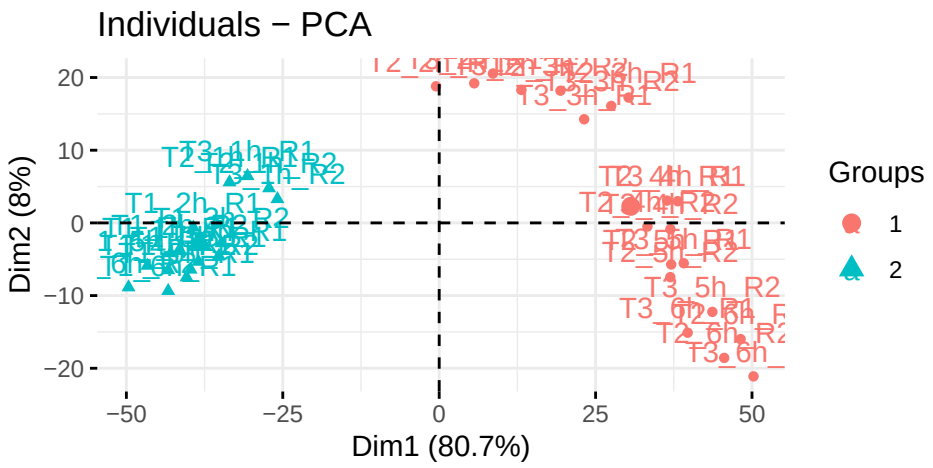


Figure 11: PCA sur les individus avec les clusters de kmeans

On peut distinguer facilement les deux cluster retenus.

-Le premier cluster représente majoritairement le traitement T1 peu importe l'heure ainsi que le traitement T2 et T3 à l'heure 1h. Ceci est valable pour les deux replicats, ce qui semble logique.

-le deuxième cluster représente majoritairement le traitment T2 et T3 à partir de l'heure 2h.

3.1.2 Classification Hiérarchique

Pour la classification hiérarchique, nous avons decidé d'utiliser la mesure d'agrégation de Ward. Nous avons fait ce choix car nous avons des données quantitatives qui ont toutes le même nombre d'individu. Observons le dendogramme obtenu :

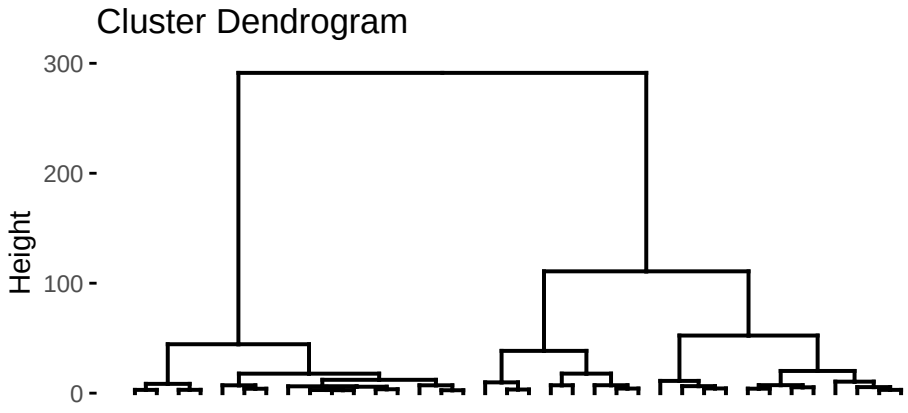


Figure 12: Dendrogramme

En observant le dendogramme, on voit que l'on peut choisir 2,3 ou bien 9 classes. Pour cela, nous allons utiliser le critère index.G1 afin de déterminer à quel endroit il faut couper le dendogramme pour obtenir les classes.

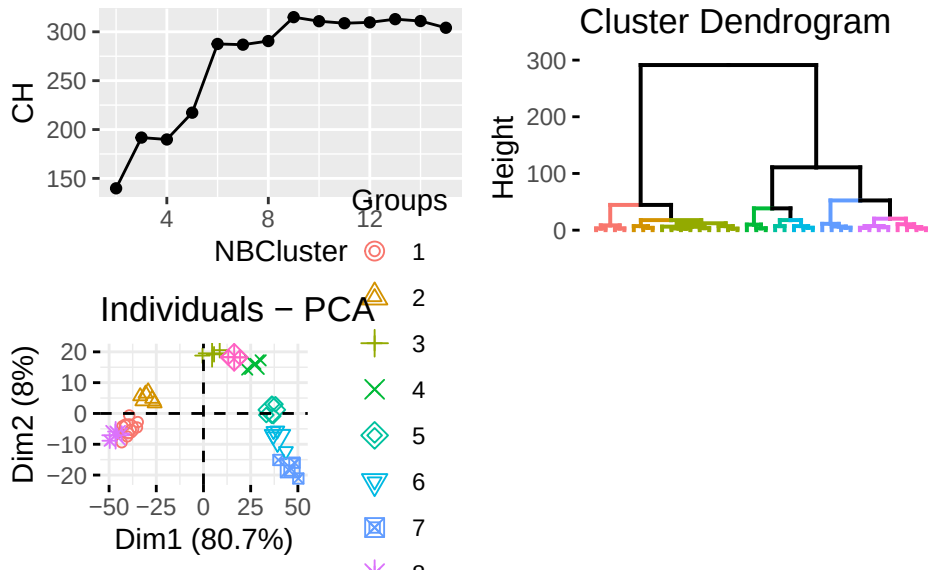


Figure 13: Graphs avec le nombre de classe optimal

Comme on cherche à maximiser le critère Calinski, on retient ici 9 classes. Nous comparerons ces 9 classes avec les autres résultats de clustering par la suite.

3.1.3 Modèle de mélange

Nous allons utilser une méthode de Clustering appelée "modèle de mélange" qui consiste à considérer un jeu de données contenant plusieurs sous-populations indépendantes et ayant leur propre loi de distribution. Le jeu de données est alors vu comme un mélange de toutes ces distributions. Un point x appartiendra à la classe K uniquement si la probabilité qu'il appartienne à cette classe est plus grande que pour les autres classes.

Pour réaliser cela, nous utilisons le fonction R Mclust. Nous lui demandons de tester les différents modèles allant de 2 classes à 15 classes selon le critere BIC, peu importe la forme de la classe. Voici les résultat obtenus :

Par défaut Mclust utilise le critère BIC pour choisir quel est le meilleur modèle (nombre de classes et forme des distributions), il va chercher à minimiser le critère BIC dans chacun des cas. D'après le graphique, on voit que c'est un modèle avec une forme de distribution "VEI" à 3 classes qui a été retenu.

Comme pour les K-means, le premier cluster contient les variables T1 à toutes les heures et les variables T2 et T3 à l'heure T1. Le deuxième cluster contient toutes les variables T2 et T3 aux heures 2h et 3h. Enfin, le dernier cluster contient toutes les variables T2 et T3 aux heures 4h, 5h et 6h.

