Rapport de projet Analyse de données & Eléments de modélisation statistique

Xiaoya Wang, Mickael Song, Yessine Jmal, Florian Grivet

Contents

Tout ce qui se trouve dans le Rmarkdown mais pas dans le pdf est indiqué par ce symbole : (%%)

1 Introduction

On observe pour G = 1615 gènes d'une plante modèle les valeurs suivantes :

$$Y_{gtsr} = log_2(Xgtsr + 1) - log_2(Xgt0 + 1)$$

avec
- $Xgtsr$ la mesure d'expression du gène g ∈ {G1, . . . , G1615} pour le traitement t ∈ {T1, T2, T3} pour le réplicat r ∈ {R1, R2} et au temps s ∈ {1h, 2h, 3h, 4h, 5h, 6h}
- $Xgt0$ l'expression du gène g pour un traitement de référence t0

Nous allons répartir l'étude de ce jeu de données en 4 parties :
- Analyse du jeu de données
- Clustering
- Etude de l'expression des gènes pour le traitement T3 à 6h
- Etude de l'expression des gènes pour le traitement T1 à 6h

2 Analyse du jeu de données

Nous allons dans cette partie effectuer une analyse des statistiques descriptives et préparer le jeu de données afin d'en sortir les variables redondantes, transformations, outliers et visualiser le jeu de données dans un espace de faible dimension (en particulier l'aspect réplicat biologique, l'effet traitement et l'effet temps)

2.1 Statistiques descriptives et préparation du jeu de données

Table 1: Les premières lignes du jeu de données.

	T1_1h_R1	T1_2h_R1	T1_3h_R1	T1_4h_R1	T1_5h_R1	T1_6h_R1	T2_1h_R1	T2_2h_R1
G1	0.17	0.68	-0.18	0.08	0.00	0.50	-0.59	0.19
G2	0.19	0.82	-0.05	0.18	0.47	-0.76	0.38	2.51
G3	-0.05	-0.03	0.26	-0.32	-0.39	0.42	0.21	-1.00
G4	-0.23	-0.75	-0.24	-0.70	-0.12	-0.38	0.41	0.23
G5	-0.21	-0.69	-0.18	-0.07	0.52	0.45	-0.45	-1.51
G6	-0.62	-0.86	-0.02	-0.14	0.49	0.45	-0.57	-1.48

Le jeu de données contient 1615 individus et 36 variables, toutes quantitatives.
Les attributs du jeu de données sont :
T1_1h_R1, T1_2h_R1, T1_3h_R1, T1_4h_R1, T1_5h_R1, T1_6h_R1, T2_1h_R1, T2_2h_R1, T2_3h_R1, T2_4h_R1, T2_5h_R1, T2_6h_R1, T3_1h_R1, T3_2h_R1, T3_3h_R1, T3_4h_R1, T3_5h_R1, T3_6h_R1, T1_1h_R2, T1_2h_R2, T1_3h_R2, T1_4h_R2, T1_5h_R2, T1_6h_R2, T2_1h_R2, T2_2h_R2, T2_3h_R2, T2_4h_R2, T2_5h_R2, T2_6h_R2, T3_1h_R2, T3_2h_R2, T3_3h_R2, T3_4h_R2, T3_5h_R2, T3_6h_R2

Avec le résultat de la commande python `datapy.isnull().sum()`, on voit bien que notre jeu de données est complet. (%%)

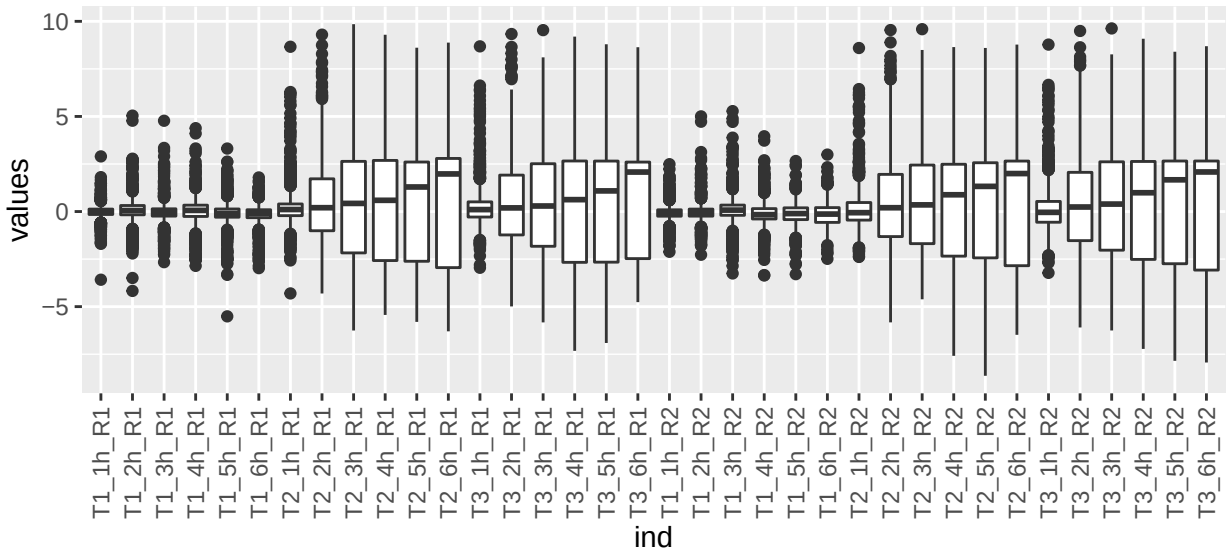


Figure 1: Boxplots des 36 variables

On remarque dans la figure 1 que les boxplots du traitement 2 et du traitement 3 ne sont pas centrés, ils ont donc un effet non nul sur les gènes. En étudiant la forme des boxplots, on remarque une dissymétrie pour ces deux traitements, des nombreux outliers ainsi qu'une forte variabilité entre les individus.
Les boxplots du traitement 2 et du traitement 3 sont d'ailleurs similaires, quelque soit le réplicat. On peut donc faire l'hypothèse que ces deux traitements donnent des résultats similaires. Le traitement 1 est quant à lui beaucoup plus réduit et centré en 0. Ce traitement semble donc ne pas avoir d'effet sur les gènes. Les boxplots du traitement 1 sont symétriques mais possèdent beaucoup d'outliers.

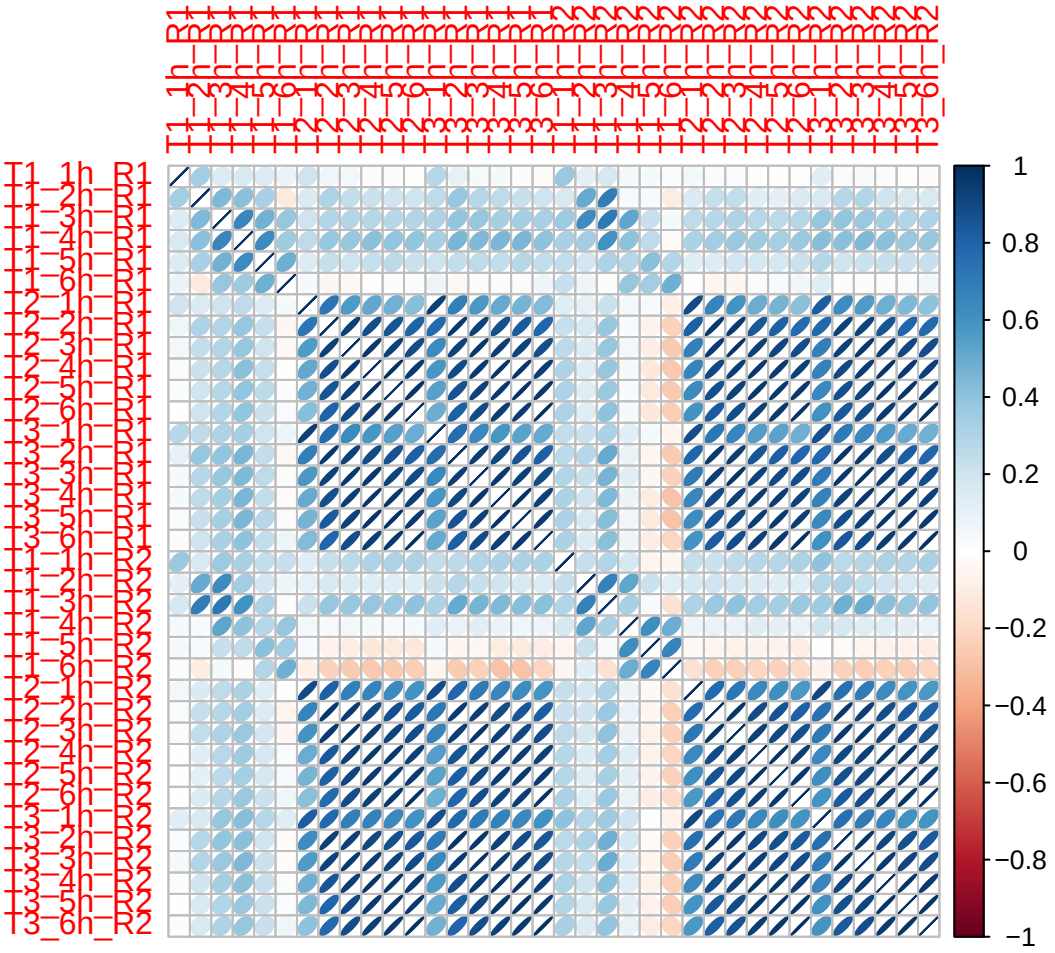


Figure 2: Graphique des corrélations des 36 variables

La figure 2 des corrélations nous confirme bien l'hypothèse précédente, les traitements 2 et 3 sont fortement corrélés alors que ces traitements semblent totalement décorellés du traitements 1.
On peut également noter le fait que, pour un traitement donné, le réplicat 1 et le réplicat 2 sont fortement correlé entre eux, ce qui est cohérent puisque ce sont des réplicats biologiques. On pourra donc par la suite uniquement faire notre étude uniquement sur 1 seul réplicat, sans perdre trop d'informations.

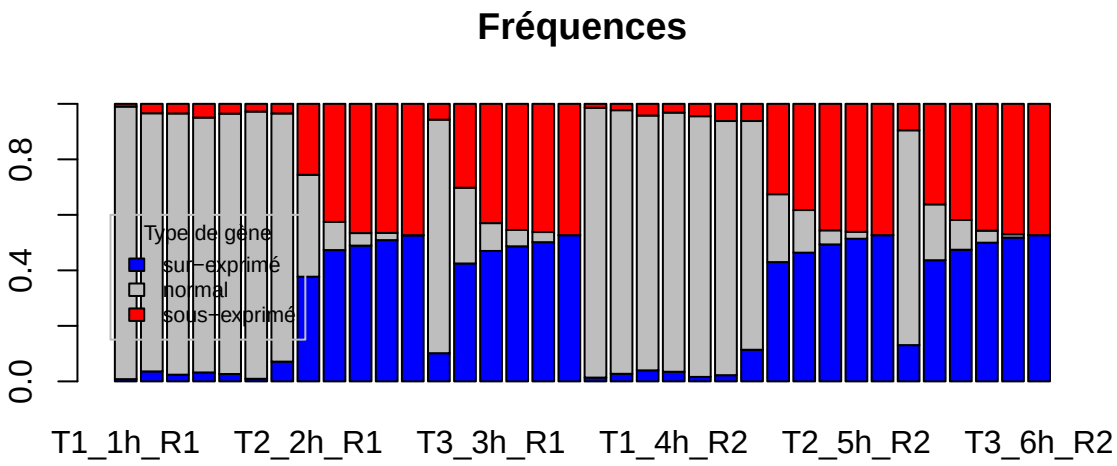


Figure 3: Fréquence de l'expression des gènes sous-exprimés, normaux et sur-exprimés en fonction des traitements

La graphique 3 représente la fréquence des gènes "sous-exprimés", "normaux" et "sur-exprimés" pour chaque traitement à toute heure sur les deux réplicats.
Il appuie notre hypothèse que les traitements 2 et 3 sont similaires et que le traitement 1 n'a pas beaucoup d'effet.

