High Dimensional Classification with combined Adaptive Sparse PLS and Logistic Regression

Ghislain Durif, Laurent Modolo, Jakob Michaelsson, Jeff E. Mold, Sophie Lambert-Lacroix, Franck Picard

High Dimensional Classification with combined Adaptive Sparse PLS and Logistic Regression

G. Durif [1,2,*], L. Modolo [1,3,4], J. Michaelsson [4], J. E. Mold [4], S. Lambert-Lacroix [5] and F. Picard [1]

[1]LBBE, UMR CNRS 5558, Université Lyon 1, F-69622 Villeurbanne, France,
[2]INRIA Grenoble Alpes, THOTH team, F-38330 Montbonnot, France,
[3]LBMC UMR 5239 CNRS/ENS Lyon, F-69007 Lyon, France,
[4]Department of Cell and Molecular Biology, Karolinska Institutet, Stockholm, Sweden,
[5]UMR 5525 Université Grenoble Alpes/CNRS/TIMC-IMAG, F-38041 Grenoble, France.

*To whom correspondence should be addressed.

Abstract

Motivation: The high dimensionality of genomic data calls for the development of specific classification methodologies, especially to prevent over-optimistic predictions. This challenge can be tackled by compression and variable selection, which combined constitute a powerful framework for classification, as well as data visualization and interpretation. However, current proposed combinations lead to unstable and non convergent methods due to inappropriate computational frameworks. We hereby propose a computationally stable and convergent approach for classification in high dimensional based on sparse Partial Least Squares (sparse PLS).

Results: We start by proposing a new solution for the sparse PLS problem that is based on proximal operators for the case of univariate responses. Then we develop an adaptive version of the sparse PLS for classification, called logit-SPLS, which combines iterative optimization of logistic regression and sparse PLS to ensure computational convergence and stability. Our results are confirmed on synthetic and experimental data. In particular we show how crucial convergence and stability can be when cross-validation is involved for calibration purposes. Using gene expression data we explore the prediction of breast cancer relapse. We also propose a multicategorial version of our method, used to predict cell-types based on single-cell expression data.

Availability: Our approach is implemented in the `plsgenomics` R-package.

Contact: ghislain.durif@inria.fr

Supplementary information: Supplementary materials are available at *Bioinformatics* online.

1 Introduction

Molecular classification is at the core of many recent studies based on Next-Generation Sequencing data. For instance, the genomic characterization of diseases based on genomic signatures has been one *Grail* for many studies to predict patient outcome, survival or relapse (Guedj *et al.*, 2012). Moreover, following the recent advances of sequencing technologies, it is now possible to isolate and sequence the genetic material from a single cell (Stegle *et al.*, 2015). Single-cell data give the opportunity to characterize the genomic diversity between the individual cells of a specific population. However, in both cases, the specific context of high dimensionality constitutes a major challenge for the development of new statistical methodologies (Marimont and Shapiro, 1979; Donoho, 2000). Indeed, the number of recorded variables p (as gene expression) being far larger than the sample size n, classical regression or classification methods are inappropriate (Aggarwal *et al.*, 2001; Hastie *et al.*, 2009), due to spurious dependencies between variables, that lead to singularities in the optimization processes, with neither unique nor stable solution.

This challenge calls for the development of specific statistical tools, such as the following dimension reduction approaches: (i) Compression methods that search for a representation of the data in lower dimensional space and (ii) Variable selection methods, based on a parsimony hypothesis, i.e., among all recorded variables, a lot are supposed to be uninformative and can be considered as noise to be removed from the model. For instance, the Partial Least Squares (PLS) regression (Wold, 1975; Wold *et al.*, 1983) is a compression approach appropriate for linear regression, especially with highly correlated covariates, that constructs new components, i.e. latent directions, explaining the response. An example of sparsity-based approach is the Lasso (Tibshirani, 1996) where coefficients of less relevant variables are shrunk to zero thanks to a ℓ_1

penalty in the optimization procedure. Eventually, sparse PLS (SPLS) regression (Lê Cao *et al.*, 2008; Chun and Keleş, 2010) combines both compression and variable selection to reduce dimension. It introduces a selection step based on the Lasso in the PLS framework, constructing new components as sparse linear combinations of predictors. It occurs as well that combining compression and "sparse" approach improves the efficiency of prediction and the accuracy of selection. Such an association (compression and selection) is also relevant for data visualization, a crucial challenge when considering high dimensional data. Existing SPLS methods are based on resolutions of approximations of the associated optimization problem. In this work, we first propose a new formulation of the sparse PLS optimization problem with a simple exact resolution, derived from proximal operators (Bach *et al.*, 2012). We also introduce an adaptive sparsity-inducing penalty, inspired from the adaptive Lasso (Zou, 2006), to improve the variable selection accuracy.

SPLS has shown excellent performance for regression with a continuous response, but its adaptation to classification is not straightforward. Chung and Keleş (2010) or Lê Cao *et al.* (2011) proposed to use sparse PLS as a preliminary dimension reduction step before a standard classification method, such as discriminant analysis (SPLS-DA) or logistic regression, following previous approaches using classical PLS for molecular classification (Nguyen and Rocke, 2002; Boulesteix, 2004). Their approach gives interesting results in SNPs data analysis (Lê Cao *et al.*, 2011) or in tumor classification (Chung and Keleş, 2010).

Another method for classification consists in using logistic regression (binary or multicategorial) (McCullagh and Nelder, 1989), for which optimization is achieved via the Iteratively Reweighted Least Squares (IRLS) algorithm (Green, 1984). However its convergence is not guaranteed especially in the high dimensional case. Computational convergence is a crucial issue when estimating parameters, as non-convergent methods may lead to unstable and inconsistent estimations, impacting analysis interpretation and reproducibility, especially when tuning hyper-parameters by cross-validation.

The combination of logistic regression and (sparse) PLS could lead to a classification method processing dimension reduction based on lower space representation and variable selection. However, the combination of such iterative algorithms is not necessarily straightforward, due to convergence issues. Performing compression with SPLS on the categorical response as a first step before logistic regression remains counter-intuitive, because SPLS was designed to handle a continuous response within homoskedastic models. Based on the generalized PLS by Marx (1996) or Ding and Gentleman (2005), Chung and Keleş (2010) proposed to use sparse PLS within the IRLS iterations to solve reweighted least squares at each step, however we will see that convergence issues remain. Fort and Lambert-Lacroix (2005) proposed to use a Ridge regularization (Eilers *et al.*, 2001) to ensure the convergence of the IRLS algorithm and to use the classical PLS to estimate predictor coefficients by using a continuous pseudo-response generated by the IRLS algorithm. We will develop a similar approach based on sparse PLS.

Our new SPLS-based approach, called logit-SPLS, combines compression and variable selection in a GLM framework. We show the accuracy, the computational stability and convergence of our method, compared with other state-of-the-art approaches on simulations. Especially, we show that compression increases variable selection accuracy, and that our method is more stable regarding the choice of hyper-parameters by cross-validation, contrary to other methods processing classification with sparse PLS. Thus, our method is the only one that correctly performs considering all criteria (prediction, selection, stability), whereas all the other approaches present a weak spot. Our simulations illustrate the interest of both selection and compression over selection or compression only. Our work was implemented in the existing R-package `plsgenomics`, available on the CRAN.

We will first introduce our adaptive sparse PLS approach. Then, we will develop and discuss our classification framework based on Ridge IRLS and adaptive sparse PLS for logistic regression. We will finish by a comparative study and eventually two applications of our method: (i) binary classification to predict breast cancer relapse after 5 years based on gene expression data, with an illustration of data visualization through compression, (ii) prediction of cell types with multinomial classification based on single-cell expression profiles. To do so, we extend our approach to the multi-group case, based on a "one-class vs a reference" type of multi-classification. One strength of our approach is to propose a sparse PLS that admits a closed-form solution in both binary and multi-group classifications. This leads to computationally efficient procedures in both cases, contrary to sparse PLS-DA approaches for instance, that are based on a multivariate response sparse PLS algorithm in the multi-group case, for which there is no closed-form solution (c.f. Chung and Keleş, 2010; Lê Cao *et al.*, 2011).

2 Compression and selection in the GLM framework

We first define the sparse PLS and introduce a new formulation of the associated optimization problem, based on proximal operator. Contrary to existing approaches, this formulation provides a simple resolution of the covariance maximization problem associated to sparse PLS. Then, we propose an adaptive version of the sparse PLS selection step. Eventually, we will develop our approach to combine sparse PLS and logistic regression.

2.1 Proximal sparse PLS

Let $(\mathbf{x}_i, \xi_i)_{i=1}^n$ be a n-sample, with $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_n)^T \in \mathbb{R}^n$ a continuous response and $\mathbf{x}_i \in \mathbb{R}^p$ a set of p covariates, gathered in the matrix $\mathbf{X}_{n\times p} = [\mathbf{x}_1^T, \ldots, \mathbf{x}_p^T]^T$. The PLS solves a linear regression problem. We consider centered data $\boldsymbol{\xi}_c$ and $\mathbf{X}_c$ to neglect the intercept and the model $\boldsymbol{\xi}_c = \mathbf{X}_c\boldsymbol{\beta}_{\setminus 0} + \boldsymbol{\varepsilon}$, with the coefficients $\boldsymbol{\beta}_{\setminus 0} \in \mathbb{R}^p$. The metric in the observation space $\mathbb{R}^n$ is weighted by the matrix $\mathbf{V}_{n\times n}$.

In the univariate response case, the PLS (Boulesteix and Strimmer, 2007) consists in constructing K components $\mathbf{t}_k \in \mathbb{R}^n$ that explain the response, and defined as linear combinations of predictors, i.e. $\mathbf{t}_k = \mathbf{X}\mathbf{w}_k$ with weight vectors $\mathbf{w}_k \in \mathbb{R}^p$ $(k = 1, \ldots, K)$. These weights $\mathbf{w}_k$ are defined to maximize the empirical covariance of the corresponding components $\mathbf{t}_k$ with the response $\boldsymbol{\xi}_c$. Other PLS algorithms consider the maximization of the squared covariance, however both definitions are equivalent in the univariate response case (De Jong, 1993; Boulesteix and Strimmer, 2007). To exclude the inherent noise induced by non pertinent covariates in the model, the sparse PLS (Lê Cao *et al.*, 2008; Chun and Keleş, 2010) introduces a variable selection step into the PLS framework. It constructs "sparse" weight vectors, whose coordinates are required to be null for covariates that are irrelevant to explain the response. Following the Lasso principle (Tibshirani, 1996), the shrinkage to zero is achieved with a ℓ_1 norm penalty in the covariance maximization problem:

$$\widehat{\mathbf{w}}(\lambda_s) = \underset{\mathbf{w}\in\mathbb{R}^p}{\operatorname{argmin}} \left\{ -\widehat{\operatorname{Cov}}(\mathbf{X}_c\mathbf{w}, \boldsymbol{\xi}_c) + \lambda_s \|\mathbf{w}\|_1 \right\}, \qquad (1)$$

under the constraints $\|\mathbf{w}\|_2 = 1$ and orthogonality between components, with the sparsity parameter $\lambda_s > 0$. The empirical covariance between $\boldsymbol{\xi}_c$ and $\mathbf{t} = \mathbf{X}_c\mathbf{w}$ is explicitly $\widehat{\operatorname{Cov}}(\mathbf{X}_c\mathbf{w}, \boldsymbol{\xi}_c) = \mathbf{w}^T\mathbf{c}$, where $\mathbf{c} = \mathbf{X}_c^T\mathbf{V}\boldsymbol{\xi}_c \in \mathbb{R}^p$ is the empirical covariance $\widehat{\operatorname{Cov}}(\mathbf{X}_c, \boldsymbol{\xi}_c)$, depending on the metric weighted by $\mathbf{V}$ ($\mathbf{c}$ is a vector because the response is univariate).

Different methodologies (Lê Cao *et al.*, 2008; Chun and Keleş, 2010) have been proposed to solve the optimization problem (1). However, both approaches give an approximate solution. We propose a new approach to

exactly solve this problem in the univariate response case. In the standard PLS algorithm, $\mathbf{w}$ is proven to be the dominant singular vector of the empirical covariance $\mathbf{c}$. In the univariate response case (PLS1 algorithm), $\mathbf{c}$ is univariate and $\mathbf{w} \propto \mathbf{c}$. Thus, we introduce the following equivalent formulation of the penalized problem (1):

$$\widehat{\mathbf{w}}(\lambda_s) = \underset{\mathbf{w}\in\mathbb{R}^p}{\operatorname{argmin}} \Big\{ \frac{1}{2}\|\mathbf{c} - \mathbf{w}\|_2^2 + \lambda_s \|\mathbf{w}\|_1 \Big\}, \tag{2}$$

under the constraints $\|\mathbf{w}\|_2 = 1$ and orthogonality between components (the equivalence between (1) and (2) is shown in the Supp. Mat.). We consider a range of values for λ_s so that the problem (2) admits a solution.

Resolution. Applying the method of Lagrange multipliers, the problem (2) becomes ($\mu > 0$):

$$\underset{\substack{\mathbf{w}\in\mathbb{R}^p \\ \mu>0}}{\operatorname{argmin}} \Big\{ \frac{1}{2}\|\mathbf{c} - \mathbf{w}\|_2^2 + \lambda_s \|\mathbf{w}\|_1 + \mu \left(\|\mathbf{w}\|_2^2 - 1\right) \Big\}. \tag{3}$$

The method of Lagrange multipliers was proposed by Witten *et al.* (2009) or Tenenhaus *et al.* (2014) for different decomposition problems. The objective is continuous and convex, thus the strong duality holds and the solutions of primal (2) and dual (3) problems are equivalent. The resolution of the dual problem is based on proximal (or proximity) operators defined as the solution of the following problem (Bach *et al.*, 2012):

$$\underset{\mathbf{w}\in\mathbb{R}^p}{\operatorname{argmin}} \Big\{ \frac{1}{2}\|\mathbf{c} - \mathbf{w}\|_2^2 + f(\mathbf{w}) \Big\}, \tag{4}$$

for any fixed $\mathbf{c} \in \mathbb{R}^p$, any function $f : \mathbb{R}^p \to \mathbb{R}$. It is denoted by $\operatorname{prox}_f(\mathbf{c})$. When $f(\cdot)$ corresponds to the Elastic Net penalty (combination of ℓ_1 and ℓ_2 penalty), i.e. $f(\mathbf{w}) = \frac{\mu}{2}\sum_{j=1}^p |w_j|^2 + \lambda \sum_{j=1}^p |w_j|$ (with $\lambda > 0$ and $\mu > 0$), the closed-form solution of problem (4) is explicitely given by the proximal operator $\operatorname{prox}_{\frac{\mu}{2}\|\cdot\|_2^2 + \lambda\|\cdot\|_1}(\mathbf{c})$ whose coordinates are defined by (Yu, 2013, Theo. 4):

$$\operatorname{prox}_{\frac{\mu}{2}\|\cdot\|_2^2 + \lambda_s\|\cdot\|_1}(\mathbf{c}) = \Big(\frac{1}{1+\mu} \operatorname{sgn}(c_j) \left(|c_j| - \lambda\right)_+ \Big)_{j=1:p}. \tag{5}$$

This corresponds to the normalized soft-thresholding operator applied to the covariance vector $\mathbf{c}$. When choosing $\mu = \mu^*$ so that $\mathbf{w}^* = \operatorname{prox}_{\frac{\mu}{2}\|\cdot\|_2^2 + \lambda\|\cdot\|_1}(\mathbf{c})$ has a unitary norm, the pair $(\mathbf{w}^*, \mu^*)$ given by the proximal operator (5) with $\lambda = \lambda_s$ is a candidate point and then a solution (by convexity) for the dual problem (3). Hence, the SPLS weights used to construct the SPLS components are given by $\mathbf{w}^* \in \mathbb{R}^p$. This new resolution of the sparse PLS problem is a general result, that also applies in the case of the standard homoskedastic linear model by replacing $\mathbf{V}$ by the $n \times n$ identity matrix. This is consistent with the derivation of the sparse PLS by Chun and Keleş (2010), but provides a more direct resolution framework. In addition, λ_s is renormalized to lie in $[0, 1]$ (c.f. Chun and Keleş, 2010).

