Créer et Valider une signature diagnostique à partir de biopuces

Partie 2 : Créer une signature diagnostique

Yohann.Foucher@univ-nantes.fr

Equipe d'Accueil 4275 "Biostatistique, recherche clinique et mesures subjectives en santé", Université de Nantes

MASTER 1 Bionformatique et Biostatistique - UE données omics

Plan

Plan

Méthodologie classiquement utilisée : le modèle logistique

Y : v.a. à expliquer binaire (1/0, malade/non-malade).

$$logit(Pr(Y = 1|X_1, X_2, ..., X_p)) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_p X_p + \epsilon$$

où :

- $X_1, X_2, ..., X_p$ sont les variables explicatives (biomarqueurs, caractéristiques cliniques, etc.)
- β_0 est l'intercept
- $\beta_1, \beta_2, ..., \beta_p$ sont des coefficients de régression
- ϵ est l'erreur du modèle

Interprétation

- X_k est une variable quantitative, alors $exp(\beta_k)$ correspond à l'odds ratio pour une augmentation de 1 unité du biomarqueur k.
  - → Si $\beta_k > 0$ alors la probabilité d'appartenir à modalité d'intérêt augmente quand l'expression du gène augmente.
  - → Si $\beta_k < 0$ alors la probabilité d'appartenir à modalité d'intérêt diminue quand l'expression du gène augmente.
- Il est donc naturel d'écrire le score prédictif suivant :

$$S = \beta_1 X_1 + \beta_2 X_2 + ... + \beta_p X_p$$

  - → Plus S est grand, plus le sujet a de chance d'appartenir à $Y = 1$.
  - → Plus S est grand, plus le sujet a de chance d'appartenir à $Y = 0$.

Problème

- Comment sélectionner les variables à inclure dans la signature puisque la p-value n'est pas un bon indicateur dans notre contexte ?
- Rappel : p-value = probabilité que l'effet observé ne soit pas dû au hasard.
- Nous souhaitons conserver une variable dans le score si elle apporte une information relevante pour la discrimination de deux groupes.

La preuve par l'exemple *

- 128 patients avec un diagnostic de leucémie aiguë lymphoblastique (ALL).
- Des puces contenant l'expression de 12625 gènes ont été collectés.
- Objectif de l'exemple : Identifier un ensemble de gènes prédisant la rémission après traitement.

∗. S. Chiaretti et al. Gene expression profile of adult T-cell acute lymphocytic leukemia identifies distinct subsets of patients with different response to therapy and survival. Blood, 2004, Vol. 103, No. 7

Exemple : Rémission d'une leucémie


```
> library(Biobase)
> library(multtest)
> library(ALL)
> library(genefilter)
> data(ALL)
> dim(exprs(ALL))

[1] 12625   128

> X<-exprs(ALL)
> pheno<-pData(ALL)
```

Exemple : Rémission d'une leucémie


```
> ffun<-filterfun(pOverA(p=0.2, A=100), cv(a=0.7, b=4))
> filt<-genefilter(2^X, ffun)
> rem<-1*(as.character(pheno$remission)=="CR")
> filtX<-X[filt,!is.na(rem)]
> rem <- rem[!is.na(rem)]
> rem.boot<-MTP(X = filtX, Y=rem, test = "t.twosamp.equalvar",
+  robust=TRUE, alpha=0.05, B = 20, get.cutoff = TRUE)

running bootstrap...
iteration = 20

> sum(rem.boot@adjp<=0.05)

[1] 8
```

