

PARTIAL LEAST SQUARES FOR CLASSIFICATION AND FEATURE SELECTION IN MICROARRAY GENE EXPRESSION DATA

G. FORT

SUMMARY

Advances in high-density DNA microarray technology allows monitoring of thousands of gene expression levels. One important application of gene expression microarray is classification and feature selection. When classification is based on polychotomous discrimination, the high-dimensional setting and the collinearity of the variables necessitate the development of robust regression techniques such as methods based on Partial Least Squares (PLS). The objective of this paper is to review extensions of PLS regression to generalized linear regression and to compare them when applied to classification and feature selection in Microarrays.

Keywords. Partial Least Squares, Microarray gene expression, Logistic regression, Polychotomous discrimination, Feature selection.

Affiliation. G. Fort, CNRS (LMC-IMAG), 51, rue des Mathématiques, BP 53, 38041 Grenoble Cedex 9, France, *email:* Gersende.Fort@imag.fr, *tel:* (33)476514553, *fax:* (33)476631263.

Acknowledgements. I am very grateful to Professor A. Antoniadis for useful discussions, for his insights and his many comments on earlier version of the paper.

Programs. The MATLAB codes on which the numerical results are based, are publicly available : <http://www-lmc.imag.fr/lmc-sms/Gersende.Fort/GLM/PLSforGLM.html>.

1. INTRODUCTION

The objective of the present work is to review some extensions of Partial Least Squares regression to Partial Least Squares generalized linear regression and to compare them when used in the “large p , small n ” framework. More precisely, we restrict our attention to binary and multinomial logistic regression models and consider applications to classification and feature selection in high-dimensional regression problems.

PLS is both a dimension reduction method and a regression method in linear models. Roughly speaking, it consists in sequentially constructing super-covariates *i.e.* linear combinations of the covariates, which are predictive of the response variable. Unlike the Principal Component Analysis components, the PLS super-covariates depend on the response variable (Wold (1975)). An introduction to the structure of PLS can be found in Helland (1988), a statistical view in Helland (1990), a study of the PLS geometry in Phatak and De Jong (1997) and some theoretical properties (some of them relative to the shrinkage property of PLS) in De Jong (1995); Goutis (1996); Lingjaerde and Christophersen (2000); Phatak et al. (2002). PLS has been used extensively in chemometrics for prediction and identification of latent structure models. Chemometrics data are characterized by highly collinear predictor variables and PLS revealed to be robust to deal with these data sets (Naes and Martens (1985); Frank and Friedman (1993)). Gene expression microarray data have a similar data structure : covariates are highly collinear and the number of covariates far exceed the number of observations. One important application of microarrays is classification of samples into categories; ; a reliable and precise classification of human malignancies is essential for successful treatment. Statistical analysis of these data thus requires the development of new methodologies or modification of existing ones. A second question of interest is the identification of the genes that really contribute to the discrimination process. This naturally suggests a classification procedure based on regression; in this paper, we will consider the logistic or polychotomous discrimination method. Such a procedure requires an estimate of the regression coefficient, and inference in such models is usually solved by Maximum Likelihood (ML) and in practice, relies on the Iteratively Reweighted Least Squares (IRLS)

algorithm. Unfortunately, when the number of covariates is far larger than the number of observations, the ML estimate does not exist.

To overcome the curse of the dimension and the high collinearity, it has been proposed to substitute the ML estimate by some PLS estimate. This approach requires the extension of PLS to generalized regression. The algorithms derived in Nguyen and Rocke (2002b), Marx (1996), Bastien et al. (2004) for the binary case and in Nguyen and Rocke (2002a) for the multi-class case, incorporate PLS in the classical IRLS scheme. The algorithms proposed by Ding and Gentleman (2004) and Fort and Lambert-Lacroix (2005) incorporate both PLS and a regularization technique in the IRLS scheme.

Any inferential method in regression models is a *black box*, with input arguments the response vector and the design matrix, and with output variable, an estimate of the regression coefficients. The interest of a new inferential method is both based (a) on the technical ability to return an output variable, whatever the input arguments are, and (b) on the ability to provide an answer to the statistical problem.

The first objective of this contribution is to study the different extensions on a technical point of view. Section 3 (resp. Section 5) is devoted to the extensions of PLS to binary logistic regression (resp. multi-class logistic regression) : we give the algorithms, discuss computational aspects, and in some cases, we point out that the existence and unicity of the estimate, given the input arguments, strongly depend upon some technical parameters (such as the initial point or the maximal number of iterations in iterative schemes).

The second objective is to compare the different extensions when applied to classification of microarray data. To that goal, we compare the error rate of the logistic (resp. polychotomous) discrimination methods when the estimate of the regression coefficients raises from the extensions of PLS. This is done through Leave One Out and Resampling analyses on real data sets : Colon data (binary case, Section 4), NCI60 data (multi-class case, Section 6). The regression coefficients allow the identification of the covariables that are decisive in the prediction equation; this information can be exploited to build a feature selection procedure, in order to identify a small subset of informative genes highly correlated to the outcome. Feature selection will be the second approach for the comparison of some extensions of PLS. We will run a feature selection algorithm based

on Recursive Feature Elimination (Guyon et al. (2002)), on the Colon data set.

We start with basic ingredients : Section 2 is devoted to the description of the logistic model, the IRLS algorithm, different PLS programs used in this paper, the polychotomous discrimination method and the feature selection algorithm. It also contains a short description of the data sets.

2. BASIC INGREDIENTS

The unfamiliar reader may refer to Fahrmeir and Tutz (2001) for a general definition and presentation of GLM.

Notations. By convention, vectors are column vectors; for a vector u , u_k denotes its k -th coordinate. $\mathbb{1}_n$ is the \mathbb{R}^n -valued constant vector with coordinates 1, and for two integers $a < b$, $a : b$ is the vector with components $(a, a + 1, \dots, b - 1, b)$. For a matrix A , $A_{i,j}$ is the element (i, j) , $A_{:,j}$ is the column $\#j$, and $A_{i,:}$ is the row $\#i$. If u is a vector, $A_{u,:}$ (resp. $A_{:,u}$) is the matrix formed by picking out the rows of A (resp. the columns) indexed by u . If u_1, u_2 are two vectors, A_{u_1, u_2} is the matrix $[A_{u_1,:}]_{:, u_2}$. If u_1, \dots, u_κ are \mathbb{R}^n -valued vectors, $[u_1 \dots u_\kappa]$ is the $(n \times \kappa)$ matrix with j -th column u_j . A' denotes the transpose matrix, A^+ the Moore-Penrose pseudo-inverse matrix. For a positive-definite matrix A , \sqrt{A} is its principal square root and for a square matrix, $|A|$ is the determinant. We denote by Id_n the $(n \times n)$ identity matrix, and, for some vector u , by $\text{Diag}(u)$ the diagonal matrix with entries the elements of u . Finally, $\|\cdot\|$ is the Euclidean norm and $\langle \cdot; \cdot \rangle$ the usual scalar product.

2.1. Binary and Multinomial logistic regression. Let c be a positive integer, Y be a $\{0, 1, \dots, c\}$ -valued random variable and z be a \mathbb{R}^{p+1} -valued vector of regressors. Let $\theta \in \mathbb{R}^{c(p+1)}$ be the parameter of the model, and henceforth referred to as the vector of regression coefficients. θ can be read as the concatenation of c vectors $\theta^{(y)} \in \mathbb{R}^{p+1}$, $1 \leq y \leq c$. The distribution of Y is given by

$$\forall y \in \{0, 1, \dots, c\}, \quad \mathbb{P}(Y = y|z; \theta) = \pi_y(\theta), \quad \text{with} \quad \sum_{y=0}^c \pi_y(\theta) = 1,$$

where π_y is related to the linear predictor $\eta_y(\theta) = \mathbf{z}'\theta^{(y)}$ through the link function

$$\pi_y(\theta) = h(\eta_y(\theta)) \quad \text{and} \quad h(\eta_y) = \frac{\exp(\eta_y)}{1 + \sum_{l=1}^c \exp(\eta_l)}. \quad (1)$$

By convention, $\theta^{(0)}$ is the null \mathbb{R}^{p+1} -valued vector. Equivalently, one can define a binary-valued random vector $\tilde{\mathbf{Y}} \in \{0, 1\}^c$ by the relations

$$\begin{aligned} \forall y \in \{1, \dots, c\}, \quad & \left[\tilde{Y}_y = 1 \quad \text{and} \quad \tilde{Y}_l = 0, l \in \{1, \dots, c\} \setminus \{y\} \right], \quad \text{iff} \quad \mathbf{Y} = y, \\ & \left[\tilde{Y}_l = 0, l \in \{1, \dots, c\} \right], \quad \text{iff} \quad \mathbf{Y} = 0, \end{aligned}$$

By definition, we have $\mathbb{E}_\theta[\tilde{Y}_l] = \pi_l(\theta)$, where \mathbb{E}_θ denotes the conditional expectation (conditionally to \mathbf{z}) assuming θ to be the true value of the parameter.

Throughout the paper, the vector of regressors \mathbf{z} is of the form $[1 \ x']'$, *i.e.* it contains an intercept term and p covariates.

2.2. Inference by Maximum Likelihood. The inference approach in GLM is usually based on the maximum likelihood method.

2.2.1. Block matrices. We observe n independent realizations $(\tilde{\mathbf{Y}}^{(k)}, \mathbf{z}^{(k)})_{1 \leq k \leq n}$ of $(\tilde{\mathbf{Y}}, \mathbf{z})$, respectively collected in a response vector $\mathbf{Y} \in \{0, 1\}^{nc}$ and in a design matrix $\mathbf{Z} \in \mathbb{R}^{cn \times c(p+1)}$ defined by

$$\begin{aligned} \mathbf{Y}' &= [\mathbf{Y}_1 \cdots \mathbf{Y}_{nc}] \quad \text{where} \quad \mathbf{Y}_{\iota_k + j} = \tilde{Y}_j^{(k)}, \quad \text{and} \quad \iota_k = (k-1)c, \\ \mathbf{Z}_{\iota_k + 1 : \iota_k + c, :} &= \begin{bmatrix} \mathbf{z}^{(k)'} & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & \cdots & 0 & \mathbf{z}^{(k)'} \end{bmatrix} \in \mathbb{R}^{c \times c(p+1)}, \end{aligned} \quad (2)$$

for all $1 \leq k \leq n$, $1 \leq j \leq c$. All the covariates are collected in a data matrix $\mathbf{X} \in \mathbb{R}^{n \times p}$ such that the i -th row contains $x^{(i)}$. \mathbf{X} is assumed to be standardized : each column is centered with norm 1. Let $\Pi(\theta) \in \mathbb{R}^{nc}$ defined by $\Pi_k(\theta) = h((\mathbf{Z}\theta)_k)$ for all $k \in \{1, \dots, nc\}$, so that $\Pi(\theta) = \mathbb{E}_\theta[\mathbf{Y}]$. The log-likelihood is given by

$$l(\theta) = \mathbf{Y}'\mathbf{Z}\theta + \sum_{k=1}^n \ln \left(1 - \sum_{l=1}^c \Pi_{\iota_k + l}(\theta) \right). \quad (3)$$

2.2.2. *Case 1 : \mathbf{Z} is full column-rank.* When the ML estimate exists and is unique, the solution to the normal equation $\mathbf{Z}'(\mathbf{Y} - \Pi(\theta))$ is usually computed by a Newton-Raphson algorithm. Let \mathbf{W} be a $\mathbb{R}^{(cn) \times (cn)}$ block-diagonal matrix with k -th block $\mathbf{W}_k \in \mathbb{R}^{c \times c}$, $1 \leq k \leq n$,

$$\mathbf{W}_k = \begin{bmatrix} \Pi_{\iota_k+1} (1 - \Pi_{\iota_k+1}) & -\Pi_{\iota_k+1} \Pi_{\iota_k+2} & \cdots & -\Pi_{\iota_k+1} \Pi_{\iota_k+c} \\ -\Pi_{\iota_k+1} \Pi_{\iota_k+2} & \Pi_{\iota_k+2} (1 - \Pi_{\iota_k+2}) & \cdots & -\Pi_{\iota_k+2} \Pi_{\iota_k+c} \\ \cdots & \cdots & \cdots & \cdots \\ -\Pi_{\iota_k+c} \Pi_{\iota_k+1} & -\Pi_{\iota_k+c} \Pi_{\iota_k+2} & \cdots & \Pi_{\iota_k+c} (1 - \Pi_{\iota_k+c}) \end{bmatrix}. \quad (4)$$

Upon noting that the Hessian of the log-likelihood is $-\mathbf{Z}'\mathbf{W}(\theta)\mathbf{Z}$, we have $\hat{\theta}^{\text{ML}} = \lim_t \theta^t$ where the Newton-Raphson sequence $(\theta^t)_t$ is produced by the iterative scheme

IRLS $[\mathbf{Y}, \mathbf{Z}]$

Initialization : choose $\theta^0 \in \mathbb{R}^{c(p+1)}$,

While $\|\mathbf{Z}'(\mathbf{Y} - \Pi(\theta^t))\| \geq \text{threshold}$,

$$\psi(\theta^t) = \mathbf{Z}\theta^t + \mathbf{W}(\theta^t)^{-1} (\mathbf{Y} - \Pi(\theta^t)),$$

$$\theta^{t+1} = \theta^t + \{\mathbf{Z}'\mathbf{W}(\theta^t)\mathbf{Z}\}^{-1} \mathbf{Z}' (\mathbf{Y} - \Pi(\theta^t)) = \{\mathbf{Z}'\mathbf{W}(\theta^t)\mathbf{Z}\}^{-1} \mathbf{Z}'\mathbf{W}(\theta^t)\psi(\theta^t).$$

End.