The resolution of the problem (2) allows to compute $\mathbf{w}_1$ and construct the first components $\mathbf{t}_1$. At step $k > 1$, $\mathbf{w}_k$ is computed by solving Eq. 2, using a "deflated" version of $\mathbf{X}_c$ and $\boldsymbol{\xi}_c$, i.e. the residuals of the respective regression of $\mathbf{X}_c$ and $\boldsymbol{\xi}_c$ onto the previous components $[\mathbf{t}_\ell]_{\ell=1}^{k-1}$, guaranteeing the orthogonality between components. The active set of selected variables up to component K is a subset of $\{1, \ldots, p\}$, defined as the variables with a non null weights in $[\mathbf{w}_k]_{k=1}^K$, and denoted by $\mathcal{A}_K = \cup_{k=1}^K \{j, w_{jk} \neq 0\}$. Eventually, the estimation $\widehat{\boldsymbol{\beta}}_{\setminus 0}^{\text{SPLS}}$ of $\boldsymbol{\beta}_{\setminus 0}$ in the model $\boldsymbol{\xi}_c = \mathbf{X}_c\boldsymbol{\beta}_{\setminus 0} + \boldsymbol{\varepsilon}$ is given by the weighted PLS regression of $\boldsymbol{\xi}_c$ onto the selected variables in the active set $\mathcal{A}_K$. The coefficient $\widehat{\beta}_j^{\text{SPLS}}$ is set to zero if the predictor $j \in \{1, \ldots, p\}$ is not in $\mathcal{A}_K$. Indeed, following the definition of the SPLS regression, the sparse structure of the weight vectors $[\mathbf{w}_k]_{k=1}^K$ directly induces the sparse structure of $\widehat{\boldsymbol{\beta}}_{\setminus 0}^{\text{SPLS}}$. The variables selected to construct the new components $[\mathbf{t}_k]_{k=1}^K$ are the ones that contribute the most to the response and correspond to those with non-null entries in the true vector $\boldsymbol{\beta}_{\setminus 0}$.

2.2 Adaptive sparse PLS

We also propose to adjust the ℓ_1 constraint to further penalize the less significant variables, which can lead to a more accurate selection process. Such an approach is inspired by component wise penalization as adaptive Lasso (Zou, 2006). We use the weights $\mathbf{w}^{\text{PLS}} \in \mathbb{R}^p$ from classical PLS (without sparsity constraint) to adapt the ℓ_1 penalty on the weight vector $\mathbf{w}^{\text{SPLS}}$. The ℓ_1 penalty in problem (2) becomes $\text{Pen}_{\text{ada}}(\mathbf{w}) = \lambda_s \sum_{j=1}^p \gamma^j |w_j|$, with $\gamma^j = 1/|w_j^{\text{PLS}}|$ to account for the significance of the predictor j (higher weights in absolute values correspond to more important variables). The closed-form solution accounts for the adaptive penalty and remains the soft-thresholding operator applied to $\mathbf{c}$ but with parameter $\lambda_s \times \gamma^j$ for j^{th} predictor (c.f. Supp. Mat.). We called this method adaptive sparse PLS.

2.3 Ridge-based logistic regression and logistic regression

We now present our approach based on sparse PLS for logistic regression.

The Logistic Regression model. We now consider a n-sample $(\mathbf{x}_i, y_i)_{i=1}^n$ with y_i being a label variable in $\{0, 1\}$, gathered in $\mathbf{y} = (y_1, \ldots, y_n)^T$. We use the Generalized Linear Models (GLM) framework (McCullagh and Nelder, 1989) to relate the predictors to the random response variable Y_i, using the logistic link function, such that $\text{logit}(\pi_i) = \beta_0 + \mathbf{x}_i^T\boldsymbol{\beta}_{\setminus 0}$, with $\pi_i = \mathbb{E}[Y_i]$, $\text{logit}(x) = \log(x/(1-x))$, and $\boldsymbol{\beta} = (\beta_0, \beta_1 \ldots, \beta_p)^T = \{\beta_0, \boldsymbol{\beta}_{\setminus 0}\}$. In the sequel, $\mathbf{Z} = [(1, \ldots, 1)^T, \mathbf{X}]$. With $\eta_i = \mathbf{z}_i^T\boldsymbol{\beta}$, the log-likelihood of the model is defined by $\log \mathcal{L}(\boldsymbol{\beta}) = \sum_{i=1}^n [y_i\eta_i - \log\{1 + \exp(\eta_i)\}]$, and the coefficients $\boldsymbol{\beta} \in \mathbb{R}^{p+1}$ are estimated by maximum likelihood (MLE).

The Ridge IRLS algorithm. Optimization relies on a Newton-Raphson iterative procedure (McCullagh and Nelder, 1989) to construct a sequence $(\widehat{\boldsymbol{\beta}}^{(t)})_{t\geq 1}$, whose limit $\widehat{\boldsymbol{\beta}}^{\infty} \in \mathbb{R}^{p+1}$ (if it exists) is the estimation of $\boldsymbol{\beta}$. The Iteratively Reweighted Least Squares (IRLS) algorithm (Green, 1984) explicitly defines $(\widehat{\boldsymbol{\beta}}^{(t)})_{t\geq 1}$ as the solutions of successive weighted least squares regressions of a pseudo-response $\boldsymbol{\xi}^{(t)} \in \mathbb{R}^n$ onto the predictors at each iteration t. The pseudo-response is linearly generated from the predictors based on previous iterations, c.f. Eq (6). However, when $p > n$, the matrix $\mathbf{Z}$ is singular, which leads to optimization issues. Le Cessie and Van Houwelingen (1992) proposed to optimize a Ridge penalized log-likelihood, i.e. $\log \mathcal{L}(\boldsymbol{\beta}) - (\lambda_R/2)\ \boldsymbol{\beta}^T\widehat{\boldsymbol{\Sigma}}\boldsymbol{\beta}$, with $\widehat{\boldsymbol{\Sigma}}$ the diagonal empirical variance matrix of $\mathbf{Z}$ and $\lambda_R > 0$ the Ridge parameter. A unique solution of this regularized problem always exists and is computed by the Ridge IRLS (RIRLS) algorithm (Eilers *et al.*, 2001), where the weighted regression at each IRLS iteration is replaced by a Ridge weighted regression, hence:

$$\begin{cases} \widehat{\boldsymbol{\beta}}^{(t+1)} = (\mathbf{Z}^T\mathbf{V}^{(t)}\mathbf{Z} + \lambda_R\,\widehat{\boldsymbol{\Sigma}})^{-1}\mathbf{Z}^T\mathbf{V}^{(t)}\boldsymbol{\xi}^{(t)}, \\ \boldsymbol{\xi}^{(t+1)} = \mathbf{Z}\widehat{\boldsymbol{\beta}}^{(t)} + \left(\mathbf{V}^{(t)}\right)^{-1}\left[\mathbf{y} - \boldsymbol{\pi}^{(t)}\right], \end{cases} \tag{6}$$

with the estimated probabilities $\widehat{\boldsymbol{\pi}}^{(t)} = (\widehat{\pi}_i^{(t)})_{i=1}^n$, i.e. $\widehat{\pi}_i^{(t)} = \text{logit}^{-1}\big(\mathbf{z}_i^T\widehat{\boldsymbol{\beta}}^{(t)}\big)$ for each Y_i, and $\mathbf{V}^{(t)} = \text{diag}\big(\widehat{\pi}_i^{(t)}(1 - \widehat{\pi}_i^{(t)})\big)_{i=1}^n$ is the diagonal empirical variance matrix of $(Y_i)_{i=1}^n$ at step t.

Following the definition of $\boldsymbol{\xi}^{(t)}$, the (R)IRLS algorithm produces a pseudo-response $\boldsymbol{\xi}^{\infty}$ as the limit of the sequence $(\boldsymbol{\xi}^{(t)})_{t\geq 1}$, verifying $\boldsymbol{\xi}^{\infty} = \mathbf{Z}\widehat{\boldsymbol{\beta}}^{\infty} + \boldsymbol{\varepsilon}$, where $\widehat{\boldsymbol{\beta}}^{\infty}$ is the solution of the likelihood optimization, and $\boldsymbol{\varepsilon}$ is a noise vector of covariance matrix $(\mathbf{V}^{\infty})^{-1}$, with $\mathbf{V}^{\infty}$ the limit of the matrix sequence $(\mathbf{V}^{(t)})_{t\geq 1}$.

Sparse PLS regression. The pseudo-response $\boldsymbol{\xi}^\infty$ produced by Ridge IRLS depends on predictors through a linear model. Following the approach by Fort and Lambert-Lacroix (2005), we propose to use the sparse PLS regression on $\boldsymbol{\xi}^\infty$ to process dimension reduction and estimate $\boldsymbol{\beta} \in \mathbb{R}^{p+1}$ in the logistic model $\mathbb{E}[Y_i] = \text{logit}^{-1}(\beta_0 + \mathbf{x}_i^T \boldsymbol{\beta}_{\setminus 0})$. In this case, the ℓ_2 metric (in the observation space) is weighted by the empirical inverse covariance matrix $\mathbf{V}^\infty$, to account for the heteroskedasticity of noise $\boldsymbol{\varepsilon}$. To neglect the intercept in the SPLS step, we consider the centered version of $\mathbf{X}$ and $\boldsymbol{\xi}^\infty$, regarding the metric weighted by $\mathbf{V}^\infty$, denoted by $\mathbf{X}_c$ and $\boldsymbol{\xi}_c^\infty$. The intercept β_0 will be estimated later.

The estimates $\widehat{\boldsymbol{\beta}}_{\setminus 0}^{\text{SPLS}} \in \mathbb{R}^p$ are renormalized to correspond to the non-centered and non-scaled data, i.e. $\widehat{\boldsymbol{\beta}}_{\setminus 0} = \widehat{\boldsymbol{\Sigma}}^{-1/2} \widehat{\boldsymbol{\beta}}_{\setminus 0}^{\text{SPLS}}$ giving the estimation $\widehat{\boldsymbol{\beta}}_{\setminus 0}$ in the original logistic model . The intercept β_0 is estimated by $\widehat{\beta}_0 = \bar{\xi}^\infty - \bar{\mathbf{x}}^T \widehat{\boldsymbol{\beta}}_{\setminus 0}$ where $\bar{\xi}^\infty$ and $\bar{\mathbf{x}}$ are respectively the sample average of the pseudo-response and the sample average vector of predictors regarding the metric weighted by $\mathbf{V}^\infty$. Our method can be summarized as follow:

1. $(\boldsymbol{\xi}^\infty, \mathbf{V}^\infty) \leftarrow \text{RIRLS}(\mathbf{X}, \mathbf{y}, \lambda_R)$
2. Center $\mathbf{X}$ and $\boldsymbol{\xi}^\infty$ regarding the scalar product weighted by $\mathbf{V}^\infty$
3. $\left(\widehat{\boldsymbol{\beta}}_{\setminus 0}^{\text{SPLS}}, \mathcal{A}_K, [\mathbf{t}_k]_{k=1}^K\right) \leftarrow \text{SPLS}(\mathbf{X}, \boldsymbol{\xi}^\infty, K, \lambda_s, \mathbf{V}^\infty)$
4. Renormalization of $\widehat{\boldsymbol{\beta}} = \{\widehat{\beta}_0, \widehat{\boldsymbol{\beta}}_{\setminus 0}\}$

The label $\widehat{y}_{\text{new}}$ of new observations $\mathbf{x}_{\text{new}} \in \mathbb{R}^p$ (non-centered and non-scaled) is predicted through the logit function thanks to the estimates $\widehat{\boldsymbol{\beta}} = \{\widehat{\beta}_0, \widehat{\boldsymbol{\beta}}_{\setminus 0}\}$. Note that $\mathbf{x}_{\text{new}}$ does not need to be centered nor scaled thanks to the intercept parameter $\widehat{\beta}_0$ and to the renormalization of the coefficient estimates in the algorithm.

Our method estimates predictor coefficients $\boldsymbol{\beta}$ in the logistic model by sparse PLS regression of a pseudo-response, considered as continuous and therefore in accordance with the conceptual framework of PLS, while completing compression and variable selection simultaneously. An additional interest is that the iterative optimization in the RIRLS algorithm does not depend on the number of components K nor on the sparsity parameter λ_s. Consequently, the convergence of our method is robust to the choice of K and λ_s by definition, contrary to other approaches for logistic regression based on sparse PLS (c.f. Supp. Mat. section A.3). Our approach will be called logit-SPLS in the following while the method by Fort and Lambert-Lacroix (2005) will be called logit-PLS.

2.4 Tuning sparsity by stability selection

Our logit-SPLS approach depends on three hyper-parameters: the sparsity parameter λ_s, the Ridge parameter λ_R and the number of components K. We first propose to tune all the parameters by 10-fold cross-validation (to reduce the sampling dependence). Details about the choice of the grid of candidates values for $(\lambda_s, \lambda_R, K)$ are given in Supp. Mat. (c.f. sections A.5.1 and A.6.1).

In addition, we propose to adapt the stability selection method developed by Meinshausen and Bühlmann (2010), to the sparse PLS framework. The interest of this approach is to avoid choosing a value for the sparsity parameter λ_s to find the degree of the sparsity in the model, i.e. to select the relevant predictors. In this framework, the grid of all parameter candidate values for $(\lambda_s, \lambda_R, K)$ is denoted by Λ. The principle consists in fitting the model for all points $\boldsymbol{\ell} \in \Lambda$, then estimating the probability $p_j^{\boldsymbol{\ell}}$ for each covariate j to be selected over 100 resamplings of size $n/2$ depending on $\boldsymbol{\ell}$, i.e. the probability for predictor j to be in the set $\widehat{S}_{\boldsymbol{\ell}} = \{j,\ \widehat{\beta}_j(\boldsymbol{\ell}) \neq 0\}$, where $\widehat{\boldsymbol{\beta}}(\boldsymbol{\ell}) \in \mathbb{R}^p$ are the corresponding estimated coefficients. Finally, the procedure retains the predictors that are in the set $\widehat{S}_{\text{stable}}$ of stable selected variables, defined as $\{j,\ \max_{\boldsymbol{\ell} \in \Lambda}\{p_j^{\boldsymbol{\ell}}\} \geq \pi_{\text{thr}}\}$, where π_{thr} is a threshold value. This means that predictors with high selection probability are kept and predictors with low selection probability are discarded.