Pour confirmer nos résultats, le clustering que nous avons obtenu avec le critère BIC, nous allons utiliser un deuxième critère de sélection avec les modèles mélanges : le critère ICL.

On voit que les critères BIC et ICL, donnent exactement le même clustering. On est donc confiant pour dire que notre clustering est plutôt "bon" dans notre situation.

3.1.4 Comparaison des différents clustering pour les variables

En utilisant la méthode des modèles de mélange, on trouve un cluster à 8 classes, ce qui est différent de tous les autres algorithmes de clustering où on trouve 2 classes.

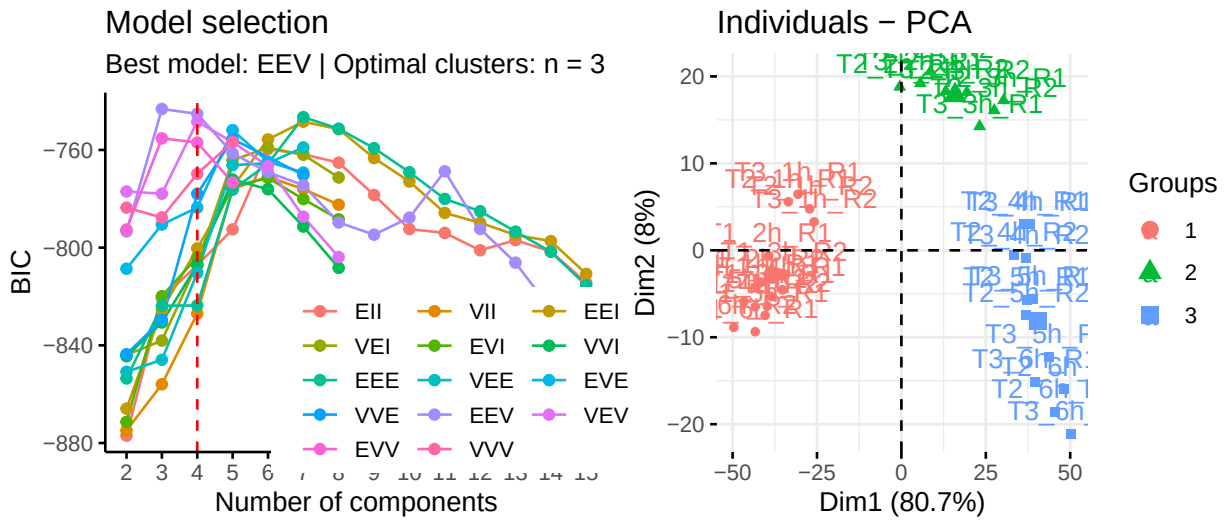


Figure 14: Des graphs pour le choix du meileur modèle

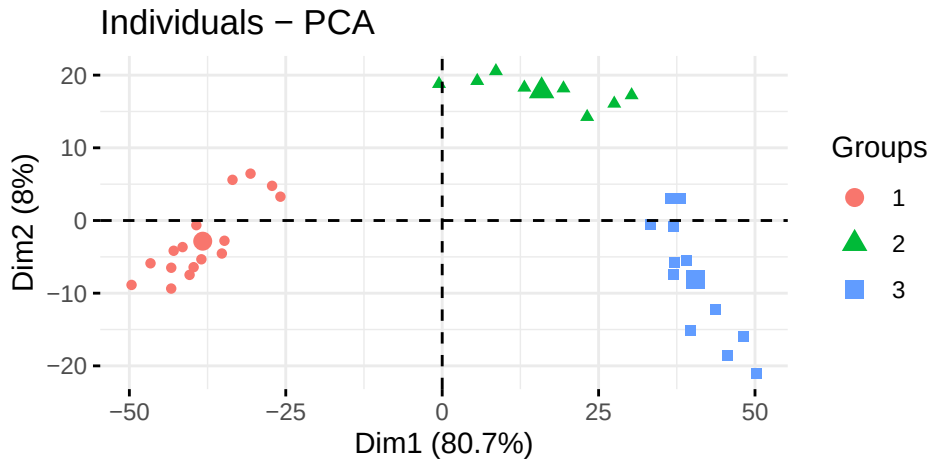


Figure 15: Classification choisi par ICL

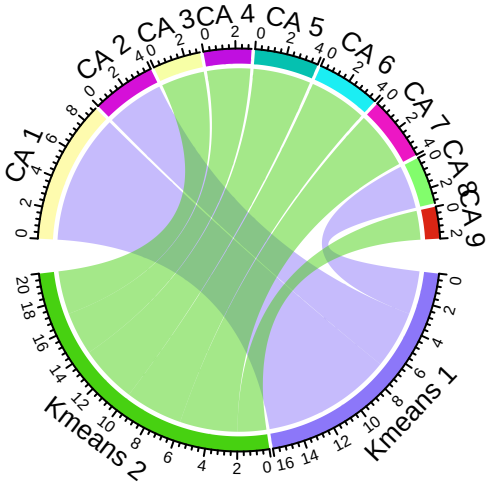


Figure 16: ChordDiagram

3.2 Classification des gènes

Ici, l'objectif de cette partie est de trouver des groupes de gènes qui se comportent de la même manière au cours du temps et des différents traitements. Comme pour la partie precédente, nous avons realisé une ACP sur le jeu de données pour nous donner une idée de la "forme" de notre jeu de données. Le nombre de variables étant de 36, nous avons choisi de ne pas réduire le nombre de dimensions pour effectuer les différentes méthodes clustering.

3.2.1 Avec K-means

Pour commencer, nous avons utilisé une méthode de K-means. Nous allons utiliser le même procédé pour que le clustering sur les variables, c'est-à-dire un n_start=15 et une mise en place de deux méthodes(variation de l'inertie Intra-classe et Silhouette) afin de déterminer le nombre de classes optimal.

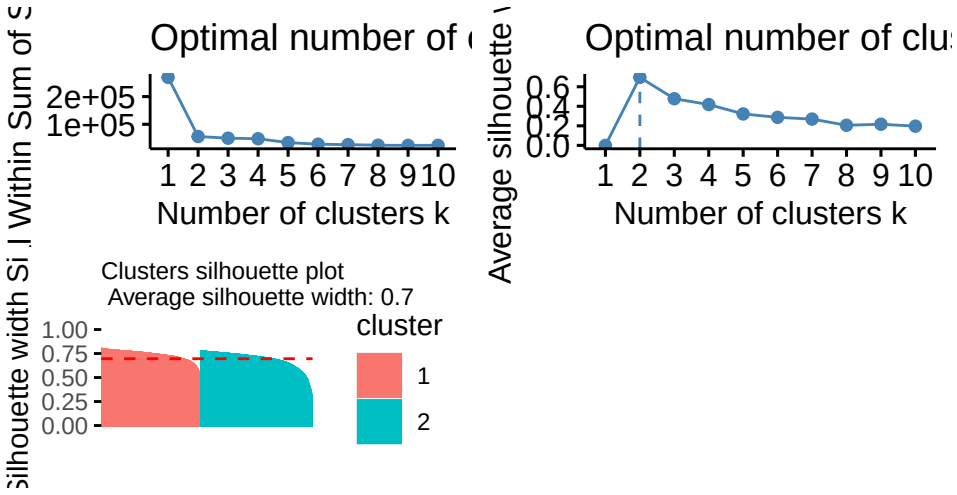


Figure 17: Choix du nombre de centre optimal

On observe qu'avec le critère Silhouette, le nombre de classes optimal est de deux classes. De plus, d'après le deuxième graphique, on remarque qu'on est plutôt confiant sur l'appartenance des points dans les classes puisque qu'on a une moyenne du critère Silhouette de 0.7. Observons le clustering obtenue dans les deux premier plan de l'ACP :

