

# Données breast tumors

Julien JACQUES

18 September, 2023

```
library(mixOmics)
data(breast.tumors)
X=breast.tumors$gene.exp
colnames(X)=breast.tumors$genes$name
Y=as.factor(breast.tumors$sample$treatment)
```

On va chercher à partir des niveaux d'expression de gènes à prédire si l'échantillon a été mesuré avant (BE) ou après (AF) le traitement.

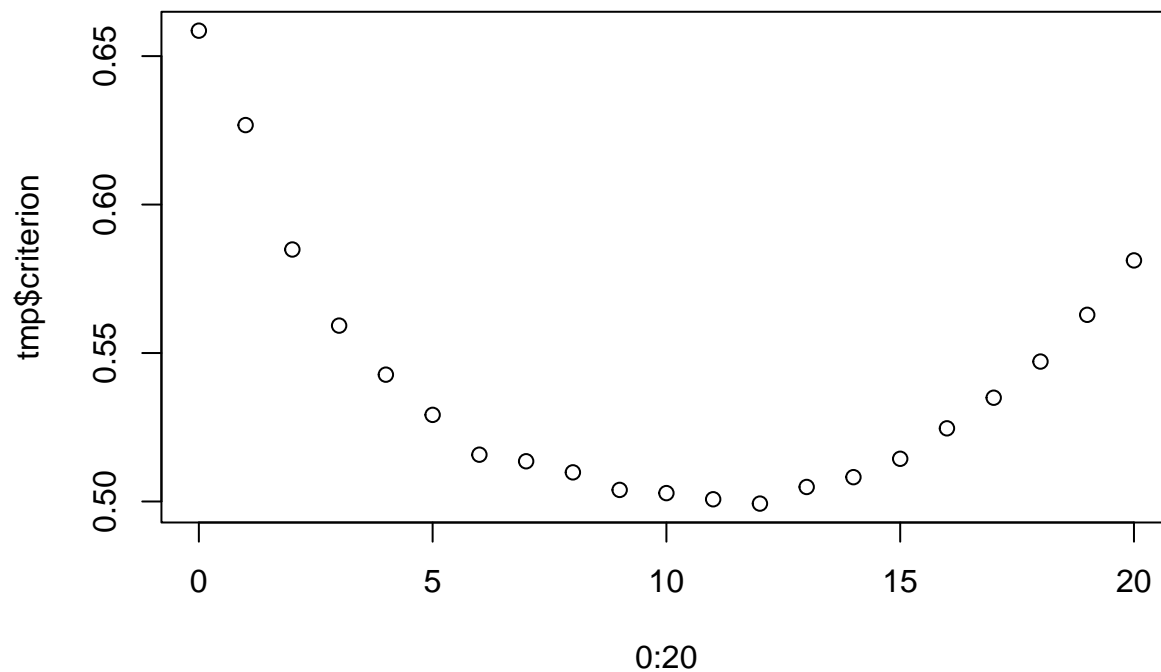
## Imputations des données manquantes

On va commencer par imputer les données manquantes par la moyenne des individus de la même classe

```
for (j in 1:ncol(X)){
  for (i in 1:nrow(X)){
    if (is.na(X[i,j])) X[i,j]=mean(X[Y==Y[i],j],na.rm=TRUE)
  }
}
```

ou mieux en utilisant le package suivant :

```
library(missMDA)
X=breast.tumors$gene.exp
colnames(X)=breast.tumors$genes$name
tmp=estim_ncpPCA(X,ncp.max=20)
plot(0:20,tmp$criterion)
```



```
X=imputePCA(X,ncp=tmp$ncp)$completeObs
```

## Random Forest

```
library(randomForest)
```

Estimons un modèle random forest.

```
model <- randomForest(X,Y,ntree=2000)
model$confusion
```

```
##      AF BE class.error
## AF   9 11   0.5500000
## BE   3 24   0.1111111
```

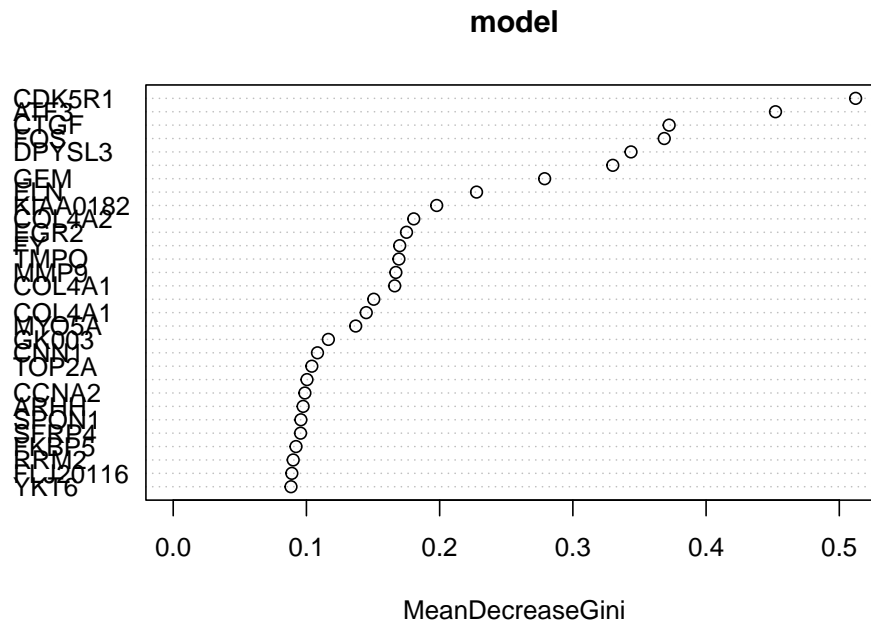
il y a une classe que l'on prédit plutôt bien (les BE), l'autre est par contre pas bien prédite du tout. Le taux de bon classement global est :

```
sum(diag(model$confusion))/sum(model$confusion)
```

```
## [1] 0.6923884
```

