La méthode PLS

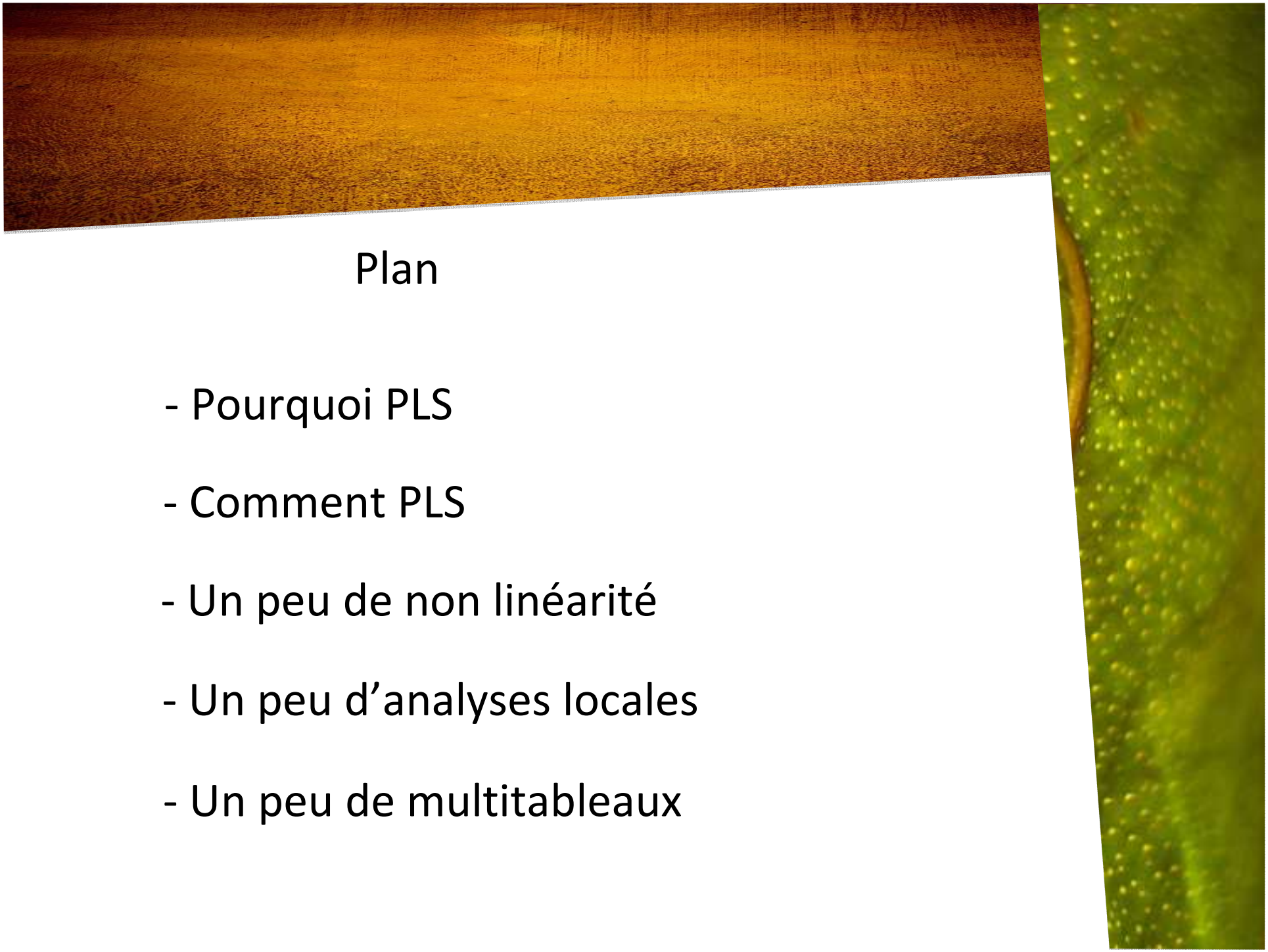
à travers quelques applications,
modifications
et
extensions.

Robert Sabatier

sabatier@univ-montp1.fr

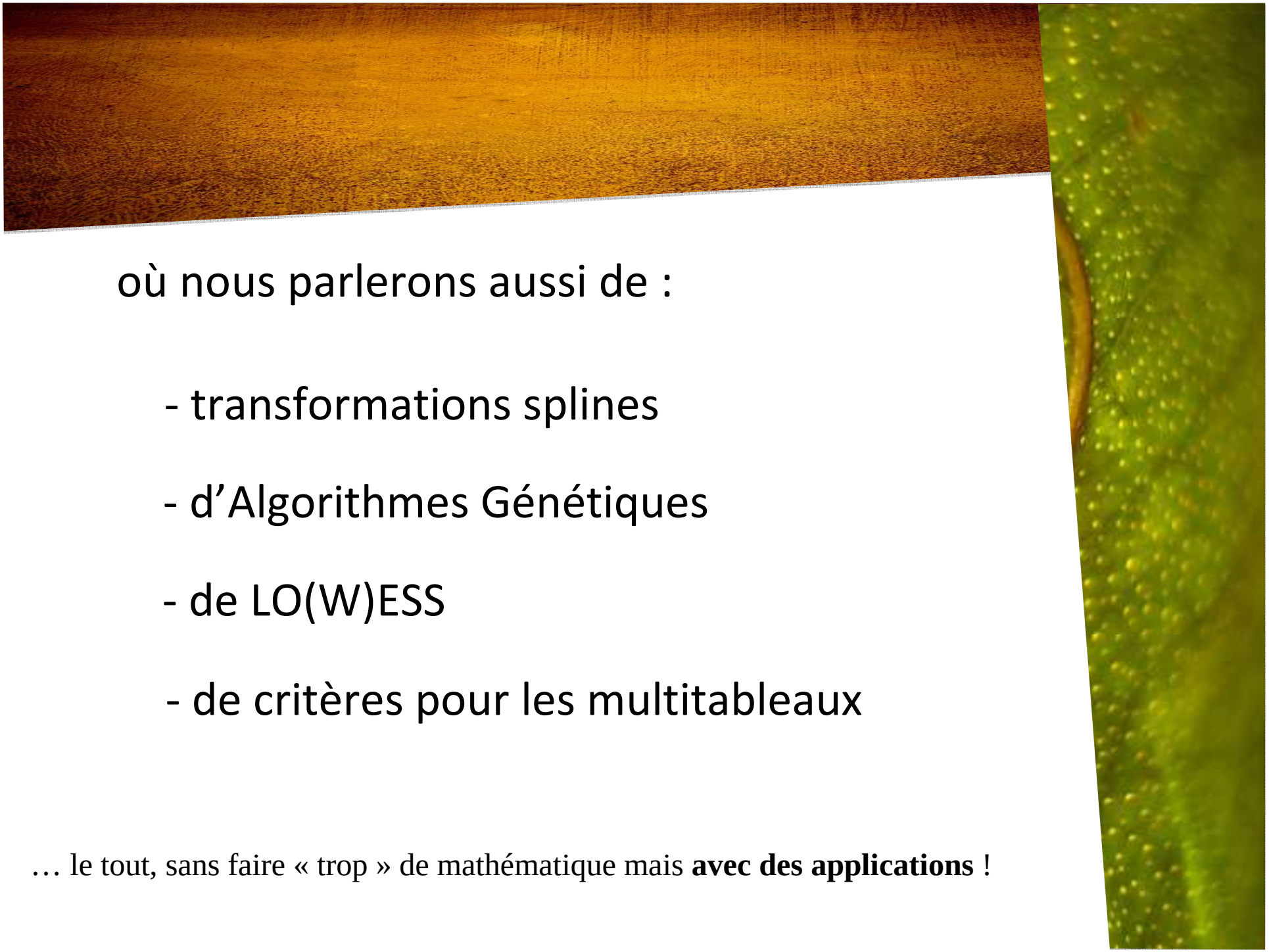
EA 2415

IRMA 17 Décembre 2010



Plan

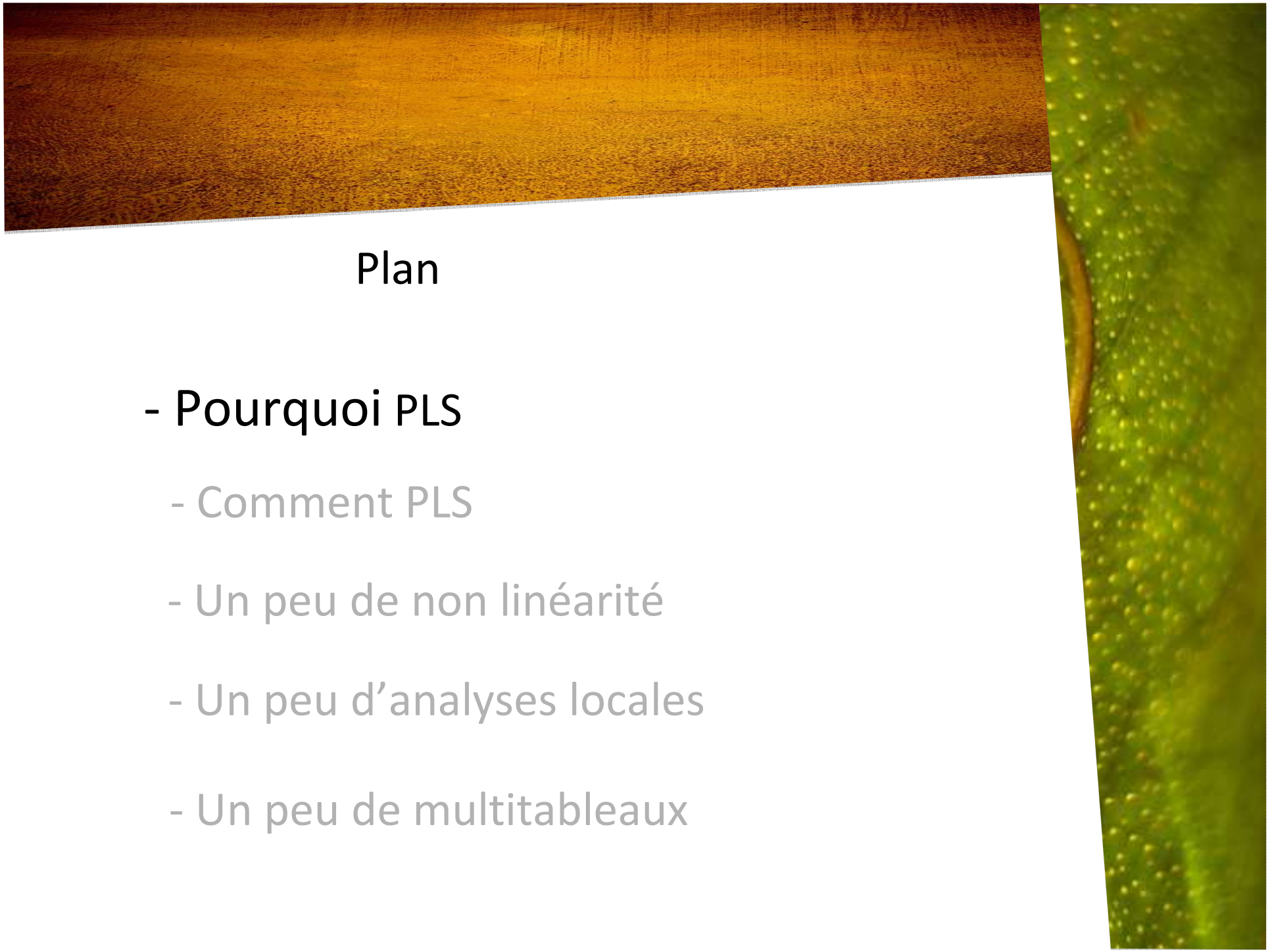
- Pourquoi PLS
- Comment PLS
- Un peu de non linéarité
- Un peu d’analyses locales
- Un peu de multitableaux



où nous parlerons aussi de :

- transformations splines
- d'Algorithmes Génétiques
- de LO(W)ESS
- de critères pour les multitableaux

… le tout, sans faire « trop » de mathématique mais **avec des applications** !



