Analyse différentielle de puces à ADN. Comparaison entre méthodes wrapper et filter.

Vincent Guillemot, Laurent Le Brusquet, Arthur Tenenhaus, Vincent Frouin

Analyse différentielle de puces à ADN
Comparaison entre méthodes wrapper et filter

Vincent Guillemot [†] [‡], Laurent Le Brusquet [‡], Arthur Tenenhaus [†] , Vincent Frouin [†]

[†] CEA, iRCM, Laboratoire d'Exploration Fonctionnelle des Génomes, F-91000, France.
[‡] Supélec, Department of Signal and Electronic Systems, F-91190, France.

24 octobre 2008

Résumé

Dans le cadre de données d'expression génétique, nous nous intéressons aux méthodes qui permettent d'identifier les gènes significativement différentiellement exprimés entre deux situations biologiques. Nous allons comparer une méthode classique d'analyse par tests d'hypothèses à des méthodes d'analyse différentielle par régression régularisée. La difficulté de ce genre de jeu de données est la profusion de variables (les gènes) pour assez peu d'individus (les profils d'expression). La stratégie usuelle consiste à mettre en œuvre autant de tests qu'il y a de variables et de considérer que les variables principales sont celles qui ont la « meilleure » p-value. Une stratégie alternative pourrait consister à choisir de classer les variables non plus en fonction de leur significativité (pour un test), mais plutôt de le classer suivant leur poids dans le modèle régularisé obtenu. Dans la bibliographie, les premières méthodes sont dites *filter* [1], les deuxièmes sont plutôt dites *wrapper* [2]. Un bon aperçu de ce que sont les méthodes *wrapper* et *filter* est donné dans [9]. Le cadre ressemble à celui de l'apprentissage supervisé, car on dispose de profils d'expression géniques pour si possible l'ensemble du génome d'un organisme, chaque puce appartenant à une classe (situation biologique particulière).

L'implémentation des méthodes évoquées dans ce rapport a été effectuée sous R [16].

1 Introduction

La technologie des puces à oligonucléotides développée par la société Affymetrix permet d'observer l'activité transcriptomique d'un ensemble de cellules prélevé sur un organisme eucaryote. Après amplification et marquage par un marqueur fluorescent, l'ARN extrait de ces cellules va être déposé sur la puce, permettant aux brins d'ARN présents dans la préparation de s'hybrider aux sondes de 25 paires de bases présentes à la surface de la puce. Deux sortes de sondes appariées ont été déposées : à une sonde dite PM (Perfect Match), spécifique d'un transcrit, correspond une sonde MM (MisMatch) identique sauf au niveau de la 13^e paire de base, qui a été mutée. Ainsi les sondes PM s'hybrident spécifiquement avec un transcrit particulier, mais peuvent également s'hybrider avec des brins d'ARN non spécifiques de ce transcrit. Cette hybridation non-spécifique est mesurée à l'aide des sondes MM. Pour un même transcrit, on appelle l'ensemble des paires de sondes PM et MM le Probe set.

Le prétraitement des puces à oligonucléotides consiste en une succession d'étapes permettant une correction du bruit, une normalisation entre les lames d'un même échantillon, l'estimation de l'indice d'expression et enfin le résumé de l'ensemble des PM et MM d'un Probe set en une seule valeur caractéristique de l'expression d'un gène [7]. Ces différentes étapes sont représentées séquentiellement sur le diagramme 1 dans l'ordre suivi usuellement. Dans la suite, nous considérons que le prétraitement a déjà été effectué.

La sélection des variables d'intérêts, dont une présentation est faite par [13], est une étape clef du traitement des données de puces à ADN. Elle est notamment préliminaire à toute analyse par des techniques de classification, supervisée ou non, ainsi qu'à l'inférence de réseaux de régulation génétique. Les méthodes *wrapper* sont intéressantes selon plusieurs points de vue. D'une part, elles permettent de considérer l'ensemble de l'information présente dans le jeu de données pour procéder à la sélection

1. *filter* car on obtient un critère de sélection de gènes, mais en ne considérant les variables qu'isolément.
2. *wrapper* car on élabore un critère de sélection en prenant en compte un modèle d'interaction entre variables.

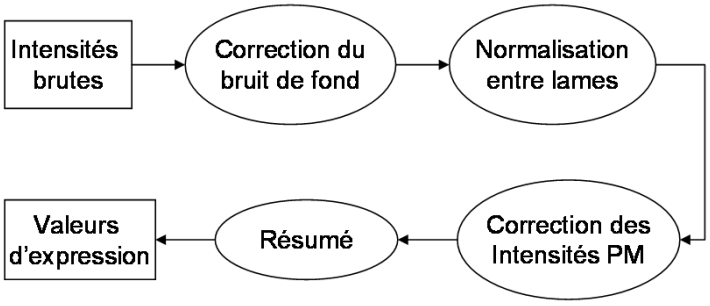


FIGURE 1 – Diagramme des différentes étapes du prétraitement pour une puce à oligonucléotides [7].

de variables. D'autre part, elles peuvent présenter des caractéristiques comparables aux méthodes *filter* (par exemple dans le cas de la régression des moindres carrés partiels, voir [4]). Il est de plus possible de calculer un modèle de prédiction permettant d'attribuer automatiquement une classe à un nouvel individu. Nous allons tout d'abord présenter les jeux de données sur lesquels nous allons travailler : ce seront des jeux de données pour lesquels nous connaissons la vérité terrain (les variables d'intérêt). Ensuite nous verrons plus en détail le fonctionnement des méthodes *filter* puis *wrapper* que nous utiliserons. Enfin nous comparerons ces algorithmes grâce à des courbes ROC.

