Statistical Methods for the Analysis of High-Dimensional and Massive Data using R

Benoit Liquet[1,2], Matthew Sutton[2], Pierre Lafaye de Micheaux[3], Boris Heljbum[4], Rodolphe Thiébaut[4]

[1] University de Pau et Pays de l'Adour, LMAP, E2S-UPPA.
[2] ARC Centre of Excellence for Mathematical and Statistical Frontiers,
[3] UNSW,
[4] Inria, SISTM

Contents

Integrative Analysis

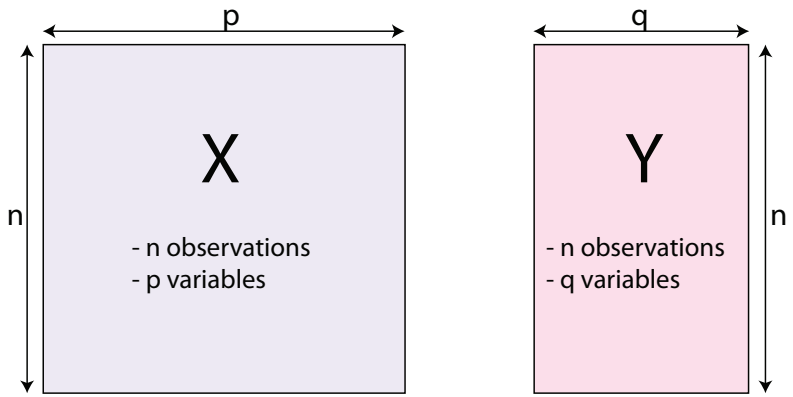
Wikipedia. **Data integration** "involves **combining data** residing in different sources and providing users with a unified view of these data. This process becomes significant in a variety of situations, which include both commercial and **scientific** domains".

System Biology. **Integrative Analysis:** Analysis of heterogeneous types of data from inter-platform technologies.

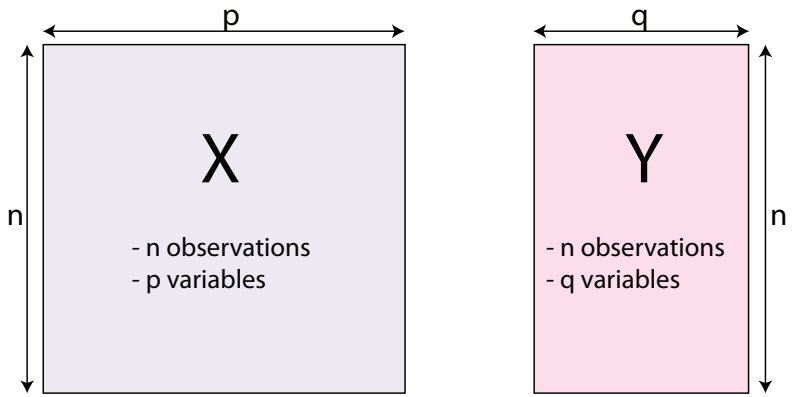
Goal. Combine multiple types of data:

- Contribute to a better understanding of biological mechanisms.
- Have the potential to improve the diagnosis and treatments of complex diseases.

Example: Data definition

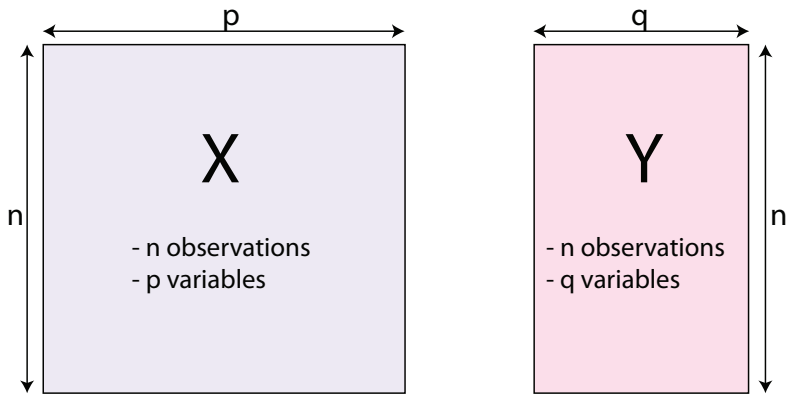


Example: Data definition



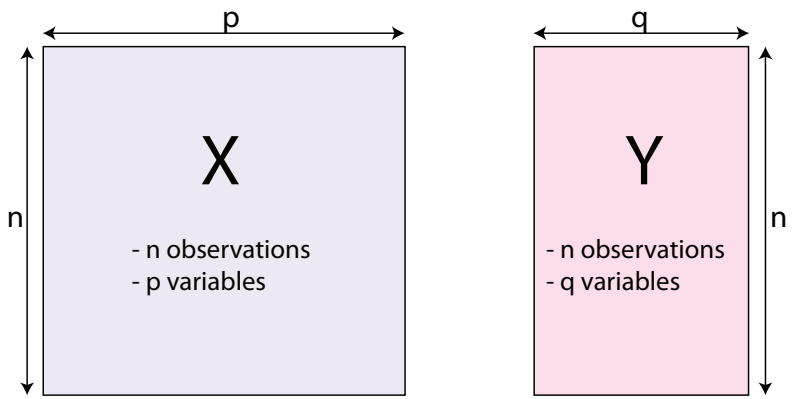
- "Omics." **Y** matrix: gene expression, **X** matrix: SNP (single nucleotide polymorphism). Many others such as proteomic, metabolomic data.

Example: Data definition



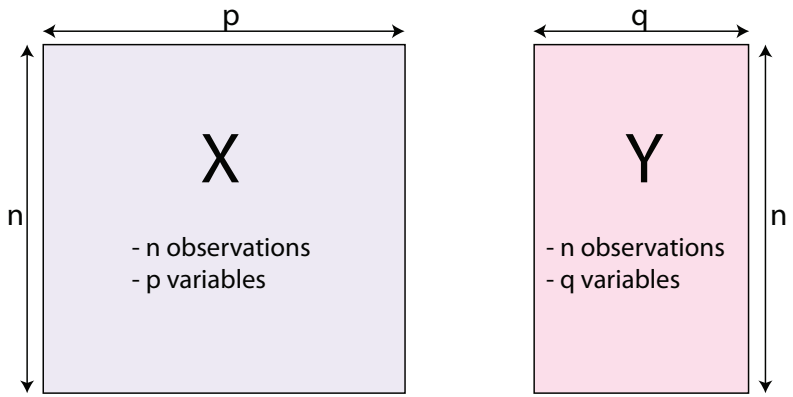
- "Omics." **Y** matrix: gene expression, **X** matrix: SNP (single nucleotide polymorphism). Many others such as proteomic, metabolomic data.
- "Neuroimaging". **Y** matrix: behavioral variables, **X** matrix: brain activity (e.g., EEG, fMRI, NIRS)

Example: Data definition



- "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: DTI (Diffusion Tensor Imaging), **X** matrix: SNP

Example: Data definition



- "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: DTI (Diffusion Tensor Imaging), **X** matrix: SNP
- "Ecology/Environment." **Y** matrix: Water quality variables , **X** matrix: Landscape variables

Data: Constraints and Aims

- Main constraint: colinearity among the variables, or situation with $p > n$ or $q > n$.

Data: Constraints and Aims

- Main constraint: colinearity among the variables, or situation with $p > n$ or $q > n$.
- Two Aims:
  1. Symmetric situation. Analyze the association between two blocks of information. Analysis focused on shared information.

Data: Constraints and Aims

- Main constraint: colinearity among the variables, or situation with $p > n$ or $q > n$.
- Two 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.

Data: Constraints and Aims

- Main constraint: colinearity among the variables, or situation with $p > n$ or $q > n$.
- Two 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

Data: Constraints and Aims

- Main constraint: colinearity among the variables, or situation with $p > n$ or $q > n$.
- Two 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 $\boldsymbol{\xi} = \boldsymbol{Xu}$, $\omega = \boldsymbol{Yv}$ with maximal covariance.

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

  - 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).

PLS and sparse PLS

Classical PLS

- Output of PLS: H pairs of latent variables (ξ_h, ω_h), $h = 1, \ldots, H$.
- Reduction method ($H << min(p, q)$). But no variable selection for extracting the most relevant (original) variables from each latent variable.

PLS and sparse PLS

Classical PLS

- Output of PLS: H pairs of latent variables (ξ_h, ω_h), $h = 1, \ldots, H$.
- Reduction method ($H << min(p, q)$). But no variable selection for extracting the most relevant (original) variables from each latent variable.

sparse PLS

- sparse PLS selects the relevant SNPs
- Some coefficients u_ℓ are equal to 0

$$\xi_h = u_1 \times SNP_1 + \underbrace{u_2}_{=0} \times SNP_2 + \underbrace{u_3}_{=0} \times SNP_3 + \cdots + u_p \times SNP_p$$

- The sPLS components are linear combinations of the selected variables

Group structures within the data

- Natural example: Categorical variables form a group of dummy variables in a regression setting.