The average number of selected variables over the entire grid Λ, is denoted by q_Λ, and defined as $q_\Lambda = \mathbb{E}[\#\{\cup_{\lambda \in \Lambda} \widehat{S}_\lambda\}]$. Meinshausen and Bühlmann (2010, Theo. 1) provided a bound on the expected number of wrongly stable selected variables (equivalent to false positives) in $\widehat{S}_{\text{stable}}$, depending on the threshold π_{thr}, the expectation q_Λ and the number p of covariates:

$$\mathbb{E}[\text{FP}] \leq \frac{1}{2\pi_{\text{thr}} - 1} \frac{q_\Lambda^2}{p} \tag{7}$$

where FP is the number of false positives i.e. $\text{FP} = \#\{S_0^c \cap \widehat{S}_{\text{stable}}\}$ and S_0 the unknown set of true relevant variables. This results is derived under some reasonable conditions that are discussed in Supp. Mat. (section A.2). Following the recommendation of Meinshausen and Bühlmann (2010, p. 424), we use Eq. 7 to determine the range of the parameter grid Λ to avoid too many false positives (corresponding to a weak ℓ_1 penalization). Indeed, since the number of false positives is controlled by q_Λ, we automatically exclude candidate points $\boldsymbol{\ell} = (\lambda_s, \lambda_R, K)$ corresponding to small λ_s (near 0) for which there is no selection and for which all variables contribute to the mode, so that we can control q_Λ. Without removing these points, q_Λ and the number of false positives are too high. For instance, when the threshold probability π_{thr} is set to 0.9, Λ is defined as a subset of the parameter grid so that $q_\Lambda = \sqrt{0.8\, p\, \rho_{\text{error}}}$. In practice, q_Λ is unknown but can be estimated by the empirical average number of selected variables over all $\boldsymbol{\ell} \in \Lambda$. In this context, the expected number of false positives will be lower than ρ_{error} (in practice, we set $\rho_{\text{error}} = 10$). Details about the candidate values for $(\lambda_s, \lambda_R, K)$ are given in Supp. Mat. (section A.6.2).

A clear interest here is that we do not have to choose a specific value for the hyper-parameters, instead we retain the variables that are selected by most of the models when exploring the grid of candidate values for hyper-parameters (including K).

3 Simulation study

We assess the performance of our approach for prediction, compression and variable selection compared to state-of-the-art methods that were previously introduced. We also use a "baseline" method, called GLMNET (Friedman *et al.*, 2010), that performs variable selection, by solving the GLM likelihood maximization with ℓ_1 norm penalty for selection and ℓ_2 norm penalty for regularization, also known as the Elastic Net approach (Zou and Hastie, 2005). We compare different approaches based on (sparse) PLS for classification (c.f. Tab. 1 and Supp. Mat. section A.3 and A.4 for details).

Simulation design. Our simulated data are constructed to assess the interest of compression and variable selection for prediction performance. The simulations are inspired from Zou *et al.* (2006), Shen and Huang (2008) or Chung and Keleş (2010). The purpose is to control the redundancy within predictors, and the relevance of each predictor to explain the response. We consider a predictor matrix $\mathbf{X}$ of dimension $n \times p$, with $n = 100$ fixed, and $p = 100, 500, 1000, 2000$, so that we examine different high dimensional models. The true vector coefficients $\boldsymbol{\beta}^*$ is generated to be sparse, the sparsity structure is thus known. Hence, it is possible to assess whether a method selects the relevant predictors or not. The response variable Y_i for observation i is a Bernoulli variable, with parameter $\pi_i^* = \text{logit}^{-1}(\mathbf{x}_i^T \boldsymbol{\beta}^*)$. The pattern of data simulation and the tuning of hyper-parameters are detailed in Supp. Mat. (section A.5). Regarding other methods, we use the range of parameters recommended by their respective authors and the cross-validation procedures supplied in the corresponding packages.

Table 1. The different algorithms to process dimension reduction by (sparse) PLS in the framework of the logistic regression.

Method	Algorithm	Sparse?	Reference
GPLS	(S)PLS inside the IRLS algorithm	×	Ding and Gentleman (2005)
SGPLS		✓	Chung and Keleş (2010)
PLS-log	(S)PLS before logistic regression	×	Wang *et al.* (1999), Nguyen and Rocke (2002)
SPLS-log		✓	Chung and Keleş (2010)
logit-PLS	(S)PLS on the pseudo-response after the RIRLS algorithm	×	Fort and Lambert-Lacroix (2005)
logit-SPLS		✓	**Our algorithm**

Ridge penalty ensures convergence. Convergence is crucial when combining PLS and IRLS algorithm as pointed by Fort and Lambert-Lacroix (2005). With the analysis of high dimensional data and the use of selection in the estimating process, it becomes even more essential to ensure the convergence of the optimization algorithms, otherwise the output estimates may not be relevant. Our simulations show that the Ridge regularization systematically ensures the convergence of the IRLS algorithm in our method (logit-SPLS), for any configuration of simulation: $p = n, p > n$, high or low sparsity, high or low redundancy (see Tabs. A.3 and A.2 in Supp. Mat.). On the contrary, approaches that use (sparse) PLS before or within the IRLS algorithm (resp. SPLS-log and (S)GPLS) encounter severe convergence issues.

Whereas the SPLS-log or (S)GPLS approaches were designed to overcome convergence issues, it appears that they do not, which questions the reliability of the results supplied by these methods. Then, it confirms the interest of the Ridge regularization to ensure the convergence of the IRLS algorithm. Moreover, this convergence seems to be fast (around 15 iterations even when $p = 2000$), which depicts an interesting outcome for computational time. For instance, the tuning of three parameters in the logit-SPLS approach is less costly thanks to the fast convergence of the algorithm. Although both SGPLS and SPLS-log methods are based on two parameters, they iterates further (until the limit set by the user) which is less computationally efficient, especially with high dimensional data. On this matter, details regarding computation times are given in Supp. Mat. (section A.5.2).

Adaptive selection improves cross-validation stability. A cross-validation procedure would be expected to be stable under multiple runs, i.e. the chosen values must not be variable when running the procedure many times on the same sample. Otherwise, selection and prediction become uncertain and not suitable for experiment reproducibility. We quantified the standard deviation of the sparse parameter λ_s chosen by cross-validation for the three sparse PLS methods (SGPLS, SPLS-log and our logit-SPLS) when repeating the procedure on the same samples. The standard deviation (all three methods consider the same range of values for λ_s) is smaller for our approach (c.f. Tab. 2) than for other methods. Thus, the cross-validation procedure in our adaptive method is more stable than other SPLS approaches. A similar comment can be made regarding the choice of the number K of components (c.f. Fig. A.1 in Supp. Mat.). This behavior can be linked to the convergence of the different approaches. The methods with convergence issues (SGPLS and SPLS-log) present a higher cross-validation instability, whereas our method (logit-SPLS) converges efficiently and shows a better cross-validation stability. Similarly, the variable selection accuracy, defined as the proportion of rightly selected and rightly non selected variables (Chong and Jun, 2005), is also influenced by the cross-validation stability and the convergence of the method. Indeed, the standard deviation of the selection accuracy (computed across multiple runs) is higher for the less stable and less convergent methods (SGPLS and SPLS-log) compared to our logit-SPLS approach (c.f. Tab. 2).

Table 2. Comparing computational stability between sparse PLS approaches. $\widehat{\sigma}(\widehat{\lambda}_s)$ stands for the estimated standard deviation of the tuned hyper-parameter λ_s (over repetitions on the same simulated data set), which measures the stability of the hyper-parameter tuning by cross-validation. $\widehat{\sigma}(\text{acc.})$ stands for the estimated standard deviation of the accuracy in variable selection, which measures the stability of the selection steps. The results are presented for different model dimensions (p).

Method	$p = 100$		$p = 2000$	
	$\widehat{\sigma}(\widehat{\lambda}_s)$	$\widehat{\sigma}(\text{acc.})$	$\widehat{\sigma}(\widehat{\lambda}_s)$	$\widehat{\sigma}(\text{acc.})$
logit-spls	**0.09**	**0.11**	**0.11**	**0.09**
sgpls	0.17	0.14	0.15	0.12
spls-log	0.23	0.12	0.21	0.17

Compression and selection increase prediction accuracy. We now assess the importance of compression and variable selection for prediction performance. We consider the prediction accuracy, evaluated through the prediction error rate. A first interesting point is that the prediction performance of compression methods is improved by the addition of a selection step: logit-SPLS, SGPLS and SPLS-DA perform better than logit-PLS, GPLS and PLS-DA respectively (c.f. Tab. 3). In addition, sparse PLS approaches also present a lower classification error rate than the GLMNET method that performs variable selection only. These two points support our claim that in any case compression and selection should be both considered for prediction. Similar results are observed for other configurations of simulated data (c.f. Supp. Mat. section A.5.2). All different SPLS-based approaches show similar prediction performance, even methods that are not converging (SPLS-log or SGPLS) compared to our adaptive approach logit-SPLS. Thus, checking prediction accuracy only may not be a sufficient criterion to assess the relevance of a method. The GPLS method is a good example of non-convergent method (c.f. Tab. 3 and Tab. A.2 in Supp. Mat.) that presents high variability and poor performance regarding prediction.

Actually, the combination of Ridge IRLS and sparse PLS in our method ensures convergence and provides good prediction performance (prediction error rate at 10% on average) even in the most difficult configurations $n = 100$ and $p = 2000$, which makes it an appropriate framework for classification.

Compression increases selection accuracy. A sparse model will be useful if characterized by good prediction performances but also if the selected covariates are the genuine important predictors that explain the response. To assess the selection accuracy, we compare the selected predictors returned by each sparse method to the set of relevant ones used to construct the response, i.e. with a non zero coefficient β_j^* in our simulation model. We consider sensitivity and specificity (Chong and Jun, 2005), respectively

Table 3. Prediction error and selection sensitivity/specificity (if relevant) when $p = 2000$, for non-sparse or sparse approaches (delimited by the line). Results for other values of p are joined in Supp. Mat. (section A.5.2).

Method	Prediction error	Selection sensitivity	Selection specificity	Selection accuracy
gpls	0.49 ± 0.31	/	/	/
pls-da	0.20 ± 0.07	/	/	/
logit-pls	0.17 ± 0.07	/	/	/
glmnet	0.16 ± 0.07	0.27	0.98	0.74
logit-spls	$\mathbf{0.11 \pm 0.06}$	**0.63**	**0.86**	**0.79**
sgpls	0.11 ± 0.05	0.80	0.75	0.81
spls-da	0.12 ± 0.06	0.82	0.74	0.81
spls-log	0.12 ± 0.05	0.83	0.75	0.81

the proportion of true positive and true negative regarding the selected variables.

A first striking point is that, in our simulations (see Tab. 3 and Tabs. A.4, A.5, A.6 in Supp. Mat.), the baseline GLMNET presents a very low sensitivity and a very high specificity (low true positive and low false positive rates), meaning that it selects a small number of predictors (that are relevant), which leads to a lower accuracy compared to SPLS-based approaches. Thus, using approaches that combine compression and variable selection such as sparse PLS has a true impact on selection accuracy, compared to "selection-only" approach such as GLMNET.

Then, we focus on the comparison of the different sparse PLS approaches. On the one hand, our method logit-SPLS selects less irrelevant predictors since the false positive rate is lower (higher specificity), compared to other SPLS approaches. On the other hand, SGPLS, SPLS-log and SPLS-DA select more true positives (higher sensitivity). Since all methods achieve a similar level of accuracy, this result clearly illustrates a difference of strategy regarding variable selection. The balance between sensitivity and specificity indicates that our method logit-SPLS selects predictors which are more likely to be relevant, discarding most of the non-pertinent predictors, while other approaches tend to select more predictors with higher false positive rate. With high dimensional data set (large p), we are generally interested in highly sparse model, thus it is an advantage to have a sharper control on the false positive rate, as in our method. In addition, the relative good sensitivity of other sparse PLS approaches (SGPLS and SPLS-log) is also balanced by a selection process that is less stable than ours, as the standard deviation of the accuracy is higher over simulations (as previously mentioned, see Tab. 2).

4 Classification of breast tumors using adaptive sparse PLS for logistic regression

We consider a publicly available data set on breast cancer (Guedj *et al.*, 2012) containing the level expression of 54613 genes for 294 patients affected by breast cancer. We focus on the relapse after 5 years, considering a $\{0, 1\}$ valued response, if the relapse occurred or not. There were 214 patients without relapse and 80 with a relapse. We reduce the number of genes by considering the top 5000 most differentially expressed genes, by using a standard t-test with a Benjamini-Hochberg correction. Computation details (resamplings, cross-validation, stability selection, training and test set definition) are joined in Supp. Mat. (c.f. section A.6).

Convergence and stability with Ridge IRLS and adaptive sparse PLS. The Ridge IRLS algorithm confirms its usual convergence (see Tab. 4). Other approaches based on SPLS (SGPLS and SPLS-log) again encounter severe issues and almost never converge. Following a similar pattern, our adaptive selection is far more stable under the tuning of the sparsity parameter λ_s by cross-validation than any other approach using sparse PLS (Tab.4), as the precision on this hyper-parameter value is the highest for our method, illustrating less variability in the tuning over repetitions.

Table 4. Averaged prediction error, convergence percentage over 100 resamplings and standard deviation of cross-validated λ_s.

Method	Prediction error	Conv. perc.	s.d. λ_s
glmnet	0.27 ± 0.04	/	/
logit-pls	0.26 ± 0.05	100%	/
logit-spls	0.23 ± 0.06	100%	0.15
logit-spls-ad	$\mathbf{0.19 \pm 0.04}$	**100%**	**0.15**
sgpls	0.5 ± 0.21	5%	0.18
spls-log	0.18 ± 0.04	1%	0.19

Interest of adaptive selection for prediction and selection. Regarding prediction performance, the adaptive version of our algorithm logit-SPLS gives better results (c.f. Tab. 4) which highlights the interest of adaptive selection. It can also be noted that our approach performs better on prediction than both logit-PLS (compression only) and GLMNET (selection only), which again supports the interest of using both compression and variable selection. The SGPLS method does not confirm its performance on our simulatiosn with poor and highly variable results, illustrating the potential lack of stability of non-convergent method. Only the SPLS-log method achieves a classification that is as good as our adaptive method. However this point will be counterbalanced by its assessment over the other criteria in the following.