2 Différentes méthodes d'analyse différentielle

Soit X la matrice des profils d'expression de dimensions $n \times g$ où n est le nombre d'observations (les puces à ADN) et g le nombre de gènes ou de transcrits (variables). Le but d'une analyse différentielle est de classer les transcrits dans le but de pouvoir sélectionner les gènes différentiellement exprimés entre deux populations de profils d'expression, étiquetés par la variable à expliquer y (vecteur de dimension n). y peut être une variable qualitative (par exemple pour différentier des patients sains de patients malades), ou une variable quantitative (par exemple une échelle temporelle pour une étude de cinétique ou encore une échelle de doses pour une étude de l'effet d'un composé xénobiotique sur une population de cellules). L'analyse différentielle classique, de type *filter*, est effectuée par des techniques d'analyse univariée, et plus particulièrement par des tests d'hypothèses. L'hypothèse $\mathcal{H}_0$ est qu'il n'y a aucune différence, en moyenne, entre les deux situations biologiques étudiées. Le résultat est une p-value pour chaque transcrit. Une autre approche, de type *wrapper*, est d'estimer le modèle linéaire $y = X\beta$. Une régression simple amène une infinité de solutions possibles : la matrice $X^T X$ n'est pas inversible (au plus de rang n). Une régularisation de cette matrice est donc nécessaire. Nous allons mettre en œuvre les plus classiques *i.e.* les régularisations de type L_1, L_2, L_1L_2 et PLS, et nous verrons comment calculer les coefficients de la régression pour ces méthodes :

$$LASSO \quad : \quad \hat{\beta}^{LASSO} = \arg\min_{\beta \in \mathbb{R}^g} \|y - X\beta\|_2 + \lambda_1 \|\beta\|_1 \tag{1}$$

$$Ridge \quad : \quad \hat{\beta}^{Ridge} = \arg\min_{\beta \in \mathbb{R}^g} \|y - X\beta\|_2 + \lambda_2 \|\beta\|_2 \tag{2}$$

$$L_1L_2 \quad : \quad \hat{\beta}^{L_1L_2} = \arg\min_{\beta \in \mathbb{R}^g} \|y - X\beta\|_2 + \lambda_1 \|\beta\|_1 + \lambda_2 \|\beta\|_2 \tag{3}$$

$$PLS \quad : \quad \hat{\beta}^{PLS}_{(h)} = \arg\min_{\beta \in \mathcal{K}_h} \left\{ \|y - X\beta\|^2 \right\} \tag{4}$$

où $\|x\|_1 = \sum_i |x_i|$ et $\|x\|_2 = \sqrt{\sum_i (x_i)^2}$ pour un vecteur x quelconque.

Il est d'usage de classer les gènes suivant la p-value qui leur est attribuée : les gènes dont la p-value est en-dessous d'un seuil donné sont considérés comme différentiellement exprimés. De même, nous considérons que plus le coefficient i du vecteur $\left|\hat{\beta}^{OLS}\right|$ est grand, plus le gène i est considéré comme différentiellement exprimé, il a en effet une plus grande importance dans l'explication des profils d'expression étudiés.

Nous verrons plus en détail la résolution de ces différents problèmes d'optimisation et quelques astuces utilisées pour gagner à la fois du temps de calcul et de la place en mémoire.

3 Présentation des données

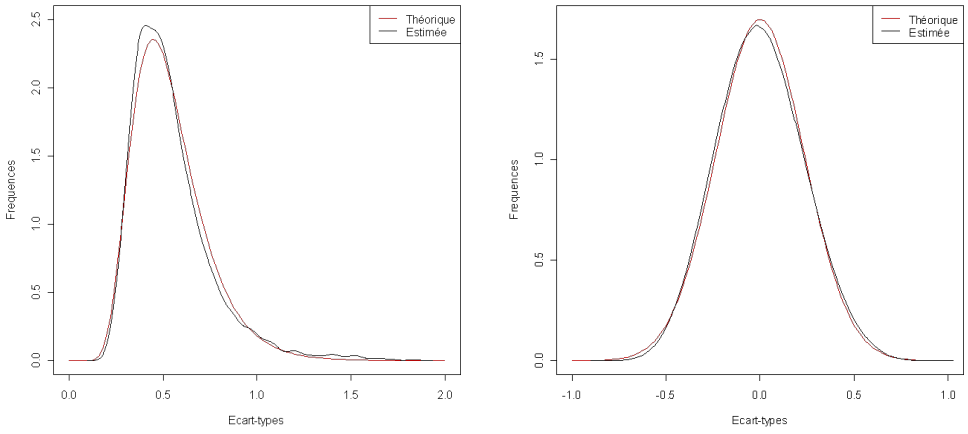
Nous appliquerons ces différentes méthodes à des jeux de données de la littérature et un jeu de données de simulation. Pour ces deux jeux de données particuliers, les gènes différentiellement exprimés sont parfaitement connus. Les paramètres de ces jeux de données sont résumés dans le tableau 3.

Jeu de données	Nombre d'individus	Nombre de gènes	Nombre de gènes différentiellement exprimés
Simulation	20	1000	100
Spike In	14	22300	42

TABLE 1 – Paramètres des jeux de données étudiés.

3.1 Données de simulation

Dans toute étude transcriptomique, la méconnaissance de la vérité terrain (*i.e.* les gènes qui seront différentiellement exprimés) rend la qualification d'une analyse différentielle hasardeuse. Il s'agit donc, dans cette partie, de décrire une structure de données s'approchant au mieux, sur le plan général, de données réelles. On se place dans le cas où le jeu de données de simulation est partagé en deux classes, représentant deux phénomènes biologiques différents. Soit k ($k = 1, 2$) la classe à laquelle appartient le profil X_k, $X_k \sim \mathcal{N}(\boldsymbol{\mu}_k, \Sigma)$, Σ, la matrice de variance-covariance des profils, permettant de modéliser les interactions entre gènes. Pour générer Σ, nous choisissons d'observer empiriquement la distribution des termes de la matrice de covariance empiriques de donnes réelles. La densité estimée des termes de la matrice de corrélation empirique de données réelles est représentée sur la figure 2(b). Un test de Kolmogorov-Smirnov nous indique, avec une confiance de 46%, que ce profil est Gaussien. De la même façon, nous pouvons estimer que la distribution des écart-type du jeu de données de Golub et al. [8] s'approche, avec une confiance de 30% [3], d'une loi lognormale (cf. figure 2(a)).



(a) Densités empiriques et estimées pour les écart-types des gènes

(b) Densités empiriques et estimées pour les termes de la matrice de corrélation

FIGURE 2 – Estimation empirique de la densité des variances des gènes ainsi que celle des termes de la matrice de corrélation. Les données sont issues d'une étude sur le diagnostic par puces à ADN de deux types de cancer très proches au niveau symptomatique [8].

La procédure de génération de données simulées va donc être la suivante : nous allons tout d'abord générer une matrice de corrélation entre gènes, de taille $g \times g$, et la transformer en matrice de variance-covariance, en s'assurant notamment que le résultat est défini positif (sinon la matrice calculée est rendue définie positive). Puis nous allons générer les profils d'expression des deux classes, avec des vecteurs moyennes qui seront différents suivant la classe :

3. Les p-values calculées (46% et 30%) sont empiriquement représentatives d'échantillons suivant les lois testées, quand un échantillon est par exemple tiré d'une loi uniforme et comparé à une loi gaussienne, on obtient des p-values significativement nulles (de l'ordre de 10^{-16})

– si $k = 1$, chaque variable aura une moyenne nulle
– si $k = 2$, 10% des variables auront une moyenne non nulle constante

Pour plus de clarté, nous avons représenté schématiquement une matrice de profils d'expression sur la figure 3.