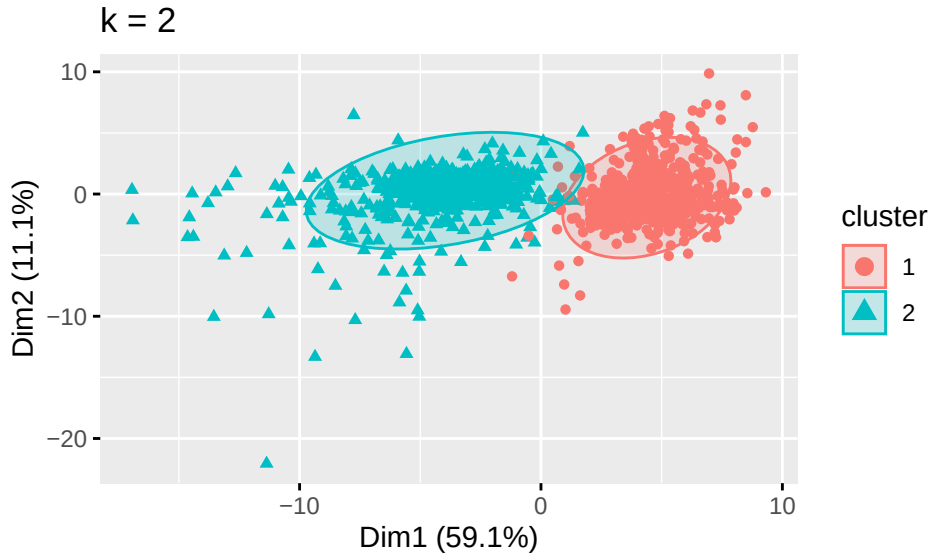


Figure 18: ACP avec 2 classes

On observe que les deux classes obtenues sont les deux classes que l'on pouvait distinguer visuellement sur l'ACP donc c'est) dire des gènes sous-exprimés pour le cluster 2 et des gènes sur-exprimés pour le cluster 1.

Malgrès cela, nous pensons qu'il est préferable de choisir un nombre de classes égal à 5 car dans ce cas, nous cherchons à distinguer différents groupes de gènes. Ainsi, une classification plus "fine" semble meilleure. Mais avant de faire ce choix, observons si le clustering à 2 classes est un regroupement des classes du clustering à 5 classes. Pour cela, regardons la table des effectifs par classe suivant :

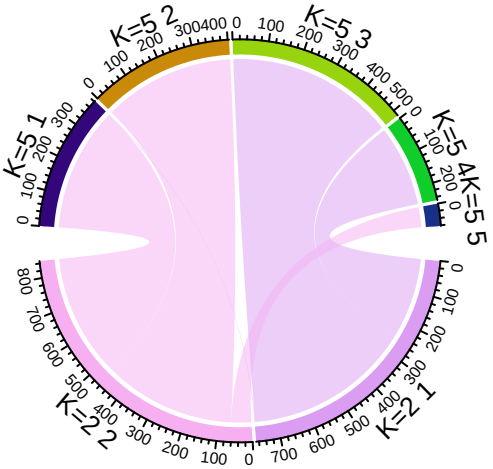


Figure 19: ChordDiagram

On voit bien que la classe 1 du clustering à deux classes contient entièrent les trois premières classes du clustering à 5 classes. De la même manière, la deuxième classe du clustering à deux classes contient les deux autres classes du clustering à 2 classes. Donc notre hypothèse de prendre 5 classes reste cohérente.

Maintenant que nous avons trouvé un clustering, nous allons interpreter les différents groupes. Pour cela, nous allons tracer les profils moyens par classe aux différents instant et au différents traitements :

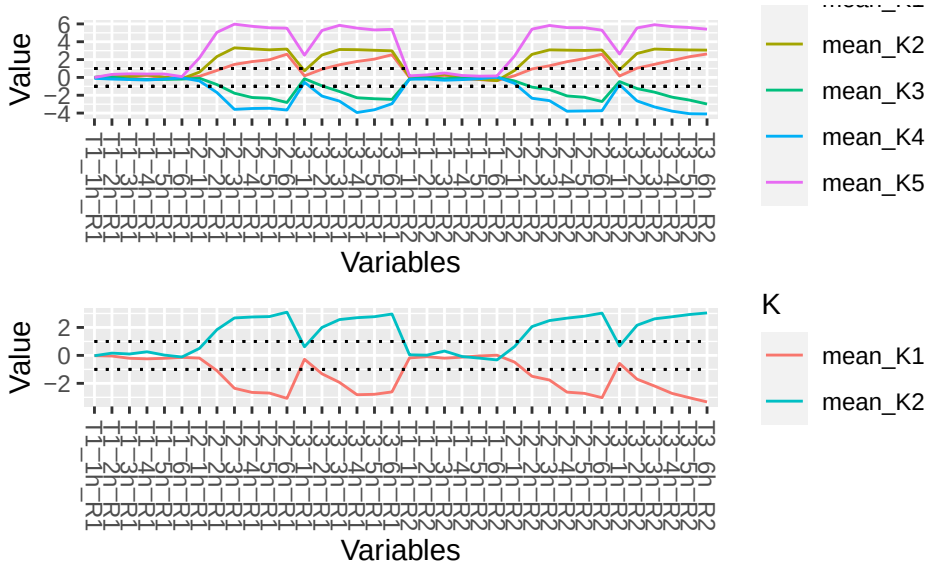
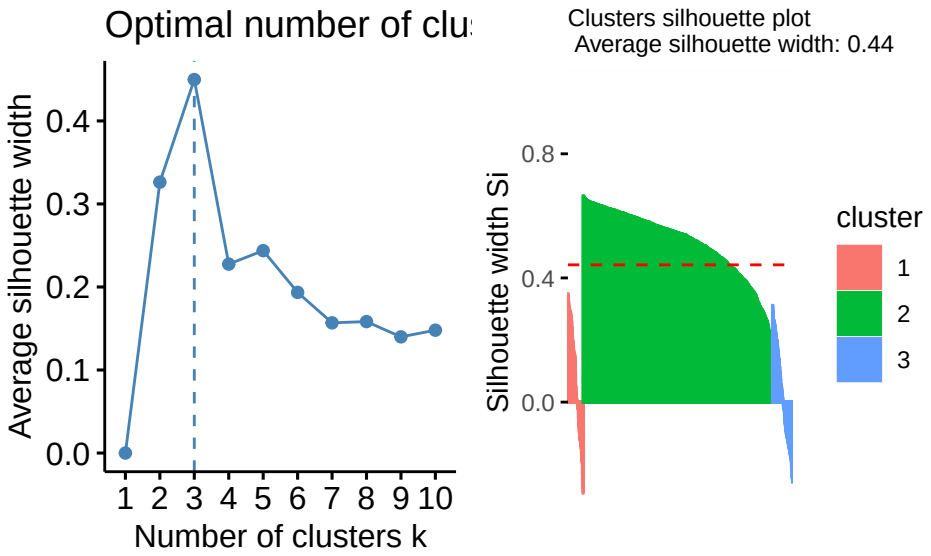


