Script Histology

DIAMBAN Mamadou Lamine

30/11/2019

```
test <- readRDS("data_test_histology.rds")
for ( v in c("sex", "tissue_status", "histology", "os_months", "dead", "dead_at_24_months", "t", "n", "m
    test[[ v ]] <- as.factor(x = test[[ v ]])
}

d <- readRDS("data_learn.rds")

head(d[,1:6])
```

```
##                     age sex tissue_status histology os_months  dead
## TCGA-44-6145-01A 62.32   F       tumoral TCGA-LUAD     19.83 false
## TCGA-97-A4LX-01A 81.96   M       tumoral TCGA-LUAD     20.47 false
## TCGA-44-6148-01A 60.65   M       tumoral TCGA-LUAD     23.47 false
## TCGA-49-6767-01A 46.87   F       tumoral TCGA-LUAD     22.57 false
## TCGA-78-7150-01A 59.98   M       tumoral TCGA-LUAD     22.20  true
## TCGA-67-3772-01A 82.16   F       tumoral TCGA-LUAD     19.10 false
```

Nous disposons d'un jeux de données de 546 individus et 1012 variables. Le but étant de prédire les valeurs histologiques en utilisant les valeurs d'expression génique, les attributs histologiques et cliniques fournis. Pour cela, nous commençons par quelques statistiques descriptives avant de proposer un modèle.

Statistiques descriptives

```
table(d$histology, useNA="ifany")
```

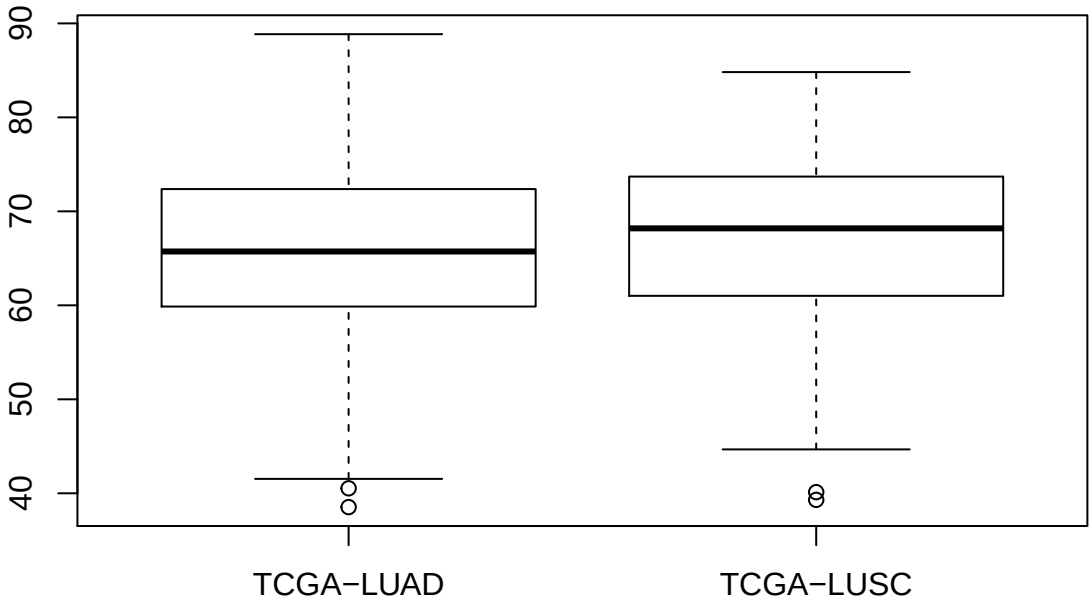
```
##
## TCGA-LUAD TCGA-LUSC
##       273       273
```

La variable histologie contient 2 modalités à part égale: TCGA-LUAD(273) et TCGA-LUSC(273)

```
table(d$sex,d$histology, useNA="ifany")
```

```
##
##     TCGA-LUAD TCGA-LUSC
##   F       152        71
##   M       121       202
```

```r
boxplot(d$age~d$histology)
```



J'ai transformé la variable `histology` en quantitative. La modalité **TCGA-LUAD** est codé en **1** et la modalité **TCGA-LUSC** en **0** puis j'ai sélectionné le nom des gènes.

```r
h <-  as.numeric(d$histology == "TCGA-LUAD")

gs <-  colnames(d)[13:1012]
```