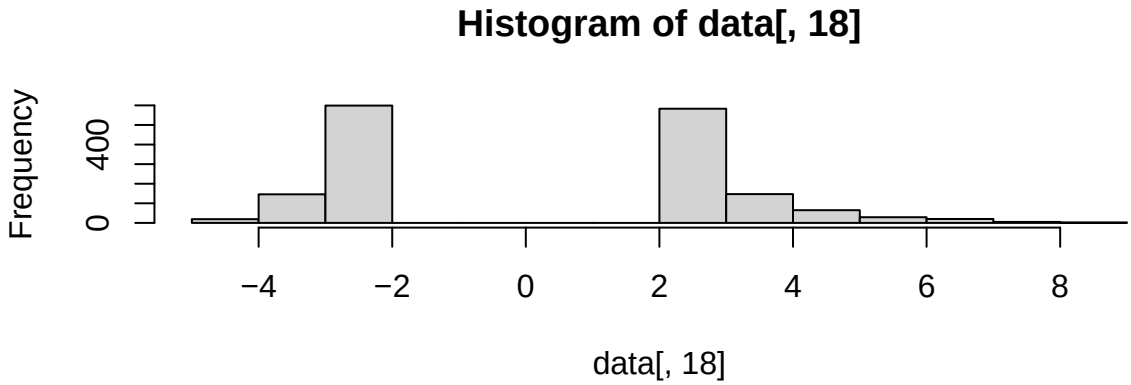


Figure 4: Fréquence de la valeur des gènes du traitement 3 à l'heure 6

On remarque qu'à la dernière heure (6h) du traitement 3, tous les gènes sont soit très sur-exprimé (valeurs ≥ 2), soit très sous-exprimé (valeurs ≤ 2). Les gènes du jeu de données ont donc été choisis en fonction de T3_6H.

2.2 Analyse en composante principale

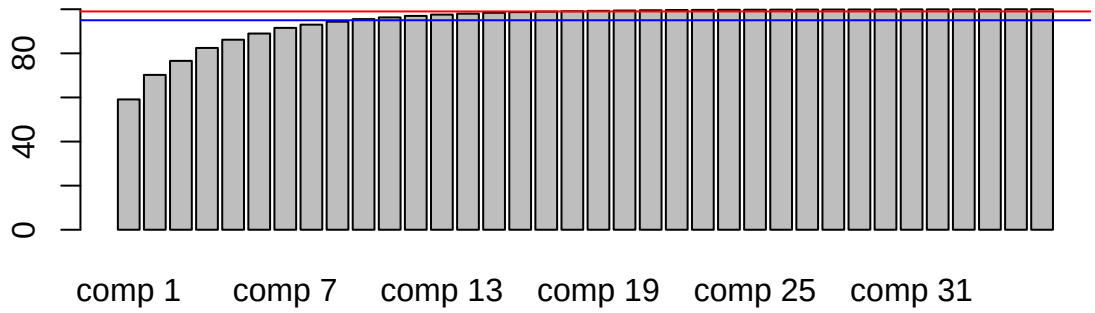


Figure 5: Variance expliquée cumulée (en %) des différentes composantes principales

D'après le graphique 5, on note que :
- Pour avoir 95% de l'information, on peut réduire nos données à 10 dimensions.
- Pour avoir 99% de l'information il suffit de se placer en dimension 18.

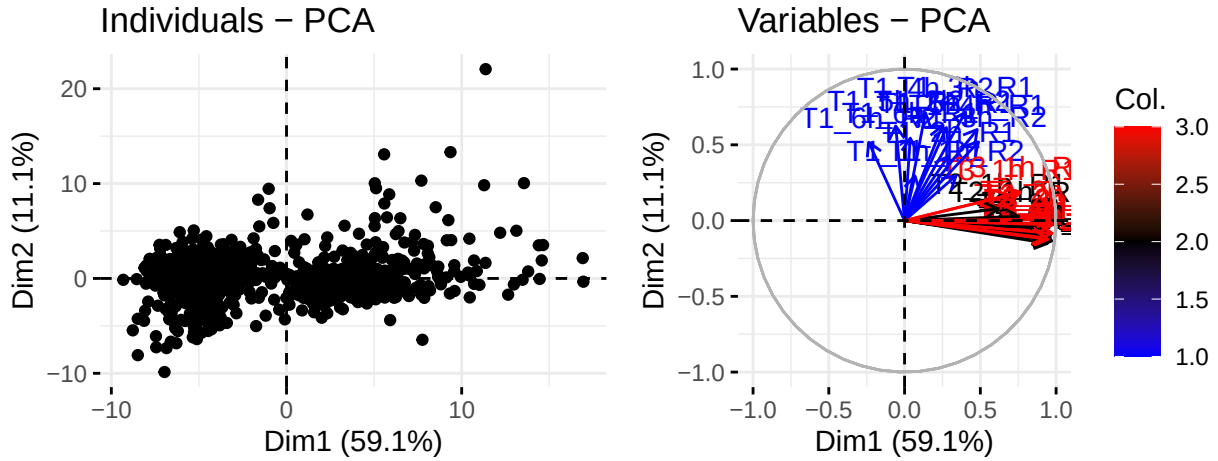


Figure 6: Visualitaion de l'ACP sur les deux premières composantes principales, pour les individus (à gauche) et pour les variables (à droite)

La première composante principale de la figure 6 nous dit si un gène réagit au traitement 2 ou 3. Si le gène réagit fortement à l'un de ces traitements, il se trouve aux extrémités du cercle (si le gène devient très sous-exprimé ou très sur-exprimé) et s'il ne réagit pas beaucoup à l'un de ces traitement il se trouve au milieu du cercle.

La deuxième composante principale nous dit si un gène réagit au traitement 1. Si le gène réagit fortement à ce traitement, c'est-à-dire s'il est sur-exprimé (resp. sous-exprimé), il se retrouve en haut (resp. en bas) du cercle. Par contre, si le gène ne réagit pas beaucoup au traitement 1, il se trouve au milieu.

2.2.1 Analyse en composante principale sur les données transposées

```
data_transpose = datapy.T
data_transpose_scale = StandardScaler().fit_transform(data_transpose)
pca_transpose = PCA()
acp_transpose = pca_transpose.fit(data_transpose_scale)
eig_transpose = pd.DataFrame(
    {
        "Dimension" : ["Dim" + str(x + 1) for x in range(36)],
        "Variance expliquée" : pca_transpose.explained_variance_,
        "% variance expliquée" : np.round(pca_transpose.explained_variance_ratio_ * 100),
        "% cum. var. expliquée" : np.round(np.cumsum(pca_transpose.explained_variance_ratio_) * 100)
    }
)
```

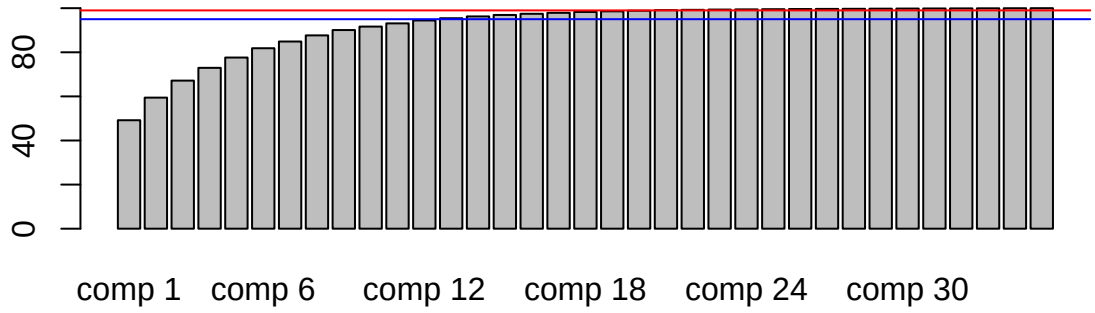
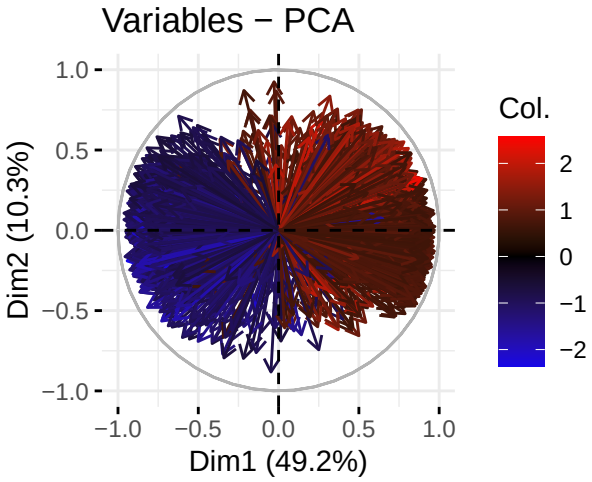
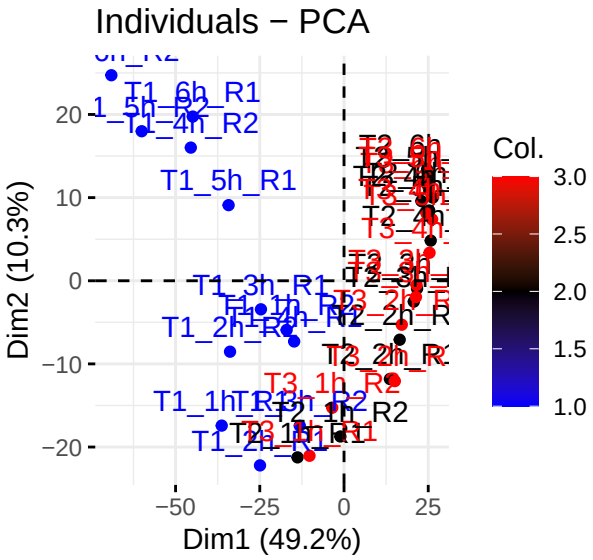


Figure 7: Variance expliquée cumulée (en %) des différentes composantes principales

D'après le graphique 7, on note que :
Pour avoir 95% de l'information, on peut réduire nos données à 13 dimensions.
Pour avoir 99% de l'information il suffit de se placer en dimension 21.





Commentaires à faire

3 Clustering

Pour mieux comprendre les relations entre les variables et les gènes dans les différentes conditions, nous allons utiliser, dans cette partie, différentes méthodes de clustering pour obtenir une classification des variables "T_xxh_Rx" et des gènes ayant des profils d'expression similaires.

3.1 Obtention d'une classification des variables "T_x_xh_Rx"

On reprend les 3 premières composantes principales de l'ACP effectué précedemment sur les variables. Les premières 3 composantes principales résument 90% de l'information.

```
##   comp 1   comp 2   comp 3   comp 4   comp 5
## 80.67096 88.69766 90.72446 92.45233 93.87476
```

3.1.1 K means

Avant d'appliquer l'algorithme K-means sur notre jeu de données, nous allons déterminer le nombre optimal de classes. Pour cela, nous allons tracer l'évolution de l'inertie intraclasse en fonction du nombre de classes.