Regarding variable selection, the stability selection analysis (see Fig. 1) shows that, when the number of false positives is bounded (on average), our approach logit-SPLS selects more genes than any other approach (SGPLS, SPLS-log and GLMNET). Hence, we discover more true positives (because the number of false positives is bounded), unraveling more relevant genes than other approaches. This again illustrates the good performance of our method for selection. More generally, approaches that use sparse PLS, i.e. performing selection and compression, select more variables than GLMNET with the same false positive rate, thus retrieving more true positives than GLMNET which performs only selection. This again supports our previously developed idea that compression and selection are both very suitable for high dimensional data analysis. We recall that the curves in Fig. 1 correspond to the number of variables that are selected by most models when exploring the grid of candidate values for hyper-parameters (including K). Additional results regarding the overlap between the genes selected by the different methods and the list of selected genes with their score (i.e. the maximum estimated probability of selection) are given in Supp. Mat. (section A.6.2).

Efficient compression to discriminate the response. To assess the interest of our approach for data visualization, we represent the score of the observations on the first two components, i.e. the point cloud $(t_{i1}, t_{i2})_{i=1}^n$. The points are colored according to their Y-labels. An efficient compression technique would separate the Y-classes with fewer components. We fit the different compression-based approach (when the number of components is set to $K = 2$). We use PCA as a reference for compression and data visualization, based on unsupervised learning contrary to other compared approaches. Fig. 2 represents the first two components computed by logit-PLS, logit-SPLS, SGPLS, SPLS-log and PCA. It appears that the first two components from our logit-SPLS are sufficient to easily separate the two Y-classes. On the contrary, other sparse PLS approaches do not achieve a similar efficiency in the

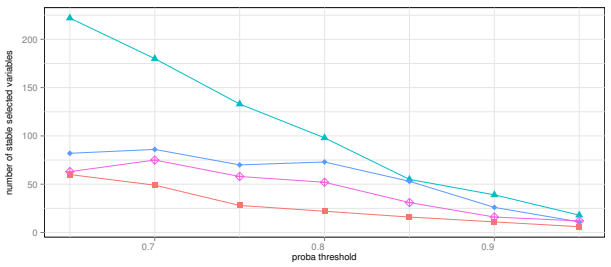

Fig. 1. Number of variables in the set of stable selected variables versus the threshold π_{thr}, when forcing the average number of false positives to be smaller than $\rho_{\text{error}} = 10$. Methods: glmnet ($-\blacksquare-$), logit-spls-adapt ($-\blacktriangle-$), sgpls ($-\blacklozenge-$), spls-log ($-\oplus-$). Note: here, all hyper-parameters (including K) vary across the grid of candidate values Λ (c.f. Supp. Mat. section A.6.2).

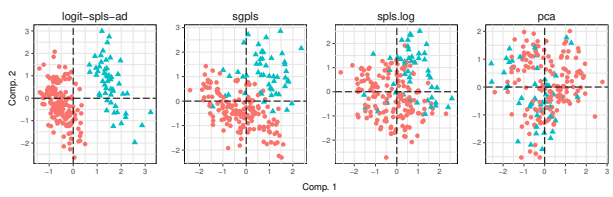

Fig. 2. Observations scores on the first two components for the different methods. The points are shaped according to the value of the response: 0 (●) and 1 (▲). The scores are normalized for comparison.

compression process. Thus, our method turns out to be very efficient for data visualization, especially compared to principal component analysis.

5 Characterization of T lymphocyte types based on single-cell data

We generalized our approach to the multicategorial case and developed a new method, called multinomial-SPLS (or MSPLS), that was applied to the prediction of cell types using single cell expression data (Stegle *et al.*, 2015; Gawad *et al.*, 2016). Our approach (detailed in Supp. Mat., section A.7) is based on a direct extension of the logistic model. It is specifically a "one-class vs a reference" type of multi-classification, in which the membership probabilities of each class (except the reference) are estimated based on linear combinations of the predictors. The membership probability of the reference class is then deduced from the rest. The resolution is derived from our logit-SPLS method. One interest is that our multi-group classification approach uses a univariate response sparse PLS algorithm (that admits a closed-form solution, c.f. section 2), contrary to sparse multigroup PLS-DA for instance (c.f. Supp. Mat. section A.3).

Understanding the mechanisms of an adaptive immune response is of great interest for the creation of new vaccines. This response is made possible thanks to antigen-specific "effector" T cells capable of recognizing and killing infected cells, and to the long-lasting "memory" T cells that will constitute a repertoire for later secondary immune responses. These two types of T cells have then been described as 4 sub-groups: CM, TSCM ("Memory"), TEMRA, EM, ("Effector Memory"). Generally speaking, CM and TSCM can be considered as "Memory" cells and TEMRA and EM can be considered as "Effectors" as CM/TSCM and EM/TEMRA share significant functional overlap with each other (Willinger *et al.*, 2005a; Gattinoni *et al.*, 2011). Understanding the transcriptomic diversity of T cells constitutes a new challenge to better characterize the short and long-term vaccinal responses, as T cells are increasingly recognized as being highly heterogeneous populations (Newell *et al.*, 2012). However, these investigations have been limited by current practices that consist in defining those 4 cell types based by drawing non-overlapping gates on the 2D-space defined two surface markers only: CCR7 and CD45RA (Sallusto *et al.*, 1999). Consequently this rule leads to the selection of a fraction of cells that only correspond to cells with the most extreme values of markers, which ignores the complexity of a T cell population sampled from real blood.

We developed a SPLS-based multi-categorical classification to better characterize the transcriptomic diversity that supports the 4 different cell types of T cells. This approach aims at classifying more cells, and at inferring the type of the non-identified cells. To do so, we considered the measurements of 11 surface markers (CCR7, CD45RA, CD27, IL7R, FAS, CD49F, PD1, CD57, CD3E, CD8A), along with the expression of the corresponding genes. All these measurements were available on the single-cell basis. We will show that even in this low dimensional case, the use of variable selection will help to improve the accuracy of the results. In the following, hyper-paramaters (including K) were tuned by cross-validation. Details about the candidate values for $(\lambda_s, \lambda_R, K)$ are given in Supp. Mat. (section A.8.1).

We developed the following two-step analysis. We started by considering the measures of the 11 surface markers and the expression of the 11 associated genes. The multinomial-SPLS was trained on a subset of cells that were tagged manually, and used to predict the types of the unknown cells (136 annotated over 943 cells). On this training set of 136 cells, including 44 CM and 28 TSCM cells (i.e. 72 "Memory" cells), 30 EM and 34 TEMRA cells (i.e. 64 "Effector" cells), a 5-fold cross-validation procedure (with 50 repetitions) is used to tune the hyper-parameters. The cross-validation prediction error over the resamplings was $\sim 6\%$.Fig. A.3 in Supp. Mat. shows that the cells in the training set are well discriminated in this first step. In addition, our SPLS procedure selected the proteins CCR7 and CD45RA in 100% of the runs, which is coherent with the manual annotation of the cells based on these two markers.

In a second step we wanted to enrich the set of genes that discriminate cell types. To proceed we considered the expression of all genes of these predicted cell types, and performed a differential analysis from which we retained 61 differentially expressed genes (corresponding to a 5% FDR). By considering these 61 genes added to the first 22 markers considered for the first prediction step, we performed the MSPLS-based prediction on the complete data set annotated by our first prediction. Our method selected 8 new biologically relevant genes (more details in Supp. Mat. section A.8.2) with a cross-validation prediction error rate over re-samplings (again 5-fold cross-validation) of $\sim 16\%$ (on the whole data set, not only considering the most extreme phenotypes). The main interests of this two-step procedure were to be computationally efficient and to narrow the list of potential genes of interest, which was conclusive since this second prediction greatly improved the biological relevance of the predicted cell types by accounting for more information than the one contained in the classical markers like CCR7 and provided us with new insight to better understand the T cells immune response.

Fig. 3 illustrates the representation of the cells in the latent dimensional space computed by the multinomial PLS in the second step of prediction. The reference class is "CM". The SPLS computes latent directions discriminating each other class ("EM", "TEMRA" and "TSCM" respectively) versus the reference class (c.f. Supp Mat.). The cells are represented on the first two components for the three different pairs: "CM versus EM", "CM versus TEMRA" and "CM versus TSCM". The latent components clearly discriminate the group of cells in the three different cases, which confirms the result of the second prediction based both on markers and differentially expressed genes. The different groups are clearly identified but there is no gap between them, contrary to the representation

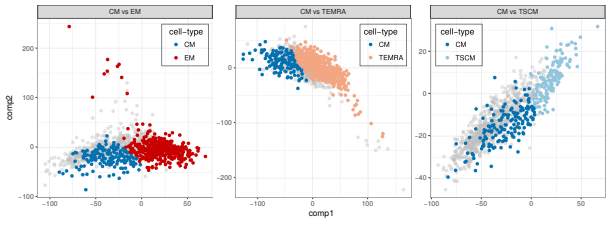

Fig. 3. Cell scores on the first two PLS components in the latent space that discriminate between the reference class ("CM") and each other class separately ("EM", "TEMRA" and "TSCM" respectively, from left to right). T cells are identified by their predicted types after the second prediction step.

of the cells in the training set for the the first prediction (c.f. Supp. Mat.). This indicates that the multinomial-SPLS was able to predict the type of the lost common cells based on the most extreme phenotypes.

This application highlights the interest of dimension reduction by compression and variable selection, even when dealing with low dimensional data. It can also be noted that, even when using sparse approaches, a step of pre-selection is always useful, especially in the analysis of single-cell expression data, which are very noisy compared to standard RNA-seq data, because of the important inter-cellular diversity.

6 Conclusion

We have introduced a new formulation of sparse PLS and proposed an adaptive version of our algorithm to improve the selection process. Using proximal operators, we provide an explicit resolution framework with a closed-form solution based on soft-thresholding operators.

In addition, we developed a method that performs compression and variable selection suitable for classification. It combines Ridge regularized Iterative Least Square algorithm and sparse PLS in the logistic regression context. It is particularly appropriate for the case of high dimensional data, which appears to be a crucial issue nowadays, for instance in genomics. Our main consideration was to ensure the convergence of IRLS algorithm, which is a critical point in logistic regression. Another concern was to properly incorporate into the GLM framework a dimension reduction approach such as sparse PLS.

Ridge regularization ensures the convergence of the IRLS algorithm, which is confirmed in our simulations and tests on experimental data sets. Applying adaptive sparse PLS as a second step on the pseudo-response produced by IRLS respects the definition of PLS regression for continuous response. Moreover, the combination of compression and variable selection increases the prediction performance and selection accuracy of our method, which turns out to be more efficient than state-of-the-art approaches that do not use both dimension reduction techniques. Such a combination also improves the compression process, illustrated by the efficiency of our method for data visualization compared to standard supervised or unsupervised approaches. Furthermore it appears that previous procedures using sparse PLS with logistic regression encounter convergence issues linked to a lack of stability inthe cross-validation parameter tuning process, highlighting the crucial importance of convergence when dealing with iterative algorithms.

It can be noted that our approach can be used to include additional covariates in the model. For example, we used a combination of surface marker levels and gene expression levels in the single cell data analysis. On this matter, an interesting research direction would be to work on a Least Square-Partial Least Square (LS-PLS) approach, in which some part of the predictors are compressed into PLS components and some others are not. There have been recent advances regarding LS-PLS for logistic regression (see Bazzoli and Lambert-Lacroix, 2016). However, to our knowledge, there is no work on a potential LS-SPLS method, even in the regression case.

In addition, an interesting extension of our work would be to investigate theoretical properties of the sparse PLS regression (especially regarding its consistency or any oracle properties). Deriving such properties would be an opportunity to assess the underlying statistical properties of our method and remains an open question.

Funding

This work was supported by the french National Resarch Agency (ANR) as part of the "Algorithmics, Bioinformatics and Statistics for Next Generation Sequencing data analysis" (ABS4NGS) ANR project [grant number ANR-11-BINF-0001-06] and as part of the "MACARON" ANR project [grant number ANR-14-CE23-0003]. It was performed using the computing facilities of the computing center LBBE/PRABI.

References

Aggarwal, C. C., Hinneburg, A., and Keim, D. A. (2001). On the surprising behavior of distance metrics in high dimensional space. In *International Conference on Database Theory*, pages 420–434. Springer.

Bach, F., Jenatton, R., Mairal, J., and Obozinski, G. (2012). Optimization with Sparsity-Inducing Penalties. *Found. Trends Mach. Learn.*, **4**(1), 1–106.

Barker, M. and Rayens, W. (2003). Partial least squares for discrimination. *Journal of chemometrics*, **17**(3), 166–173.

Bazzoli, C. and Lambert-Lacroix, S. (2016). Classification using LS-PLS with logistic regression based on both clinical and gene expression variables. *Preprint*.

Boulesteix, A.-L. (2004). PLS dimension reduction for classification with microarray data. *Statistical applications in genetics and molecular biology*, **3**(1), 1–30.

Boulesteix, A.-L. and Strimmer, K. (2007). Partial least squares: A versatile tool for the analysis of high-dimensional genomic data. *Briefings in bioinformatics*, **8**(1), 32–44.

Chong, I.-G. and Jun, C.-H. (2005). Performance of some variable selection methods when multicollinearity is present. *Chemometrics and Intelligent Laboratory Systems*, **78**(1), 103–112.

Chun, H. and Keleş, S. (2010). Sparse partial least squares regression for simultaneous dimension reduction and variable selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **72**(1), 3–25.

Chung, D. and Keleş, S. (2010). Sparse Partial Least Squares Classification for High Dimensional Data. *Statistical Applications in Genetics and Molecular Biology*, **9**(1).

De Jong, S. (1993). SIMPLS: An alternative approach to partial least squares regression. *Chemometrics and intelligent laboratory systems*, **18**(3), 251–263.

Ding, B. and Gentleman, R. (2005). Classification Using Generalized Partial Least Squares. *Journal of Computational and Graphical Statistics*, **14**(2), 280–298.

Donoho, D. L. (2000). High-dimensional data analysis: The curses and blessings of dimensionality. *AMS Math Challenges Lecture*, pages 1–32.

Eilers, P. H., Boer, J. M., van Ommen, G.-J., and van Houwelingen, H. C. (2001). Classification of microarray data with penalized logistic regression. In *BiOS 2001 The International Symposium on Biomedical Optics*, pages 187–198. International Society for Optics and Photonics.

Eksioglu, E. M. (2011). Sparsity regularised recursive least squares adaptive filtering. *IET signal processing*, **5**(5), 480–487.

Firth, D. (1993). Bias reduction of maximum likelihood estimates. *Biometrika*, **80**(1), 27–38.

Fort, G. and Lambert-Lacroix, S. (2005). Classification using partial least squares with penalized logistic regression. *Bioinformatics*, **21**(7), 1104–1111.

Fort, G., Lambert-Lacroix, S., and Peyre, J. (2005). Réduction de dimension dans les modèles linéaires généralisés: application à la classification supervisée de données issues des biopuces (in french). *Journal de la société française de statistique*, **146**(1-2), 117–152.

Friedman, J., Hastie, T., and Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of statistical software*, **33**(1), 1.