Group structures within the data

- Natural example: Categorical variables form a group of dummy variables in a regression setting.
- 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

- Natural example: Categorical variables form a group of dummy variables in a regression setting.
- 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 that variables are divided into groups:

- Example: p SNPs grouped into K genes ($X_j = \mathrm{SNP}_j$)

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

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

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

Group PLS

Aim: select groups of variables taking into account the data structure

Group PLS

Aim: select groups of variables taking into account the data structure

- PLS components
  $\xi_h = u_1 \times X_1 + u_2 \times X_2 + u_3 \times X_3 + \cdots + u_p \times X_p$
- sparse PLS components (sPLS)
  $\xi_h = u_1 \times X_1 + \underbrace{u_2}_{=0} \times X_2 + \underbrace{u_3}_{=0} \times X_3 + \cdots + u_p \times X_p$

Group PLS

Aim: select groups of variables taking into account the data structure

- PLS components
  $\xi_h = u_1 \times X_1 + u_2 \times X_2 + u_3 \times X_3 + \cdots + u_p \times X_p$
- sparse PLS components (sPLS)
  $\xi_h = u_1 \times X_1 + \underbrace{u_2}_{=0} \times X_2 + \underbrace{u_3}_{=0} \times X_3 + \cdots + u_p \times X_p$
- group PLS components (gPLS)

$$\xi_h = \overbrace{\underbrace{u_1}_{=0} X_1 + \underbrace{u_2}_{=0} X_2}^{module_1} + \overbrace{\underbrace{u_3}_{\neq 0} X_3 + \underbrace{u_4}_{\neq 0} X_1 + \underbrace{u_5}_{\neq 0} X_5}^{module_2} + \cdots + \overbrace{\underbrace{u_{p-1}}_{=0} X_{p-1} + \underbrace{u_p}_{=0} X_p}^{module_K}$$

$\hookrightarrow$ select groups of variables; either all the variables within a group are selected or none of them are selected

Group PLS

Aim: select groups of variables taking into account the data structure

- PLS components
  $\xi_h = u_1 \times X_1 + u_2 \times X_2 + u_3 \times X_3 + \cdots + u_p \times X_p$
- sparse PLS components (sPLS)
  $\xi_h = u_1 \times X_1 + \underbrace{u_2}_{=0} \times X_2 + \underbrace{u_3}_{=0} \times X_3 + \cdots + u_p \times X_p$
- group PLS components (gPLS)

$$\xi_h = \overbrace{\underbrace{u_1}_{=0} X_1 + \underbrace{u_2}_{=0} X_2}^{module_1} + \overbrace{\underbrace{u_3}_{\neq 0} X_3 + \underbrace{u_4}_{\neq 0} X_1 + \underbrace{u_5}_{\neq 0} X_5}^{module_2} + \cdots + \overbrace{\underbrace{u_{p-1}}_{=0} X_{p-1} + \underbrace{u_p}_{=0} X_p}^{module_K}$$

  $\hookrightarrow$ select groups of variables; either all the variables within a group are selected or none of them are selected

... does not achieve sparsity within each group ...

Sparse Group PLS

Aim: combine both sparsity of groups and within each group.
Example: $\boldsymbol{X}$ matrix = genes. We might be interested in identifying particularly important genes in pathways of interest.

- sparse PLS components (sPLS)

$$\xi_h = u_1 \times X_1 + \underbrace{u_2}_{=0} \times X_2 + \underbrace{u_3}_{=0} \times X_3 + \cdots + u_p \times X_p$$

- group PLS components (gPLS)

$$\xi_h = \overbrace{\underbrace{u_1}_{=0} X_1 + \underbrace{u_2}_{=0} X_2}^{module_1} + \overbrace{\underbrace{u_3}_{\neq 0} X_3 + \underbrace{u_4}_{\neq 0} X_1 + \underbrace{u_5}_{\neq 0} X_5}^{module_2} + \cdots + \overbrace{\underbrace{u_{p-1}}_{=0} X_{p-1} + \underbrace{u_p}_{=0} X_p}^{module_K}$$

Sparse Group PLS

Aim: combine both sparsity of groups and within each group.
Example: $\boldsymbol{X}$ matrix = genes. We might be interested in identifying particularly important genes in pathways of interest.

- sparse PLS components (sPLS)

$$\xi_h = u_1 \times X_1 + \underbrace{u_2}_{=0} \times X_2 + \underbrace{u_3}_{=0} \times X_3 + \cdots + u_p \times X_p$$

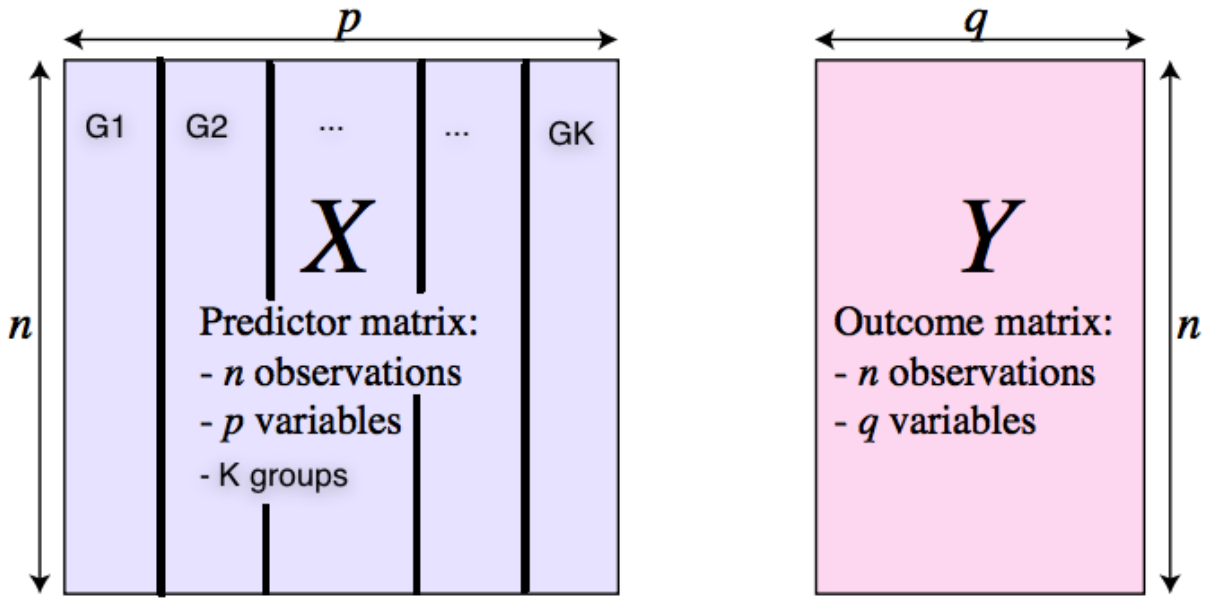
- group PLS components (gPLS)

$$\xi_h = \overbrace{\underbrace{u_1}_{=0} X_1 + \underbrace{u_2}_{=0} X_2}^{module_1} + \overbrace{\underbrace{u_3}_{\neq 0} X_3 + \underbrace{u_4}_{\neq 0} X_1 + \underbrace{u_5}_{\neq 0} X_5}^{module_2} + \cdots + \overbrace{\underbrace{u_{p-1}}_{=0} X_{p-1} + \underbrace{u_p}_{=0} X_p}^{module_K}$$

- sparse group PLS components (sgPLS)

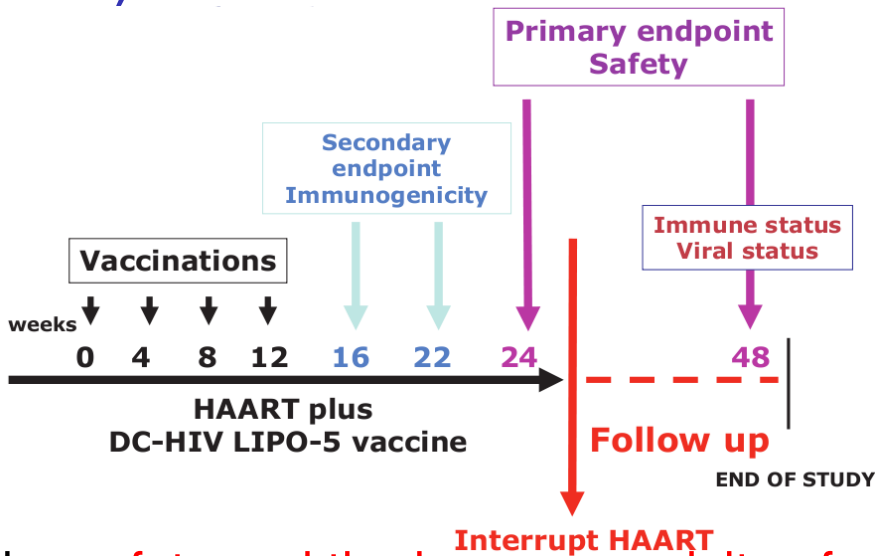
$$\xi_h = \overbrace{\underbrace{u_1}_{=0} X_1 + \underbrace{u_2}_{=0} X_2}^{module_1} + \overbrace{\underbrace{u_3}_{\neq 0} X_3 + \underbrace{u_4}_{=0} X_4 + \underbrace{u_5}_{=0} X_5}^{module_2} + \cdots + \overbrace{\underbrace{u_{p-1}}_{=0} X_{p-1} + \underbrace{u_p}_{=0} X_p}^{module_K}$$

Aims in a regression setting



- Select groups of variables taking into account the data structure; 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

Illustration: Dendritic Cells in Addition to Antiretroviral Treatment (DALIA) trial



- Evaluation of the safety and the immunogenicity of a vaccine on $n = 19$ HIV-1 infected patients.
- The vaccine was injected on weeks 0, 4, 8 and 12 while patients received an antiretroviral therapy. An interruption of the antiretrovirals was performed at week 24.
- After vaccination, a deep evaluation of the immune response was performed at week 16.
- Repeated measurements of the main immune markers and gene expression were performed every 4 weeks until the end of the trials.