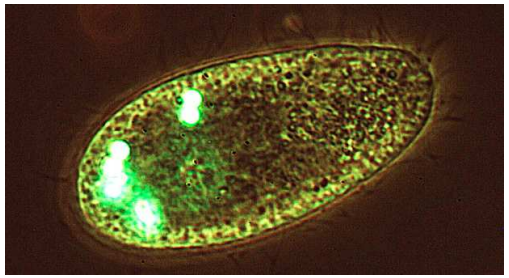
Plan

- Pourquoi PLS
  - Comment PLS
  - Un peu de non linéarité
  - Un peu d’analyses locales
  - Un peu de multitableaux

Un exemple de modélisation en biochimie

Moulton M.P. & Schultz T.W. (1986) Structure-activity relationships of selected pyridines II. Principal components analysis, *Chemosphere*, vol 16, **1**, 59-67.

Données de toxicités (**log BR** : logarithme d'inhibition à 50%) sur *Tetrahymena pyriformis* (protozoaire cilié) utilisé comme modèle en biochimie (très présent en eau douce).



Les 20 lignes correspondent à des para-substitutions sur des pyridines (c'est-à-dire des substitutions chimiques sur hydrocarbures aromatiques).

Les 6 variables explicatives sont des descripteurs moléculaires :

- **MR** : Réfractométrie Moléculaire,
- **X** : Indice de connectivité des liaisons simples,
- **Pi** : Paramètre hydrophobique,
- **sp** : Constante électronique de Hammett,
- **F** : Paramètre du champ électronique,
- **R** : Paramètre de résonance électrique

MR	X	Pi	sp	F	R	log BR
1.03	0	0	0	0	0	-1.327
5.65	0.5	0.56	-0.17	-0.04	-0.13	-0.895
10.3	1.061	1.02	-0.15	-0.05	-0.1	-0.297
6.03	0.598	0.71	0.23	0.41	-0.2	-0.862
8.88	0.35	0.86	0.23	0.44	-0.17	-0.31
6.33	0.474	-0.57	0.66	0.51	0.19	-0.819
11.18	0.704	-0.55	0.5	0.32	0.2	-0.835
6.88	0.525	-0.65	0.42	0.31	0.13	-0.159
30.33	2.364	1.05	0.43	0.31	0.16	-0.093
12.47	0.816	-0.64	0.31	0.41	-0.07	-0.814
5.47	0.289	-1.23	-0.66	0.02	-0.68	-0.439
2.85	0.224	-0.67	-0.37	0.29	-0.64	-1.587
15.55	0.816	0.18	-0.83	0.1	-0.92	-0.635
7.19	0.57	-1.03	0	0	0	-1.329
6.93	0.678	-0.32	0.45	0.33	0.15	-1.386
10.28	0.458	-0.38	0.1	0.25	-0.13	-0.547
9.81	0.493	-1.49	0.36	0.24	0.14	-1.015
25.36	1.91	1.96	-0.01	0.08	-0.08	0.664
30.01	2.264	2.01	-0.09	-0.08	-0.01	0.676
19.62	1.5	1.98	-0.2	-0.07	-0.13	0.164

Résultats de la Régression Multiple

Response: logBR

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
MR	1	4.7470	28.4300	0.0001361	***
X	1	0.0003	0.0019	0.9660416	
Pi	1	0.6167	3.6936	0.0768165	
sp	1	0.0003	0.0017	0.9677052	
F	1	0.0416	0.2489	0.6262196	
R	1	0.0181	0.1082	0.7473975	
Residuals	13	2.1706	0.1670		

Residual standard error: 0.4086 on 13 degrees of freedom
Multiple R-Squared: **0.7142**, Adjusted R-squared: 0.5823

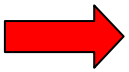
MAIS . . .

R = matrice de corrélation entre les X

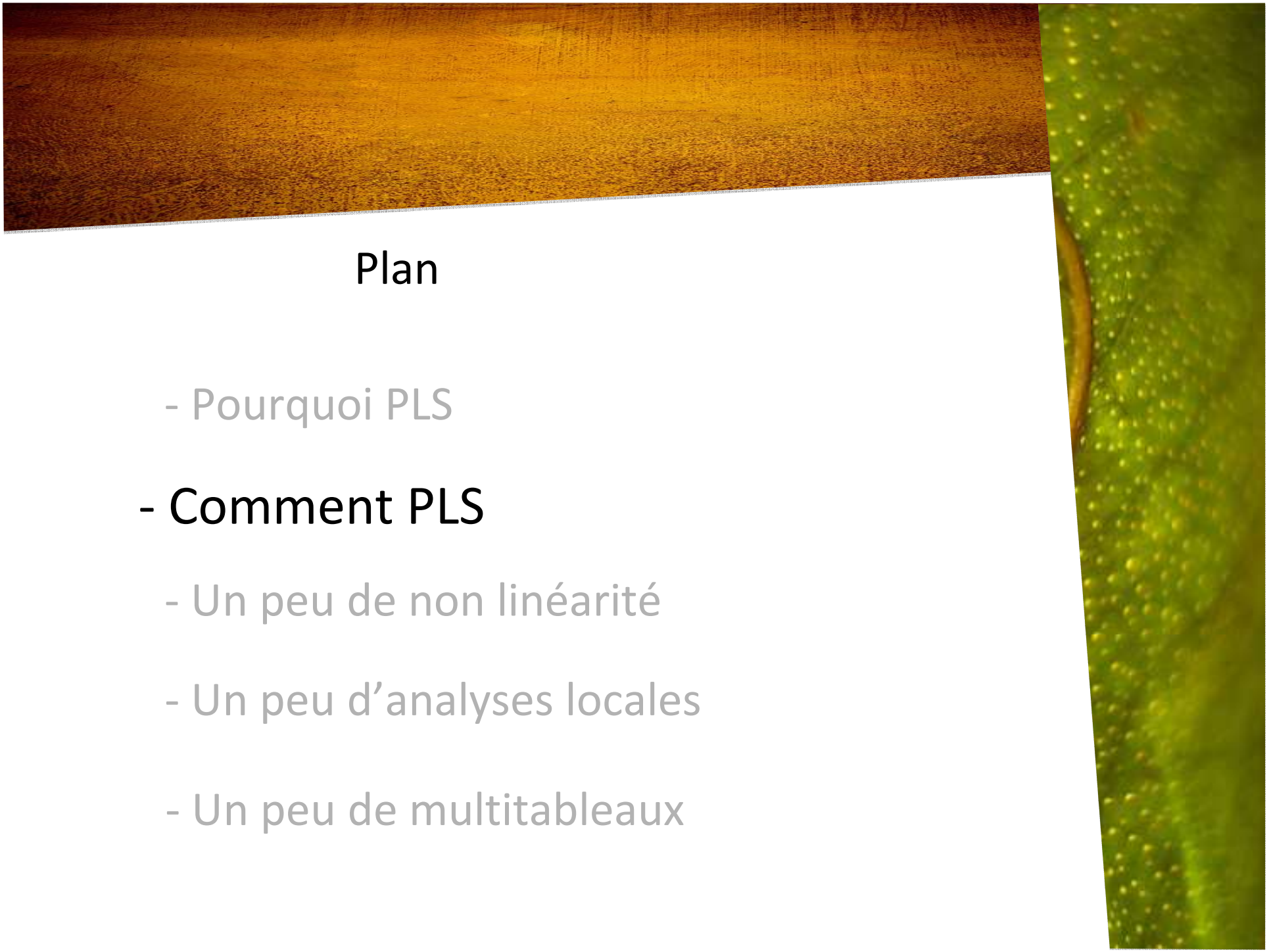
	MR	X	Pi	sp	F	R
MR	1.0000	0.9671	0.6791	-0.0052	-0.2191	0.1384
X	0.9671	1.0000	0.7282	0.0365	-0.2600	0.2134
Pi	0.6791	0.7282	1.0000	-0.1420	-0.3622	0.0362
sp	-0.0052	0.0365	-0.1420	1.0000	0.6586	0.8888
F	-0.2191	-0.2600	-0.3622	0.6586	1.0000	0.2418
R	0.1384	0.2134	0.0362	0.8888	0.2418	1.0000

valeurs propres de R : 2.7593 2.2220 0.64143 0.3517 0.0252 0.0004

c'est-à-dire les données sont « mal conditionnées » (dernière valeur propre « presque nulle »)



« le modèle est très instable »



Plan

- Pourquoi PLS

- Comment PLS

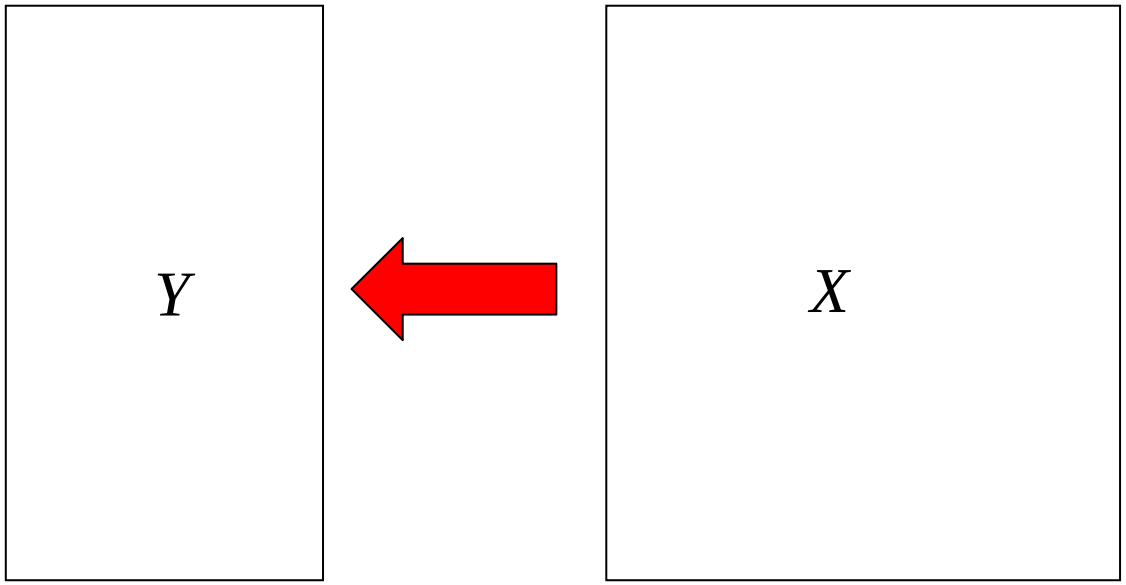
  - Un peu de non linéarité

  - Un peu d’analyses locales

  - Un peu de multitableaux

Les données, le problème et sa solution

- Prédire la matrice $Y_{n \times q}$ par $X_{n \times p}$



- Mais : si $n < p$ ou/et problèmes de colinéarité ou/et données manquantes
le modèle linéaire habituel est donc à la peine !

