Cox model in high dimensional and low sample size settings

Philippe Bastien

L'Oréal Recherche - Aulnay

pbastien@rd.loreal.com

INTRODUCTION

The evolution of cancer is more certainly linked to a complex interplay of genes rather than a single gene activity. Multivariate analysis, which can exploit the correlated pattern of gene expression display by genes behaving jointly, such as genes performing the same functions or genes operating along the same pathway, can become a very useful diagnostic tool to determine molecular predictor of survival on the basis of gene expression. However, problems encountered in multiple regression due to multicollinearity, or ill posed problems with many descriptors and only a few samples, occur in the same way when we are dealing with censored data. The proportional hazard regression model suggested by Cox in 1972 to study the relationship between the time to event and a set of covariates, in the presence of censoring, is the model most commonly used for the analysis of survival data. However, like multivariate regression models, it supposes that there are more observations than variables, complete data, and variables not strongly correlated between them. These constraints are often crippling in practice, as for example in oncology when the expression of several thousand of genes is collected from bio-ships and used as molecular predictors of survival.

Prediction in high-dimensional and low-sample size settings already arose in Chemistry in the eighties. The PLS regression (S.Wold et al., 1983; M.Tenenhaus, 1998), which could be viewed as a regularization method based on dimension reduction, was developed as a

Chemometric tool in an attempt to find reliable predictive models with spectral data. Nowadays, the difficulty encountered with the use of transcriptomic data for classification or prediction, using very large matrices, is of comparable nature. It was thus natural to use PLS regression principles in this new context.

Bastien and M.Tenenhaus (2001), showing that PLS univariate regression could be obtained as a series of simple and multiple regressions, and replacing the succession of simple and multiple linear regression by a succession of simple and multiple generalized linear regressions, extended PLS regression to any generalized linear regression and to the Cox model as a special case. Their approach is similar to that of Garthwaite (1994), but can also cope with missing data by using the principles of the NIPALS algorithm (H.Wold, 1966),
In the context of censored data, they proposed to modelize the occurrence of prematurely graying hair with data on more than 4000 adult males. Both PLS complementary log log regression and PLS Cox regression have been carried out depending on the hypothesis formulated on the data. Further improvements have been proposed in Bastien, Esposito Vinzi, and M. Tenenhaus. (2005) in the case of categorical descriptors with model validation by Bootstrap resampling.

The Cox proportional hazards model

The model assumes the following hazard function for the occurrence of an event at time *t* in the presence of censoring:

$$\lambda(t) = \lambda_0(t)\exp(X\beta)$$

where $\lambda_0(t)$ is an unspecified baseline hazard function, β the vector of the regression coefficients, and *X* the matrix of prognosis factors which will be the gene expression in the following. The event could be the death or the cancer relapse. Based on the available data, the Cox's partial likelihood can be written as:

$$PL(\beta) = \prod_{k \in D} \frac{\exp(\beta' x_k)}{\sum_{j \in R_k} \exp(\beta' x_j)} \qquad (1)$$

Where D is the set of indices of the events and R_k denotes the set of indices of the individuals at risk at time t_k

The goal is to find the coefficients β which minimize the negative log partial likelihood function. Note that when p > n, there is no unique β to maximise this partial likelihood function. Even when p ≤ n, covariates could be highly correlated and regularization may still be required in order to reduce the variances of the estimates and to improve the prediction performance.

The PLS-Cox algorithm

The algorithm consists of 4 steps:

1) computation of the m PLS components $\mathbf{t}_h$ (h = 1,…,m);
2) Cox regression on the m retained PLS components;
3) expression of the hazard function in terms of the original explanatory variables;
4) Bootstrap validation of the coefficients in the final PLS-Cox model.

The first step will be described in details below:

Let $\mathbf{X}=\{\mathbf{x}_1, \ldots, \mathbf{x}_p\}$ be the matrix of the p explanatory variables $\mathbf{x}_j$'s. The objective is to search for m PLS orthogonal components $\mathbf{t}_h$'s defined as linear combinations of $\mathbf{x}_j$.

Computation of the first PLS component $\mathbf{t}_1$

Step 1 : Compute the regression coefficient a_{1j} of $\mathbf{x}_j$ in the Cox regression on $\mathbf{x}_j$ for each variable $\mathbf{x}_j$, j = 1 to p;

Step 2 : Normalise the column vector $\mathbf{a}_1$ made by a_{1j}'s: $\mathbf{w}_1 = \mathbf{a}_1/||\mathbf{a}_1||$;

Step 3 : Compute the component $\mathbf{t}_1 = \mathbf{X}\mathbf{w}_1/\mathbf{w}_1'\mathbf{w}_1$.

Computation of the second PLS component $\mathbf{t}_2$

Step 1 : Compute the regression coefficient a_{2j} of $\mathbf{x}_j$ in the Cox regression on $\mathbf{t}_1$ and $\mathbf{x}_j$ for each variable $\mathbf{x}_j$, j = 1 to p;

Step 2 : Normalise the column vector $\mathbf{a}_2$ made by a_{2j}'s: $\mathbf{w}_2 = \mathbf{a}_2/||\mathbf{a}_2||$;

Step 3 : Compute the residual matrix $\mathbf{X}_1$ of the linear regression of $\mathbf{X}$ on $\mathbf{t}_1$;

Step 4 : Compute the component $\mathbf{t}_2 = \mathbf{X}_1\mathbf{w}_2/\mathbf{w}_2'\mathbf{w}_2$;

Step 5 : Express the component $\mathbf{t}_2$ in terms of $\mathbf{X}$: $\mathbf{t}_2 = \mathbf{X}\mathbf{w}_2^*$.

Computation of the h-th PLS component $\mathbf{t}_h$

In the previous steps, the PLS components $\mathbf{t}_1,\ldots, \mathbf{t}_{h-1}$ have been yielded. The component $\mathbf{t}_h$ is obtained by iterating the search for the second component.

Step 1 : Compute the regression coefficient a_{hj} of $\mathbf{x}_j$ in the Cox regression on $\mathbf{t}_1,\ldots, \mathbf{t}_{h-1}$ and $\mathbf{x}_j$ for each variable $\mathbf{x}_j$, j = 1 to p;

Step 2 : Normalise the column vector $\mathbf{a}_h$ made by a_{hj}'s: $\mathbf{w}_h = \mathbf{a}_h/||\mathbf{a}_h||$;

Step 3 : Compute the residual matrix $\mathbf{X}_{h-1}$ of the linear regression of $\mathbf{X}$ on $\mathbf{t}_1,\ldots, \mathbf{t}_{h-1}$;

Step 4 : Compute the component $\mathbf{t}_h = \mathbf{X}_{h-1}\mathbf{w}_h/\mathbf{w}_h'\mathbf{w}_h$;

Step 5 : Express the component $\mathbf{t}_h$ in terms of $\mathbf{X}$: $\mathbf{t}_h = \mathbf{X}\mathbf{w}_h^*$.