Exemple : Rémission d'une leucémie

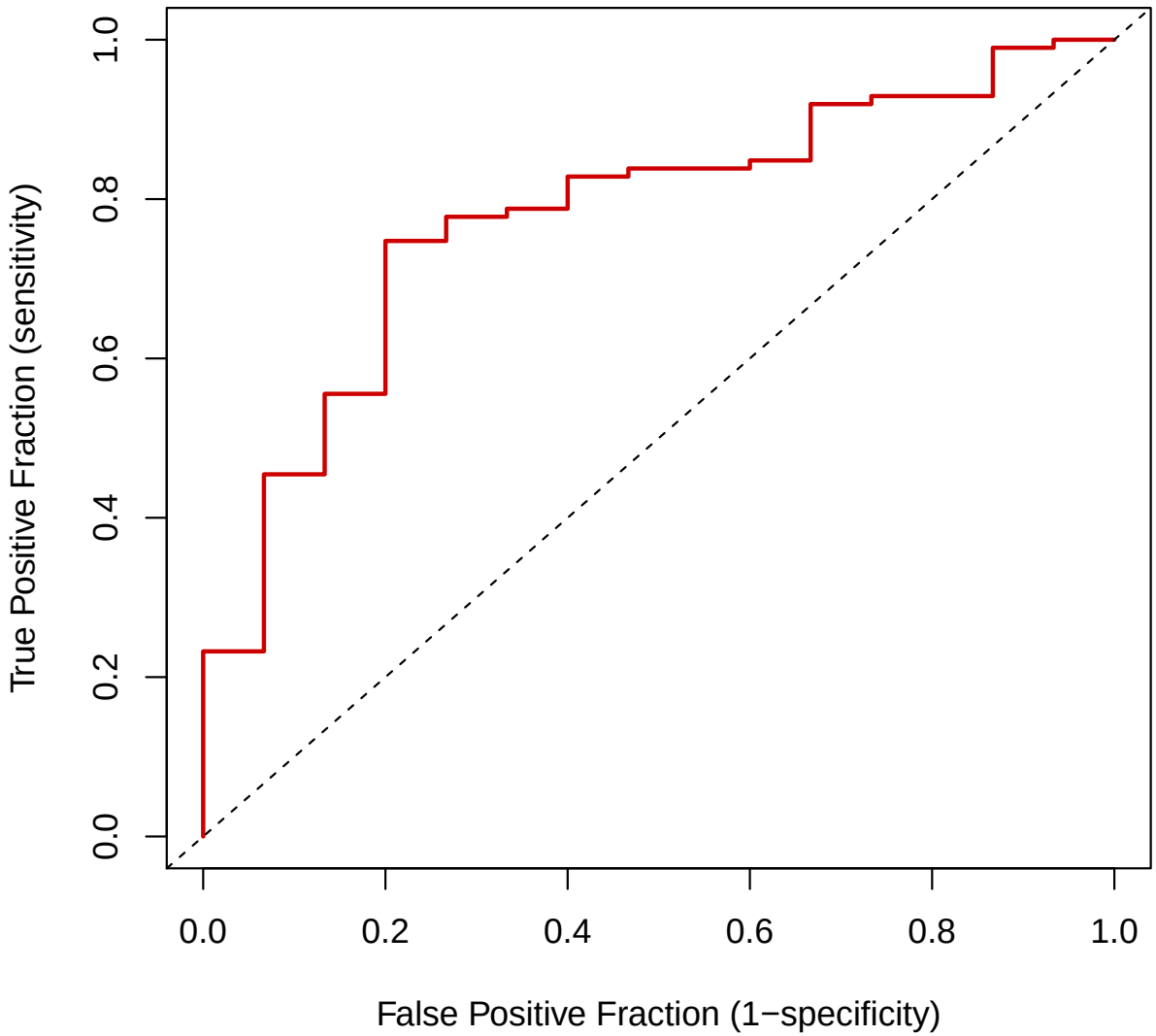

```
> indic <- rem.boot@adjp==min(rem.boot@adjp)
> if(sum(indic)==1) {X1 <- filtX[indic,]}  else {X1 <- filtX[indic,][1,]}
> range(X1)

[1] 5.155300 9.992673

> summary(logit.model<-glm(rem ~ X1 , binomial))$coef

             Estimate Std. Error   z value     Pr(>|z|)
(Intercept)  9.0441043  2.3182710  3.901228 9.570588e-05
X1          -0.9616988  0.2978995 -3.228265 1.245433e-03

> exp(summary(logit.model)$coef[2,1])

[1] 0.382243

> S <- logit.model$coef[2] * X1
>
```

- L'OR associé à ce gène est de 0.38.
- Pour une diminution de 2 unités, le risque est multiplié par 6.84.
- Pas mal... Mais quelle est l'aire sous la courbe ROC correspondante ?

Exemple : Rémission d'une leucémie


```
> se<-function(x) { return(sum(1*(S[rem==1]>x))/sum(rem==1)) }
> sp<-function(x) { return(sum(1*(S[rem==0]<=x))/sum(rem==0)) }
> temp<-sort(unique(S))
> resultats<-data.frame(
+  sp=sapply(temp, sp),
+  sp1=1-sapply(temp, sp),
+  se=sapply(temp, se), seuils=temp)
> plot(c(1, resultats$sp1, 0), c(1, resultats$se, 0),
+  xlab="False Positive Fraction (1-specificity)",
+  ylab="True Positive Fraction (sensitivity)",
+  type="n")
> lines(c(1, resultats$sp1, 0), c(1, resultats$se, 0),
+  type="s", lwd=2, col="red3")
> abline(0,1, lty=2)
>
```

Exemple : Rémission d'une leucémie



Exemple : Rémission d'une leucémie


```
> resultats<-resultats[order(resultats$sp1, resultats$se),]
> resultats[dim(resultats)[1]+1,1]<-1
> resultats[dim(resultats)[1],2]<-1
> resultats[dim(resultats)[1],3]<-resultats[dim(resultats)[1]-1,3]
> AUC <- sum( (resultats$sp1[2:length(resultats$sp1)] -
+  resultats$sp1[1:(length(resultats$sp1)-1)] ) *
+  (resultats$se[2:length(resultats$se)]) )
> AUC
[1] 0.7784512
```

- L'AUC est "seulement" de 0.78...

Plan

Notations pour un sujet j ($j = 1, ..., N$)

- X_j est le vecteur des P variables avec $X_j = (X_{j1}, ..., X_{jP})$.
- Y_j la v.a. binaire à expliquer.
- Le modèle :

$$logit(\pi_j =) = \beta_0 + \beta_1 X_{j1} + ... + \beta_P X_{jP} + \epsilon_j$$

où $\pi_j = Pr(Y_j = 1|X_j)$

- Estimation des paramètres β en maximisant la Vraisemblance.
- On note $\ell(\beta)$ cette Vraisemblance (Likelihood).
- La vraisemblance représente la probabilité d'observer l'échantillon. On part du principe que cette probabilité devait être maximale si on a observé cet échantillon. On cherche donc les paramètres les plus vraisemblables.