On peut représenter l'importance des variables, où l'on voit que quelques unes se détachent

```
varImpPlot(model)
```



Pour évaluer la stabilité de l'importance des variables, je vais réaliser un certain nombre de forêt aléatoires, et je vais regarder si ce sont toujours les mêmes variables qui sont les plus importantes

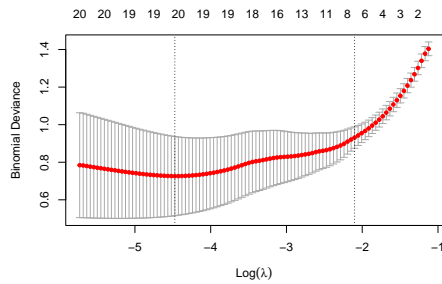
```
tmp2=NULL
for (i in 1:20){
  model <- randomForest(X,Y,ntree=2000)
  tmp=importance(model)
  # je récupère les indices des variables dont l'importance est >0.2
  tmp2=c(tmp2,which(tmp>0.2))
}
# je résume les indices dans variables importantes dans un tableau de contingence
tab=table(tmp2)
# je remplace ces indices par le nom des variables correspondantes
names(tab)=colnames(X)[as.integer(names(table(tmp2)))]
# j'affiche le tableau des noms des variables importantes par ordre décroissant
sort(tab,decreasing = T)
```

##	CTGF	GEM	ATF3	DPYSL3	FOS	CDK5R1		ELN
##	20	20	20	20	20	20	19	18
##	EGR2	FY KIAA0182						
##	17	9	3					

On arrive à identifier un petit pool de variables qui sont à chaque fois parmi les variables les plus importantes dans les différentes forêts aléatoires construites : CTGF, GEM, ATF3, DPYSL3, FOS, CDK5R1, EGR2, ELN

## Régression logistique

```
library(glmnet)
require(doMC)
registerDoMC(cores=4)
cvfit=cv.glmnet(X,Y,alpha=1,family="binomial",parallel=TRUE)
plot(cvfit)
```



```
fit=glmnet(X,Y,alpha=1,lambda=cvfit$lambda.1se,family="binomial")
# on peut regarder la performance de classement (ici optimiste car sur l'échantillon d'apprentissage)
pfit = predict(fit, X, s = cvfit$lambda.1se, type = "response")
table(pfit>.5, Y)
```

```
##          Y
##          AF BE
##  FALSE 16  0
##   TRUE   4 27
```

*# examinons les variables sélectionnées*

```
fit$beta[fit$beta@i+1]
```

```
## [1] -0.75612978 -0.10490729  0.03463505  0.10033703 -0.42885249  0.14490976
## [7] -0.21653174 -0.33241262
```

```
fit$beta@Dimnames[[1]][fit$beta@i+1]
```

```
## [1] "CTGF"    "FY"      "PDGFB"   "ODC1"    "GEM"     "COL4A1"  "FOS"     "CDK5R1"
```

On peut, comme pour les forêts aléatoires, ré-exécuter plusieurs fois le choix de lambda (qui peut varier du fait du découpage des folds en validation croisée)

```
numvar=NULL
for (i in 1:20){
  cvfit=cv.glmnet(X,Y,alpha=1,family="binomial",parallel=TRUE,nfolds=10)
  fit=glmnet(X,Y,alpha=1,lambda=cvfit$lambda.1se,family="binomial")
  numvar=c(numvar,fit$beta@i+1)
}
tab=table(numvar)
names(tab)=fit$beta@Dimnames[[1]][as.integer(names(tab))]
# j'affiche le tableau des noms des variables importantes par ordre décroissant
sort(tab,decreasing = T)
```

```
##    CTGF    GEM  CDK5R1    FOS    FY  PDGFB    ODC1  COL4A1    EGR2  CTNNAL1
##     20     20     16     12     7    5      5      5      3      2
##   EDR2   ATF3  S100A1  FGFR1   ECE1
##     2     2     2     1     1
```

Là encore un pool de variables ressort, relativement cohérent avec ce que l'on a obtenu avec les forêts aléatoires.

En croisant ainsi plusieurs exécution de l'estimation de nos modèles, cela permet d'avoir une idée assez sûre des variables les plus importantes