Remarks

1) Computation of the PLS component $\mathbf{t}_h$ can be simplified by setting to 0 those regression coefficients a_{hj} that are not significant. Only significant variables will then contribute to the computation of the PLS component.

2) The number m of PLS components to be retained may be chosen by cross-validation on the predictive power of the model or by observing that the component $\mathbf{t}_{m+1}$ is not significant because none of the coefficients $a_{m+1,j}$ is significantly different from 0.

3) The proposed algorithm may cope with missing data. Let $\mathbf{x}_{h-1,i}$ be the column vector obtained by transposing the i-th row of $\mathbf{X}_{h-1}$. The value $\mathbf{t}_{hi} = \mathbf{x}_{h-1,i}'\mathbf{w}_h/\mathbf{w}_h'\mathbf{w}_h$ of the i-th case on the component $\mathbf{t}_h$ represents the slope of the OLS line without constant term related to the cloud of points ($\mathbf{w}_h$, $\mathbf{x}_{h-1,i}$). This slope may be computed even when there are missing data. In such a case, in computing the h-th PLS component, the denominator of Step 4 is computed only on the data available also for the numerator.

Other approaches

Another approach to extend PLS regression to survival data, by means of generalized linear models, was done by Park *et al.* (2002). They reformulated the Cox model as a Poisson model for the censored indicator variable using Whitehead (1980) who showed the equivalence of the Poisson model and the survival Cox model, the likelihoods being proportional at their maximum. Park et al. then applied the formulation of PLS proposed by Marx (1996) for the generalized linear models to derive the PLS components. Marx used the fact that in the context of exponential family, maximum likelihood estimates are obtained by an iterative reweighted least squares procedure. Its approach consists of replacing the iterative weighted least squares step by a sequence of PLS regressions. However the reformulation of the censored problem as a Poisson regression increases the dimension of the problem. Multiple observations must be created for each single individual depending on the number of distinct failure times where the individual is at risk. Moreover when the number of covariates is large the algorithm may fail to converge (Li and Gui, 2004).

In Nguyen and Rocke (2002), the so-called Partial Least Squares Proportional Hazard regression is proposed for the application to gene expression data from DNA microarrays. Their proposal actually consists of a two-stage strategy of analysis: PLS regression of survival time on the predictors at the first stage, in order to reduce data dimensionality and extract PLS components; proportional hazard regression model at the second stage in order to estimate the survival distribution. This two-stage strategy does not take into account the censoring information in the estimation of PLS components, thus inducing bias in their estimates.

More recently, Li and Gui (2004) proposed a formulation of the PLS Cox algorithm named Partial Cox Regression described below, very similar to the one presented by Bastien and Tenenhaus (2001). They also proposed to generalized the Garthwaite approach but with an alternative weighting scheme for the determination of the PLS components using a pseudo generalized covariance. As a direct extension of the Garthwaite approach they algorithm does not cope with missing data.

Partial Cox Regression algorithm

Let $\mathbf{X_1}=\{\mathbf{x}_{11}, \ldots, \mathbf{x}_{p1}\}$ be the matrix of the *p* centred explanatory variables $\mathbf{x}_j$'s

Computation of the first PLS component $\mathbf{t}_1$

Step 1 : Compute the regression coefficient a_{1j} of $\mathbf{x}_{1j}$ in the Cox regression on $\mathbf{x}_{1j}$ for each variable $\mathbf{x}_{1j}$, j = 1 to p;

Step 2 : Compute the component $\mathbf{t}_1 = \sum_{j=1}^{p} w_{1j} a_{1J} x_{1j}$ with $w_{1j} = \dfrac{\operatorname{var}(x_{1j})}{\sum_{l=1}^{p} \operatorname{var}(x_{1l})}$

Computation of the second PLS component $\mathbf{t}_2$

Step 0 : Compute the residual matrix $\mathbf{X}_2$ of the linear regression of $\mathbf{X_1}$ on $\mathbf{t}_1$;

$$x_{2j} = x_{1j} - \frac{x'_{1j}t_1}{t'_1t_1}t_1$$

Step 1 : Compute the regression coefficient a_{2j} of $\mathbf{x}_{2j}$ in the Cox regression on $\mathbf{t}_1$ and $\mathbf{x}_{2j}$ for each variable $\mathbf{x}_{2j}$, j = 1 to p;

Step 2 : Compute the component $\mathbf{t}_2 = \sum_{j=1}^{p} w_{2j}a_{2J}x_{2j}$ with $w_{2j} = \frac{\mathrm{var}(x_{2j})}{\sum_{l=1}^{p}\mathrm{var}(x_{2l})}$

Computation of the h-th PLS component $\mathbf{t}_h$

In the previous steps, the PLS components $\mathbf{t}_1,\ldots,\mathbf{t}_{h-1}$ have been yielded. The component $\mathbf{t}_h$ is obtained by iterating the search for the second component.

Step 0 : Compute the residual matrix $\mathbf{X}_h$ of the linear regression of $\mathbf{X}_{\mathbf{h-1}}$ on $\mathbf{t}_{h-1}$

$$x_{hj} = x_{h-1j} - \frac{x'_{h-1j}t_{h-1}}{t'_{h-1}t_{h-1}}t_{h-1}$$

Step 1 : Compute the regression coefficient a_{hj} of $\mathbf{x}_{1j}$ in the Cox regression on $\mathbf{t}_1, \ldots, \mathbf{t}_{\mathbf{h-1}}$ and $\mathbf{x}_{hj}$ for each variable $\mathbf{x}_{hj}$, j = 1 to p;

Step 2 : Compute the component $\mathbf{t}_h = \sum_{j=1}^{p} w_{hj}a_{hj}x_{hj}$ with $w_{hj} = \frac{\mathrm{var}(x_{hj})}{\sum_{l=1}^{p}\mathrm{var}(x_{hl})}$

PLS-Cox with high dimensional data

The PLS-Cox algorithm is sequential in the estimation of the weights used in the determination of the PLS components. When the number of descriptors exceeds by far the number of observations, as it is the case with gene expression where the number of genes can reach several tens of thousand, the algorithm becomes computer-intensive and technical problems may arise. PLS, as a dot product algorithm, being invariant under orthogonal transformation of the *X* and/or *Y* variables, PLS based on the ***X*** principal components is equivalent to PLS based on the original descriptors matrix ***X***. It seems then very appealing, when dealing with very large data sets, to use PLS-Cox regression on the ***X*** principal components. This algorithm will be named PC PLS-Cox in the text. This modification of the PLS algorithm in the setting of Cox regression has been also proposed by Li and Gui (2004) in a similar approach, in combination with their PCR algorithm described above. Based on the Rosenwald et al. (2002) published dataset of gene expression from diffuse large B-cell lymphoma, they obtained better predictive performance by combining the principal components to their Partial Cox Regression algorithm.

