Scénario: intégration de données omiques et de toxicité pour le foi

Résumé

Etudes des données génomiques liver toxicity. *Comparaison et intégration de données transcriptomiques et phénotypiques de dommage du foi avec utilisation du package mixOmics : analyse en composantes principales (ACP), et* sparse *ACP, la régression PLS et* sparse *PLS, analyse discriminante PLS (PLS-DA) et* sparse *PLS-DA. Il est recommandé d'aborder les données génomiques (exploration élémentaire, classification, recherche de gènes différentiellement exprimés) par un scénario sur des données plus élémentaires. Ces points ne sont pas abordés sur ces données.*

1 Introduction

1.1 Objectif

L'hypothèse testée dans cette étude est la suivante : l'expression de certains gènes est un indicateur de dommage du foie lorsque ce dommage est créé par la dose d'une substance qui n'est pas forcément toxique.

Les données de cette étude sont utilisées pour illustrer des méthodes de sélection de variables par pénalisation Lasso couplées avec des techniques exploratoires (ACP) ou de modélisation (PLS, PLS-DA) ; Ces méthodes ainsi qu'une interprétation détaillée de ces données sont décrites par Lê Cao et al. (2008, 2011) [3] [2]. La dernière référence propose une comparaison approfondie de différentes méthodes de sélection de variables.

1.2 Données *Liver Toxicity*

Ces données évaluent l'impact sur le foie de rats d'une injection d'*acetaminophen* (paracétamol). Elles ont été acquises par Heinloth et al. (2004) (citées et normalisées par Bushel et al. (2007) [1]). Elles décrivent l'expression de 3116 gènes et 10 mesures cliniques relatives au foie de 64 rats. Les 64 sujets, tous soumis aux mêmes conditions expérimentales, ont été exposés à différentes doses de paracétamol : doses non toxiques (50 ou 150 mg/kg), doses moyennement toxiques (1500 mg/kg), doses très toxiques (2000 mg/kg). Les rats ont été ensuite euthanasiés par dioxyde de carbonne à différentes heures après avoir reçu l'injection : 6 heures, 18 heures, 24 heures ou 48 heures. Les mêmes conditions (dose, durée) sont chacunes répétées pour 4 individus.

Les données disponibles avec la librairie `mixOmics` sont présentées sous la forme d'une liste de 3 tableaux :

`gene` : (64 lignes × 3116 colonnes) donne l'expression de chacun des 3116 gènes pour chacun des 64 individus.

`clinic` : (64 lignes × 10 colonnes) contient les mesures de 10 variables cliniques pour chacun des 64 individus

`treatment` : (64 lignes × 4 colonnes) donne les informations nécessaires à l'étude relative à chacun des 64 individus : le numéro de l'animal, la dose d'acetaminophen injectée (50, 150, 1500 ou 2000mg/kg), l'heure à laquelle l'animal a été euthanasié (6h, 18h, 24h ou 48h).

Lecture des données :

```
library(mixOmics)
data(liver.toxicity)
help(liver.toxicity) # décrit les données
gene=liver.toxicity$gene
clinic=liver.toxicity$clinic
time=liver.toxicity$treatment$Time.Group
dose=liver.toxicity$treatment$Dose.Group
nomrats=dimnames(gene)[[1]]
```

2 Exploration

2.1 Unidimensionnelle

Quelques graphiques élémentaires permettent de se familiariser avec les données et se rendre compte qu'elles ont déjà été correctement normaliées sans doute par une fonction logarithme :

```
# distributions des gènes
```

```
boxplot(gene)
boxplot(data.frame(scale(gene)))
# "distributions" des rats
boxplot(data.frame(t(gene)))
# vérifier sommairement les raisons
# expliquant ces distributions
time
dose
```

2.2 Analyse en composantes principales

```
# éboulis des valeurs propres
pcatune(gene,ncomp=10)
title("Centrée et NON réduite")
pcatune(gene,ncomp=10,scale.=TRUE)
title("Centrée et réduite")
pcas.rat=pca(gene,scale.=TRUE)
pca.rat=pca(gene,scale.=FALSE)
# graphique évidemment illisible
biplot(pca.rat)
pt=(liver.toxicity$treatment$Time.Group/3)
# graphe des individus 2D ACP non réduite
plot(pca.rat$x,type="p",pch=pt,cex=1,
   col=rep(c(1,2,3,4),c(16,16,16,16)))
   legend(50, 70, c("50", "150", "1500", "2000"),
   col =c(1,2,3,4) , pch= rep(17,64), pt.cex = 1,
   title = "Dose injectee")
legend(-50, 70, c("6h", "18h", "24h", "48h"),
   pch=c(2,6,8,16) ,pt.cex = 1,
   title = "Heure d'euthanasie")
title("ACP non réduite")
# Comparer avec l'ACP réduite
x11()
plot(pcas.rat$x,type="p",pch=pt,cex=1,
   col=rep(c(1,2,3,4),c(16,16,16,16)))
   legend(50, 70, c("50", "150", "1500", "2000"),
   col =c(1,2,3,4) , pch= rep(17,64), pt.cex = 1,
   title = "Dose injectee")
legend(40, -4, c("6h", "18h", "24h", "48h"),
   pch=c(2,6,8,16) ,pt.cex = 1,
   title = "Heure d'euthanasie")
title("ACP réduite")
```