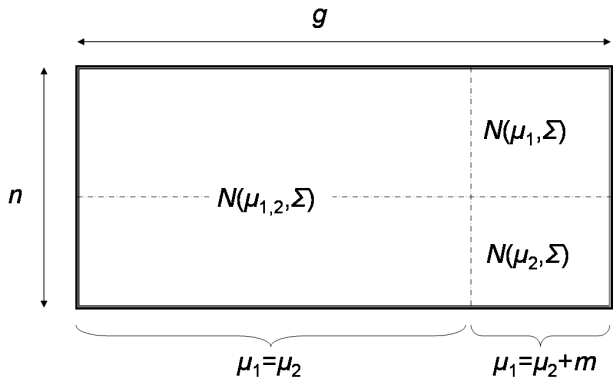


FIGURE 3 – Représentation d'une matrice de profils d'expression

L'hypothèse forte sous-tendue ici est que la matrice de covariance d'un ensemble de transcrits est le reflet direct de la régulation entre gènes. Une autre approche effectuée dans la littérature consiste à considérer plutôt la matrice des corrélations partielles entre gènes et d'en déduire la matrice de variance-covariance [19]. Un coefficient de corrélation partielle entre deux gènes est plus facilement interprétable qu'un simple coefficient de corrélation. Cependant partir d'une matrice de corrélations partielles pour générer des données *in silico* ne donne par pour l'instant des profils statistiques satisfaisants (correspondant aux densités représentées sur les figures 2(a) et 2(b)), et surtout aisément paramétrables, c'est pourquoi nous ne présenterons par ici cette technique de simulation.

3.2 Données « Spike In »

L'étude Spike in a été développée par Affymetrix pour valider l'algorithme de normalisation MAS5 à partir de fragments d'ARN correspondant à 16 Probe sets de la puce GeneChip HG-U95A. Ces fragments ont été ajoutés à des préparations d'ARN dont les concentrations sont des puissances de 2 allant de 0.125 à 512 pM [3]. La même source d'ARN a été utilisée dans tous les cas pour la préparation d'ARN. Ainsi, un petit nombre de gènes seront différentiellement exprimés tout en étant mélangés à une population classique d'ARN qui est identique pour toutes les puces. L'organisation du plan expérimental est un carré latin [1], [12], comme le représente la figure 4.

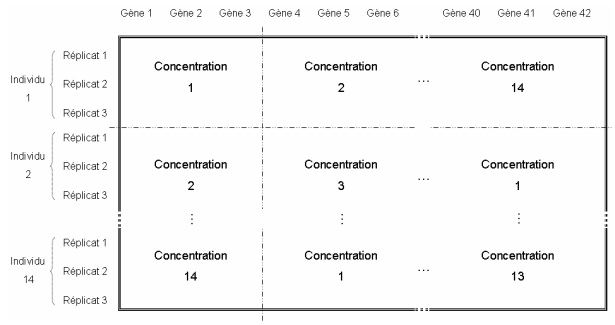


FIGURE 4 – Organisation en carré latin des données Spike In (hgu133)

Une classe de puces est ainsi constituée de 3 réplicats, qui correspondent à une certaine configuration de concentrations des 42 gènes d'intérêt. Avec ce jeu de données, on peut donc former 14 classes de 3 puces, chaque classe présentant une configuration de concentrations différente.

3. pM est l'abréviation de picoMolaire. $1\ M = 1\ mol.L^{-1}$

4 Méthodes *filter*

On fait le plus souvent l'hypothèse gaussienne pour l'évolution de l'expression d'un gène dans chaque classe. Cette hypothèse est en pratique rarement vérifiée, et parfois non vérifiable, du fait de la très petite taille des échantillons à étudier. On utilise donc parfois des tests non paramétriques, avec le risque de perdre de la puissance. Les deux tests les plus fréquemment utilisés dans la communauté scientifique produisant des puces à ADN sont le test de Student et le test de Wilcoxon-Mann-Whitney (par exemple implémentés dans le logiciel ArrayAssist pour le traitement de puces Affymetrix).

Pour la présentation des deux tests de comparaison de deux échantillons indépendants, on considérera deux échantillons x_1 et x_2 de taille respective n_1 et n_2, dont les moyennes empiriques seront notées $\overline{x}_1$ et $\overline{x}_2$ et les écarts-types empiriques s_1 et s_2.

4.1 Un test paramétrique : le test de Student

C'est en fait une variante du test de Student, le test d'Aspin-Welch [20], que l'on utilise en général. La statistique calculée est la suivante :

$$t' = \frac{\overline{x}_1 - \overline{x}_2}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}} \tag{5}$$

Sous l'hypothèse $\mathcal{H}_0$, et à condition que les échantillons suivent une loi gaussienne, la statistique t' suit une loi de Student dont le nombre de degrés de liberté m est approximé par la formule suivante :

$$\frac{1}{m} = \frac{c^2}{n_1 - 1} + \frac{(1-c)^2}{n_2 - 1} \tag{6}$$

avec :

$$c = \frac{s_1^2/n_1}{s_1^2/n_1 + s_2^2/n_2} \tag{7}$$

4.2 Un test non paramétrique : le test de Wilcoxon-Mann-Whitney [17]

L'hypothèse forte de normalité des échantillons n'est le plus souvent pas respectée, c'est pourquoi le test de Mann-Whitney sur la somme des rangs est parfois utilisé.

Il consiste à calculer, pour les échantillons réordonnés, le nombre de couples $(x_1^{(i)}, x_2^{(j)})$ tels que $x_1^{(i)} > x_2^{(j)}$ (ici nos variables sont quantitatives). On calcule ainsi une statistique $U = \text{card}\left\{(x_1^{(i)}, x_2^{(j)}), x_1^{(i)} > x_2^{(j)}\right\}$ dont on peut montrer, si les échantillons sont issus de la même population (*i.e.* sous l'hypothèse $\mathcal{H}_0$), que

$$\begin{aligned} \mathbb{E}(U) &= \frac{n_1 n_2}{2} \\ V(U) &= \frac{n_1 n_2 (n_1 + n_2 + 1)}{12} \end{aligned}$$

De plus, si n_1 et n_2 sont tous les deux supérieurs à 8, cette statistique U suit approximativement une loi Gaussienne. Dans tous les cas, on peut calculer la loi exacte de U.