The invariance property of PLS linear regression under orthogonal transformation does not hold for PLS generalized linear regression giving modified coefficient estimates.

Even if the invariance property is not an optimality criterion, an alternative algorithm sharing this property can be proposed, based on a generalisation of a linear kernel PLS algorithm, as described hereinafter.

Linear Kernel PLS algorithms

The same computational problems posed by very large matrices already arose in Chemometrics and solutions were proposed in the nineties using linear kernel variants of the PLS algorithm. The objective of these methods was to obtain PLS components by working on a condensed kernel matrix of a considerably smaller size than the original matrices X and Y. let’s note that the term kernel is used here more in an attempted to reduce the computational complexity in the input space (linear kernel), than a nonlinear transformation into a feature space as it is the case with Support Vector Machine.

The first kernel algorithm was developed by Lingren *et al.* (1993) for data matrices with many observations. Conversely, for matrices with many variables and only a few objects ($p >> n$), Rännar *et al.* (1994) developed a quick and efficient algorithm named Kernel-PLS based on the deflation of the small matrices ***XX’*** and ***YY’***, which avoid the use of the large

X matrix. They used the result of Höskuldson (1988) and Manne (1987) who demonstrated that the PLS components can be calculated as the eigenvector associated to the maximum eigenvalues of the matrices $X_{h-1}X'_{h-1}Y_{h-1}Y'_{h-1}$, where X_{h-1} and Y_{h-1} are the deflated matrices (*i.e.* residuals in the multiple regression of ***X*** and ***Y*** on the previous PLS components). These algorithms, however, does not extend to the Cox model. It was thus tempting to develop a new kernel PLS algorithm which can be then naturally extended to the Cox model.

Modified Canonical PLS algorithm:

A simple kernel PLS algorithm, named Modified Canonical PLS, can be derived based on the kernel matrix XX'=ZZ'.

Let X be the matrix of centred explanatory variables

```
Step 0      SVD of X ($X = UL^{1/2}V'$) or NIPALS in case of missing data
Let $Z_1 = Z = UL^{1/2}$
For h=1 to a,
    Step 1 :    Compute the component $t_h := Z_h Z'_h y$
    Step 2 :    Normalize the vector $t_h$ : $\|t_h\| = 1$
    Step 2      Compute the residual $Z_{h+1}$ in the linear regression of $Z_h$ on $t_{h+1}$ :
                $Z_{h+1} = (I - t_{h+1}t'_{h+1})Z_h$
```

Let's note that t_h could also be expressed as $t_h = U_{h-1}LU'_{h-1}y = \sum_i \lambda_i^2 (u'_i y)u_i$ with $Z_h = U_h L^{1/2}$, $U_h = (I - t_h t'_h)U_{h-1}$, and L the diagonal matrix of non zero eigenvalues λ_i^2

Since all calculations are done in the canonical space, obtained after one singular value decomposition, a considerable gain in speed is achieved. PLS coefficients can then be easily back transformed to the original co-ordinate system: $\beta = UL^{-1/2}V'TT'y$

Nguyen and Rocke (2004) and Nguyen (2005) have also proposed an expression for the PLS components involving eigenvectors of the matrices ***X'X*** and **XX'.** However the formulas are very complex and explicit only for the first components.

This algorithm is inspired by the "Canonical PLS regression" algorithm of de Jong et al. (2001), but with a more explicit analytical expression of the PLS components based on eigenvalues and eigenvectors of the ***X*** matrix. It amounts, to replace the Gram-Schmidt orthonormalisation procedure, used in "Canonical PLS regression" as an implicit deflation, by an explicit deflation of the singular vectors:

$$(T_1, T_2, ..., T_a) = GS(ULU'y, UL^2U'y, ..., UL^aU'y) \propto (ULU'y, U_1LU_1'y, ..., U_aLU_a'y)$$

where the symbol $\propto$ means that the terms in the left series are equal to the terms in the right series up to a normalization.

This kernel PLS algorithm can be generalized (MCPLS Cox) by replacing the dot product $Z_h'y$ in step 1 of the MCPLS algorithm by $g(Z_h, y)$ with $g(Z_h, y)$ being the coefficients of z_{hj} in the Cox regression on $t_1, ..., t_{h-1}, z_{hj}$.

In order to be compared, the PC-PCR, PC-PLS Cox, and MCPLS Cox algorithms have been expressed as linear combinations of the deflated singular vectors of the ***X*** matrix.

PC PLS-Cox $\quad t_h \propto Z_{h-1}g(Z_{h-1}, y) = U_{h-1}g(U_{h-1}, y)$

PC-PCR: $\quad t_h \propto Z_{h-1}L_{h-1}g(Z_{h-1}, y) = U_{h-1}L^{1/2}L_{h-1}L^{-1/2}g(U_{h-1}, y)$

MCPLS-Cox: $t_h \propto U_{h-1} L g(U_{h-1}, y)$

with $Z_0 = Z,\ L_h = Z'_h Z_h,\ Z_h = (I - t_h t'_h) Z_{h-1}$

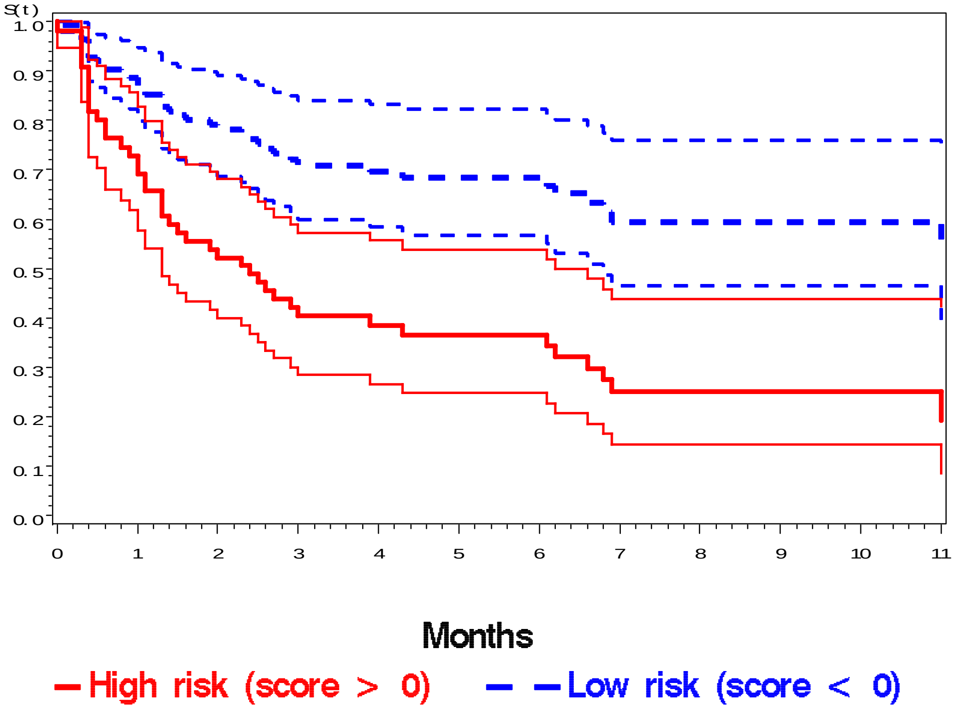
The PLS components are all expressed as $t_h \propto U_{h-1} W g(U_{h-1}, y)$, with different expressions of ***W*** according to the algorithms. This gives various trade-off between fit and stability, which correspond to different paths in the parameter space, the best one depending on the data.

