PLS for Big Data: A Unified Parallel Algorithm for Regularized Group PLS.

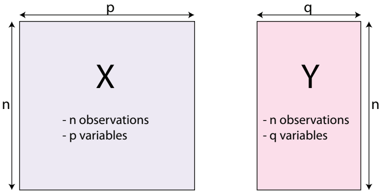
B. Liquet[1,2] and M. Sutton[2] and P. Lafaye De Micheaux[3]

[1] LMAP,E2S-UPPA, Université de Pau et des Pays de L'Adour

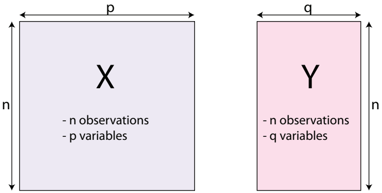
[2] ACEMS, QUT, Australia

[3] UNSW Sydney, Australia

Data Definition and Examples



Data Definition and Examples



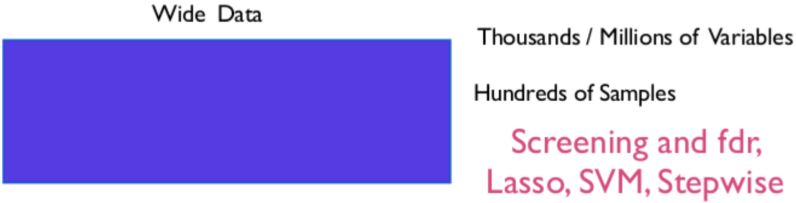
- “Omics.” **Y** matrix: gene expression, **X** matrix: SNP (single nucleotide polymorphism). Many others such as proteomic, metabolomic data.
- “neuroimaging”. **Y** matrix: behavioral variables, **X** matrix: brain activity (e.g., EEG, fMRI, NIRS)
- “neuroimaging genetics.” **Y** matrix: fMRI (Fusion of functional magnetic resonance imaging), **X** matrix: SNP
- “Ecology/Environment.” **Y** matrix: Water quality variables , **X** matrix: Landscape variables

Definition of BIG DATA

Big data vary in shape: These call for different approaches

- Wide Data
- Tall Data
- Tall and Wide

BIG DATA: Wide Data



Wide Data

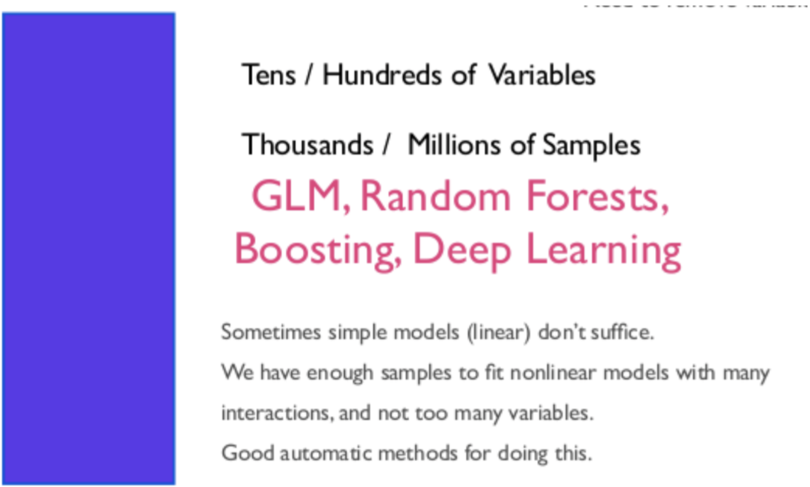
Thousands / Millions of Variables

Hundreds of Samples

Screening and fdr,
Lasso, SVM, Stepwise

We have too many variables, prone to overfitting. Need to remove variable, or regularize, or both