Pour des échantillons appariés, on préférera un test de Wilcoxon (qui peut-être également plus rapide dans certains cas) : les échantillons x_1 et x_2 sont mélangés et l'échantillon global obtenu est ordonné. La somme des rangs, W, des éléments de l'échantillon x_1 suit une loi dite de Wilcoxon pour de petits échantillons, qui peut être approximée par une loi gaussienne quand les échantillons sont de taille supérieure à 10. Sous l'hypothèse nulle :

$$\begin{aligned} \mathbb{E}(W) &= \frac{n_1 (n_1 + n_2 + 1)}{2} \\ V(W) &= \frac{n_1 n_2 (n_1 + n_2 + 1)}{12} \end{aligned}$$

4.3 Correction de l'effet « tests multiples »

Une fois la p-value calculée, il convient d'appliquer des stratégies d'estimation du taux de faux positifs, ou False Discovery Rate (FDR), du fait de la multitude de tests qui sont effectués les uns à la suite des autres. Le tableau 2 contient les notations que nous utiliserons après avoir mené un ensemble de g tests statistiques d'une hypothèse nulle.

	$\mathcal{H}_0$ acceptée	$\mathcal{H}_0$ rejetée	total
$\mathcal{H}_0$ vraie	W	V	g_0
$\mathcal{H}_0$ fausse	T	S	$g - g_0$
	$g - R$	R	g

TABLE 2 – Résultats d'un ensemble de tests d'hypothèse.

Par définition, $FWER = \Pr(V \geq 1)$ (Family Wise Error rate) et $FDR = \mathbb{E}(V/R)$ (False Discovery Rate). La procédure de Benjamini et Hochberg [2] permet un contrôle du FDR et du $FWER$ dans le cas de variables corrélées (ce qui est le cas pour des données d'expression génétique). Nous la préférerons à des procédures plus « stringentes »telles celles de Holm, Hommel, Bonferroni ou encore Benjamini et Yekutieli [3] qui ne changent rien, de façon générale, au classement des transcrits, mais qui saturent très vite à 1.

La procédure de Benjamini et Hochberg est la suivante :

```
Contrôle du FDR au niveau alpha, g est le nombre de tests effectués
Ranger les p-values dans l'ordre croissant et les numéroter
Accepter l'hypothèse nulle pour les p_values  de numéro i, p(i) < (i/g)*alpha
Rejeter les autres
```

Cette procédure ne changeant strictement pas l'ordre des p-values calculées, et comme nous ne cherchons pas encore à filtrer les gènes d'intérêt, nous nous contenterons des tests précédemment explicités.

5 Méthodes *wrapper* : quelques régressions régularisées

Les tests d'hypothèses ne peuvent prendre en compte la corrélation qui existe entre l'expression des gènes d'un jeu de données, au contraire des méthodes *wrapper*. Nous allons considérer 4 méthodes *wrapper* qui estiment de façon différente les coefficients d'un modèle linéaire. Pour chaque estimateur ($\hat{\beta}^{LASSO}$,$\hat{\beta}^{Ridge}$,$\hat{\beta}^{L_1L_2}$ et $\hat{\beta}^{PLS}$), nous allons présenter le critère qu'il optimise, si possible son expression en fonction de l'estimateur des moindres carrés $\hat{\beta}^{OLS} = (X^TX)^{-1}X^Ty$ et, de façon succincte, l'algorithme utilisé pour résoudre le problème d'optimisation.

Notations :
- X jeu de données ou variables explicatives ou encore matrice des covariables
- y variable à expliquer, dans notre cas, elle est supposée binaire
- g nombre de variables (de gènes)
- n le nombre d'échantillons (d'expériences)

Les 4 régressions régularisées que nous allons présenter sont les régressions LASSO, Ridge, L1-L2 et PLS.

5.1 Régression L1 ou LASSO

LASSO, « Least Absolute Shrinkage and Selection Operator » [21], est une méthode itérative initialisée à un $\hat{\beta}^{LASSO}$ nul et convergeant en un nombre d'itérations très inférieur à g. Elle permet de sélectionner en théorie l'ensemble des variables explicatives indépendantes qui sont le plus corrélées à la variable à expliquer.

5.1.1 Critère à optimiser et expression de l'estimateur LASSO

L'optimisation à effectuer est la suivante :

$$\hat{\beta}^{LASSO} = \arg\min_{\beta \in \mathbb{R}^g} \|y - X\beta\|_2 + \lambda_1 \|\beta\|_1 \tag{8}$$

L'estimateur LASSO $\hat{\beta}^{LASSO}$ s'exprime suivant l'estimateur des moindres carrés, dans le cas simple où la matrice $X^T X$ est diagonale [15] :

$$\hat{\beta}_i^{LASSO} = \text{sign}\left(\hat{\beta}_i^{OLS}\right) \max\left(0, \left|\hat{\beta}_i^{OLS}\right| - \lambda_1/2\right), i = 1, \ldots, g \tag{9}$$

Toujours selon [15], il est possible d'approximer l'estimateur LASSO dans le cas plus général où le nombre de variables est inférieur au nombre d'individus. Dans la configuration particulière $n \ll g$, il n'existe pas pour l'instant d'expression simple de $\hat{\beta}^{LASSO}$ en fonction de $\hat{\beta}^{OLS}$.

5.1.2 Progression de l'algorithme

Nous allons présenter l'implémentation particulière effectuée dans [6] : l'algorithme itératif LARS (Least Angle Regression). Notations :

- $\mathcal{A}$ un sous-ensemble de $\{1, \ldots, m\}$, ensemble des indices des covariables indépendantes entre elles ; on pose $|\mathcal{A}| = k$
- $X_{\mathcal{A}} = (\ldots s_j \mathbf{x}_j \ldots)_{j \in \mathcal{A}}$ matrice des covariables sélectionnées, multipliées par $s_j = \pm 1$ de dimensions $n \times k$ (n est le nombre d'individus, k est la taille de $\mathcal{A}$)
- $\mathcal{G}_{\mathcal{A}} = X'_{\mathcal{A}} X_{\mathcal{A}}$ de dimensions $k \times k$
- $A_{\mathcal{A}} = (\mathbb{1}'_{\mathcal{A}} \mathcal{G}_{\mathcal{A}} \mathbb{1}_{\mathcal{A}})^{-1/2}$, un scalaire.
- $\mathbb{1}_{\mathcal{A}}$ dimensions $k \times 1$
- $w_{\mathcal{A}}$ dimensions $k \times 1$
- $\hat{\beta}_i$, de dimensions $g \times 1$, est le vecteur des coefficients du modèle de régression estimé à l'itération i, dont seulement k composantes ont été calculées ; pour alléger l'écriture, nous omettrons de préciser qu'il s'agit en fait d'un $\hat{\beta}^{LASSO}$