DALIA trial: Question ?

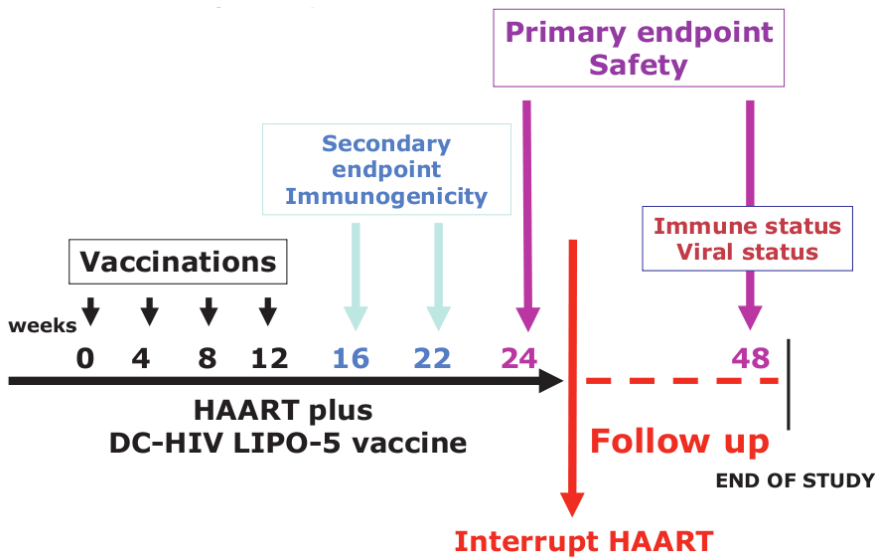
First results obtained using group of genes

- Significant change of gene expression among 69 modules over time before antiretroviral treatment interruption.

DALIA trial: Question ?

First results obtained using group of genes

- Significant change of gene expression among 69 modules over time before antiretroviral treatment interruption.
- How does the gene abundance of these 69 modules as measured at week 16 **correlate** with immune markers measured at week 16?



sPLS, gPLS and sgPLS

- Response variables **Y**= immune markers composed of $q = 7$ cytokines (IL21, IL2, IL13, IFNg, Luminex score, TH1 score, CD4).
- Predictor variables **X**= expression of $p = 5399$ genes extracted from the 69 modules.
- Use the structure of the data (modules) for gPLS and sgPLS. Each gene belongs to one of the 69 modules.
- Asymmetric situation.

Results: Modules and number of genes selected

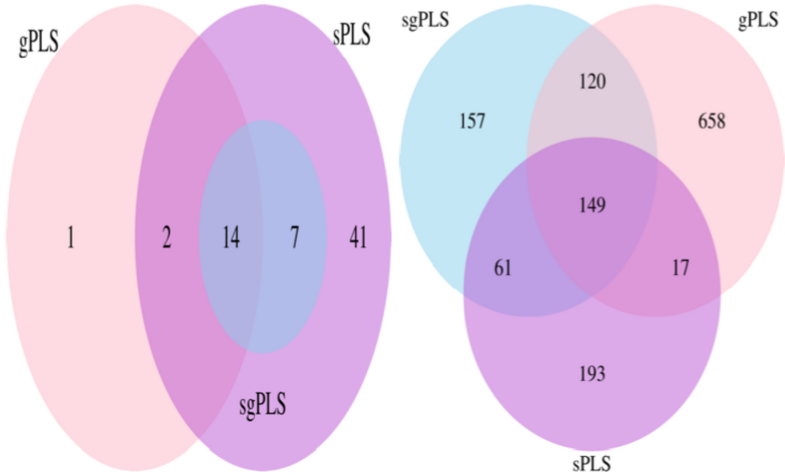
		gPLS			sgPLS			sPLS		
	size	comp1	comp2	comp3	comp1	comp2	comp3	comp1	comp2	comp3
M1.1	79	79	0	0	19	0	0	8	2	1
M3.2	126	126	0	0	41	0	0	22	0	0
M3.5	131	0	0	0	11	24	0	7	7	1
M3.6	42	42	0	0	15	0	0	6	0	0
M4.1	60	0	0	0	6	0	0	4	0	0
M4.13	72	72	0	0	26	0	0	11	0	0
M4.15	41	41	0	0	15	0	0	10	0	1
M4.2	43	43	0	0	14	0	0	7	1	1
M4.6	104	104	0	0	28	0	0	16	2	0
M5.1	214	0	0	0	46	0	0	21	2	4
M5.14	54	54	0	0	13	0	0	7	0	2
M5.15	24	24	24	0	20	0	0	18	0	0
M5.7	119	0	0	0	18	0	40	8	0	2
M6.13	38	38	0	0	10	0	0	7	0	0
M6.6	40	40	0	0	19	0	0	11	0	0
M7.1	150	150	0	0	37	0	0	19	2	2
M7.27	29	29	0	0	8	0	0	3	0	1
M4.7	82	0	0	0	0	20	0	5	7	0
M6.7	62	0	0	0	0	23	0	3	4	1
M8.59	13	0	13	0	0	4	0	0	3	0
M5.2	65	0	0	0	0	0	32	0	1	0
M4.8	53	53	0	0	0	0	0	1	0	0
M7.35	19	19	0	0	0	0	0	1	1	0
M4.11	17	0	0	17	0	0	0	0	0	0

$p = 5399$; 24 modules selected by gPLS or sgPLS on 3 scores

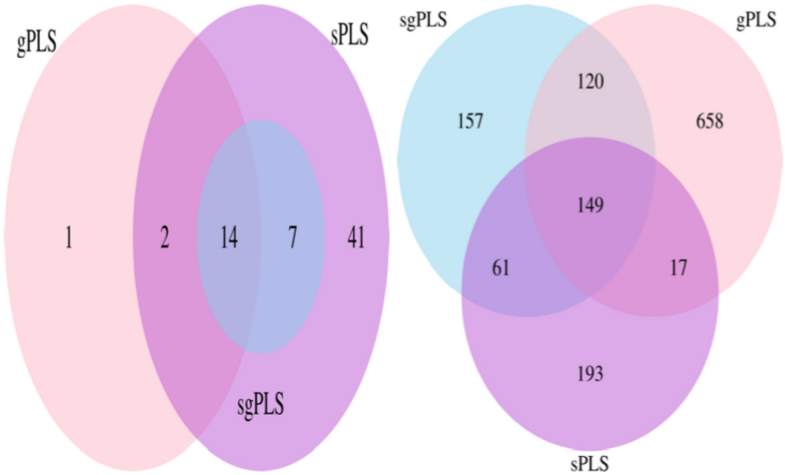
Results: Modules and number of genes selected

		gPLS			sgPLS			sPLS		
	size	comp1	comp2	comp3	comp1	comp2	comp3	comp1	comp2	comp3
M1.1	79	79	0	0	19	0	0	8	2	1
M3.2	126	126	0	0	41	0	0	22	0	0
M3.5	131	0	0	0	11	24	0	7	7	1
M3.6	42	42	0	0	15	0	0	6	0	0
M4.1	60	0	0	0	6	0	0	4	0	0
M4.13	72	72	0	0	26	0	0	11	0	0
M4.15	41	41	0	0	15	0	0	10	0	1
M4.2	43	43	0	0	14	0	0	7	1	1
M4.6	104	104	0	0	28	0	0	16	2	0
M5.1	214	0	0	0	46	0	0	21	2	4
M5.14	54	54	0	0	13	0	0	7	0	2
M5.15	24	24	24	0	20	0	0	18	0	0
M5.7	119	0	0	0	18	0	40	8	0	2
M6.13	38	38	0	0	10	0	0	7	0	0
M6.6	40	40	0	0	19	0	0	11	0	0
M7.1	150	150	0	0	37	0	0	19	2	2
M7.27	29	29	0	0	8	0	0	3	0	1
M4.7	82	0	0	0	0	20	0	5	7	0
M6.7	62	0	0	0	0	23	0	3	4	1
M8.59	13	0	13	0	0	4	0	0	3	0
M5.2	65	0	0	0	0	0	32	0	1	0
M4.8	53	53	0	0	0	0	0	1	0	0
M7.35	19	19	0	0	0	0	0	1	1	0
M4.11	17	0	0	17	0	0	0	0	0	0
M2.1	105	0	0	0	0	0	0	1	0	0
M3.1	74	0	0	0	0	0	0	1	0	0
M4.12	87	0	0	0	0	0	0	1	0	1
M4.16	79	0	0	0	0	0	0	2	0	1
M4.9	87	0	0	0	0	0	0	4	1	1
M5.10	196	0	0	0	0	0	0	3	3	0
M5.11	59	0	0	0	0	0	0	3	2	0
M5.13	147	0	0	0	0	0	0	1	2	4
M5.3	91	0	0	0	0	0	0	3	1	0
M5.4	115	0	0	0	0	0	0	3	2	2
M5.5	211	0	0	0	0	0	0	12	4	0
M5.6	126	0	0	0	0	0	0	3	2	1
M5.8	97	0	0	0	0	0	0	4	1	0
M5.9	72	0	0	0	0	0	0	4	0	0
M6.10	67	0	0	0	0	0	0	4	0	0
M6.14	33	0	0	0	0	0	0	3	0	0
M6.2	121	0	0	0	0	0	0	2	2	1
M6.20	42	0	0	0	0	0	0	1	2	0
M6.4	82	0	0	0	0	0	0	3	2	0
M6.9	35	0	0	0	0	0	0	2	1	0
M7.11	104	0	0	0	0	0	0	2	2	1
M7.12	108	0	0	0	0	0	0	4	0	0
M7.14	48	0	0	0	0	0	0	4	1	0
M7.15	78	0	0	0	0	0	0	2	0	1
M7.16	56	0	0	0	0	0	0	1	2	1
M7.2	93	0	0	0	0	0	0	4	1	0
M7.21	76	0	0	0	0	0	0	3	0	0
M7.24	65	0	0	0	0	0	0	2	0	0
M7.25	93	0	0	0	0	0	0	3	2	3
M7.26	63	0	0	0	0	0	0	2	0	0
M7.4	109	0	0	0	0	0	0	4	2	0
M7.5	132	0	0	0	0	0	0	6	5	2
M7.6	94	0	0	0	0	0	0	2	3	1
M7.8	85	0	0	0	0	0	0	3	0	0
M8.13	27	0	0	0	0	0	0	1	0	0
M8.14	27	0	0	0	0	0	0	2	1	0
M7.33	49	0	0	0	0	0	0	0	1	0
M7.7	89	0	0	0	0	0	0	0	3	1
M4.14	55	0	0	0	0	0	0	0	0	1
M4.4	68	0	0	0	0	0	0	0	0	1
M4.5	74	0	0	0	0	0	0	0	0	1