A une symétrie près, il y a semble-t-il peu de différences entre les deux représentations. La librairie permet une représentation 3D qui se justifie dans ce cas.

```
# Construction vecteur pour les formes suivant
# heure d'euthanasie :
timecol=as.vector(time)
timecol[time==6]= "c"
timecol[time==18]= "s"
timecol[time==24]= "t"
timecol[time==48]= "o"
# graphe des individus 3D
plot3dIndiv(pca.rat, cex=.6,col=rep(c(1,2,3,4),
   c(16,16,16,16)),pch=timecol)
```

Suivre l'évolution des positionnements des souris en fonction du temps de mesure par dose injectée.

2.3 ACP parcimonieuse ou *sparse*

Une version parcimonieuse est calculée afin d'obtenir une représentation lisible et plus facile à interpréter des gènes.

```
spca.rat=spca(gene,ncomp=3,keepX=rep(50, 3))
# graphe des individus 2D
plotIndiv(spca.rat,ind.names=FALSE, pch=pt ,
   cex=1 ,col=rep(c(1,2,3,4) ,c(16,16,16,16)))
legend(-0.03, 0.44, c("50", "150", "1500", "2000"),
   col =c(1,2,3,4),pch = rep(17,64), pt.cex = 1,
   title = "Dose")
legend(0.12, 0.44, c("6h", "18h", "24h", "48h"),
   pch=c(2,6,8,16),pt.cex = 1, title = "Temps")
title("sPCA")
```

```
# graphe des variables 2D
plotVar(spca.rat, comp = c(1,2),var.label =
   TRUE, cex = 0.7)
title("sPCA des donnees initiales")
# graphe des individus 3D
plot3dIndiv(spca.rat,cex=0.015,col=rep(c(1,2,3,4),
   c(16,16,16,16)),pch=timecol)
```

Reprendre les calculs en limitant encore plus le nombre de gènes présents dans l'analyse. Voir que les représentations des individus sont peu affectés alors que les graphes des variables deviennent utilisables.

3 Régression PLS et sparse PLS

L'intégration des données a pour objectif d'expliquer globalement les données cliniques par les données d'expression en utilisant la PLS de type 2 (variable Y multidimensionnelle) en mode régression.

3.1 Régression PLS

Choix de la dimension La fonction `valid()` permet d'évaluer différents critères liés à la qualité de prévision par validation croisée selon différentes options (validation croisée M-*fold* ou *leave-one-out*). Avec des dimensions raisonnables (p plus petit), il est alors possible de chercher la dimension minimisant le critère Q^2_h.

```
liver.valid.pls=valid(gene,clinic,ncomp=5,
   method="pls",mode="regression",
   criterion="all",validation="Mfold",M=10)
plot(1:5,liver.valid.pls$Q2$total)
```

La situation est ici trop complexe, nous nous contenterons d'un nombre restreint de composantes pour un objectif exploratoire.

Etudier l'erreur de prévision pour chaque variable clinique et pour un nombre donné de dimensions ($PRESS_{k,h}$)

```
plot(liver.valid.pls,criterion="MSEP",type="l",
   layout=c(3, 4))
```

Pour simplifier les représentations "exploratoires", trois dimensions sont retenues.

Estimation du modèle :

```
liver.pls=pls(gene,clinic,ncomp=3,mode="regression")
```

Représentation des individus La fonction `plotIndiv()` permet de représenter les individus en 2D en choisissant les 2 axes principaux retenus de la PLS.

```
plotIndiv(liver.pls,comp=1:2,rep.space ="X-variate")
```

Les individus sont représentés par leur nom. Pour donner plus de sens à cette représentation, ils sont identifiés par un ensemble de deux codes : une couleur (`mycol`) correspondant à la dose de paracétamol reçue, et un type de symbole correspondant à l'heure de mesure des variables d'expression et cliniques.

```
pt=time/3
mycol=rep(c(1,2,3,4),c(16,16,16,16))
plotIndiv(liver.pls, ind.names = FALSE,
   pch=pt ,cex=1 ,col=mycol)
legend(31,-15,c("50","150","1500","2000"),
   fill=c(1,2,3,4),pt.cex=1,title="Dose injectee")
legend(53,-15,c("6h","18h","24h","48h"),
   pch=c(2,6,8,16),pt.cex= 1,title="Heure de mesure")
```

Commenter la manière dont se regroupent les individus, en précisant le rôle des axes, comparer avec l'analyse en composantes principales.