Gattinoni, L., Lugli, E., Ji, Y., Pos, Z., Paulos, C. M., Quigley, M. F., Almeida, J. R., Gostick, E., Yu, Z., Carpenito, C., Wang, E., Douek, D. C., Price, D. A., June, C. H., Marincola, F. M., Roederer, M., and Restifo, N. P. (2011). A human memory T cell subset with stem cell-like properties. *Nat. Med.*, **17**(10), 1290–1297.

Gawad, C., Koh, W., and Quake, S. R. (2016). Single-cell genome sequencing: Current state of the science. *Nature Reviews Genetics*, **17**(3), 175–188.

Green, P. J. (1984). Iteratively reweighted least squares for maximum likelihood estimation, and some robust and resistant alternatives. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 149–192.

Guedj, M., Marisa, L., de Reynies, A., Orsetti, B., Schiappa, R., Bibeau, F., MacGrogan, G., Lerebours, F., Finetti, P., Longy, M., Bertheau, P., Bertrand, F., Bonnet, F., Martin, A. L., Feugeas, J. P., Bièche, I., Lehmann-Che, J., Lidereau, R., Birnbaum, D., Bertucci, F., de Thé, H., and Theillet, C. (2012). A refined molecular taxonomy of breast cancer. *Oncogene*, **31**(9), 1196–1206.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York, New York, NY, second edition.

Lê Cao, K.-A., Rossouw, D., Robert-Granié, C., and Besse, P. (2008). A sparse PLS for variable selection when integrating omics data. *Statistical applications in genetics and molecular biology*, **7**(1).

Lê Cao, K.-A., Boitard, S., and Besse, P. (2011). Sparse PLS discriminant analysis: Biologically relevant feature selection and graphical displays for multiclass problems. *BMC Bioinformatics*, **12**, 253.

Le Cessie, S. and Van Houwelingen, J. C. (1992). Ridge estimators in logistic regression. *Applied statistics*, pages 191–201.

Marimont, R. B. and Shapiro, M. B. (1979). Nearest Neighbour Searches and the Curse of Dimensionality. *IMA Journal of Applied Mathematics*, **24**(1), 59–70.

Marx, B. D. (1996). Iteratively reweighted partial least squares estimation for generalized linear regression. *Technometrics*, **38**(4), 374–381.

McCullagh, P. and Nelder, J. A. (1989). *Generalized Linear Models, Second Edition*. CRC Press.

Meinshausen, N. and Bühlmann, P. (2010). Stability selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **72**(4), 417–473.

Newell, E. W., Sigal, N., Bendall, S. C., Nolan, G. P., and Davis, M. M. (2012). Cytometry by time-of-flight shows combinatorial cytokine expression and virus-specific cell niches within a continuum of CD8+ T cell phenotypes. *Immunity*, **36**(1), 142–152.

Nguyen, D. V. and Rocke, D. M. (2002). Tumor classification by partial least squares using microarray gene expression data. *Bioinformatics*, **18**(1), 39–50.

Sallusto, F., Lenig, D., Forster, R., Lipp, M., and Lanzavecchia, A. (1999). Two subsets of memory T lymphocytes with distinct homing potentials and effector functions. *Nature*, **401**(6754), 708–712.

Shen, H. and Huang, J. Z. (2008). Sparse principal component analysis via regularized low rank matrix approximation. *Journal of multivariate analysis*, **99**(6), 1015–1034.

Stegle, O., Teichmann, S. A., and Marioni, J. C. (2015). Computational and analytical challenges in single-cell transcriptomics. *Nature Reviews Genetics*, **16**(3), 133–145.

Tenenhaus, A., Philippe, C., Guillemot, V., Le Cao, K.-A., Grill, J., and Frouin, V. (2014). Variable selection for generalized canonical correlation analysis. *Biostatistics*, **15**(3), 569–583.

Tibshirani, R. (1996). Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, **58**(1), 267–288.

Wang, C.-Y., Chen, C.-T., Chiang, C.-P., Young, S.-T., Chow, S.-N., and Chiang, H. K. (1999). A Probability-based Multivariate Statistical Algorithm for Autofluorescence Spectroscopic Identification of Oral Carcinogenesis. *Photochemistry and photobiology*, **69**(4), 471–477.

Wherry, E. J., Ha, S.-J., Kaech, S. M., Haining, W. N., Sarkar, S., Kalia, V., Subramaniam, S., Blattman, J. N., Barber, D. L., and Ahmed, R. (2007). Molecular Signature of CD8+ T Cell Exhaustion during Chronic Viral Infection. *Immunity*, **27**(4), 670–684.

Willinger, T., Freeman, T., Hasegawa, H., McMichael, A. J., and Callan, M. F. (2005a). Molecular signatures distinguish human central memory from effector memory CD8 T cell subsets. *J. Immunol.*, **175**(9), 5895–5903.

Willinger, T., Freeman, T., Hasegawa, H., McMichael, A. J., and Callan, M. F. C. (2005b). Molecular Signatures Distinguish Human Central Memory from Effector Memory CD8 T Cell Subsets. *The Journal of Immunology*, **175**(9), 5895–5903.

Witten, D. M., Tibshirani, R., and Hastie, T. (2009). A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis. *Biostatistics*, **10**(3), 515–534.

Wold, H. (1975). Soft Modeling by Latent Variables; the Nonlinear Iterative Partial Least Squares Approach. *Perspectives in Probability and Statistics. Papers in Honour of M. S. Bartlett*.

Wold, S., Martens, H., and Wold, H. (1983). The multivariate calibration problem in chemistry solved by the PLS method. In *Matrix Pencils*, pages 286–293. Springer.

Yu, Y.-L. (2013). On decomposing the proximal map. In *Advances in Neural Information Processing Systems*, pages 91–99.

Zou, H. (2006). The adaptive lasso and its oracle properties. *Journal of the American statistical association*, **101**(476), 1418–1429.

Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **67**(2), 301–320.

Zou, H., Hastie, T., and Tibshirani, R. (2006). Sparse principal component analysis. *Journal of computational and graphical statistics*, **15**(2), 265–286.

Supplementary Information

A.1 Optimization in sparse PLS

A.1.1 Reformulation of the sparse PLS problem

As previously introduced, the sparse PLS constructs components as sparse linear combination of the covariates. When considering the first components, i.e. $\mathbf{t}_1 = \mathbf{X}\mathbf{w}_1$, the weight vector $\mathbf{w}_1 \in \mathbb{R}^p$ is defined to maximize the empirical covariance between the component and the response, i.e. $\widehat{\text{Cov}}(\mathbf{X}\mathbf{w}, \boldsymbol{\xi}) \propto \mathbf{w}^T\mathbf{X}_c^T\boldsymbol{\xi}_c$ (centered $\mathbf{X}$ and $\boldsymbol{\xi}$) with a penalty on the ℓ_1-norm of $\mathbf{w}_1$ to enforce sparsity in the weights. Thus, the weight vector $\mathbf{w}_1$ is computed as the solution of the following optimization problem:

$$\begin{cases} \underset{\mathbf{w}\in\mathbb{R}^p}{\text{argmin}} \Big\{ -\mathbf{w}^T\mathbf{X}_c^T\boldsymbol{\xi}_c + \lambda_s \sum_j |w_j| \Big\}, \\ \|\mathbf{w}\|_2 = 1 \text{ (additional constraint)}, \end{cases} \tag{A.1}$$

with $\lambda_s > 0$. The problem (A.1) is equivalent to the following, when denoting the standard scalar product by $\langle \cdot, \cdot \rangle$:

$$\begin{cases} \underset{\mathbf{w}\in\mathbb{R}^p}{\text{argmin}} \Big\{ -2\langle \mathbf{w}\,,\, \mathbf{X}_c^T\boldsymbol{\xi}_c \rangle + \|\mathbf{w}\|_2^2 + 2\lambda_s \sum_j |w_j| \Big\}, \\ \|\mathbf{w}\|_2 = 1\,, \end{cases}$$

because the term $\|\mathbf{w}\|_2$ is constant thanks to the additional constraint. This new problem remains equivalent to the following:

$$\begin{cases} \underset{\mathbf{w}\in\mathbb{R}^p}{\text{argmin}} \Big\{ \|\mathbf{X}_c^T\boldsymbol{\xi}_c\|_2^2 - 2\langle \mathbf{w}\,,\, \mathbf{X}_c^T\boldsymbol{\xi}_c \rangle + \|\mathbf{w}\|_2^2 + 2\lambda_s \sum_j |w_j| \Big\}, \\ \|\mathbf{w}\|_2 = 1\,, \end{cases}$$

since the norm of the empirical covariance $\|\mathbf{X}_c^T\boldsymbol{\xi}_c\|_2^2$ is constant. Then, thanks to the Euclidean norm properties, it can be rewritten as:

$$\begin{cases} \underset{\mathbf{w}\in\mathbb{R}^p}{\text{argmin}} \Big\{ \frac{1}{2}\|\mathbf{c} - \mathbf{w}\|_2^2 + \lambda_s \|\mathbf{w}\|_1 \Big\}, \\ \|\mathbf{w}\|_2 = 1\,, \end{cases} \tag{A.2}$$

with $\mathbf{c} = \mathbf{X}_c^T\boldsymbol{\xi}_c$ and when setting $\lambda = 2\nu > 0$. Actually, in the case of a univariate response, the formulation (A.2) is natural. Indeed, in the standard (non-sparse) PLS, the optimal weight vector $\mathbf{w}$ is the normalized dominant singular vector of the covariance matrix $\mathbf{X}^T\boldsymbol{\xi}$. However, when the response is univariate, the matrix $\mathbf{X}^T\boldsymbol{\xi}$ is a vector and the solution for $\mathbf{w}$ is the normalized vector $\mathbf{X}^T\boldsymbol{\xi}$ (normalized to 1). This corresponds exactly to the solution of the problem:

$$\begin{cases} \underset{\mathbf{w}\in\mathbb{R}^p}{\text{argmin}} \|\mathbf{c} - \mathbf{w}\|_2^2\,, \\ \|\mathbf{w}\|_2 = 1\,, \end{cases}$$

(without the ℓ_1 penalty).

The solution of the penalized problem (A.3) defines the first component ($k = 1$) of the sparse PLS. We use deflated predictors and response to construct the following component ($k > 1$).

A.1.2 Resolution of the sparse PLS problem

Applying the method of Lagrange multipliers, the problem (A.2) becomes:

$$\underset{\substack{\mathbf{w}\in\mathbb{R}^p \\ \mu>0}}{\text{argmin}} \Big\{ \frac{1}{2}\|\mathbf{c} - \mathbf{w}\|_2^2 + \lambda_s \|\mathbf{w}\|_1 + \mu\left(\|\mathbf{w}\|_2^2 - 1\right) \Big\}, \tag{A.3}$$

with $\mu > 0$. The objective is continuous and convex, thus the strong duality holds and the solutions of primal and dual problems are equivalent.

To solve the problem A.3, we use proximity operator (also called proximal operator) defined as the solution of the following problem (Bach *et al.*, 2012):

$$\underset{\mathbf{w}\in\mathbb{R}^p}{\text{argmin}} \Big\{ \frac{1}{2}\|\mathbf{c} - \mathbf{w}\|_2^2 + f(\mathbf{w}) \Big\}, \tag{A.4}$$

for any fixed $\mathbf{c} \in \mathbb{R}^p$, any function $f : \mathbb{R}^p \to \mathbb{R}$. It is denoted by $\text{prox}_f(\mathbf{c})$. When $f(\cdot)$ corresponds to the Elastic Net penalty (combination of ℓ_1 and ℓ_2 penalty), i.e. when considering the problem (with $\lambda > 0$ and $\mu > 0$):

$$\underset{\mathbf{w}\in\mathbb{R}^p}{\text{argmin}} \Big\{ \frac{1}{2}\|\mathbf{c} - \mathbf{w}\|_2^2 + \frac{\mu}{2}\sum_{j=1}^p (w_j)^2 + \lambda \sum_{j=1}^p |w_j| \Big\}, \tag{A.5}$$

the closed-form solution is explicitly given by the proximal operator $\text{prox}_{\frac{\mu}{2}\|\cdot\|_2^2 + \lambda\|\cdot\|_1}$ that is in particular the composition of $\text{prox}_{\frac{\mu}{2}\|\cdot\|_2^2}$ and $\text{prox}_{\lambda\|\cdot\|_1}$ (Yu, 2013, Theo. 4), i.e.

$$\text{prox}_{\frac{\mu}{2}\|\cdot\|_2^2 + \lambda\|\cdot\|_1}(\mathbf{c}) = \text{prox}_{\frac{\mu}{2}\|\cdot\|_2^2} \circ \text{prox}_{\lambda\|\cdot\|_1}(\mathbf{c})\,.$$

Both proximal operators $\text{prox}_{\lambda\|\cdot\|_1}$ and $\text{prox}_{\frac{\mu}{2}\|\cdot\|_2^2}$ are known (Bach *et al.*, 2012), respectively being:

$$\text{prox}_{\lambda\|\cdot\|_1}(\mathbf{c}) = \Big(\text{sgn}(c_j)\left(|c_j| - \lambda\right)_+\Big)_{j=1:p},$$

$$\text{prox}_{\frac{\mu}{2}\|\cdot\|_2^2}(\mathbf{c}) = \frac{1}{1+\mu}\,\mathbf{c},$$

where $\text{sgn}(\cdot)\left(|\cdot| - \lambda\right)_+$ is the soft-thresholding operator. Eventually, the coordinates of the solution are then:

$$\text{prox}_{\frac{\mu}{2}\|\cdot\|_2^2 + \lambda\|\cdot\|_1}(\mathbf{c}) = \Big(\frac{1}{1+\mu}\,\text{sgn}(c_j)\left(|c_j| - \lambda\right)_+\Big)_{j=1:p}, \tag{A.6}$$

which correspond to the normalized soft-thresholding operator applied to the vector $\mathbf{c} = \mathbf{X}_c^T\boldsymbol{\xi}_c$.

We use the solution (A.6) of the Elastic Net problem (A.5), where $\lambda = \lambda_s$ and μ is chosen so that the solution has a unitary norm, to find a candidate point and then the solution (by convexity) of the dual problem (A.3).

Finally, we have reformulated the problem defining the sparse PLS as a least squares problem with an Elastic Net penalty and we have shown that the solution of this problem is the (normalized) soft-thresholding operator.

A.1.3 Adaptive penalty

When considering an adaptive penalty, the optimization problem associated to the sparse PLS can be similarly rewritten as:

$$\begin{cases} \underset{\mathbf{w}\in\mathbb{R}^p}{\text{argmin}} \Big\{ \frac{1}{2}\|\mathbf{c} - \mathbf{w}\|_2^2 + \sum_{j=1}^p \lambda_j |w_j| \Big\}, \\ \|\mathbf{w}\|_2 = 1\,, \end{cases} \tag{A.7}$$

with the penalty constant $\lambda_j = \lambda\gamma^j$ (c.f. main text). By a similar reasoning (continuity and convexity), it is possible to use Lagrange multiplier to resolve the problem (A.7).