Each Newton-Raphson iteration is thus a weighted regression of a pseudo-variable ψ onto the columns of \mathbf{Z} . This yields the so-called Iteratively Reweighted Least Squares algorithm (IRLS, Green (1984)), a procedure henceforth denoted IRLS $[\mathbf{Y}, \mathbf{Z}]$. The limit $\lim_t \theta^t$ does not depend upon the initial value; in the binary case, choosing θ^0 such that $\Pi(\theta^0) = (\mathbf{Y} + 0.5)/2 = 0.25((\mathbb{I}_{nc} - \mathbf{Y}) + 3\mathbf{Y})$ works well (Fahrmeir and Tutz (2001)); in the multi-class case, we suggest to fix θ^0 such that $\Pi(\theta^0) = (3 + c)^{-1}((\mathbb{I}_{nc} - \mathbf{Y}) + 3\mathbf{Y})$. It is proved in Albert and Anderson (1984); Santner and Duffy (1986); Lesaffre and Albert (1989) that the ML estimate does not necessarily exist; the existence depends on the configuration of the sample points in the observation space. Three different cases can be distinguished, namely the separation, the quasi-separation and the overlap case. Separation means that there exists $\theta \in \mathbb{R}^{c(p+1)}$ such that for $1 \leq k \leq n$, $1 \leq j \leq c$,

$$\left[(\mathbb{I}_n \mathbf{X}) \theta^{(j)} \right]_k > \left[(\mathbb{I}_n \mathbf{X}) \theta^{(l)} \right]_k, \quad \forall l \in \{0, \dots, c\} \setminus \{j\} \quad \text{iff} \quad \mathbf{Y}_{(k-1)c+j} = 1, \quad (5)$$

where by convention, $\theta^{(0)} = 0$. Quasi-separation means that (5) holds with large inequalities; overlap is the third situation. In the first two cases, the ML estimate does not exist since the likelihood is maximized on the boundary of $\mathbb{R}^{c(p+1)}$ *i.e.* when $\|\theta\|$ tends to $+\infty$. In the third case, the ML estimate exists and is unique. These situations are illustrated on Figure 1 in the binary case, when $p = 2$: we plot the n vector-valued covariates with coordinates $\mathbf{X}_{k,:}$ in \mathbb{R}^2 with a \times -mark (resp. a \circ -mark) for samples from the first class (resp. the second). Separation means that some hyperplane separates the observation space into two half-spaces, the positive (resp. negative) half-space containing the samples from class 1 (resp. class 0); quasi-separation means that some points are on the linear boundary; overlap means that the sample points can not be separated by a hyperplane.

Insert Figure 1 about here

2.2.3. Case 2 : \mathbf{Z} is not full column-rank. This situation always occurs when $p \gg n$. The log-likelihood depends on the parameter through the linear predictor $\mathbf{Z}\theta$ so that θ is not identifiable. Nevertheless, we can always (a) formulate the model with a full column-rank design matrix \mathbf{Z}^{red} and a parameter $\gamma \in \mathbb{R}^{\text{rank}(\mathbf{Z})}$ by standard matricial manipulations; (b) solve the estimation problem and obtain, when it exists, $\hat{\gamma}^{\text{ML}}$; (c) return to the initial statistical problem by defining $\hat{\theta}^{\text{ML}}$ as the minimal norm vector among all the vectors satisfying $\mathbf{Z}^{\text{red}}\hat{\gamma}^{\text{ML}} = \mathbf{Z}\theta$. Observe that when $\text{rank}(\mathbf{Z}) = nc$, which is most often the case when $p \gg n$, the solution to the normal equation verifies

$$(\mathbf{Z}^{\text{red}}\hat{\gamma})_{(k-1)c+j} = \ln \left(\frac{\mathbf{Y}_{(k-1)c+j}}{1 - \sum_{l=1}^c \mathbf{Y}_{(k-1)c+l}} \right), \quad \forall 1 \leq k \leq n, 1 \leq j \leq c,$$

which implies $\|\hat{\gamma}\| = +\infty$. Hence $\hat{\theta}^{\text{ML}}$ can not exist, and this naturally calls for a dimension reduction, *i.e.* for reducing the high p -dimensional predictor space to a lower κ -dimensional space.

2.3. Partial Least Squares (PLS). Partial Least Squares is a regression tool that combines regression and dimension reduction (Wold (1975); Helland (1988)). The most famous dimension reduction within regression, is certainly the method of Principal Component Analysis (PCA). In PCA, orthogonal linear combinations t of the covariates are

sequentially constructed to maximize the variance of the linear combination (Jolliffe (1986)). In PLS, the idea is to construct super-covariates t which are predictive of the response variable. Orthogonal linear combinations t of the covariates are sequentially constructed to maximize the covariance between t and the response variable (see Phatak and De Jong (1997) and references therein).

We first briefly describe the classical univariate method which is, in our opinion, devoted to the case when the design matrix is on the form $[\mathbb{I}_n \mathbf{X}]$. We then propose an extension of PLS for sparse design matrices on the form (2); when $c = 1$, this extension and the classical method coincide. Till now, we concatenated the n response variables in a vector of length nc ; nevertheless, some extensions of PLS to GLM are based on the array-concatenation of the responses, in a $n \times c$ matrix. Hence, we conclude this description by the mention of MPLS (Multivariate PLS), a PLS technique derived for array-valued response variables.

2.3.1. Univariate PLS : PLS $[Y, X, W, \kappa]$. Let $Y \in \mathbb{R}^n$ be a response variable and $X \in \mathbb{R}^{n \times p}$ be a design matrix, which is assumed to be standardized in columns (each column is centered with norm 1). Choose an integer $\kappa > 0$. PLS proceeds as follows :

Initialize :

$$f_0 = Y - q_0 \mathbb{I}_n, \quad \text{with} \quad q_0 = (\mathbb{I}_n' Y) / (\mathbb{I}_n' \mathbb{I}_n)$$

$$E_0 = X.$$

For $k = 1, \dots, \kappa$,

$$t_k = E_{k-1} \omega_k, \quad \text{with} \quad \omega_k = E_{k-1}' f_{k-1},$$

$$f_k = f_{k-1} - t_k q_k, \quad \text{with} \quad q_k = (t_k' f_{k-1}) (t_k' t_k)^{-1},$$

$$E_k = E_{k-1} - t_k p_k', \quad \text{with} \quad p_k = (E_{k-1}' t_k) (t_k' t_k)^{-1}.$$

End.

By construction, (t_1, \dots, t_κ) is an orthogonal family and the PLS components t_j are orthogonal to the constant vector \mathbb{I}_n . This yields a decomposition on the form

$$Y = (\mathbb{I}_n' \mathbb{I}_n)^{-1} (\mathbb{I}_n' Y) \mathbb{I}_n + q_1 t_1 + \dots + q_\kappa t_\kappa + f_\kappa = [\mathbb{I}_n \ X] \hat{\theta}^{\text{PLS}, \kappa} + f_\kappa,$$

where f_κ is orthogonal to the space spanned by $(\mathbb{1}_n, t_1, \dots, t_\kappa)$. When $Z = [\mathbb{1}_n \ X]$ is of full column-rank, $\hat{\theta}^{\text{PLS}, \kappa}$ is uniquely defined and $\hat{\theta}_{2:p+1}^{\text{PLS}, \kappa}$ is given by (see Helland (1988))

$$\hat{\theta}_{2:p+1}^{\text{PLS}, \kappa} = \Omega (P' \Omega)^{-1} Q \quad \text{with} \quad \Omega = [\omega_1 \ \dots \ \omega_\kappa], P = [p_1 \ \dots \ p_\kappa], Q = [q_1 \ \dots \ q_\kappa]'; \quad (6)$$

otherwise, application of the above algorithm with a non full column-rank matrix Z yields an estimate $\hat{\theta}^{\text{PLS}, \kappa}$ which is the minimal norm vector among all the θ such that $Y - f_\kappa = Z\theta$. In addition, $\kappa \mapsto \|\hat{\theta}^{\text{PLS}, \kappa}\|$ is non-decreasing. These assertions are proved in Appendix A (the second one results from De Jong (1995)).

There exists a maximal number of PLS components, κ_{\max} , which is lower or equal to $\text{rank}(X)$ and depends upon Y ; more precisely, κ_{\max} is equal to the number of distinct positive eigenvalues of XX' such that for some corresponding eigenvector ν_j , $\nu_j'Y \neq 0$ (Helland (1990)). When $\kappa = \kappa_{\max}$, $Y - f_{\kappa_{\max}}$ is the projection of Y on the space spanned by the columns of Z , and PLS regression is nothing more than Least Squares regression.

In the present description, projections and orthogonalities are derived and intended with respect to the Euclidean scalar product. The algorithm can be modified to take into account an eventual heteroscedasticity of the response variables, by substituting the Euclidean scalar product by a W -scalar product where W is a positive-definite matrix (Fort and Lambert-Lacroix (2005)). Henceforth, we will refer to this procedure as PLS $[Y, X, W, \kappa]$. The next two properties, used in the sequel, are trivial to verify (and the proof is omitted for brevity)

- (i) the estimate $\hat{\theta}_{(1)}^{\text{PLS}, \kappa}$ and the scores $(t_{(1),j})_j$ returned by PLS $[Y, X, W, \kappa]$ are related to those returned by PLS $[\sqrt{W}Y, \sqrt{W}X, \text{Id}_n, \kappa]$ (denoted with the subscript (2)) by

$$\hat{\theta}_{(1)}^{\text{PLS}, \kappa} = \hat{\theta}_{(2)}^{\text{PLS}, \kappa}, \quad \sqrt{W}t_{(1),j} = t_{(2),j}.$$

- (ii) for any $\alpha, \beta > 0$, the estimate returned by PLS $[Y + \alpha \mathbb{1}_n, X, \beta \text{Id}_n, \kappa]$ is equal to the estimate returned by PLS $[Y + \alpha \mathbb{1}_n, X, \text{Id}_n, \kappa]$.

2.3.2. An extension of univariate PLS : $\text{PLS}^* [Y, \mathbf{Z}, W, \kappa]$. Let $Y \in \mathbb{R}^{nc}$ be a response variable and $\mathbf{Z} \in \mathbb{R}^{(nc) \times (c(p+1))}$ be a design matrix on the form (2). When $c > 1$, \mathbf{Z} contains c columns with null entries except n coefficients equal to 1, namely the columns

$\mathbf{Z}_{\cdot,1}, \mathbf{Z}_{\cdot,p+2} \cdots, \mathbf{Z}_{\cdot,1+(c-1)(p+1)}$. We collect these columns in the $(nc \times c)$ matrix Ξ .

Despite the special structure of \mathbf{Z} , one can decide to apply the classical PLS algorithm. Nevertheless, we want the columns of Ξ to play the same role as the vector $\mathbf{1}_n$ in the classical algorithm; that is, we want (a) project Y onto Ξ , (b) consider the residual design matrix obtained by projecting the columns of \mathbf{Z} on the orthogonal of the space spanned by Ξ ; (c) define the PLS components in the space spanned by the residual design matrix. More precisely, our extension proceeds as follows :

Regress Y onto the columns of Ξ :

$$q_0 = (\Xi' \Xi)^{-1} \Xi' Y.$$

Deflate Y and \mathbf{Z} :

$$f_0 = Y - \Xi q_0,$$

$$\tilde{\mathbf{Z}} = \mathbf{Z} - \Xi (\Xi' \Xi)^{-1} \Xi' \mathbf{Z}.$$

Extract and standardize the new design matrix :

let $\bar{\mathbf{Z}}$ be the $nc \times np$ matrix formed with the non-null columns of $\tilde{\mathbf{Z}}$.

standardize the (centered) columns of $\bar{\mathbf{Z}}$ to have norm 1; let E_0 be the standardized matrix.

For $k = 1, \dots, \kappa$,

$$t_k = E_{k-1}' E_{k-1}' f_{k-1},$$

$$f_k = f_{k-1} - t_k (t_k' f_{k-1}) (t_k' t_k)^{-1},$$

$$E_k = E_{k-1} - t_k (t_k' E_{k-1}) (t_k' t_k)^{-1}.$$

End.

This yields a decomposition of the form

$$Y = \Xi q_0 + q_1 t_1 + \cdots + q_\kappa t_\kappa + f_\kappa = \mathbf{Z} \hat{\theta}^{\text{PLS}^*, \kappa} + f_\kappa.$$

Here again, $\hat{\theta}^{\text{PLS}^*, \kappa}$ is uniquely defined if \mathbf{Z} is full column-rank; otherwise, $\hat{\theta}^{\text{PLS}^*, \kappa}$ is the shortest norm vector among the admissible ones. The Euclidean geometry can be replaced by a weighted one, induced by a positive definite matrix $W \in \mathbb{R}^{nc \times nc}$. This procedure is henceforth denoted $\text{PLS}^* [Y, \mathbf{Z}, W, \kappa]$.

2.3.3. *Multivariate PLS : MPLS* $[Y^a, X, \kappa]$. Let $Y^a \in \mathbb{R}^{n \times c}$ be an array-valued response variable and X be a $\mathbb{R}^{n \times p}$ data matrix. MPLS amounts to finding two sets of weights ω, c in order to create a linear combination $t = X\omega$ of the columns of X (resp. a linear combination $u = Y^a c$ of the columns of Y^a) such that the square of their covariance is maximal under the constraints, $c'c = 1, \omega'\omega = 1$. X and Y^a are then deflated with respect to t ; the process is repeated with the deflated matrices. This yields the following algorithm : let κ be a positive integer

Let f_0 and E_0 be formed by respectively standardizing the matrices Y^a and X (the columns of f_0 and E_0 are centered with norm 1).

For $k = 1, \dots, \kappa$,

let ω_k be an eigenvector of $E'_{k-1} f_{k-1} f'_{k-1} E_{k-1}$, corresponding to the largest eigenvalue;

$$t_k = E_{k-1} \omega_k;$$

$$E_k = E_{k-1} - t_k (t'_k E_{k-1}) (t'_k t_k)^{-1};$$

$$f_k = f_{k-1} - t_k (t'_k f_{k-1}) (t'_k t_k)^{-1};$$

End.