Figure 20: Analyse sur les clusters obtenus

On observe que :

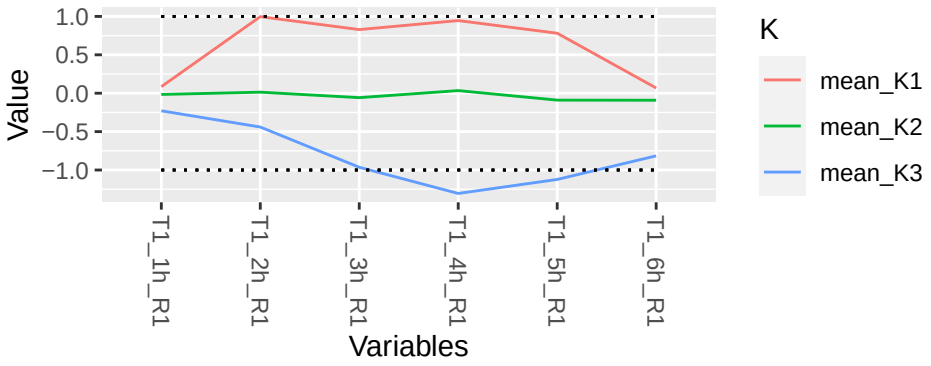
- Le cluster K4 représente essentiellement les gènes qui sont sur-exprimés, ils ont tous une réaction rapide et fortement positive aux traitements T2 et T3.
- Le cluster K3 sont les gènes qui sont très sur-exprimés, ils ont tous une réaction assez rapide et fortement positive au traitement T2 et T3 mais avec une intensité moins forte que la classe K3.
- Le cluster K1 les gènes qui sont sur-exprimés mais avec une réaction très progressive aux traitements T2 et T3.
- Le cluster K5 défini les gènes sous-exprimés. Ils ont tous une réaction progressive et négative.
- Enfin, pour le cluster K2, les gènes qui sont tres sous-exprimés. Ils ont tous une réaction rapide et fortement négative aux traitements T2 et T3.

De plus, on remarque que le clustering obtenu classe les gènes de la même façon entre le réplicat R1 et le réplicat R2, ce qui nous conforte dans l'idée qu'il existe un lien fort entre les gènes d'une même classe. Cependant, on observe le clustering différentie uniquement les gènes sur les traitements T2 et T3. Nous allons donc réaliser un clustering complémentaire uniquement sur les données du traitement T1.



On observe qu'avec le critère Silhouette, on prend 3 classes. De plus, les classes ont un nombre d'effectif très différent : K1 a 109 individus, K2 a 1362 individus et K3 en a 144.

Nous allons tracer les profils moyens par classe aux différents instants et aux différents traitements afin de mieux visualiser le rôle de chaque cluster.



On observe donc qu'il y a bien 3 clusters assez différents.

- Le cluster K1 a tendance à réagir positivement rapidement puis à rester constant pour finalement retomber à une valeur proche de 0.
- Le cluster K2 a tendance à reagir négativement et de manière plus progressive que le cluster K1 pour finalement retomber à partir de 4h.
- Le cluster K5 contient tous les gènes qui ne réagissent pas au traitement T1.

3.2.2 Avec la Classification Hiérarchique

Nous allons maintenant utiliser la classification hiérarchique pour essayer de retrouver le clustering précédent ou un nouveau clustering qui nous permettrai d'obtenir plus d'informations. Au vu du jeu de données, on choisit de prendre une distance Euclidienne et une méthode de Ward.

Il nous faut donc maintenant choisir un critère afin de savoir où le dendogramme (ce qui correspond au nombre de classes retenu) pour obtenir notre clustering. Pour cela, nous utilisons le critere “index.G1” que nous allons chercher à maximiser. Visuellement, on remarque que le nombre de classes optimal va se situer entre 2 et 10 classes.

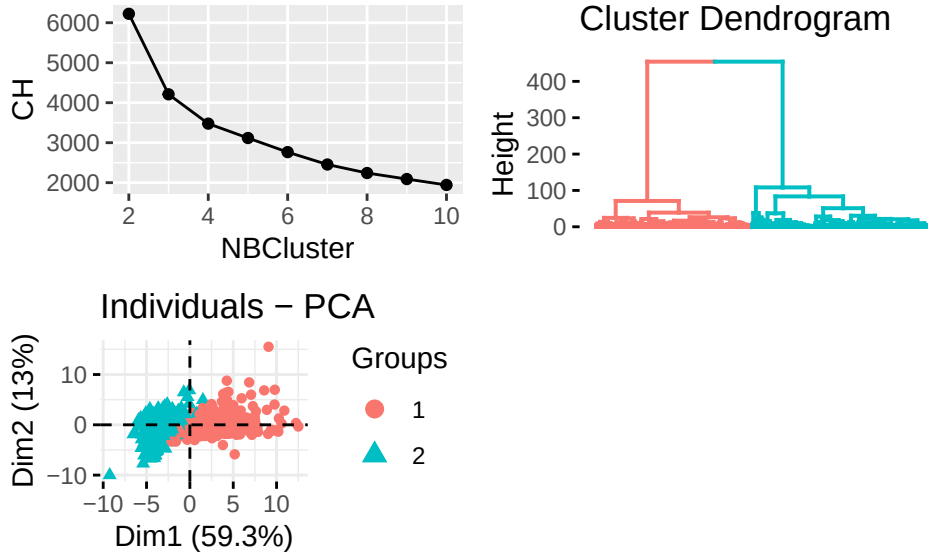


Figure 21: Le nombre de classe optimal

3.2.3 Avec les modèles de mélange

Enfin, nous allons utliser les modèles de mélange. Au vu des premiers clusterings et de la taille du jeu de données, nous allons limiter notre recherche de classes entre 2 et 15 classes mais on essaie toutes les formes de classe possible. Pour la sélection du nombre de classes, nous allons utiliser le citère BIC dans un premier temps, puis le critère ICL. Nous décidons de ne pas inclure les résultats du critère AIC dans le rapport car les résultats ne menaient à rien.

On remarque que l'on obtient un clustering VVE à 11 classes.

On obtient deux classes qui sont à chaque fois quasiment identiques.