Préparation pour la modélisation

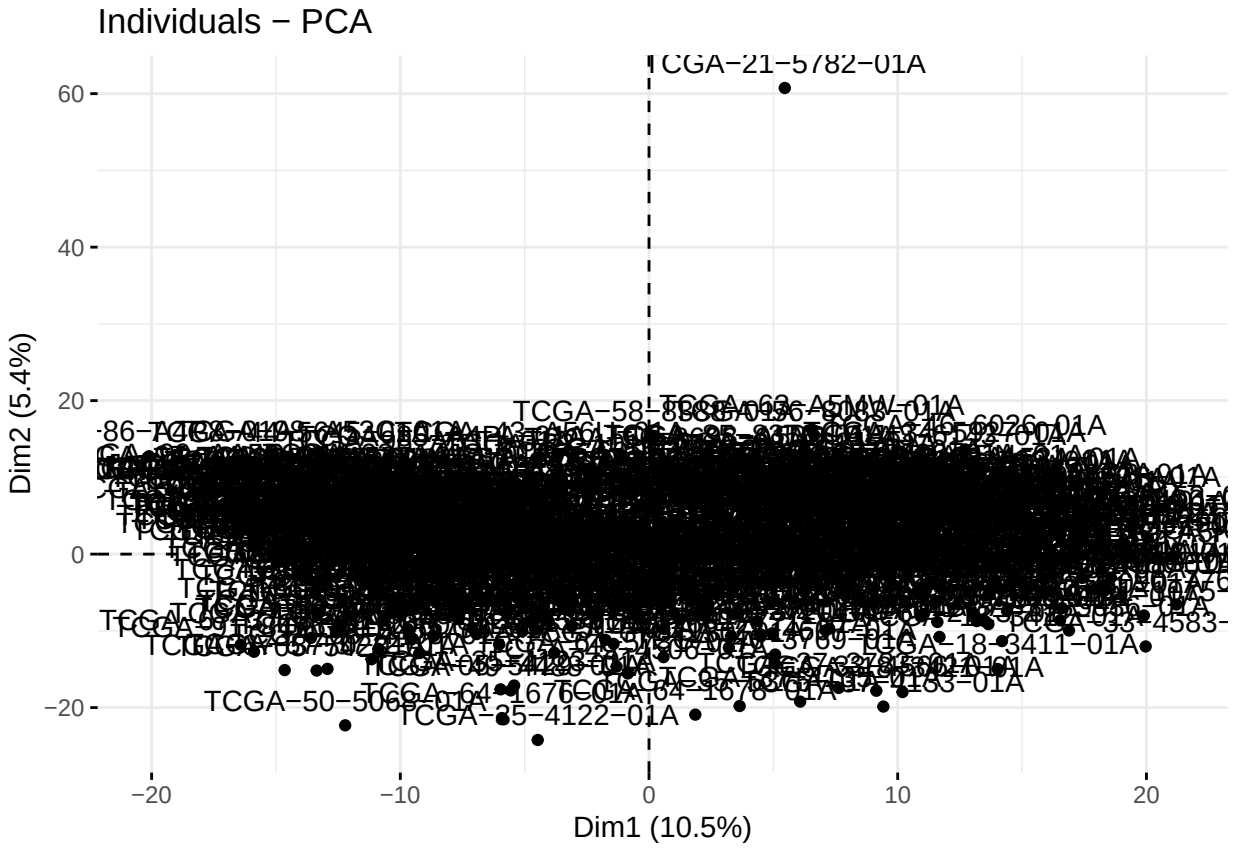
ACP

J'ai réalisé une ACP pour voir s'il y'avait des individus aberrants dans les données

```r
testd <- cbind(age=d$age, sex=d$sex, histology=d$histology, d[, gs])
testd$sex <- as.numeric(testd$sex)
testd$histology <- as.numeric(testd$histology)
library(FactoMineR)
library(factoextra)

res.pca <- PCA(testd, graph = FALSE, scale.unit = TRUE)

fviz_pca_ind(res.pca)
```



On voit que l'individu **TCGA-21-5782-01A** présente des caractéristiques peu communes avec les autres individus.
J'ai jugé alors nécessaire de le retirer du jeux de données.

```
h <- h[-which(rownames(d)=="TCGA-21-5782-01A")]
d <- d[-which(rownames(d)=="TCGA-21-5782-01A"),]
dim(d); length(h)
```

```
## [1]  545 1012
```

```
## [1] 545
```

Calcule des p-valeur de chaque gène ~ histology

J'ai fait un modèle avec chaque gène par rapport à l'histolgie pour récupérer la p-valeur et la valeur de beta.

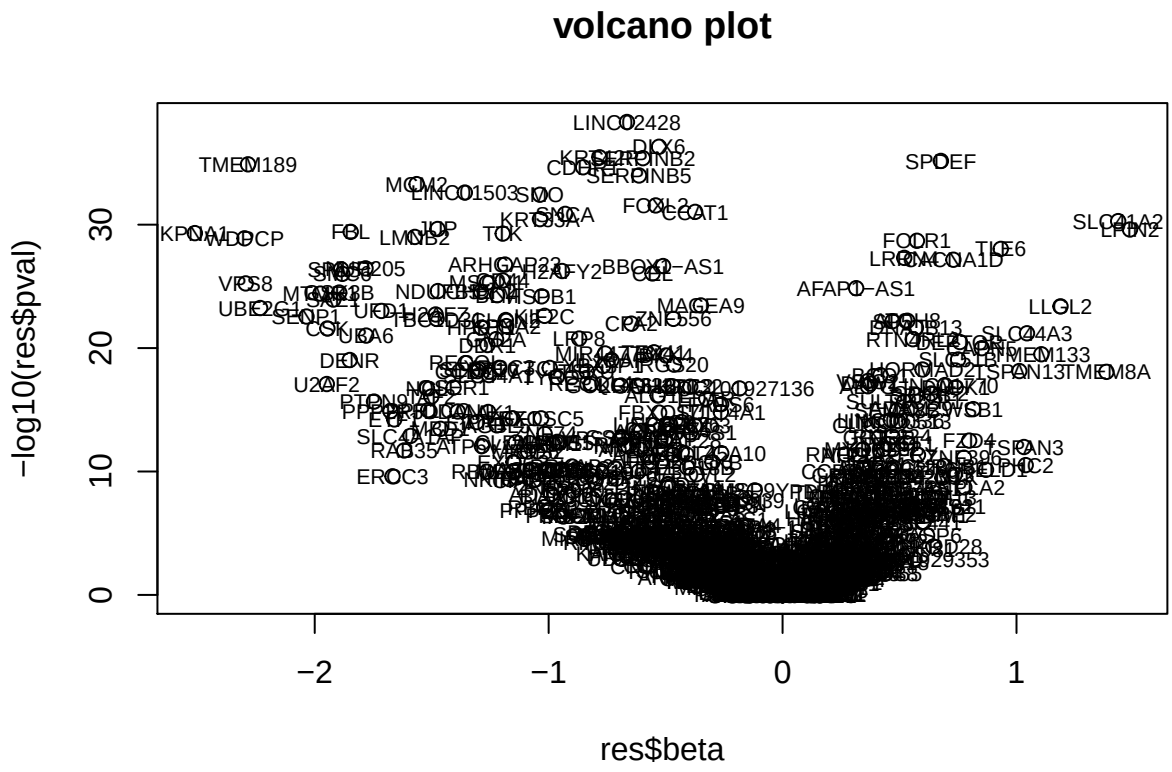
```
res = sapply(gs, function(g) {
  #print(g)
  m = glm(c(h,0,0,1,1)~c(d[[g]],0,max(d[,gs]),0,max(d[,gs])),
    family = binomial(logit))
  b = m$coefficients[[2]]
  pv = summary(m)$coefficients[2,4]
  c(pval = pv,beta = b)
})
```