BIG DATA: Tall Data



Tens / Hundreds of Variables

Thousands / Millions of Samples

GLM, Random Forests, Boosting, Deep Learning

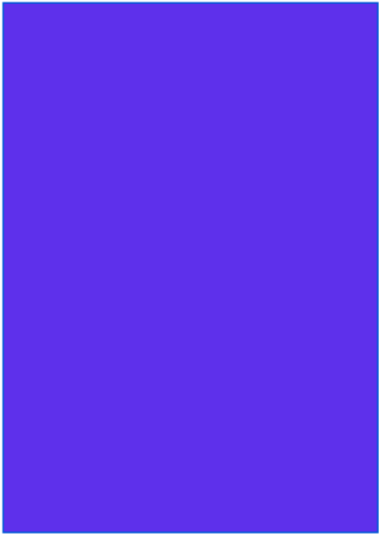
Sometimes simple models (linear) don't suffice.

We have enough samples to fit nonlinear models with many interactions, and not too many variables.

Good automatic methods for doing this.

BIG DATA: Tall and Wide Data

Tall and Wide Data



Thousands / Millions of Variables

Millions to Billions of Samples

Tricks of the Trade

Exploit sparsity

Random projections / hashing

Variable screening

Subsample rows

Divide and recombine

Case/ control sampling

MapReduce

ADMM (divide and conquer)

.
.
.

Genomics Data: Wide Data, High Dimensional Data

- Main constraint: situation with $p > n$
- Strong colinearity among the variables.

Genomics Data: Wide Data, High Dimensional Data

- Main constraint: situation with $p > n$
- Strong colinearity among the variables.

Contribution:

- Incorporation of knowledge on the structure existing in the data
- Potential grouping of the covariates is key to:
  - more accurate prediction
  - improved interpretability

Group structures within the data

- Genomics: genes within the same pathway have similar functions and act together in regulating a biological system.

↪ These genes can add up to have a larger effect

↪ can be detected as a group (i.e., at a pathway or gene set/module level).

Group structures within the data

- Genomics: genes within the same pathway have similar functions and act together in regulating a biological system.

↪ These genes can add up to have a larger effect

↪ can be detected as a group (i.e., at a pathway or gene set/module level).

We consider variables are divided into groups:

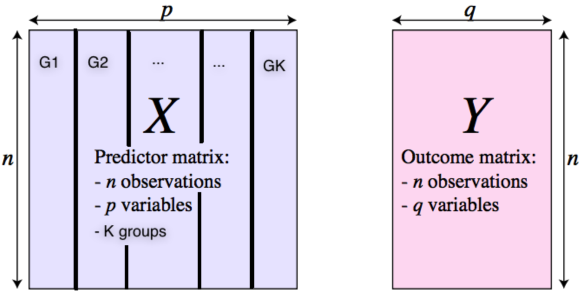
- Example p: SNPs grouped into K genes

$$\mathbf{X} = [\underbrace{SNP_1, \ldots + SNP_k}_{gene_1} \,|\, \underbrace{SNP_{k+1}, SNP_{k+2}, \ldots, SNP_h}_{gene_2} \,|\, \ldots \,|\, \underbrace{SNP_{l+1}, \ldots, SNP_p}_{gene_K}]$$

- Example p: genes grouped into K pathways/modules ($X_j = \mathrm{gene}_j$)

$$\mathbf{X} = [\underbrace{X_1, X_2, \ldots, X_k}_{M_1} \,|\, \underbrace{X_{k+1}, X_{k+2}, \ldots, X_h}_{M_2} \,|\, \ldots \,|\, \underbrace{X_{l+1}, X_{l+2}, \ldots, X_p}_{M_K}]$$

Our contribution for Multivariate phenotypes



- Select group variables taking into account the data structures; all the variables within a group are selected otherwise none of them are selected
- Combine both sparsity of groups and within each group; only relevant variables within a group are selected