This yields a decomposition of the form

$$Y = \mathbb{I}_n q'_0 + t_1 q'_1 + \dots + t_\kappa q'_\kappa + f_\kappa = [\mathbb{I}_n \ X] \hat{\Theta}^{\text{MPLS}, \kappa} + f_\kappa$$

where $q_j \in \mathbb{R}^c$ and $\hat{\Theta}^{\text{MPLS}, \kappa} \in \mathbb{R}^{(p+1) \times c}$. Column $\#j$ of $\hat{\Theta}^{\text{MPLS}, \kappa}$ is the MPLS estimate of $\theta^{(j)}$.

The reader may refer to Hoskuldsson (1988); Garthwaite (1994) for an interpretation and practical implementations of this algorithm.

2.4. Polychotomous Discrimination. Given an estimate of the regression coefficients $\hat{\theta}$, the class of a new sample characterized by a vector of covariates $x \in \mathbb{R}^p$ is predicted by

$$\hat{Y} = \operatorname{argmax}_{y \in \{0, \dots, c\}} \mathbb{P} \left(Y = y | Z = [1 \ x']; \hat{\theta} \right);$$

a rule which is, by (1), equivalent to

$$\hat{Y} = y \quad \text{iff} \quad \left[z' \hat{\theta}^{(y)} \geq z' \hat{\theta}^{(l)}, \quad \forall l \in \{0, \dots, c\} \right],$$

where, by convention, $\hat{\theta}^{(0)} = 0$. In the binary case, this method is called Logistic Discrimination. Usually, $\hat{\theta}$ is the ML estimate. Since, in the present framework $n \ll p$, the ML estimate is unlikely to exist, we substitute this estimate by one raising from extensions of PLS to GLM, detailed in Section 3 for the case $c = 1$, and in Section 5 for the case $c > 1$.

To assess the prediction, we will consider a M -fold cross-validation and/or a Resampling analysis. In a M -fold cross-validation, the data set (of size say n) is divided into M non overlapping groups of roughly same size; the model is fitted, using the samples of $M - 1$ groups combined together and is tested on the remaining one. This is repeated M times. The case $M = n$ is the so-called Leave One Out analysis. In a resampling analysis, we run $N = 100$ out of sample analyses (*i.e.* the regression model is constructed using the learning samples and outcomes of the test samples are predicted) on N random subdivisions of the data set into a learning set and a test set following a 2:1 scheme; the proportion of samples from each class in the learning set is the proportion of each class in the data set. For a given data set, the same N subdivisions are used to compare the different algorithms. Furthermore, some methods depend upon an hyperparameter (e.g. the number of PLS components κ); it is determined by Leave One Out cross validation (LOOCV) error rate for the learning set.

2.5. Feature Selection by iterative thresholding. Guyon et al. (2002) propose a feature selection algorithm in the case of binary output, based on Support Vector Machine (SVM) with Recursive Feature Elimination (RFE). Their backward selection procedure starts with all the available genes; the SVM is trained and genes having the highest vector weights are selected to form the next model. The process is repeated till removing all the genes. The number of discarded genes between two successive models is chosen by the user. This algorithm yields a family of nested models, and Guyon et al. provide several metrics of quality in order to compare them. Classically, accuracy of a model is measured by cross-validation : a proper way to evaluate the performance of a model is to divide the data set into a learning set and a test set, learn the gene selection rule on the training samples and measure the performance on the left out test samples. The test samples have to be external to the iterative gene selection process,

otherwise one introduces a selection bias when evaluating the performances (Ambroise and McLachlan (2002)).

Based on these considerations, we propose the following feature selection algorithm. Let the data set be divided into a learning set and a test set; fit the full model using the learning samples and measure its performance using the test samples. Discard the 2 genes with the lowest regression coefficient (in absolute value) and fit this new model, using again the learning set. Observe that since, in our convention, the design matrix is standardized per columns, this ranking criterion corresponds to the criterion adopted in Zhu and Hastie (2004). Repeat this process till the obtention of a model of minimal size. To test the prediction accuracy of a model, we use 10-fold cross-validation. Observe that, since there is no guarantee that the same subset of genes will be extracted at each level of the cross-validation, we test a rule characterized by a number of features and not a rule characterized by a given feature subset.

2.6. Real data sets. We will use the Colon data and the NCI60 data, publicly available at

Colon : <http://microarray.princeton.edu/oncology/affydata/index.html>

NCI60: <http://discover.nci.nih.gov/datasetsNature2000.jsp>.

and largely described resp. in Alon et al. (1999) and Scherf et al. (2000). The Colon data set contains 62 tissue samples (40 'tumor tissues' and 22 'normal tissues') with 2000 genes. The Colon data are pre-processed as described in Fort and Lambert-Lacroix (2005). This step discards some genes based on informations from the learning samples. Hence, the list of the discarded genes varies when the learning set varies, and the number of available covariates depends on the subdivision learning set / test set of the data set. The NCI60 data set contains 35 tumor samples from 5 cancer types (6 central nervous system, 8 renal, 8 melanoma, 7 colon and 6 leukemia), with 1415 genes. Missing values exist for NCI60 data : we drop out genes some genes and impute missing values as described in Ding and Gentleman (2004) so that there remain 1299 genes. Both the data sets are standardized : for each gene, the vector of expression levels from the learning samples is centered with norm 1. The same linear transformation is applied

on the vector of expression levels from the test samples (the vector is not necessarily centered with norm 1).

3. EXTENSIONS OF PLS TO GLM, IN THE BINARY CASE

In this section, $\mathbf{Z} \in \mathbb{R}^{n \times (p+1)}$ is equal to $[\mathbb{I}_n \ \mathbf{X}]$. The k -th coordinate of the vector $\Pi(\theta)$ is $(1 + \exp(-(\mathbf{Z}\theta)_k))^{-1}$ and $\mathbf{W}(\theta)$ is a diagonal matrix with k -th entry $(\Pi_k(1 - \Pi_k))(\theta)$.

3.1. Nguyen and Rocke : NR $[\mathbf{Y}, \mathbf{Z}, \kappa]$.

3.1.1. *The Nguyen & Rocke's algorithm.* The method proposed by Nguyen and Rocke (2002b) proceeds into two steps; let κ be a positive integer.

Run PLS $[\mathbf{Y}, \mathbf{X}, \text{Id}_n, \kappa]$ and return the first κ PLS components t_1, \dots, t_κ . Set $\mathbf{T}_\kappa = [\mathbb{I}_n \ t_1 \ \dots \ t_\kappa] \in \mathbb{R}^{n \times (\kappa+1)}$.

Run IRLS $[\mathbf{Y}, \mathbf{T}_\kappa]$ and return $\hat{\theta}$, the limiting value of the Newton-Raphson sequence (a regression coefficient in terms of the PLS components $(t_j)_{j \leq \kappa}$).

Express the regression in terms of the original explanatory variables and return $\hat{\theta}^{\text{NR}, \kappa}$.

Roughly speaking, a dimension reduction is first performed in order to replace the initial design matrix \mathbf{Z} by a new full column-rank design matrix \mathbf{T}_κ that collects the κ PLS covariates most informative on the output variable \mathbf{Y} . Then, a classical logistic regression is performed onto the columns of the new design matrix.

3.1.2. *Computational aspects.* Consider the singular value decomposition of $\mathbf{X} = UDV'$ where U and V are orthogonal $(n \times n)$ and $(p \times p)$ matrices and D is a $(n \times p)$ matrix with null entries except $r = \text{rank}(\mathbf{X})$ entries on the first diagonal. Replacing \mathbf{Z} for $\mathbf{Z}^{\text{red}} = [\mathbb{I}_n \ U_{:,1:r} D_{1:r,1:r}]$ in the above algorithm yields a unique estimate $\hat{\gamma}^{\text{NR}, \kappa} \in \mathbb{R}^{1+r}$, when it exists. $\hat{\theta}^{\text{NR}, \kappa}$ is the vector of minimal norm among all the vectors satisfying

$\mathbf{Z}^{\text{red}}\hat{\gamma}^{\text{NR},\kappa} = \mathbf{Z}\theta$, and is related to $\hat{\gamma}^{\text{NR},\kappa}$ through the relations

$$\hat{\theta}_1^{\text{NR},\kappa} = \hat{\gamma}_1^{\text{NR},\kappa} \quad \text{and} \quad \hat{\theta}_{2:p+1}^{\text{NR},\kappa} = V_{:,1:r}\hat{\gamma}_{2:r+1}^{\text{NR},\kappa}. \quad (7)$$

3.1.3. Existence of the estimate $\hat{\theta}^{\text{NR},\kappa}$. Whatever (\mathbf{Y}, \mathbf{Z}) , the matrix \mathbf{T}_κ is uniquely defined whenever $\kappa \leq \kappa_{\max}$. If $\kappa > \kappa_{\max}$, the PLS components $(t_j)_{j>\kappa_{\max}}$ are null vectors (up to numerical approximations).

In some cases, there exists κ_* such that IRLS $[\mathbf{Y}, \mathbf{T}_{\kappa_*}]$ can not converge : the n sample points in the observation space \mathbb{R}^κ are (quasi)-separated and the ML estimate does not exist. Observe that since the columns of the design matrix \mathbf{T}_j are pairwise orthogonal, if IRLS $[\mathbf{Y}, \mathbf{T}_{\kappa_*}]$ does not converge, then IRLS $[\mathbf{Y}, \mathbf{T}_j]$ can not converge, for any $\kappa_* \leq j \leq \kappa_{\max}$. In case of non-convergence, we decide to stop the IRLS step when separation is detected; the estimate $\hat{\theta}^{\text{NR},\kappa_*}$ is set to the current value of the Newton-Raphson sequence θ^t , a value of the parameter that correctly separates the learning samples in two classes. Applying such a rule yields an estimate that depends upon the initial value of the Newton-Raphson sequence.

3.2. Marx : IRPLS $[\mathbf{Y}, \mathbf{Z}, \nu, \kappa]$.

3.2.1. The Iteratively Reweighted PLS algorithm. The extension proposed by Marx (1996) proceeds also into two steps; for some positive integers (ν, κ) , $\nu \leq \kappa$,

Initialization :

Choose θ^0 .

Step A : While non-convergence,

Set $\psi^t = \mathbf{Z}\theta^t + \mathbf{W}(\theta^t)^{-1}(\mathbf{Y} - \Pi(\theta^t))$.

Run PLS $[\psi^t, \mathbf{X}, \mathbf{W}(\theta^t), \kappa]$ and set $\theta^{t+1} = \hat{\theta}^{\text{PLS},\kappa}$, and $\mathbf{T}_\nu = [\mathbb{I}_n \ t_1 \ \cdots \ t_\nu]$.

End.

Step B : Run IRLS $[\mathbf{Y}, \mathbf{T}_\nu]$ and return $\hat{\theta}^\nu$, the limiting value of the Newton-Raphson sequence (a regression coefficient in terms of the PLS components).

Express the regression in terms of the original explanatory variables, and return $\hat{\theta}^{\text{M},\kappa,\nu}$.

Step A is nothing else than IRLS, in which each weighted Least-Squares regression is replaced with a weighted PLS regression with a fixed number of components κ . At convergence, the first ν PLS covariates $(t_j)_{j \leq \nu}$ are collected in \mathbf{T}_ν . This new design matrix \mathbf{T}_ν is then plugged in a ML inferential scheme (Step B).

The author also discusses the choice of (κ, ν) , and initialize θ^0 as suggested in Section 2.2.

3.2.2. Computational aspects. Here again, we can substitute the original design matrix \mathbf{Z} for the matrix $\mathbf{Z}^{\text{red}} \in \mathbb{R}^{n \times (1 + \text{rank}(\mathbf{X}))}$, defined in Section 3.1.2. This yields an estimate $\hat{\gamma}^{\text{M}, \kappa, \nu}$ of the vector of regression with respect to the columns of \mathbf{Z}^{red} ; the vector of regression in terms of the original explanatory variables $\hat{\theta}^{\text{M}, \kappa, \nu}$ is then obtained as in (7).

3.2.3. Existence of the estimate $\hat{\theta}^{\text{M}, \kappa, \nu}$. Due to the PLS algorithms, κ has to be chosen lower or equal to some upper bound κ_{\max} that, in theory, depends on $(\mathbf{Z}, (\psi^t)_t, (\mathbf{W}^t)_t)$. In practice, on the considered data sets, κ_{\max} is constant and equal to $n - 1$.

When $\kappa = \kappa_{\max}$ and \mathbf{Z} is full rank, step A never converges; indeed, by definition of PLS, $\mathbf{Z}\theta^{t+1}$ is the $\mathbf{W}(\theta^t)$ -projection of $\psi^t \in \mathbb{R}^n$ onto the n -dimensional space spanned by the columns of \mathbf{Z} . This implies that for all $t \geq 0$, $\mathbf{Z}\theta^{t+1} = \psi^t$ and, component-wise,

$$(\mathbf{Z}\theta^{t+1})_k = \phi((\mathbf{Z}\theta^t)_k) \quad \text{where} \quad \phi(u) = \begin{cases} 1 + u + \exp(u), & \text{for all } k, \text{ such that } \mathbf{Y}_k = 1, \\ -1 + u - \exp(u) & \text{for all } k, \text{ such that } \mathbf{Y}_k = 0. \end{cases}$$

Step A never stops since ϕ does not have a fixed point. This non-convergence may occur when $\kappa < \kappa_{\max}$ too.

In addition, IRLS is not guaranteed to converge, but here again, we can substitute the stopping rule based on the convergence of $(\theta^t)_t$ by a stopping rule based on the detection of the separation.