Le premier concept important est : dans quelle direction faire « avancer » le vecteur des $\hat{\beta}$ à chaque itération. le principe est de sélectionner l'ensemble des variables identifiées comme étant les plus corrélées avec la variable à expliquer tout en étant indépendantes entre elles. On avancera dans la direction qui ne favorise aucune de ces variables :

$$\hat{\beta}_{i+1} = \hat{\beta}_i + \hat{\gamma} u_i \tag{10}$$

u_i est le vecteur équiangulaire combinaison linéaire des covariables à l'itération i.

Ainsi, u va être choisi unitaire et combinaison linéaire d'un sous-ensemble $\mathcal{A}$ des covariables. Ce sous-ensemble est augmenté itérativement. Pour la suite, nous caractériserons une itération non plus par son numéro (i, dans l'équation 10), mais par le sous-ensemble de covariables qui ont été sélectionnées :

$$\hat{\beta}_{\mathcal{A}^+} = \hat{\beta}_{\mathcal{A}} + \hat{\gamma}_{\mathcal{A}^+} u_{\mathcal{A}^+} \tag{11}$$

Le vecteur $u_{\mathcal{A}^+}$ est une combinaison linéaire des covariables indicées dans $\mathcal{A}^+$: $X_{\mathcal{A}^+} w_{\mathcal{A}^+}$. Il doit avoir une norme unitaire et surtout doit être tel que sa corrélation avec toutes les covariables sélectionnées est la même. En considérant ces contraintes :

$$\begin{aligned} u_{\mathcal{A}^+} &= X_{\mathcal{A}^+} w_{\mathcal{A}^+} \\ u'_{\mathcal{A}^+} u_{\mathcal{A}^+} &= 1 \\ X'_{\mathcal{A}^+} u_{\mathcal{A}^+} &= A_{\mathcal{A}^+} \mathbb{1}_{\mathcal{A}^+} \end{aligned}$$

on obtient $w_{\mathcal{A}^+} = A_{\mathcal{A}^+} \mathcal{G}_{\mathcal{A}^+}^{-1} \mathbb{1}_{\mathcal{A}^+}$, avec $A_{\mathcal{A}^+} = (\mathbb{1}'_{\mathcal{A}^+} \mathcal{G}_{\mathcal{A}^+} \mathbb{1}_{\mathcal{A}^+})^{-1/2}$ et $\mathcal{G}_{\mathcal{A}^+} = X'_{\mathcal{A}^+} X_{\mathcal{A}^+}$.

5.1.3 Choix de l'ensemble des covariables actives $\mathcal{A}$

Pour déterminer cet ensemble, les covariables qui ont la plus grande corrélation avec les résidus sont sélectionnées, une par une, à chaque itération. Soit $\hat{c} = (c_j)_{j \in \mathcal{A}} = X'(y - \hat{\beta}_{\mathcal{A}})$. On aura alors $\mathcal{A}^+ = \{j : |\hat{c}_j| = \max_j |\hat{c}_j|\}$. Au passage, on pose $\widehat{C} = \max_j |\hat{c}_j|$ et $a = X'_{\mathcal{A}} u_{\mathcal{A}}$.

5.1.4 Choix du pas de variation d'une itération à l'autre

$\hat{\gamma}$ se calcule ainsi sur les covariables sélectionnées à l'étape courante :

$$\hat{\gamma} = \min_{j \in \mathcal{A}^c}{}^{+} \left\{ \frac{\widehat{C} - \hat{c}_j}{A_{\mathcal{A}} - a_j}, \frac{\widehat{C} + \hat{c}_j}{A_{\mathcal{A}} + a_j} \right\} \tag{12}$$

$\min^+$ signifie que le minimum n'est calculé que sur les composantes positives, quel que soit j.

$\hat{\gamma}$ est le plus petit réel positif tel qu'une nouvelle covariable d'indice $\hat{j}$ rejoigne l'ensemble des covariables actives $\mathcal{A}$ [6].

5.1.5 Modification de LARS pour LASSO

[6] ont montré que quelques modifications de LARS permettent de mettre en œuvre LASSO. On pose $\tilde{\gamma} = \min_{\gamma_j > 0} \gamma_j$. Pour des raisons de changement de signe entre $\hat{\beta}_j$ et c_j, on ne peut pas avoir de γ supérieur à $\tilde{\gamma}$. Si cela arrive, l'algorithme LASSO enlève la covariable en cause (disons j_0), et fixe pour de bon son coefficient à 0 ; de plus, l'incrémentation de $\hat{\beta}$ se fera avec un poids $\tilde{\gamma}$:

$$\hat{\beta}_{\mathcal{A}^+} = \hat{\beta}_{\mathcal{A}} + \tilde{\gamma}_{\mathcal{A}^+} u_{\mathcal{A}^+} \text{ et } \mathcal{A}^+ = \mathcal{A} - j_0 \tag{13}$$

5.2 Régression L2 : ridge.lm Ridge

5.2.1 Critère à minimiser et expression de l'estimateur

L'obtention de l'estimateur Ridge se fait en minimisant un critère quadratique avec une contrainte quadratique :

$$\hat{\beta}^{Ridge} = \arg\min_{\beta \in \mathbb{R}^g} \|y - X\beta\|_2 + \lambda_2 \|\beta\|_2 \tag{14}$$

Le critère à minimiser est convexe et différentiable, ainsi on peut donc exprimer facilement l'estimateur Ridge en fonction de X, y, ou encore en fonction de $\hat{\beta}^{OLS}$: $\hat{\beta}^{Ridge} = (X^T X + \lambda_2 I_g)^{-1} X^T y$ où I_g est la matrice identité de taille $g \times g$, et donc $\hat{\beta}^{Ridge} = (X^T X + \lambda_2 I_g)^{-1} X^T X \hat{\beta}^{OLS}$.