Our contribution for Multivariate phenotypes

- Sparse Group PLS : SNP $\subset$ Gene or Gene $\subset$ Pathways

Liquet B., Lafaye de Micheaux P., Hejblum B. and Thiebaut R., (2016) *Group and Sparse Group Partial Least Square Approaches Applied in Genomics Context.* **Bioinformatics**, 32(1), 35–42.

```r
library(sgPLS)
```

- Sparse Group subgroup PLS : SNP $\subset$ Gene $\subset$ Pathways

M. Sutton, R. Thiebaut, and B. Liquet. (2018) *Sparse group subgroup Partial Least Squares with application to genomics data*. Statistics in Medicine.

```r
install_github("sgsPLS", "matt-sutton")
```

Our contribution for Multivariate phenotypes

- Sparse Group PLS : SNP $\subset$ Gene or Gene $\subset$ Pathways

Liquet B., Lafaye de Micheaux P., Hejblum B. and Thiebaut R., (2016) *Group and Sparse Group Partial Least Square Approaches Applied in Genomics Context.* **Bioinformatics**, 32(1), 35–42.

```
library(sgPLS)
```

- Sparse Group subgroup PLS : SNP $\subset$ Gene $\subset$ Pathways

M. Sutton, R. Thiebaut, and B. Liquet. (2018) *Sparse group subgroup Partial Least Squares with application to genomics data*. Statistics in Medicine.

```
install_github("sgsPLS", "matt-sutton")
```

Main ideas:

- combining L_1 and L_2 penalties into the optimization function
- **Sparse Group Penalties:**

$$\lambda_1 \sum_{g=1}^{G} \sqrt{p_g} ||\boldsymbol{\beta}_g||_2 + \lambda_2 ||\boldsymbol{\beta}||_1$$

Why PLS ?

Aims:

1. Symmetric situation. Analyze the association between two blocks of information. Analysis focused on shared information.
2. Asymmetric situation. **X** matrix= predictors and **Y** matrix= response variables. Analysis focused on prediction.

Why PLS ?

Aims:

1. Symmetric situation. Analyze the association between two blocks of information. Analysis focused on shared information.
2. Asymmetric situation. **X** matrix= predictors and **Y** matrix= response variables. Analysis focused on prediction.

- Partial Least Square Family: dimension reduction approaches

Why PLS ?

Aims:

1. Symmetric situation. Analyze the association between two blocks of information. Analysis focused on shared information.
2. Asymmetric situation. **X** matrix= predictors and **Y** matrix= response variables. Analysis focused on prediction.

- Partial Least Square Family: dimension reduction approaches
  - PLS finds pairs of latent vectors $\boldsymbol{\xi} = \boldsymbol{Xu}$, $\boldsymbol{\omega} = \boldsymbol{Yv}$ with maximal covariance.

$$e.g., \quad \xi = u_1 \times SNP_1 + u_2 \times SNP_2 + \cdots + u_p \times SNP_p$$

  - Symmetric situation and Asymmetric situation.
  - Matrix decomposition of **X** and **Y** into successive latent variables.

Latent variables: are not directly observed but are rather inferred (through a mathematical model) from other variables that are observed (directly measured). Capture an underlying phenomenon (e.g., health).

How it works ?

Now some mathematics ...

PLS family

PLS = Partial Least Squares or Projection to Latent Structures\ $ $\
Four main methods coexist in the literature:

(i) Partial Least Squares Correlation (PLSC) also called PLS-SVD;

(ii) PLS in mode A (PLS-W2A, for Wold's Two-Block, Mode A PLS);

(iii) PLS in mode B (PLS-W2B) also called Canonical Correlation Analysis (CCA);

(iv) Partial Least Squares Regression (PLSR, or PLS2).

PLS family

PLS = Partial Least Squares or Projection to Latent Structures\ $ $\
Four main methods coexist in the literature:

(i) Partial Least Squares Correlation (PLSC) also called PLS-SVD;
(ii) PLS in mode A (PLS-W2A, for Wold's Two-Block, Mode A PLS);
(iii) PLS in mode B (PLS-W2B) also called Canonical Correlation Analysis (CCA);
(iv) Partial Least Squares Regression (PLSR, or PLS2).