3.3. Ding and Gentleman : IRPLSF $[\mathbf{Y}, \mathbf{Z}, \kappa]$.

3.3.1. The Iteratively Reweighted PLS-Firth algorithm. Bull et al. (2001) propose an algorithm close the ML inferential approach, to make robust the ML estimate in cases of small samples, when \mathbf{Z} is a full column-rank matrix. They prone the use of the

Firth-penalized ML estimate which is defined as the unique maximum of the penalized log-likelihood function $l_F^*(\theta) = l(\theta) - 0.5 \ln |\mathbf{Z}'\mathbf{W}(\theta)\mathbf{Z}|$ where l is given by (3), and the regularization term $-0.5 \ln |\mathbf{Z}'\mathbf{W}(\theta)\mathbf{Z}|$ is minimal at $\theta = 0$. The maximum is computed by a Newton-Raphson algorithm, and each loop of this iterative algorithm can be understood as a weighted least squares regression of some so-called pseudo-variable ψ^t onto the columns of \mathbf{Z} .

Ding and Gentleman (2004) extends this regularization technique to the high-dimensional regression framework $n \ll p$ by substituting the weighted least squares regression by a weighted PLS one. This yields the following algorithm. Let κ be a positive integer.

Initialization :

Choose θ^0 .

While non convergence,

Set $H^t = \sqrt{\mathbf{W}(\theta^t)}\mathbf{Z}(\mathbf{Z}'\mathbf{W}(\theta^t)\mathbf{Z})^+\mathbf{Z}'\sqrt{\mathbf{W}(\theta^t)}$, and let h^t be the diagonal matrix with diagonal entries $(H_{kk}^t)_{1 \leq k \leq n}$.

Set $\tilde{\mathbf{W}}(\theta^t) = (\text{Id}_n + h^t)\mathbf{W}(\theta^t)$.

Set $\psi^t = \mathbf{Z}\theta^t + [\tilde{\mathbf{W}}(\theta^t)]^{-1}[(\text{Id}_n + 0.5h^t)\mathbf{Y} - (\text{Id}_n + h^t)\Pi(\theta^t)]$.

Run PLS $[\psi^t, \mathbf{X}, \tilde{\mathbf{W}}(\theta^t), \kappa]$ and set $\theta^{t+1} = \hat{\theta}^{\text{PLS}, \kappa}$.

End.

Return $\hat{\theta}^{\text{DG}, \kappa} = \lim_t \theta^t$.

The authors also provide programs in R, available at <http://www.bioconductor.org/>, in which they initialize their algorithm by setting $\mathbf{Z}\theta^0 = \psi^0 = 0.75\mathbf{Y} + 0.25(\mathbb{I}_n - \mathbf{Y})$.

3.3.2. Computational aspects. Here again, we can substitute the original design matrix \mathbf{Z} for the matrix $\mathbf{Z}^{\text{red}} \in \mathbb{R}^{n \times (1 + \text{rank}(\mathbf{X}))}$, defined in Section 3.1.2. This yields an estimate $\hat{\gamma}^{\text{DG}, \kappa}$ of the regression coefficients with respect to the columns of \mathbf{Z}^{red} ; the vector of regression in terms of the original explanatory variables is then obtained as in (7).

3.3.3. Existence of the estimate $\hat{\theta}^{\text{DG}, \kappa}$. Due to the PLS algorithms, there exists an upper bound for the value κ , denoted κ_{max} , which depends upon $(\mathbf{Z}, (\psi^t)_t, (\mathbf{W}(\theta^t))_t)$. In practice, κ_{max} is constant over the iterations.

When $\kappa = \kappa_{\max}$, the above algorithm maximizes the function $\theta \mapsto l(\theta) - 0.5 \ln |\mathbf{Z}'\mathbf{W}(\theta)\mathbf{Z}|^+$, where for some positive semi-definite matrix A , $|A|^+$ stands for the product of the positive eigenvalues. Upon noting that $\partial_{\theta_k} \ln |AA'| = \text{Trace}((A'A)^{-1} A' \partial_{\theta_k} A)$ for any matrix A such that $A'A$ invertible (Bates (1983)), the gradient is given by $Z'((\text{Id}_n + 0.5h)\mathbf{Y} - (\text{Id}_n + h)\Pi(\theta))$ where h is a diagonal matrix with diagonal equal to that of the hat matrix $H = \sqrt{\mathbf{W}}\mathbf{Z}(\mathbf{Z}'\mathbf{W}\mathbf{Z})^+\mathbf{Z}'\sqrt{\mathbf{W}}$. When \mathbf{Z} is full rank, which is in practice the case when $n \ll p$, h is the identity matrix, and the normal equations possess an explicit solution : $\hat{\theta}^{\text{DG}, \kappa_{\max}} = \mathbf{Z}^+ (\ln 3 \mathbf{Y} - \ln 3 (\mathbb{1}_n - \mathbf{Y}))$.

When $\kappa < \kappa_{\max}$, the algorithm is a kind of truncated Newton-Raphson algorithm : starting from θ^t , a Newton-Raphson iteration is performed and the new value of the parameter is projected onto a subspace of dimension κ spanned by the PLS components, a subspace which may be different at each iteration. This algorithm may not converge; on some examples, we have sometimes observed a cyclic behavior, *i.e.* the existence of, say, two points $\theta^{\infty,1}, \theta^{\infty,2}$ such that for all sufficiently large t , $\theta^{2t} = \theta^{\infty,1}$ and $\theta^{2t+1} = \theta^{\infty,2}$.

3.4. Fort and Lambert-Lacroix : RPLS $[\mathbf{Y}, \mathbf{Z}, \lambda, \kappa]$.

3.4.1. *The Ridge-PLS algorithm.* The algorithm proposed in Fort and Lambert-Lacroix (2005) divides into two steps. Let \mathbf{R}_s be a diagonal $s \times s$ matrix with diagonal entries $[0 \ 1 \ \dots \ 1]$, and λ, κ be resp. a positive real number and a positive integer.

Initialization :

Choose θ^0 .

Step A : While non convergence,

set $\psi^t = \mathbf{Z}\theta^t + \mathbf{W}(\theta^t)^{-1} (\mathbf{Y} - \Pi(\theta^t))$,

set $\theta^{t+1} = (\mathbf{Z}'\mathbf{W}(\theta^t)\mathbf{Z} + \lambda\mathbf{R}_{p+1})^{-1} \mathbf{Z}'\mathbf{W}(\theta^t)\psi^t$.

End.

Step B : Run PLS $[\psi^\infty, \mathbf{X}, \mathbf{W}(\theta^\infty), \kappa]$ and return $\hat{\theta}^{\text{FL}, \lambda, \kappa} = \hat{\theta}^{\text{PLS}, \kappa}$.

Step A is a Newton-Raphson algorithm to optimize the ridge-penalized ML criterion $l_R^*(\theta) = l(\theta) - 0.5\lambda\|\mathbf{R}_{p+1} \theta\|^2$. The pseudo-variable at convergence of this iterative procedure, ψ^∞ has a linear structure on the form $\mathbf{Z}\theta^\infty + \epsilon$, where conditionally on θ^∞

being the true value of the parameter, ϵ is a zero-mean noise with dispersion matrix $\mathbf{W}(\theta^\infty)^{-1}$. PLS is then called with input response variables ψ^∞ and a weight matrix $\mathbf{W}(\theta^\infty)$ which takes into account the heteroscedasticity of the noise ϵ .

The authors also provide programs in MATLAB and R (resp. available at <http://www-lmc.imag.fr/lmc-sms/Gersende.Fort,Sophie.Lambert>), in which the algorithm is initialized as in Section 2.2, by setting $\Pi(\theta_0) = 0.75\mathbf{Y} + 0.25(\mathbb{I}_n - \mathbf{Y})$.

3.4.2. Computational aspects. Set $\tilde{\theta}_1 = \theta_1$ and $\tilde{\theta}_{2:p+1} = V'\theta_{2:p+1}$, where V is defined in Section 3.1.2. We have $l_R^*(\theta) = l_R^{*,\text{red}}(\tilde{\theta}_{1:r+1}) - 0.5 \sum_{k=r+2}^{p+1} (\tilde{\theta}_k)^2$ where for $\gamma \in \mathbb{R}^{r+1}$,

$$l_R^{*,\text{red}}(\gamma) = \mathbf{Y}'\mathbf{Z}^{\text{red}}\gamma + \sum_{k=1}^n \ln(1 - \Pi_k^{\text{red}}(\gamma)) - 0.5\lambda\|\mathbf{R}_{r+1}\gamma\|^2,$$

\mathbf{Z}^{red} and r are defined in Section 3.1.2 and $\Pi_k^{\text{red}}(\gamma) = (1 + \exp(-(\mathbf{Z}^{\text{red}}\gamma)_k))^{-1}$. This implies that θ maximizes l_R^* if and only if $\tilde{\theta}_{1:r+1}$ maximizes $l_R^{*,\text{red}}$ and $\tilde{\theta}_k = 0$, $r+2 \leq k \leq p+1$. As a consequence, in Step A, we can replace \mathbf{Z} for \mathbf{Z}^{red} . At convergence, this yields $\gamma^\infty \in \mathbb{R}^{r+1}$, a vector of regression with respect to the columns of \mathbf{Z}^{red} ; the vector of regression in terms of the original explanatory variables is the shortest norm vector among all the θ satisfying $\mathbf{Z}^{\text{red}}\gamma^\infty = \mathbf{Z}\theta$ and is thus obtained as in (7). Since the same substitution can be done in the PLS step, all the steps of the above algorithm can be run with \mathbf{Z}^{red} instead of \mathbf{Z} ; we obtain $\hat{\gamma}^{\text{FL},\lambda,\kappa}$ and deduce $\hat{\theta}^{\text{FL},\lambda,\kappa}$ as in (7).

3.4.3. Existence of $\hat{\theta}^{\text{FL},\lambda,\kappa}$. The function $l^{*,\text{red}}$ is strictly concave and tends to $-\infty$ when $\|\gamma\| \rightarrow +\infty$ (coercivity); the maximum exists and is unique. This means that, in Step A, any converging sequence has the same limit whatever θ^0 ; since the PLS estimate is uniquely defined given the entries $(\psi^\infty, \mathbf{X}, \mathbf{W}, \kappa)$, $\hat{\gamma}^{\text{FL},\lambda,\kappa}$ exists and is unique. And so $\hat{\theta}^{\text{FL},\lambda,\kappa}$ is.

When $\lambda = 0$, $\lim_t \theta^t$ is the ML estimate, and as discussed in Section 2.2, it never exists if $\text{rank}(\mathbf{Z}) = n$ which is in practice the case when $n \ll p$. Step A never converges thus explaining the condition $\lambda > 0$. When $\lambda \rightarrow +\infty$, $\lim_t \theta^t$ tends to $[\ln(\bar{y}/(1-\bar{y})), 0, \dots, 0]'$ where $\bar{y} = n^{-1} \sum_{k=1}^n \mathbf{Y}_k$; i.e. $\lim_t \Pi(\theta^t)$ is the ML estimate of the probability of success when the observations are independent and identically distributed Bernoulli variables. The weight matrix tends to $\omega = \bar{y}(1-\bar{y})\text{Id}_n$ and $\hat{\theta}^{\text{FL},+\infty,\kappa}$ is related to $\hat{\theta}^{\text{PLS},\kappa}$, the

estimate returned by PLS $[\mathbf{Y}, \mathbf{X}, \text{Id}_n, \kappa]$ by

$$\hat{\theta}^{\text{FL}, +\infty, \kappa} = \frac{1}{\bar{y}(1 - \bar{y})} \hat{\theta}^{\text{PLS}, \kappa} + \left(\ln\left(\frac{\bar{y}}{1 - \bar{y}}\right) - \omega^{-1} \bar{y} \right) [1, 0, \dots, 0]'$$

This discussion evidences that λ has to be chosen sufficiently large, but not too large.

3.4.4. Choice of λ . The weakness of the method RPLS, compared to some previous ones, is that it depends on two parameters (λ, κ) , while the previous methods only depend on κ . Fort and Lambert-Lacroix (2005) propose to determine λ at the end of Step A (independently of κ), by choosing the value that minimizes the BIC criterion

$$\text{BIC}(\lambda, \theta) = -2l(\theta) + \log(n) \text{Trace} \left(\sqrt{\mathbf{W}}(\theta) \mathbf{Z} (\mathbf{Z}' \mathbf{W}(\theta) \mathbf{Z} + \lambda \mathbf{R}_{p+1})^{-1} \mathbf{Z}' \sqrt{\mathbf{W}}(\theta) \right),$$

evaluated at $\theta = \lim_t \theta^t$, a limit depending on λ . In practice, the criterion is minimized on a range chosen by the user; we observed that, in some cases, the BIC criterion is minimal when $\lambda \rightarrow +\infty$ so that λ is set to the upper limit of the range. In the following applications, the BIC criterion is evaluated on 61 \log_{10} -linearly spaced points within the range $[10^{-3}, 10^3]$. In the literature, the choice of λ by minimization of a GCV criterion is often advocated; in the present case, this criterion is equal to $\sum_{k=1}^n \{\Pi_k^{-2} \mathbb{I}_{\mathbf{Y}_k=1} + (1 - \Pi_k)^{-2} \mathbb{I}_{\mathbf{Y}_k=0}\}$, which is minimal when $\Pi_k = \mathbb{I}_{\mathbf{Y}_k=1} + (1 - \mathbb{I}_{\mathbf{Y}_k=0})$. When \mathbf{Z} is full row-rank, this occurs by choosing $\lambda = 0$ (*i.e.* when $\theta = \hat{\theta}^{\text{ML}}$ which is of infinite norm) so that the GCV criterion is not pertinent for the present framework.