4 Modélisation

4.1 Etude de l'expression des gènes pour le traitement T3 à 6h

4.1.1 Etude de l'expression des gènes pour le traitement T3 à 6h en fonction des autres expressions des gènes pour le traitement T3 :

On commence par modéliser l'expression des gènes pour le traitement T3 à 6h par les différents temps pour le traitement T3 fixé en réalisant une régression linéaire multiple.

$$\begin{cases} T3\_6h\_R2_i = \theta_0 + \theta_1 * T3\_1h\_R2_i + \cdots + \theta_5 * T3\_5h\_R2_i + \varepsilon_i, \quad i = 1, \ldots, k = 5 \\ (\varepsilon_i) \text{ i.i.d } \mathcal{N}(0, \sigma^2) \end{cases}$$

D'abord, on commence par vérifier les 4 hypothèses: $H1$: les ε_i sont : - centrées, - de variance constante, - indépendantes, - suivent une loi normale.

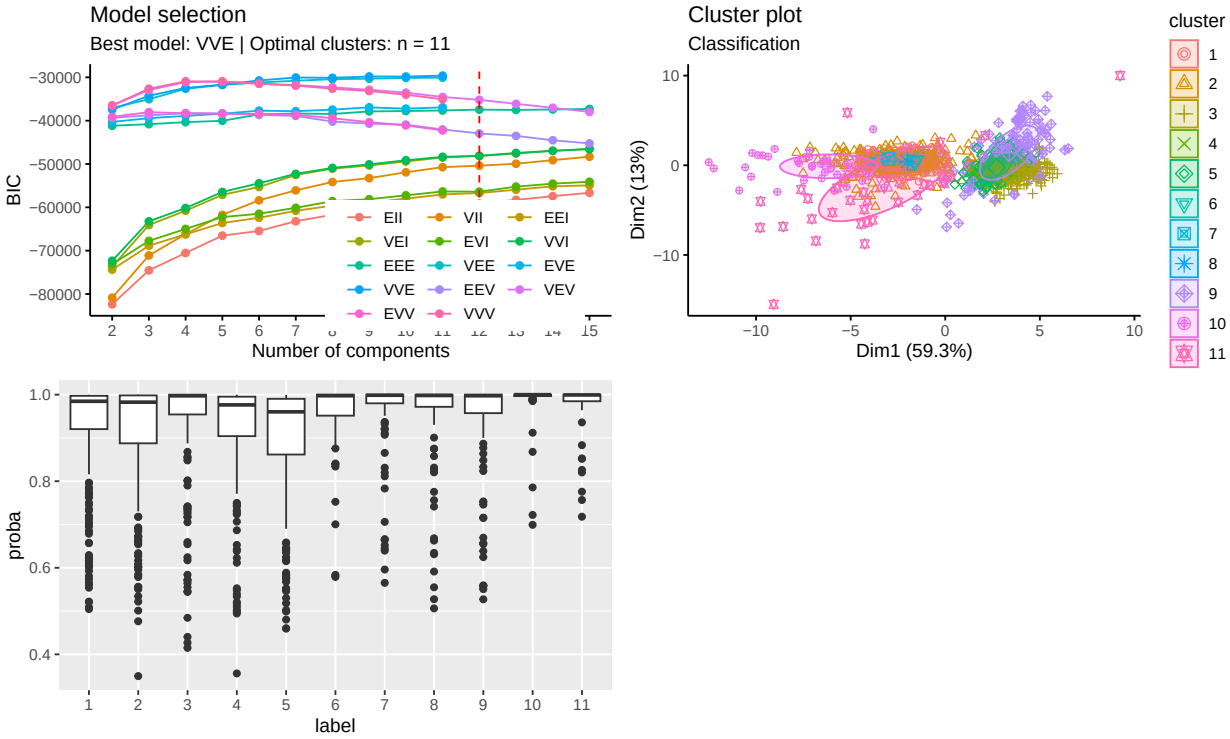


Figure 22: Analyse sur le le modèle optimal

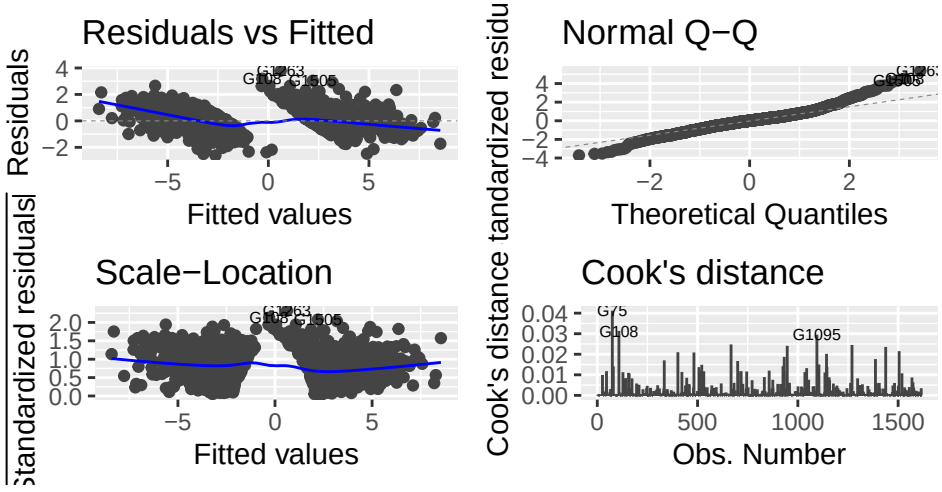


Figure 23: Graphiques de vérifications des hypothèses sur les erreurs

Les hypothèses précédentes sont validées. De plus, on a le modèle complet pour l'expression des gènes à 6h pour le traitement T3 fixé ci-dessous :

Table 2: Reduced Summary of Linear Model

term	estimate	p.value
(Intercept)	-0.0644084	0.0003974
T3_1h_R2	0.1357981	0.0000001
T3_2h_R2	-0.2436986	0.0000000
T3_3h_R2	0.1888772	0.0000030
T3_4h_R2	-0.1857737	0.0000193
T3_5h_R2	1.1786223	0.0000000

On remarque grâce aux p-valeurs des tests de Student (voir sorties dans le Rmd) qu'aucune des variables du modèle complet ne peuvent être supprimée avec une erreur à 5%.

4.1.2 Etude de l'expression des gènes pour le traitement T3 à 6h en fonction des autres expressions des gènes pour tous les traitements :

Maintenant, nous allons modéliser l'expression des gènes pour le traitement T3 à 6h par les différents temps les différents traitements par une regréssion linéaire multiple.

$$\begin{cases} T3\_6h\_R2_i = \theta_0 + \theta_1 * T1\_1h\_R2_i + \cdots + \theta_15 * T3\_5h\_R2_i + \varepsilon_i, \quad i = 1, \ldots, k = 15 \\ (\varepsilon_i) \text{ i.i.d } \mathcal{N}(0, \sigma^2) \end{cases}$$

De même que précédemment, les hypothèses sont vérifiées (les ε_i sont centrés, de variance constante, indépendants et suivent une loi normale).