Définition des contributions individuelles à la Vraisemblance

- Soit un échantillon de taille N ($j = 1, ..., N$).
- On suppose tous les sujets indépendants les uns des autres.
- La probabilité d'observer l'échantillon est alors le produit des probabilités d'observer chaque sujet ($Pr(A, B) = Pr(A) \times Pr(B)$ si A et B sont indépendants).
- On parle de contribution individuelle à la vraisemblance, $\ell_j(\beta)$.
- La seule difficulté est de définir ces contributions en fonction de la v.a. à expliquer.
- Deux type de sujets :
  - Si $Y_j = 1$, alors $\ell_j(\beta) = \pi_j$.
  - Si $Y_j = 0$, alors $\ell_j(\beta) = 1 - \pi_j$.

Définition de la Vraisemblance, $\ell(\beta)$

$$\ell(\beta) = \prod_j \ell_j(\beta)$$

$$\Downarrow$$

$$\ell(\beta) = \prod_j \pi_j^{y_j} \times (1-\pi_j)^{1-y_j}$$

$$\Downarrow$$

$$\hat{\beta} = \text{ arg max}_\beta \Big\{ \ell(\beta) \Big\}$$

Un petit exemple pour comprendre le principe


```
> Data <- data.frame(y=c(1,1,0,0), x=c(1, 2, 3, 4))
> Data
  y x
1 1 1
2 1 2
3 0 3
4 0 4
> pj <- function(b0, b1, co)
+ { exp(b0 + b1 * co)/(1+exp(b0 + b1 * co)) }
> pj(b0=0.3, b1=1.5, co=Data$x)
[1] 0.8581489 0.9644288 0.9918374 0.9981671
> pj(b0=0.3, b1=-1.5, co=Data$x)
[1] 0.231475217 0.062973356 0.014774032 0.003334807
> pj(b0=0.3, b1=1.5, co=Data$x)^Data$y
[1] 0.8581489 0.9644288 1.0000000 1.0000000
> (1-pj(b0=0.3, b1=1.5, co=Data$x))^(1-Data$y)
[1] 1.000000000 1.000000000 0.008162571 0.001832939
> prod((pj(b0=0.3, b1=1.5, co=Data$x)^Data$y) *
+ ((1-pj(b0=0.3, b1=1.5, co=Data$x))^(1-Data$y)))
[1] 1.238249e-05
>
```

Un petit exemple pour comprendre le principe


```
> temp.b1 <- seq(-5,5,by=0.01)
> temp.l <- rep(-99, length(temp.b1))
> for(i in 1:length(temp.l))
+ {
+ temp.l[i] <- prod((pj(b0=0.3, b1=temp.b1[i], co=Data$x)^Data$y) *
+ ((1-pj(b0=0.3, b1=temp.b1[i], co=Data$x))^(1-Data$y)))
+ }
> cbind(temp.b1, temp.l)[1:10,]
```

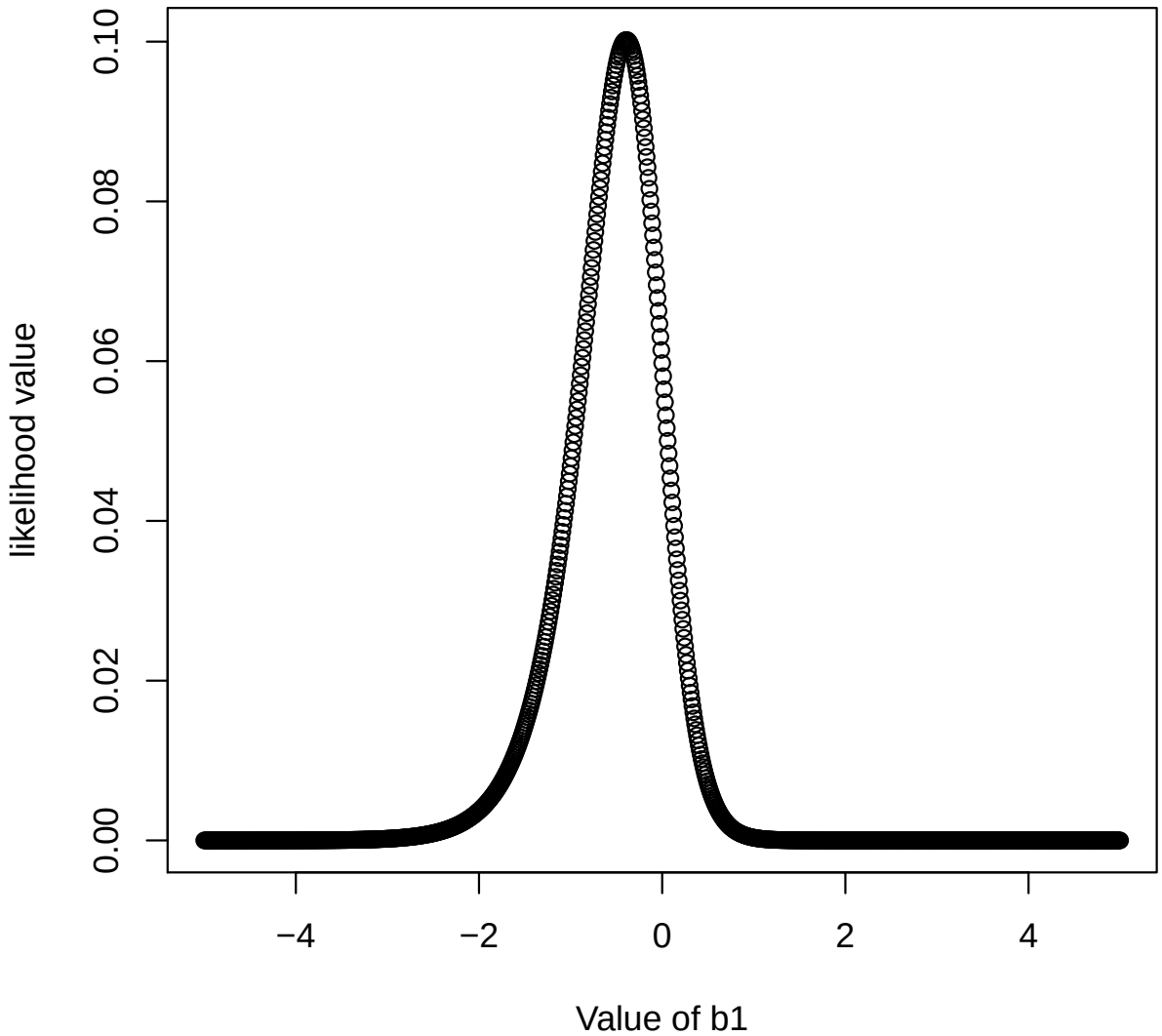
```
      temp.b1       temp.l
 [1,]   -5.00 5.523324e-07
 [2,]   -4.99 5.691011e-07
 [3,]   -4.98 5.863784e-07
 [4,]   -4.97 6.041797e-07
 [5,]   -4.96 6.225208e-07
 [6,]   -4.95 6.414181e-07
 [7,]   -4.94 6.608883e-07
 [8,]   -4.93 6.809490e-07
 [9,]   -4.92 7.016178e-07
[10,]   -4.91 7.229133e-07
```

```
> plot(temp.b1, temp.l, ylab="likelihood value", xlab="Value of b1")
> temp.b1[temp.l==max(temp.l)]
```

```
[1] -0.39
```

```
>
```