- (i),(ii) and (iii) are symmetric while (iv) is asymmetric.
- Different objective functions to optimise.
- Good news: all use the singular value decomposition (SVD).

Singular Value Decomposition (SVD)

Definition

Let a matrix $\mathcal{M} : p \times q$ of rank r:

$$\mathcal{M} = \mathcal{U}\Delta\mathcal{V}^{\mathsf{T}} = \sum_{l=1}^{r} \delta_l \boldsymbol{u}_l \boldsymbol{v}_l^{\mathsf{T}}, \tag{1}$$

- $\mathcal{U} = (\boldsymbol{u}_l) : p \times p$ and $\mathcal{V} = (\boldsymbol{v}_l) : q \times q$ are two orthogonal matrices which contain the normalised left (resp. right) singular vectors
- $\Delta = \mathrm{diag}(\delta_1, \ldots, \delta_r, 0, \ldots, 0)$: the ordered singular values $\delta_1 \geqslant \delta_2 \geqslant \cdots \geqslant \delta_r > 0$.

Note: fast and efficient algorithms exist to solve the SVD.

Connexion between SVD and maximum covariance

We were able to describe the optimization problem of the **four** PLS methods as:

$$(\boldsymbol{u}^*, \boldsymbol{v}^*) \quad = \quad \underset{\|\boldsymbol{u}\|_2=\|\boldsymbol{v}\|_2=1}{\operatorname{argmax}} \; Cov(\mathbf{X}_{h-1}\boldsymbol{u}, \mathbf{Y}_{h-1}\boldsymbol{v}), \qquad h = 1, \ldots, H.$$

Matrices $\mathbf{X}_h$ and $\mathbf{Y}_h$ are obtained recursively from $\mathbf{X}_{h-1}$ and $\mathbf{Y}_{h-1}$.

Connexion between SVD and maximum covariance

We were able to describe the optimization problem of the **four** PLS methods as:

$$(\boldsymbol{u}^*, \boldsymbol{v}^*) = \underset{\|\boldsymbol{u}\|_2=\|\boldsymbol{v}\|_2=1}{\operatorname{argmax}} Cov(\mathbf{X}_{h-1}\boldsymbol{u}, \mathbf{Y}_{h-1}\boldsymbol{v}), \qquad h = 1, \ldots, H.$$

Matrices $\mathbf{X}_h$ and $\mathbf{Y}_h$ are obtained recursively from $\mathbf{X}_{h-1}$ and $\mathbf{Y}_{h-1}$.

The four methods differ by the deflation process, chosen so that the above scores or weight vectors satisfy given constraints.

Connexion between SVD and maximum covariance

We were able to describe the optimization problem of the **four** PLS methods as:

$$(\boldsymbol{u}^*, \boldsymbol{v}^*) \quad = \quad \underset{\|\boldsymbol{u}\|_2 = \|\boldsymbol{v}\|_2 = 1}{\operatorname{argmax}} \; Cov(\mathbf{X}_{h-1}\boldsymbol{u}, \mathbf{Y}_{h-1}\boldsymbol{v}), \qquad h = 1, \ldots, H.$$

Matrices $\mathbf{X}_h$ and $\mathbf{Y}_h$ are obtained recursively from $\mathbf{X}_{h-1}$ and $\mathbf{Y}_{h-1}$.

The four methods differ by the deflation process, chosen so that the above scores or weight vectors satisfy given constraints.