Table 3: Summary of Linear Model

term	estimate	p.value
(Intercept)	-0.1056965	0.0000000
T1_1h_R2	0.0995677	0.0001494
T1_2h_R2	0.0032632	0.9040695
T1_3h_R2	0.0275281	0.2205777
T1_4h_R2	0.0325371	0.1745174
T1_5h_R2	-0.1850272	0.0000000
T1_6h_R2	0.0705961	0.0017656
T2_1h_R2	0.0571210	0.0160569
T2_2h_R2	0.0211564	0.3957984
T2_3h_R2	-0.0582911	0.0147770
T2_4h_R2	-0.0241517	0.2774255
T2_5h_R2	-0.2359026	0.0000000
T2_6h_R2	0.8489089	0.0000000
T3_1h_R2	-0.0200560	0.3510226
T3_2h_R2	0.0047948	0.8564334
T3_3h_R2	0.0241732	0.3326871
T3_4h_R2	-0.0305609	0.1783551
T3_5h_R2	0.4652246	0.0000000

Puisque le modèle complet présente plusieurs variables qui peuvent être supprimées avec une erreur de 5% (qui on une p-valeur > 0.05), on peut simplifier le modèle. Cependant, puisque plusieurs d'entre-elles peuvent être annulées, nous avons utilisé des procédures de choix de modèles pour sélectionner les variables significatives. On va ici comparer la sélection de variable obtenue par différents critères : BIC, AIC et Cp de Mallows.

La méthode backward consiste à commencer avec toutes les variables et à les supprimer une à une. La méthode forward consiste à commencer avec une seule variable et à ajouter progressivement des variables supplémentaires.

Les deux méthodes peuvent être utilisées pour sélectionner les variables les plus pertinentes et sont équivalentes dans notre cas. Nous avons donc choisi pour la suite la méthode backward.

Pour les méthodes de sélection de variables, nous avons utilisé la méthode bic, aic et le Cp de Mallows.

Table 4: Variable selectionnées avec CP

term	estimate	p.value
(Intercept)	-0.1061919	0.0000000
T1_1h_R2	0.0884944	0.0001905
T1_3h_R2	0.0405882	0.0166802
T1_5h_R2	-0.1660468	0.0000000
T1_6h_R2	0.0718861	0.0006484
T2_1h_R2	0.0472833	0.0001481
T2_3h_R2	-0.0499371	0.0031227
T2_5h_R2	-0.2410115	0.0000000
T2_6h_R2	0.8458119	0.0000000
T3_3h_R2	0.0312338	0.1000109
T3_4h_R2	-0.0390566	0.0607784
T3_5h_R2	0.4612992	0.0000000

Table 5: Variable selectionnées avec BIC

term	estimate	p.value
(Intercept)	-0.1063485	0.0000000
T1_1h_R2	0.0842082	0.0003423
T1_3h_R2	0.0483964	0.0017717
T1_5h_R2	-0.1613348	0.0000000
T1_6h_R2	0.0666999	0.0014433
T2_1h_R2	0.0508089	0.0000395
T2_3h_R2	-0.0379863	0.0000335
T2_5h_R2	-0.2499635	0.0000000
T2_6h_R2	0.8463700	0.0000000
T3_5h_R2	0.4485327	0.0000000

Table 6: Variable selectionnées avec AIC

term	estimate	p.value
(Intercept)	-0.1061919	0.0000000
T1_1h_R2	0.0884944	0.0001905
T1_3h_R2	0.0405882	0.0166802
T1_5h_R2	-0.1660468	0.0000000

term	estimate	p.value
T1_6h_R2	0.0718861	0.0006484
T2_1h_R2	0.0472833	0.0001481
T2_3h_R2	-0.0499371	0.0031227
T2_5h_R2	-0.2410115	0.0000000
T2_6h_R2	0.8458119	0.0000000
T3_3h_R2	0.0312338	0.1000109
T3_4h_R2	-0.0390566	0.0607784
T3_5h_R2	0.4612992	0.0000000

On a déduit 3 nouveaux sous modèles. On a le même modèle en utilisant AIC et CP. On va comparer maintenant les sous-modèles obtenus par BIC et AIC avec le modèle de base en effectuant des tests de Fisher (le modèle obtenu par BIC est un sous modèle de celui obtenu par AIC) :

Le tableau suivant résume les résultats des tests qui ont été faits :

test	p_value
AIC_vs_Complet	4.887079e-01
BIC_vs_AIC	1.131544e-01
BIC_vs_Complet	2.79821e-01

On fait un test de fisher de sous-modèle entre le modèle BIC et le modèle complet. On a une p-valeur = 0.2798, donc on accepte le modèle obtenu par BIC.

$$\begin{cases} T3\_6h\_R2_i = \theta_0 + \theta_1 * T1\_1h\_R2_i + \theta_2 * T1\_3h\_R2_i + \theta_3 * T1\_5h\_R2_i + \theta_4 * T1\_6h\_R2_i + \theta_5 * T2\_1h\_R2_i \\ +\theta_6 * T2\_3h\_R2_i + \theta_7 * T2\_5h\_R2_i + \theta_8 * T2\_6h\_R2_i + \theta_9 * T3\_5h\_R2_i + \varepsilon_i, \quad i = 1, \ldots, k = 9 \\ (\varepsilon_i) \text{ i.i.d } \mathcal{N}(0, \sigma^2) \end{cases}$$

Par un test de fisher entre le modèle AIC et le modèle complet, on a accepte le modèle AIC. Finalement, en comparant BIC avec AIC, on trouve une p-valeur égale à 0.1132. Ainsi, on accepte le modèle obtenu par BIC.

4.2 Modèle linéaire généralisé pour T3 à 6h

On remarque que pour $T3_6h_R2$ les gènes sont soit sur-exprimés soit sous-exprimés, ce qui est en accord avec les résultats trouvés dans les statistiques descriptives. Donc on va voir si une régression logistique peut bien expliquer cette variable.

D'abord, on change T3_6h_R2 d'une variable quantitative à une variable binaire qui prend 1 lorsque le gène est sur-exprimé sinon c'est 0. Cette dernière va suivre une loi binomiale donc pour la suite, on va utiliser une régression logistique. En effet, lorsque les variables dans une régression logistique sont corrélées, cela peut causer un phénomène connu sous le nom de "multicollinéarité". Cela se produit lorsque deux ou plusieurs variables indépendantes dans un modèle de régression sont fortement corrélées entre elles. Cela peut poser des problèmes pour estimer les coefficients de régression et il peut également être difficile de déterminer lesquelles des variables ont réellement un effet sur le résultat. Dans le cas de la multicollinéarité, l'estimation des coefficients de régression devient instable et l'algorithme peut ne pas converger. Une façon de résoudre ce problème est de retirer une des variables corrélées du modèle ou d'utiliser une forme différente de régularisation. On peut également utiliser une autre méthode qui consiste à utiliser l'ACP pour extraire des composantes non corrélées des variables corrélées, et d'utiliser ces composantes non corrélées en tant qu'entrée pour le modèle de régression logistique.