- Hermann Wold (1966) propose la méthode Partial Least Square (PLS).

- si $q = 1$, PLS1 si $q > 1$ PLS2 (ou simplement PLS).

Attention

- ne pas confondre PLS avec NIPALS (Nonlinear Iterative PArtial Least Square)

Algorithme itératif (genre puissance itérée) qui permet de trouver les valeurs propres et vecteurs propres d'une matrice à diagonaliser en effectuant des déflations. Permet également de tenir compte des valeurs manquantes (ce n'est pas le seul). Utilisé pour l'ACP, PLS, l'Analyse Canonique, la Régression Multiple etc…
C'est celui que nous allons donner pour présenter PLS1 et c'est H. Wold en 1966 qui l'a introduit.

- ne pas confondre PLS avec SIMPLS (Straightforward Implementation of stastistically inspire

Algorithme équivalent à PLS1 (dans le cas d'un seul *y*) mais sinon « proche » de NIPALS. Introduit par de Jong en 1993.
C'est celui que nous allons donner pour présenter PLS2

- ne pas confondre PLS avec SIMCA-P (Soft Independent Modelling by Cross Analogy)

Nom du logiciel dans lequel a été implémenté NIPALS, ainsi que d'autres programmes. Longtemps le seul … écrit par l'équipe de S. Wold.

Principe de la PLS1

- **Une méthode factorielle et linéaire** : recherches successives de h (h = 1, …, A) combinaisons linéaires des variables de départ : $t_h = Xw_h$, liées avec y, et ensuite les utiliser pour modéliser y linéairement.

- **Le problème est donc** :

$$\max_{w_h} \left\{ \text{cov}(y, t_h = Xw_h) \right\}$$

sous les contraintes :

$$w_h^t w_h = \left\| w_h \right\|^2 = 1$$

$$t_h^t t_l = 0, \text{pour } l \in \left\{ 1, 2, ..., (h-1) \right\}$$

- **Remarque** : on verra plus loin le choix de A (appelé dimension ou rang du modèle).

Construction de la première composante

$h = 1$

- **Les données** : $X = (x_1, x_2, \ldots, x_p)$, p variables explicatives et une, y, à expliquer.

- **Objectif :** construire une composante (combinaison linéaire des variables) de X liée avec y.

On cherche : $t_1 = Xw = w_{11}x_1 + w_{12}x_2 + \ldots + w_{1p}x_p = \sum_{j=1}^{p} w_{1j}x_j$

- **Solution** : $w_{1j} = \dfrac{\mathrm{cov}(x_j, y)}{\sqrt{\sum_{k=1}^{p} \mathrm{cov}^2(x_k, y)}}$ ➡ une « nouvelle variable » t_1

- **Retour à** y : régression simple de y par t_1 :

$y = c_1 t_1 + y^{(1)} = c_1 w_{11} x_1 + \ldots + c_1 w_{1p} x_p + y^{(1)}$, avec : $c_1 = \dfrac{\mathrm{cov}(t_1, y)}{\mathrm{cov}(t_1, t_1)}$

d'où le modèle : $\hat{y}^1 = c_1 t_1 = P_{t_1}(y) = \sum_{j=1}^{p} \hat{\beta}_j^{(1)} x_j$

et le résidu associé : $y^{(1)} = P_{t_1}^{\perp}(y) = (Id - P_{t_1})(y)$

Construction des composantes suivantes

$h = 2$

Objectif : construire une nouvelle combinaison linéaire des variables de X, non corrélée à t_1 (apport d'information nouvelle), qui explique au mieux le résidu de y, $y^{(1)} = y - c_1 t_1$

Calcul des résidus :

- pour y : $y^{(1)} = P_{t_1}^{\perp}(y) = (Id - P_{t_1})(y)$

- pour x_j : analogue, calcul de la régression simple de x_j sur t_1 : $x_j = P_{t_1}(x_j) + P_{t_1}^{\perp}(x_j)$
  c'est-à-dire : $x_j = p_{1j} t_1 + x_j^{(1)}$

On cherche donc : $t_2 = w_{21} x_1^{(1)} + w_{22} x_2^{(1)} + \ldots + w_{2p} x_p^{(1)}$

$$= w_{21}\left(x_1 - p_{11} t_1\right) + w_{22}\left(x_2 - p_{12} t_1\right) + \ldots + w_{2p}\left(x_p - p_{1p} t_1\right)$$

$$= \sum_{j=1}^{p} w_{2j}\left(x_j - p_{1j} t_1\right) = \sum_{j=1}^{p} w_{2j}\left(x_j - p_{1j}\left(\sum_{k=1}^{p} w_{1k} x_k\right)\right) = \sum_{j=1}^{p} \tilde{w}_{2j} x_j$$

Construction des composantes suivantes (fin)

- Calcul des coefficients :

Par analogie avec la première composante, on obtient : $w_{2j} = \dfrac{\mathrm{cov}(x_j^{(1)}, y^{(1)})}{\sqrt{\sum_{k=1}^{p} \mathrm{cov}^2(x_k^{(1)}, y^{(1)})}}$

- Retour à *y* : régression simple de y - $c_1 t_1$ sur t_2

d'où, après quelques manipulations algébriques, on obtient :

$$y = P_{t_2}\left(\hat{y}^1 + y^{(1)}\right) + y^{(2)} = P_{t_2}\left(P_{t_1}(y)\right) + y^{(2)} = \left(P_{t_1} + P_{t_2}\right)(y) + y^{(2)} = c_1 t_1 + c_2 t_2 + y^{(2)}$$

et le modèle de dimension 2 : $\hat{y}^2 = \left(P_{t_1} + P_{t_2}\right)(y) = c_1 t_1 + c_2 t_2 = \sum_{j=1}^{p} \hat{\beta}_j^{(2)} x_j$

On peut recommencer la procédure jusqu'à obtenir $h = A$ composantes.

On note également que, par construction, nous avons : $\mathrm{var}(\hat{y}^{(h)}) \leq \mathrm{var}(\hat{y}^{(h+1)})$

c'est-à-dire, à chaque nouvelle étape, la **variance expliquée croit**.

Choix de la dimension du modèle (A)

- **Par validation croisée** (en général, pour n « pas trop élevé », leave-one out).

- **Erreur sur le jeu d'apprentissage** Residual Sum of Squares : $RSS_h = \sum_{i=1}^{n} \left(y_i - \hat{y}_i^h \right)^2$

  $\hat{y}_i^h = \left(\sum_{l=1}^{h} P_{t_k} \right)(y_i)$ est la prédiction, de l'observation i, avec un modèle à h composantes.

  RSS_h ↘ quand h ↗ .

- **Erreur de prédiction** Predicted REsidual Sum of Squares : $PRESS_h = \sum_{i=1}^{n} \left(y_i - \hat{y}_{(-i)}^h \right)^2$

  où $\hat{y}_{(-i)}^h$ est la valeur prédite, pour l'observation i, en construisant le modèle à h composantes sans utiliser cette observation.

  $PRESS_h$ ↘ tant qu'on améliore le modèle et ↗ quand on commence à faire du surajustement : ➔ choix du nombre optimal de composantes.

- Si on choisit $A = p$ composantes, on obtient le modèle classique des moindres carrés (Régression Multiple, s'il existe) d'où le nom de PLS. Mais, bien sûr en général, $A \leq p$

Et ce dont on ne parlera pas …

- l'écriture du modèle, de dimension h, en fonction des x_j

- données manquantes

- R^2 par validation croisée

- Q^2(cum)

- $VIP_{h,j}$

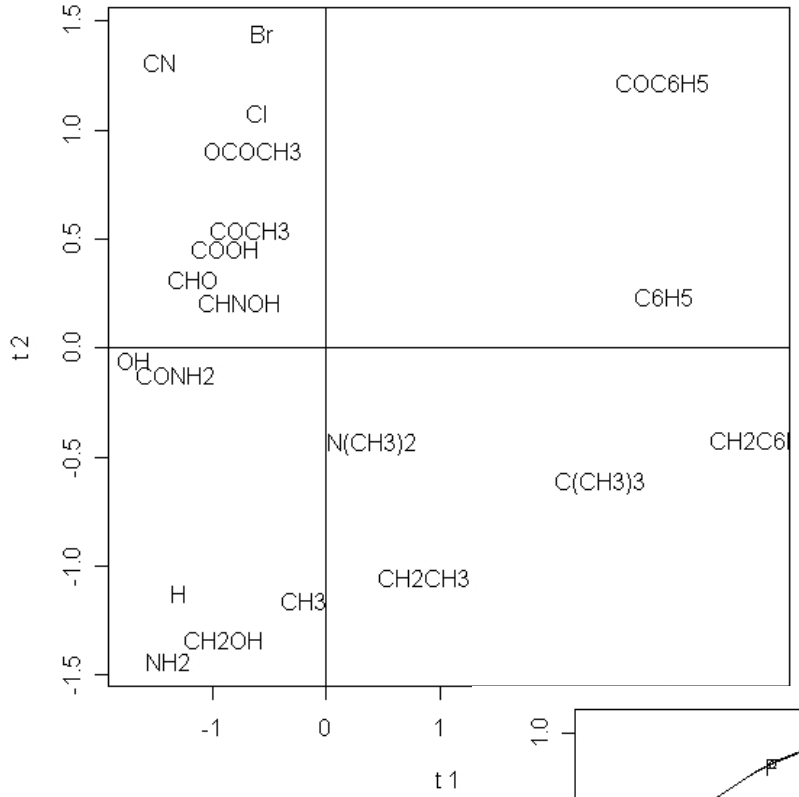
- DModX et DModY

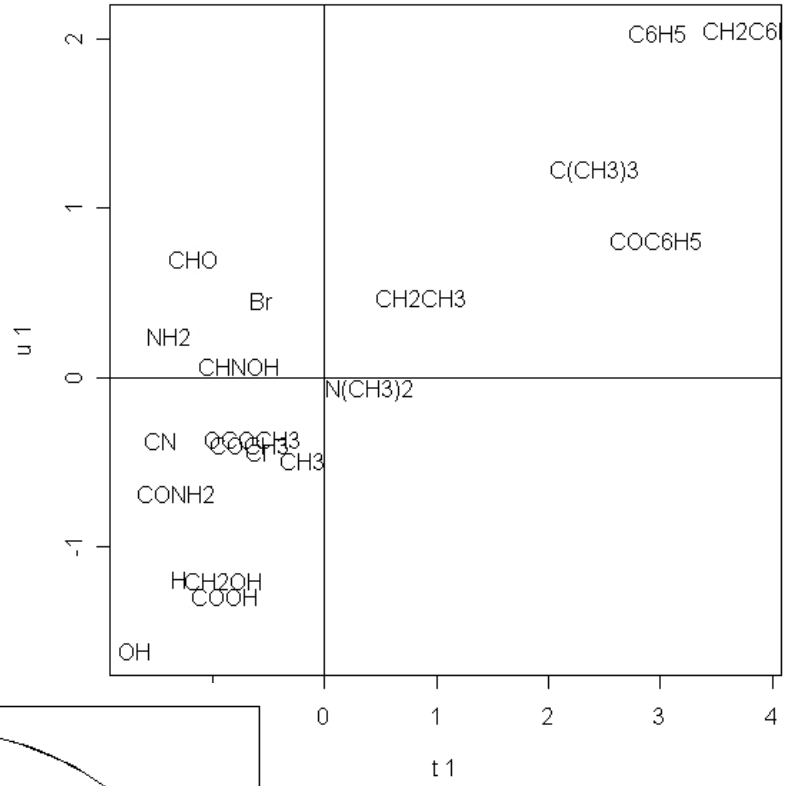
- autres algorithmes pour PLS1, voir Andersson (2009)