5.2.2 Algorithme utilisé

L'algorithme utilisé couramment (par exemple dans le package MASS de R) propose d'évaluer une décomposition en valeurs singulières de la matrice des données X : $X = X_s = UDV^T$, de sorte que

- U est de dimensions $n \times n$ et $U^T U = I_n$
- V est de dimensions $p \times n$ et $V^T V = I_n$
- D est une matrice réelle diagonale de dimensions $n \times n$

Ainsi, pour le cas dans lequel nous nous sommes placés ($n \ll p$), l'inversion de matrice est grandement facilitée, puisqu'on peut exprimer de nouveau $\hat{\beta}^{Ridge} = V(D^2 + \lambda_2 I_n)^{-1} UDy$.

5.3 Régression L1-L2 : la solution « elastic net » [22]

Soit λ_1 et λ_2 les deux paramètres de régularisation pour lesquels l'optimisation doit être effectuée :

$$\hat{\beta}^{L_1 L_2} = \arg\min_{\beta \in \mathbb{R}^g} \|y - X\beta\|_2 + \lambda_1 \|\beta\|_1 + \lambda_2 \|\beta\|_2 \tag{15}$$

On définit une nouvelle matrice de données X^* ainsi qu'une nouvelle variable explicative y^* :

$$X^*_{(n+p)\times p} = (1+\lambda_2)^{-1/2} \begin{pmatrix} X \\ \sqrt{\lambda_2} I_p \end{pmatrix}, y^*_{(n+p)} = \begin{pmatrix} y \\ 0 \end{pmatrix} \tag{16}$$

Si on pose de plus $\tilde{\lambda} = \lambda_1/(1+\lambda_2)$ et $\beta^* = \sqrt{1+\lambda_2}\beta$, alors on peut récrire l'estimateur en un nouvel estimateur de type LASSO :

$$\hat{\beta}^{L_1 L_2} = \arg\min_{\beta^* \in \mathbb{R}^g} \|y^* - X^* \beta^*\|_2 + \tilde{\lambda} \|\beta^*\|_1 \tag{17}$$

Il en ressort que toutes les propriétés intéressantes (sélection de variables) de l'algorithme de LASSO sont également transposables à l'algorithme elastic net. La mise en œuvre en pratique est donc également la même.

5.4 Régression PLS

Notons $s = X^T y$, $C = X^T X$ et $K_h = [s\ Cs\ C^2 s\ ...\ C^{h-1}s]$, qui est la matrice (dite de Krylov) dont les colonnes sont formées par les vecteurs $C^i s$, $i = 1, \ldots, h-1$.

L'idée générale de la régression PLS est d'extraire des données X un ensemble de composantes orthogonales $\mathcal{T} = t_1, ..., t_m$ qui serviront de nouveaux prédicteurs pour prédire la variable à expliquer y. La première composante PLS t_1 est construite de manière à maximiser la covariance avec y ; les composantes suivantes t_h maximisent également la covariance avec y avec la contrainte additionnelle d'être orthogonale au h-1 composantes précédentes.

5.4.1 Expression de l'estimateur PLS

On peut montrer que $\hat{\beta}_h^{PLS}$ peut s'exprimer par :

$$\begin{aligned} \hat{\beta}_h^{PLS} &= K_h (K_h^T C K_h)^{-1} K_h^T s && (18) \\ &= K_h (K_h^T C K_h)^{-1} K_h^T C C^{-1} s && (19) \\ &= K_h (K_h^T C K_h)^{-1} K_h^T C \hat{\beta}^{OLS}. && (20) \end{aligned}$$

La démonstration de la première égalité 19 est donnée, par exemple, dans [10, 14]. On en déduit que $\hat{\beta}_h^{PLS}$ s'obtient par résolution du problème d'optimisation suivant :

$$\hat{\beta}_h^{PLS} = \underset{\beta \in \mathcal{K}_h(C,s)}{\operatorname{argmin}} \|y - X\beta\|^2 \quad (21)$$

où $\mathcal{K}_h(C, s) = \text{vect}\{s, Cs, C^2 s, ..., C^{h-1}s\}$

5.4.2 Algorithme : SIMPLS

L'algorithme présenté succinctement ici est inspiré de celui exposé par [5]. À l'état initial, on a $A_0 = X^T y$, $M_0 = X^T X$ et $C_0 = I$. Pour k allant de 1 à h (h étant le nombre de composantes PLS que l'on souhaite calculer) :

1. Calculer q_k, le vecteur propre dominant de la matrice $A_k A_k^T$
2. $w_k = A_k q_k$, $c_k = w_k^T M_k w_k$, $w_k = w_k / \sqrt{c_k}$
3. $p_k = M_k w_k$, $q_k = A_k^T w_k$
4. $v_k = C_k p_k$ et $v_k = v_k / \|v_k\|$
5. $C_{k+1} = C_k - v_k v_k^T$ et $M_{k+1} = M_k - p_k p_k^T$
6. $A_{k+1} = C_k A_k$

6 Résultats

Nous évaluons la qualité de l'analyse différentielle pour un gène donné en comparant
- soit la p-value qui lui est attribuée
- soit la valeur absolue du coefficient de régression qui lui est associée

avec le caractère binaire « différentiellement exprimé » connu. Pour pouvoir observer cette comparaison, nous avons décidé d'utiliser des courbes ROC.

6.1 Choix des paramètres de régularisation

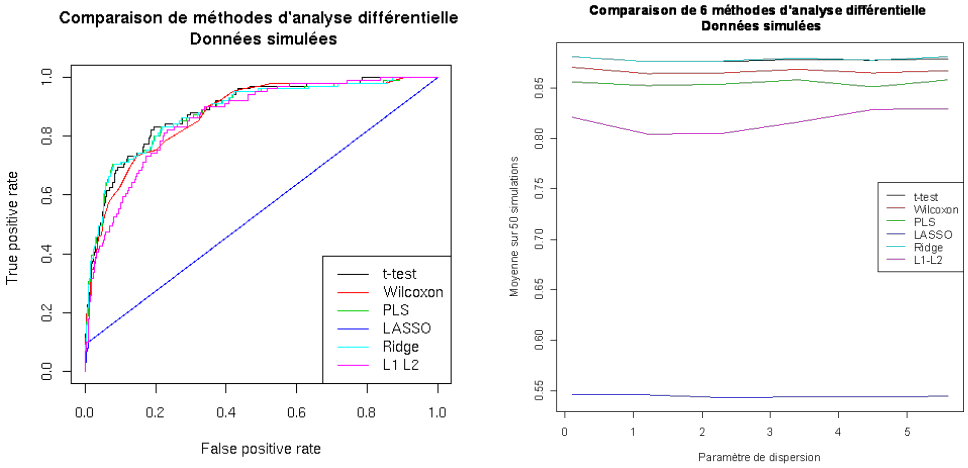
L'estimation du modèle de régression linéaire nécessite l'ajustement de coefficients de régularisation :
- un seul pour contrôler la norme de l'estimateur pour les régressions LASSO et Ridge
- deux pour contrôler conjointement les normes de type 1 et 2 de l'estimateur L1-L2
- le nombre de composantes PLS pour la régression PLS

Les valeurs de ces coefficients qui minimisent l'erreur quadratique moyenne de prédiction sont déterminées par validation croisée.