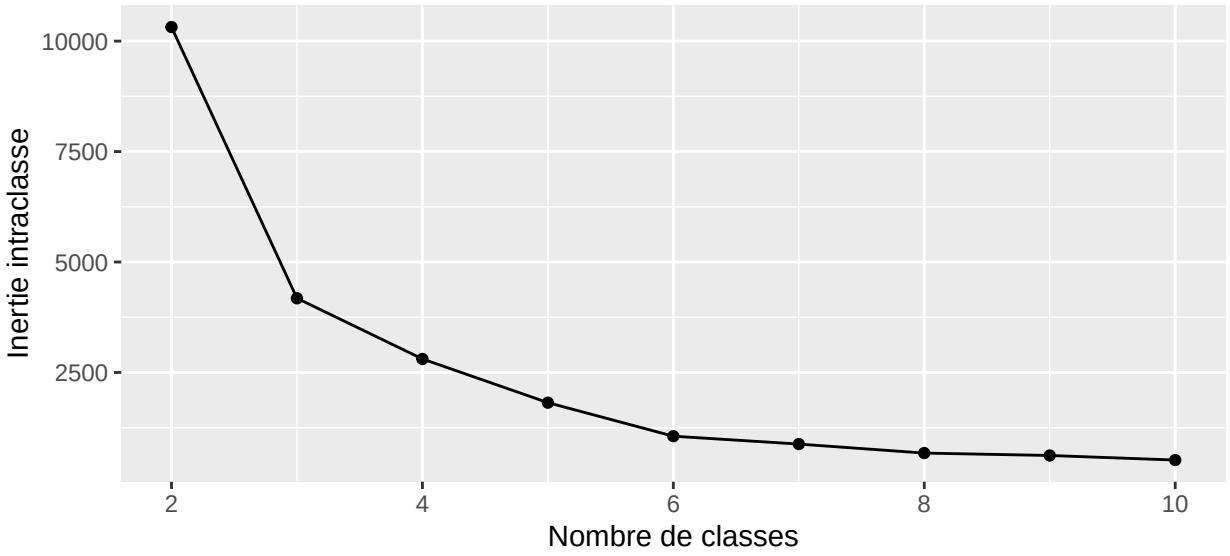


Figure 8: Evolution de l'inertie intraclasse en fonction du nombre de classes

On voit sur la figure 8 qu'il y a un coude pour K = 3. On retient donc 3 classes. Nous avons aussi appliqué sur le Rmd le critère silhouette, qui lui nous as donné un pic à K = 2, soit 2 classes.
On a représenté sur la figure 9 les résultats de kmeans sur les 2 premières composantes principales de l'ACP.

On remarque que le cluster 1 regroupe les traitements T2 et T3 du réplicat R1/R2 à 1h et le traitement T1 du réplicat R1/R2 pour toute les heures. Le cluster 2 regoupe les traitements T2 et T3 de 2h et 3h, Le cluster 3 regroupe quant à lui tous les traitements de 4h à 6h.
On retrouve les mêmes informations qu'avant: le traitement T2 et T3 réagissent de façon similaire. Comme T3 est une combinaison des traitements T1 et T2, on peut dire que c'est T2 qui domine les effets du traitement T3.
Cependant, à l'heure débutante (1h), les traitements T2/T3 ont le même effet que T1 sur les gènes, on peut supposer que l'effet du traitement T2/T3 se déroule graduellement donc n'a pas encore manifesté à 1h.

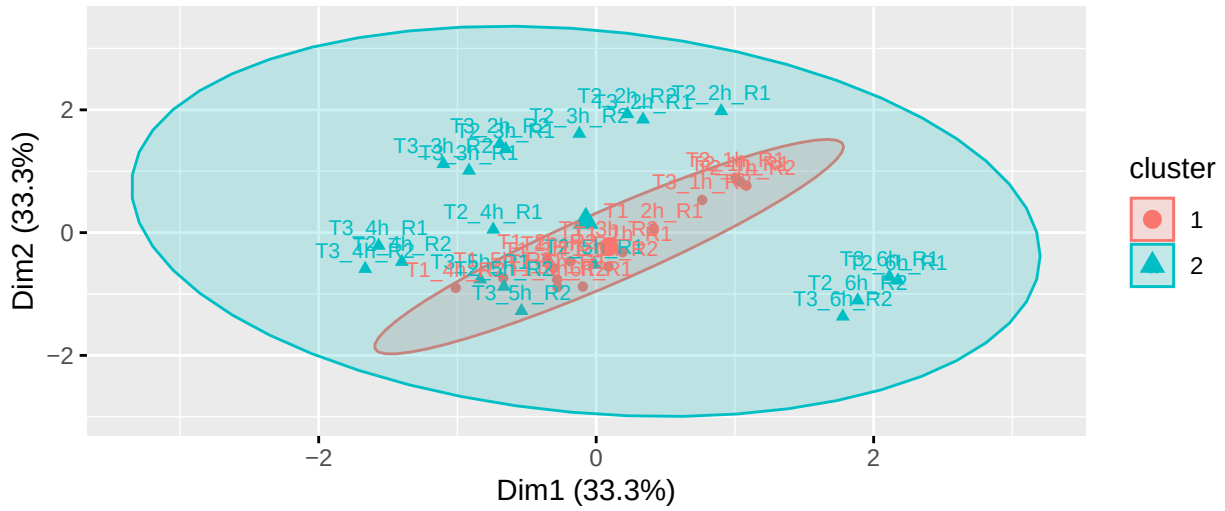


Figure 9: Résultat du clustering Kmeans sur les 2 axes de l'ACP

Effectuons maintenant un diagramme de silhouette afin de déterminer l'homogéité de nos 3 clusters.
On voit sur le diagramme de silhouette figure 10 que les clusters 2 et 3 ont des scores de silhouette inférieurs au cluster 1. La cohésion des points du cluster 1 est donc plus grande. C'est-à-dire que le cluster 1 a moins d'outliers et est plus proches de son centre de gravité que les deux autres clusters. On peut cependant noter que le score de silhouette de chaque cluster est supérieure à 0.5, les partitions de chaque cluster sont globalement homogènes.

```
##   cluster size ave.sil.width
## 1       1   16          0.83
## 2       2   20          0.60
```

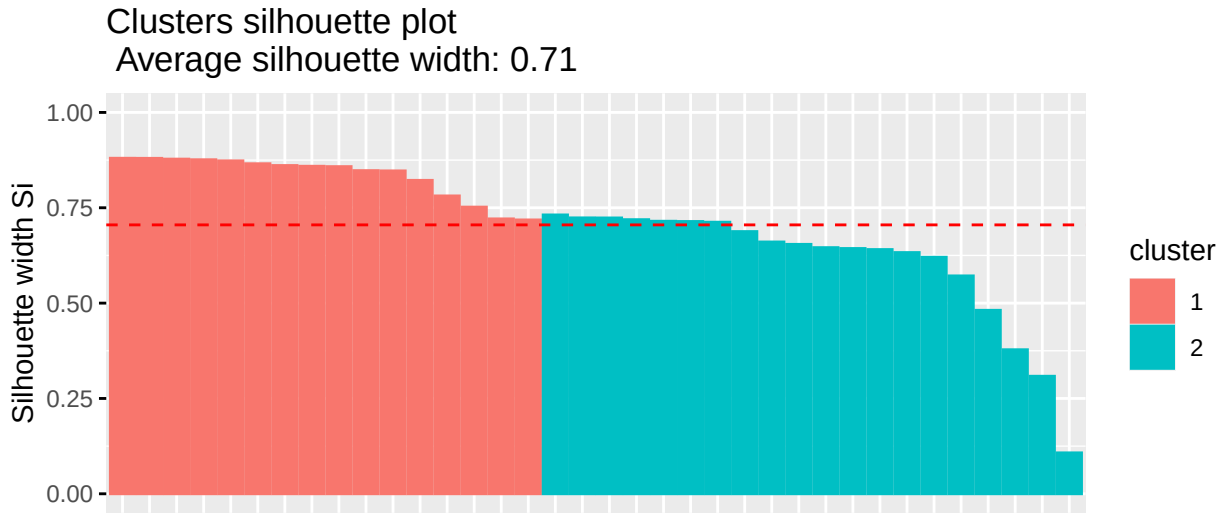
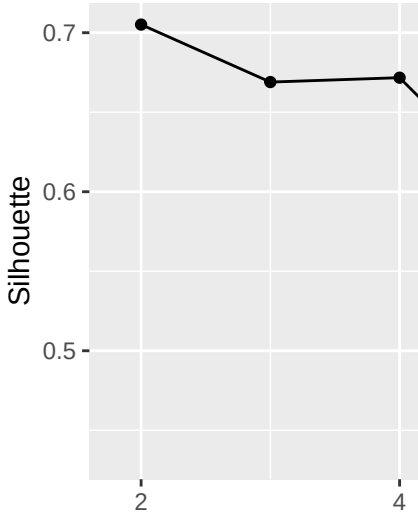


Figure 10: Graphique des silhouettes scores pour nos 3 clusters

3.1.2 PAM



On visualise le nombre de classes optimal par le critère Silhouette avec l'algorithme PAM.
On obtient le même résultat: 3 classes, et ils sont identiques qu'avec k-means.

```
##
##      1  2  3
##   1 16  0  0
##   2  0  8 12
```

3.1.3 Classification hiérarchique

Effectuons maintenant une classification hiérarchique avec la mesure d'agregation de Ward.

On détermine le nombre de classes à retenir avec l'indice de Calinski_Harabasz figure 12. Il y a un pic pour K = 3, on retient donc 3 classes.

```
## KElbowVisualizer(ax=<AxesSubplot: >,
##                  estimator=AgglomerativeClustering(n_clusters=9), k=(2, 10),
##                  metric='calinski_harabasz', timings=False)
```

On trace maintenant la distribution des variables en fonction de la classification en 3 classes avec la mésure d'agrégation Ward.

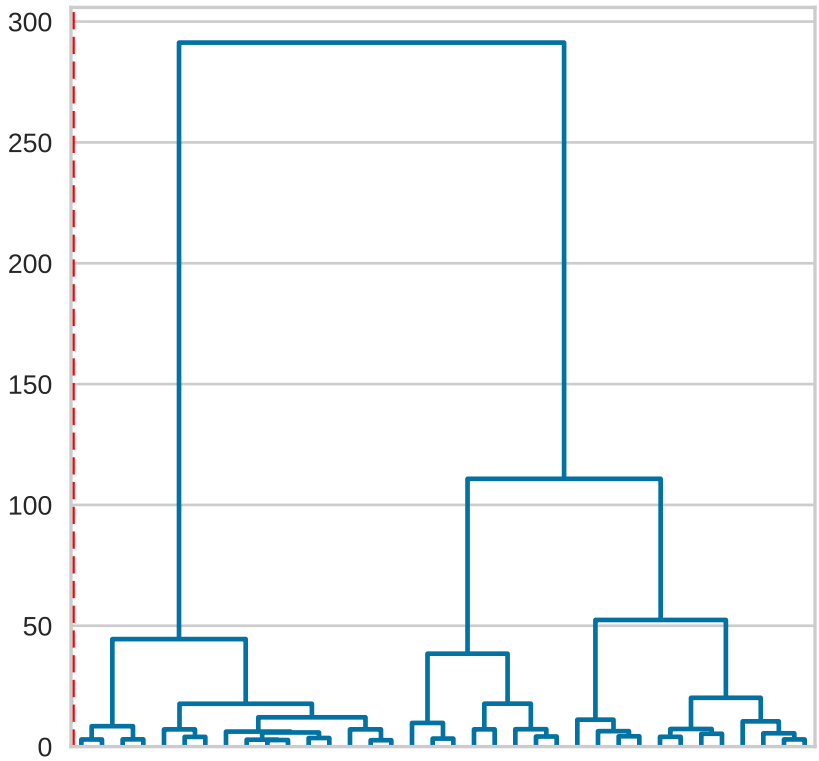


Figure 11: Dendogramme de nos données avec classification hiérarchique avec la mesure d'agrégation de Ward