Results: Venn diagram

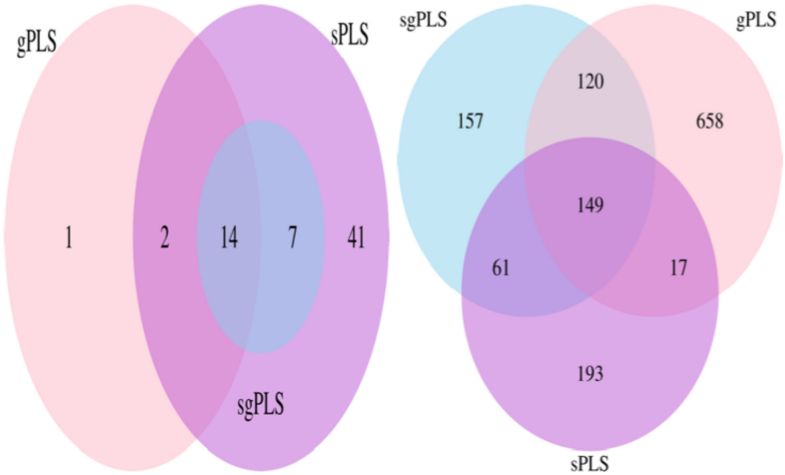


Results: Venn diagram



- sgPLS selects slightly more genes than sPLS (respectively 487 and 420 genes selected)
- But sgPLS selects fewer modules than sPLS (respectively 21 and 64 groups of genes selected)
- Note: all the 21 groups of genes selected by sgPLS were included in those selected by sPLS.
- sgPLS selects slightly more modules than gPLS (4 more, 14/21 in common).

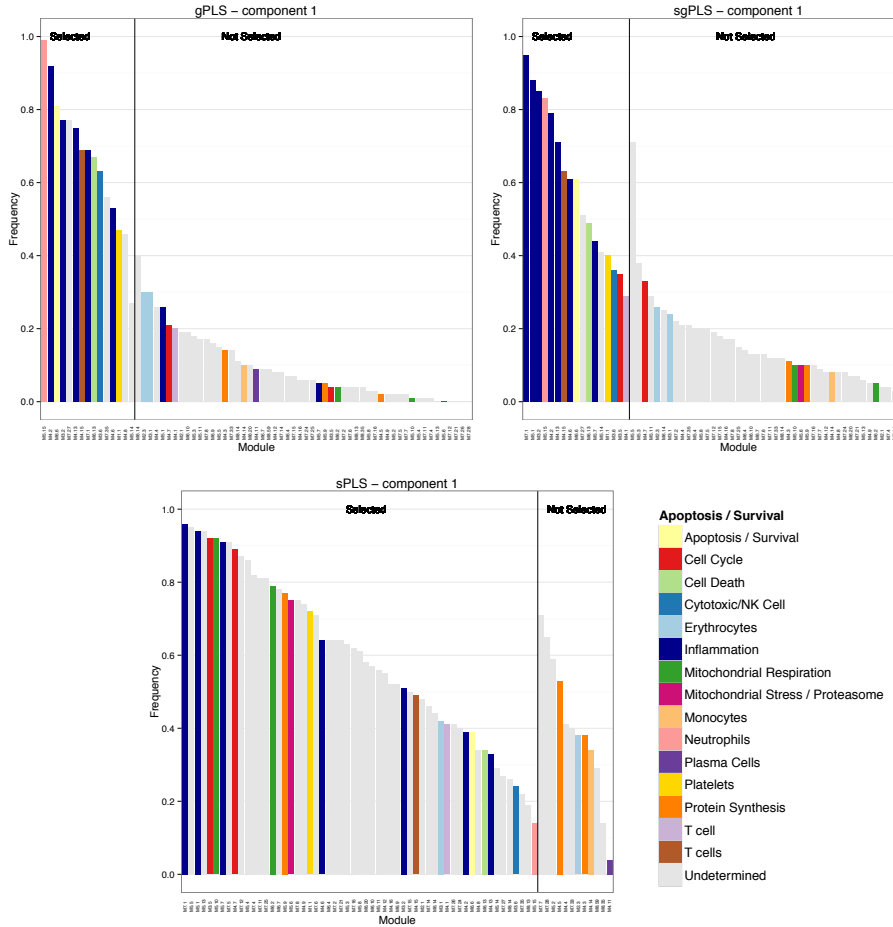
Results: Venn diagram



- sgPLS selects slightly more genes than sPLS (respectively 487 and 420 genes selected)
- But sgPLS selects fewer modules than sPLS (respectively 21 and 64 groups of genes selected)
- Note: all the 21 groups of genes selected by sgPLS were included in those selected by sPLS.
- sgPLS selects slightly more modules than gPLS (4 more, 14/21 in common).
- However, gPLS leads to more genes selected than sgPLS (944)
- In this application, the sgPLS approach led to a parsimonious selection of modules and genes that sound very relevant biologically

Chaussabel's functional modules: http://www.biir.net/public_wikis/module_annotation/V2_Trial_8_Modules

Stability of the variable selection (100 bootstrap samples)



Stability of the variable selection assessed on 100 bootstrap samples on DALIA-1 trial data, for the gPLS, sgPLS and sPLS procedures respectively. For each procedure, the modules selected on the original sample are separated from those that were not.

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 1

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:

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

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:

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

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 $\boldsymbol{\mathcal{M}}_{h-1} = \mathbf{X}_{h-1}^{\mathsf{T}}\mathbf{Y}_{h-1}$:

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

Connexion between SVD and maximum covariance

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

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

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 $\boldsymbol{\mathcal{M}}_{h-1} = \mathbf{X}_{h-1}^{\mathsf{T}}\mathbf{Y}_{h-1}$:

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

Why is this useful ?

SVD properties

Theorem 2

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

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

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

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

SVD properties

Thus, solving

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

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:

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

  $\hookrightarrow$ 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:

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

  $\hookrightarrow$ 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

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

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).

Sparse PLS models

For cases (i)–(iv),

- Aim: obtaining sparse weight vectors $\boldsymbol{u}_h$ and $\boldsymbol{v}_h$.
- Associated component scores (i.e., latent variables) $\boldsymbol{\xi}_h := \mathbf{X}_{h-1}\boldsymbol{u}_h$ and $\boldsymbol{\omega}_h := \mathbf{Y}_{h-1}\boldsymbol{v}_h$, $h = 1, \ldots, H$, for a small number of components.
- Recursive procedure with objective function involving $\mathbf{X}_{h-1}$ and $\mathbf{Y}_{h-1}$
  $\hookrightarrow$ decomposition (approximation) of the original matrices $\mathbf{X}$ and $\mathbf{Y}$:

$$\mathbf{X} = \boldsymbol{\Xi}_H \mathcal{C}_H^{\mathsf{T}} + \mathcal{F}_{X,H}, \qquad \mathbf{Y} = \boldsymbol{\Omega}_H \mathcal{D}_H^{\mathsf{T}} + \mathcal{F}_{Y,H}, \tag{3}$$

  where $\boldsymbol{\Xi} = (\boldsymbol{\xi}_h)$ and $\boldsymbol{\Omega} = (\boldsymbol{\omega}_h)$.
- For the regression mode, we have the multivariate linear regression model

$$\mathbf{Y} = \mathbf{X}\widehat{\mathcal{B}}_{PLS} + \mathcal{E},$$

  with $\widehat{\mathcal{B}}_{PLS} = \mathcal{U}_H(\mathcal{C}_H^{\mathsf{T}}\mathcal{U}_H)^{-1}\mathcal{P}_H\mathcal{D}_H^{\mathsf{T}}$ and $\mathcal{E}$ is a matrix of residuals.