The solution at step h is obtained by computing **only the first** triplet $(\delta_1, \boldsymbol{u}_1, \boldsymbol{v}_1)$ of singular elements of the SVD of $\mathcal{M}_{h-1} = \mathbf{X}_{h-1}^{\mathsf{T}}\mathbf{Y}_{h-1}$:

$$(\boldsymbol{u}^*, \boldsymbol{v}^*) = (\boldsymbol{u}_1, \boldsymbol{v}_1)$$

Connexion between SVD and maximum covariance

We were able to describe the optimization problem of the **four** PLS methods as:

$$(\boldsymbol{u}^*, \boldsymbol{v}^*) \quad = \quad \underset{\|\boldsymbol{u}\|_2 = \|\boldsymbol{v}\|_2 = 1}{\operatorname{argmax}} \; Cov(\mathbf{X}_{h-1}\boldsymbol{u}, \mathbf{Y}_{h-1}\boldsymbol{v}), \qquad h = 1, \ldots, H.$$

Matrices $\mathbf{X}_h$ and $\mathbf{Y}_h$ are obtained recursively from $\mathbf{X}_{h-1}$ and $\mathbf{Y}_{h-1}$.

The four methods differ by the deflation process, chosen so that the above scores or weight vectors satisfy given constraints.

The solution at step h is obtained by computing **only the first** triplet $(\delta_1, \boldsymbol{u}_1, \boldsymbol{v}_1)$ of singular elements of the SVD of $\mathcal{M}_{h-1} = \mathbf{X}_{h-1}^{\mathbf{T}}\mathbf{Y}_{h-1}$:

$$(\boldsymbol{u}^*, \boldsymbol{v}^*) = (\boldsymbol{u}_1, \boldsymbol{v}_1)$$

Why is this useful ?

SVD properties

Theorem

Eckart-Young (1936) states that the (truncated) SVD of a given matrix $\mathcal{M}$ (of rank r) provides the best reconstitution (in a least squares sense) of $\mathcal{M}$ by a matrix with a lower rank k:

$$\min_{\mathcal{A} \text{ of rank } k} \|\mathcal{M} - \mathcal{A}\|_F^2 = \left\| \mathcal{M} - \sum_{\ell=1}^{k} \delta_\ell \boldsymbol{u}_\ell \boldsymbol{v}_\ell^\mathsf{T} \right\|_F^2 = \sum_{\ell=k+1}^{r} \delta_\ell^2 .$$

If the minimum is searched for matrices $\mathcal{A}$ of rank 1, which are under the form $\widetilde{\boldsymbol{u}}\widetilde{\boldsymbol{v}}^\mathsf{T}$ where $\widetilde{\boldsymbol{u}}$, $\widetilde{\boldsymbol{v}}$ are non-zero vectors, we obtain

$$\min_{\widetilde{\boldsymbol{u}}, \widetilde{\boldsymbol{v}}} \left\| \mathcal{M} - \widetilde{\boldsymbol{u}}\widetilde{\boldsymbol{v}}^\mathsf{T} \right\|_F^2 = \sum_{\ell=2}^{r} \delta_\ell^2 = \left\| \mathcal{M} - \delta_1 \boldsymbol{u}_1 \boldsymbol{v}_1^\mathsf{T} \right\|_F^2 .$$

SVD properties

Thus, solving

$$\underset{\widetilde{\boldsymbol{u}},\widetilde{\boldsymbol{v}}}{\operatorname{argmin}} \left\| \mathcal{M}_{h-1} - \widetilde{\boldsymbol{u}}\widetilde{\boldsymbol{v}}^{\mathsf{T}} \right\|_F^2 \tag{2}$$

and norming the resulting vectors gives us $\boldsymbol{u}_1$ and $\boldsymbol{v}_1$. This is another approach to solve the PLS optimization problem.