Volcano plot

Puis, pour pouvoir sélectionner les gènes qui pourraient être adéquats à notre modèle, je me suis basé sur le volcano plot en ne prenant que les variables qui étaient très significatives et les variables qui se trouvaient sur les extrémités.

```
res = t(res)
res = as.data.frame(res)

plot(res$beta, -log10(res$pval), main="volcano plot")
text(res$beta, -log10(res$pval), rownames(res), cex = .7)
```



Modélisation

J'ai réalisé un premier modèle en ne prenant que les gènes dont le log de leur p-valeur est supérieur à 30. Puis j'ai utilisé un step pour trouver le meilleur modèle parmi les gènes choisis.

```
model1 <- glm(
    formula = h ~ LINC02428 + DLX6 + SERPINB2 + SERPINB5 + CDHR1 + KRT33A +
        TMEM189 + MCM2 + SPDEF + LINC01503 + SLC41A2 + SMO + FOXL2 + KRT42P + SNCA
  , data = d
  , family = binomial(link = 'logit')
)

summary(object = model1)
```

```
##
## Call:
## glm(formula = h ~ LINC02428 + DLX6 + SERPINB2 + SERPINB5 + CDHR1 +
```

```
##     KRT33A + TMEM189 + MCM2 + SPDEF + LINC01503 + SLC41A2 + SMO +
##     FOXL2 + KRT42P + SNCA, family = binomial(link = "logit"),
##     data = d)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1690  -0.0256   0.0018   0.0913   4.0046
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) 50.762560  11.799734   4.302 1.69e-05 ***
## LINC02428    0.059030   0.132112   0.447 0.655008
## DLX6         0.101137   0.112153   0.902 0.367175
## SERPINB2    -0.294767   0.116936  -2.521 0.011711 *
## SERPINB5    -0.260354   0.102167  -2.548 0.010824 *
## CDHR1       -0.744957   0.200051  -3.724 0.000196 ***
## KRT33A       0.643204   0.196101   3.280 0.001038 **
## TMEM189     -1.514472   0.550635  -2.750 0.005952 **
## MCM2        -1.367364   0.410264  -3.333 0.000859 ***
## SPDEF        0.453069   0.135777   3.337 0.000847 ***
## LINC01503   -0.858652   0.257461  -3.335 0.000853 ***
## SLC41A2     -0.212476   0.349890  -0.607 0.543674
## SMO         -0.084881   0.213658  -0.397 0.691166
## FOXL2       -0.003449   0.128671  -0.027 0.978613
## KRT42P       0.049867   0.171005   0.292 0.770584
## SNCA        -0.586179   0.225044  -2.605 0.009195 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 755.53  on 544  degrees of freedom
## Residual deviance: 106.59  on 529  degrees of freedom
## AIC: 138.59
##
## Number of Fisher Scoring iterations: 9
```

```
step(model1, direction = "backward")
```

```
## Start:  AIC=138.59
## h ~ LINC02428 + DLX6 + SERPINB2 + SERPINB5 + CDHR1 + KRT33A +
##     TMEM189 + MCM2 + SPDEF + LINC01503 + SLC41A2 + SMO + FOXL2 +
##     KRT42P + SNCA
##
##             Df Deviance    AIC
## - FOXL2      1   106.59 136.59
## - KRT42P     1   106.67 136.68
## - SMO        1   106.75 136.75
## - LINC02428  1   106.79 136.79
## - SLC41A2    1   106.96 136.96
## - DLX6       1   107.42 137.42
## <none>           106.59 138.59
## - SERPINB2   1   112.95 142.95
## - SERPINB5   1   114.10 144.10
## - TMEM189    1   114.70 144.70
```

```
## - SNCA       1   115.08 145.08
## - KRT33A     1   118.17 148.17
## - SPDEF      1   120.32 150.32
## - LINC01503  1   120.48 150.48
## - MCM2       1   120.76 150.76
## - CDHR1      1   125.02 155.02
##
## Step:  AIC=136.59
## h ~ LINC02428 + DLX6 + SERPINB2 + SERPINB5 + CDHR1 + KRT33A +
##     TMEM189 + MCM2 + SPDEF + LINC01503 + SLC41A2 + SMO + KRT42P +
##     SNCA
##
##             Df Deviance    AIC
## - KRT42P     1   106.67 134.68
## - SMO        1   106.76 134.76
## - LINC02428  1   106.79 134.79
## - SLC41A2    1   106.97 134.97
## - DLX6       1   107.43 135.43
## <none>           106.59 136.59
## - SERPINB2   1   113.10 141.10
## - SERPINB5   1   114.11 142.11
## - TMEM189    1   114.74 142.74
## - SNCA       1   115.20 143.20
## - KRT33A     1   118.37 146.37
## - LINC01503  1   120.50 148.50
## - SPDEF      1   120.59 148.59
## - MCM2       1   120.82 148.82
## - CDHR1      1   126.31 154.31
##
## Step:  AIC=134.68
## h ~ LINC02428 + DLX6 + SERPINB2 + SERPINB5 + CDHR1 + KRT33A +
##     TMEM189 + MCM2 + SPDEF + LINC01503 + SLC41A2 + SMO + SNCA
##
##             Df Deviance    AIC
## - SMO        1   106.83 132.83
## - LINC02428  1   106.93 132.93
## - SLC41A2    1   107.17 133.17
## - DLX6       1   107.81 133.81
## <none>           106.67 134.68
## - SERPINB2   1   113.42 139.43
## - SERPINB5   1   114.24 140.24
## - TMEM189    1   115.02 141.02
## - SNCA       1   115.66 141.66
## - KRT33A     1   118.94 144.94
## - LINC01503  1   120.63 146.63
## - SPDEF      1   120.99 146.99
## - MCM2       1   122.77 148.76
## - CDHR1      1   126.50 152.50
##
## Step:  AIC=132.83
## h ~ LINC02428 + DLX6 + SERPINB2 + SERPINB5 + CDHR1 + KRT33A +
##     TMEM189 + MCM2 + SPDEF + LINC01503 + SLC41A2 + SNCA
##
##             Df Deviance    AIC
```