The algorithm

Main steps of the iterative algorithm

1. $\mathbf{X}_0 = \mathbf{X},\ \mathbf{Y}_0 = \mathbf{Y} \quad h = 1$
2. $\mathcal{M}_{h-1} := \mathbf{X}_{h-1}^{\mathsf{T}} \mathbf{Y}_{h-1}$.
3. SVD: extraction of the first pair of singular vectors $\boldsymbol{u}_h$ and $\boldsymbol{v}_h$.
4. Sparsity step. Produces sparse weights $\boldsymbol{u}_{\text{sparse}}$ and $\boldsymbol{v}_{\text{sparse}}$.
5. Latent variables: $\boldsymbol{\xi}_h = \mathbf{X}_{h-1}\boldsymbol{u}_{\text{sparse}}$ and $\omega_h = \mathbf{Y}_{h-1}\boldsymbol{v}_{\text{sparse}}$
6. Slope coefficients:
   - $\mathbf{c}_h = \mathbf{X}_{h-1}^{\mathsf{T}}\boldsymbol{\xi}_h / \boldsymbol{\xi}_h^{\mathsf{T}}\boldsymbol{\xi}_h$ for both modes
   - $\mathbf{d}_h = \mathbf{Y}_{h-1}^{\mathsf{T}}\boldsymbol{\xi}_h / \boldsymbol{\xi}_h^{\mathsf{T}}\boldsymbol{\xi}_h$ for "PLSR regression mode"
   - $\mathbf{e}_h = \mathbf{Y}_{h-1}^{\mathsf{T}}\omega_h / \omega_h^{\mathsf{T}}\omega_h$ for "PLS mode A"
7. Deflation:
   - $\mathbf{X}_h = \mathbf{X}_{h-1} - \boldsymbol{\xi}_h \boldsymbol{c}_h^{\mathsf{T}}$ for both modes
   - $\mathbf{Y}_h = \mathbf{Y}_{h-1} - \boldsymbol{\xi}_h \boldsymbol{d}_h^{\mathsf{T}}$ for "PLSR regression mode"
   - $\mathbf{Y}_h = \mathbf{Y}_{h-1} - \omega_h \mathbf{e}_h^{\mathsf{T}}$ for "PLS mode A"
8. If $h = H$ stop, else $h = h + 1$ and goto step 2.

sparse PLS (sPLS)

In sPLS, the optimisation problem to solve is

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

- $\|\boldsymbol{\mathcal{M}}_h - \boldsymbol{u}_h \boldsymbol{v}_h^{\mathsf{T}}\|_F^2 = \sum_{i=1}^{p} \sum_{j=1}^{q} (m_{ij} - u_{ih} v_{jh})^2$,
- $\boldsymbol{\mathcal{M}}_h = \mathbf{X}_h^{\mathsf{T}} \mathbf{Y}_h$ for each iteration h.
- $P_{\lambda_{1,h}}(\boldsymbol{u}_h) = \sum_{i=1}^{p} 2\lambda_1^h |u_i|$ and $P_{\lambda_{2,h}}(\boldsymbol{v}_h) = \sum_{j=1}^{q} 2\lambda_2^h |v_i|$

sparse PLS (sPLS)

In sPLS, the optimisation problem to solve is

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

- $\|\boldsymbol{\mathcal{M}}_h - \boldsymbol{u}_h \boldsymbol{v}_h^{\mathsf{T}}\|_F^2 = \sum_{i=1}^{p} \sum_{j=1}^{q} (m_{ij} - u_{ih} v_{jh})^2$,
- $\boldsymbol{\mathcal{M}}_h = \mathbf{X}_h^{\mathsf{T}} \mathbf{Y}_h$ for each iteration h.
- $P_{\lambda_{1,h}}(\boldsymbol{u}_h) = \sum_{i=1}^{p} 2\lambda_1^h |u_i|$ and $P_{\lambda_{2,h}}(\boldsymbol{v}_h) = \sum_{j=1}^{q} 2\lambda_2^h |v_i|$

Iterative solution. Applying the thresholding function $g^{\text{soft}}(x, \lambda) = \text{sign}(x)(|x| - \lambda)_+$

- to the vector $\boldsymbol{\mathcal{M}} \boldsymbol{v}_h$ componentwise to get $\boldsymbol{u}_h$.
- to the vector $\boldsymbol{\mathcal{M}}^{\mathsf{T}} \boldsymbol{u}_h$ componentwise to get $\boldsymbol{v}_h$.

group PLS (gPLS)

- $\mathbf{X}$ and $\mathbf{Y}$ can be divided respectively into K and L sub-matrices (groups) $\mathbf{X}^{(k)}$: $n \times p_k$ and $\mathbf{Y}^{(l)}$: $n \times q_l$.
- Same idea as Yuan and Lin (2006), we use group lasso penalties:

$$P_{\lambda_1}(\boldsymbol{u}) = \lambda_1 \sum_{k=1}^{K} \sqrt{p_k} \|\boldsymbol{u}^{(k)}\|_2 \quad \text{and} \quad P_{\lambda_2}(\boldsymbol{v}) = \lambda_2 \sum_{l=1}^{L} \sqrt{q_l} \|\boldsymbol{v}^{(l)}\|_2,$$

where $\boldsymbol{u}^{(k)}$ (resp. $\boldsymbol{v}^{(l)}$) is the weight vector associated to the k-th (resp. l-th) block.

In gPLS, the optimisation problem to solve is

$$\sum_{k=1}^{K} \sum_{l=1}^{L} \left\| \boldsymbol{\mathcal{M}}^{(k,l)} - \boldsymbol{u}^{(k)} \boldsymbol{v}^{(l)^{\mathsf{T}}} \right\|_F^2 + P_{\lambda_1}(\boldsymbol{u}) + P_{\lambda_2}(\boldsymbol{v}),$$

- $\boldsymbol{\mathcal{M}}^{(k,l)} = \mathbf{X}^{(k)} \mathbf{Y}^{(l)^{\mathsf{T}}}$.

Remark if the k-th block is composed by only one variable then

$$\|\boldsymbol{u}^{(k)}\|_2 = \sqrt{(u^{(k)})^2} = |u^{(k)}|.$$

group PLS (gPLS)

Theorem 3

Solution of the group PLS optimisation problem is given by:

$$\boldsymbol{u}^{(k)} = \left(1 - \frac{\lambda_1}{2} \frac{\sqrt{p_k}}{\|\boldsymbol{\mathcal{M}}^{(k,\cdot)}\boldsymbol{v}\|_2}\right)_+ \boldsymbol{\mathcal{M}}^{(k,\cdot)}\boldsymbol{v} \qquad \text{(for fixed } \boldsymbol{v}\text{)}$$

and

$$\boldsymbol{v}^{(l)} = \left(1 - \frac{\lambda_2}{2} \frac{\sqrt{q_l}}{\|\boldsymbol{\mathcal{M}}^{(\cdot,l)^\intercal}\boldsymbol{u}\|_2}\right)_+ \boldsymbol{\mathcal{M}}^{(\cdot,l)^\intercal}\boldsymbol{u} \qquad \text{(for fixed } \boldsymbol{u}\text{)}.$$

Note: we will iterate until convergence of $\boldsymbol{u}^{(k)}$ and $\boldsymbol{v}^{(l)}$, using alternatively one of the above formulas.

sparse group PLS: sparsity within groups

- Following Simon et al. (2013), we introduce sparse group lasso penalties:

$$
\begin{aligned}
P_{\lambda_1}(\boldsymbol{u}) &= (1-\alpha_1)\lambda_1 \sum_{k=1}^{K} \sqrt{p_k}\|\boldsymbol{u}^{(k)}\|_2 + \alpha_1\lambda_1\|\boldsymbol{u}\|_1, \\
P_{\lambda_2}(\boldsymbol{v}) &= (1-\alpha_2)\lambda_2 \sum_{l=1}^{L} \sqrt{q_l}\|\boldsymbol{v}^{(l)}\|_2 + \alpha_2\lambda_2\|\boldsymbol{v}\|_1.
\end{aligned}
$$

sparse group PLS (sgPLS)

Theorem 4

Solution of the sparse group PLS optimisation problem is given by:
$\boldsymbol{u}^{(k)} = 0$ if

$$\left\| g^{\text{soft}}\left(\boldsymbol{\mathcal{M}}^{(k,\cdot)}\boldsymbol{v}, \lambda_1\alpha_1/2\right)\right\|_2 \leqslant \lambda_1(1-\alpha_1)\sqrt{p_k},$$

otherwise

$$\boldsymbol{u}^{(k)} = \frac{1}{2}\left[g^{\text{soft}}\left(\boldsymbol{\mathcal{M}}^{(k,\cdot)}\boldsymbol{v}, \lambda_1\alpha_1/2\right) - \lambda_1(1-\alpha_1)\sqrt{p_k}\,\frac{g^{\text{soft}}\left(\boldsymbol{\mathcal{M}}^{(k,\cdot)}\boldsymbol{v}, \lambda_1\alpha_1/2\right)}{\|g^{\text{soft}}\left(\boldsymbol{\mathcal{M}}^{(k,\cdot)}\boldsymbol{v}, \lambda_1\alpha_1/2\right)\|_2}\right].$$

We have $\boldsymbol{v}^{(l)} = 0$ if

$$\left\| g^{\text{soft}}\left(\boldsymbol{\mathcal{M}}^{(\cdot,l)^{\mathsf{T}}}\boldsymbol{u}, \lambda_2\alpha_2/2\right)\right\|_2 \leqslant \lambda_2(1-\alpha_2)\sqrt{q_l}$$

and

$$\boldsymbol{v}^{(l)} = \frac{1}{2}\left[g^{\text{soft}}\left(\boldsymbol{\mathcal{M}}^{(\cdot,l)^{\mathsf{T}}}\boldsymbol{u}, \lambda_1\alpha_1/2\right) - \lambda_2(1-\alpha_2)\sqrt{q_l}\,\frac{g^{\text{soft}}\left(\boldsymbol{\mathcal{M}}^{(\cdot,l)^{\mathsf{T}}}\boldsymbol{u}, \lambda_2\alpha_2/2\right)}{\|g^{\text{soft}}\left(\boldsymbol{\mathcal{M}}^{(\cdot,l)^{\mathsf{T}}}\boldsymbol{u}, \lambda_2\alpha_2/2\right)\|_2}\right]$$

otherwise.