Application

The DLBCL published dataset of Rosenwald et al. (2002) has been reanalysed using the PC-PLS Cox model. This dataset includes a total of 240 patients with DLBCL, including 138 patient deaths during the follow-ups with median death time of 2.8 years and 30% of right censored survival time. The gene expression measurements of 7399 genes are available for the analysis.

The PC-PLS Cox regression provides three significant PLS components. The PLS components being centred, the score function $T\beta$, derived from the Cox model on the three PC-PLS Cox components, is centred on 0 which represents the average risk. Therefore subjects with a positive risk score belong to the high risk group, and conversely subjects with a negative risk score belong to the low risk group.

The graph below displays the survival curves with their associated 95% confidence intervals for the low and high risk groups, based on the test set. We observe a significant difference in the risk of death between the high and low groups.



Non linear kernel PLS-Cox algorithm

Rosipal and Trejo (2001) were the first to propose a nonlinear extension of PLS regression using kernels. Assuming a nonlinear transformation of the input variables $\{x_i\}_{i=1}^{n}$ into a feature space F, i.e. a mapping $\Phi : x_i \in R^N \to \Phi(x_i) \in F$ they goal was to construct a linear PLS regression model in F. They derived an algorithm, named KPLS, for non-linear kernel PLS models by performing the PLS regression on $\Phi(X)$. It amounts to replace in the expression of PLS components the product *XX’* by $\Phi(X)\Phi(X)'$ using the so-called kernel trick which permits the computation of dot products in high-dimensional feature spaces using simple functions defined on pairs of input patterns: $\Phi(x_i)\Phi(x_j)' = K(x_i, x_j)$

Using the kernel functions corresponding to the canonical dot product in the feature space allows avoiding non linear optimization and the use of simple linear algebra.

However, this KPLS algorithm doesn't allow extension to generalised linear regression.

A better solution for generalization came 2 years later from Bennett and Embrecht (2003) who proposed, with their Direct Kernel PLS algorithm, to perform PLS regression on the kernel matrix K instead of $\Phi(X)$. DKPLS corresponds to a low rank approximation of the kernel matrix. As shown by Lewi (1995), latent vectors extracted from a data table are the same as those derived from the corresponding distance matrix up to a transformation of the later into a variance-covariance matrix. Moreover, A.Tenenhaus et al.(2006) demonstrated that for one dimensional output response PLS of $\Phi(X)$ (KPLS) is equivalent to PLS on $K^{1/2}$ (DKPLS).

Following Bennett and Embrechts, A.Tenenhaus et al recently proposed KLPLS, a kernelized version of generalized PLS regression (Bastien, Esposiro Vinzi, and M. Tenenhaus, 2005) in the framework of logistic regression, as an extension of PLS-logistic regression to non linear settings.

Using the previous works, it becomes straightforward to derive a (non linear) Kernel PLS Cox algorithm by replacing in the PLS Cox algorithm the X matrix by the kernel matrix K. Moreover, the introduction of an intercept when constructing the latent variables avoids the kernel centring used in the DKPLS algorithm (Bennett and Embrecht 2003, A.Tenenhaus et al. 2006.

The KPLS Cox algorithm is composed of 3 steps.

1/ Computation of the kernel matrix

2/ Computation of the PLS components

3/ Cox regression on the retained PLS components

The main kernel functions are:

Polynomial kernel $K(u,v) = \left(\langle u,v\rangle + c\right)^d$

Gaussian kernel $K(u,v) = \exp\left(-\frac{\left\|\langle u,v\rangle\right\|^2}{2\sigma^2}\right)$

The linear kernel $K(u,v) = \langle u,v\rangle$ appears as a particular polynomial kernel. As mentioned by A.Tenenhaus and al., the choice of the kernel function defines the relative position of the data points in the feature space.

Let’s note that non-linear kernel regression loses the explanation with the original descriptors, unlike linear kernel PLS regression. This could limit the interpretation of the results, as for example in genomics when PLS regression may be used to determine the genes belonging to transcriptomic signatures linked to the response.

Application

In order to demonstrate the performance of the KPLS Cox algorithm when dealing with non linearity in the relationship between input variables and the hazard function, data showing dramatic non linearity have been simulated.

Data have been simulated in order that independent processes represented by unobserved latent variables L_k are responsible for the systematic variation of both the survival time and the predictor variables with a highly non linear relationship between the hazard function and the latent variables. The predictor covariance matrix has a block structure with variables in the same group being highly correlated with each other, while between groups correlations are small. Moreover, groups of variables represented by their associated latent variables could be not related to the hazard function in order to include some noise in the data. The application in the context of transcriptomic data is straightforward with the variables corresponding to gene expression and the latent variables to the associated biological functions. Moreover, we suppose that the survival time is non-linearly linked to the biological functions. The aim is to find a molecular signature which could be used as a prognosis factor of survival.

We first generate for each latent variable k variables with some defined inter-correlation pattern represented by the matrix R.

To generate correlated multivariate normal data of k variables with a desired population inter-correlation pattern as represented by the matrix R, we take the following steps (Kaiser & Dickman 1962, Fan et al 2002):

1 Perform a PCA on the R correlation matrix in order to extract k principal components resulting in a matrix F of size kxk.

2 Generate k uncorrelated standardized random variables, each with N observations and then transpose to a kxN dimension matrix X.

3 Premultiply the uncorrelated data matrix X with the factor pattern matrix F. The resultant Z matrix ($Z_{(kxN)}=F_{(kxk)}\ X_{(kxN)}$) contains N observations on k correlated variables, as if the N observations were sampled from a population with population correlation pattern R. Then transpose back to an Nxk data matrix.

In the following, we suppose that the correlation between variables associated with the same latent variable is constant. By choosing their respective values, we can specify the percentage of variability explained by the first factorial axes of the descriptors matrix X.

Next, survival times are generated satisfying the proportional hazard model. We took survival times $Y_i = Y_{0i} exp(r)\ where\ Y_{0i} \approx Exp(\lambda_y)$ and censoring times $Z_i = Z_{0i}\exp(r)$ where $Z_{0i} \approx Exp(\lambda_z)$.