In order to explicitly derive the solution, we will use the proximal operator that is solution of the following problem:

$$\underset{\mathbf{w}\in\mathbb{R}^p}{\text{argmin}} \Big\{ \frac{1}{2}\|\mathbf{c} - \mathbf{w}\|_2^2 + \frac{\mu}{2}\sum_{j=1}^p (w_j)^2 + \sum_{j=1}^p \lambda_j |w_j| \Big\}, \tag{A.8}$$

with $\mu > 0$.

If $f_1(\mathbf{w}) = \sum_j \lambda_j |w_j|$, it can be shown that the solution of the problem (A.4) when considering $f = f_1$ is given by:

$$\text{prox}_{f_1}(\mathbf{c}) = \Big(\text{sgn}(c_j)\left(|c_j| - \lambda_j\right)_+\Big)_{j=1:p},$$

because the subgradient of f_1 is given by $\nabla^s f_1(\mathbf{w}) = \left(\lambda_j \,\text{sgn}(w_j)\right)_{j=1}^p$ (Eksioglu, 2011).

The link between subgradient and proximal operator is described in Bach *et al.* (2012). In particular, $\mathbf{w}^* = \text{prox}_f(\mathbf{c})$ if and only if $\mathbf{c} - \mathbf{w}^* \in \partial f(\mathbf{w}^*)$ for any couple $(\mathbf{c}, \mathbf{w}^*) \in \mathbb{R}^p \times \mathbb{R}^p$, where $\partial f(\mathbf{w}^*)$ is the subdifferential of f at point $\mathbf{w}^*$, i.e. the set of all subgradients $\nabla^s f(\mathbf{w}^*)$ of f at point $\mathbf{w}^*$. If f is differentiable in $\mathbf{w}^*$, then the only subgradient is the gradient $\nabla f(\mathbf{w}^*)$.

The proximal operator corresponding to the function $f_2(\mathbf{w}) = \frac{\mu}{2}\sum_j (w_j)^2$ is known (c.f. previously):

$$\text{prox}_{f_2}(\mathbf{c}) = \frac{1}{1+\mu}\,\mathbf{c}.$$

Eventually, thanks to theorem 4 in Yu (2013), the solution of problem (A.8) is explicitly defined as the combination of prox_{f_1} and prox_{f_2}:

$$\text{prox}_{f_1+f_2}(\mathbf{c}) = \text{prox}_{f_2} \circ \text{prox}_{f_1}(\mathbf{c})\,.$$

for any $\mathbf{c} \in \mathbb{R}^p$. Thus, the solution of the problem (A.8) is given by:

$$\text{prox}_{f_1+f_2}(\mathbf{c}) = \Big(\frac{1}{1+\mu}\,\text{sgn}(c_j)\left(|c_j| - \lambda_j\right)_+\Big)_{j=1:p},$$

with $\mathbf{c} = \mathbf{X}_c^T \boldsymbol{\xi}_c$.

We choose μ so that the norm of the solution is unitary to find a candidate point and thus the solution (by convexity) of the adaptive problem (A.7).

A.2 Conditions for stability selection

The result by Meinshausen and Bühlmann (2010) regarding the expected number of wrongly selected variables is derived for $\Lambda \subset \mathbb{R}^+$ under two conditions: (i) assuming that the indicators $\left(\mathbf{1}_{\{j \in \widehat{S}_\ell\}}\right)_{j \in S_0^c}$ are exchangeable for any $\ell \in \Lambda$. (ii) The original procedure of selection is not worst than random guessing. The first assumptions assumes that the considered method does not "prefer" to select some covariates rather than some other in the set of the non-pertinent predictors. This hypothesis seems reasonable in our SPLS framework. The second one is verified according to the results on our simulations (c.f. section 3). Moreover, in the method we consider, the grid of hyper-parameters lies in $(\mathbb{R}^+)^3$, however the parameter that truly influences the sparsity of the estimation is the parameter $\lambda_s \in \mathbb{R}^+$. Therefore, the sparse PLS appears to be a reasonable framework to apply the concept of stability selection.

A.3 Comparison with state-of-the-art approaches

In the literature, other methodologies have been proposed to adapt (sparse) PLS for binary classification. We detail here different approaches based on (sparse) PLS and GLMs, especially regarding the potential issues raised by the combination of two optimization frameworks.

PLS and GLMs. To overcome the convergence issue in the IRLS algorithm, Marx (1996) proposed to solve the weighted least square problem at each IRLS step with a PLS regression, i.e. $\widehat{\boldsymbol{\beta}}^{(t+1)}$ is computed by weighted PLS regression of the pseudo-response $\boldsymbol{\xi}^{(t)}$ onto the predictors $\mathbf{X}$. However, such iterative scheme does not correspond to the optimization of an objective function. Hence, the convergence of the procedure cannot be guaranteed and the potential solution is not clearly defined.

Alternatively, Wang *et al.* (1999) and Nguyen and Rocke (2002) proposed to achieve the dimension reduction before the logistic regression. Their algorithm use the PLS regression as a preliminary compression step. The components $[\mathbf{t}_k]_{k=1}^K$ in the subspace of dimension K are then used in the logistic regression instead of the predictors. Therefore, the IRLS algorithm does not deal with high dimensional data (as $K < p$). In this context, the PLS algorithm treats the discrete response as continuous. Such approach seems counter-intuitive as it neglects the definition of PLS to resolve a linear regression problem and it ignores the inherent heteroskedastic context. This algorithm is called PLS-log in the following. It can be noted that Nguyen and Rocke (2002) or Boulesteix (2004) also proposed to use discriminant analysis as a classifier after the PLS step. This method, known as PLS-DA, is not directly linked to the GLM framework but we cite it as an alternative for classification with PLS-based approaches. It can be noted that Barker and Rayens (2003) proposed a slightly different implementation of PLS-DA, which is however equivalent to Boulesteix's approach in the binary response case, since they both rely on equivalent univariate response PLS algorithms (De Jong, 1993; Boulesteix and Strimmer, 2007).

Then, Ding and Gentleman (2005) proposed the GPLS method. They introduced a modification in Marx's algorithm based on the Firth procedure (Firth, 1993), in order to avoid the non-convergence and the potential infinite parameter estimation in logistic regression. However, this approach is also characterized by the absence of an explicit optimization criterion. Eventually, as introduced previously, Fort and Lambert-Lacroix (2005) proposed to integrate the dimension reduction PLS step after a Ridge regularized IRLS algorithm. We presented the adaptation of such methodology in the context of sparse PLS in the previous section.

Sparse PLS and GLMs. More recently, based on the SPLS algorithm by Chun and Keleş (2010), Chung and Keleş (2010) presented two different approaches. The first one, called SGPLS, is a direct extension from the GPLS algorithm by Ding and Gentleman (2005). It solves the successive weighted least square problems of IRLS using a sparse PLS regression, with the idea that variable selection reduces the model complexity and helps to overwhelm numerical singularities. Unfortunately, our simulations will show that convergence issues remain. Indeed, the use of SPLS does not resolve the issue link to the absence of an associated optimization problem. The second approach is a generalization of the PLS-log algorithm and uses sparse PLS to reduce the dimension before running the logistic regression on the SPLS components. This method will be called SPLS-log.In both case, i.e. in SGPLS and SPLS-log , the iterative optimization in the IRLS algorithm or modified IRLS algorithm does depend on the number K of components and on the sparsity parameter λ_s. Thus, the convergence of the algorithm is potentially affected by the choice of the hyper-parameters.

Eventually, we cite the SPLS-DA method developed by (Chung and Keleş, 2010) or Lê Cao *et al.* (2011). Generalizing the approach from Boulesteix (2004), they used sparse PLS as a preliminary dimension reduction step before a discriminant analysis. In the binary response case, thanks to the equivalence between Boulesteix (2004) and Barker and Rayens (2003) works, the sparse extension of Barker and Rayens' PLS-DA for binary classification corresponds to the work of Chung and Keleş (2010) or Lê Cao *et al.* (2011). A disadvantage of sparse PLS-DA approaches is that, in the multi-group classification case, they both rely on multivariate response sparse PLS algorithms, which do not admit a closed-form solution. On the contrary, our approach uses a univariate response sparse PLS algorithm (which admits a closed-form solution, c.f. main text) in both binary and multi-group classifications, being computationally efficient in both cases.

A.4 Performance evaluation

In order to assess the performance of our method, we compare it to other state-of-the-art approaches taking into account sparsity and/or performing compression. We eventually use a "reference" method, called GLMNET (Friedman *et al.*, 2010), that performs variable selection, by solving the GLM likelihood maximization penalized by ℓ_1 norm penalty for selection and ℓ_2 norm penalty for regularization, also known as the Elastic Net approach (Zou and Hastie, 2005). Computations were performed using the software environment for statistics `R`. The GPLS approach used in our computation comes from the archive of the former `R`-package `gpls`, the methods logit-PLS and PLS-DA from the package `plsgenomics`, SGPLS, SPLS-log and SPLS-DA from the package `spls`, GLMNET from the package `glmnet`.

A.5 Complements on the simulation study

A.5.1 Simulation design

We consider a predictor matrix $\mathbf{X}$ of dimension $n \times p$, with $n = 100$ fixed, and $p = 100, 500, 1000, 2000$, so that we examine low and high dimensional models. To simulate redundancy within predictors, $\mathbf{X}$ is partitioned into k^* blocks (10 or 50 in practice) denoted by $\mathcal{G}_k$ for block k. Then for each predictor $j \in \mathcal{G}_k$, X_{ij} is generated depending on a latent variable H_k as $X_{ij} = H_{ik} + F_{ij}$, with $H_{ik} \sim \mathcal{N}(0, \sigma_H^2)$ and some noise $F_{ij} \sim \mathcal{N}(0, \sigma_F^2)$. The correlation between the blocks is regulated by σ_H^2, the higher σ_H^2 the less dependency. In the following we consider $\sigma_H / \sigma_F = 2$ or 1/3.

The true vector of predictor coefficients $\boldsymbol{\beta}^*$ is structured according to the blocks $\mathcal{G}_k$ in $\mathbf{X}$. Actually, ℓ^* blocks in $\boldsymbol{\beta}^*$ are randomly chosen among the k^* ones to be associated with non null coefficients (with $\ell^* = 1$ or $k^*/2$). All coefficients within the ℓ^* designated blocks are constant (with value 1/2). In our model, the relevant predictors contributing to the response will be those with non zero coefficient, and our purpose will be to retrieve them via selection. The response variable Y_i for observation i is sampled as a Bernoulli variable, with parameter π_i^* that follows a logistic model: $\pi_i^* = \text{logit}^{-1}(\mathbf{x}_i^T \boldsymbol{\beta}^*)$.

For our method, the parameter values that are tuned by cross-validation are the following: the number of components K varies from 1 to 10, candidate values for the Ridge parameter λ_R in RIRLS are 31 points that are $\log_{10}$-linearly spaced in the range $[10^{-2}; 10^3]$, candidate values for the sparse parameter λ_s are 10 points that are linearly spaced in the range $[0.05; 0.95]$. Other SPLS approaches (SGPLS and SPLS-log) only depend on hyper-parameters (λ_s, K) for which candidate values are the same as for our method. Regarding GLMNET, we let the procedure chooses by itself the grid of hyper-parameters, as recommended by the authors in the documentation.

A.5.2 Additional simulation results

Convergence. Tab. A.3 summarized the convergence of the different methods (logit-SPLS, SGPLS and SPLS-log) during the cross-validation procedure (including the tuning of K) on the simulations, depending on the number of predictors p. Our approach logit-SPLS always converges on contrary to other SPLS approaches for logistic regression.

Tab. A.2 summarized the convergence of the different methods on the simulations, when fitting the model after tuning the hyper-parameters (including K) by cross-validation, depending on the number p of predictors. Again, our approach logit-SPLS always converges on contrary to other SPLS approaches for logistic regression.

In addition, Tab. A.3 shows the percentage of convergence for the different SPLS approaches across cross-validation repeated runs. We see a similar pattern as in Tab. A.2, only our method logit-SPLS almost certainly converge.

Table A.1. Averaged percentage of model that converged during cross-validation tuning of hyper-parameters for different values of p.

Method	$p = 100$	$p = 500$	$p = 1000$	$p = 2000$
sgpls	37	34	33	33
spls-log	44	67	71	74
logit-spls	**100**	**100**	**100**	**100**

Table A.2. Averaged percentage of model fitting that converged over 75 simulations for different values of p. Hyper-parameters are tuned by cross-validation.

Method	$p = 100$	$p = 500$	$p = 1000$	$p = 2000$
gpls	66	59	61	56
sgpls	33	23	23	23
spls-log	84	52	39	32
logit-spls	**100**	**100**	**100**	**100**

Table A.3. Averaged precentage of runs that converged across repeated cross-validations (tuning of all hyper-parameters, including K).

Method	$p = 100$	$p = 500$	$p = 1000$	$p = 2000$
sgpls	37	34	33	33
spls-log	44	67	71	74
logit-spls	**100**	**100**	**100**	**100**

Cross-validation stability. Fig. A.1 illustrates the stability of the cross-validation procedure for the different SPLS approaches regarding the number of components. Our approach logit-SPLS always chooses $K = 1$, while other SPLS approaches mostly returns $K = 1$. A first comment can be made on the stability of the cross-validation procedure. Our approach is also more stable regarding the choice of K compared to other SPLS methods. A second comment is that, as explained in the manuscript, the stability of the cross-validation is directly linked to convergence of the method (c.f. Tab. A.2). Our method always converges on our simulations and is thus more stable regarding cross-validation than other SPLS approaches that do not converge most of the time and are less stable when tuning hyper-parameters. In addition, based on these results, we decided to set $K = 1$ and only tune the sparsity parameter λ_s and the Ridge parameter λ_R when evaluating the performance of the different approaches (Tab. 3 in the manuscript) to save computation time.

Prediction and selection Tabs. A.4, A.5 and A.6 collects the results regarding performance in prediction and selection (sensitivity, specificity, accuracy) for the different approaches compared in the simulation study, and for data respectively simulated with $p = 100, 500, 1000$. These results are consistent with the case $p = 2000$ presented in the manuscript. In details, approaches that combines compression and variable selection (sparse PLS) achieve better prediction performance than compression only (PLS) or selection only (GLMNET) approaches. Regarding selection, sparse PLS is generally better in term of selection sensitivity (true positive rate) compared to GLMNET, which is too conservative. However, our approach logit-SPLS seems to select less false positives compared to other SPLS approaches, since the specificity is higher for a similar accuracy level.