6.2 Résultats sur les données de simulation

La figure 5(a) permet de visualiser les performances en termes de spécificité et de sensibilité des différentes méthodes d'analyse différentielle que nous avons abordées :

- les test d'hypothèses et les régressions Ridge et PLS ont des performances équivalentes.
- LASSO ne semble toujours adapté qu'à la sélection d'un nombre extrêmement restreint de gènes différentiellement exprimés.
- avec la régression L1-L2, nous avions espéré combiner les avantages des régressions Ridge et LASSO, mais ce n'est pas le cas. Les paramètres de régularisation qui ont été choisis par validation croisée ne permettent ni de garder un nombre restreint de gènes, ni de bénéficier des bonnes performances de la régression Ridge. Sans compter que cette méthode est très gourmande en temps de calcul.



(a) Exemple de courbes ROC pour différentes méthodes d'analyse différentielle

(b) Aires en dessous des courbes pour les méthodes étudiées

FIGURE 5 – La comparaison des méthodes détaillées, à gauche sous la forme de courbes ROC, à droite en termes d'AUC.

Sur la figure 5(b), on peut observer l'évolution des moyennes des AUC pour les méthodes utilisées pour un certain nombre de simulations différentes (on choisit dix matrices de corrélation différentes en termes de dispersions de leurs coefficients [4], et pour chacune de ces matrices, on effectue 50 simulations différentes, pour chaque simulation, on peut calculer une aire en dessous de la courbe ROC (AUC Area Under Curve), puis on en calcule la moyenne sur ces 50 simulations).

6.3 Résultats sur les données Spike In

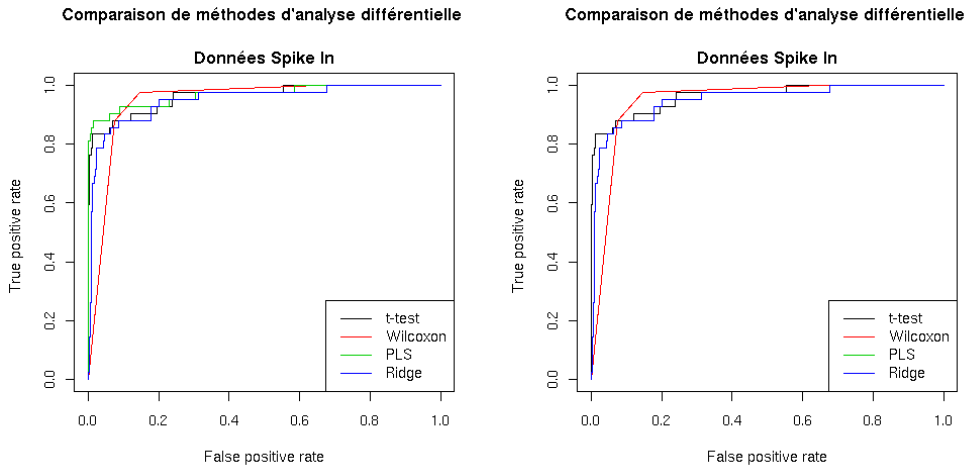
La figure 6 présente les résultats obtenu pour un exemple (*i.e.* une configuration particulière) de deux classes de trois puces sur lesquelles sont effectuées les analyses différentielles que nous avons estimées les meilleures, on remarque que ces performances sont assez différentes :

La différence entre les courbes 6(a) et 6(b) est la standardisation des données. La régression PLS fonctionne, dans le cas des données Spike In, bien mieux sur des données qui ne sont pas standardisées, et donne même de meilleurs résultats que les autres méthodes.

Enfin, et ce qui n'est toujours pas explicable par des données simulées, lorsque l'on considère en moyenne les performances des deux méthodes sur l'ensemble des configurations possibles pour les données Spike In, la PLS-R est plus efficace. Nous pouvons effectuer les remarques suivantes, d'après les comportements des différentes méthodes étudiées :

- quand les données sont centrées et réduites : le modèle PLS à 1 composante équivaut à un t-test [4], le modèle PLS à un nombre optimal de composantes équivaut à la régression Ridge

4. Cette dispersion est en fait déterminée par l'écart-type de la loi des coefficients de la matrice de variance-covariance. Mais comme cette première matrice, générée aléatoirement, subit une transformation en vue de devenir définie positive, nous préférons utiliser le terme dispersion plutôt qu'écart-type.



(a) Exemple de courbes ROC, les données ne sont pas centrées réduites

(b) Exemple de courbes ROC, les données sont centrées réduites

FIGURE 6 – La comparaison des méthodes les plus performantes, sur un exemple issu des données Spike In. Les données sont normalisées 6(a) ou non 6(b) : cela a une influence sur les performances de la PLS.

– quand les données ne sont pas réduites : le modèle à 1 composante équivaut au *Fold-Change*[5], le modèle PLS à un nombre de composantes optimal ne ressemble à aucune méthode étudiée, mais semble se situer à mi-chemin, en termes de performances, entre le Fold-Change et le t-test

7 Conclusion et perspectives

Méthodes utilisée : La comparaison que nous avons effectuée entre méthodes *wrapper* et méthodes *filter* ne nous permettent pas de conclure nettement à une supériorité des unes par rapport aux autres :

– théoriquement, on peut montrer que la plupart des méthodes utilisées sont équivalentes lorsque certaines hypothèses (notamment l'indépendance entre variables) sont vérifiées : elles permettent de sélectionner les variables explicatives les mieux corrélées avec la variable à expliquer
– la régression PLS ou la régression Ridge donnent, sur des données simulées, les mêmes performances que les tests d'hypothèses
– sur des données non simulées, le résultat est plus mitigé : il semblerait que les méthodes *wrapper* soient plus performantes. Mais il faut bien garder à l'esprit que les profils d'expression présents dans les données Spike In sont assez caricaturaux : seulement une poignée de gènes sont différentiellement exprimés, et avec des concentrations très fortes, qui ne s'observent pas dans des données réelles de la littérature. La solution pour se sortir de ce cadre serait plutôt d'utiliser des modèles de classification qui permettraient de rendre compte d'autres sortes de liens entre gènes que de la simple corrélation d'expression (linéaire), par exemple en utilisant des méthodes de régressions plus élaborées ou en introduisant des noyaux : les méthodes de régression régularisées comme Ridge ou PLS peuvent être « kernélisées » [18], [11].