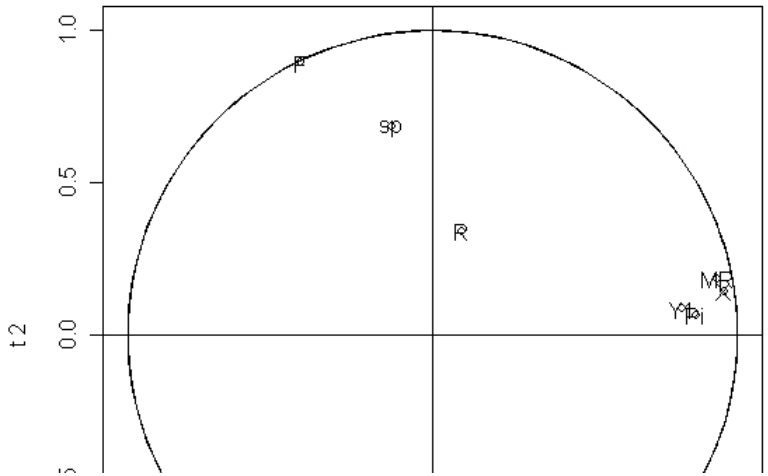
Retour aux données de biochimie
sorties numériques de PLS1

```
axe numéro  1
   cov(t,y) = 1.354   r(t,y) = 0.818
   var(t)   = 2.742   var(y) = 1
   Inertie de Y expliquée = 0.6693
   corrélations t et variables de Y
                0.8181
   corrélations t et variables de X
                 0.9347  0.9563  0.8654 -0.1332 -0.4318  0.0974

axe numéro  2
   cov(t,y) = 0.076   r(t,y) = 0.146
   var(t)   = 0.816   var(y) = 0.331
   Inertie de Y expliquée = 0.0071
                  cumulée = 0.6764
   corrélations t et variables de Y
                0.0842
   corrélations t et variables de X
                0.1820 0.1413 0.0632 0.6829 0.8943 0.3398

. . . . .


axe numéro  6
   cov(t,y) = 0.001   r(t,y) = 0.094
   var(t)   = 0.0004   var(y) = 0.288
   Inertie de Y expliquée = 0.0026
                  cumulée = 0.7142
```

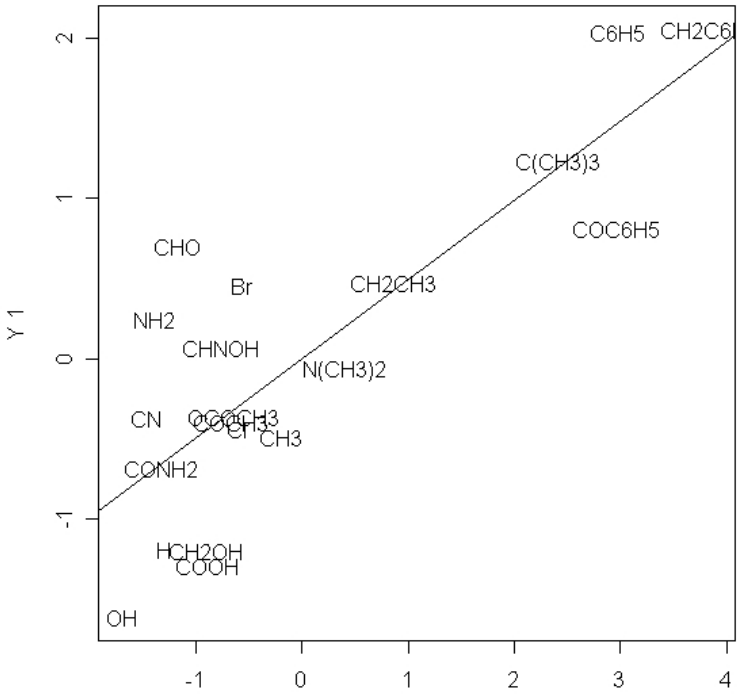
Retour aux données de biochimie
sorties graphiques de PLS1

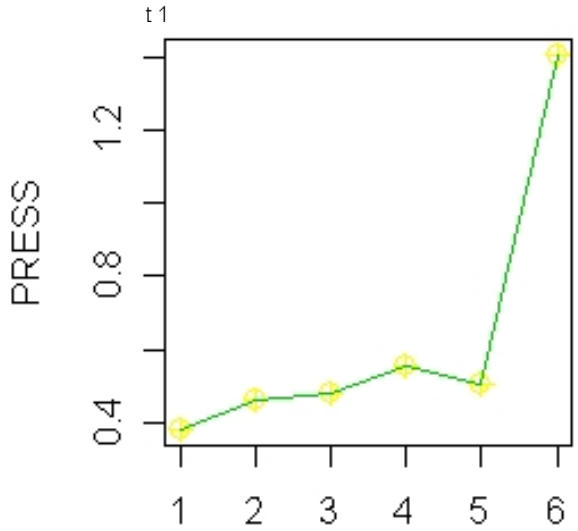


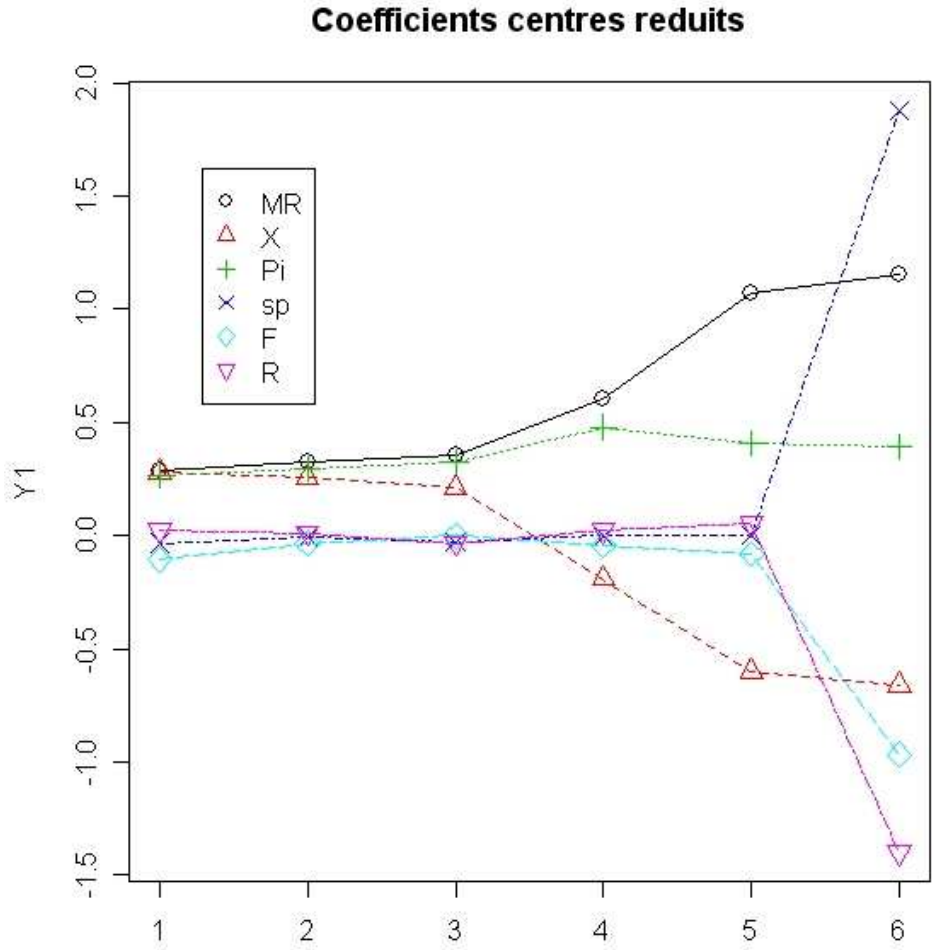




Retour aux données de biochimie
sorties graphiques de PLS1 (suite)







Et pour la PLS2

- Présence de **plusieurs variables à prédire** (Y = matrice), mais même démarche.

- Une méthode linéaire : utilisation de combinaisons linéaires des variables explicatives ET des variables à prédire.

- Une méthode factorielle : choix des coefficients des combinaisons de sorte à construire des composantes qui maximisent la covariance entre composante en X et composante en Y.

- Utilise, comme intermédiaire de calcul, dans l'une de ses versions, une technique « un peu » oubliée de l'AD, la méthode Inter-battery de Tucker (1958), mise au point dans un contexte totalement différent.

- PLS1 est un cas particulier de PLS2.

- La présentation choisie, est l'une des possibles… (algorithme SIMPLS de de Jong)

Construction de la première composante de PLS2

$h = 1$

- Les données : une matrice $X = (x_1, x_2, \ldots, x_p)$ de dimension $(n \times p)$ et Y de dimension $(n \times q)$

- Objectif : construire une combinaison linéaire des variables de X et une combinaison linéaire des Y qui maximisent leur covariance.

On cherche des combinaisons linéaires des X et des Y telles que :

$$\max_{w_1, c_1} \left\{ \text{cov}\left(t_1 = X w_1, u_1 = Y c_1 \right) \right\} \quad \text{avec} \quad \left\| w_1 \right\|^2 = \left\| c_1 \right\|^2 = 1$$

On obtient les vecteurs solutions, w_1 et c_1, par diagonalisation (analyse Inter-battery de Tucker) :

$$\left(Y^t X \right)^t \left(Y^t X \right) w_1 = \lambda_1 w_1 \qquad \left(X^t Y \right)^t \left(X^t Y \right) c_1 = \lambda_1 c_1$$

Attention : premier vecteur propre uniquement (mais même valeur propre pour les deux), avec : $\text{cov}(t_1, u_1) = \lambda_1$

Et pour les autres composantes de PLS2

$h = 2, \ldots, A$

- Calcul des résidus :

pour X : $X^{(h)} = (Id - P_{t_{h-1}})X^{(h-1)}$

pour Y : $Y^{(h)} = (Id - P_{t_{h-1}})Y^{(h-1)}$

avec : $X^{(1)} = X$, et $Y^{(1)} = Y$

- Calcul des composantes :

On cherche les vecteurs w_h et c_h, tels que : $\max_{w_h, c_h} \left\{ \text{cov}\left(X^{(h)} w_h, Y^{(h)} c_h \right) \right\}$

avec : $\|w_h\|^2 = \|c_h\|^2 = 1$

On obtient les vecteurs solutions, w_h et c_h, par diagonalisation (analyse Interbattery de Tucker) :

$$\left(Y^{(h)t} X^{(h)} \right)^t \left(Y^{(h)t} X^{(h)} \right) w_h = \lambda_1^{(h)} w_h \qquad \left(X^{(h)t} Y^{(h)} \right)^t \left(X^{(h)t} Y^{(h)} \right) c_h = \lambda_1^{(h)} c_h$$

Quelques propriétés des composantes de PLS2

$h = 2, \ldots, A$

Il y en a énormément, plus ou moins immédiates, plus ou moins utiles en pratique…

- **Les composantes** t_h sont orthogonales (non corrélées) : $t_h^t t_l = 0, \text{ pour } l = 1, 2, ..., (h-1)$

- **Les composantes** (t_h et u_h), à chaque étape, sont vecteurs propres de matrices.