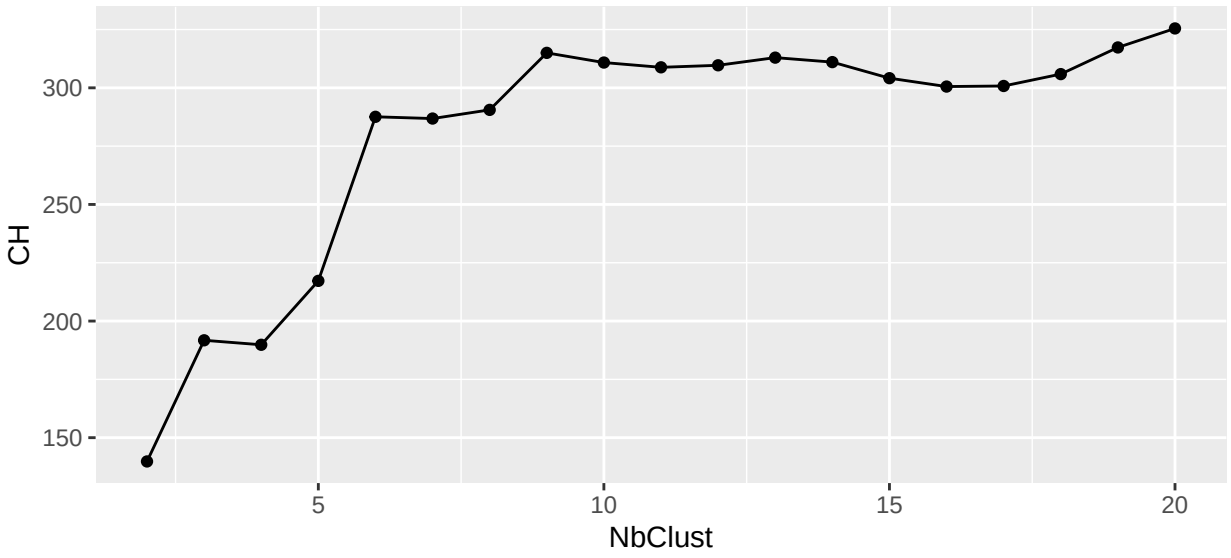


Figure 12: Indice de Calinski-Harabasz en fonction du nombre de classe

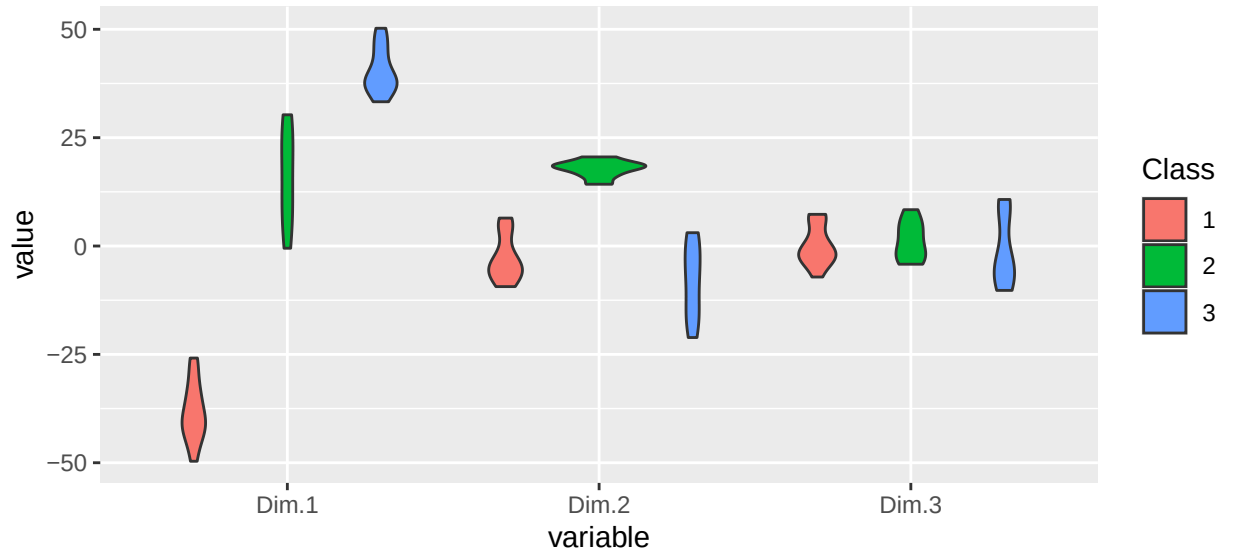


Figure 13: Distribution des variables en 3 classes avec la mesure de Ward

3.1.3.1 Modèle de mélange

BIC : figure 14

```
## Warning: `gather_()` was deprecated in tidyr 1.2.0.
## Please use `gather()` instead.
```

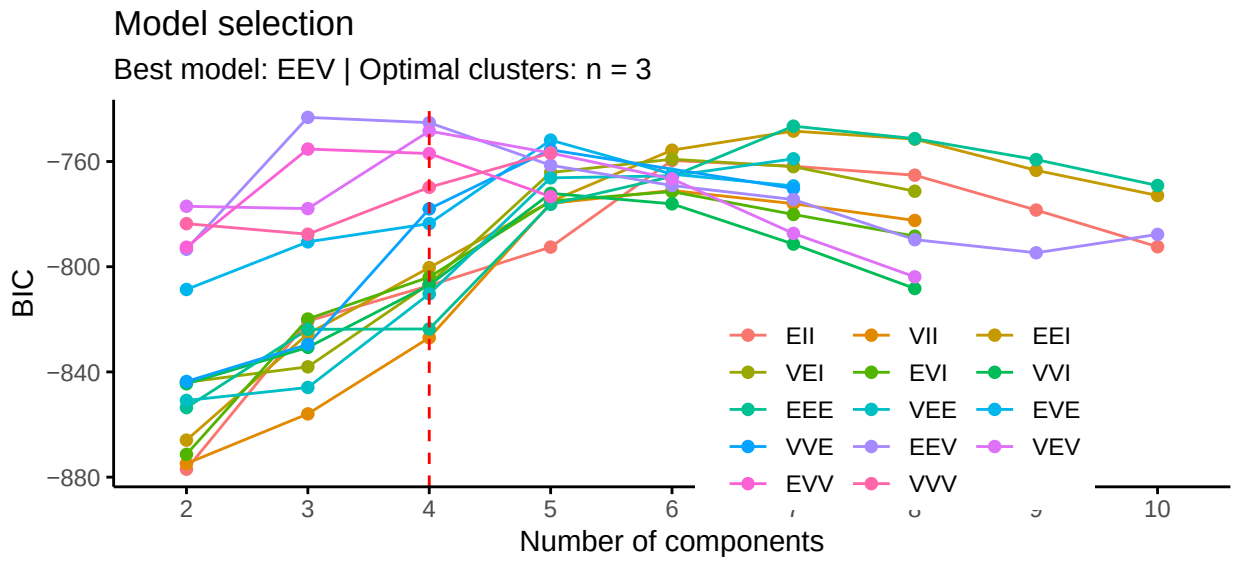


Figure 14: Modèle de mélange BIC

ICL : figure **??**

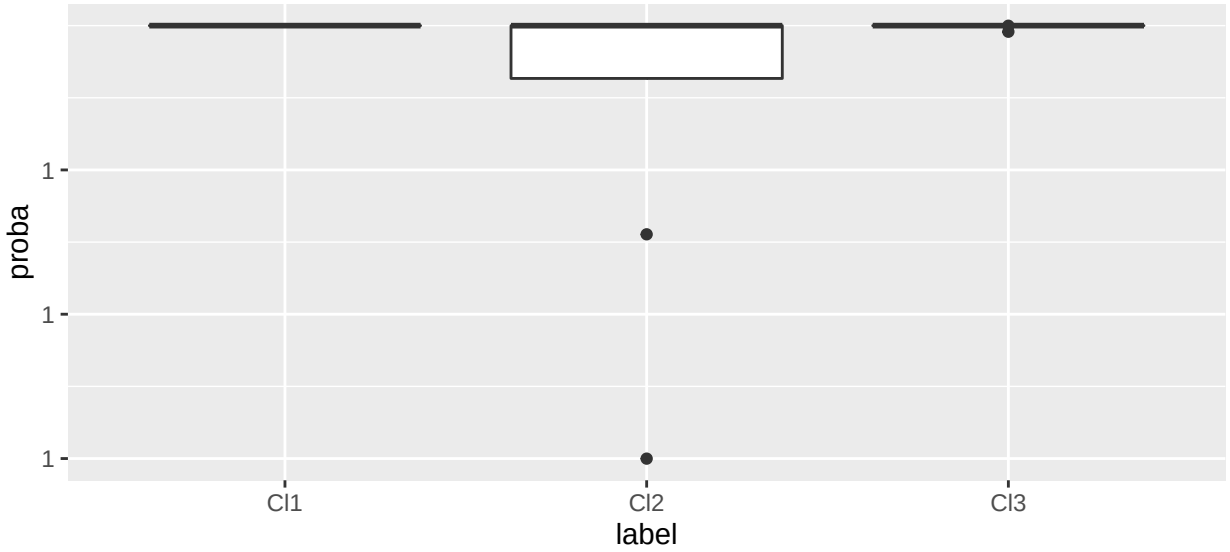
```
## Best ICL values:
##              EEV,3       EEV,4       EEE,7
## ICL      -743.2232 -745.239478 -746.582550
## ICL diff    0.0000   -2.016256   -3.359329
```

Si on retient 3 classes: CAH, modèle de mélange BIC et ICL donnent le même résultat de classification selon ARI = 1:

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

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

On visualise les 3 clusters en boxplot des probabilité d'appartenance et dans le plan de l'ACP:



Tous les individus ont bien été classé dans chacun des classes car les boxplots sont aplati à 1.

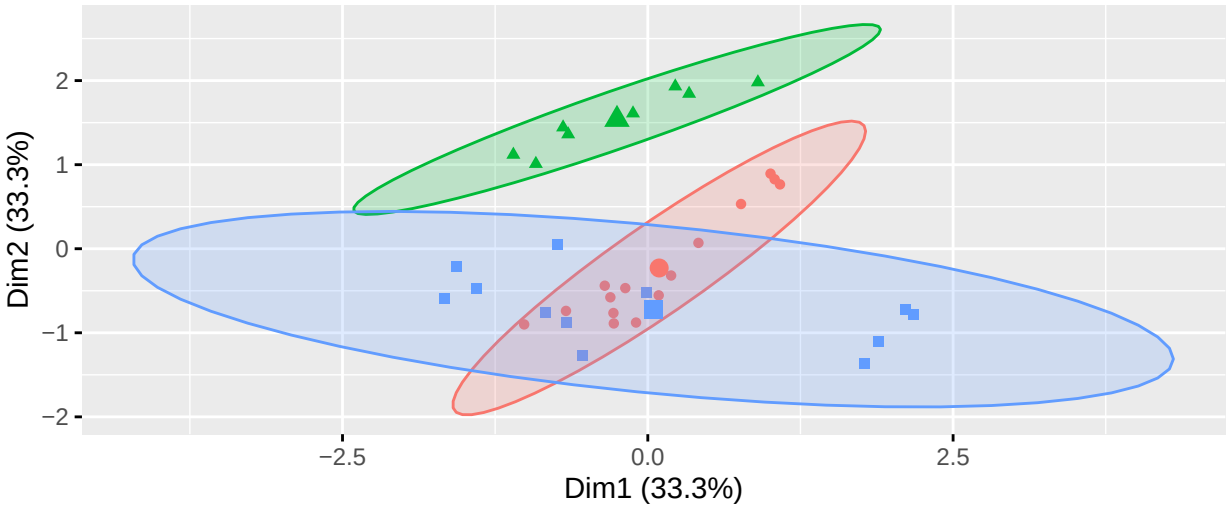
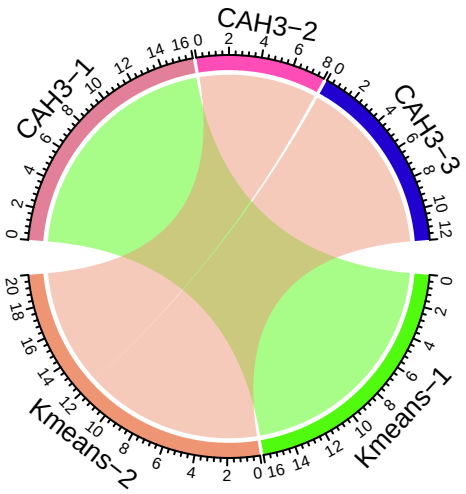


Figure 15: Visualisation des 3 clusters sur le plan de l'ACP

On compare les deux classifications: k = 2 avec kmeans et k=3 avec CAH Ward:

```
##          clust2
```

```
## clust1     CAH3-1 CAH3-2 CAH3-3
##   Kmeans-1     16      0      0
##   Kmeans-2      0      8     12
```



On voit que le cluster 1 de k-means est égale aux cluster 2 et 3 de CAH, qui représente 2h et 3h du traitement T2/T3 du réplicat R1/R2. Les gènes du traitemetns T2/T3 dans les heures aux milieux(2h, 3h) ont donc des comportements différents que l'heure débutante(1h) et heures finales(5h,6h), ce qui correspond à l'analyse précédante.