Simulations : Plusieurs paramètres pourront être modulés de façon à se rapprocher de données réelles :

– le caractère différentiellement exprimé est constant en moyenne, alors qu'il est certainement plus diffus
– la proportion de gènes différentiellement exprimés est à discuter, bien que la littérature s'accorde à ne considérer que 5 ou 10% de gènes différentiellement exprimés
– les profils statistiques que nous avons utilisés pour générer les matrices de corrélation représentent de façon simpliste le fait qu'il existe des liens entre gènes. On pourrait avoir comme point de départ un réseau de régulation, inspiré de la réalité, qui déterminerait les interactions entre gènes.

5. En respectant les notations du paragraphe sur les méthodes *filter*, le Fold-Change FC, pour un gène donné est calculé simplement de la manière suivante :

$$FC = \overline{x}_1 - \overline{x}_2 \tag{22}$$

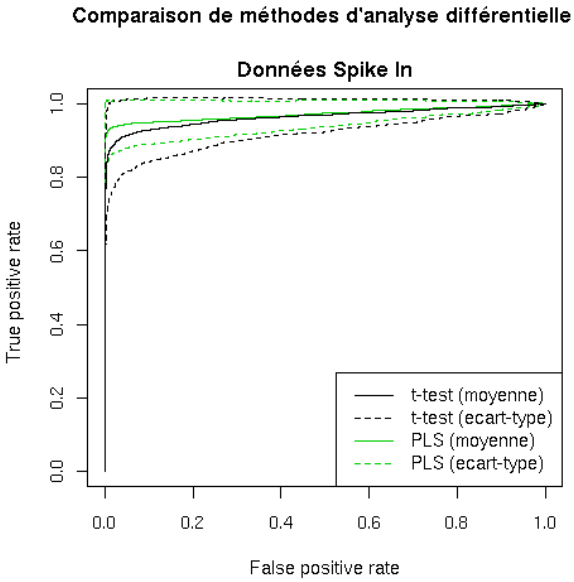


FIGURE 7 – Moyennes des différentes courbes ROC pour toutes les configurations possibles : comparaison entre le t-test et la régression PLS (sans standardisation)

Comment comparer des analyses différentielles sur des jeux de données réels : Quand la réalité n'est pas connue, la méthode classique est de comparer les listes de gènes après des analyses différentielles entre elles. Plutôt que d'adopter cette approche de comparaison relative, il serait préférable d'adopter une méthode plus absolue qui consiste à comparer les erreurs de généralisation après élaboration d'un modèle de classification : cette méthode n'est pas applicable aux tests d'hypothèses et présuppose de travailler dans un contexte de validation croisée, cela nécessite donc des jeux de données d'aux moins quelques dizaines de profils d'expression dans chaque classe pour être cohérent.

Références

[1] AffyTeam, *Guide to probe logarithmic intensity error (plier) estimation*, Tech. report, Affymetrix, 2005.

[2] Y. Benjamini and Y. Hochberg, *Controlling the false discovery rate : a practical and powerful approach to multiple testing*, Journal of the Royal Statistical Society B **57** (1995), 289–300.

[3] Yoav Benjamini and Daniel Yekutieli, *Quantitative trait loci analysis using the false discovery rate.*, Genetics **171** (2005), no. 2, 783–790.

[4] Anne-Laure Boulesteix, *Pls dimension reduction for classification with microarray data.*, Stat Appl Genet Mol Biol **3** (2004), Article33.

[5] S. de Jong, *Simpls : An alternative approach to partial least squares regression.*, Chemometrics and Intelligent Laboratory Systems **18** (1993), 251–253.

[6] Bradley Efron, Trevor Hastie, Lain Johnstone, and Robert Tibshirani, *Least angle regression*, (2002).

[7] Johannes M. Freudenberger, *Comparison of background correction and normalization procedures for high-density oligonucleotide microarrays*, Ph.D. thesis, UniversitÃd't Leipzig, January 2005.

[8] T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, C. D. Bloomfield, and E. S. Lander, *Molecular classification of cancer : class discovery and class prediction by gene expression monitoring.*, Science **286** (1999), no. 5439, 531–537.

[9] Isabelle Guyon and André Elisseeff, *An introduction to variable and feature selection*, J. Mach. Learn. Res. **3** (2003), 1157–1182.

[10] I. S. Helland, *On the structure of Partial Least Squares regression*, Communications in Statistics Simulation and Computation **17** (1988), 581–607.

[11] L. Hoegaerts, J. A. K. Suykens, and B. De Moor, *Kernel pls variants for regression*, (2003).

[12] Rafael A Irizarry, Benjamin M Bolstad, Francois Collin, Leslie M Cope, Bridget Hobbs, and Terence P Speed, *Summaries of Affymetrix GeneChip probe level data.*, Nucleic Acids Res **31** (2003), no. 4, e15.

[13] Ron Kohavi and George H. John, *Wrappers for feature subset selection*, (1998).

[14] R. Manne, *Analysis of Two Partial Least Squares Algorithms for Multivariate Calibration*, Chemometrics and Intelligent Laboratory Systems **2** (1987), 187–197.

[15] Michael R. Osborne, Brett Presnell, and Berwin A. Turlach, *On the lasso and its dual*, (1999).

[16] R Development Core Team, *R : A language and environment for statistical computing*, 2007, ISBN 3-900051-07-0.

[17] Gilbert Saporta, *Probabilitï£¡s, analyse de donnï£¡es et statistiques*, 2006.

[18] C. Saunders, A. Gammerman, and V. Vovk, *Ridge regression learning algorithm in dual variables*, (1998).

[19] J. Schäfer and K. Strimmer, *An empirical Bayes approach to inferring large-scale gene association networks*, Bioinformatics **21** (2005), 754–764.

[20] Michel Tenenhaus, *Mï£¡thodes pour dï£¡crire, expliquer et prï£¡voir*, 2 ed., 2007.

[21] Robert Tibshirani, *Regression shrinkage and selection via the lasso*, (1998).

[22] H. Zou and T. Hastie, *Regularization and variable selection via the elastic net*, Journal of the Royal Statistical Society : Series B **67** (2005), 301–320.