```
## - LINC02428  1   107.12 131.12
## - SLC41A2    1   107.22 131.22
## - DLX6       1   107.87 131.87
## <none>           106.83 132.83
## - SERPINB2   1   113.44 137.44
## - SERPINB5   1   114.24 138.24
## - TMEM189    1   115.58 139.57
## - SNCA       1   115.86 139.86
## - KRT33A     1   119.10 143.10
## - SPDEF      1   121.07 145.07
## - LINC01503  1   121.87 145.87
## - MCM2       1   122.77 146.77
## - CDHR1      1   126.55 150.54
##
## Step:  AIC=131.12
## h ~ DLX6 + SERPINB2 + SERPINB5 + CDHR1 + KRT33A + TMEM189 + MCM2 +
##     SPDEF + LINC01503 + SLC41A2 + SNCA
##
##             Df Deviance    AIC
## - SLC41A2    1   107.59 129.59
## - DLX6       1   108.14 130.14
## <none>           107.12 131.12
## - SERPINB2   1   113.73 135.73
## - SERPINB5   1   114.34 136.34
## - TMEM189    1   115.58 137.57
## - SNCA       1   115.95 137.95
## - KRT33A     1   119.37 141.37
## - SPDEF      1   121.07 143.07
## - LINC01503  1   121.89 143.89
## - MCM2       1   122.78 144.78
## - CDHR1      1   126.57 148.57
##
## Step:  AIC=129.59
## h ~ DLX6 + SERPINB2 + SERPINB5 + CDHR1 + KRT33A + TMEM189 + MCM2 +
##     SPDEF + LINC01503 + SNCA
##
##             Df Deviance    AIC
## - DLX6       1   108.39 128.39
## <none>           107.59 129.59
## - SERPINB2   1   113.81 133.81
## - SERPINB5   1   115.55 135.55
## - SNCA       1   116.00 136.00
## - TMEM189    1   116.03 136.03
## - KRT33A     1   120.32 140.32
## - SPDEF      1   121.21 141.21
## - LINC01503  1   121.91 141.91
## - MCM2       1   124.98 144.98
## - CDHR1      1   126.88 146.88
##
## Step:  AIC=128.39
## h ~ SERPINB2 + SERPINB5 + CDHR1 + KRT33A + TMEM189 + MCM2 + SPDEF +
##     LINC01503 + SNCA
##
##             Df Deviance    AIC
```

```
## <none>           108.39 128.39
## - SERPINB2   1   115.56 133.56
## - SNCA       1   116.00 134.00
## - TMEM189    1   116.08 134.08
## - SERPINB5   1   116.20 134.20
## - KRT33A     1   120.89 138.89
## - SPDEF      1   121.57 139.57
## - LINC01503  1   122.56 140.56
## - MCM2       1   125.48 143.48
## - CDHR1      1   127.51 145.51
```

```
##
## Call:  glm(formula = h ~ SERPINB2 + SERPINB5 + CDHR1 + KRT33A + TMEM189 +
##     MCM2 + SPDEF + LINC01503 + SNCA, family = binomial(link = "logit"),
##     data = d)
##
## Coefficients:
## (Intercept)     SERPINB2     SERPINB5        CDHR1       KRT33A
##     42.8870      -0.2721      -0.2485      -0.6742       0.6553
##     TMEM189         MCM2        SPDEF    LINC01503         SNCA
##     -1.3635      -1.1394       0.4237      -0.8314      -0.4964
##
## Degrees of Freedom: 544 Total (i.e. Null);  535 Residual
## Null Deviance:       755.5
## Residual Deviance: 108.4     AIC: 128.4
```

```
model1 <- glm(
    formula = h ~ SERPINB2 + SERPINB5 + CDHR1 + KRT33A + TMEM189 + MCM2 + SPDEF +
    LINC01503 + SNCA
  , data = d
  , family = binomial(link = 'logit')
)

pred <- predict.glm(object = model1, newdata = test, type = "response")
idx  <- pred <= 0.5
pred[  idx ] <- h==1
pred[ !idx ] <- h==0
table(pred, useNA = "ifany")
```

```
## pred
##  0  1
## 52 48
```

J'ai utilisé le test de Howmer-Lemeshow pour déterminer si les probabilités prévues diffèrent des probabilités observées d'une façon que ne prévoit pas la loi binomiale.

```
ResourceSelection::hoslem.test (h, model1$fitted.values)
```

```
##
##  Hosmer and Lemeshow goodness of fit (GOF) test
##
## data:  h, model1$fitted.values
## X-squared = 43.176, df = 8, p-value = 8.14e-07
```

La p-valeur du modèle1 est inférieure au risque $\alpha = .05$. Le modèle n'est donc pas adéquat.