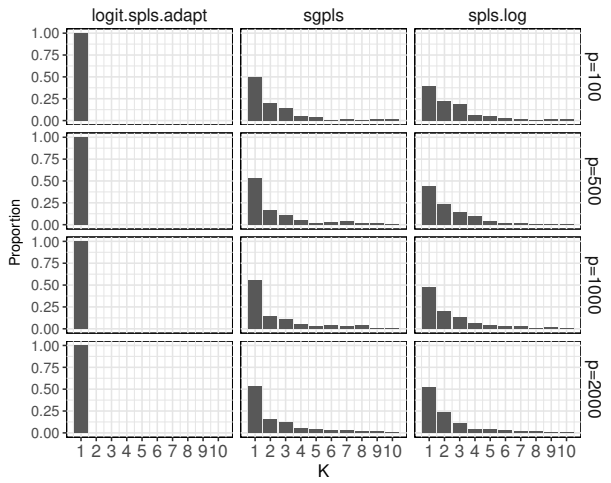

Fig. A.1. Values chosen for K by cross-validation over repetitions for the different SPLS approaches for different values of p (with $n = 100$) on simulated data.

Table A.4. Prediction error and selection sensitivity/specificity (if relevant) when $p = 100$, for non-sparse or sparse approaches (delimited by the line).

Method	Prediction error	Selection sensitivity	Selection specificity	Selection accuracy
gpls	0.51 ± 0.30	/	/	/
pls-da	0.22 ± 0.08	/	/	/
logit-pls	0.20 ± 0.08	/	/	/
glmnet	0.17 ± 0.08	0.77	0.76	0.71
logit-spls	$\mathbf{0.14 \pm 0.07}$	**0.78**	**0.86**	**0.83**
sgpls	0.14 ± 0.07	0.86	0.77	0.83
spls-da	0.15 ± 0.07	0.88	0.75	0.83
spls-log	0.12 ± 0.07	0.87	0.75	0.82

Table A.5. Prediction error and selection sensitivity/specificity (if relevant) when $p = 500$, for non-sparse or sparse approaches (delimited by the line).

Method	Prediction error	Selection sensitivity	Selection specificity	Selection accuracy
gpls	0.47 ± 0.31	/	/	/
pls-da	0.22 ± 0.08	/	/	/
logit-pls	0.19 ± 0.07	/	/	/
glmnet	0.18 ± 0.07	0.49	0.93	0.74
logit-spls	$\mathbf{0.13 \pm 0.07}$	**0.69**	**0.85**	**0.80**
sgpls	0.12 ± 0.06	0.81	0.76	0.81
spls-da	0.14 ± 0.07	0.82	0.75	0.81
spls-log	0.13 ± 0.06	0.83	0.77	0.81

Computation time. Tab. A.7 shows the averaged computation time for the cross-validation runs of the different approaches on simulated data where $n = 100$ and $p = 100, 500, 1000, 2000$. Each run was performed on the cluster grid of the LBBE, equipped with standard multi-core CPU with frequency between 2 and 2.5 GHz. For each method, each cross-validation runs did used a single core of a single CPU for two reason: (i) we did perform massive simultaneous runs on the cluster). (ii) It was a fare basis for comparison, because the different packages that we used propose different degrees of parallelization in their implementation. It is important

Table A.6. Prediction error and selection sensitivity/specificity (if relevant) when $p = 1000$, for non-sparse or sparse approaches (delimited by the line).

Method	Prediction error	Selection sensitivity	Selection specificity	Selection accuracy
gpls	0.48 ± 0.31	/	/	/
pls-da	0.21 ± 0.07	/	/	/
logit-pls	0.18 ± 0.07	/	/	/
glmnet	0.17 ± 0.06	0.37	0.96	0.74
logit-spls	$\mathbf{0.13 \pm 0.06}$	**0.66**	**0.85**	**0.80**
sgpls	0.12 ± 0.06	0.80	0.77	0.81
spls-da	0.13 ± 0.06	0.82	0.75	0.81
spls-log	0.13 ± 0.06	0.83	0.75	0.81

Table A.7. Averaged computation time (in seconds) of cross-validation runs on a single-core of a standard CPU, when considering simulated data where $n = 100$ and $p = 100, 500, 1000, 2000$.

Method	$p = 100$	$p = 500$	$p = 1000$	$p = 2000$
glmnet	4.69	4.85	5.39	6.59
logit-spls	**72.98**	**223.13**	**452.21**	**706.86**
sgpls	79.41	284.62	541.86	1103.32
spls-log	3.63	11.17	20.74	37.30

to note that our approach logit-SPLS can run on multi-core architecture, which improves the results presented below.

GLMNET is the most efficient method because its implementation relies on `fortran` and `C` codes, interfaced with `R`. SPLS-log is also quite efficient (less than a minute in all cases). Indeed, it uses the `glm` function from `R` that is encoded in `C`. However, as mentioned earlier and in the paper, this function did encounter convergence issues in many cases. Our method logit-SPLS is slower since the cross-validation takes between ~ 1 min. (when $p = 100$) and ~ 11 min. (when $p = 2000$) in average. We can make two comments here: (i) our approach needs to calibrate an additional hyper-parameter λ_R, however this additional cost is reasonable (a few minutes). (ii) The fast convergence of our approach ensures a lower computation time compared to the SGPLS approach, despite the additional hyper-parameter.

In addition, it can be noted that we are currently working on a `C++` implementation of our algorithm, which is expected to speed up the computations compared to the `R` implementation.

Eventually, Tab. A.8 presents the averaged computation time to fit a single model for the different approaches on simulated data where $n = 100$ and $p = 100, 500, 1000, 2000$. Each run was performed on the cluster grid of the LBBE, equipped with standard multi-core CPU with frequency between 2 and 2.5 GHz. For each method, each model fitting runs did used a single core of a single CPU.

All methods are computationally efficient to fit a single model, except for SGPLS. The non-convergence of this approach requires that the algorithm iterates further, until the limit set by the users. It can be noted that the cost of additional iterations in the case of SPLS-log is counter-balanced by the efficient use of the `glm` function. However, it does not guarantee its convergence (c.f. previously).

Table A.8. Averaged computation time (in seconds) of a single fit run on a single-core of a standard CPU, when considering simulated data where $n = 100$ and $p = 100, 500, 1000, 2000$.

Method	$p = 100$	$p = 500$	$p = 1000$	$p = 2000$
glmnet	0.01	0.03	0.06	0.11
logit-spls	**0.05**	**0.17**	**0.35**	**0.60**
sgpls	0.89	3.70	7.86	17.92
spls-log	0.03	0.09	0.19	0.40

A.6 Complements on the breast cancer data analysis

A.6.1 Computation details

We applied the methods GLMNET, logit-PLS, logit-SPLS (adaptive or not), SGPLS and SPLS-log to our data set. We fit each model over a hundred resamplings, where observations are randomly split into training and test sets with a 70%/30% ratio. For the prediction task, on each resampling, the parameter values of each method are tuned by 10-fold cross-validation on the training set, respecting the following grid (for our method logit-SPLS) $K \in \{1, \ldots, 8\}$, candidate values for the Ridge parameter λ_R in RIRLS are 31 points that are $\log_{10}$-linearly spaced in the range $[10^{-2}; 10^3]$, candidate values for the sparse parameter λ_s are 10 points that are linearly spaced in the range $[0.05; 0.95]$. Other SPLS approaches (SGPLS and SPLS-log) only depend on hyper-parameters (λ_s, K) for which candidate values are the same as for our method. Regarding GLMNET, we let the procedure chooses by itself the grid of hyper-parameters, as recommended by the authors in the documentation.

A.6.2 Stability selection

Hyper-parameter grid. In the study of the stability selection on the breast cancer data set, regarding our approach logit-SPLS, as a basis, we use the same grid Λ of candidates values for $(\lambda_s, \lambda_R, K)$ as in the cross-validation case (c.f. section A.6.1). As stated in the manuscript, the grid is then reduced to control the false positive expected number. For other SPLS approaches, we apply the same procedure, based on the grid (λ_s, K). Regarding GLMNET, the grid of candidate values for the penalty parameter is chosen by the procedure itself, but then we apply the same framework to extract the set of stable selected variables (as detailed in the manuscript, section 2.4).

Selected genes. The overlap between the genes selected by the different approaches based on the stability selection procedure (for a threshold $\pi_{\text{thr}} = 0.75$) are given in Fig. A.2. We can make two comments: (i) The 28 genes selected by GLMNET are all retrieved by our approach logit-SPLS (over 133 selected genes). In addition, genes with higher selection score (i.e. maximum estimated probability to be selected) are the same between the two methods (c.f. Tabs. A.9 and A.10). Thus, the selection procedure based on our logit-SPLS method is consistent with our baseline GLMNET. (ii) Genes selected by other SPLS-based approaches (SPLS-log and SGPLS) are not consistent with the ones selected by GLMNET nor by logit-SPLS. On the contrary, they select 50 common genes over respectively 58 and 70 selected genes for SPLS-log and SGPLS. However, the reliability of these results is questioned because of the non-convergence of these two methods (c.f. Tab. 4 in the manuscript). It can be noted that similar observations (consistency between logit-SPLS and GLMNET, SPLS-log and SGPLS are different) can be made for other level of probability threshold π_{thr}.

Tabs. A.9 and A.10 give the list of genes that were selected respectively by GLMNET and logit-SPLS thanks to the stability selection procedure

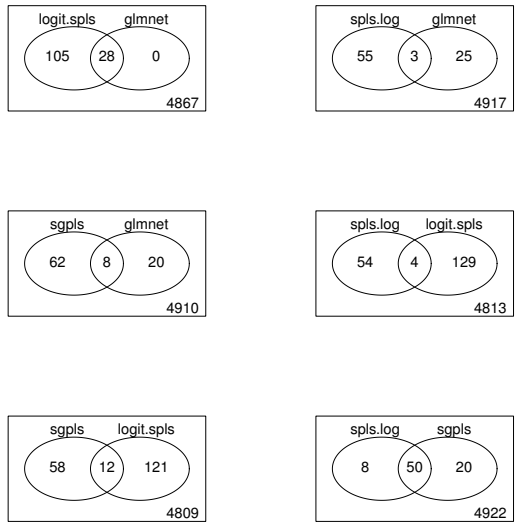

Fig. A.2. Overlap between the genes selected by the different methods thanks to the stability selection procedure, when taking a threshold $\pi_{\text{thr}} = 0.75$.

applied to the breast cancer data set (in particular to the 5000 most differentially expressed genes between the two conditions relapse or not). Genes are identified by their ProbeID on Affymetrix U133-Plus2.0 chip (c.f. Guedj *et al.*, 2012). Gene identification (Symbool, Entrezid and Name) is recovered thanks to the `annotate` and `hgu133plus2.db` R-package, that are available on Bioconductor (`https://www.bioconductor.org`). Some ProbeID were not identified and correspond to blank line. On contrary, other ProbeID seem to correspond to two genes and are present twice.

A.7 Sparse PLS for multi-group classification

We generalize our approach to a multi-categorical response. This problem is known as multinomial logistic regression or polytomous regression (McCullagh and Nelder, 1989) and will be called multinomial sparse PLS in the sequel.

A.7.1 Multinomial logistic regression

The response y_i takes its values in a discrete set $\{0, \ldots, G\}$ corresponding to $G + 1$ groups or classes of observations. The associated variable Y_i $(i = 1, \ldots, n)$ follows a multi-categorical distribution where $\mathbb{P}(Y_{ij} = g \,|\, \mathbf{x}_i) = \pi_{ig}$ for any class g. Based on a direct generalization of the logistic model, a class of reference is set (generally the class 0) and for each class $g \neq 0$, the probability π_{ig} that $Y_i = g$ depends on a linear combination of predictor such as:

$$\log\left(\frac{\pi_{ig}}{\pi_{i0}}\right) = \mathbf{z}_i^T \boldsymbol{\beta}_g, \tag{A.9}$$

with a specific vector of coefficient $\boldsymbol{\beta}_g \in \mathbb{R}^{p+1}$ for each class $g = 1, \ldots, G$. Indeed, the probabilities $(\pi_{ig})_{g=1:G}$ determine the probability π_{i0} since $\sum_{g=0}^{G} \pi_{ig} = 1$. A column of 1s is added in the matrix $\mathbf{Z}$

Table A.9. List of the 28 genes selected by GLMNET thanks to the stability selection procedure (at threshold $\pi_{\text{thr}} = 0.75$) on the breast cancer data set. Genes are sorted by selection score (maximum estimated probability to be selected). Genes are identified by their ProbeID on Affymetrix U133-Plus2.0 chip.

PROBEID	SYMBOL	ENTREZID	GENE NAME	Selection score
217048_at				0.99
1553561_at	TAS2R50	259296	taste 2 receptor member 50	0.97
233227_at	KIAA1109	84162	KIAA1109	0.97
218307_at	RSAD1	55316	radical S-adenosyl methionine domain containing 1	0.97
241034_at	GLS	2744	glutaminase	0.97
211870_s_at	PCDHA3	56145	protocadherin alpha 3	0.95
211870_s_at	PCDHA2	56146	protocadherin alpha 2	0.95
1561665_at	LOC100421171	100421171	thyroid hormone receptor interactor 11 pseudogene	0.95
216738_at				0.92
236899_at				0.91
227240_at	NGEF	25791	neuronal guanine nucleotide exchange factor	0.91
234739_at				0.91
1560522_at	DLGAP1-AS3	201477	DLGAP1 antisense RNA 3	0.89
1554988_at	SLC9C2	284525	solute carrier family 9 member C2 (putative)	0.86
217360_x_at	IGHA1	3493	immunoglobulin heavy constant alpha 1	0.85
217360_x_at	IGHG1	3500	immunoglobulin heavy constant gamma 1 (G1m marker)	0.85
217360_x_at	IGHG3	3502	immunoglobulin heavy constant gamma 3 (G3m marker)	0.85
217360_x_at	IGHM	3507	immunoglobulin heavy constant mu	0.85
217360_x_at	IGHV4-31	28396	immunoglobulin heavy variable 4-31	0.85
229485_x_at	SHISA3	152573	shisa family member 3	0.85
239052_at				0.85
242870_at				0.84
228776_at	GJC1	10052	gap junction protein gamma 1	0.83
244849_at	SEMA3A	10371	semaphorin 3A	0.83
217391_x_at				0.82
229215_at	ASCL2	430	achaete-scute family bHLH transcription factor 2	0.82
235945_at				0.82
1554708_s_at	SPATA6L	55064	spermatogenesis associated 6 like	0.79
208777_s_at	PSMD11	5717	proteasome 26S subunit, non-ATPase 11	0.79
213651_at	INPP5J	27124	inositol polyphosphate-5-phosphatase J	0.78
225792_at	HOOK1	51361	hook microtubule tethering protein 1	0.78
1570136_at				0.76
1560692_at	VSTM2A-OT1	285878	VSTM2A overlapping transcript 1	0.75
1560692_at	VSTM2A	222008	V-set and transmembrane domain containing 2A	0.75

to incorporate the intercept in the linear combination $\mathbf{z}_i^T \boldsymbol{\beta}_g$. The log-likelihood can be be explicitly formulated:

$$\log \mathcal{L}(\boldsymbol{\beta}) = \sum_{i=1}^{n} \left\{ \sum_{g=1}^{G} y_{ig}\, \mathbf{z}_i^T \boldsymbol{\beta}_g - \log\left(1 + \textstyle\sum_{g=1}^{G} \exp(\mathbf{z}_i^T \boldsymbol{\beta}_g)\right) \right\}, \tag{A.10}$$

where the binary variable $y_{ig} = \mathbf{1}_{\{y_i = g\}}$ indicates the class of the observation i ($\mathbf{1}_{\{\mathcal{A}\}}$ is the indicator function valued in $\{0, 1\}$, indicating if the statement $\mathcal{A}$ is true (1) or false (0).)