Comme en ACP, il est possible de représenter les individus en 3D à l'aide de la fonction `plot3dIndiv()`. Les symboles utilisés en 2D n'étant pas acceptés ici, d'autres sont sélectionnés (`timecol`) :

```
timecol=as.vector(time)
timecol[time==6]= "c" # circle
timecol[time==18]= "s" # square
timecol[time==24]= "t" # triangle
timecol[time==48]= "o" # ovale
plot3dIndiv(liver.pls,cex=30,col=mycol,pch=timecol)
```

Représentation des variables Les corrélations entre variables initiales et variables finales peuvent être représentées avec la fonction `plotVar()`. Comme il y a trop de variables d'expression (gènes), seules les variables cliniques sont représentées :

```
plotVar(liver.pls, comp = 1:2, Y.label = TRUE,
   X.label = FALSE)
```

En confrontant les résultats de ces deux graphiques, lister les variables cliniques dont la variation est très corrélée à la dose de paracétamol injectée. Pour le vérifier, il suffit par exemple de tracer ces variables en fonction de la variable `dose` à l'aide de la fonction `plot()`.

3.2 Régression *sparse* PLS

Une sélection des gènes va permettre une représentation lisible et interprétable afin d'expliquer globalement les données cliniques en fonction des données d'expression. C'est donc une version *sparse* de la régression PLS qui est utilisée en se limitant à 3 composantes.

On décide de retenir 50 gènes par dimension pour les données d'expression. En revanche, comme on a peu de variables cliniques, on ne les sélectionne pas et on en garde 10 par dimension. La commande à utiliser est donc :

```
liver.spls=spls(gene,clinic,ncomp=3,
   mode="regression",keepX=rep(50,3),keepY=rep(10,3))
```

```
plotIndiv(liver.spls, ind.names = FALSE,
   pch=pt ,cex=1 ,col=mycol)
legend(0.1,0.4,c("50","150","1500","2000"),
   fill=c(1,2,3,4),pt.cex=1,title="Dose injectee")
legend(0.25,0.4,c("6h","18h","24h","48h"),
   pch=c(2,6,8,16),pt.cex=1,title="Heure de mesure")
```

Comparer avec la PLS.

Représentation des variables Pour la représentation des variables, le nombre de gènes est beaucoup moins important et on peut donc les représenter par leur nom en modifiant les options de `plotVar()`. On peut ainsi identifier les gènes associés à un axe.

```
plotVar(liver.spls, comp = 1:2, Y.label = TRUE,
   X.label = TRUE)
```

Petit coup d'oeil dans la boite noire Examiner les attributs de la variable résultat `liver.spls`. Essayer par exemple de récupérer la liste des gènes sélectionnés pour l'axe 1.

```
attributes(liver.spls)
liver.spls$loadings$Y
liver.spls$load$X[abs(liver.spls$load$X[,1])>0.1,]
```

Erreur de prévision Les commandes ci-dessous permettent de comparer les erreurs de prévision obtenues avec la PLS et la sparse PLS :

```
liver.valid.spls=valid(gene, clinic, ncomp = 4,
   method="spls",mode="regression",criterion=
   "all",validation = "Mfold", M=10)
par(mfrow = c(3, 4))
for(i in 1:10){
    spls.rmsep=sqrt(liver.valid.spls$MSEP[i, ])
    pls.rmsep=sqrt(liver.valid.pls$MSEP[i, ])
    matplot(cbind(spls.rmsep, pls.rmsep),
       type = "l", ylab = "RMSEP",
        lwd = 2, xlab = "dim", axes = FALSE)
    axis(1, 1:4, labels = 1:4)
    axis(2)
    title(main=paste(rownames(liver.valid.spls
    $MSEP)[i]))
}
```

On voit ainsi que, indépendamment de l'éclairage que les deux méthodes apportent sur les ressemblances entre individus, la plupart des variables sont mieux prévues en utilisant la Sparse PLS que la PLS. Si l'objectif prédictif

est la priorité de l'étude, il est également conseillé de faire varier le nombre de variables par dimension de la sparse PLS afin de minimiser l'erreur de prévision.