3.5. Bastien, Esposito Vinzi and Tenenhaus : PLSGLR $[\mathbf{Y}, \mathbf{Z}, \kappa]$.

3.5.1. The PLS Generalized Linear Regression algorithm. Bastien et al. (2004) develop a method in the case $n > p$, for a full column-rank design matrix $\mathbf{Z} = [\mathbb{I}_n \mathbf{X}]$. The method is based on the following observation : PLS defines t_1 by the relation $\sum_{j=1}^p X_{:,j} \langle X_{:,j}, \mathbf{Y} \rangle$ where $\langle X_{:,j}, \mathbf{Y} \rangle$ is, up to the multiplicative term $\|X_{:,j}\|^2$, the ordinary Least Squares regression coefficient of \mathbf{Y} on $X_{:,j}$. The idea is to extend PLS to GLM by replacing this ordinary regression by a generalized linear regression, and to iterate the mechanism to construct $(t_j)_{1 \leq j \leq \kappa}$.

Bastien (2004) apply their algorithm to Cox model in the context of highly multidimensional data ($n \ll p$). We apply their algorithm to the design matrix \mathbf{Z} . Their method divides into two steps; let κ be a positive integer.

Initialization :

Set $\mathbf{E}^0 \in \mathbb{R}^{n \times p}$ be the centered covariate matrix ($\mathbf{E}_0 = \mathbf{X} - n^{-1} \mathbb{1}_n \mathbb{1}_n' \mathbf{X}$).

Step A : Construction of the PLS components

For $k = 0, \dots, \kappa - 1$,

For $j = 1, \dots, p$,

Run IRLS $[\mathbf{Y}, [\mathbb{1}_n \ t_1 \ \dots \ t_k \ \mathbf{E}_{:,j}^k]]$ and return $a_{k+1,j}$, the limiting value of the Newton-Raphson sequence.

Set $w_{k+1,j} = a_{k+1,j} \|\mathbf{E}_{:,j}^k\|^2$.

End.

Set $t_{k+1} = \mathbf{E}^k w_{k+1,:} \|w_{k+1,:}\|^{-1}$ and $\mathbf{E}^{k+1} = \mathbf{E}^k - t_{k+1} t_{k+1}' \mathbf{E}^k \|t_{k+1}\|^{-2}$.

End.

Step B : Run IRLS $[\mathbf{Y}, [\mathbb{1}_n \ t_1 \ \dots \ t_\kappa]]$ and return $\hat{\boldsymbol{\theta}}^\kappa$ the limiting value of the Newton-Raphson sequence, a regression coefficient in terms of the PLS components $(t_j)_{1 \leq j \leq \kappa}$.

Express the regression in terms of the original explanatory variables, and return $\hat{\boldsymbol{\theta}}^{\text{B},\kappa}$.

By convention, the matrix $[t_1 \ \dots \ t_0 \ \mathbf{E}_{:,j}^0]$ is the column matrix $\mathbf{E}_{:,j}^0$. Bastien et al. also discuss the choice of κ , and propose a simplification of the computation of the PLS components which consists in setting to zero the non-significant coefficients $a_{k,j}$.

Contrary to the four previous methods, this method is not invariant by re-parameterization; substituting \mathbf{Z} for \mathbf{Z}^{red} in the above procedure, yields $\hat{\boldsymbol{\gamma}}^{\text{B},\kappa}$ such that $\mathbf{Z} \hat{\boldsymbol{\theta}}^{\text{B},\kappa} \neq \mathbf{Z}^{\text{red}} \hat{\boldsymbol{\gamma}}^{\text{B},\kappa}$.

3.5.2. Existence of the estimate $\hat{\boldsymbol{\theta}}^{\text{B},\kappa}$. Here again, the different IRLS algorithms are not guaranteed to converge; they can be stopped when separation is detected. Observe that if for some κ , the convergence problems only occur in Step B when regressing \mathbf{Y} on $\mathbf{T}_\kappa = [\mathbb{1}_n \ t_1 \ \dots \ t_\kappa]$, then none of the IRLS procedures of Step A with $k > \kappa$ can converge. Indeed, if there exists $\boldsymbol{\theta} \in \mathbb{R}^{\kappa+1}$ such that for all $k = 1, \dots, n$, $(\mathbf{T}_\kappa \boldsymbol{\theta})_k \geq 0$

iff $\mathbf{Y}_k = 1$, then there exists $\tilde{\theta} \in \mathbb{R}^{\kappa+2}$ such that for all k , $([\mathbf{T}_\kappa \mathbf{E}_{:,j}^\kappa] \tilde{\theta})_k \geq 0$ iff $\mathbf{Y}_k = 1$. This phenomenon naturally exhibits an upper bound for the set of the admissible values of the hyperparameter κ .

4. APPLICATION : BINARY CLASSIFICATION OF MICROARRAYS

This section is devoted to the comparison of the different estimates in terms of the classification rule on the Colon data set. All the genes remaining after the pre-processing step are included in the model.

We run the different extensions of PLS for some values of κ : due to the dimension of the data sets, we think that κ has to be chosen small enough in order to perform a dimension reduction; this is the reason why we choose κ lower or equal to 6.

4.1. A Leave One Out analysis. We report in Table 1 the total number of misclassified samples over the 62 successive test sets (columns T), and the mean number of misclassified samples in the 62 learning sets of size 61 (columns L). We indicate by the sign (*) results which have to be carefully considered for some reasons detailed below. In the last row of the table, we report the number of misclassified samples when for each of the 62 analysis, we choose $\kappa \in \{1, \dots, 6\}$ that minimizes the number of misclassified test samples. In other words, the last row gives the number of samples that are systematically misclassified, whatever $\kappa \in \{1, \dots, 6\}$.

Insert Table 1 about here

The number of covariates included in the regression model depends on the subdivision learning set/ test set, due to the pre-processing procedure; in practice, it is in the range $\{1200, \dots, 1224\}$, with mean 1221.40.

The NR algorithm. For $\kappa = 1, 2, 3$, all the IRLS calls converge; for $\kappa = 4$, separation is detected for one subdivision (namely, when the test set contains sample #55 or N36); for $\kappa = 5, 6$, all the IRLS steps are stopped when separation is detected. The results given for $\kappa = 4, 5, 6$ thus depend on the initialization of IRLS and in that sense are not significative. Samples N34,36 and T33,36 are systematically misclassified, whatever

the value of κ is.

The IRPLS algorithm. For $\kappa = 1, 2$, we observe on the 62 subdivisions, a cyclic behavior in Step A and convergence of IRLS in Step B. For $\kappa = 3$, some of the paths in Step A do not converge; for $\kappa = 4, 5, 6$, none of the paths in Step A converge. As a consequence, we only report the results obtained for $\kappa = 1, 2$, when Step A is stopped after $t_{\max} = 200$ iterations, but insist on the fact that these results depend on t_{\max} . Samples $N34, 36$ and $T33, 36$ are systematically misclassified, whatever the value of κ is.

The IRPLSF algorithm. For $\kappa = 1$, we observe a periodic behavior in 28 cases : $\hat{\theta}^{\text{DG}, \kappa}$ thus depends on the maximal number of iterations t_{\max} allowed in the iterative part of the procedure. The results reported in Table 1 are obtained with $t_{\max} = 200$. For $\kappa = 2, \dots, 6$, the iterative part converges. Samples $N34, 36$ and $T33, 36$ are systematically misclassified, whatever the value of κ is.

The RPLS algorithm. The mean value of the hyper-parameter λ over the 62 analysis is 24.70. Samples $N8, 34, 36$ and $T33, 36$ are systematically misclassified, whatever the value of κ is.

The PLSGLR algorithm. The PLS super-covariates t_1, t_2, t_3 are perfectly defined, since the IRLS algorithms all converge; for t_4 , some regressions in Step A are stopped because separation is detected; for t_5, t_6 , separation systematically occurs. Step B always converges for $\kappa = 1, 2$, and for $\kappa = 3$, it is stopped due to detection of separation in one case (namely, when the test set contains sample #55 *i.e.* $N36$). For $\kappa = 4, 5, 6$, separation is systematically detected. The results given for $\kappa = 4, 5, 6$ thus depend on the initialization of IRLS and in that sense are not significative. Samples $N34, 36$ and $T33, 36, 37$ are systematically misclassified, whatever the value of κ is.

4.2. A Resampling analysis. For PLSGLR, when determining the value of the hyper-parameter κ by LOOCV training set error rate, the minimum is found over the values of κ such that separation never occurs in all the IRLS calls of Step A; and when this is never the case, the default value is 1. For IRPLSF, this minimum is over the values of κ such that Step A converges.

Figure 2[left] shows the boxplot of the test set error rate based on the 100 subdivisions, for the NR, IRPLSF, RPLS and PLSGLR algorithms; and for four other methods :

DLDA (Diagonal Linear Discriminant Analysis), DQDA (Diagonal Quadratic DA), k -nearest neighbors (k -NN) and weighted k -NN (k -WNN). For the last two algorithms, the distance is the Euclidean one, the number of neighbors k is chosen by LOOCV training set error rate in the grid $\{1, 3, \dots, 19\}$, and for k -WNN, the weight of each gene is given by the square of the t -statistic computed on the learning samples (which is equal to the ratio of the between-groups to within-groups sum of squares). This t -test score ranks genes based on their individual predictive ability. Table 2 shows the mean error rate and its standard deviation, and the mean value of κ (k for k -NN and k -WNN).

Insert Figure 2 and 2 about here

4.3. Conclusion. These analyses show that DLDA, DQDA and k -NN behave poorly : they really suffer from the dimensionality of the problem, from the multicollinearity of the design matrix, and from the noise due to the presence of irrelevant genes. They do not perform neither dimension reduction nor regularization. Comparison of k -NN and k -WNN shows that introduction of all genes with an equal importance greatly disturb the classifier; smoothing out the role of the genes with weak t -test score drops noise and improves the performances of the k -NN classifier. The boxplot shows that k -WNN, NR, IRPLSF and RPLS have an equivalent behavior. The last three methods have the great advantage of providing an estimate of the regression coefficients, a crucial knowledge for the identification of genes that really contribute to the classification process, and for feature selection. In Section 7, we compare NR, IRPLSF and RPLS when applied to the feature selection scheme presented in Section 2.5.

The Colon data set is often studied in the Microarrays literature; we point out that the above results of the Leave One Out analysis corroborate earlier observations. In Alon et al. (1999), classification is based on a deterministic-annealing algorithm and samples N8, 12, 34 and T2, 30, 33, 36, 37 are misclassified. In Furey et al. (2000), the classification is based on SVM and samples N8, 34, 36 and T30, 33, 36 are misclassified.

5. EXTENSION OF PLS TO GLM, IN THE MULTI-CLASS CASE

5.1. Nguyen and Rocke : MNR $[\mathbf{Y}, \mathbf{Z}, \kappa]$.

5.1.1. *The Multiple NR algorithm.* The method proposed by Nguyen and Rocke (2002a) proceeds into two steps; let κ be a positive integer. Denote by \mathbf{Y}^a the array-concatenation of the response variables : $\mathbf{Y}_{k,:}^a$ is $\tilde{\mathbf{Y}}^{(k)}$, $1 \leq k \leq n$.

Run MPLS $[\mathbf{Y}^a, \mathbf{X}, \kappa]$ and return the first κ PLS components t_1, \dots, t_κ . Set $\mathbf{T}_\kappa = [\mathbb{1}_n \ t_1 \ \dots \ t_\kappa] \in \mathbb{R}^{n \times (\kappa+1)}$.

Run IRLS $[\mathbf{Y}, \mathbf{T}_\kappa]$ and return $\hat{\boldsymbol{\theta}}$, the limiting value of the Newton-Raphson sequence (a regression coefficient in terms of the PLS components $(t_j)_{j \leq \kappa}$).

Express the regression in terms of the original explanatory variables and return $\hat{\boldsymbol{\theta}}^{\text{NR}, \kappa}$.

As for the binary case, a dimension reduction is first performed in order to replace the initial design matrix \mathbf{X} by a new full column-rank design matrix \mathbf{T}_κ that collects the κ PLS covariates most informative on the output variable \mathbf{Y}^a . Then, a classical logistic regression is performed onto the columns of the new design matrix.

5.1.2. *Computational aspects.* Consider the singular value decomposition of $\mathbf{X} = UDV'$ where U and V are unitary matrices and D is a $(n \times p)$ matrix with null entries except $r = \text{rank}(\mathbf{X})$ entries on the first diagonal. Replacing \mathbf{X} for $\mathbf{X}^{\text{red}} = U_{:,1:r} D_{1:r,1:r}$ in the above algorithm yields a unique estimate $\hat{\gamma}^{\text{NR}, \kappa} \in \mathbb{R}^{c(1+r)}$, when exists. $\hat{\boldsymbol{\theta}}^{\text{NR}, \kappa}$ is related to $\hat{\gamma}^{\text{NR}, \kappa}$ through the relations

$$[\hat{\boldsymbol{\theta}}^{\text{NR}, \kappa}]_1^{(j)} = [\hat{\gamma}^{\text{NR}, \kappa}]_1^{(j)} \quad \text{and} \quad [\hat{\boldsymbol{\theta}}^{\text{NR}, \kappa}]_{2:(p+1)}^{(j)} = V_{:,1:r} [\hat{\gamma}^{\text{NR}, \kappa}]_{2:(r+1)}^{(j)},$$

for all $j \in \{1, \dots, c\}$.

5.1.3. *Existence of the estimate $\hat{\boldsymbol{\theta}}^{\text{NR}, \kappa}$.* The comments for the binary case (Section 3.1.3) remain valid for the multi-class case.

5.2. Ding and Gentleman : MIRPLSF $[\mathbf{Y}, \mathbf{Z}, \kappa]$.

5.2.1. *The Multiple IRPLSF algorithm.* The multi-class algorithm follows the same lines as the two-class algorithm, and is based on a PLS within IRLS scheme, till convergence (Ding and Gentleman (2004)). We point out that the implementation of PLS

differs from the different programs given in Section 2.3 (univariate PLS, its extension PLS* and the multivariate PLS). Briefly, they use a univariate PLS in which the initialization step is omitted, *i.e.* they set $f_0 = \mathbf{Y}$ and $E_0 = \mathbf{Z}$. This means that at each PLS iteration, the PLS score is chosen in the space spanned by all the columns of \mathbf{Z} (including the binary-valued columns due to the addition of an intercept term in the model); usually, the PLS score is in the space spanned by the columns of \mathbf{Z} and orthogonal to the subspace spanned by the binary-valued columns. We refer to this implementation as PLS_{dg}.