Un petit exemple pour comprendre le principe



Problèmes lors de la construction d'un score

- Sélection sur la p-value inadéquate.
- Trop de paramètres à estimer (à partir des données omics).
- Colinéarité de certaines variables (à partir des données omics).

La pénalisation de lasso †

$$\hat{\beta} = \arg\max_{\beta} \left\{ \ell(\beta) - \lambda \sum_{p=1}^{P} |\beta_p| \right\}$$

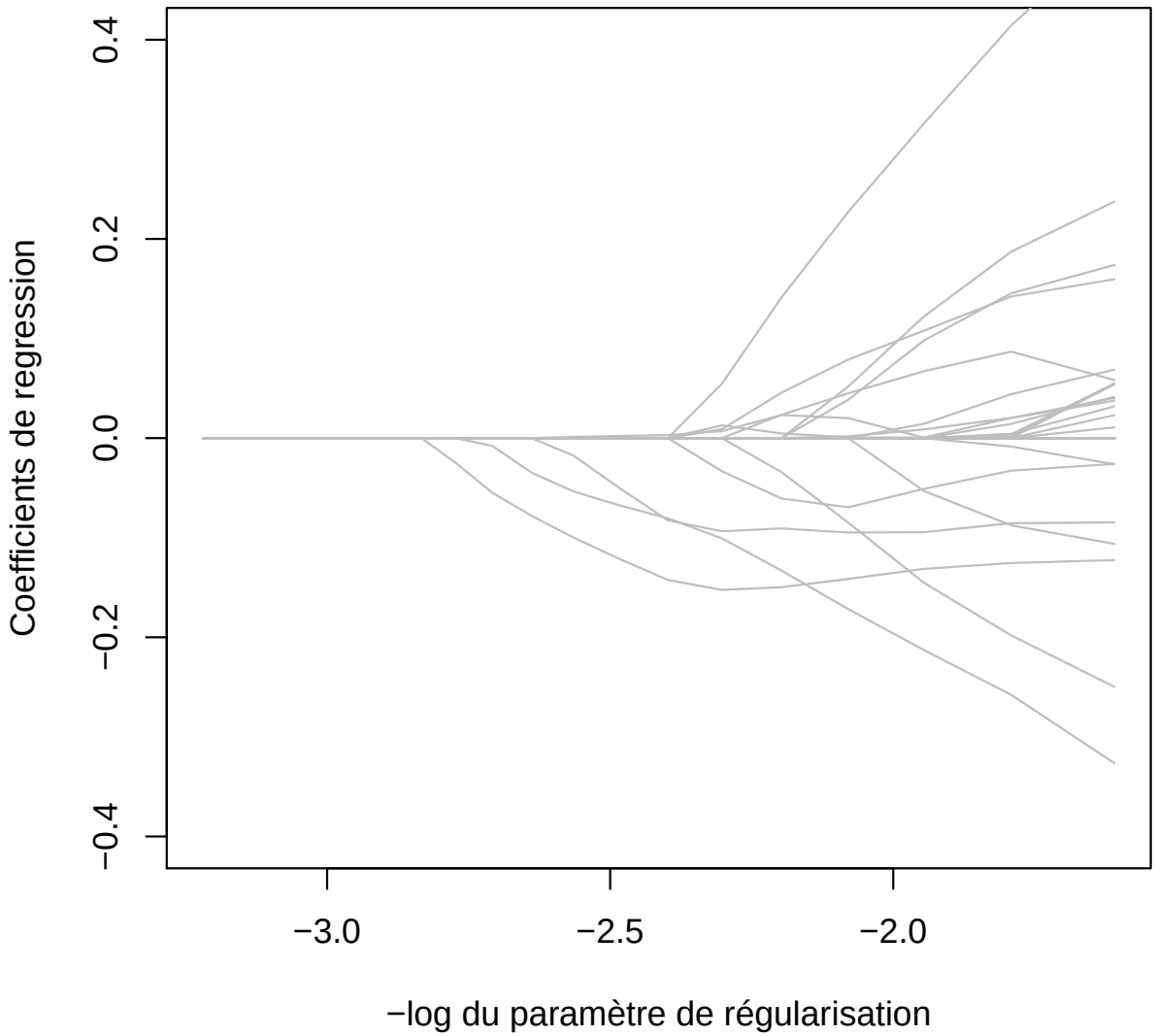
- λ est le paramètre de régularisation (*tuning parameter*).
- Permet d'obtenir un modèle parcimonieux :
  - Restriction de la valeur de certains coefficients.
  - Annulation de la valeur des coefficients associés aux variables les moins informatives.