4.2.1 Première méthode

On enlève toutes les variables corrélés avec T3_6h_R2. Donc on va l'exprimer avec que le traitment T1 avec et sans interaction.

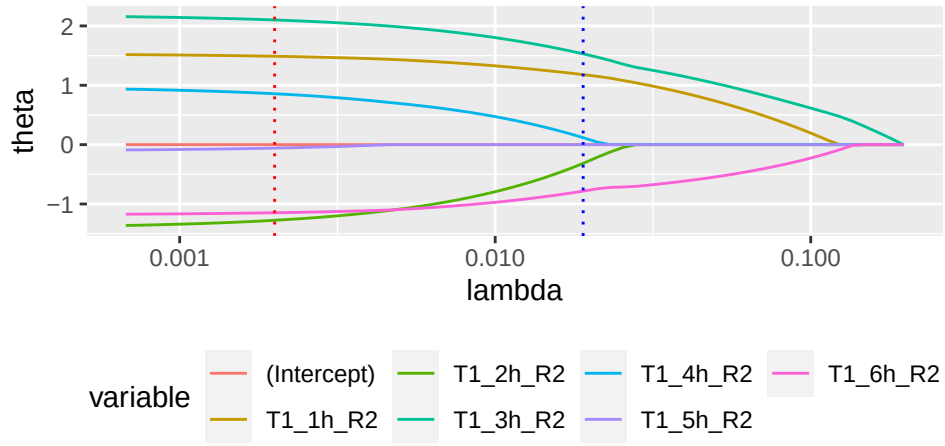


Figure 24: Selection de variable avec Lasso

```
## [1] 0.1919505

##
##       0   1
##   0 111  31
##   1  31 150

## [1] 0.8080495
```

On trouve 20% de risque d'erreur et donc une accurance de 80%. On va tester le modèle avec intéraction.

```
## Analysis of Deviance Table
##
## Model 1: T3_6h_R2 ~ T1_1h_R2 + T1_2h_R2 + T1_3h_R2 + T1_4h_R2 + T1_5h_R2 +
##     T1_6h_R2
## Model 2: T3_6h_R2 ~ (T1_1h_R2 + T1_2h_R2 + T1_3h_R2 + T1_4h_R2 + T1_5h_R2 +
##     T1_6h_R2)^2
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1      1608     1665.4
## 2      1593     1507.5 15   157.89 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

On a également testé le modèle avec interactions et on a trouvé sensiblement les mêmes résultats. On a aussi un pseudo R^2 plus élevé pour le modèle avec interaction. De plus, nous avons réalisé un test de sous-modèle entre celui avec interactions et sans interactions. On trouve une p-valeur«0.05. Donc on va garder le modèle avec intéractions. On remarque aussi qu'on peut enlever quelques interactions grâce au Z-test.

La régression logistique ne fontionne que lorsqu'on prend que T1 et on arrive à exprimer et prédire T3_6h_R2.

4.2.2 Deuxième méthode

On garde toutes les variables et on utilise une sélection de variables avec la régression Lasso et on trouve les résultats suivant :

Selected_Variables
T2_6h_R2
T3_4h_R2
T3_5h_R2

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 764   0
##          1   0 851
##
##                Accuracy : 1
##                  95% CI : (0.9977, 1)
##     No Information Rate : 0.5269
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 1
##
##  Mcnemar's Test P-Value : NA
##
##             Sensitivity : 1.0000
##             Specificity : 1.0000
##          Pos Pred Value : 1.0000
##          Neg Pred Value : 1.0000
##              Prevalence : 0.4731
##          Detection Rate : 0.4731
##    Detection Prevalence : 0.4731
##       Balanced Accuracy : 1.0000
##
##        'Positive' Class : 0
##
```

On a une précision de 100% et on a un pseudo-R^2=1 ce qui veut dire qu'on est dans le cas de overfitting. Donc ce modèle n'est pas très adapté pour prédire T3_6h_R2. Les variables sont trop corrélées.

4.2.3 3-ème méthode

On utilise les données obtenus de l'ACP pour essayer d'éviter le problème de correlation entre les variables.

```
## [1] 0.005247925
```

On a un pseudo R^2 trop faible donc on peut déduire que ce n'est pas un bon modèle pour expliquer la variable T3_6h_R2.

Conclusion: Le meilleur choix est de ne garder que le traitement T1 pour prédire T3.

4.3 Etude de l'expression des gènes pour le traitement T1 à 6h

On procède de même façon en commençant par modéliser l'expression des gènes pour le traitement T1 à 6h par les différents temps pour le traitement T1 fixé. On effectue une regréssion linéaire multiple.

$$\begin{cases} T1\_6h\_R2_i = \theta_0 + \theta_1 * T1\_1h\_R2_i + \cdots + \theta_5 * T1\_5h\_R2_i + \varepsilon_i, \quad i = 1, \ldots, k = 5 \\ (\varepsilon_i) \text{ i.i.d } \mathcal{N}(0, \sigma^2) \end{cases}$$

Après avoir effectué des tests de nullité, on déduit le modèle suivant :

Table 9: Selected variables of Linear Model

term	estimate	p.value
(Intercept)	-0.0493897	2.10e-06
T1_2h_R2	0.1240513	3.84e-05
T1_3h_R2	-0.2861604	0.00e+00
T1_4h_R2	0.2307127	0.00e+00
T1_5h_R2	0.5595213	0.00e+00

4.3.1 Etude de l'expression des gènes pour le traitement T1 à 6h en fonction des autres expressions des gènes pour tous les traitements