- **Il existe d'autres orthogonalités** (c_h, w_h, …).

- **Le modèle de rang A** pour la variable k de Y : $\left(\hat{Y}^k\right)^{(A)} = \left(\sum_{h=1}^{A} P_{t_h}\right) Y^k = \sum_{j=1}^{p} \hat{\beta}_j^{A,k} x_j$

  et donc, si $A = p$, on retrouve la Régression Multiple (si elle est possible).

- Il n'est pas forcément utile de centrer simultanément X et Y.

- Choix optimal de A, données manquantes, etc… analogue à PLS1.

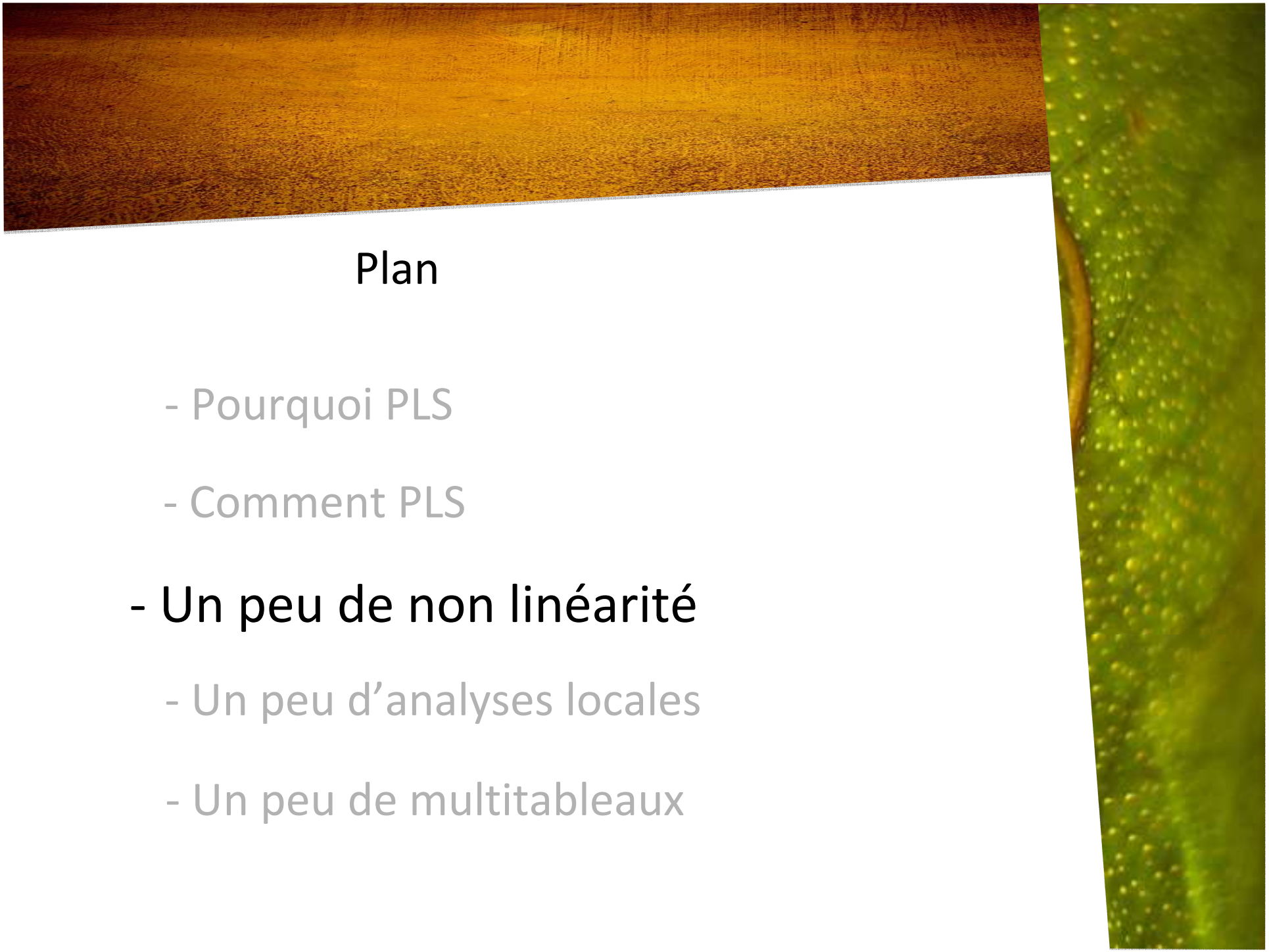
. . .

Donc PLS c’est …

- Une méthode, simple et efficace pour résoudre des problèmes de multi-colinéarité.
- Permet, par le calcul de composantes orthogonales, de bien gérer la dimension du modèle, à l’aide de la validation croisée.

mais PLS n’est pas …

- Une méthode pour gérer la non linéarité.
- Une méthode pour gérer les dépendances locales.
- Une méthode pour pratiquer la « discrimination » de façon optimale (PLS-DA).



Plan

- Pourquoi PLS
- Comment PLS
- Un peu de non linéarité
- Un peu d’analyses locales
- Un peu de multitableaux

Un exemple de Régression Multiple « a problème » et la méthode PLS1 pas mieux …

Frank I.E. (1995) Modern nonlinear regression methods, *Chemometrics and Intelligent Laboratory System*, **27**, 1-9.

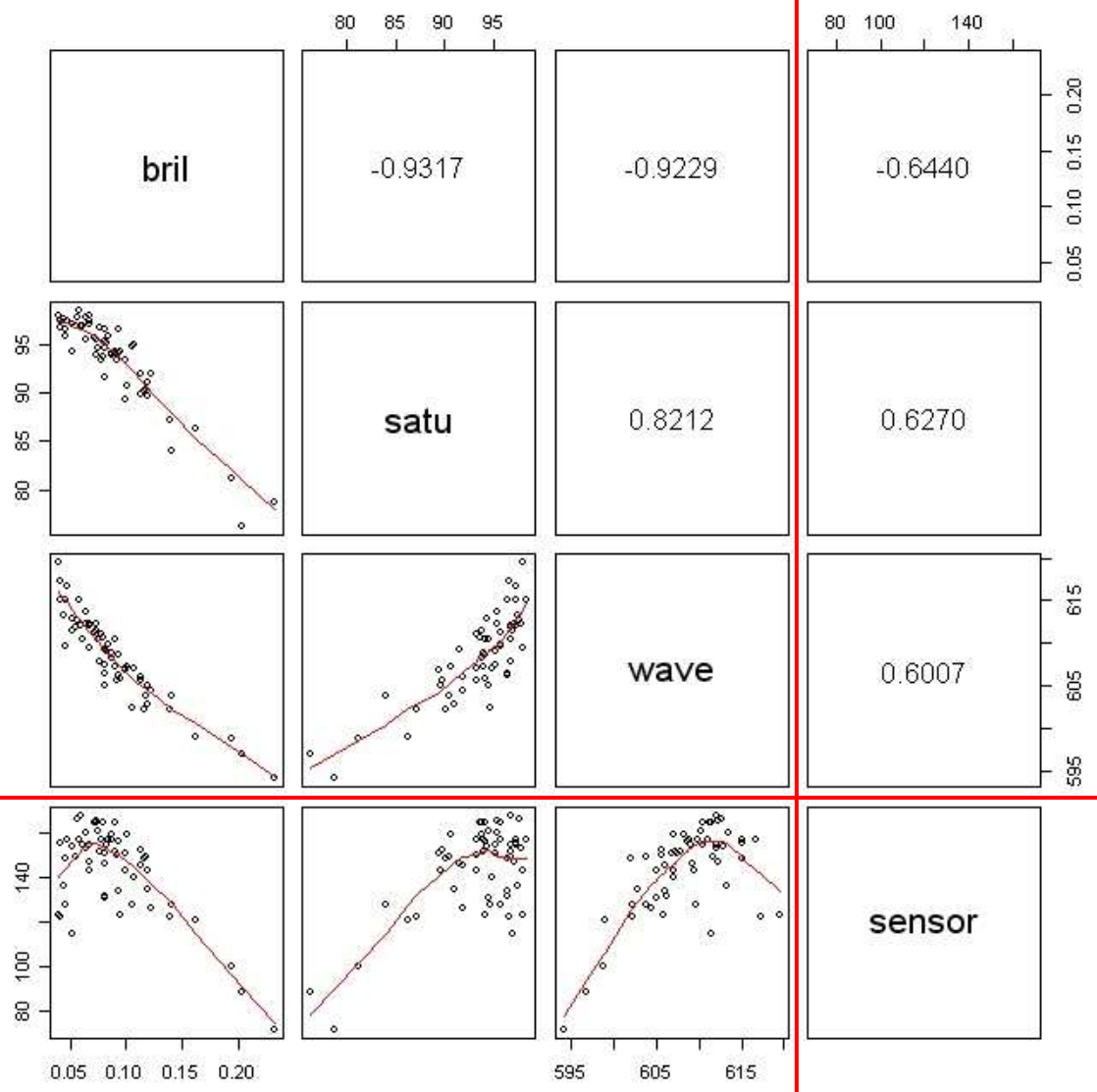
Un échantillon de 58 vins Barbaresco (vin rouge sec produit dans la région d'Alba - Piémont) décrit par trois paramètres spectrométriques : brillance (**bril**), saturation (**satu**) et longueur d'onde dominante (**wave**), sont utilisés pour prédire le score sensoriel (**sensor**).
Chaque valeur du score est la moyenne donnée par vingt experts.

bril	satu	wave	score
0.099	89.36	606.88	150.8
0.139	87.15	602.19	122.8
0.057	97.74	612.75	166.7
0.059	98.43	615.14	157.1
0.068	98	609.43	143.0
0.107	94.98	607.06	140.1
0.101	90.65	607.12	159.7
0.092	93.38	605.65	150.2
0.113	89.81	605.68	152.8
0.082	96.5	606.33	131.5
0.06	96.89	612.08	168.3
0.194	81.23	598.77	100.1
0.075	94.62	610.36	160.9
0.064	97.82	612.17	153.2
0.093	94.12	608.57	156.3
0.047	97.35	616.78	157.4
0.095	94.15	605.79	123.5
0.119	91.02	602.87	134.6
0.042	96.67	617.3	122.4
0.122	91.96	604.44	126.3
0.087	94.08	608.88	157.4
0.082	94.6	605.04	130.7
	…. .		
	…. .		