En suite, à partir du précédent modèle, j'ai rajouté des variables qui sont en lien avec l'histology. Puis avec

la fonction `step` j'ai sélectionné le meilleur modèle.

```
model2 <- glm(
    formula = h ~ SERPINB5  + CDHR1 + MCM2 + LINC01503 + TMEM8A + WDPCP + ERCC3 + UBE2G1 + KPNA1
  , data = d
  , family = binomial(link = 'logit')
)

#h ~ SERPINB5  + CDHR1 + TMEM8A + WDPCP + KPNA1
#h ~ SERPINB5  + CDHR1 + MCM2 + LINC01503 + TMEM8A + WDPCP + ERCC3 + UBE2G1 + KPNA1
#step(model2, direction = "backward")
pred <- predict.glm(object = model2, newdata = test, type = "response")
idx  <- pred <= 0.5
pred[  idx ] <- h==1
pred[ !idx ] <- h==0
table(pred, useNA = "ifany")
```

```
## pred
##  0  1
## 50 50
```

```
ResourceSelection::hoslem.test (h, model2$fitted.values)
```

```
##
##  Hosmer and Lemeshow goodness of fit (GOF) test
##
## data:  h, model2$fitted.values
## X-squared = 76.203, df = 8, p-value = 2.831e-13
```

Ce modèle prédit très bien les classes. Mais comme le précédent, il n'est pas adéquat.

```
model3 <- glm(
    formula = h ~ SERPINB5  + CDHR1 + WDPCP + MCM2 + UBE2G1 + TMEM8A
  , data = d
  , family = binomial(link = 'logit')
)

pred <- predict.glm(object = model3, newdata = test, type = "response")
idx  <- pred <= 0.5
pred[  idx ] <- h==1
pred[ !idx ] <- h==0
table(pred, useNA = "ifany")
```

```
## pred
##  0  1
## 51 49
```

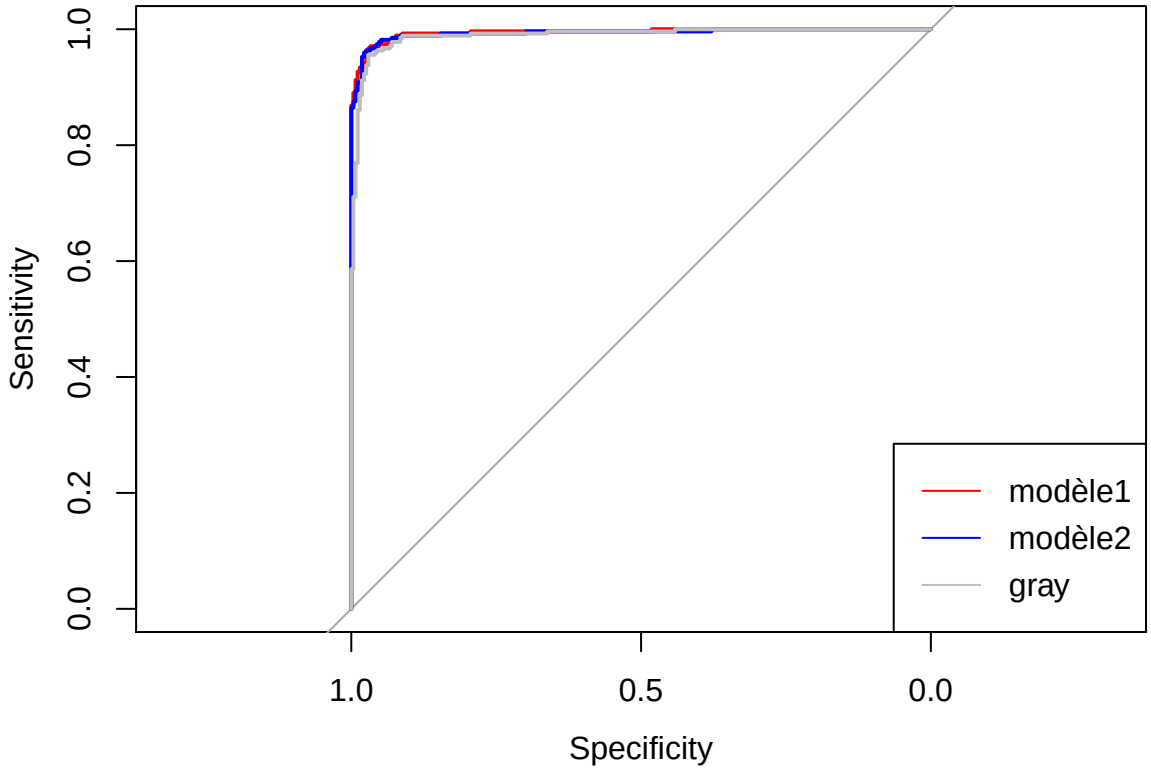
```
ResourceSelection::hoslem.test (h, model3$fitted.values)
```

```
##
##  Hosmer and Lemeshow goodness of fit (GOF) test
##
## data:  h, model3$fitted.values
## X-squared = 5.9097, df = 8, p-value = 0.6574
```

La p-valeur du test de Hosmer-Lemeslow est > 0.05, on consreve H0 et les valeurs prédites et observées concordent bien, le modèle est bon.