The derivations are detailed in Ding and Gentleman (2004). Let κ be a positive integer.

Initialization :

Choose θ^0 .

While non convergence,

Set $H^t = \sqrt{\mathbf{W}(\theta^t)}\mathbf{Z}(\mathbf{Z}'\mathbf{W}(\theta^t)\mathbf{Z})^+\mathbf{Z}'\sqrt{\mathbf{W}(\theta^t)}$, and let h^t be the diagonal matrix with diagonal entries $(H_{kk}^t)_{1 \leq k \leq n}$.

Define \bar{h}^t , a diagonal $(nc \times nc)$ matrix with $((k-1)c+j)$ -th diagonal entry $\sum_{l=1}^c h_{(k-1)c+l}^t$, $1 \leq k \leq n$ and $1 \leq j \leq c$.

Set $\tilde{\mathbf{W}}(\theta^t) = \mathbf{W}(\theta^t)(\text{Id}_{nc} + 0.5(h^t + \bar{h}^t))$.

Set $\psi^t = \mathbf{Z}\theta^t + [\tilde{\mathbf{W}}(\theta^t)]^{-1}[(\text{Id}_{nc} + 0.5h^t)\mathbf{Y} - (\text{Id}_{nc} + 0.5(h^t + \bar{h}^t))\Pi(\theta^t)]$.

Run PLS_{dg} $[\psi^t, \mathbf{Z}, \tilde{\mathbf{W}}(\theta^t), \kappa]$ and set $\theta^{t+1} = \hat{\theta}^{\text{PLS}_{\text{dg}}, \kappa}$.

End.

Return $\hat{\theta}^{\text{DG}, \kappa} = \lim_t \theta^t$.

The authors also provide programs in R (available at <http://www.bioconductor.org/>), in which the algorithm is initialized by setting $\psi^0 = 0.75\mathbf{Y} + 0.25(\mathbb{I} - \mathbf{Y})$, and by drawing at random a diagonal matrix for the initial value of $\tilde{\mathbf{W}}$.

5.2.2. Computational aspects. To speed up the implementation of this method, consider the singular value decomposition of \mathbf{Z} , $\mathbf{Z} = \mathbf{U}\mathbf{D}\mathbf{V}'$ where \mathbf{U} and \mathbf{V} are unitary matrices and \mathbf{D} is a diagonal matrix of the same dimension as \mathbf{Z} . Let $r = \text{rank}(\mathbf{Z})$. We can substitute the original design matrix \mathbf{Z} for the matrix $\mathbf{Z}^{\text{red}} = \mathbf{U}_{:,1:r}\mathbf{D}_{1:r,1:r}$. This yields an estimate $\hat{\gamma}^{\text{DG}, \kappa}$ of the regression coefficients with respect to the columns of \mathbf{Z}^{red} ; the

vector of regression in terms of the original explanatory variables is then obtained by $\hat{\theta}^{\text{DG},\kappa} = V_{:,1:r} \hat{\gamma}^{\text{DG},\kappa}$.

5.2.3. Existence of the estimate $\hat{\theta}^{\text{DG},\kappa}$. Due to the PLS algorithms, there exists an upper bound for the value κ , which theoretically depends upon $(\mathbf{Z}, (\psi^t)_t, (\mathbf{W}(\theta^t))_t)$, but, in the considered applications, κ_{\max} is constant over the iterations.

When $\kappa = \kappa_{\max}$, the above algorithm maximizes the function $\theta \mapsto l(\theta) - 0.5 \ln |\mathbf{Z}'\mathbf{W}(\theta)\mathbf{Z}|^+$. By following the same lines as in Section 3.3.3, it may be shown that when \mathbf{Z} is full column rank, which is in practice the case when $n \ll p$, the maximum has an explicit expression given by

$$\hat{\theta}^{\text{DG},\kappa_{\max}} = \ln(3) \mathbf{Z}^+ \left\{ \mathbf{Y} - (\mathbb{I}_{nc} - \mathbf{Y}) .* (\mathbb{I}_{nc} - \bar{\mathbf{Y}}) \right\},$$

where $\bar{\mathbf{Y}}$ is a $\{0, 1\}$ -valued vector defined by $\bar{\mathbf{Y}}_{(k-1)c+j} = \sum_{l=1}^c \mathbf{Y}_{(k-1)c+l}$, $1 \leq k \leq n$, $1 \leq j \leq c$; and $.*$ denotes the element-by-element multiplication.

When $\kappa < \kappa_{\max}$, the algorithm is a kind of truncated Newton-Raphson algorithm which is not guaranteed to converge.

5.3. Fort and Lambert-Lacroix : MRPLS $[\mathbf{Y}, \mathbf{Z}, \lambda, \kappa]$.

5.3.1. The Multiple RPLS algorithm. The following algorithm follows that same lines as RPLS for the binary case, except that, due to the special form of the design matrix \mathbf{Z} , we use PLS* instead of the usual univariate PLS. Let $\tilde{\mathbf{R}}_{cs}$ be a diagonal $cs \times cs$ matrix with diagonal obtained by c replications of the vector $[0 \ 1 \ \cdots \ 1] \in \mathbb{R}^s$; and λ, κ be resp. a positive real number and a positive integer.

Initialization :

Choose θ^0 .

Step A : While non convergence,

$$\begin{aligned} \text{set } \psi^t &= \mathbf{Z}\theta^t + \mathbf{W}(\theta^t)^{-1} (\mathbf{Y} - \Pi(\theta^t)), \\ \text{set } \theta^{t+1} &= \left(\mathbf{Z}'\mathbf{W}(\theta^t)\mathbf{Z} + \lambda\tilde{\mathbf{R}}_{c(p+1)} \right)^{-1} \mathbf{Z}'\mathbf{W}(\theta^t)\psi^t. \end{aligned}$$

End.

Step B : Run PLS* $[\psi^\infty, \mathbf{Z}, \mathbf{W}(\theta^\infty), \kappa]$ and return $\hat{\theta}^{\text{FL},\lambda,\kappa} = \hat{\theta}^{\text{PLS*},\kappa}$.

Step A is a Newton-Raphson algorithm to optimize the ridge-penalized ML criterion $l_R^*(\theta) = l(\theta) - 0.5\lambda\|\tilde{\mathbf{R}}_{c(p+1)}\theta\|^2$. Programs in MATLAB are available (available at <http://www-lmc.imag.fr/lmc-sms/Gersende.Fort>), in which the algorithm is initialized by setting $\Pi(\theta_0) = (3 + c)^{-1}((\mathbb{I}_{nc} - \mathbf{Y}) - 3\mathbf{Y})$.

5.3.2. Computational aspects. To speed up the algorithm, one can replace the $(nc \times c(p+1))$ matrix \mathbf{Z} for a $(nc \times c(r+1))$ matrix \mathbf{Z}^{red} where $r = \text{rank}(\mathbf{X})$. To that goal, let UDV' be the singular value decomposition of \mathbf{X} (see Section 3.1.2); construct \mathbf{Z}^{red} as in (2) from the rows of $[\mathbb{I}_n \ U_{:,1:r} D_{1:r,1:r}]$ instead of the rows of $[\mathbb{I}_n \ \mathbf{X}]$. Running the algorithm with \mathbf{Z}^{red} yields $\hat{\gamma}^{\text{FL},\lambda,\kappa} \in \mathbb{R}^{c(r+1)}$, a vector of regression with respect to the columns of \mathbf{Z}^{red} . The vector of regression in terms of the original explanatory variables is the shortest norm vector among all the θ satisfying $\mathbf{Z}^{\text{red}}\hat{\gamma}^{\text{FL},\lambda,\kappa} = \mathbf{Z}\theta$ and is obtained by

$$[\hat{\theta}^{\text{FL},\lambda,\kappa}]_1^{(j)} = [\hat{\gamma}^{\text{FL},\lambda,\kappa}]_1^{(j)}, \quad [\hat{\theta}^{\text{FL},\lambda,\kappa}]_{2:(p+1)}^{(j)} = V_{:,1:r} [\hat{\gamma}^{\text{FL},\lambda,\kappa}]_{2:(r+1)}^{(j)},$$

for all $j \in \{1, \dots, c\}$.

5.3.3. Existence of $\hat{\theta}^{\text{FL},\lambda,\kappa}$. Here again, it may be proved that given (\mathbf{Y}, \mathbf{Z}) , $\hat{\theta}^{\text{FL},\lambda,\kappa}$ is unique; the proof is on the same lines as the proof in the binary case (Section 3.4.3).

When $\lambda = 0$, and $\text{rank}(\mathbf{Z}) = nc$, Step A never converges thus explaining the condition $\lambda > 0$. When $\lambda \rightarrow \infty$, $\lim_t \theta^t$ tends to a vector with c non-null entries such that

$$\forall j \in \{1, \dots, c\}, \quad [\lim_t \theta^t]_{(j-1)(p+1)+1} = \ln \left(\frac{\bar{y}_j}{1 - \sum_{l=1}^c \bar{y}_l} \right), \quad \bar{y}_j = n^{-1} \sum_{k=1}^n \mathbf{Y}_{(k-1)c+j};$$

so that $\mathbf{W}(\theta^\infty)$ tends to a block diagonal matrix with k -th block given by $\omega = \text{diag}(\bar{\mathbf{y}}) - \bar{\mathbf{y}}\bar{\mathbf{y}}'$ where $\bar{\mathbf{y}}' = [\bar{y}_1 \ \dots \ \bar{y}_c]$. Hence $[\hat{\theta}^{\text{FL},+\infty,\kappa}]^{(j)}$, the estimate of the j -th block of the parameter $\theta^{(j)}$ is given by

$$[\hat{\theta}^{\text{FL},+\infty,\kappa}]^{(j)} = [\hat{\theta}^{\text{PLS},\kappa}]^{(j)} + \left\{ \ln \left(\frac{\bar{y}_j}{1 - \sum_{l=1}^c \bar{y}_l} \right) - \{\omega^{-1}\bar{\mathbf{y}}\}_j \right\} [1, 0, \dots, 0]'$$

where $\hat{\theta}^{\text{PLS},\kappa}$ is the PLS estimate returned by PLS* $[\mathbf{W}(\theta^\infty)^{-1}\mathbf{Y}, \mathbf{Z}, \mathbf{W}(\theta^\infty), \kappa]$. In practice, one can fix λ to the value that minimizes the BIC criterion

$$\text{BIC}(\lambda, \theta) = -2l(\theta) + \log(nc) \text{Trace} \left(\sqrt{\mathbf{W}}(\theta) \mathbf{Z} (\mathbf{Z}' \mathbf{W}(\theta) \mathbf{Z} + \lambda \tilde{\mathbf{R}}_{c(p+1)})^{-1} \mathbf{Z}' \sqrt{\mathbf{W}}(\theta) \right),$$

evaluated at $\theta = \lim_t \theta^t$, a limit depending on λ . We will do so in the following applications, and will minimize the criterion on 61 \log_{10} -linearly spaced points within the range $[10^{-3}, 10^3]$.

6. APPLICATION : MULTI-CLASS CLASSIFICATION OF MICROARRAYS

We compare MNR, MIRPLSF and MRPLS when applied to polychotomous discrimination, on the NCI60 data set. We first run a leave one out analysis based on MNR, MIRPLSF, MRPLS; we report the number of misclassified test samples (column T), and the mean number over the n loops of misclassified learning samples (column L) by MNR, MRPLS and MIRPLSF, for different values of κ . The last column indicates the number of samples that are systematically misclassified, whatever the value of κ is.

We then run a resampling analysis based on MNR, MIRPLSF, MRPLS and on k -NN and k -WNN. For the nearest neighbor methods, we choose the Euclidean distance; in k -WNN, the weight of a gene is equal to its between-groups to within-groups sum of squares. For MNR, MIRPLSF and MRPLS, (resp. NN methods), the hyper-parameter κ (resp. k) is chosen by LOOCV training set error rate, within the range $\{1, \dots, 6\}$ (resp. $\{1, 3, \dots, 19\}$). For MIRPLSF, the minimum is over the values of κ such that Step A converges. We report the mean value and the standard deviation of the test set error rate, and the mean value of the hyper-parameter (κ or k). We also give a measure of accuracy of the prediction based on the *contrast*. For a vector $\pi = (\pi_0, \pi_1, \dots, \pi_c)$ of the class probability, we define the contrast by $\sum_{j=2}^{c+1} (\pi_{[1]} - \pi_{[j]}) = (c+1) \{\pi_{[1]} - (c+1)^{-1}\}$ where $\pi_{[j]}$ denotes the sorted components : $\pi_{[1]} \geq \dots \geq \pi_{[c+1]}$. A large value of the contrast means that the probability of being from the class associated to $\pi_{[1]}$ is far larger than the probability of the other classes. Since the predicted class is the class associated to $\pi_{[1]}$, the quantity *Contrast* is indicative of the classification confidence : the larger it is, the more confident the classification is. We report the mean value of $\{\hat{\pi}_{[1]} - (c+1)^{-1}\}$, when the mean is over all the estimated vectors per subdivision, and over the 100 subdivisions.

Table 3 shows the result of the leave one out analysis, when all the available genes are included in the model ($p = 1299$). Step A of MIRPLSF always converges for this data

set. Sample *Me LOXIMVI* is systematically misclassified, whatever the algorithm and the value of κ .

Insert Table 3 about here

We run a resampling analysis and include all the available genes in the model. The results are displayed on Figure 3 and Table 4.