Diagramme croisé entre les variables de X et y pour les données de food chemestry

linéarité entre les X

« non linéarité » entre y et les X



Résultats de la Régression Multiple

Response: **sensor**

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
bril	1	8994.1	8994.1	38.7254	7.471e-08 ***
satu	1	119.7	119.7	0.5155	0.4759
wave	1	32.6	32.6	0.1403	0.7094
Residuals	54	12541.6	232.3		

Residual standard error: 15.24 on 54 degrees of freedom
Multiple R-Squared: **0.4217**, Adjusted R-squared: 0.3896

Conclusions :

- évidentes non linéarités, d'où mauvaise prédiction,

- si l'on calcule le R^2 par cross-validation, il est égal à : **0.26** !

PLS1 « usuel » sur les données de food chemistry

axe numéro 1

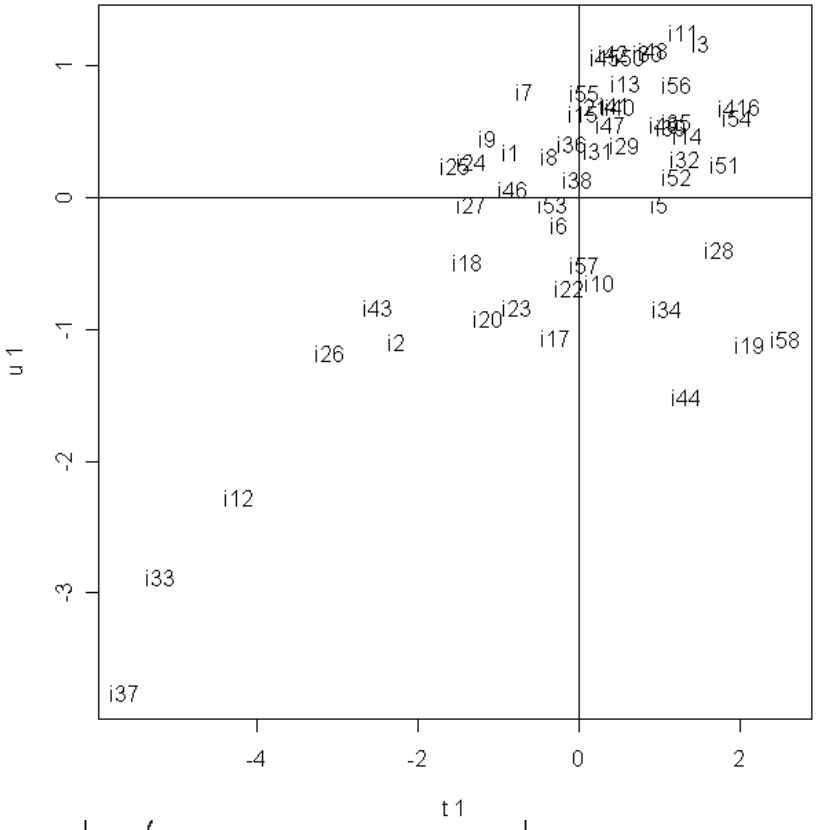
cov(t,u) = 1.081 r(t,u) = 0.648
var(t) = 2.784 var(u) = 1
Inertie de Y expliquée = **0.4198**
corrélations t et variables de Y
0.6480
corrélations t et variables de X
-0.9882 0.9537 0.9479

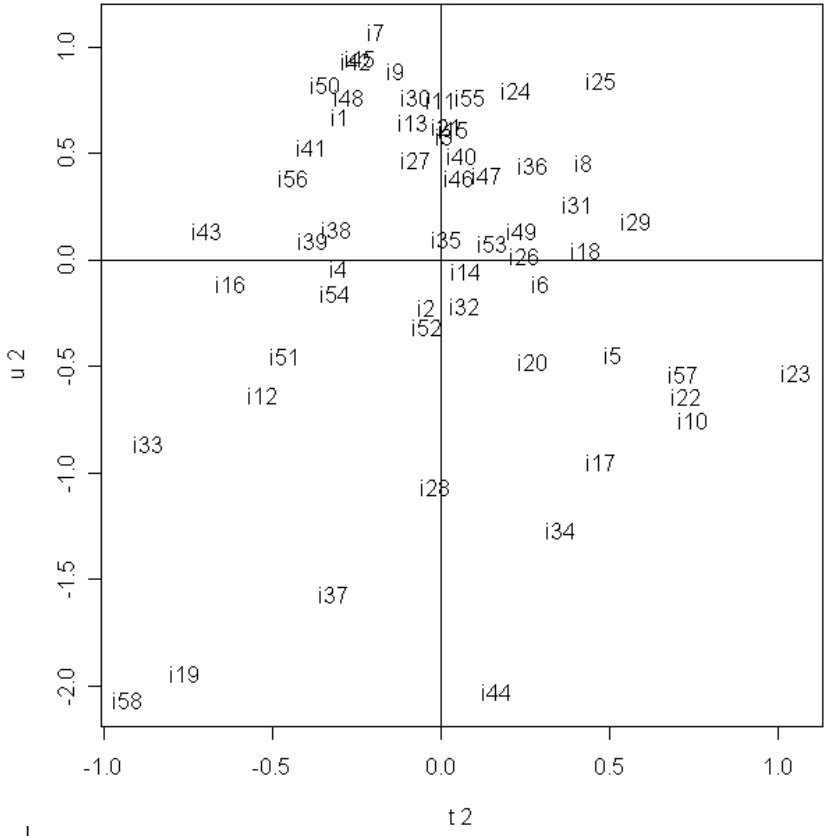
---

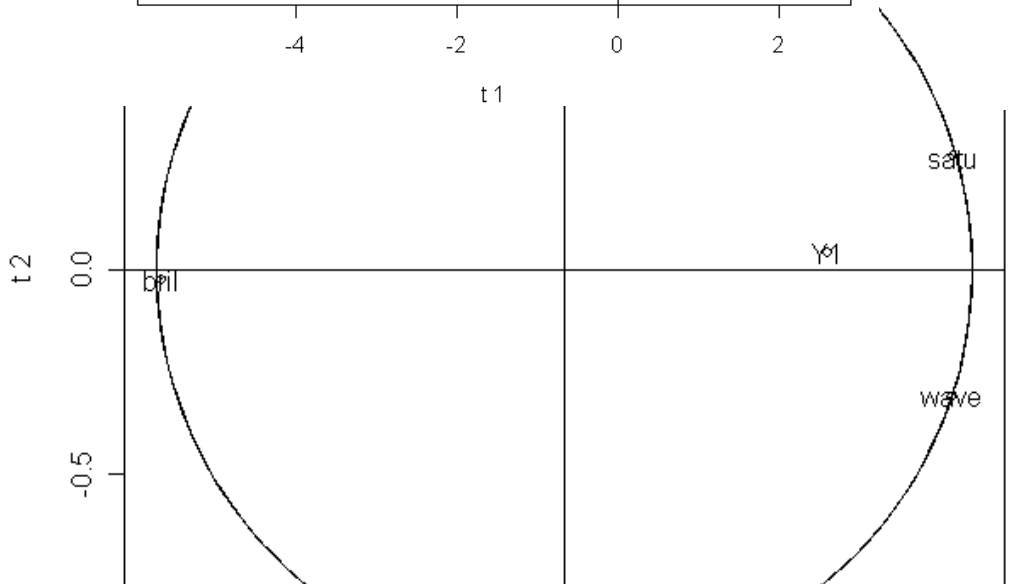
axe numéro 2

cov(t,u) = 0.016 r(t,u) = 0.053
var(t) = 0.170 var(u) = 0.580
Inertie de Y expliquée = **0.0016**
cumulée = **0.4214**
corrélations t et variables de Y
0.0404
corrélations t et variables de X
-0.0257 0.2764 -0.3161
corrélations entre le u calculé et les précédents
0.7617

Sorties du PLS1 linéaire







Conclusions analogues, bien sûr !

Introduction de la non linéarité dans PLS

- Déjà fait un certain nombre de fois. En premier dans la liaison entre les *t* et les *u*
  puis pas mal d'autres méthodologies comme : CART, ACE, SMART, MARS, ASPLS, PLSS, SVM, K-PLS…

- Mais, quels types de transformations choisir et comment les paramétrer et les optimiser ?
  Aucune méthode ne semble vraiment s'imposer.

Nous allons proposer :

un **modèle additif** (en les variables transformées),
avec des **transformations splines** (pour chaque variable explicative),
le **degré** (pour chaque variable explicative),
son **nombre de nœuds**,
la **positions de ses nœuds**,
et le **rang du modèle PLS**,

et … tout sera optimisé **simultanément**… !?

Prise en compte de la non linéarité avec des fonctions splines de régression

- Utilisation des **splines de régression** à cause de leur simplicité (par rapports à celles de lissage ou autres…) pour introduire la non linéarité dans PLS1 et PLS2.

- Une fonction spline est un **polynôme de degré *d* par morceaux** (définis par des **nœuds**).

- Toute fonction spline $s(x)$, de degré connu et de nœuds fixés, peut se mettre sous la forme d'une somme : $s(x) = \sum_{l=1}^{r} \beta_l B_l(x)$

et l'ensemble : $\{B_l(x)\}_{l=1,\ldots,r}$ est la base d'un espace vectoriel de dimension $r = d + K + 1$

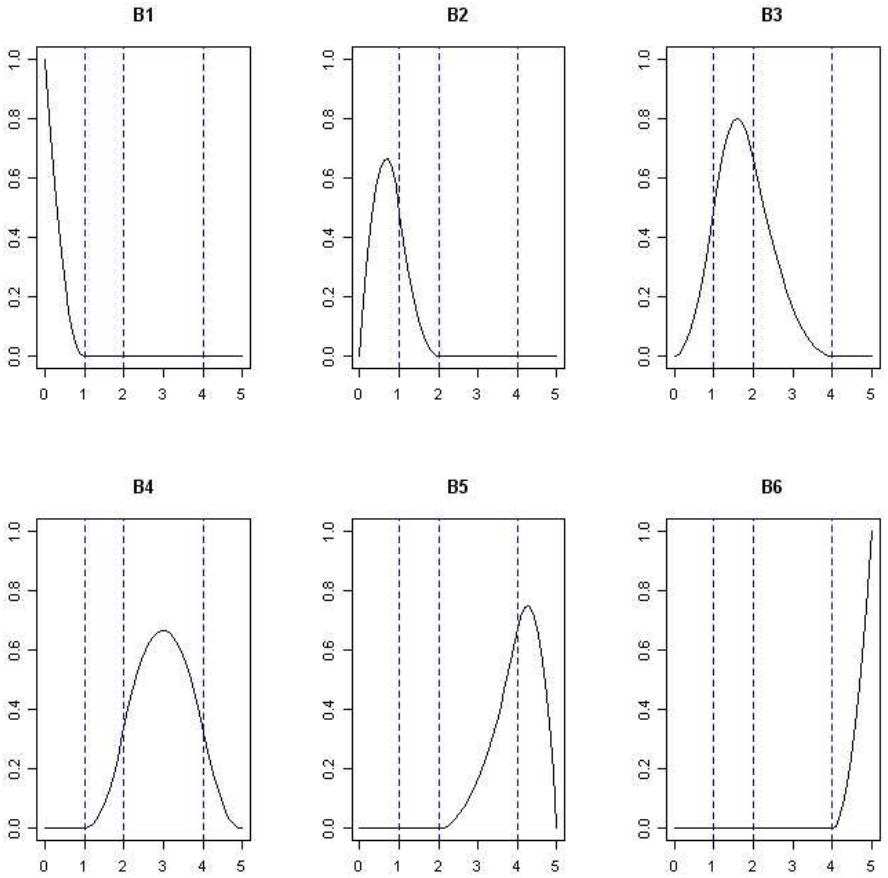
d est le degré des polynômes, K le nombre de nœuds intérieurs, dite **base des B-splines**.