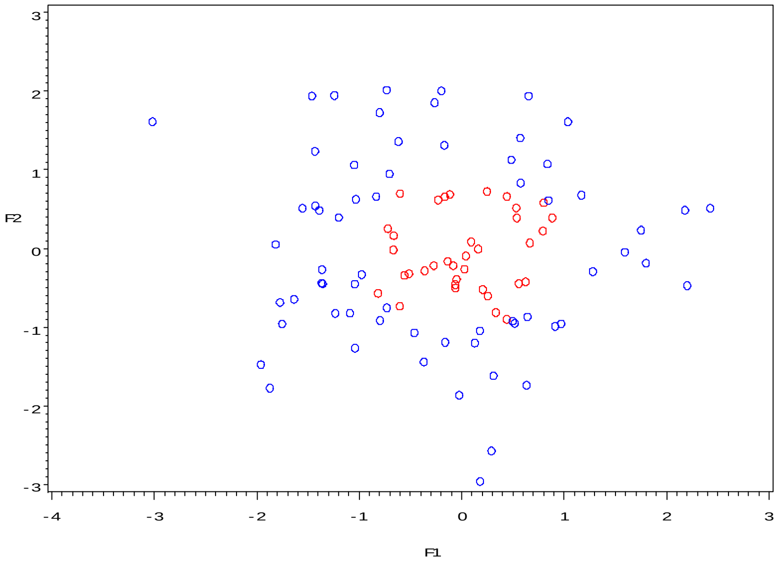
The observed times are $T_i = \min(Y_i, Z_i)$. We took $\lambda_y = 0.5\ and\ \lambda_z = 1$ which gives 1/3 censoring rate.

In the following r is chosen as a non linear function of the latent variables. In the example we choose three latent variables F1 to F3 formed from three groups of 100 variables each with correlation matrices R1 to R3 characterised by their off diagonal between variables correlations of 0.7, 0.6, and 0.5 respectively. The third group being not associated with survival. A PCA on the 300 simulated data shows eigenvalues of respectively 69.7, 60.85, and 49.87 associated to the first three principal axes.

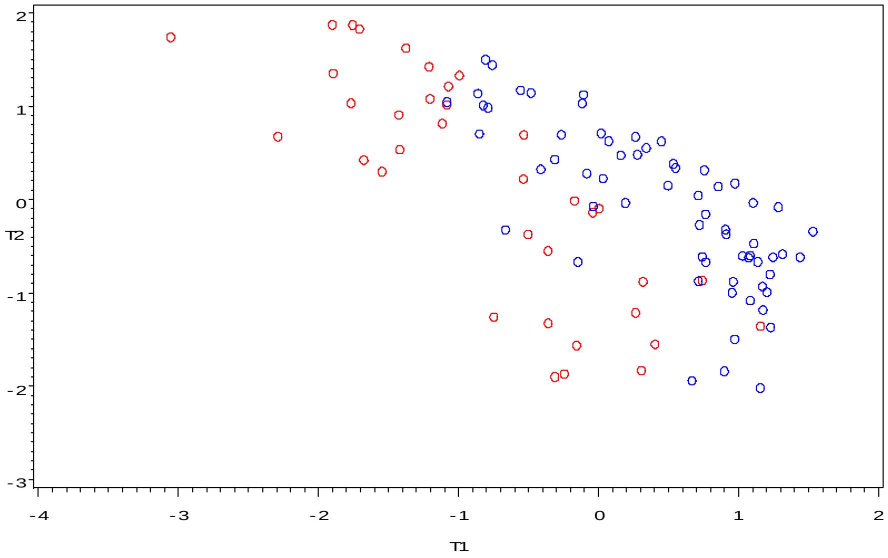
Let $r = F_1^2 + F_2^2$

The data have been separated in working (n=200) and test (n=100) sets.

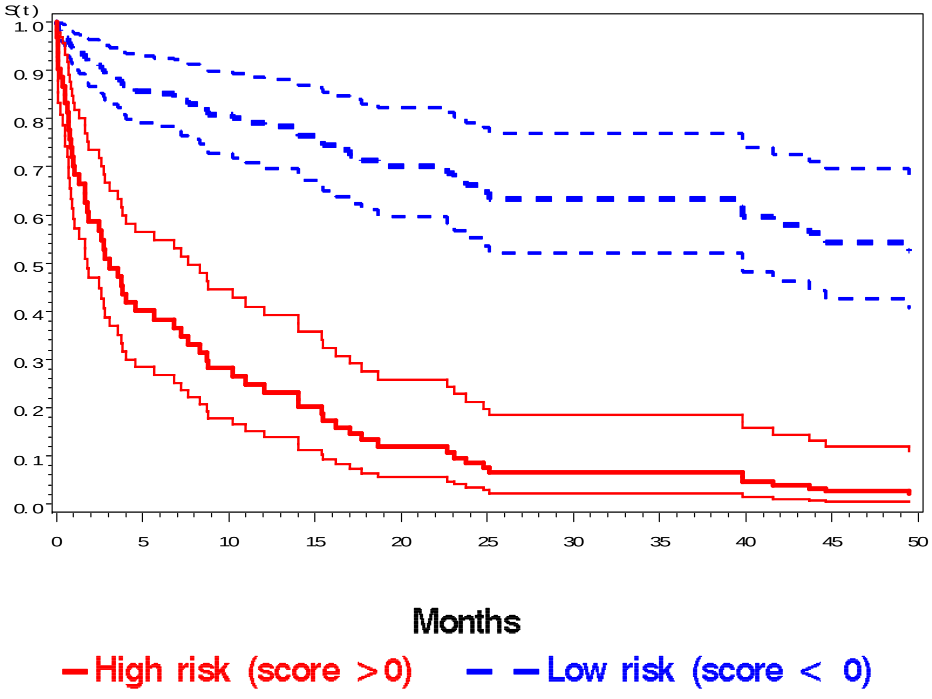
Graph 1 displays the projection on F1 and F2 of the 100 simulated test samples. Red circles correspond to data with low values for r (< median), and blue circles the otherwise.



The KPLS Cox model has two significant components. Graph 2 shows the projection of the test samples onto the first two KPLS Cox components. We observe a quasi linear separation of the blue and red circles in the feature space, the overlap being due to the noise brought by the third group of variables not linked to survival.



Graph 3 displays the survival curves with the associated confidence intervals for the low risk and high risk groups, based on the test set. Let's note that the PSL components being centred, the score function $T\beta$, derived from the Cox model on the KPLS Cox components, is centred on 0 which represents the average risk. Therefore subjects with a positive score function are said to be at high risk, and conversely subjects with a negative score function are said to be at low risk. We observe a highly significant difference in the risk of death between the high and low groups.