3.2 Obtention d'une classification des gènes ayant des profils d'expression similaires (co-exprimés) dans les différentes conditions

On travaille avec les données non transposé pour la classification des gènes.

3.2.1 k-means

Implementons l'algorithme K-means.
Commençons par déterminer le nombre de classe optimal. Nous avons implémenté sur la figure 16 le gap statistic en fonction du nombre de classe. On trouve un pic pour K = 2. On choisi donc 2 classes. La méthode silhouette et la méthode des coudes (inertie intraclasse) ont également été implémenté sur le Rmd et nous donnent le même résultat.

```
## Warning: pas de convergence en 10 itérations
```

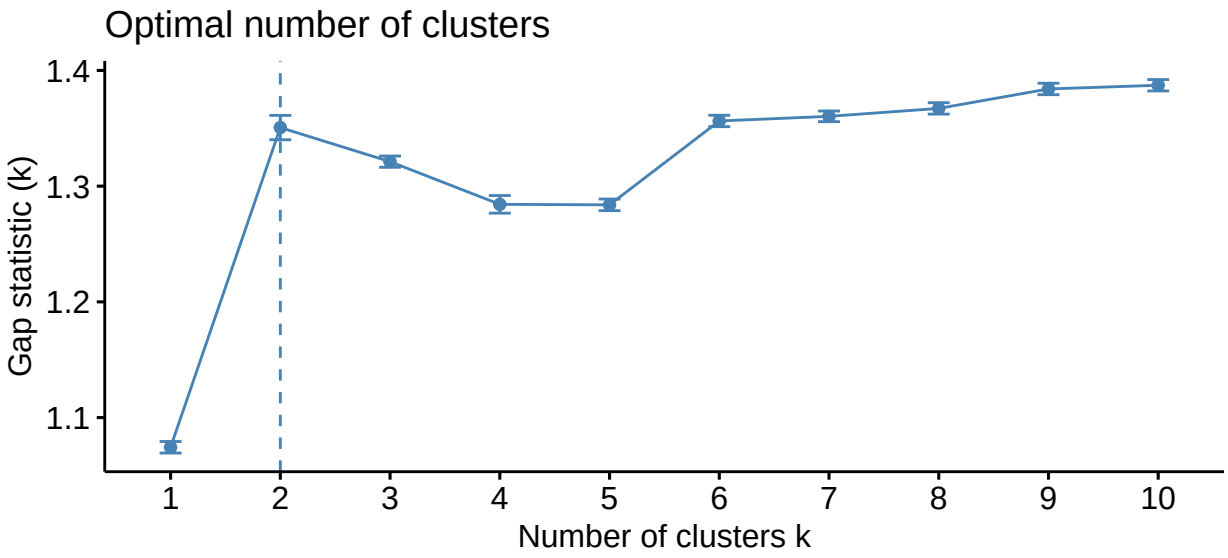
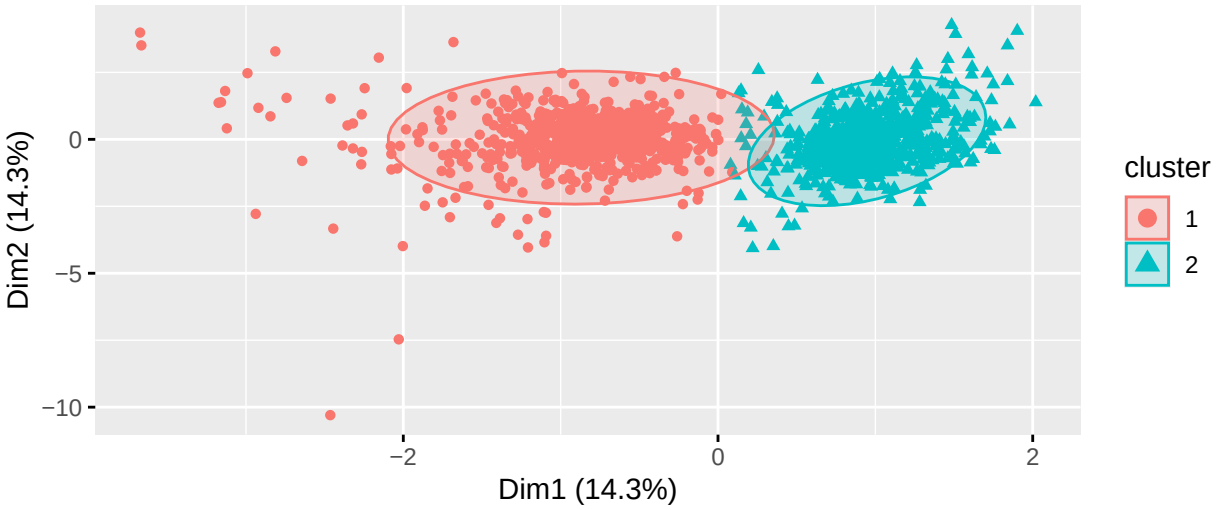
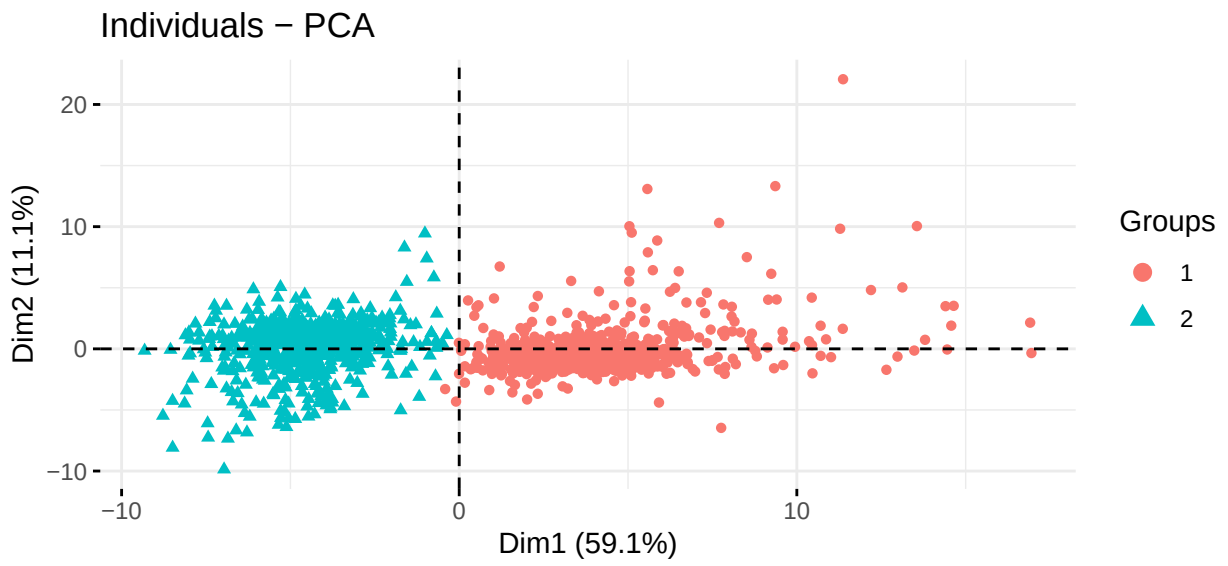


Figure 16: Gap statistique en fonction du nombre de cluster



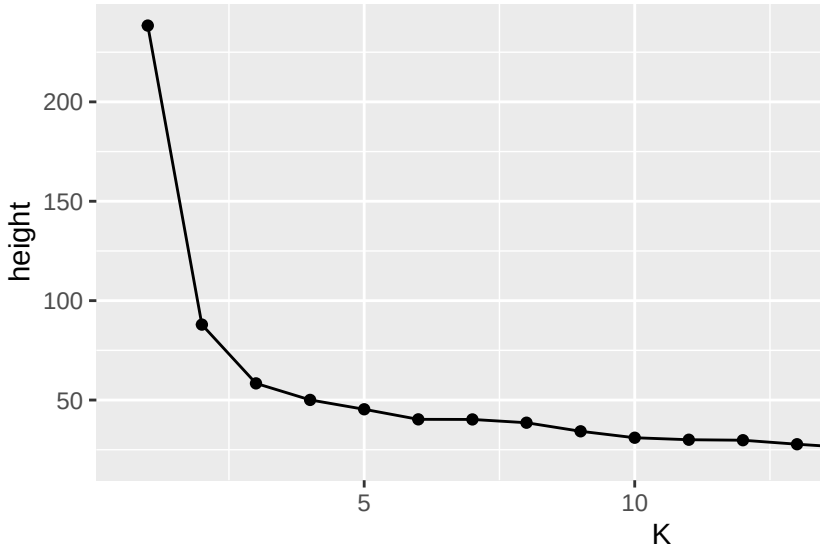


On visualise la classification obtenue sur le plan de l'ACP figure **??**. L'axe 1 de l'ACP sépare bien les deux clusters.

3.2.2 Classification hiérarchique avec mesure de Ward

Effectuons maintenant une classification hiérarchique des gènes avec la mesure d'agregation de Ward.

Afin de déterminer le nombre de classe optimal pour la classification hiérarchique, on trace la hauteur du dendogramme (fig **??**) en fonction du nombre de classe K.



La figure \ref{ref:{height3}) nous donne 2 ou 3 classes. On retrouve le même résultat avec l'indice de Calinski-Harabasz et l'indice Silhouette implémenté sur le Rmd.