†. R Tibshirani. Regression shrinkage and selection via the Lasso. Journal of the Royal Statistical Society : Series B, 58 :267-288, 1996

Exemple : Rémission d'une leucémie


```
> library(penalized)
> N <- dim(filtX)[2]; N

[1] 114

> G <- dim(filtX)[1]; G

[1] 430

> lambda <- seq(5, 25, by=1)
> Beta <- matrix(0, ncol=G, nrow=length(lambda))
> for (i in 1:length(lambda)) {
+ pen1 <- penalized(rem, penalized = t(filtX),
+  lambda1=lambda[i], model="logistic", trace = FALSE)
+
+ Beta[i,]  <- coefficients(pen1, "all")[-1] }
> dim(Beta)

[1]  21 430

> plot(log(1/lambda), Beta[,1], xlab="-log du paramètre de régularisation", ylab="C
> for (i in 1:G) { lines(log(1/lambda), Beta[,i],  type="l", col="gray", lwd=1) }
>
```

Exemple : Rémission d'une leucémie



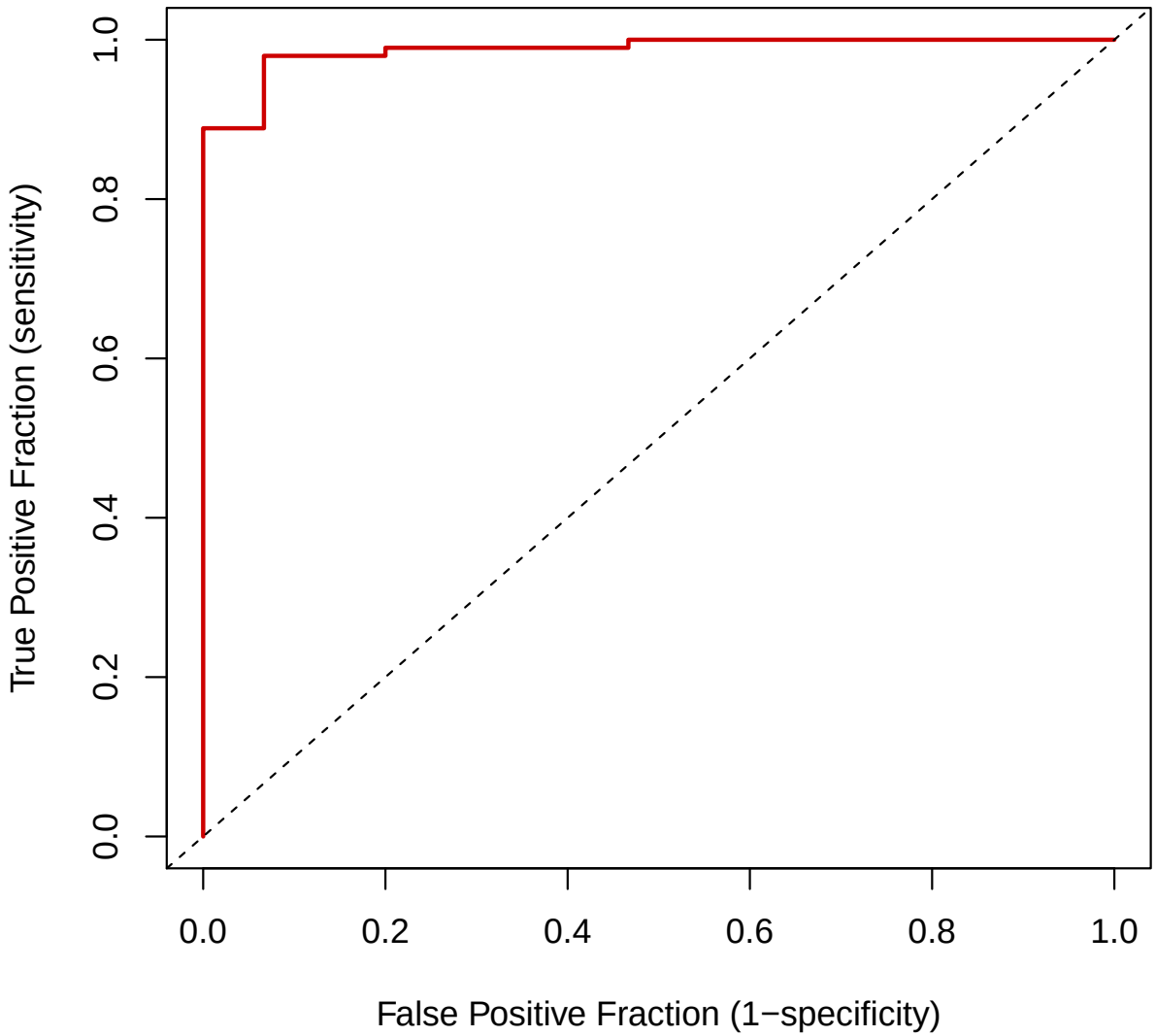
Problème : Quelle valeur de λ choisir ?


```
> # lambda =5
> pen1 <- penalized(rem, penalized = t(filtX),
+  lambda1=5, model="logistic", trace = FALSE)
> sum(coefficients(pen1, "all")!=0)

[1] 23

> S<-predict(pen1,  t(filtX))
>
```