Similar proof (see our paper in Bioinformatics, 2016).

R package: sgPLS

- `sgPLS` package implements sPLS, gPLS and sgPLS methods: `http://cran.r-project.org/web/packages/sgPLS/index.html`
- Includes some functions for choosing the tuning parameters related to the predictor matrix for different sparse PLS model (regression mode).
- Some simple code to perform a sgPLS:

```
model.sgPLS <- sgPLS(X, Y, ncomp = 2, mode = "regression",
                     keepX = c(4, 4), keepY = c(4, 4),
                     ind.block.x = ind.block.x ,
                     ind.block.y = ind.block.y,
                     alpha.x = c(0.5, 0.5),
                     alpha.y = c(0.5, 0.5))
```

- Last version also includes sparse group Discriminant Analysis.

Extension of sparse group PLS

Taking into account one more layer in the group structure:

- Example: SNP $\subset$ Gene $\subset$ Pathways

Extension of sparse group PLS

Taking into account one more layer in the group structure:

- Example: SNP $\subset$ Gene $\subset$ Pathways
- Longitudinal study

Group structures within the data}

- Gene Module: genes within the same pathway have similar functions and act together to regulate the biological system.

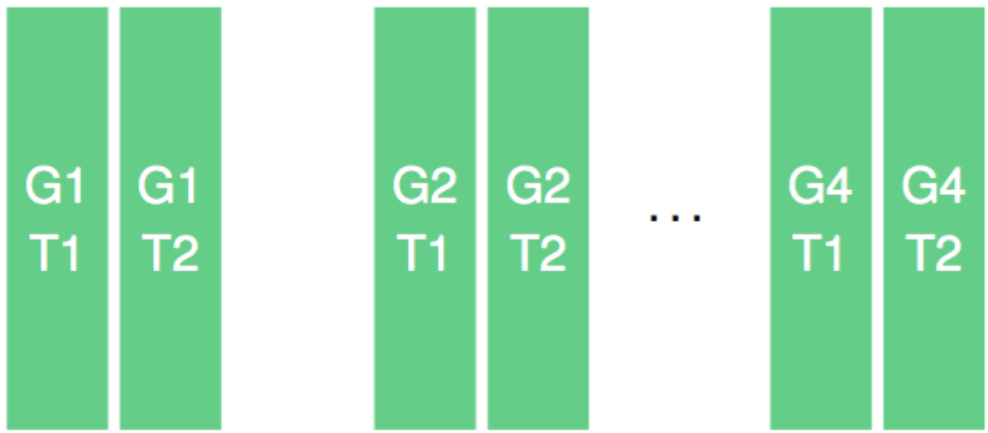
$$\mathbf{X} = [\underbrace{\text{gene}_1, \ldots, \text{gene}_k}_{G1} \mid \cdots \mid \underbrace{\text{gene}_{l+1}, \ldots, \text{gene}_p}_{G4}]$$

Longitudinal group structures:

- Time index: genes within the same pathway at the same time index have similar functions in regulating a biological system.2

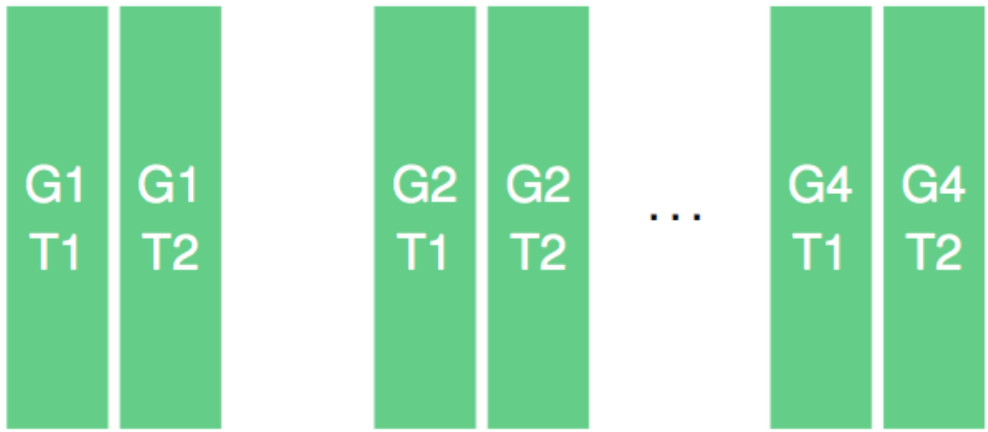
$$\mathbf{G1} = [\underbrace{\text{gene}_1, \ldots, \text{gene}_k}_{G1T1} \mid \underbrace{\text{gene}_1, \ldots, \text{gene}_k}_{G1T2}]$$

Longitudinal group structures:



$$\mathbf{X} = [\underbrace{G1T1, G1T2}_{G1} \mid \underbrace{G2T1, G2T2}_{G2} \mid \cdots \mid \underbrace{G4T1, G4T2}_{G4}]$$

Aims:



- Identify important modules at a group level, important times at a subgroup level and single genes at an individual level.

Sparse Models: sgsPLS

sparse group subgroup PLS

$$\xi = \overbrace{\underbrace{\overbrace{0 \times X_1 + 0 \times X_2}^{Time\ 1} + \overbrace{0 \times X_1 + 0 \times X_2}^{Time\ 2}}_{Module\ 1} + \cdots}$$

$$+ \overbrace{\underbrace{\overbrace{u_{p-1} \times X_{p-1} + 0 \times X_p}^{Time\ 1} + \overbrace{0 \times X_{p-1} + 0 \times X_p}^{Time\ 2}}_{Module\ k}}$$

Sparse Models: sgsPLS

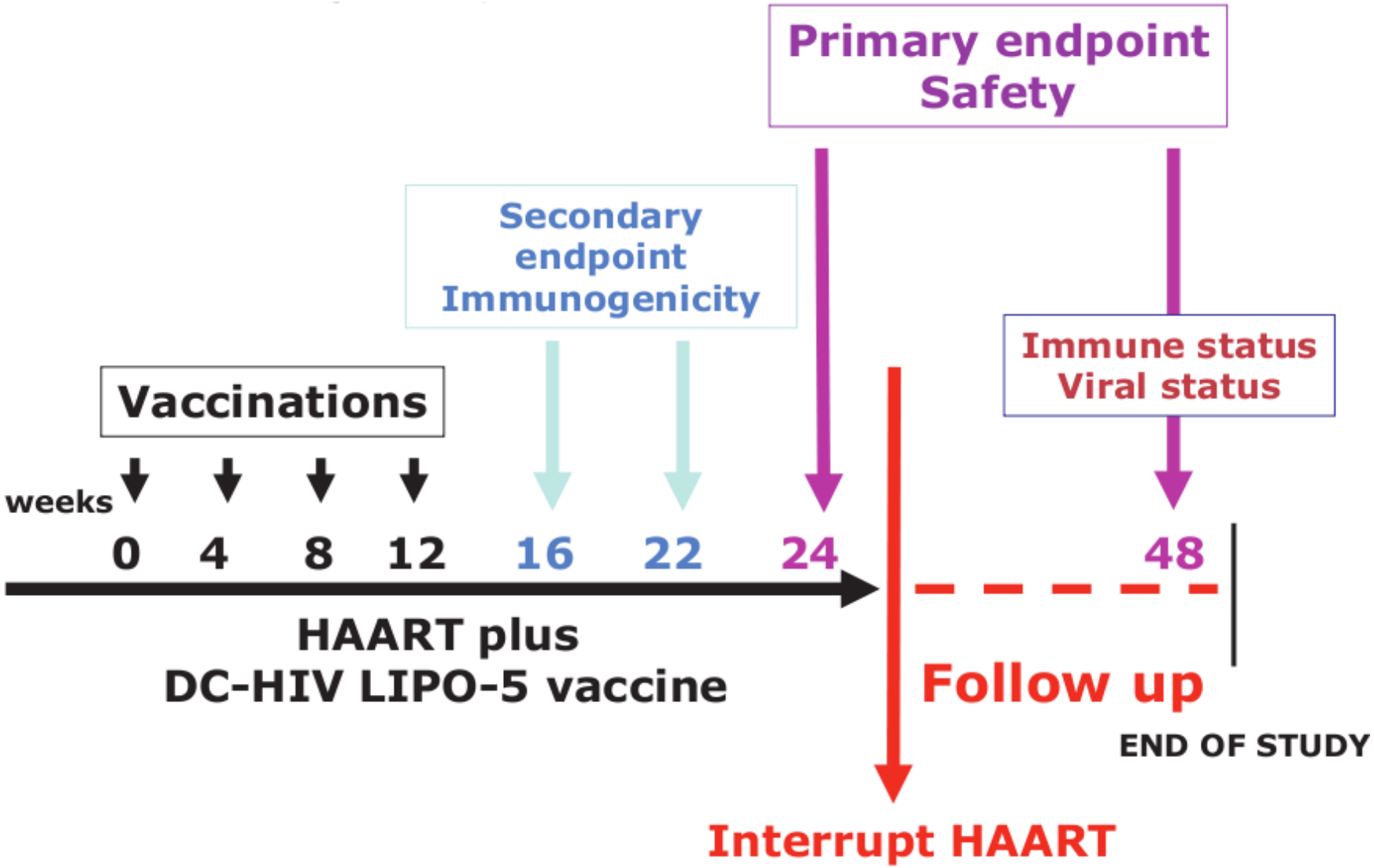
Optimisation of the weights

- X-score $\xi_h = \mathbf{X}_{h-1}\mathbf{u}_h$, Y-score $\omega_h = \mathbf{Y}_{h-1}\mathbf{v}_h$