- Ainsi, si l'on veut ajuster une fonction spline comme fonction dépendant de x, une fois déterminé le degré et le nombre de nœuds, il faut estimer les $\{\beta_l\}_{l=1,\ldots,r}$

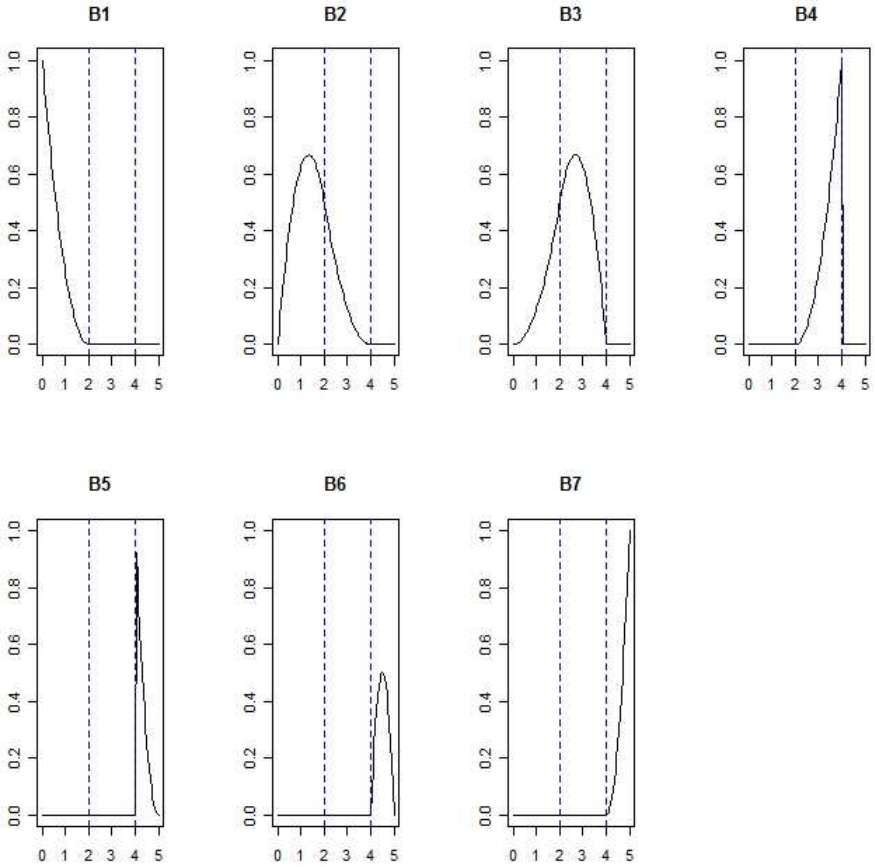
- Les splines de régression ont beaucoup d'autres propriétés mathématiques… dont on ne parlera pas !

Graphique des éléments de deux bases de B-splines de régression

$d = 2$ et $K = 3$

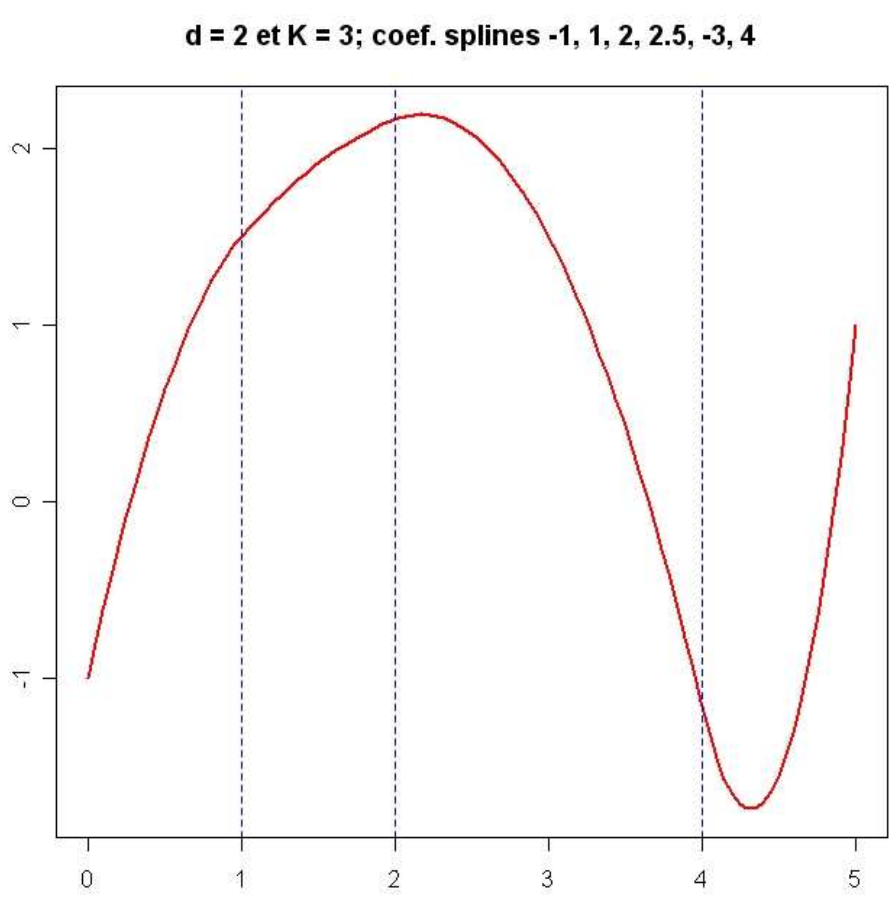


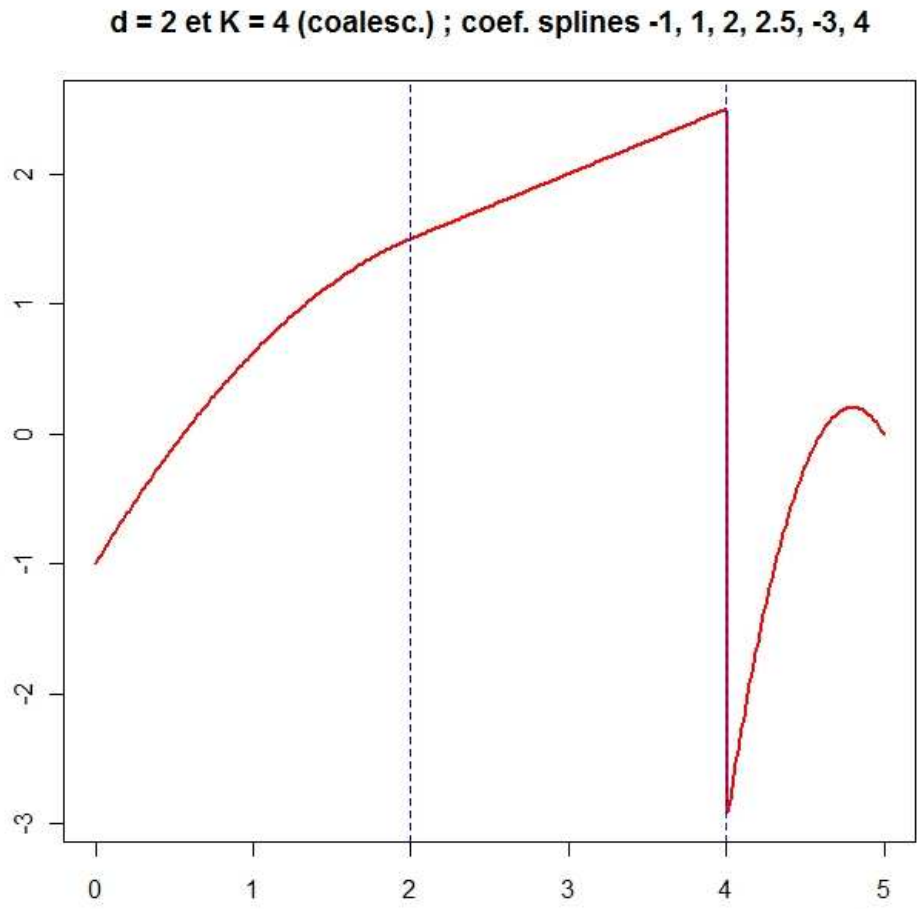
$d = 2$, $K = 4$ et **coalescence**



Graphique de fonctions spline
degré 2 avec 3 et 4 noeuds

$$s(x) = -B_1(x) + B_2(x) + 2B_3(x) + 2.5B_4(x) + 3B_5(x) + B_6(x)$$





Le problème de PLS spline

C'est, pour nous, construire un **modèle additif**, pour chaque, Y^j ($j^{\text{ème}}$ variable à expliquer), de **rang** A, en utilisant une **transformation spline de régression optimale** pour chaque variable explicative X^i.

$$Y^j = \hat{Y}_A^j + \varepsilon^j, \text{ avec } \hat{Y}_A^j = f_A^{j,1}(X^1) + f_A^{j,2}(X^2) + \ldots + f_A^{j,p}(X^p)$$

dans lequel $f_A^{j,i}(X^i) = \sum_{k=1}^{r_i} \hat{\beta}_{k,A}^{j,i} B_k^i(X^i)$, où $r_i = d_i + K_i + 1$

et $\hat{\beta}_{k,A}^{j,i}$ est le coefficient de la transformée spline, optimisé par PLSS.

Rq 1 : **Problème combinatoire « complexe »** car il y a $\sum_{i=1}^{p} r_i$ paramètres à optimiser, plus la position des « nœuds intérieurs » de chaque variable explicative et le rang A !

Rq 2 : Si les degrés et nombres de nœuds intérieurs sont fixés, c'est PLS entre Y et X, $X = \left[B_1^1, B_2^1, \ldots, B_{r_p}^p \right]$. C'est la méthode PLSS (1 ou 2).

La solution proposée
la méthode PLSS-GA

- On se propose de résoudre le problème précédent en utilisant un **Algorithme Génétique** (AG), car la **complexité** du modèle est trop importante !