4 PLS-DA des données d'expression sur la dose

4.1 PLS-DA

On a vu que les analyses précédentes, en s'intéressant au lien entre données d'expression et données cliniques, permettaient indirectement de séparer les individus en fonction de la dose de paracétamol reçue, et, pour la Sparse PLS, de faire ressortir les gènes dont les variations sont liées à cette dose. Si cette classification en fonction de la dose reçue est l'objectif principal de l'étude, des outils tels que la PLS-DA et la Sparse PLS-DA paraissent toutefois plus adaptés, puisqu'ils permettent de prédire une variable de type classe (ici la dose) en fonction de variables quantitatives (ici les expressions de gènes). Pour utiliser ces méthodes, on doit transformer la variable `dose`, qui est au départ numérique, en une variable de type **factor**, puis utiliser les fonctions `plsda()` et `splsda()`. Par exemple ici :

```
liver.plsda=plsda(gene,as.factor(dose),ncomp = 3)
liver.splsda=splsda(gene,as.factor(dose),
   ncomp = 3,keepX=rep(50,3))
```

Représentation des résultats Commenter les graphes des individus obtenus par les deux méthodes. En vous aidant du graphe des variables, donner des gènes sur-exprimés pour les doses fortes.

```
# Représentation des individus
plotIndiv(liver.splsda, ind.names = FALSE,
   pch=pt ,cex=1 ,col=mycol)
legend(-.15,.3,c("50","150","1500","2000"),
   fill=c(1,2,3,4),pt.cex=1,title="Dose")
# représentation des variables
plotVar(liver.splsda, comp = 1:2)
```

Erreur de classification En utilisant la fonction `valid()`, comparer les erreurs de classification obtenus par les deux méthodes. Ces résultats sont-ils cohérents avec les représentations des individus ?

```
plotIndiv(liver.plsda, ind.names = FALSE, pch=pt ,
   cex=1, col=rep(c(1,2,3,4),c(16,16,16,16)))
legend(-47,5, c("50", "150", "1500", "2000"),
   col =c(1,2,3,4), pch =rep(17,64), pt.cex = 1,
   title = "Dose injectee")
legend(18, -7, c("6h", "18h", "24h", "48h"),
   pch = c(2,6,8,16),pt.cex =1,
   title = "Heure d'euthanasie");
title("PLS-DA facteur=dose")

plotIndiv(liver.splsda, ind.names = FALSE, pch=pt ,
   cex=1, col=rep(c(1,2,3,4),c(16,16,16,16)))
legend(-47,5, c("50", "150", "1500", "2000"),
   col =c(1,2,3,4), pch =rep(17,64), pt.cex = 1,
   title = "Dose injectee")
legend(18, -7, c("6h", "18h", "24h", "48h"),
   pch = c(2,6,8,16),pt.cex =1,
   title = "Heure d'euthanasie");
title("PLS-DA facteur=dose")

# graphe des individus 3D
plot3dIndiv(liver.splsda,cex=1,
   col=rep(c(1,2,3,4),c(16,16,16,16)),pch=timecol)

# fonction du temps

liver.plsda=plsda(gene,as.factor(time),ncomp = 3)
liver.splsda=splsda(gene,as.factor(time),
   ncomp = 3,keepX=rep(50,3))


plotIndiv(liver.plsda, ind.names = FALSE, pch=pt ,
   cex=1 ,col=rep(c(1,2,3,4),c(16,16,16,16)))
```

```
legend(-47,5, c("50", "150", "1500", "2000"),
   col =c(1,2,3,4), pch =rep(17,64), pt.cex = 1,
   title = "Dose injectee")
legend(18, -7, c("6h", "18h", "24h", "48h"),
   pch = c(2,6,8,16),pt.cex =1,
   title = "Heure d'euthanasie");
title("PLS-DA facteur=temps")

plotIndiv(liver.splsda, ind.names = FALSE, pch=pt ,
cex=1 ,col=rep(c(1,2,3,4),c(16,16,16,16)))
   legend(-47,5, c("50", "150", "1500", "2000"),
   col =c(1,2,3,4), pch =rep(17,64), pt.cex = 1,
   title = "Dose injectee")
legend(18, -7, c("6h", "18h", "24h", "48h"),
   pch = c(2,6,8,16),pt.cex =1,
   title = "Heure d'euthanasie");
title("PLS-DA facteur=temps")

plot3dIndiv(liver.splsda,cex=30,
   col=rep(c(1,2,3,4),c(16,16,16,16)),pch=timecol)
```

Références

[1] A. Buja, T. Hastie et R. Tibshirani, *Linear smoothers and additive models*, The Annals of Statistics **17** (1989), 453–555.

[2] K. A. Lê Cao, S. Boistard et P. Besse, *Sparse PLS Discriminant Analysis : biologically relevant feature selection and graphical displays for multiclass problems*, BMC Bioinformatics **12** (2011), n° 253.

[3] K. A. Lê Cao, D. Rossouw, C. Robert-Granié et P. Besse, *A sparse PLS for variable selection when integrating Omics data*, Statistical Applications in Genetics and Molecular Biology **7** (2008), n° 35.