$$\max_{\mathbf{v}_h,\ \mathbf{u}_h} \ Cov(\mathbf{Xu}, \mathbf{Yv}) - \lambda_1 \sum_{k=1}^{K} \|\mathbf{u}^{(k)}\|_2 - \lambda_2 \sum_{k=1}^{K} \sum_{a=1}^{A_k} \|\mathbf{u}^{(k,a)}\|_2 - \lambda_3 \|\mathbf{u}\|_1$$

such that $\mathbf{v}_h^T \mathbf{v}_h \leqslant 1$ and $\mathbf{u}_h^T \mathbf{u}_h \leqslant 1$.

DALIA application



Preliminary results – selected variables

- 19 modules, 784 genes total of 1452 selected variables.

	size.group	consistent	W4	W8	W12	W16
M3.2	126	9	53	42	31	42
M4.1	60	0	24	7	5	7
M4.13	72	3	35	15	16	27
M4.15	41	6	17	11	13	15
M4.2	43	5	15	7	10	14
M4.6	104	5	33	26	26	28
M4.7	82	2	31	15	15	16
M5.1	214	9	36	40	35	47
M5.14	54	3	26	8	8	13
M5.15	24	14	18	18	19	20
M5.5	211	2	77	25	27	30
M5.7	119	3	31	15	13	19
M6.13	38	2	13	8	8	10
M6.14	33	1	7	8	5	8
M6.6	40	2	12	8	17	21
M6.9	35	1	15	5	4	4
M7.1	150	9	33	35	25	41
M7.27	29	1	11	2	3	8
M8.14	27	0	6	4	8	2

R Package

sgsPLS Available now on GITHUB

```
library(devtools)
install_github("sgspls", "matt-sutton")
```

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%

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.

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 $\mathcal{M}_{h-1} = \mathbf{X}_{h-1}^{\mathsf{T}} \mathbf{Y}_{h-1}$:

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 $\mathcal{M}_{h-1} = \mathbf{X}_{h-1}^{\mathsf{T}} \mathbf{Y}_{h-1}$:

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

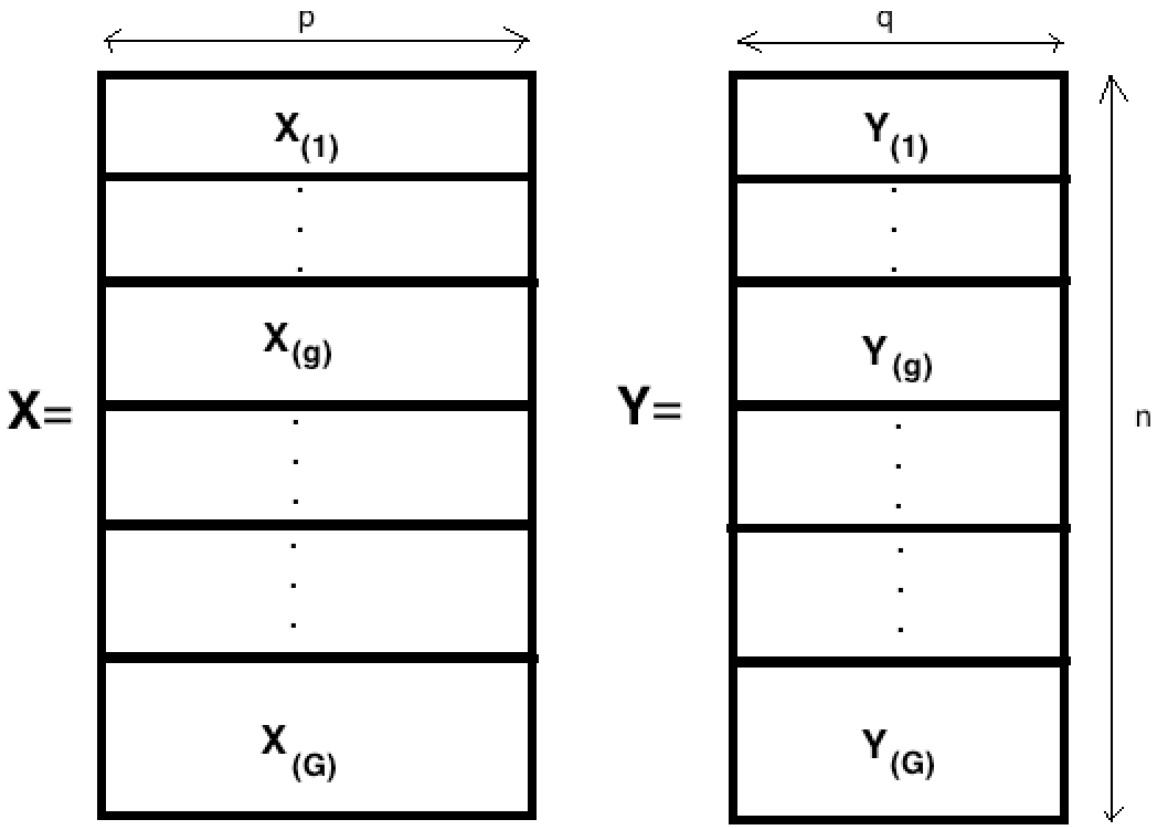
This matrix fits into memory.

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

- No need to load the big matrices **X** and **Y**
- Use memory-mapped files (called “filebacking”) through the big-memory 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 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 big-memory 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.

Illustration of group PLS with Big-Data

- Simulated: **X** (5GB) and **Y** (5GB);
- $n = 560,000$ observations, $p = 400$ and $q = 500$;
- Linked by two latent variables, made up of sparse linear combinations of the original variables;
- Both **X** and **Y** have a group structure: 20 groups of 20 variables for **X** and 25 groups of 20 variables for **Y**;
- Only 4 groups in each data set are relevant, 5 variables in each of these groups are not relevant.

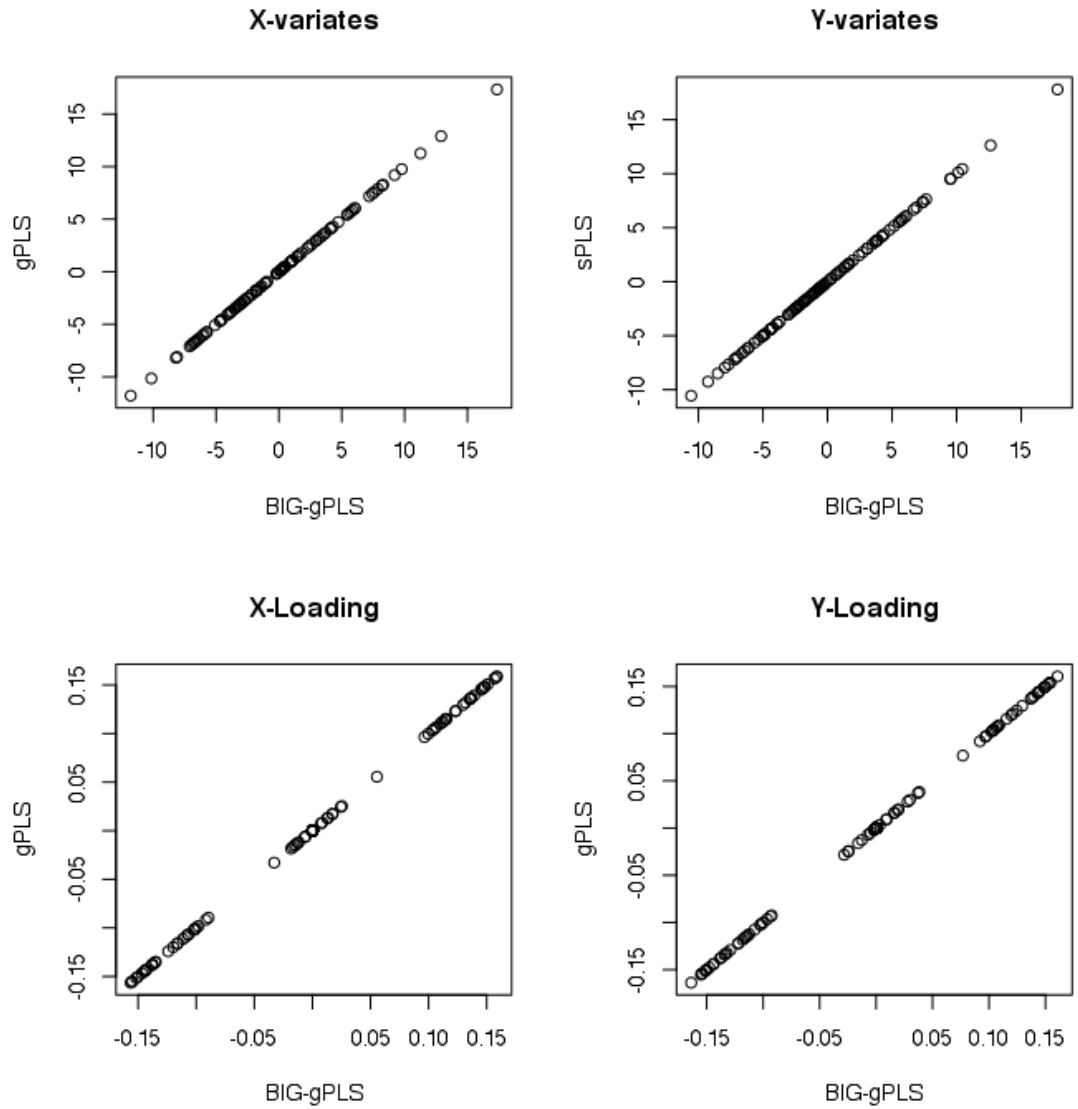


Figure 1: Comparison of gPLS and BIG-gPLS (for small $n = 1,000$)