Towards sparse PLS

- Shen and Huang (2008) connected (2) (in a PCA context) to least square minimisation in regression:

$$\left\|\mathcal{M}_{h-1} - \widetilde{\boldsymbol{u}}\widetilde{\boldsymbol{v}}^{\mathsf{T}}\right\|_F^2 = \left\|\underbrace{\mathrm{vec}(\mathcal{M}_{h-1})}_{\boldsymbol{y}} - \underbrace{(\mathcal{I}_p \otimes \widetilde{\boldsymbol{u}})\widetilde{\boldsymbol{v}}}_{\mathcal{X}\beta}\right\|_2^2 = \left\|\underbrace{\mathrm{vec}(\mathcal{M}_{h-1})}_{\boldsymbol{y}} - \underbrace{(\widetilde{\boldsymbol{v}} \otimes \mathcal{I}_q)\widetilde{\boldsymbol{u}}}_{\mathcal{X}\beta}\right\|_2^2 .$$

↪ Possible to use many existing variable selection techniques using regularization penalties.

Towards sparse PLS

- Shen and Huang (2008) connected (2) (in a PCA context) to least square minimisation in regression:

$$\left\|\mathcal{M}_{h-1}-\widetilde{\boldsymbol{u}}\widetilde{\boldsymbol{v}}^{\mathsf{T}}\right\|_F^2=\left\|\underbrace{\mathrm{vec}(\mathcal{M}_{h-1})}_{\boldsymbol{y}}-\underbrace{(\mathcal{I}_p\otimes\widetilde{\boldsymbol{u}})\widetilde{\boldsymbol{v}}}_{\mathcal{X}\beta}\right\|_2^2=\left\|\underbrace{\mathrm{vec}(\mathcal{M}_{h-1})}_{\boldsymbol{y}}-\underbrace{(\widetilde{\boldsymbol{v}}\otimes\mathcal{I}_q)\widetilde{\boldsymbol{u}}}_{\mathcal{X}\beta}\right\|_2^2.$$

  $\hookrightarrow$ Possible to use many existing variable selection techniques using regularization penalties.

We propose iterative **alternating** algorithms to find normed vectors $\widetilde{\boldsymbol{u}}/\|\widetilde{\boldsymbol{u}}\|$ and $\widetilde{\boldsymbol{v}}/\|\widetilde{\boldsymbol{v}}\|$ that minimise the following penalised sum-of-squares criterion

$$\left\|\mathcal{M}_{h-1}-\widetilde{\boldsymbol{u}}\widetilde{\boldsymbol{v}}^{\mathsf{T}}\right\|_F^2+P_\lambda(\widetilde{\boldsymbol{u}},\widetilde{\boldsymbol{v}}),$$

for various penalization terms $P_\lambda(\widetilde{\boldsymbol{u}},\widetilde{\boldsymbol{v}})$.

$\hookrightarrow$ We obtain several sparse versions (in terms of the weights $\boldsymbol{u}$ and $\boldsymbol{v}$) of the four methods (i)–(iv).

Regularized PLS scalable for BIG-DATA

What happens in a MASSIVE DATA SET context?

Regularized PLS scalable for BIG-DATA

What happens in a MASSIVE DATA SET context?

Massive datasets. The size of the data is large and analysing it takes a significant amount of time and computer memory.

Emerson & Kane (2012). Dataset considered large if it exceeds 20% of the RAM (Random Access Memory) on a given machine, and massive if it exceeds 50%

Tall Data

Case of a lot of observations: two massive data sets **X**: $n \times p$ matrix and **Y**: $n \times q$ matrix due to a large number of observations.

We suppose here that n is very large, but not p nor q.

Tall Data

Case of a lot of observations: two massive data sets **X**: $n \times p$ matrix and **Y**: $n \times q$ matrix due to a large number of observations.

We suppose here that n is very large, but not p nor q.

PLS algorithm mainly based on the SVD of $\mathcal{M}_{h-1} = \mathbf{X}_{h-1}^{\mathbf{T}} \mathbf{Y}_{h-1}$:

Tall Data

Case of a lot of observations: two massive data sets $\mathbf{X}$: $n \times p$ matrix and $\mathbf{Y}$: $n \times q$ matrix due to a large number of observations.