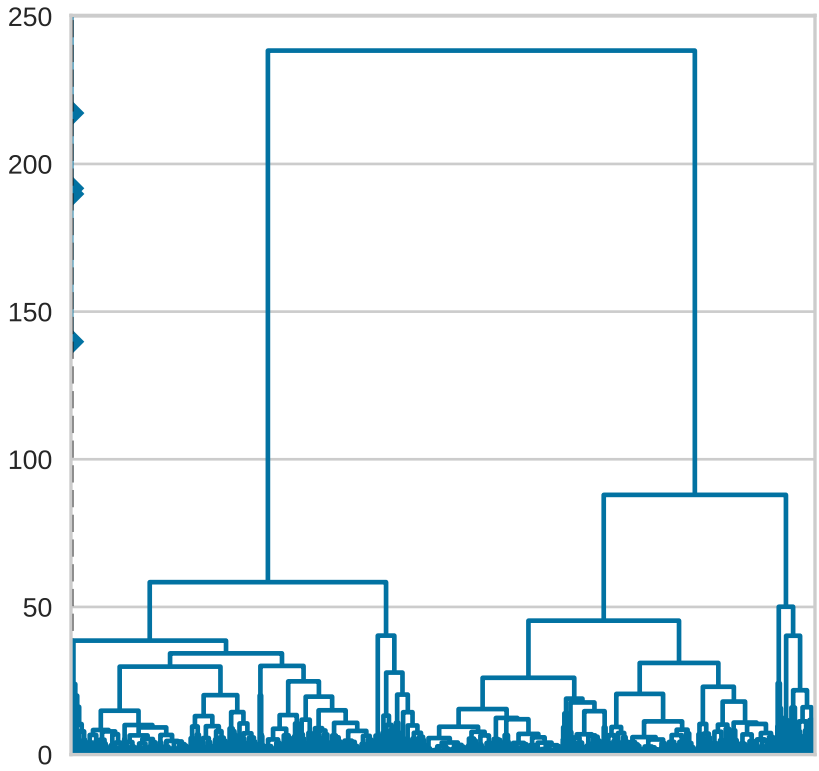
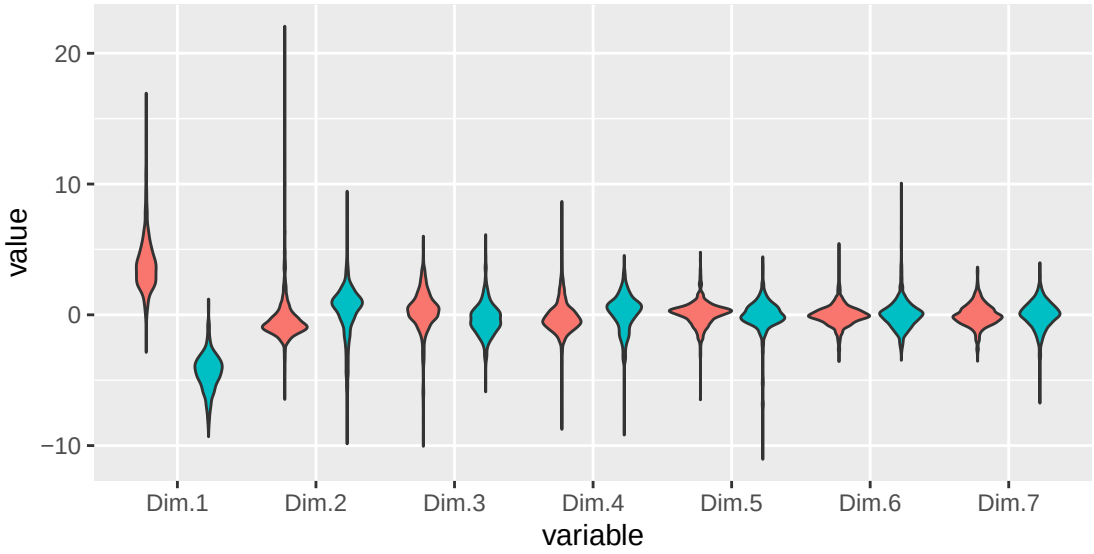
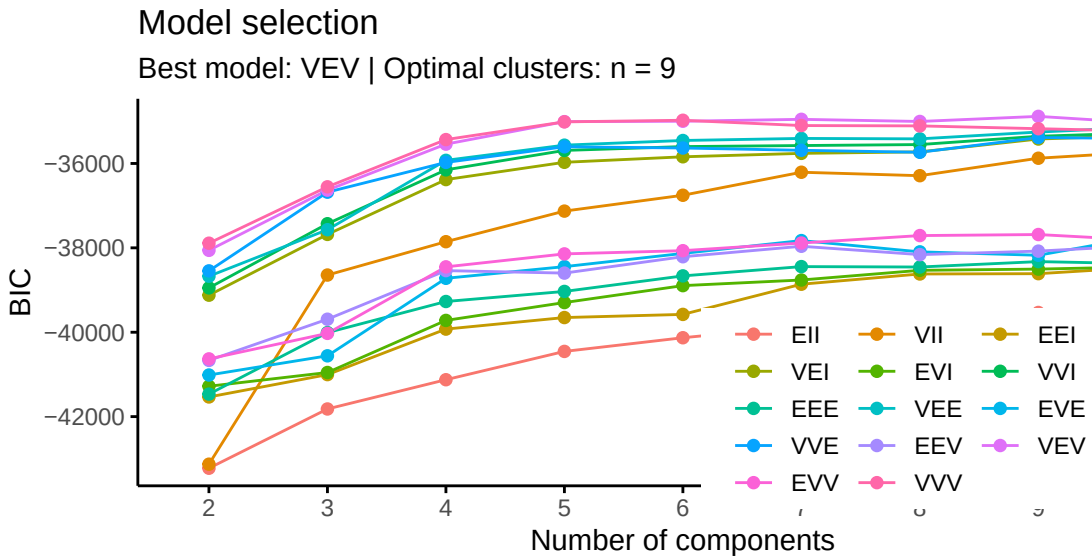


Figure 17: Dendogramme des gènes avec classification hiérarchique avec la mesure d'agrégation de Ward



Visualisation avec 2 classes:

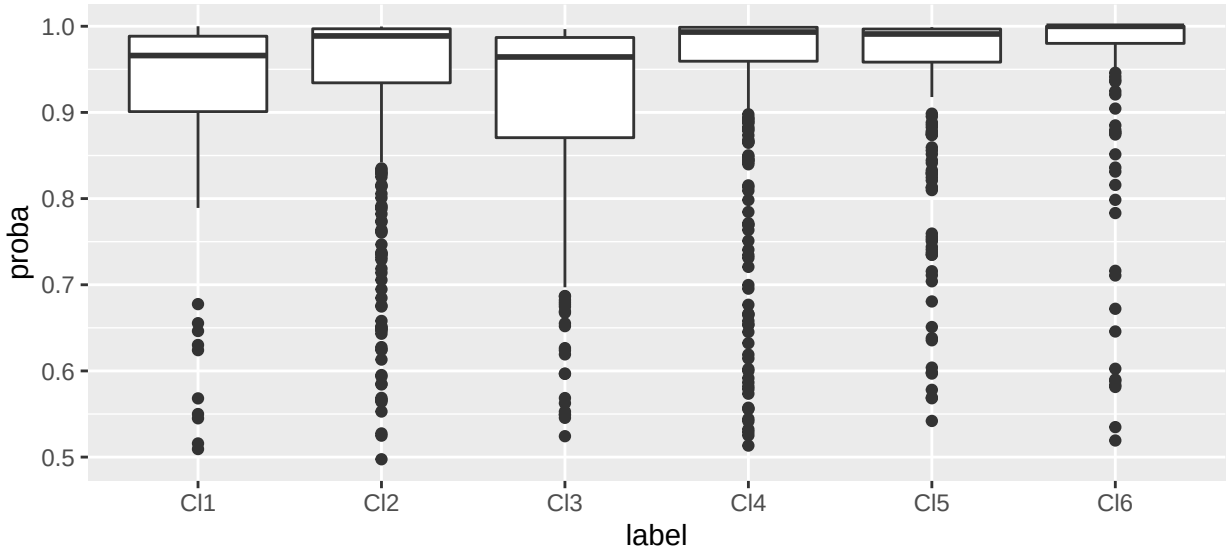
3.2.3 Modèle de mélange



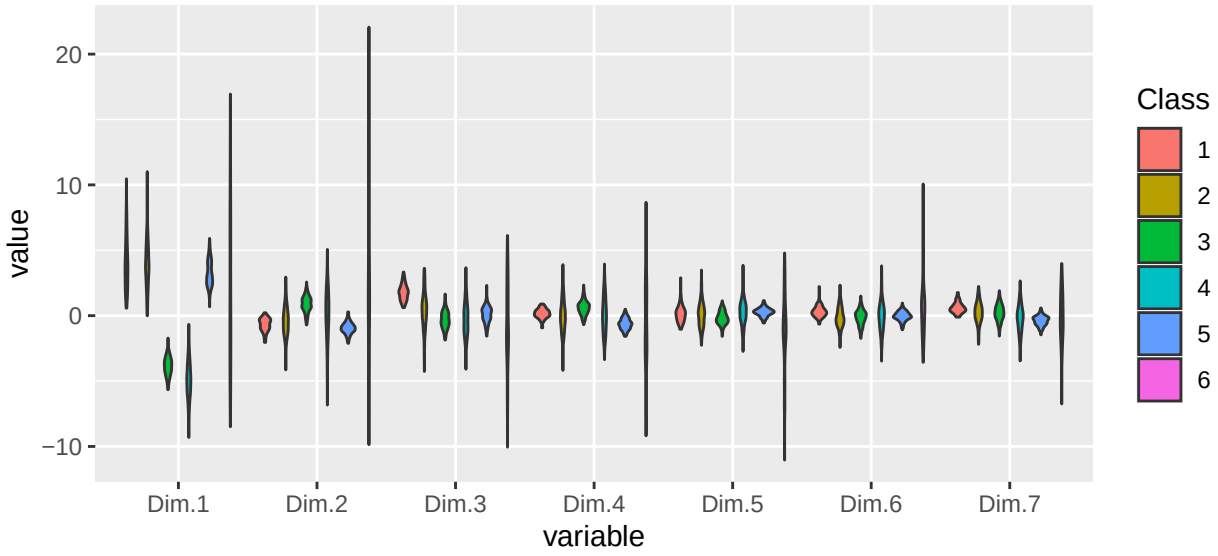
Modèle mélange avec BIC: ??

Modèle mélange avec ICL: ??

```
## Best ICL values:
##              VVV,6        VEV,5       VVV,5
## ICL      -35216.24 -35240.78587 -35244.1317
## ICL diff      0.00    -24.54394    -27.8898
```



La classification n'a pas été bien faite, il y a beaucoup de outliers avec probabilité d'appartenance jusqu'à 50%.



Transformation en data qualitative avec modalités -1, 0 et 1:

D'après analyse descriptive précedante, on sait que T3_6h_R2 possède que des gènes sur-exprimé et sous-exprimé, donc en comparant avec T3_6h_R2, on en déduit que les deux cluters qu'on a obtenu sépare les profils d'expression non-similaires (sur et sous-exprimé), qui veut dire regroupe les profils co-exprimés:

```
##
##      -1   1
##   1   3 839
##   2 761  12
```

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

Nous allons dans cette partie étudier l'expression des gènes pour le traitement T3 à 6h. Nous allons notamment évaluer les temps clés qui influencent l'expression des gènes et étendre cette analyse à tous les traitements et temps. Nous allons également découvrir les facteurs prédictifs qui permettent de distinguer les gènes sur-exprimés et les gènes sous-exprimés pour le traitement T3 à 6 heures.

4.1 Modèle linéaire

Nous allons étudier l'expression des gènes pour le traitement T3 à 6 heures par un modèle linéaire par rapport aux autres heures.

```
##
## Call:
## lm(formula = T3_6h_R2 ~ ., data = T3R2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.6068 -0.3814  0.0046  0.3445  3.7964
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.06441    0.01815  -3.549 0.000397 ***
## T3_1h_R2     0.13580    0.02491   5.451 5.79e-08 ***
## T3_2h_R2    -0.24370    0.03783  -6.441 1.56e-10 ***
## T3_3h_R2     0.18888    0.04027   4.691 2.95e-06 ***
## T3_4h_R2    -0.18577    0.04335  -4.285 1.93e-05 ***
## T3_5h_R2     1.17862    0.02970  39.686  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7061 on 1609 degrees of freedom
## Multiple R-squared:  0.9563, Adjusted R-squared:  0.9561
## F-statistic:  7035 on 5 and 1609 DF,  p-value: < 2.2e-16
```

Pour identifier les temps qui ont une réelle influence sur l'expression des gènes à ce stade nous effectuons une sélection de variable avec différents critères.

On a réalisé notre sélection de variables avec tous les critères (BIC, adjr2, Cp) et avec les méthodes forward et backward. Nous avons eu les mêmes résultats.

On garde toutes les variables mais on observe quand même une gradation. Le temps précédent (5h) est le plus influent suivi du temps de démarrage (1h, 2h). On peut faire l'hypothèse d'une périodicité de temps sur l'influence des traitements sur les gènes. Il faudrait tester cette sélection de variable sur plus d'heures afin valider ou non cette hypothèse.

4.1.1 Etude sur tous les traitements et tous les temps

Réalisons maintenant la même étude mais cette fois ci sur tous les traitements et tous les temps.