Penalized Cox regression

Other approaches to deal with high dimensional and low-sample size data in the framework of the Cox regression is to use penalized partial likelihood, including both L_1 (Lasso) and L_2 (Ridge) penalized estimations. Li and Luan (2003) where the first to investigate the L_2 penalized estimation. They developed a kernel Cox regression model for relating gene expression profiles to censored phenotypes in terms of function estimation in reproducing kernel Hilbert spaces. However, one limitation of the L_2 penalized estimation of the Cox model is that it uses all the genes in the prediction and does not provide a way of selecting relevant genes for prediction. On the other hand, in addition to improving on prediction accuracy through shrinkage, the nature of the L_1 constraint is such that interpretation is enhanced by “zeroing out” many covariates.

Cox-LASSO procedure

In the context of censored data, Tibshirani (1997) extended the LASSO procedure for variable selection with the Cox's proportional hazards model. Based on the Cox's partial likelihood (1) the lasso estimate β can be expressed as:

$$\beta = \arg\max l(\beta), subject\ to \sum_{j=1}^{p} \left|\beta_j\right| \le s \tag{2}$$

Where s is a tuning parameter that determine how many coefficients are shrunk to zero.

Let $\eta = \beta^T x, \mu = \frac{\partial l}{\partial \eta}, A = -\frac{\partial^2 l}{\partial \eta \eta^T}, and\ z = \eta + A^{-}\mu$ where $x = (x_1, ..., x_n)$ is the matrix of covariates.

Let $(z-\eta)^T A(z-\eta)$ be a one-term Taylor series expansion for $l(\beta)$, Tibshirani proposed a 4 steps iterative procedure to solve (2)

Step 1 Fix s and initialize $\hat{\beta} = 0$

Step 2 Compute η, μ, A, and z based on the current value of $\hat{\beta}$

Step 3 Minimize $(z - \beta^T X)^T A(z - \beta^T X)$ subject to $\sum_{j=1}^{p} \left|\beta_j\right| \le s$

Step 4 repeat step 2 and 3 until $\hat{\beta}$ does not change.

The strategy for solving step 3 is to express the usual Newton-Raphson update as an iterative reweighted least squares step, and then replace the weighted least squares step by a constrained weighted least squares procedure. However the quadratic programming procedure used in step 3 starts from the least squares solution and hence cannot be applied when p > n.

LARS-LASSO procedure

In 2004, Efron *et al.* proposed a highly efficient procedure called LARS for Least Angle Regression for variable Selection. This procedure can be used to perform variable selection with large matrices. The LARS procedure starts with all coefficients equal to zero, and finds the predictor most correlated with the response. It takes then the largest step in the direction of this predictor until another predictor becomes as much correlated with the current residual. LARS then proceeds in a direction equiangular between the two predictors until a third variable in turn becomes as much correlated with the current residual, as the previously selected ones. The stopping rule is based on a Cp-type criterion. At each step LARS adds one covariate to the model, so that after k steps, k of the $\hat{\beta}$ are non-zero. LARS is computationally thrifty. The computational cost for the entire steps is of the same order as that described for the usual unconstrained Least Squares solution for the full set of covariates.

Moreover, LARS can be modified to provide solution for the LASSO procedure. Using the connection between LARS and LASSO, Gui and Li (2005a) proposed the LARS-LASSO for gene selection in high-dimension and low-sample settings. Using a Choleski factorization of A, they transformed the step 3 minimisation in a constrained version of Ordinary least Squares which can be solved by the LARS-LASSO procedure.

Step 3 (modified) Minimize $(y-\beta^T\hat{X})^T(y-\beta^T\hat{X})$ subject to $\sum_{\mathrm{j=1}}^{\mathrm{p}}\left|\beta_{\mathrm{j}}\right| \le s$

Where $y = Tz,\ \hat{X} = TX, and\ A = TT^T$

They used their procedure in oncology for identifying important genes that are related to survival time, and for building a parsimonious Cox model for predicting the survival of future patients.

Deviance residuals

However the IRWLS iterations performed in the Cox-LASSO procedure counter balanced the efficiency of the LARS-LASSO algorithm and render the Gui and Li (2005a) algorithm computationally costly. Segal (2005), at the expense of some approximation, proposed to speed-up the calculations. He proposed to replace the survival times by the deviance residuals, a normalized version of Martingal residuals that result from fitting a null (intercept only) Cox regression model. The deviance residual is a measure of excess of death and can therefore be interpreted as measure of hazard. Moreover, Segal showed that the expression to be minimized in step 3 of the Cox-LASSO procedure can be approximate, at a first order Taylor approximation, by the deviance residual sum of squares.

$$(z-\beta^T X)^T A(z-\beta^T X) \approx RSS(\hat{D})$$

Therefore, in order to perform the Cox-Lasso procedure, initially compute the null deviance residuals and use these as outcomes for the LARS-LASSO algorithm.

Moreover both residual deviance and LARS-LASSO procedure are available in R or SPLUS with coxph() and lars(), and in SAS with the Proc PHREG and the new proc GLMSELECT procedure.

We propose to use the same idea in the setting of Partial least Squares. An alternative formulation of the PLS Cox model could be derived by fitting the deviance residuals with a simple univariate PLS regression. We will compare both models based on the evaluation of their predictive performance using time dependant ROC curves.

Predictive accuracy

Cross-validated residual sum of squares used in order to assess how well the model fits or predicts the outcomes can no longer be used with censored data. Censored data are characterised by both status at the end of the follow-up period and the length of follow-up.

Therefore extended predictive accuracy criteria based for continuous data (R²) and binary (ROC curves) may be used in the case of censored data. Schemper and Henderson (2000), build on earlier works that extend R² to Cox model, proposed to characterize the proportion of variation explained by the covariates. Heagerty and Zheng (2003) proposed new time-dependent predictive accuracy summaries based on time specific versions of sensitivity and specificity. This later approach is used in the following where the survival time will be characterized by a counting process representation $N_i(t) = 1(T_i \le t)$. The area under the ROC curves, which measures the probability that a marker value for a randomly selected case exceeds the marker value for a randomly selected control, are particularly useful for comparing the discriminatory capacity of different potential biomarkers. For survival there are several potential extensions of sensitivity and specificity. Heagerty and Zheng (2003) proposed new time-dependent ROC curves based on Incident/Dynamic definition of sensitivity and specificity.

$$\mathit{sensitivity}(c,t):\ P(M_i > c / T_i = t) = P(M_i > c / dN_i = 1)$$

$$\mathit{specificity}(c,t):\ P(M_i \le c / T_i = t) = P(M_i \le c / N_i = 0)$$

with M a predictor score function (M=***Xb***).

Sensitivity measures the expected fraction of subjects with marker greater than *c* among the sub-population of individuals who die at time *t*, while specificity measures the fraction of subjects with a marker less than or equal to *c* among those who survive beyond time *t*.