Courbe ROC

```
ROC1 = pROC::roc(response=h, model1$fitted.values)
ROC2 = pROC::roc(response=h, model2$fitted.values)
ROC3 = pROC::roc(response=h, model3$fitted.values)
plot(ROC1, xlim=c(1,0), col=2)
lines(ROC2, xlim=c(1,0), col=4)
lines(ROC3, xlim=c(1,0), col="gray")
legend("bottomright", lty=1, c("modèle1", "modèle2", "gray"), col=c("2", "4", "gray"))
```



Le modèle 1 et le modèle 2 ont seniblement le même AUC(proche de 1) et sont globalement meilleur que le modèle 3. Mais comme les premiers sont surement surestimés, j'ai gardé le modèle 3.

Anova entre le modele 3 et le modele 2

```
anova(model3, model2, test = "Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: h ~ SERPINB5 + CDHR1 + WDPCP + MCM2 + UBE2G1 + TMEM8A
## Model 2: h ~ SERPINB5 + CDHR1 + MCM2 + LINC01503 + TMEM8A + WDPCP + ERCC3 +
##     UBE2G1 + KPNA1
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1       538     134.41
## 2       535     111.62  3    22.79 4.466e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Le test est très significatif. Les termes utilisés dans le modèle final sont donc adéquats.

Test de Wald

```
lapply(names(model3$model)[-1], function(x) lmtest::waldtest(model3, x))
```

```
## [[1]]
## Wald test
##
## Model 1: h ~ SERPINB5 + CDHR1 + WDPCP + MCM2 + UBE2G1 + TMEM8A
## Model 2: h ~ CDHR1 + WDPCP + MCM2 + UBE2G1 + TMEM8A
##   Res.Df Df      F    Pr(>F)
## 1    538
## 2    539 -1 25.259 6.836e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## [[2]]
## Wald test
##
## Model 1: h ~ SERPINB5 + CDHR1 + WDPCP + MCM2 + UBE2G1 + TMEM8A
## Model 2: h ~ SERPINB5 + WDPCP + MCM2 + UBE2G1 + TMEM8A
##   Res.Df Df      F    Pr(>F)
## 1    538
## 2    539 -1 27.225 2.589e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## [[3]]
## Wald test
##
## Model 1: h ~ SERPINB5 + CDHR1 + WDPCP + MCM2 + UBE2G1 + TMEM8A
## Model 2: h ~ SERPINB5 + CDHR1 + MCM2 + UBE2G1 + TMEM8A
##   Res.Df Df      F    Pr(>F)
## 1    538
## 2    539 -1 14.967 0.0001228 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## [[4]]
## Wald test
##
## Model 1: h ~ SERPINB5 + CDHR1 + WDPCP + MCM2 + UBE2G1 + TMEM8A
## Model 2: h ~ SERPINB5 + CDHR1 + WDPCP + UBE2G1 + TMEM8A
##   Res.Df Df      F    Pr(>F)
## 1    538
## 2    539 -1 12.357 0.0004763 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## [[5]]
## Wald test
##
```

```
## Model 1: h ~ SERPINB5 + CDHR1 + WDPCP + MCM2 + UBE2G1 + TMEM8A
## Model 2: h ~ SERPINB5 + CDHR1 + WDPCP + MCM2 + TMEM8A
##   Res.Df Df      F Pr(>F)
## 1    538
## 2    539 -1 17.155  4e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## [[6]]
## Wald test
##
## Model 1: h ~ SERPINB5 + CDHR1 + WDPCP + MCM2 + UBE2G1 + TMEM8A
## Model 2: h ~ SERPINB5 + CDHR1 + WDPCP + MCM2 + UBE2G1
##   Res.Df Df      F  Pr(>F)
## 1    538
## 2    539 -1 5.4581 0.01984 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Tous les termes sont significatifs aux seuil $\alpha = .05$