Exemple : Rémission d'une leucémie


```
> se<-function(x) { return(sum(1*(S[rem==1]>x))/sum(rem==1)) }
> sp<-function(x) { return(sum(1*(S[rem==0]<=x))/sum(rem==0)) }
> temp<-sort(unique(S))
> resultats<-data.frame(
+  sp=sapply(temp, sp),
+  sp1=1-sapply(temp, sp),
+  se=sapply(temp, se), seuils=temp)
> plot(c(1, resultats$sp1, 0), c(1, resultats$se, 0),
+  xlab="False Positive Fraction (1-specificity)",
+  ylab="True Positive Fraction (sensitivity)",
+  type="n")
> lines(c(1, resultats$sp1, 0), c(1, resultats$se, 0),
+  type="s", lwd=2, col="red3")
> abline(0,1, lty=2)
>
```

Exemple : Rémission d'une leucémie



Exemple : Rémission d'une leucémie


```
> resultats<-resultats[order(resultats$sp1, resultats$se),]
> resultats[dim(resultats)[1]+1,1]<-1
> resultats[dim(resultats)[1],2]<-1
> resultats[dim(resultats)[1],3]<-resultats[dim(resultats)[1]-1,3]
> AUC <- sum( (resultats$sp1[2:length(resultats$sp1)] -
+  resultats$sp1[1:(length(resultats$sp1)-1)] ) *
+  (resultats$se[2:length(resultats$se)]) )
> AUC
[1] 0.9872054
```

- L'AUC est "seulement" de 0.99 : sur-ajustement.

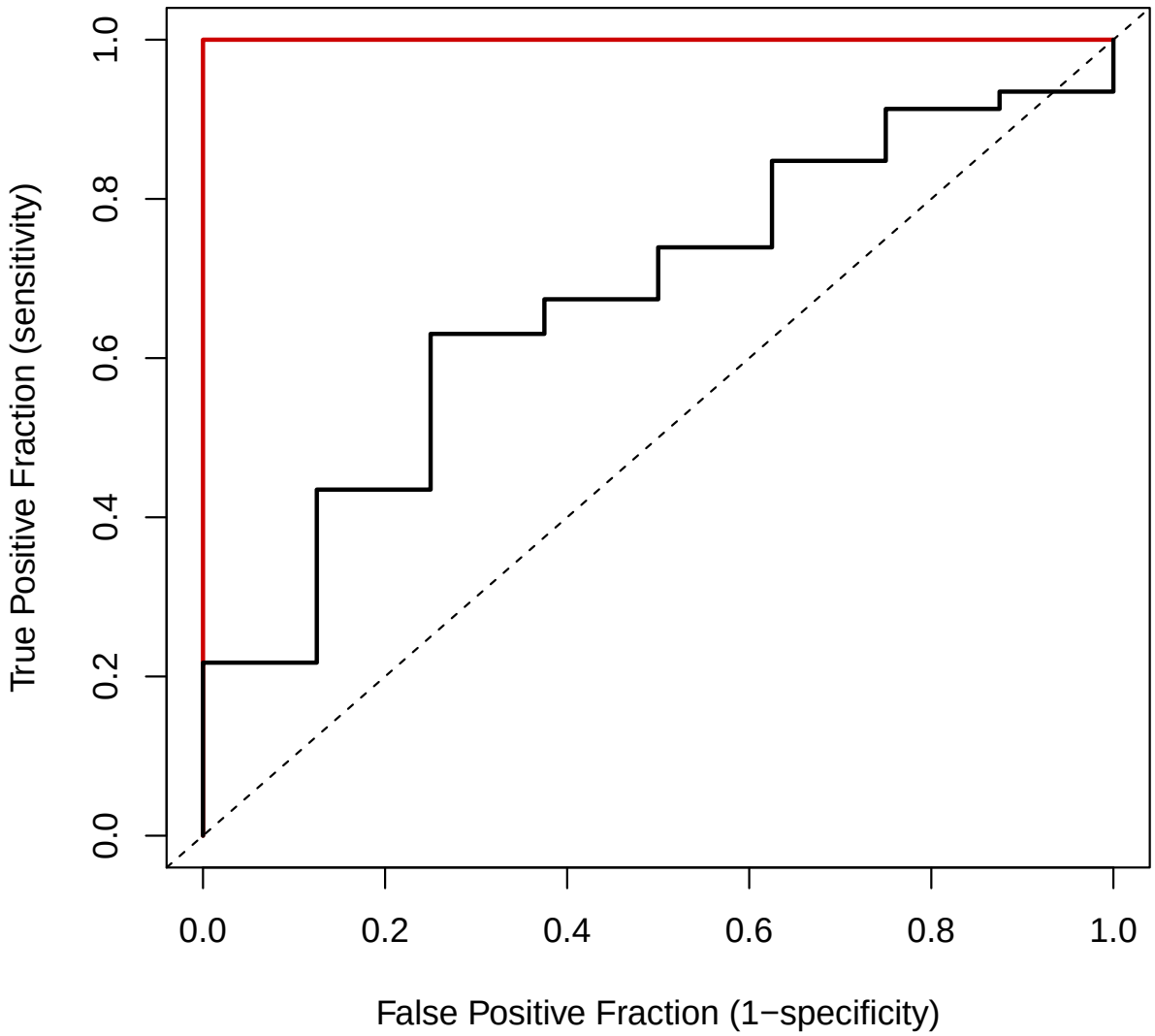
Démonstration du sur-ajustement


```
> indic <- rbinom(length(rem), 1, p=0.5)
> rem1 <- rem[indic==1]
> rem0 <- rem[indic==0]
> filtX1 <- t(filtX)[indic==1,]
> filtX0 <- t(filtX)[indic==0,]
> # lambda =3
> pen1 <- penalized(rem1, penalized = filtX1,
+  lambda1=3, model="logistic", trace = FALSE)
> sum(coefficients(pen1, "all")!=0)

[1] 15

> S1<-predict(pen1,  filtX1)
> S0<-predict(pen1,  filtX0)
>
```