Using the true and false positive rate functions $TP_t(c) = \mathit{sensitivity}(c,t)$ and $FP_t(c) = 1 - \mathit{specificity}(c,t)$ allows the ROC curve to be written as:

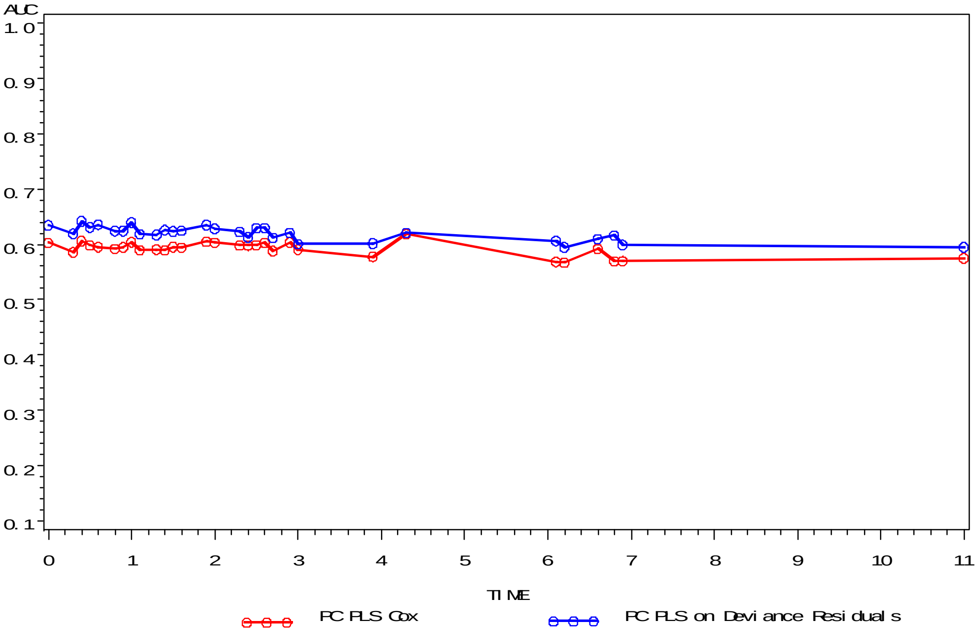
$$ROC_t(p) = TP_t(FP_t)^{-1}(p) \ \text{ with } (FP_t)^{-1}(p) \ = \ \inf_c \left\{ c \ : \ FP_t(c) \le p \right\}$$

$$AUC(t) = \int_0^1 ROC_t(p)dp \ = P\left\{ M_j > M_k / T_j = t, T_k > t \right\}$$

Larger AUC at time *t* based on the risk score function M indicates better predictability of time to event at time t as measured by sensitivity and specificity evaluated at time t.

It is worthy to mention that, time-dependent ROC curves are related to standard Kendall's tau concordance summary (Heagerty and Zheng,2003)

The PLS-Cox model and the PLS regression with deviance residuals as a response, have been compared based on the Rosenwald data. The graph below shows the time-dependant AUC for the test sample, as a criterion to assess the relative predictive performance of the two models.



Both methods show quite similar predictive performance. This confirms the close agreement shown by Segal between the LARS-LASSO Cox procedure of Gui and Li (2005a) and his method based on the deviance residuals.

Threshold Gradient Descent

Friedman and Popescu (2004) showed that PLS regression (or Ridge regression) tends to produce regression coefficients $\hat{\beta}=\left\{\hat{\beta}_1,\ldots,\hat{\beta}_p\right\}$ for which the $\left|\hat{\beta}_j\right|$ of highly correlated variables are shrunk to a common value whereas the LARS-LASSO procedure produces an opposite effect. Let's the true parameter β^* represents a point in the parameter space. The goal of a model selection procedure is to construct a path in the parameter space such that some of the points on the path are close to the parameter β^*. Hence if β^* has highly disparate absolute values, the LARS-LASSO procedure would likely to produce paths in the parameter space that come close to β^*, whereas if the components of β^* have roughly equal absolute values, PLS regression (or Ridge regression) will produce closer paths.

However for situations in between Friedman and Popescu proposed a generalized gradient descent algorithm with a threshold parameter $0 \le \tau \le 1$ that controls dispersion in the absolute coefficient values. Smaller values of τ create paths closer to PLS regression (or Ridge regression), whereas larger values produce paths closer to LARS-LASSO.

Following Friedman and Pospescu, Gui and Li (2005b) proposed to extend the Threshold Gradient Descent to the Cox model. Gui and Li found two major limitations of the LARS-LASSO procedure. First, the number of predictors selected cannot be greater than the sample size, and secondly, the LARS-LASSO procedure tends to select only one variable from a group with high pair-wise correlations which is a major limitation in transcriptomic data when the goal is to select all genes which are most related to survival.

Specifically, for any threshold value $0 \le \tau \le 1$, the threshold gradient descent algorithm for Cox model involves the following steps:

Step 1 $\quad \beta(0) = 0,\ \ \nu = 0$

Step 2 $\quad$ Calculate $\eta,\ \mu,\ g(\nu) = \dfrac{\partial \log PL(\beta)}{\partial \beta}$ for the current β

Step 3 $\quad f_j(\nu) = I\left[\left|g_j(\nu)\right| \ge \tau \max\limits_{0 \le k \le n} \left|g_k(\nu)\right|\right]$ where I is an indicator function.

Step 4 Update $\beta(\nu+\Delta\nu)=\beta(\nu)+\Delta\nu.g(\nu).f(\nu)$

$\nu=\nu+\Delta\nu$

Step 5 repeats steps 2 to 4 until convergence.

With $g(\nu)=X\mu$

Let $\mu=\{\mu_1,...,\mu_n\}$ and $\eta=X\beta$

$$\mu_i=\frac{\partial\log PL(\beta)}{\partial\eta_i}=\delta_i-\exp(\eta_i)\sum_{k\in C_i}\frac{d_k}{\sum_{j\in R_k}\exp(\eta_j)}$$

Where d_k is the number of events at time t_k and $C_k=\{k:i\in R_k\}$ denotes the risk sets containing individual *i*.

Then $g(\nu)=\dfrac{\partial\log PL(\beta)}{\partial\beta}=X\mu$

Gui and Li (2005b) noted that compared to the LARS-LASSO estimate of the Cox model this TGD procedure is computationally fast and does not involve matrix inversion.

Threshold PLS

A similar approach could be used with PLS Cox regression by selecting, in the construction of the *h*th PLS component, only the variables x_i with a significant Wald test in the multivariate Cox regression on x_i *and* $t_1,..,t_{h-1}$. The significant threshold α of the testing procedure could be used to limit the number of selected predictor variables with 1- α acting as τ in the TGD procedure. There are some arguments to motivate the use of such shrinkage procedure when dealing with transcriptomic data. From a biological point of view, one should expect that only a small subset of the genes is relevant to predict survival. Many of the thousands of genes are not informative with regard to the hazard function and contribute only to reduce the predictive performance by introducing noise in the data.