D'après la figure 19 (et les autres figures qui ont été réalisé sur le R-markdown), on trouve que :
- On sélectionne les variables suivantes pour T1: 1h, 3h, 5h, 6h, pour T2: 1h, 3h, 5h, 6h et pour T3: 5h.

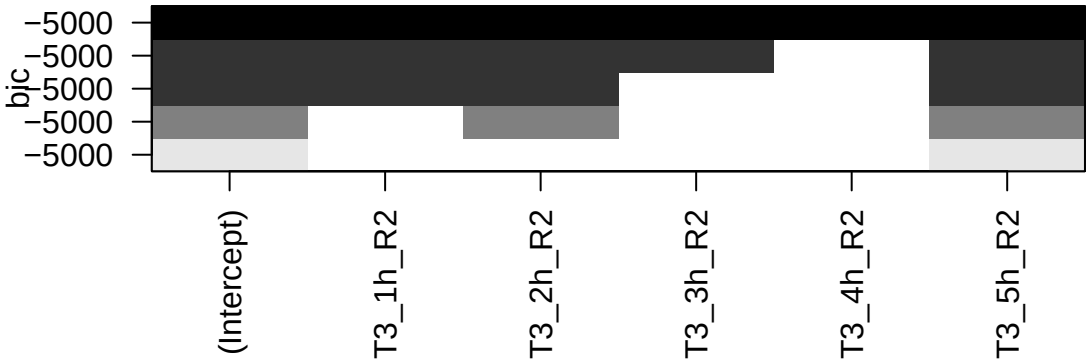


Figure 18: Selection de variable du traitement 3 selon le critère BIC et la méthode backward

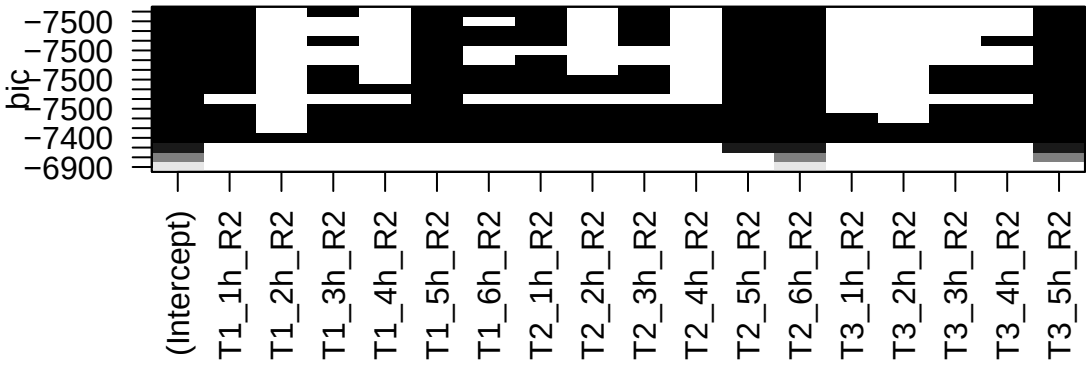


Figure 19: Selection de variable sur tous les traitement selon le critère BIC et la méthode backward

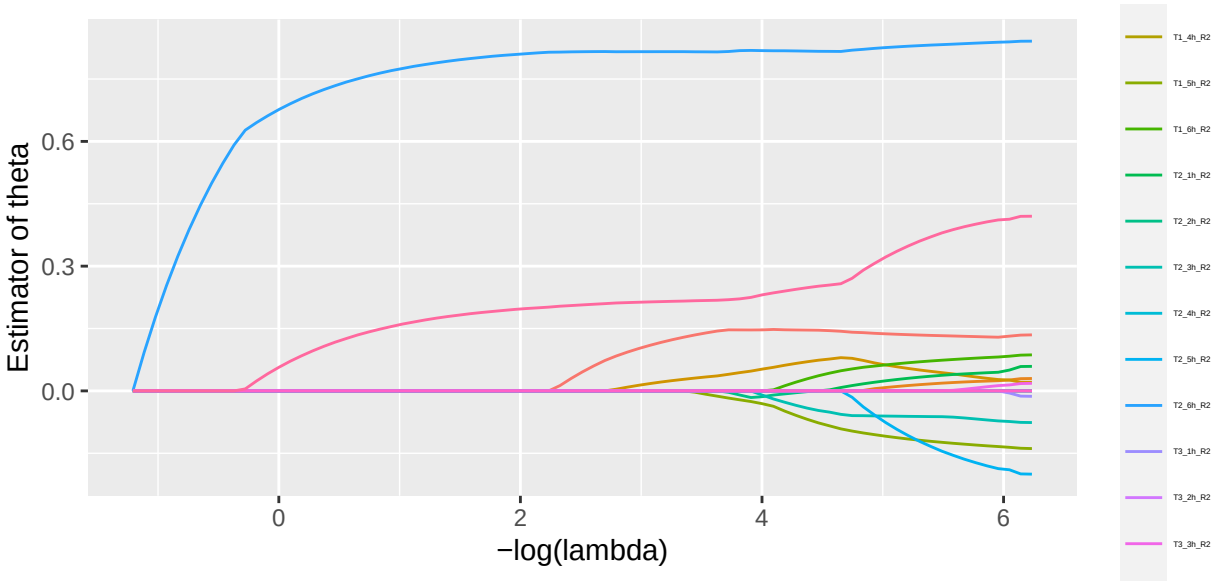
Cela rejoint l'analyse descriptive précedente : les gènes qui ont eu le traitement T2 ou le traitemet T3 ont des comportements similaires.
- On retrouve, par ailleurs, les résultats de l'analyse de la figure 18 puisque les heures les plus influentes sont les heures les plus proches de 6h.

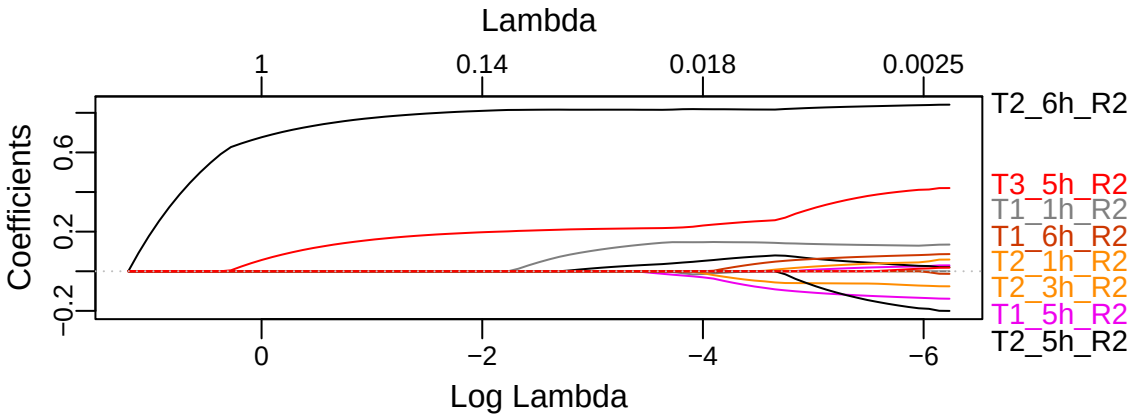
On cherche maintenant à valider ce sous-modèle en comparant avec le modèle de départ :

```
## Analysis of Variance Table
##
## Model 1: T3_6h_R2 ~ T1_1h_R2 + T1_3h_R2 + T1_5h_R2 + T1_6h_R2 + T2_1h_R2 +
##     T2_3h_R2 + T2_5h_R2 + T2_6h_R2 + T3_5h_R2
## Model 2: T3_6h_R2 ~ T1_1h_R2 + T1_2h_R2 + T1_3h_R2 + T1_4h_R2 + T1_5h_R2 +
##     T1_6h_R2 + T2_1h_R2 + T2_2h_R2 + T2_3h_R2 + T2_4h_R2 + T2_5h_R2 +
##     T2_6h_R2 + T3_1h_R2 + T3_2h_R2 + T3_3h_R2 + T3_4h_R2 + T3_5h_R2
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1   1605 169.72
## 2   1597 168.68  8    1.0357 1.2256 0.2798
```

La p-valeur est égale 0.2798 et est supérieure à 0.05, on ne rejette don pas H0 au risque de 5%, on accepte donc le sous modèle.

4.1.2 Lasso





On voit que les variables les plus affectantes sont: -T1: 1h, 3h, 5h -T2: 3h, 4h, 5h, 6h -T3: 5h

4.2 Modèle linéaire généralisé

On veut chercher les variables prédictives qui permettent de discriminer les gènes sur-exprimés (Y>1) des gènes sous-exprimés (Y<-1) à 6h pour le traitement T3.
La sortie est binaire, nous allons donc chercher les variables prédictives par une régression logistique sur le réplicat 2 uniquement (puisque nous avions montré précedemment que le réplicat 1 était similaire en comportement au réplicat 2).

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
##
## Call:
## glm(formula = T36HR2_binomial$T3_6h_R2 ~ ., family = binomial(link = "logit"),
##     data = T36HR2_binomial, control = glm.control(maxit = 100))
##
## Deviance Residuals:
##        Min          1Q      Median          3Q         Max
## -5.840e-06  -2.110e-08   2.110e-08   2.110e-08   5.176e-06
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.334e+00  6.111e+04       0        1
## T1_1h_R2    -4.964e-01  1.913e+05       0        1
## T1_2h_R2    -2.698e+00  1.760e+05       0        1
## T1_3h_R2     5.340e+00  1.298e+05       0        1
## T1_4h_R2     1.060e+00  1.984e+05       0        1
## T1_5h_R2    -1.204e+00  2.134e+05       0        1
## T1_6h_R2    -3.859e-01  1.714e+05       0        1
## T2_1h_R2    -4.752e-01  1.925e+05       0        1
## T2_2h_R2     1.326e+00  1.541e+05       0        1
## T2_3h_R2     2.075e+00  1.655e+05       0        1
## T2_4h_R2    -4.261e+00  1.722e+05       0        1
## T2_5h_R2    -1.334e+00  1.624e+05       0        1
## T2_6h_R2     1.402e+01  9.531e+04       0        1
## T3_1h_R2    -2.748e-01  1.805e+05       0        1
## T3_2h_R2     4.596e-01  2.126e+05       0        1
## T3_3h_R2    -1.538e+00  1.525e+05       0        1
## T3_4h_R2     9.621e-01  1.485e+05       0        1
## T3_5h_R2     3.683e+00  1.932e+05       0        1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2.2342e+03  on 1614  degrees of freedom
## Residual deviance: 4.5579e-10  on 1597  degrees of freedom
## AIC: 36
##
## Number of Fisher Scoring iterations: 29
```

En prenant en compte toutes les variables, le modèle linéaire généralisé n'arrive pas à bien ajuster le modèle.
On remarque que toutes les pvaleurs sont égales à 1.
Ceci est probablement dû au fait que les variables sont très liées les unes aux autres.

Cependant, d'après la table lorsqu'on fait une sélection de variable "backward" sur notre modèle, on obtient que T3_6h_R2 peut s'expliquer par les variables : T1_4h_R2, T1_6h_R2, T2_5h_R2, T3_3h_R2.

Peut importe les combinaisons de traitement qu'on prend en pour expliquer T3_6h_R2, on obtient la même erreur (des pvaleurs toutes égales à 1) sauf lorsqu'on prend seulement le traitement 1.
Dans ce cas, on obtient que T3_6h_R2 s'explique par T1_1h_R2, T1_2h_R2, T1_3h_R2, T1_4h_R2, T1_6h_R2 par une sélection de variable "backward".