Démonstration du sur-ajustement


```
> se1<-function(x) { return(sum(1*(S1[rem1==1]>x))/sum(rem1==1)) }
> sp1<-function(x) { return(sum(1*(S1[rem1==0]<=x))/sum(rem1==0)) }
> se0<-function(x) { return(sum(1*(S0[rem0==1]>x))/sum(rem0==1)) }
> sp0<-function(x) { return(sum(1*(S0[rem0==0]<=x))/sum(rem0==0)) }
> temp<-sort(unique(c(S1, S0)))
> resultats<-data.frame(
+  sp1=1-sapply(temp, sp1),
+  se1=sapply(temp, se1),
+  sp0=1-sapply(temp, sp0),
+  se0=sapply(temp, se0),
+  seuils=temp)
> plot(c(1, resultats$sp1, 0), c(1, resultats$se1, 0),
+  xlab="False Positive Fraction (1-specificity)",
+  ylab="True Positive Fraction (sensitivity)",
+  type="n")
> lines(c(1, resultats$sp1, 0), c(1, resultats$se1, 0),
+  type="s", lwd=2, col="red3")
>  lines(c(1, resultats$sp0, 0), c(1, resultats$se0, 0),
+  type="s", lwd=2, col=1)
> abline(0,1, lty=2)
>
```

Démonstration du sur-ajustement



Estimation du paramètre λ par validation croisée ‡

- Division de l'échantillon en K groupes ($k = 1, ..., K$).
- Soit $\hat{\beta}_{(-k)}(\lambda)$ l'estimation de β sans le $k$ième groupe.
- Soit $\ell_{(-k)}(\hat{\beta}_{(-k)}(\lambda))$ la log vraisemblance partielle.
- Soit $\ell(\hat{\beta}_{(-k)}(\lambda))$ la log vraisemblance partielle avec tous les patients mais avec les paramètres précédents.

$$CV(\lambda) = \sum_{k=1}^{K} \left\{ \ell(\hat{\beta}_{(-k)}(\lambda)) - \ell_{(-k)}(\hat{\beta}_{(-k)}(\lambda)) \right\}$$

$$\hat{\lambda} = \arg\max_{\lambda} \left\{ CV(\lambda) \right\}$$

- Remarque : éviter minimisation AIC ou BIC.

‡. Bovelstad et al. Predicting survival from microarray data - a comparative study. Bioinformatics, 23 :2080-7, 2007.