In the same state of mind, Huang et al.(2004) proposed a Penalized PLS regression. They replaced the coefficient b_i of x_i in the computation of the PLS components by $\widehat{b_i} = sign(b_i)\left(\left|b_i\right| - \Delta\right)_+$,with Δ a shrinkage parameter to be determined and $f_+ = f$ if $f > 0$ and $f_+ = 0$ if $f \leq 0$. However it seems to make more sense from a statistical point of view to shrink the coefficients based on their significance rather than on their magnitude.

References

[1] Bastien P., Tenenhaus M., 2001. PLS generalised linear regression, Application to the analysis of life time data. In *PLS and Related Methods, Proceedings of the PLS'01 International Symposium* , CISIA-CERESTA Editeur, Paris, pp. 131-140.

[2] Bastien P., Esposito Vinzi V., and Tenenhaus M., (2005). PLS generalised linear regression, *Computational Statistics & Data Analysis,*48: 17-46

[3] Cox, D.R. 1972. Regression models and life tables. *Journal of the Royal Statistical Society* B. ; 74187-220.

[4] De Jong S., Wise B. and Ricker N., 2001. Canonical partial least squares and continuum power regression. *Journal of. Chemometrics*; 15:85-100.

[5] Efron B., Johnston I., Hastie T., and Tibshirani R., 2004. Least angle regression. *Annals of Statistics*, in press.

[6] Fan X., Felsovalyi A., Sivo S., Keenan S.C. 2002, SAS for Monte Carlo Studies: A guide for quantitative Researchers, SAS publishing, Cary, NY.

[7] Friedman J.H. and Popescu B.E., 2004. Gradient Directed Regularization, Technical report, Statistics department, Standford University.

[8] Garthwaite P.H., 1994. An Interpretation of Partial least Squares. *Journal of the American Statistical Association*, 89(425):122-127.

[9] Gui J. and Li H., 2004. Penalizes Cox regression Analysis in the High-Dimensional and Low-sample Size Settings, with Applications to microarray gene Expression Data. *Center for Bioinformatics & Moleculad Biostatistics*, University of California, San Francisco.

[10] Gui J. and Li H., 2005a. Penalized Cox regression analysis in the high-dimensional and low-sample size settings, with application to microarray gene expression data. *Bioinformatics*; 21 (13): 3001-3008.

[11] Gui J. and Li H., 2005b. Threshold gradient descent method for censored data regression with application in pharmacogenomics, Pacific Symposium Biocomputing 272-83.

[12] Heagerty P. and Zheng Y., 2003. Survival Model Predictive Accuracy and ROC Curves. *UW Biostatistics Working Paper Series*, paper 219, the Berkeley Electronic Press.

[13] Höskuldsson, 1988. A. PLS regression methods. *J. Chemometrics* , 2:211-228.

[14] Huang X. and Pan W. 2003. Linear regression and two-class classification with gene expression data. *Bioinformatics*, 19:2072-2078.

[15] Huang X., Pan W., Park S., han X., Miler L., and Hall J. (2004). Modeling the relationship between LVAD support time and gene expression changes in the human heart by penalized least squares *Bioinformatics*, 20 (6): 888-894.

[16] Kaiser H.F. and Dickman K. 1962. Sample and population score matrices and sample correlation matrices from an arbitrary population correlation matrix. *Psychometrika* 27: 179-182.

[17] Li. and Luan Y., 2003. Kernel Cox regression models for linking gene expression profiles to censored survival data. *Pacific Symposium of Biocomputing*, 8:65-76.

[18] Lewi, P.J. 1995 Pattern recognition, reflection from a chemometric point of view. *Chemometrics and Intelligent laboratory System*, 28, 23-33.

[19] Li H. and Gui J., 2004. Partial Cox regression analysis for high-dimension microarray gene expression data. *Bioinformatics*; 20:208-215.

[20] Lingren F., Geladi P., and Wold S., 1993. The kernel algorithm for PLS. *Journal of. Chemometrics*; 7: 45-59.

[21] Manne R., 1987. Analysis of Two Partial-Least-Squares Algorithms for Multivariate Calibration. *Chemometrics and Intelligent Laboratory System*; 2:187-197.

[22] Nguyen D.V. and Rocke D., 2002. Partial least squares proportional hazard regression for application to DNA microarray survival data. *Bioinformatics*, 18:1625-1632.

[23] Nguyen D.V. and Rocke D., 2004. On partial least squares dimension reduction for microarry-based classification: a simulation study, Comput. Stat. Data Anal. 46: 407.

[24] Nguyen D.V., 2005. Partial least squares dimension reduction for microarray gene expression data with a censored response. *Mathematical Biosciences*, 193:119-137.

[25] Park. P.J., Tian L., and Kohane I.S., 2002. Linking gene expression data with patient survival times using partial least squares. *Bioinformatics*, 18:S120-S127.

[26] Rânnar S., Geladi P., Lingren F. and Wold S., 1994. A PLS kernel Algorithm for data sets with many variables and few objects. Part II : cross-validation, missing data and exemples. *Journal of. Chemometrics*; 9:459-470.

[27] Rosipal, R. and Trejo, L.J., 2001. Kernel Partial Least Squares Regression in Reproducing Kernel Hilbert space. *Journal of machine Learning research* 2, 97-123.

[28] Rosenwald A. *et al.*, 2002. The use of molecular profiling to predict survival for diffuse large-B-cell lymphoma. *The New England Journal of Medicine*, 346:1937-1947.

[29] Segal M. R., 2005. Microarray Gene Expression Data with Linked Survival Phenotypes: Diffuse large-B-Cell Lymphoma Revisited. Technical report, Center for Bioinformatic & Molecular Biostatistics, University of California San Francisco.

[30] Tenenhaus A., Giron A., Viennet E., Béra M., Saporta G., and Fertil B., 2005. Kernel Logistic PLS: a tool for supervised nonlinear dimensionality reduction and binary classification, *Computational Statistics and data Analysis*, on press.

[31] Tenenhaus M., 1998. *La régression PLS.* Technip, Paris.

[32] Tibshirani R., 1997. The lasso method for variable selection in the Cox model. *Statistics in Medicine*, 16:385-395.

[33] Whitehead J., 1980. Fitting Cox's regression Model to Survival Data using Glim. *Appl. Statist*, 29:268-275.

[34] Wold H., 1966. Estimation of principal components and related models by iterative least squares. *In Krishainaah, P.R. (ed), Multivariate Analysis*. New Academic Press, New York , pp. 391-420.

[35] Wold S., Martens H. and Wold H., 1983. The multivariate calibration problem in chemistry solved by the PLS method. *In Proc. Conf. Matrix Pencils, Ruhe A. & Kåstrøm B. (Eds), March 1982, Lecture Notes in Mathematics*, Springer Verlag, Heidelberg , p. 286-293.