Insert Figures 3 and Table 4 about here

On Figure 3, the boxes have a large line at the median value. (Quasi)-separation often occurs in the IRLS step of MNR; to illustrate the sensibility of the results to the initial value of IRLS, we re-run the resampling analyses, by initializing IRLS at $\theta^0 = 0$. The mean test set error rate is 0.047, with standard deviation 0.063, the mean value of κ and of the contrast are resp. equal to 3.21 and 0.639.

6.1. Conclusion. Classification in the NCI data set is a difficult task, due to the presence of many classes and very few samples from each class. In the resampling analysis, the test set error rate in k -NN is minimized for small values of k (k close to 1); this is a consequence of the definition of the learning set, which contains a very small number of samples from each class. In these unfavorable conditions, methods based on dimension reduction by PLS seem to provide better results, and among them, MIRPLSF looks more stable. The value of *Contrast* show that the classification confidence is far more important for MNR and MRPLS than it is for MIRPLSF.

7. APPLICATION : FEATURE SELECTION FOR BINARY-VALUED RESPONSE VARIABLE

We run the RFE algorithm described in Section 2.5 when the extensions are based on NR, IRPLSF and RPLS. This yields NR-RFE, IRPLSF-RFE and RPLS-RFE. Starting from the full model, we apply RFE and produce a model of size 1024 followed by the nested models of size 1022, 1020, \dots , p_{\min} . The PLS extensions are applied with different values of κ on which p_{\min} depends; $p_{\min} = 2$ when $\kappa = 1, 2$, $p_{\min} = 4$ when $\kappa = 3, 4$ and $p_{\min} = 6$ when $\kappa = 5, 6$. Each model is evaluated with three metrics

proposed by Guyon et al. (2002) : (i) the *test set success rate* **Suc**; (ii) the *acceptation rate* **Acc**, that complements the rejection rate defined as the fraction of samples that have to be discarded to obtain zero error; (iii) the *extremal margin* **Ext**, difference between the smallest linear predictor over the 1-class samples and the largest linear predictor over the 0-class samples, rescaled by the largest difference between the linear predictors. By definition, $0 < \text{Suc} < 1$, $0 < \text{Acc} < 1$ and $\text{Ext} < 1$. Figure 4 is a graphical representation of **Acc** and **Ext**. A value of any of this criterion close to 1 is indicative of the quality, in terms of a low confidence of wrong prediction (**Acc**) and a large confidence of correct prediction (**Ext**).

Insert Figure 4 about here

We sort the models based on different signed quantities : **Suc**, 0.5 Suc Ext , 0.5 Suc Acc , and \mathcal{Q} which corresponds to the signed surface of a triangle defined by the points with coordinates

$$E = (\text{Ext}, 0) \quad S = \text{Suc} (\cos(2\pi/3), \sin(2\pi/3)) \quad A = \text{Acc} (\cos(4\pi/3), \sin(4\pi/3));$$

more precisely, \mathcal{Q} is the sum of the 'surface' of the triangles SOE, EOA, AOS where by convention, the 'surface' of SOE and EOA is negative iff $\text{Ext} < 0$. Hence, $\mathcal{Q} > 0$ iff $\text{Ext} > -\text{Suc Acc} (\text{Suc} + \text{Acc})^{-1}$, as illustrated on Figure 5.

Insert Figure 5 about here

Table 5 displays the results of a 10-fold cross-validation: for the four criteria, and the different algorithms, we report the best value of the mean criterion among all the considered models (column 'value'), the size of the best model (column 'p') and the value of the hyperparameter κ for which it is reached (column ' (κ) '). The mean of the criterion is over the 10 values obtained at each step of the cross-validation. We consider RFE based on the NR estimate when NR is initialized from θ^0 given in Section 2.2; the results are on row 'NR (init1)'. When learning the NR estimate for the different nested models with $\kappa = 2$ (resp. 3, 4, 5, 6), separation occurs in 0.33% (resp. 43.91%, 87.25%, 100%, 100%) of the analyses; for $\kappa = 1$, it never occurs. To test the robustness of the NR-RFE to the initial value, we start the NR algorithms from $\theta^0 = 0$; the results are on row

'NR (init2)'. This study points out the sensibility of NR-RFE to the initial value, and more generally, the weakness of the Nguyen & Rocke's approach. We then study the performances of the IRPLSF-RFE algorithm; when learning the IRPLSF estimate for the different nested models with $\kappa = 1$ (resp. 2, 3, 4, 5, 6), the algorithm converges for 91.19% (resp. 99.82%, 90.80%, 99.90%, 99.98%) of the analyses. Here again, we test the robustness to the initial value by modifying the maximal number of iterations in the iterative part of IRPLSF. The results are similar (see row 'IRPLSF (init2)'), thus illustrating the stability of IRPLSF-RFE with respect to its non-convergence pathology. Ranking the models by the **Suc**-value select quite large models; ranking the models by the \mathcal{Q} -value yields small models. The model that maximizes \mathcal{Q} results from a compromise between the quantities **Suc**, **Acc** and **Ext**, that is, it takes into account the correct prediction, the large confidence in the correct predictions and the low confidence in the wrong predictions. When sorted by the \mathcal{Q} -value, the optimal model selected by IRPLSF-RFE has a success rate **Suc** = 0.8738, an acceptance and an extremal rates equal to **Acc** = 0.6452 and **Ext** = 0.0646. For the optimal model selected by RPLS-RFE, we have **Suc** = 0.8881, **Acc** = 0.6071 and **Ext** = 0.0163. These optimal models are resp. among the top 10% (resp. top 2.5%) when models are sorted by the **Suc**-value, thus showing that the consideration of the more general criterion \mathcal{Q} does not greatly penalize the predictive quality of the selected model (see Figures 6 and 7[right]).

Insert Table 5 and Figures 6 and 7 about here

8. CONCLUSION

We discussed the different PLS extensions to GLM on a technical point of view, and compare them when applied to classification and feature selection in Microarrays. The extensions proposed by Marx (1996) and Bastien et al. (2004) really present technical problems and do not perform well when applied to microarray data. The extensions proposed by Nguyen and Rocke (2002b,a) suffer from the separation problem : classification and feature selection greatly depend upon the initialization of some maximization procedure on which their methods rely. The extensions by Ding and Gentleman (2004) and Fort and Lambert-Lacroix (2005) seem to be the most promising extensions : the

simulations demonstrate their very interesting performances when applied to binary classification and feature selection for binary output variables; the interest of the methods is less evident when applied to multi-class classification, but this may be explained by the fact that the number of samples from some classes is very small (three, four, \dots). We observed that the classification and the feature selection methods are not sensible to the asymptotical cyclic behavior of the iterative algorithm proposed by Ding and Gentleman.

The robustness of the methods by Ding and Gentleman (2004) and Fort and Lambert-Lacroix (2005) stresses the pertinence of combining a regularization step and a dimension reduction step, when dealing with high dimensional regression problem with highly collinear regressors. The Firth penalty and the Ridge penalty are both maximal at the origin, thus attracting the estimate of the regression coefficient to the null vector. When feature selection is the question of interest, one is interested in sparse models. This naturally suggests the use of a more selective regularization step : for example, the Ridge-penalization step and a thresholding penalization could be combined in order to fight the high-collinearity of the design matrix, and to do shrinkage and automatic variable selection simultaneously. This will be the next step of our work.

APPENDIX A. PLS WITH A NON FULL COLUMN-RANK DATA MATRIX

Let \mathbf{X} be a non full column-rank standardized ($n \times p$) matrix (each column is centered with norm 1). Let $Y \in \mathbb{R}^n$ and W be a $n \times n$ symmetric positive definite matrix. Consider the singular value decomposition of \mathbf{X} , $\mathbf{X} = UDV'$, where, by convention, U and V are unitary matrices. Define

$$r = \text{rank}(\mathbf{X}), \quad \tilde{U} = U_{:,1:r}, \quad \tilde{D} = D_{1:r,1:r}, \quad \tilde{V} = V_{:,1:r},$$

so that $\mathbf{X} = UDV' = \tilde{U}\tilde{D}\tilde{V}'$, $\tilde{U}'\tilde{U} = \text{Id}_r$ and $\tilde{V}'\tilde{V} = \text{Id}_r$. Finally, denote by $E_k, f_k, t_k, \omega_k, p_k, q_k$ (resp. $\tilde{E}_k, \tilde{f}_k, \tilde{t}_k, \tilde{\omega}_k, \tilde{p}_k, \tilde{q}_k$) the quantities produced by PLS $[Y, \mathbf{X}, W, \kappa]$ (resp. by PLS $[Y, \tilde{U}\tilde{V}, W, \kappa]$).

Lemma 1. $E_0 = \tilde{E}_0\tilde{V}'$, $f_0 = \tilde{f}_0$ and for all $1 \leq k \leq \kappa$,

$$E_k = \tilde{E}_k\tilde{V}', \quad f_k = \tilde{f}_k, \quad t_k = \tilde{t}_k, \quad \omega_k = \tilde{V}\tilde{\omega}_k, \quad p_k = \tilde{V}\tilde{p}_k \quad q_k = \tilde{q}_k.$$

The proof is trivial and is omitted for brevity; it consists in replacing \mathbf{X} for $\tilde{U}\tilde{D}\tilde{V}'$ and in using the relations $\tilde{U}'\tilde{U} = \text{Id}_r$ and $\tilde{V}'\tilde{V} = \text{Id}_r$.

Proposition 2. *When \mathbf{X} is a non full column-rank matrix, centered in columns, the PLS estimate $\hat{\theta}^{\text{PLS},\kappa}$ is the shortest (Euclidean) norm vector among all the solutions satisfying $Y - f_\kappa = [\mathbb{I}_n \ \mathbf{X}] \theta$.*

Proof. By lemma 1, $\Omega (P' \Omega)^{-1} Q = \tilde{V} \tilde{\Omega} (\tilde{P}' \tilde{\Omega})^{-1} \tilde{Q}$ where $\tilde{\Omega}$, \tilde{P} and \tilde{Q} are defined as Ω, P, Q (see Eq.(6)), from the quantities $\tilde{\omega}_k, \tilde{p}_k, \tilde{q}_k$. Hence,

$$\hat{\theta}_{2:p+1}^{\text{PLS},\kappa} = \tilde{V} \tilde{\theta}_{2:r+1}^{\text{PLS},\kappa}, \quad (8)$$

where $\hat{\theta}^{\text{PLS},\kappa}$ and $\tilde{\theta}^{\text{PLS},\kappa}$ denote resp. the PLS estimates returned by PLS $[Y, \mathbf{X}, W, \kappa]$ and by PLS $[Y, \tilde{U}\tilde{V}, W, \kappa]$. Since X is centered i.e. $\mathbb{I}_n' X = 0$, all the solutions to the equation $Y - f_\kappa = [\mathbb{I}_n \ \mathbf{X}] \theta$ have the same first component θ_1 . The remaining p components differ; the shortest Euclidean norm solution satisfies $[V' \theta_{2:p+1}]_{1:r} = \tilde{\theta}_{2:r+1}^{\text{PLS},\kappa}$ and $[V' \theta_{2:p+1}]_j = 0$ for all $r+1 \leq j \leq p$. Hence, $\theta_{2:p+1} = \tilde{V} \tilde{\theta}_{2:r+1}^{\text{PLS},\kappa}$ and from (8), $\hat{\theta}_{2:p+1}^{\text{PLS},\kappa}$ is the shortest norm solution. \square

Proposition 3. $\kappa \mapsto \|\hat{\theta}^{\text{PLS},\kappa}\|$ is non decreasing.

Proof. For a full column-rank matrix \mathbf{X} , this result is proved by De Jong (1995). For a non full column-rank matrix, using the same notations as in the proof above, the De Jong's result states that $\kappa \mapsto \tilde{\theta}^{\text{PLS},\kappa}$ is non decreasing; since the columns of \tilde{V} are pairwise orthogonal with norm 1, we have $\|\tilde{\theta}^{\text{PLS},\kappa}\| = \|\hat{\theta}^{\text{PLS},\kappa}\|$ thus concluding the proof. \square

REFERENCES

- Albert, A. and Anderson, J. (1984). On the Existence of Maximum Likelihood Estimates in Logistic Regression Models. *Biometrika*, 71(1):1–10.
- Alon, U., Barkai, N., Notterman, D., Gish, K., Ybarra, S., Mack, D., and Levine, A. (1999). Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proc. Natl. Acad. Sci. USA*, 96(12):6745–6750.

- Ambroise, C. and McLachlan, G. (2002). Selection bias in gene extraction on the basis of microarray gene-expression data. *PNAS*, 99(10):6562–6566.
- Bastien, P. (2004). PLS-Cox model : application to gene expression. In *Proceedings in Computational Statistics*, pages 655–662. Physica-Verlag, Springer.
- Bastien, P., Esposito Vinzi, V., and Tenenhaus, M. (2004). PLS generalised linear regression. *Computational Statistics & Data Analysis*, 48(1):17–46.
- Bates, D. (1983). The derivative of $|X'X|$ and its uses. *Technometrics*, 25(4):373–376.
- Bull, S., Mak, C., and Greenwood, C. (2001). A modified score function estimator for multinomial logistic regression in small samples. *Computational Statistics & Data Analysis*, 39:57–74.
- De Jong, S. (1995). PLS shrinks. *Journal of Chemometrics*, 9:323–326.
- Ding, B. and Gentleman, R. (2004). Classification Using Generalized Partial Least Squares. Technical Report 5, Bioconductor Project Working Papers. <http://www.bepress.com/bioconductor/paper5>.
- Fahrmeir, L. and Tutz, G. (2001). *Multivariate statistical modelling based on generalized linear models*. 2nd ed. Springer Series in Statistics. New York.
- Fort, G. and Lambert-Lacroix, S. (2005). Classification using Partial Least Squares with Penalized Logistic Regression. *To appear in Bioinformatics*.
- Frank, I. and Friedman, J. (1993). A statistical view of some chemometrics regression tools, with discussion. *Technometrics*, 35(2):109–148.
- Furey, T., Cristianini, N., Duffy, N., Bednarsky, D., Schummer, M., and Haussler, D. (2000). Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics*, 16:906–914.
- Garthwaite, P. (1994). An interpretation of partial least squares. *Journal of the American Statistical Association*, 89(425):122–127.
- Goutis, C. (1996). Partial Least Squares algorithm yields shrinkage estimators. *The Annals of Statistics*, 24(2):816–824.
- Green, P. (1984). Iteratively Reweighted Least Squares for Maximum Likelihood Estimation, and some Robust and Resistant Alternatives. *Journal of the Royal Statistical Society, Series B*, 46(2):149–192.