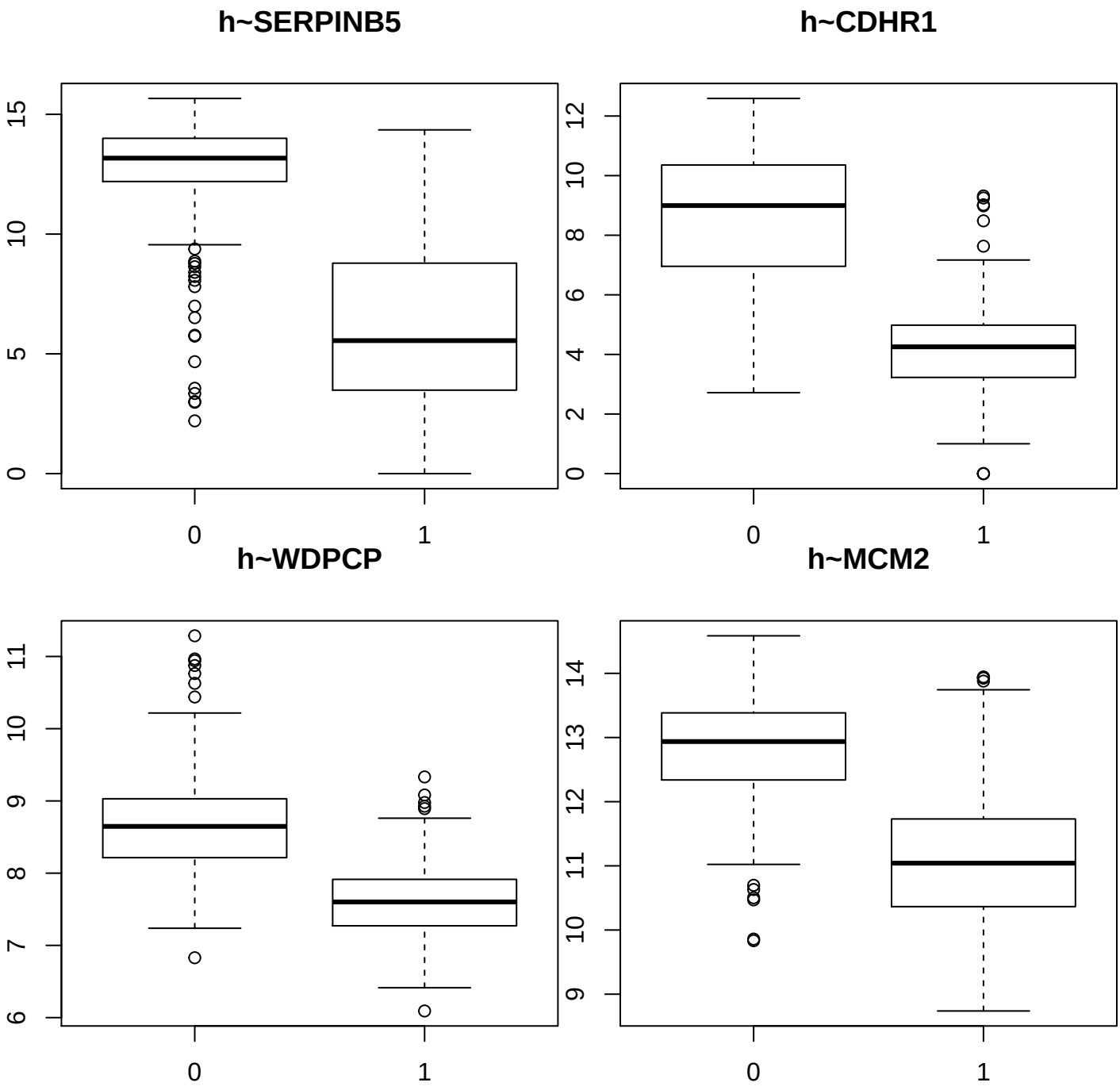
Conclusion

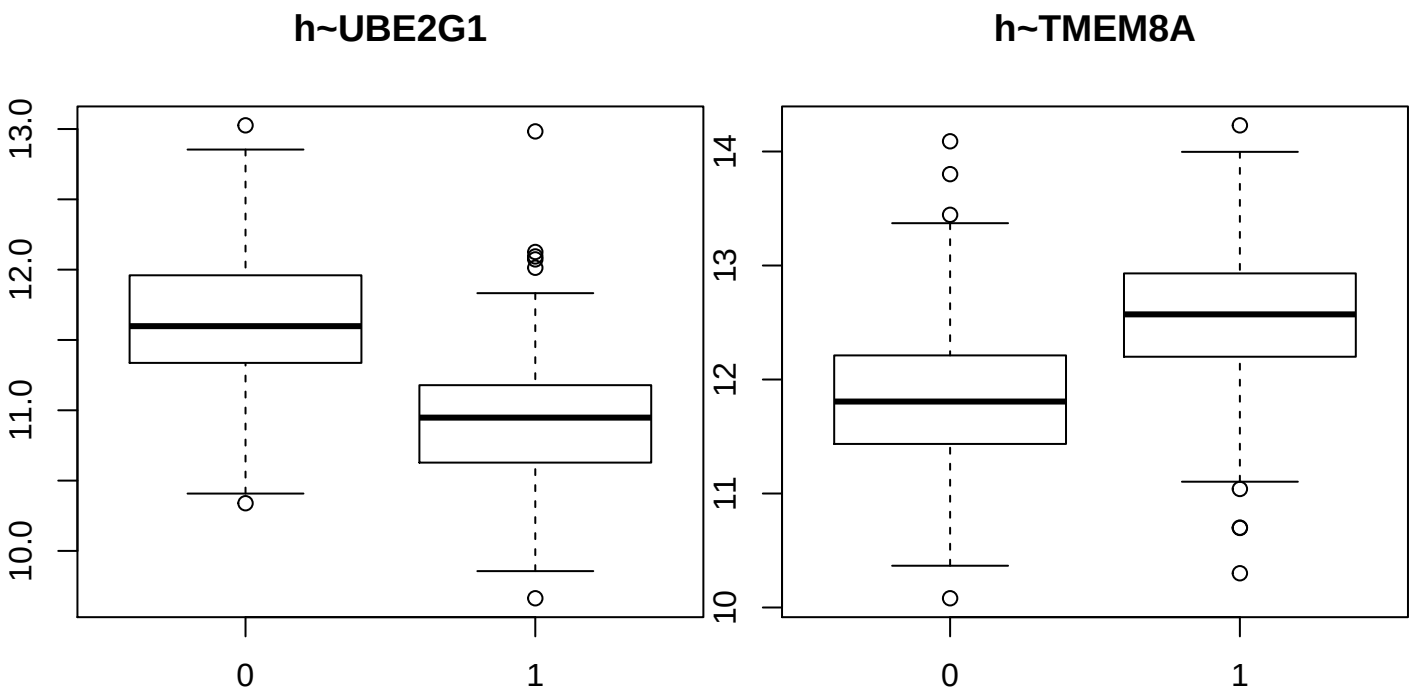
```
summary(object = model3)
```

```
##
## Call:
## glm(formula = h ~ SERPINB5 + CDHR1 + WDPCP + MCM2 + UBE2G1 +
##     TMEM8A, family = binomial(link = "logit"), data = d)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.8660  -0.0696   0.0064   0.1404   3.5890
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) 70.67897   10.95922   6.449 1.12e-10 ***
## SERPINB5    -0.36614    0.07285  -5.026 5.01e-07 ***
## CDHR1       -0.72329    0.13862  -5.218 1.81e-07 ***
## WDPCP       -1.78240    0.46072  -3.869 0.000109 ***
## MCM2        -0.81481    0.23179  -3.515 0.000439 ***
## UBE2G1      -2.50682    0.60524  -4.142 3.45e-05 ***
## TMEM8A      -0.86884    0.37190  -2.336 0.019478 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 755.53  on 544  degrees of freedom
## Residual deviance: 134.41  on 538  degrees of freedom
## AIC: 148.41
##
## Number of Fisher Scoring iterations: 8
```