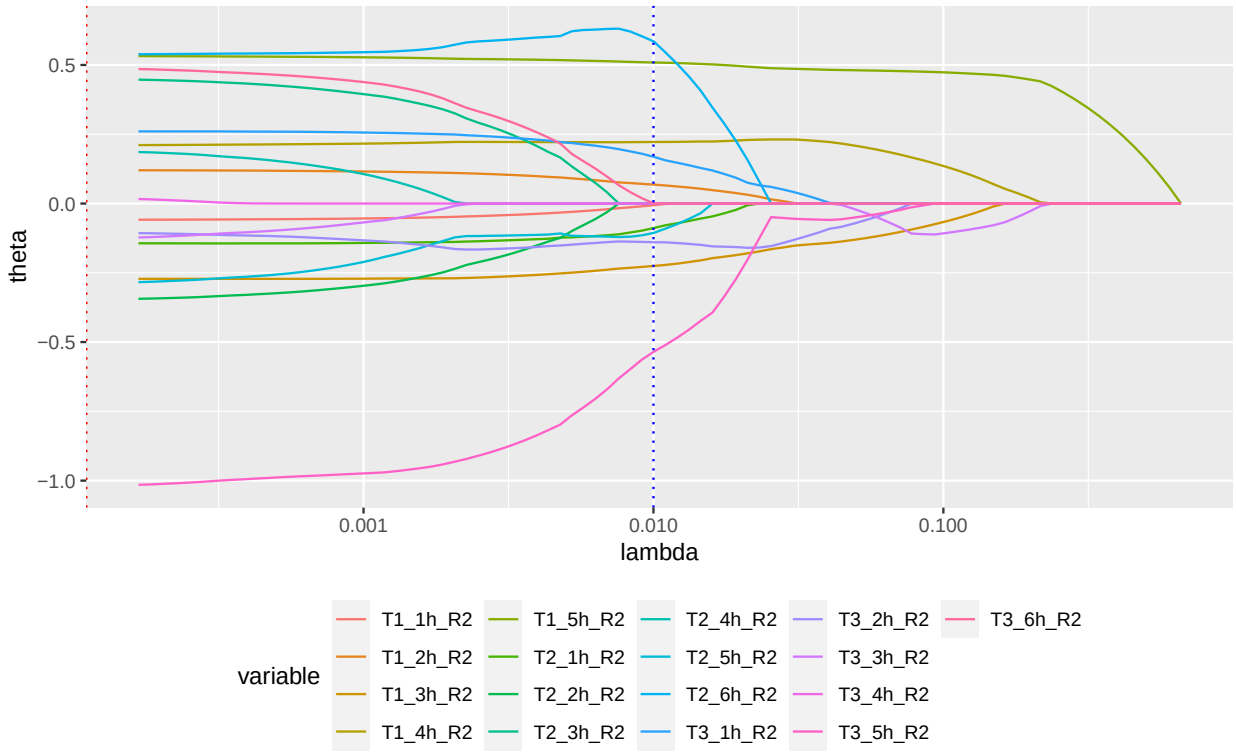
Maintenant on va modéliser l'expression des gènes pour le traitement T1 à 6h par les différents temps pour les différents traitements.

$$\begin{cases} T1\_6h\_R2_i = \theta_0 + \theta_1 * T1\_1h\_R2_i + \cdots + \theta_15 * T3\_5h\_R2_i + \varepsilon_i, \quad i = 1, \ldots, k = 15 \\ (\varepsilon_i) \text{ i.i.d } \mathcal{N}(0, \sigma^2) \end{cases}$$

Table 10: Selected variables of Linear Model

term	estimate	p.value
(Intercept)	-0.0514190	0.0000032
T1_1h_R2	-0.0901953	0.0019286
T1_2h_R2	0.1389152	0.0000033
T1_3h_R2	-0.2395640	0.0000000
T1_4h_R2	0.2129738	0.0000000
T1_5h_R2	0.5853892	0.0000000
T2_1h_R2	-0.0827656	0.0016124
T2_2h_R2	-0.0921768	0.0008144
T2_3h_R2	0.1065700	0.0000551
T2_4h_R2	0.0401996	0.1023429
T2_5h_R2	-0.0570989	0.0554852
T2_6h_R2	0.0939566	0.0005709
T3_1h_R2	0.1354878	0.0000000
T3_2h_R2	-0.0232303	0.4283951
T3_3h_R2	-0.0316035	0.2525273
T3_4h_R2	0.0090624	0.7185084
T3_5h_R2	-0.1870921	0.0000000
T3_6h_R2	0.0864963	0.0017656

On va effectuer une sélection de variables en utilisant la régression Lasso comme méthode de sélection :



On garde les variables suivantes en les pénalisant avec le lambda minimum :

```
## 18 x 1 sparse Matrix of class "dgCMatrix"
##                         s1
## (Intercept)  1.554878e-17
## T1_1h_R2    -5.956673e-02
## T1_2h_R2     1.208494e-01
## T1_3h_R2    -2.707376e-01
## T1_4h_R2     2.080233e-01
## T1_5h_R2     5.340386e-01
## T2_1h_R2    -1.405376e-01
## T2_2h_R2    -3.713317e-01
## T2_3h_R2     4.748870e-01
## T2_4h_R2     2.125620e-01
## T2_5h_R2    -3.042662e-01
## T2_6h_R2     5.218571e-01
## T3_1h_R2     2.581913e-01
## T3_2h_R2    -8.480684e-02
## T3_3h_R2    -1.613104e-01
## T3_4h_R2     4.109062e-02
## T3_5h_R2    -1.041028e+00
## T3_6h_R2     5.127471e-01
```

On n'arrive pas à réduire le modèle avec la méthode lasso, donc on va tester une autre méthode :

On obtient deux modèles à partir des méthodes BIC et AIC comme méthode de sélection de variables.

Dans le cas présent, le modèle trouvé avec la méthode BIC est un sous modèle de celui trouvé avec la méthode AIC.

On effectue des tests de sous-modèle de Fisher afin de comparer les modèles obtenus car les modèles sont sous-modèle les uns des autres.

On accepte dans un premier temps le modèle AIC car on a une p-valeur = 0.6966 » 0.05. Et finalement, on accepte le modèle obtenu par BIC (une p-valeur=0.09).

4.4 Modèle Linéaire Généralisé pour T1 à 6h

```
## # weights:  57 (36 variable)
## initial  value 1419.407077
## iter  10 value 379.507572
## iter  20 value 303.371562
## iter  30 value 285.433722
## iter  40 value 284.366093
## final  value 284.362271
## converged
```

```
##           actual
## predicted  -1   0   1
##        -1   5   1   0
##        0   15 292   3
##        1    0   5   2
```

```
## [1] 0.9256966
```

On a une précision de 92%. On prédit bien la variable T1_6h_R2. Cependant, on a un petit souci avec les factors 1 et -1 car on a un manque de valeurs par rapport au facteur 0. La prédiction n'est donc pas trop fiable. Si on prend le cas du facteur 1, on a plus de 50% d'erreur. On en déduit que, avec ce modèle, on a une difficulté à prédire T1_6h_R2 quand les réponses sont sur/sous exprimées.

5 Conclusion

Lors de ce projet, nous avons pu mettre en pratique les notions étudiées lors du cours de modélisation statistique et analyse des données. Les différentes méthodes et algorithmes que nous avons appliqué permettent d'avoir une analyse plus approfondie de nos données.

D'abord, nous avons commencé par réaliser quelques statistiques descriptives sur notre jeu de données, pour pouvoir ensuite faire une classification en implémentant différentes méthodes qui nous ont permis d'identifier deux classes pour les gènes sur-exprimées et sous-exprimées. Ensuite, en comparant entre plusieurs algorithmes, nous avons pu sélectionner des modèles linéaires qui dépendent uniquement des temps affectant réellement l'expression des gènes à 6h pour les traitements T3 et T1. Nous avons également testé plusieurs modèles pour trouver les variables prédictives qui permettent de discriminer entre les gènes sur- / sous-/non-exprimés à 6h pour les traitements T1 et T3.

Enfin, ce projet a montré l'importance de chacune des étapes citées précédemment, pour mieux comprendre notre jeu de données et avoir des conclusions solides pour poursuivre dans une étude plus approfondie de nos données.