- Guyon, I., Weston, J., Barnhill, S., and Vapnik, V. (2002). Gene Selection for Cancer Classification using Support Vector Machines. *Machine Learning*, 46:389–422.
- Helland, I. (1988). On the structure of Partial Least Squares Regression. *Communications in Statistics. Simulation and Computation.*, 17(2):581–607.
- Helland, I. (1990). Partial Least Squares Regression and Statistical Models. *Scandinavian Journal of Statistics*, 17(2):97–114.
- Hoskuldsson, P. (1988). PLS Regression Methods. *Journal of Chemometrics*, 2:211–228.
- Jolliffe, I. (1986). *Principal Component Analysis*. Springer Verlag.
- Lesaffre, E. and Albert, A. (1989). Partial separation in logistic discrimination. *Journal of the Royal Statistical Society, Series B*, 51:109–116.
- Lingjaerde, O. and Christophersen, N. (2000). Shrinkage structure of Partial Least Squares. *Scandinavian Journal of Statistics*, 27:459–473.
- Marx, B. D. (1996). Iteratively Reweighted Partial Least Squares estimation for Generalized Linear Regression. *Technometrics*, 38(4):374–381.
- Naes, T. and Martens, H. (1985). Comparison of prediction methods for multicollinear data. *Communications in Statistics. Simulation and Computation.*, 14:545–576.
- Nguyen, D. and Rocke, D. (2002a). Multi-class cancer classification via Partial Least Squares with gene expression profiles. *Bioinformatics*, 18(9):1116–1226.
- Nguyen, D. and Rocke, D. (2002b). Tumor classification by Partial Least Squares using microarray gene expression data. *Bioinformatics*, 18(1):39–50.
- Phatak, A. and De Jong, S. (1997). The geometry of partial least squares. *Journal of Chemometrics*, 11:311–338.
- Phatak, A., Reilly, P., and Penlidis, A. (2002). The asymptotic variance of the univariate pls estimator. *Linear Algebra and its Applications*, 354:245–253.
- Santner, T. and Duffy, D. (1986). A note on A. Albert and J.A. Anderson’s Conditions for the Existence of Maximum Likelihood Estimates in Logistic Regression Models. *Biometrika*, 73(3):755–758.
- Scherf, U., Ross, D., Waltham, M., Smith, L., Lee, J., Tanabe, L., Kohn, K., Reinhold, W., Myers, T., Andrews, D., Scudiero, D., Eisen, M., Sausville, E., Pommier, Y., Botstein, D., Brown, P., , and J.N., W. (2000). A gene expression database for the molecular pharmacology of cancer. *Nature Genetics*, 24(3):236–244.

- Wold, H. (1975). Soft modelling by latent variables: The non-linear iterative partial least squares (NIPALS) approach. In *Perspect. Probab. Stat., Pap. Honour M. S. Bartlett Occas. 65th Birthday*, pages 117–142.
- Zhu, J. and Hastie, T. (2004). Classification of Gene Microarrays by Penalized Logistic Regression. *Biostatistics*, 5:427–443.

APPENDIX B. TABLES AND FIGURES

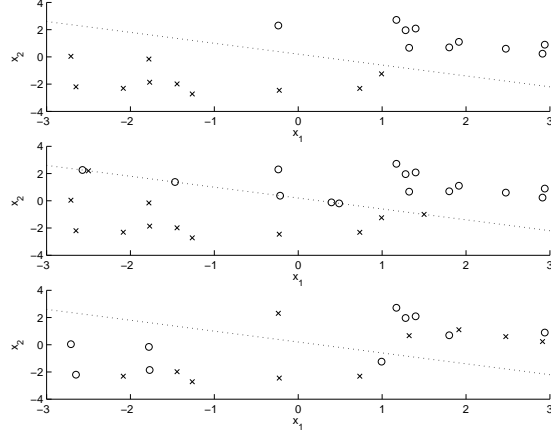


FIGURE 1. n points with coordinates $(\mathbf{Z}_2^{(k)}, \mathbf{Z}_3^{(k)})$ and label $\mathbf{Y}_k = 0$ (with a \times -mark) or label $\mathbf{Y}_k = 1$ (with a \circ mark). [Top] Separation : $\exists \hat{\theta}, \forall k, (\mathbf{Z}\hat{\theta})_k > 0$ if $\mathbf{Y}_k = 1$ and $(\mathbf{Z}\hat{\theta})_k < 0$ if $\mathbf{Y}_k = 0$. [Middle] Quasi-separation : $\exists \hat{\theta}, \forall k, (\mathbf{Z}\hat{\theta})_k \geq 0$ if $\mathbf{Y}_k = 1$ and $(\mathbf{Z}\hat{\theta})_k \leq 0$ if $\mathbf{Y}_k = 0$. [Bottom] Overlap : none of the two previous cases.

κ	NR		IRPLS		PLSGLR		IRPLSF		RPLS	
	T	L	T	L	T	L	T	L	T	L
1	20	13.90	12	8.27 (*)	19	13.71	9	6.77 (*)	18	10.29
2	8	7.85	8	6.60 (*)	9	7.84	8	7.93	8	7.97
3	7	4.01	-	-	13	2.42	7	5.00	7	5.00
4	10	1.92 (*)	-	-	14	0 (*)	7	2.00	8	2.02
5	8	0 (*)	-	-	10	0 (*)	7	0	9	0
6	11	0 (*)	-	-	13	0 (*)	11	0	11	0
min	4	-	-	-	5	-	4	-	5	-

TABLE 1. *Colon data*. For different methods and different values of κ , number of misclassified samples in the test set (column T) and mean number of misclassified samples in the learning set (column L).

	NR	IRPLSF	RPLS	PLSGLR	k-WNN	k-NN	DLDA	DQDA
mean	0.163	0.148	0.153	0.290	0.160	0.241	0.286	0.338
std	0.064	0.062	0.060	0.112	0.072	0.067	0.140	0.141
κ	3.27	3	2.82	1.01	5.64	7.37	-	-

TABLE 2. *Colon data*. Test set error rate : mean value and standard deviation (std). The last row shows the mean value of κ (or k for k-NN and k-WNN)

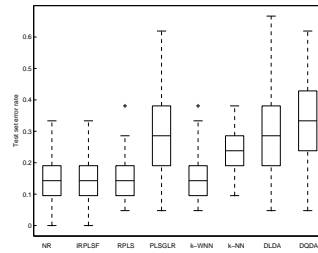


FIGURE 2. *Colon data*. Test set error rate in the resampling Analysis.

	MNR		MIRPLSF		MRPLS	
κ	T	L	T	L	T	L
1	12	10.4	7	6.80	7	6.77
2	4	0 (*)	2	0	3	0
3	5	0 (*)	2	0	2	0
4	1	0 (*)	2	0	2	0
5	2	0 (*)	2	0	2	0
6	2	0 (*)	2	0	2	0
min	1	-	1	-	1	-

TABLE 3. *NCI data* $p = 1299$. For different methods and different values of κ , number of misclassified samples in the test set (column T) and mean number of misclassified samples in the learning set (column L).

	MNR	MIRPLSF	MRPLS	k-NN	k-WNN
mean	0.054	0.043	0.046	0.056	0.055
std	0.062	0.055	0.058	0.060	0.062
κ	3.03	3.25	3.34	1.14	1.34
\mathcal{C}	0.578	0.146	0.553	-	-

TABLE 4. *NCI data* $p = 1299$. Test set error rate : mean value and standard deviation (std). The last two rows give the mean value of the parameter κ (or k for k-NN and k-WNN), and of the contrast \mathcal{C} .

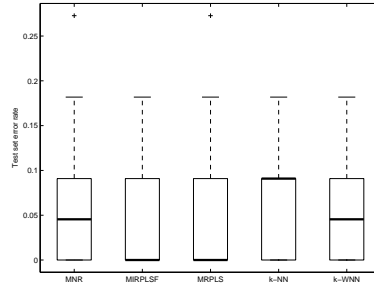


FIGURE 3. *NCI data* $p = 1299$. Test set error rate for the resampling analysis

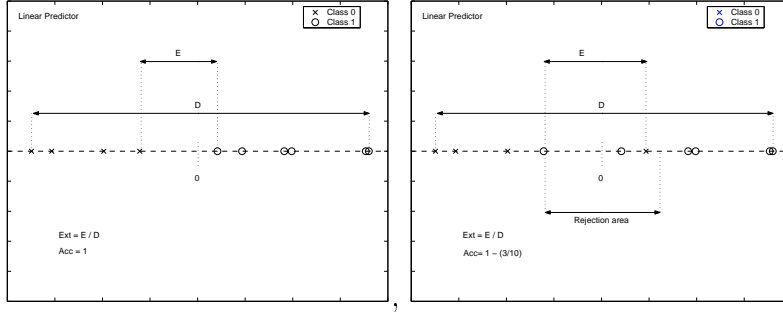


FIGURE 4. Estimate of the linear predictor for 10 samples, with true class '0' (drawn with a \times mark) and true class '1' (drawn with a \circ mark): a sample is classified 'class 1' iff the linear predictor is positive. [left] No errors : $\text{Suc}=1$, $\text{Ext} > 0$ and $\text{Acc} = 1$. [right] Two errors : $\text{Suc}=1-(2/10)$, $\text{Ext} < 0$ and $\text{Acc} = 1-(3/10)$ since there are 3 points in the Rejection area.

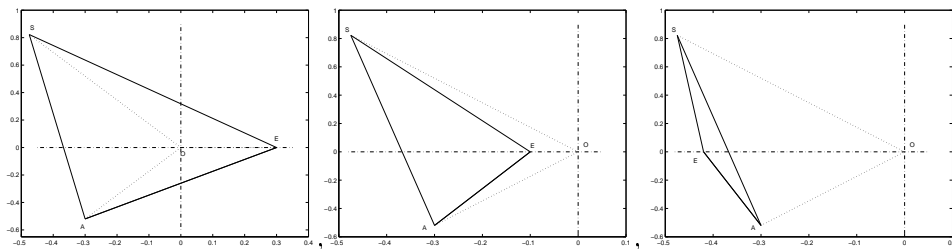


FIGURE 5. Q corresponds to the (signed) surface of the triangle with solid lines. In the first two cases, the surface is positive; in the last one, it is negative.

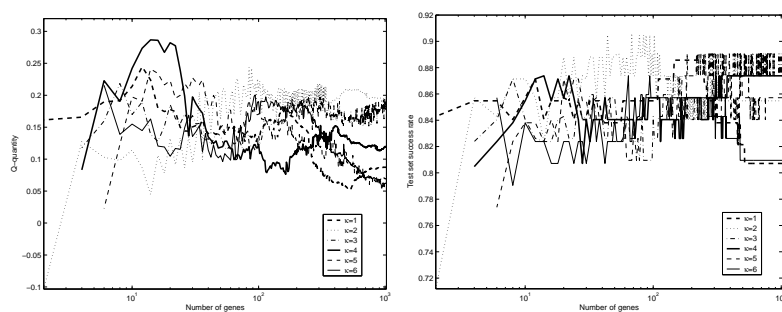


FIGURE 6. *Colon data*. Q -quantity and test set success rate, for the different nested models selected by IRPLSF-RFE.

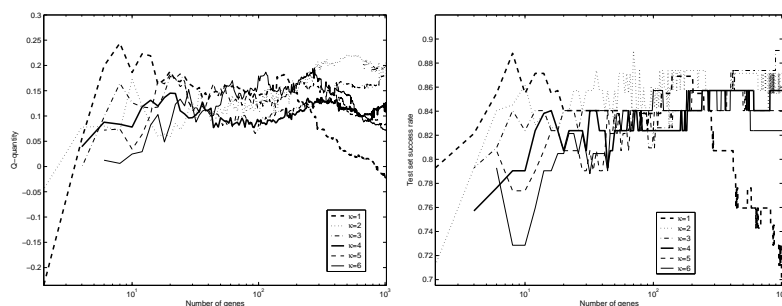


FIGURE 7. *Colon data*. Q -quantity and test set success rate, for the different nested models selected by RPLS-RFE.

	Q		0.5 (Suc Ext)		0.5 (Suc Acc)		Suc	
	value	p (κ)	value	p (κ)	value	p (κ)	value	p (κ)
NR (init 1)	0.3354	12 (4)	0.0569	12 (4)	0.3021	28 (6)	0.9048	64 (2)
NR (init 2)	0.3047	18 (3)	0.0577	8 (4)	0.2952	78 (6)	0.8929	422 (3)
IRPLSF (init 1)	0.2867	14 (4)	0.0378	20 (4)	0.2868	158 (1)	0.9048	44 (2)
IRPLSF (init 2)	0.2867	14 (4)	0.0378	20 (4)	0.2868	158 (1)	0.9048	32 (2)
RPLS	0.2440	8 (1)	0.0072	8 (1)	0.2814	140 (1)	0.8905	70 (2)

TABLE 5. *Colon data*. Feature selection : optimal model exhibited by NR-RFE, IRPLSF-RFE and RPLS-RFE for different measures of the quality : we report the value of the quality 'value', the size of the model 'p' and the value of the hyper-parameter ' κ ' with which the optimum is reached.