It is possible to rearrange the data in order to formulate a vectorized version of the loss (A.10), and express the multinomial logistic regression as a logistic regression of a binary response $\mathcal{Y} \in \{0,1\}^{n\,G}$ against a matrix of rearranged covariates $\mathcal{Z} \in \mathbb{R}^{n\,G \times (p+1)\,G}$. The response vector $\mathcal{Y}$ of length $n\,G$ is defined as follows:

$$\mathcal{Y} = \Big((y_{1g})_{g=1:G}, \ldots, (y_{ig})_{g=1:G}, \ldots, (y_{ng})_{g=1:G} \Big)^T,$$

where $y_{ig} = \mathbf{1}_{\{y_i = g\}}$ as previously mentioned. The new covariate matrix $\mathcal{Z}$ of dimension $n\,G \times (p+1)\,G$ is defined by blocks as:

$$\mathcal{Z} = \Big[\mathcal{Z}_1^T, \ldots, \mathcal{Z}_i^T, \ldots, \mathcal{Z}_n^T \Big]^T,$$

where each block i is constructed by G diagonal repetitions of the row $\widetilde{\mathbf{x}}_i$ from the original covariate matrix $\mathbf{Z}$, i.e.

$$\mathcal{Z}_i = \begin{pmatrix} 1\; x_{i1}\; \ldots\; x_{ip} & & 0 \\ & \ddots & \\ 0 & & 1\; x_{i1}\; \ldots\; x_{ip} \end{pmatrix} \Bigg\} G \text{ repeats of row } \mathbf{z}_i^T.$$

The coefficient vectors $\boldsymbol{\beta}_g \in \mathbb{R}^{p+1}$ (for $g = 1, \ldots, G$) are also reorganized in the vector $\mathbf{B} \in \mathbb{R}^{(p+1)\,G}$ as:

$$\mathbf{B} = \Big((\beta_{0g})_{g=1:G}, \ldots, (\beta_{jg})_{g=1:G}, \ldots, (\beta_{pg})_{g=1:G} \Big)^T,$$

where $(\beta_{jg})_{j=0:p}$ are the coordinates of $\boldsymbol{\beta}_g$, so that the response $\mathcal{Y}$ depends on the linear combination $\mathcal{Z}\,\mathbf{B}$.

Thanks to this reformulation, it is possible to adapt the Ridge IRLS algorithm to estimate the coefficients $\mathbf{B}$ and infer the probabilities π_{ig} that observations y_i belongs to the class g. The algorithm that we call MRIRLS is detailed in Fort *et al.* (2005).

A.7.2 Multinomial SPLS

The vectorized formulation of the MIRLS algorithm allows to use our SPLS-based dimension reduction approach. As in the binary case, the

Table A.10. List of the top 50 genes (over 133) selected by logit-SPLS thanks to the stability selection procedure (at threshold $\pi_{\text{thr}} = 0.75$) on the breast cancer data set. Genes sorted by selection score (maximum estimated probability to be selected). Genes are identified by their ProbeID on Affymetrix U133-Plus2.0 chip.

PROBEID	SYMBOL	ENTREZID	GENE NAME	Selection score
1553561_at	TAS2R50	259296	taste 2 receptor member 50	1.00
218307_at	RSAD1	55316	radical S-adenosyl methionine domain containing 1	1.00
1560522_at	DLGAP1-AS3	201477	DLGAP1 antisense RNA 3	0.99
217048_at				0.99
211870_s_at	PCDHA3	56145	protocadherin alpha 3	0.99
211870_s_at	PCDHA2	56146	protocadherin alpha 2	0.99
233227_at	KIAA1109	84162	KIAA1109	0.99
220098_at	HYDIN	54768	HYDIN, axonemal central pair apparatus protein	0.98
220098_at	HYDIN2	100288805	HYDIN2, axonemal central pair apparatus protein (pseudogene)	0.98
234739_at				0.98
1561665_at	LOC100421171	100421171	thyroid hormone receptor interactor 11 pseudogene	0.97
216738_at				0.97
241034_at	GLS	2744	glutaminase	0.97
227240_at	NGEF	25791	neuronal guanine nucleotide exchange factor	0.97
1554988_at	SLC9C2	284525	solute carrier family 9 member C2 (putative)	0.95
1560692_at	VSTM2A-OT1	285878	VSTM2A overlapping transcript 1	0.95
1560692_at	VSTM2A	222008	V-set and transmembrane domain containing 2A	0.95
217360_x_at	IGHA1	3493	immunoglobulin heavy constant alpha 1	0.95
217360_x_at	IGHG1	3500	immunoglobulin heavy constant gamma 1 (G1m marker)	0.95
217360_x_at	IGHG3	3502	immunoglobulin heavy constant gamma 3 (G3m marker)	0.95
217360_x_at	IGHM	3507	immunoglobulin heavy constant mu	0.95
217360_x_at	IGHV4-31	28396	immunoglobulin heavy variable 4-31	0.95
236899_at				0.95
239052_at				0.95
242870_at				0.95
228507_at	PDE3A	5139	phosphodiesterase 3A	0.95
229081_at	SLC25A13	10165	solute carrier family 25 member 13	0.95
1562030_at	LOC284898	284898	uncharacterized LOC284898	0.94
227379_at	MBOAT1	154141	membrane bound O-acyltransferase domain containing 1	0.94
225792_at	HOOK1	51361	hook microtubule tethering protein 1	0.93
234792_x_at	IGHA1	3493	immunoglobulin heavy constant alpha 1	0.93
234792_x_at	IGHV4-31	28396	immunoglobulin heavy variable 4-31	0.93
244849_at	SEMA3A	10371	semaphorin 3A	0.93
217697_at				0.93
1556937_at				0.92
1569126_at	CCNC	892	cyclin C	0.92
228776_at	GJC1	10052	gap junction protein gamma 1	0.92
229485_x_at	SHISA3	152573	shisa family member 3	0.92
232920_at	KIAA1656	85371	KIAA1656 protein	0.92
232920_at	CCDC157	550631	coiled-coil domain containing 157	0.92
1563057_at				0.91
1568666_at	PLIN5	440503	perilipin 5	0.91
1570116_at				0.91
238824_at	RPS29	6235	ribosomal protein S29	0.91
243583_at				0.91
206349_at	LGI1	9211	leucine rich glioma inactivated 1	0.91
211064_at	ZNF493	284443	zinc finger protein 493	0.91
231913_s_at	BRCC3	79184	BRCA1/BRCA2-containing complex subunit 3	0.91
1570136_at				0.89
206202_at	MEOX2	4223	mesenchyme homeobox 2	0.89

MIRLS algorithm (penalized by Ridge) produces a continuous pseudo-response (at the convergence) that is suitable for the sparse PLS regression. Thus, our approach, called multinomial-SPLS, directly extends our algorithm logit-SPLS to the multinomial logistic regression. It estimates the linear coefficients $\mathbf{B}$ by sparse PLS, processing compression and variable selection simultaneously. Then, these estimated coefficients are used to get an estimation of the probabilities π_{ig}. Our procedure is directly inspired from the approach by Fort *et al.* (2005) that extended the algorithm logit-PLS (Fort and Lambert-Lacroix, 2005) to the multi-categorical cases.

In this context, the SPLS step considers: i) the pseudo-response $\boldsymbol{\xi} \in \mathbb{R}^{n\,G}$ constructed from the reformulated response $\mathcal{Y}$, ii) the centered version $\mathcal{X}_c$ of the modified covariate matrix $\mathcal{X}$ defined by:

$$\mathcal{X} = \left[\mathcal{X}_1^T, \ldots, \mathcal{X}_i^T, \ldots, \mathcal{X}_n^T\right]^T,$$

where each block i is constructed by G diagonal repetitions of the row $\widetilde{\mathbf{x}}_i$ from the original covariate matrix $\mathbf{X}$, i.e.

$$\mathcal{X}_i = \left.\begin{pmatrix} x_{i1} \ \ldots \ x_{ip} & & 0 \\ & \ddots & \\ 0 & & x_{i1} \ \ldots \ x_{ip} \end{pmatrix}\right\} G \text{ repeats of row } \mathbf{x}_i^T.$$

It corresponds to the matrix $\mathcal{Z}$ where the terms 1 corresponding to the intercept have been removed. Thus, the coefficients $(\beta_{0g})_{g=1:G}$ are estimated afterward. These coefficients are ultimately used to compute the class membership probabilities for each observation, following the model (A.9). In prediction task, an observation is assigned to the class with the highest predicted probability.

At this point, we mention that the error rate that we consider in this case (especially for tuning of hyper-parameters by V-fold cross-validation, with $V = 5$ or 10) is the standard error rate, i.e. the proportion of overall mismatches, that previously used in the `plsgenomics` R-package for multi-class PLS classification.

A.7.3 SPLS components

Since the sparse PLS is applied on the modified covariate matrix $\mathcal{X} \in \mathbb{R}^{n\,G \times p\,G}$, the constructed SPLS components represent the matrix $\mathcal{X}$ in a lower dimensional subspace, and not the original matrix $\mathbf{X}$. However, it is possible to obtain a low dimensional representation of the original covariates. Indeed, thanks to the construction of the matrix $\mathcal{X}$, the SPLS weight vectors $\mathbf{w}_k \in \mathbb{R}^{p\,G}$ are partitioned as follows:

$$\mathbf{w}_k = \left((w_{j1}^k)_{j=1:p}, \ldots, (w_{jg}^k)_{j=1:p}, \ldots, (w_{jG}^k)_{j=1:p}\right)^T,$$

for $k = 1, \ldots, K$. Thus when multiplying the original predictor matrix $\mathbf{X}$ by the weights matrix $\left[(w_{jg}^k)_{j=1:p}\right]_{k=1:K} \in \mathbb{R}^{p \times K}$, we obtain a representation of the observations in a lower dimensional space of dimension K, as a matrix $\mathbf{T}_g \in \mathbb{R}^{n \times K}$. The matrix $\mathbf{T}_g$ represents the directions that discriminate the class g versus the class reference 0.

A.7.4 State-of-the-art

It can be noted that Ding and Gentleman (2005) presented a version of the GPLS method suitable for multinomial logistic regression, i.e. the linear regression inside the iteration of the MIRLS algorithm are processed by weighted PLS regression. Chung and Keleş (2010) introduced a similar algorithm based on sparse PLS (extension of the SGPLS algorithm). However, we used exclusively our multinomial SPLS algorithm in the data analysis. Indeed, based on the conclusions from the binary case, our approach showed better results regarding prediction performance on an experimental data set. Moreover, the dimension of the data is drastically increased because of the rearrangement since the number of observations becomes $n\,G$ and the number of covariates becomes $p\,G$. It is therefore necessary to account for the computational cost and to give priority to computationally efficient methods. In particular, thanks to the Ridge penalty, we showed that our approach converges quickly, hence reducing the time of computation.

A.8 Complements on the single T cell data analysis

A.8.1 Computation details

On each resampling, the parameter values of each method are tuned by 10-fold cross-validation on the training set, respecting the following grid $K \in \{1, \ldots, 4\}$, candidate values for the Ridge parameter λ_R in RIRLS are 10 points that are $\log_{10}$-linearly spaced in the range $[10^{-2}; 10^{3}]$, candidate values for the sparse parameter λ_s are 10 points that are linearly spaced in the range $[0.05; 0.95]$.

A.8.2 Additional results single cell data analysis

Training in the first step of prediction. The manual identification of cells is mainly based on the level of the CCR7 markers. The identified cells mostly correspond to the most extreme values of CCR7 level. The set of manually identified cells constitutes the training set for the first step of prediction based on multinomial sparse PLS. Fig. A.3 illustrates the representation of the cells in the training set according to the first two PLS components. The distinction between the reference class ("CM") and both classes from the group of "Effector" cells ("EM" and "TEMRA") is clearly apparent in the latent subspace, since there is an important gap between the different groups of cells. It confirms that the cells in the training set correspond to the most extreme phenotypes that appear clearly different.

Genes selection by sparse PLS. The genes that are selected by the multinomial-SPLS during the second round of prediction (as explained in the manuscript) are the following: "CCL4", "CCR7", "CST7", "GNLY", "GZMB", "KLRD1", "LTB", "S100A4". These genes have been identified as genes involved in the phenotype ("Effector" or "Memory") of T-cells (Wherry *et al.*, 2007; Willinger *et al.*, 2005b). In particular, "CCR7" and "LTB" are associated to "Memory" cells, while "CCL4", "CST7", "GNLY", "GZMB" and "KLRD1" characterized "Effector" cells.

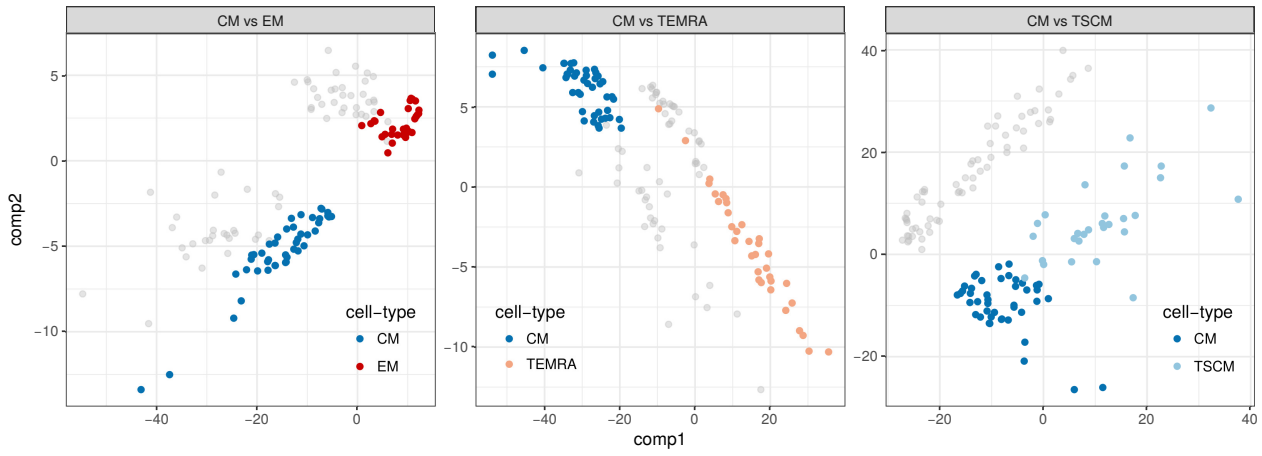

Fig. A.3. Cell scores on the first two PLS components in the latent space that discriminate between the reference class ("CM") and each other class separately ("EM", "TEMRA" and "TSCM" respectively, from left to right). Restriction to the T cells in the training set before the first prediction step.