Tous les termes de notre modèle sont significatifs. Les gènes qui influencent le plus l'histology sont des gènes à codage de protéine. Notamment les gène SERPINB5, CDHR1 et MCM2.
D'autre part ces gènes contribuent de manière négative au cancer du poumon. Plus les individus possédent ces gènes moins ils ont de chance d'avoir ce type de cancer.

```
layout(matrix(1, 1), respect=TRUE)
lapply(names(model3$model)[-1], function(x) boxplot(d[[x]]~h, main = paste0("h~", x)))
```






```
## [[1]]
## [[1]]$stats
##           [,1]      [,2]
## [1,]  9.555734  0.000000
## [2,] 12.195348  3.485071
## [3,] 13.170795  5.552120
## [4,] 13.997947  8.786891
## [5,] 15.661810 14.350780
##
## [[1]]$n
## [1] 272 273
##
## [[1]]$conf
##          [,1]     [,2]
## [1,] 12.99810 5.045129
## [2,] 13.34349 6.059112
##
## [[1]]$out
##  [1] 3.346046 8.069036 3.564086 6.509203 8.775252 6.994055 3.007708
##  [8] 5.739543 8.869054 2.202656 5.776971 7.806473 9.378726 4.672918
## [15] 8.402219 8.230399 8.637738 2.977542
##
## [[1]]$group
##  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##
## [[1]]$names
## [1] "0" "1"
##
##
## [[2]]
## [[2]]$stats
##           [,1]     [,2]
```

```
## [1,]  2.717064 1.002465
## [2,]  6.956431 3.228137
## [3,]  8.995179 4.255055
## [4,] 10.357149 4.981630
## [5,] 12.589091 7.167783
##
## [[2]]$n
## [1] 272 273
##
## [[2]]$conf
##          [,1]     [,2]
## [1,] 8.669385 4.087375
## [2,] 9.320973 4.422734
##
## [[2]]$out
## [1] 8.482416 8.987066 7.631785 9.249732 0.000000 9.020470 9.318075 0.000000
## [9] 0.000000
##
## [[2]]$group
## [1] 2 2 2 2 2 2 2 2 2
##
## [[2]]$names
## [1] "0" "1"
##
##
## [[3]]
## [[3]]$stats
##           [,1]     [,2]
## [1,]  7.237395 6.413256
## [2,]  8.214805 7.271239
## [3,]  8.647655 7.601733
## [4,]  9.029793 7.914088
## [5,] 10.217009 8.760857
##
## [[3]]$n
## [1] 272 273
##
## [[3]]$conf
##          [,1]     [,2]
## [1,] 8.569578 7.540259
## [2,] 8.725732 7.663206
##
## [[3]]$out
##  [1] 10.941736 10.763132 10.965402 11.286029 10.438810 10.627921  6.829012
##  [8] 10.874295  8.893284  6.092205  8.978071  9.332314  9.082405  8.926535
##
## [[3]]$group
##  [1] 1 1 1 1 1 1 1 1 2 2 2 2 2 2
##
## [[3]]$names
## [1] "0" "1"
##
##
## [[4]]
```

```
## [[4]]$stats
##          [,1]     [,2]
## [1,] 11.02246  8.73824
## [2,] 12.33860 10.36435
## [3,] 12.93704 11.04175
## [4,] 13.38409 11.73067
## [5,] 14.58506 13.74419
##
## [[4]]$n
## [1] 272 273
##
## [[4]]$conf
##          [,1]     [,2]
## [1,] 12.83688 10.91109
## [2,] 13.03720 11.17241
##
## [[4]]$out
##  [1] 10.502463  9.832927 10.696931  9.858491 10.631026 10.469600 13.925154
##  [8] 13.943915 13.934151 13.878847
##
## [[4]]$group
##  [1] 1 1 1 1 1 1 2 2 2 2
##
## [[4]]$names
## [1] "0" "1"
##
##
## [[5]]
## [[5]]$stats
##          [,1]      [,2]
## [1,] 10.40776  9.856332
## [2,] 11.33685 10.627938
## [3,] 11.59830 10.947742
## [4,] 11.95980 11.178612
## [5,] 12.85386 11.832894
##
## [[5]]$n
## [1] 272 273
##
## [[5]]$conf
##          [,1]     [,2]
## [1,] 11.53862 10.89508
## [2,] 11.65798 11.00040
##
## [[5]]$out
## [1] 10.33902 13.02607  9.66333 12.12550 12.09393 12.07286 12.98304 12.01360
##
## [[5]]$group
## [1] 1 1 2 2 2 2 2 2
##
## [[5]]$names
## [1] "0" "1"
##
##
```

```
## [[6]]
## [[6]]$stats
##          [,1]     [,2]
## [1,] 10.36644 11.10359
## [2,] 11.43473 12.19967
## [3,] 11.80677 12.57143
## [4,] 12.21151 12.93074
## [5,] 13.37134 13.99707
##
## [[6]]$n
## [1] 272 273
##
## [[6]]$conf
##          [,1]     [,2]
## [1,] 11.73235 12.50152
## [2,] 11.88118 12.64134
##
## [[6]]$out
## [1] 10.08110 13.44467 13.80153 14.08964 10.69802 10.70017 10.30056 14.22952
## [9] 11.03991
##
## [[6]]$group
## [1] 1 1 1 1 2 2 2 2 2
##
## [[6]]$names
## [1] "0" "1"
```