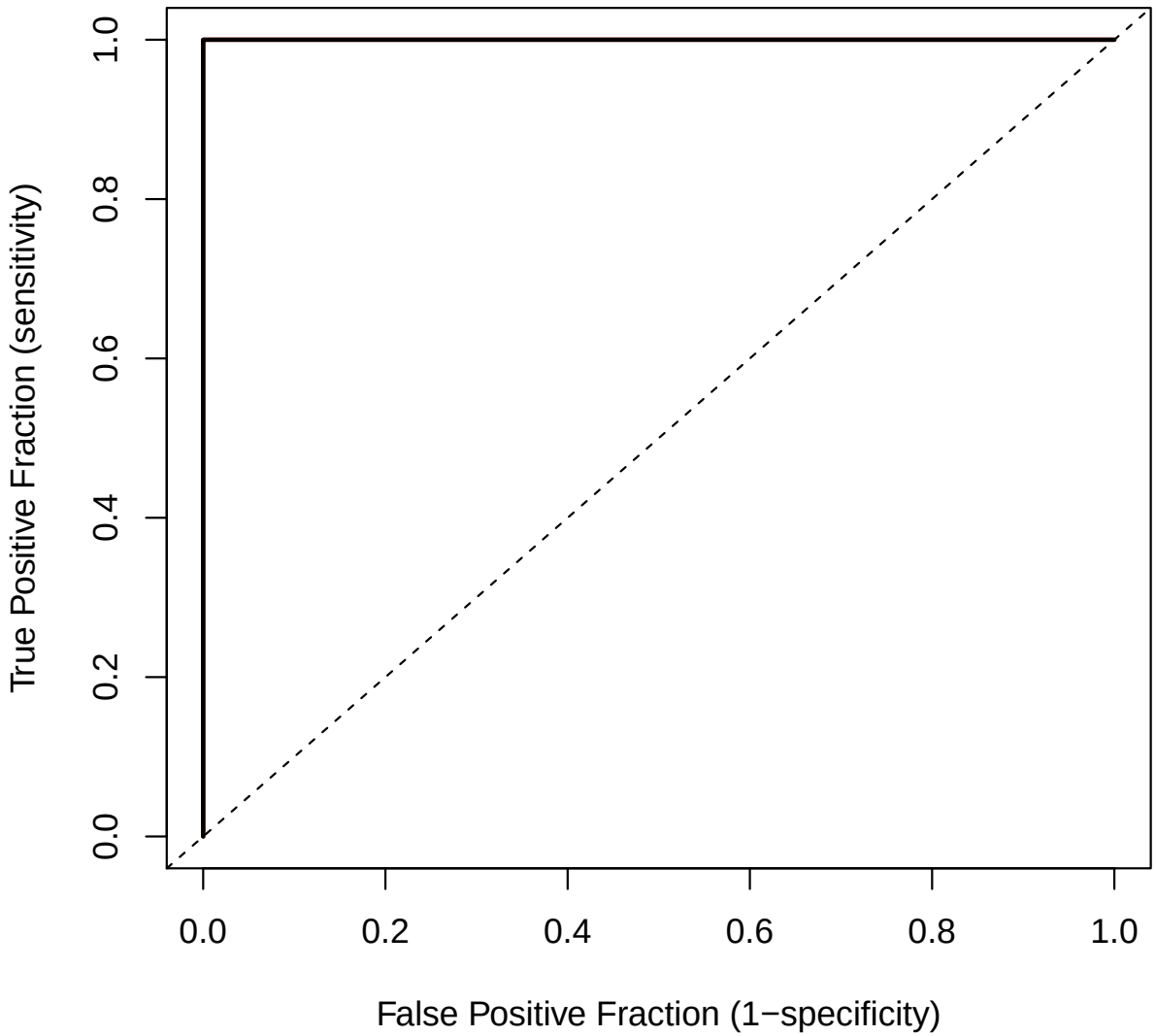
Application de la cross-validation pour estimer λ


```
> opt1 <- optL1(rem1, penalized = filtX1, model="logistic", trace = FALSE)
> opt1$lambda # 23.82089

[1] 14.80879

> # lambda =3
> pen1 <- penalized(rem1, penalized = filtX1,
+  lambda1=opt1$lambda, model="logistic", trace = FALSE)
> sum(coefficients(pen1, "all")!=0)

[1] 1

> S1<-predict(pen1,  filtX1)
> S0<-predict(pen1,  filtX0)
>
```

Application de la cross validation pour estimer λ


```
> se1<-function(x) { return(sum(1*(S1[rem1==1]>x))/sum(rem1==1)) }
> sp1<-function(x) { return(sum(1*(S1[rem1==0]<=x))/sum(rem1==0)) }
> se0<-function(x) { return(sum(1*(S0[rem0==1]>x))/sum(rem0==1)) }
> sp0<-function(x) { return(sum(1*(S0[rem0==0]<=x))/sum(rem0==0)) }
> temp<-sort(unique(c(S1, S0)))
> resultats<-data.frame(
+  sp1=1-sapply(temp, sp1),
+  se1=sapply(temp, se1),
+  sp0=1-sapply(temp, sp0),
+  se0=sapply(temp, se0),
+  seuils=temp)
> plot(c(1, resultats$sp1, 0), c(1, resultats$se1, 0),
+  xlab="False Positive Fraction (1-specificity)",
+  ylab="True Positive Fraction (sensitivity)",
+  type="n")
> lines(c(1, resultats$sp1, 0), c(1, resultats$se1, 0),
+  type="s", lwd=2, col="red3")
>  lines(c(1, resultats$sp0, 0), c(1, resultats$se0, 0),
+  type="s", lwd=2, col=1)
> abline(0,1, lty=2)
>
```

Application de la cross-validation pour estimer λ



Problèmes liés à cette séparation apprentissage et validation

- Aggravation de la faible puissance et des faibles tailles d'échantillons.
- Intervalles de confiance plus larges.
- Répétition de l'opération jusqu'à l'obtention de bons résultats.

Techniques de ré-échantillonnage