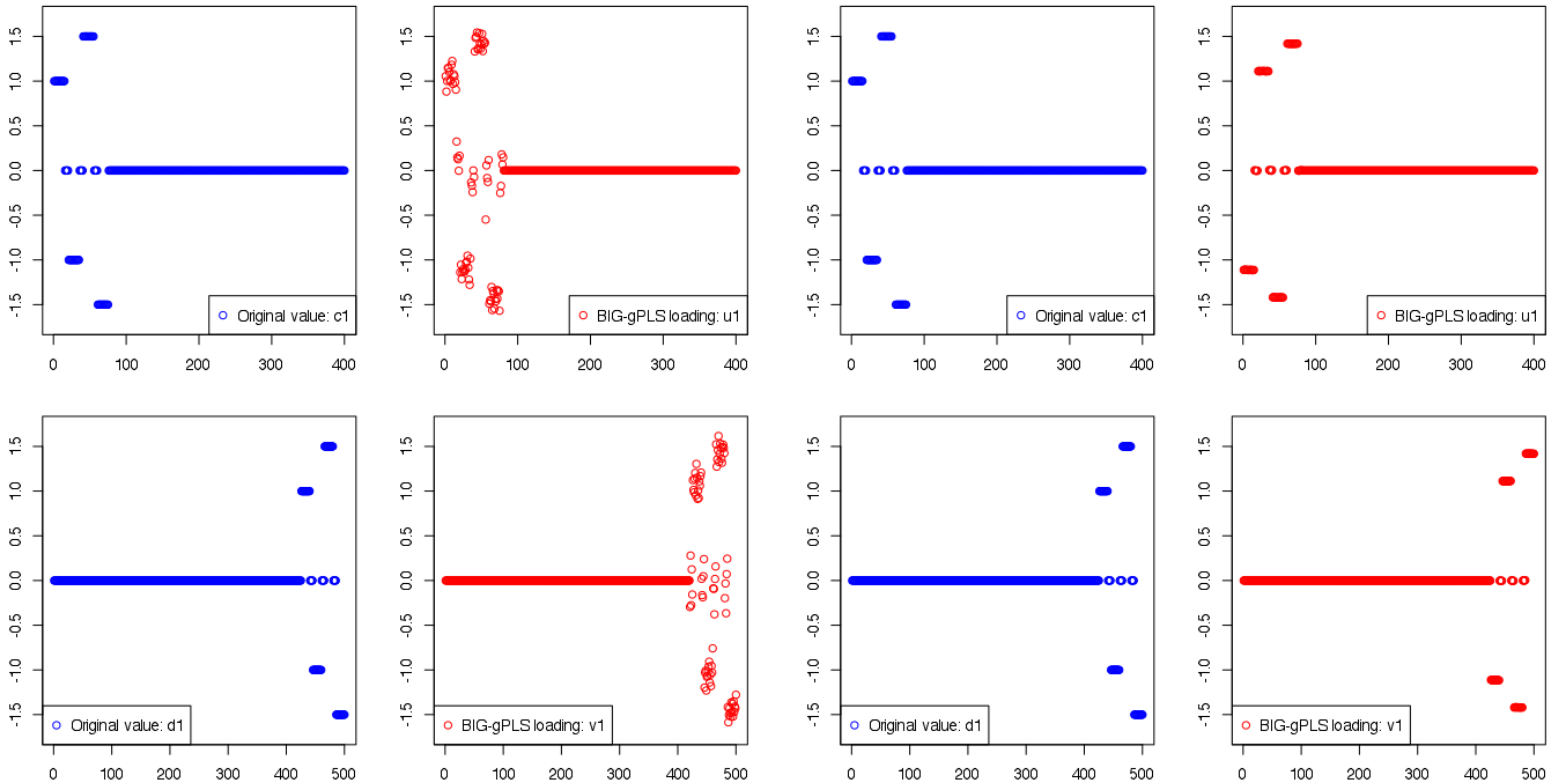
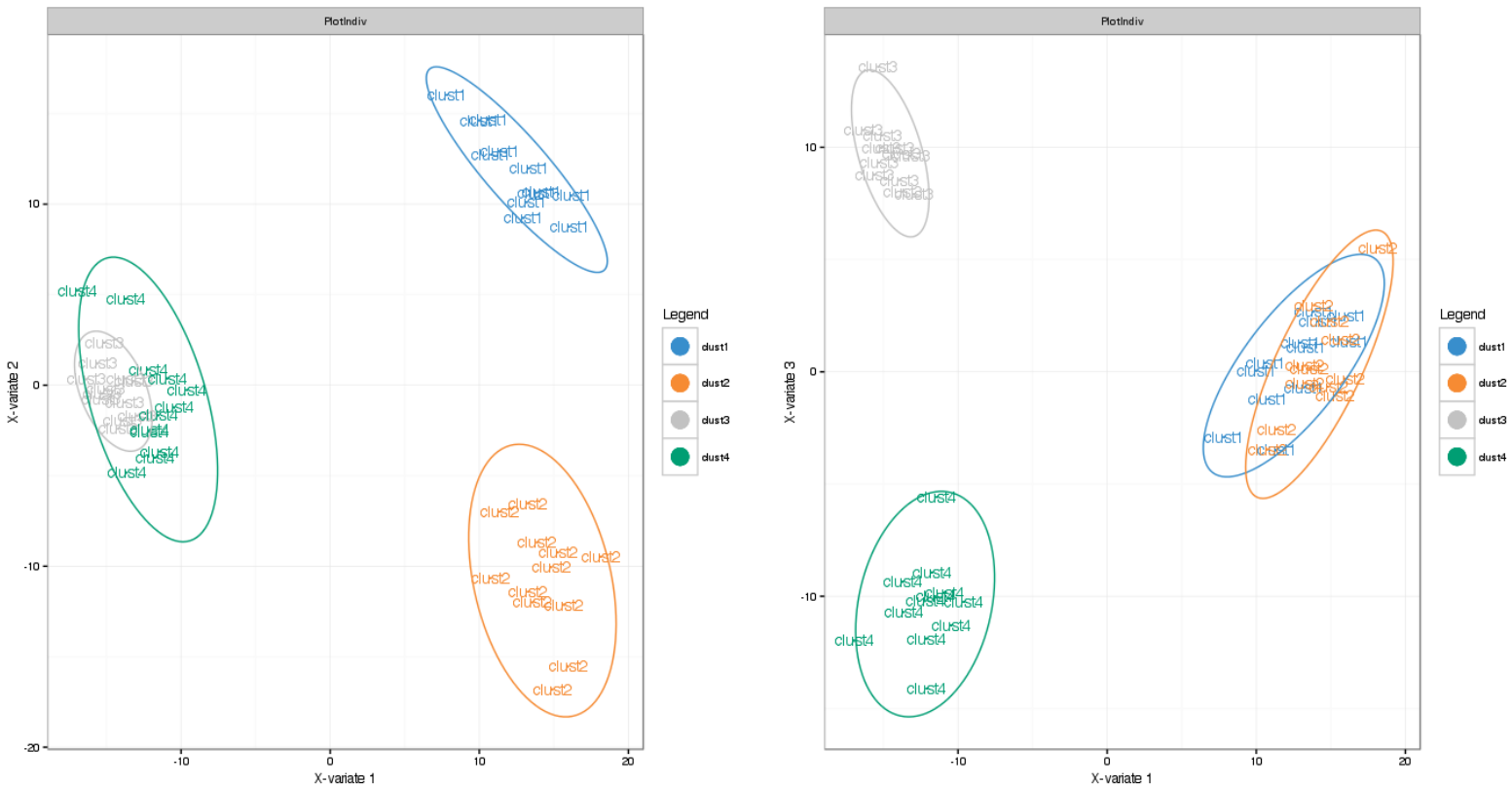


Figure 2: Use of BIG-gPLS. Left: small n. Right: Large n. Blue: truth. Red: Recovered.

Regularised PLS Discriminant Analysis

Categorical response variable becomes a dummy matrix in PLS algorithms:



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.
- bigsgPLS Available now on GITHUB:

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

- We have also offered a version of this algorithm for any combination of large values of n, p and q.

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.