We suppose here that n is very large, but not p nor q.

PLS algorithm mainly based on the SVD of $\mathcal{M}_{h-1} = \mathbf{X}_{h-1}^{\mathbf{T}} \mathbf{Y}_{h-1}$:

Dimension of $\mathcal{M}_{h-1}$: $p \times q$ matrix !!

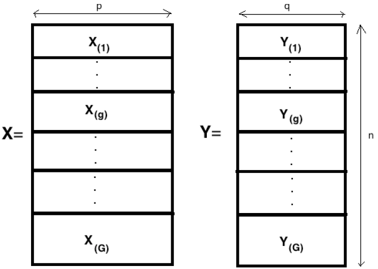
This matrix fits into memory.

But **not X** nor **Y**.

Computation of $\mathcal{M} = \mathbf{X}^\mathsf{T}\mathbf{Y}$ by chunks

$$\mathcal{M} = \mathbf{X}^\mathsf{T}\mathbf{Y} = \sum_{g=1}^{G} \mathbf{X}_{(g)}^\mathsf{T}\mathbf{Y}_{(g)}$$

All terms fit (successively) into memory!



Computation

Computation of $\mathcal{M} = \mathbf{X}^T\mathbf{Y}$ by chunks using R

- No need to load the big matrices **X** and **Y**
- Use memory-mapped files (called “filebacking”) through the bigmemory package to allow matrices to exceed the RAM size.
- A big.matrix is created which supports the use of shared memory for efficiency in parallel computing.
- foreach: package for running in parallel the computation of $\mathcal{M}$ by chunks

Computation

Computation of $\mathcal{M} = \mathbf{X}^T\mathbf{Y}$ by chunks using R

- No need to load the big matrices **X** and **Y**
- Use memory-mapped files (called "filebacking") through the bigmemory package to allow matrices to exceed the RAM size.
- A big.matrix is created which supports the use of shared memory for efficiency in parallel computing.
- foreach: package for running in parallel the computation of $\mathcal{M}$ by chunks

Regularized PLS algorithm:

- Computation of the components ("Scores"):

$$\mathbf{Xu}\ (n \times 1) \text{ and } \mathbf{Yv}\ (n \times 1)$$

- Easy to compute by chunks and store in a big.matrix object.

Concluding Remarks and Take Home Message

- We were able to derive a simple unified algorithm that perfoms standard, sparse, group and sparse group versions of the four classical PLS algorithms (i)–(iv). (And also PLSDA.)
- We used big memory objects, and a simple trick that makes our procedure scalable to big data (large *n*).
- We also parallelized the code for faster computation.
- We have also offered a version of this algorithm for any combination of large values of *n*, *p* and *q*.

sgPLS Available on CRAN

sgsPLS and bigsgPLS Available now on GITHUB:

```r
library(devtools)
install_github("sgsPLS","bigsgPLS", "matt-sutton")
```

References

- Yuan M. and Lin Y. (2006) *Model Selection and Estimation in Regression with Grouped Variables*. **Journal of the Royal Statistical Society: Series B (Statistical Methodology)**, 68 (1), 49–67.
- Simon N., Friedman J., Hastie T. and Tibshirani R. (2013) *A Sparse-group Lasso.* **Journal of Computational and Graphical Statistics**, 22 (2), 231–245.
- Liquet B., Lafaye de Micheaux P., Hejblum B. and Thiebaut R., (2016) *Group and Sparse Group Partial Least Square Approaches Applied in Genomics Context.* **Bioinformatics**, 32(1), 35–42.
- Lafaye de Micheaux P., Liquet B. and Sutton M., *PLS for Big Data: A Unified Parallel Algorithm for Regularized Group PLS.* (Submitted) https://arxiv.org/abs/1702.07066
- M. Sutton, R. Thiebaut, and B. Liquet. (2018) *Sparse group subgroup Partial Least Squares with application to genomics data*. Statistics in Medicine.