```
##
## Call:
## glm(formula = T3_6h_R2 ~ ., family = binomial(link = "logit"),
##     data = T1R2_T3_6h_R2)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.1162  -0.8334   0.3096   0.7980   3.6184
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.04232    0.06495  -0.652    0.515
## T1_1h_R2     1.66907    0.20184   8.269  < 2e-16 ***
## T1_2h_R2    -1.46318    0.20939  -6.988 2.79e-12 ***
## T1_3h_R2     2.20413    0.17324  12.723  < 2e-16 ***
## T1_4h_R2     0.88017    0.18395   4.785 1.71e-06 ***
## T1_5h_R2    -0.04548    0.18239  -0.249    0.803
## T1_6h_R2    -1.17064    0.16260  -7.199 6.05e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2234.2  on 1614  degrees of freedom
## Residual deviance: 1665.4  on 1608  degrees of freedom
## AIC: 1679.4
##
## Number of Fisher Scoring iterations: 5
```

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

Nous allons dans cette partie étudier l'expression des gènes pour le traitement T1 à 6h. Nous allons notamment repérer les temps influent l'expression de ces gènes ainsi que les variables prédictives qui permettent de discriminer les gènes sur-exprimés des gènes sous-exprimés, à 6h pour le traitement T1.

5.1 Modèle linéaire

On a réalisé notre sélection de variables avec tous les critères (BIC, adjr2, Cp) et avec les méthodes forward et backward. Nous avons eu les mêmes résultats :

On garde toutes les variables sauf T1_1h_R2.

On cherche à valider ce sous-modèle :

```
## Analysis of Variance Table
##
## Model 1: T1_6h_R2 ~ T1_2h_R2 + T1_3h_R2 + T1_4h_R2 + T1_5h_R2
```

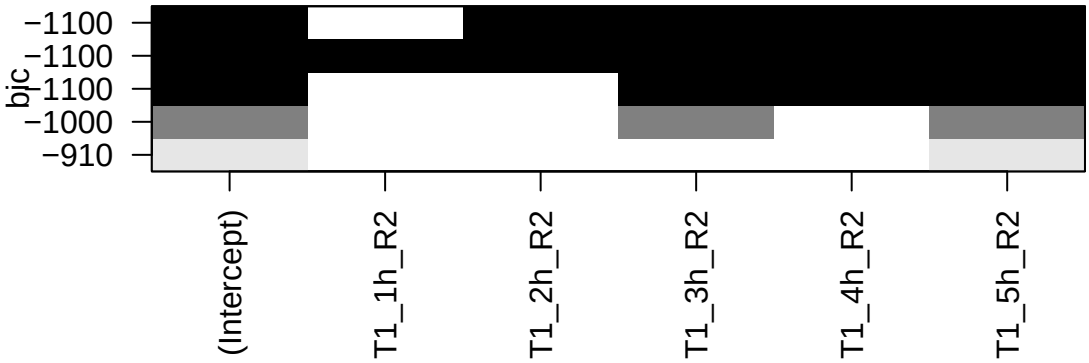


Figure 20: Selection de variable du traitement 1 selon le critère BIC et la méthode backward

```
## Model 2: T1_6h_R2 ~ T1_1h_R2 + T1_2h_R2 + T1_3h_R2 + T1_4h_R2 + T1_5h_R2
##   Res.Df    RSS Df Sum of Sq     F Pr(>F)
## 1   1610 257.34
## 2   1609 257.29  1  0.051971 0.325 0.5687
```

p-valeur $= 0.5687 > 0.05$, on ne rejette pas H0 au risque de 5%, on valide le sous-modèle.

5.1.1 Etude sur tous les traitements et tous les temps

```
choix=regsubsets(T1_6h_R2~., data = R2, nbest = 1, nvmax = 18, method = "backward")
plot(choix,scale = "bic")
```

D'après la figure 21 (et les autres figures qui ont été réalisé sur le R-markdown), on trouve que :
- On sélectionne les variables suivantes pour T1: 1h, 2h, 3h, 4h, 5h, 6h, pour T2: 1h, 2h, 3h, 6h et pour T3: 1h, 5h, 6h.

- On retrouve, par ailleurs, les résultats de l'analyse de la figure 20 puisque les heures les plus influentes sont les heures les plus proches de 6h.

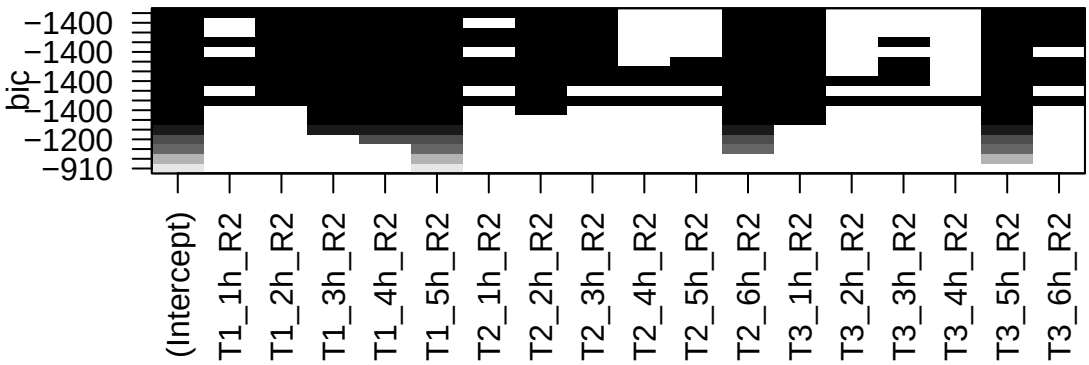


Figure 21: Selection de variable sur tous les traitement selon le critère BIC et la méthode backward

On cherche maintenant à valider ce sous-modèle en comparant avec le modèle de départ :

```
## Analysis of Variance Table
##
## Model 1: T1_6h_R2 ~ T1_1h_R2 + T1_2h_R2 + T1_3h_R2 + T1_4h_R2 + T1_5h_R2 +
##     T2_1h_R2 + T2_2h_R2 + T2_3h_R2 + T2_6h_R2 + T3_1h_R2 + T3_5h_R2 +
##     T3_6h_R2
## Model 2: T1_6h_R2 ~ T1_1h_R2 + T1_2h_R2 + T1_3h_R2 + T1_4h_R2 + T1_5h_R2 +
##     T2_1h_R2 + T2_2h_R2 + T2_3h_R2 + T2_4h_R2 + T2_5h_R2 + T2_6h_R2 +
##     T3_1h_R2 + T3_2h_R2 + T3_3h_R2 + T3_4h_R2 + T3_5h_R2 + T3_6h_R2
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1   1602 207.91
## 2   1597 206.68  5    1.2324 1.9045 0.0906 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

p-valeur = 0.09 > 0.05, on ne rejette pas H0 au risque de 5%, on accepte le sous-modèle.

On voit que l'expression des gènes à 6h pour le traitement T1 est affecté par - les heures finales (3h, 4h, 5h) du traitement T1 - les heures débutantes (1h, 2h, 3h) et finale(6h) du traitements T2 - les heures débutantes (1h) et finales (5h, 6h) du traitement T3.

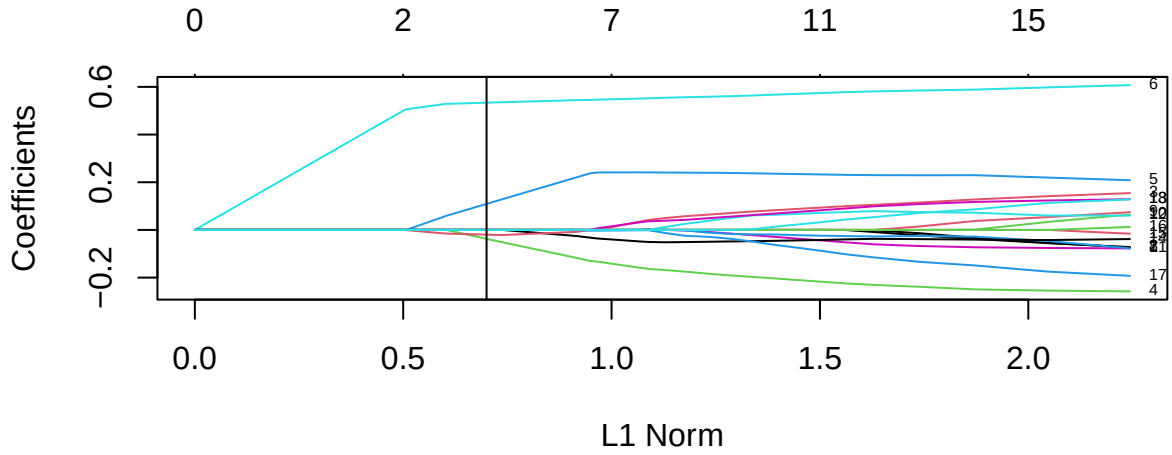
5.1.2 Lasso

```
lambda_seq=seq(0,1,0.001)
x = model.matrix(T1_6h_R2~.,data=R2)
y = data$T1_6h_R2
```

```
fitlasso <- glmnet(x, y , alpha = 1, lambda = lambda_seq, family=c("gaussian"), intercept=F)
plot(fitlasso,label= TRUE)
```

```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values
```

```
abline(v = 0.7)
```



On voit que les variables les plus affectantes sont ??

5.2 Modèle linéaire généralisé

On veut chercher les variables prédictives qui permettent de discriminer les gènes sur-exprimés (Y>1), les gènes sous-exprimés (Y<-1), et les gènes non-exprimés à 6h pour le traitement T1.
La sortie est binaire, nous allons donc chercher les variables prédictives par une régression logistique sur le réplicat 2 uniquement (puisque nous avions montré précedemment que le réplicat 1 était similaire en comportement au réplicat 2).

```
## Call:
## multinom(formula = Y ~ ., data = dfmodel, trace = F)
##
## Coefficients:
##              (Intercept)    T1_1h_R2   T1_2h_R2   T1_3h_R2   T1_4h_R2  T1_5h_R2
## sous-exprime   -4.905461  0.03987164 -0.6813844  0.6556245 -0.5017774 -3.256918
## sur-exprime    -5.280181 -0.81065181  0.4547041 -1.4371988  2.4496719  1.417759
##                T2_1h_R2  T2_2h_R2   T2_3h_R2    T2_4h_R2  T2_5h_R2  T2_6h_R2
## sous-exprime  0.3941555 0.5988196 -0.6684853 -0.09437566 0.4183293 0.2568884
## sur-exprime  -0.3300347 0.4580245 -0.5060857 -0.51678735 1.1800846 0.3912118
##                T3_1h_R2   T3_2h_R2  T3_3h_R2    T3_4h_R2   T3_5h_R2   T3_6h_R2
## sous-exprime -0.7000935 -0.2273308 0.7588703 -0.08713926  0.4014076 -0.9723514
## sur-exprime   0.2728899 -1.0812748 1.0944825  0.15461699 -2.1851137  0.8626062
##
## Std. Errors:
```

```
##              (Intercept)  T1_1h_R2  T1_2h_R2  T1_3h_R2  T1_4h_R2  T1_5h_R2
## sous-exprime   0.3156944 0.3813541 0.4058223 0.3367225 0.3741317 0.3577547
## sur-exprime    0.4140214 0.6247254 0.5856648 0.4807367 0.4765345 0.5190984
##               T2_1h_R2  T2_2h_R2  T2_3h_R2  T2_4h_R2  T2_5h_R2  T2_6h_R2
## sous-exprime 0.3631824 0.4340794 0.4363998 0.3669008 0.4648246 0.4446681
## sur-exprime  0.5459498 0.4953546 0.5387247 0.5351794 0.6314579 0.4908601
##               T3_1h_R2  T3_2h_R2  T3_3h_R2  T3_4h_R2  T3_5h_R2  T3_6h_R2
## sous-exprime 0.3039686 0.4338753 0.4426995 0.4133474 0.4388571 0.4447722
## sur-exprime  0.5180650 0.5888178 0.6086898 0.4801942 0.5948142 0.5228239
##
## Residual Deviance: 699.067
## AIC: 771.067
```

En faisant une sélection de modèle en mode “backward” on peut exprimer T1_6h_R2 par : T1_3h_R2, T1_4h_R2, T1_5h_R2, T2_2h_R2, T2_5h_R2, T3_1h_R2, T3_5h_R2, T3_6h_R2