- Il faut définir une fonction, le **fitness**, qui va être optimisé par l'AG :

$$fit(s) = Rv(Y, \hat{Y}_{A(s)}(s)) + a\left[\left(\alpha_1 \sum_{i=1}^{p} K_i(s) + \alpha_2\right) + \left(\beta_1 \sum_{i=1}^{p} d_i(s) + \beta_2\right)\right] + b\left(\gamma_1 A(s) + \gamma_2\right)$$

dans lequel $Rv(Z,T) = \dfrac{tr\left(ZZ^tTT^t\right)}{\sqrt{tr\left(\left(ZZ^t\right)^2\right)}\sqrt{tr\left(\left(TT^t\right)^2\right)}} \in [0,1]$ est le coefficient Rv d'Escoufier.

Dans le *fit*, les coefficients (a et b) sont choisis pour « équilibrer » les quatre composantes, mais chacune appartient à l'intervalle [0 , 1] (grâce aux $\alpha_1, \ldots, \gamma_2$).

- Le vecteur s caractérisant une solution est donné par :

$$s = \left(A, K_1, K_2, \ldots, K_p, d_1, d_2, \ldots, d_p, l_1^1, \ldots, l_1^{K_1}, NA, NA, l_2^1, \ldots, l_p^{K_p}\right)$$

position des noeuds

Les Algorithmes Génétiques (AG)

Heuristiques d'optimisation, introduits par Holland (1975).
Inspirés des mécanismes de la **sélection naturelle**.

Déroulement général d'un AG :

- **Une solution** est un vecteur (de réels) contenant les paramètres de la fitness à optimiser.
- Construction d'une **population initiale** de nombreuses (plusieurs dizaines voire centaines) solutions potentielles aussi hétérogène que possible T_{pop}.
- **Evolution de cette population** par trois mécanismes dont deux probabilistes (avec probabilités π_m, π_c), qui vont faire évoluer cette population (taille fixe) :

- mutation
- croisement } Indépendants du problème d'optimisation,

- sélection } Permet de conserver les solutions les plus intéressantes (au sens de la *fitness*).

- Récupération de la population finale quand elle a convergé (après N_{gene} itérations).

Application de PLSS1-AG sur les données de food chemistry

Paramètres de l'AG :

L'AG a été utilisé 50 fois indépendamment, avec les paramètres suivants :

. $N_{gene} = 300$,

. $T_{pop} = 200$,

. $\pi_c = 0.5$,

. $\pi_m = 0.9$.

Il y a eu convergence de la fonction *fit* à chaque fois.

Caractéristiques des cinquante solutions PLSS1-GA :

$A = 4$ dans 83% des cas,

$d = 1$ dans tous les cas, pour les trois variables,

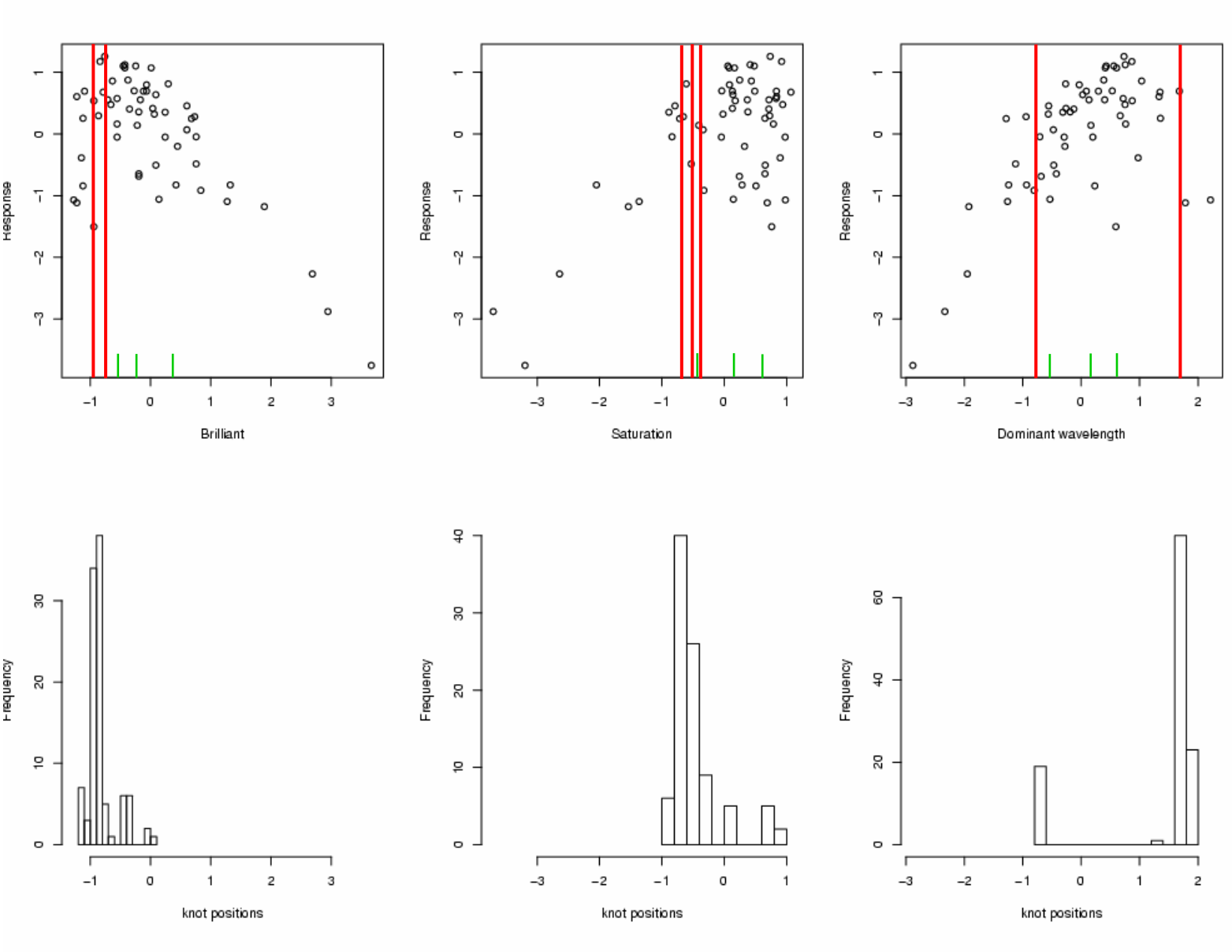
nombre de nœuds intérieurs : **bril** 2 (82% des cas), **satu** 2 ou 3, **wave** 2 ou 3.

Comparaisons de différentes méthodes :

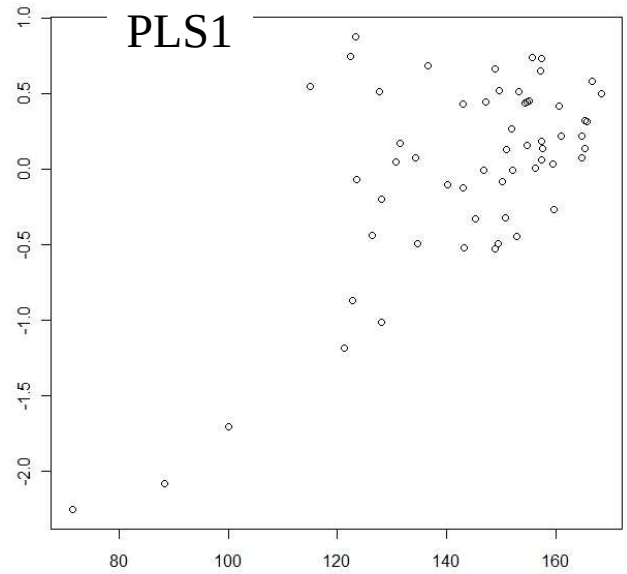
noyau Gaussien et $A = 9, \sigma = 0.03$

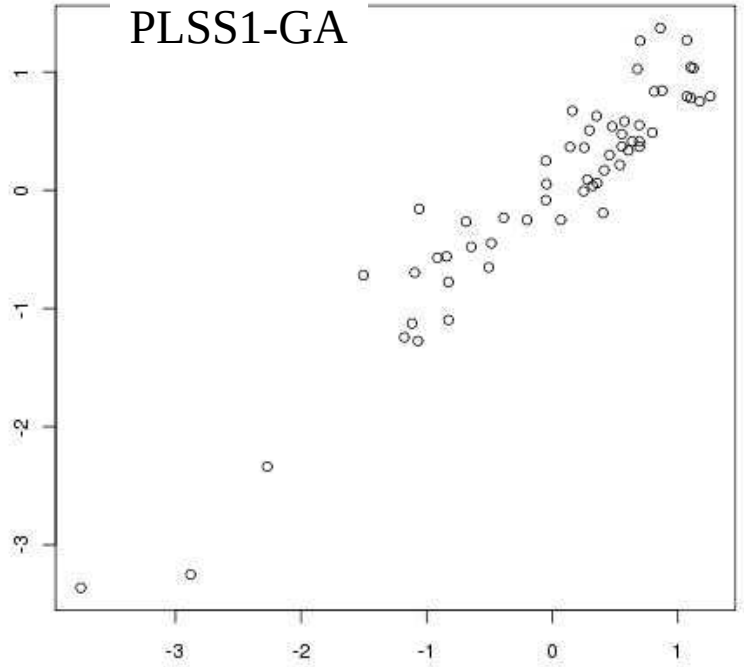
	PLSS1-GA	**OLS**	**MARS**	**ACE**	**SMART**	**CART**	**K-PLS**
R^2	0.90	0.42	0.78	0.80	0.80	0.80	0.7317
R^2_{CV}	0.85	0.26	0.72	0.69	0.69	0.62	0.7258

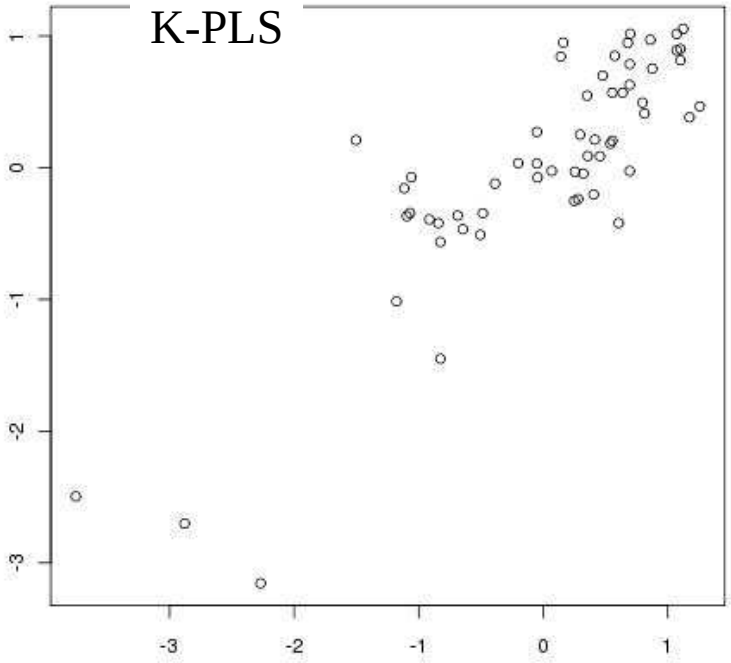
Stabilité des nœuds intérieurs des trois variables explicatives pour les 50 essais de PLSS1-GA



Graphiques de la modélisation de *Y* par PLS1
PLSS1-GA et K-PLS



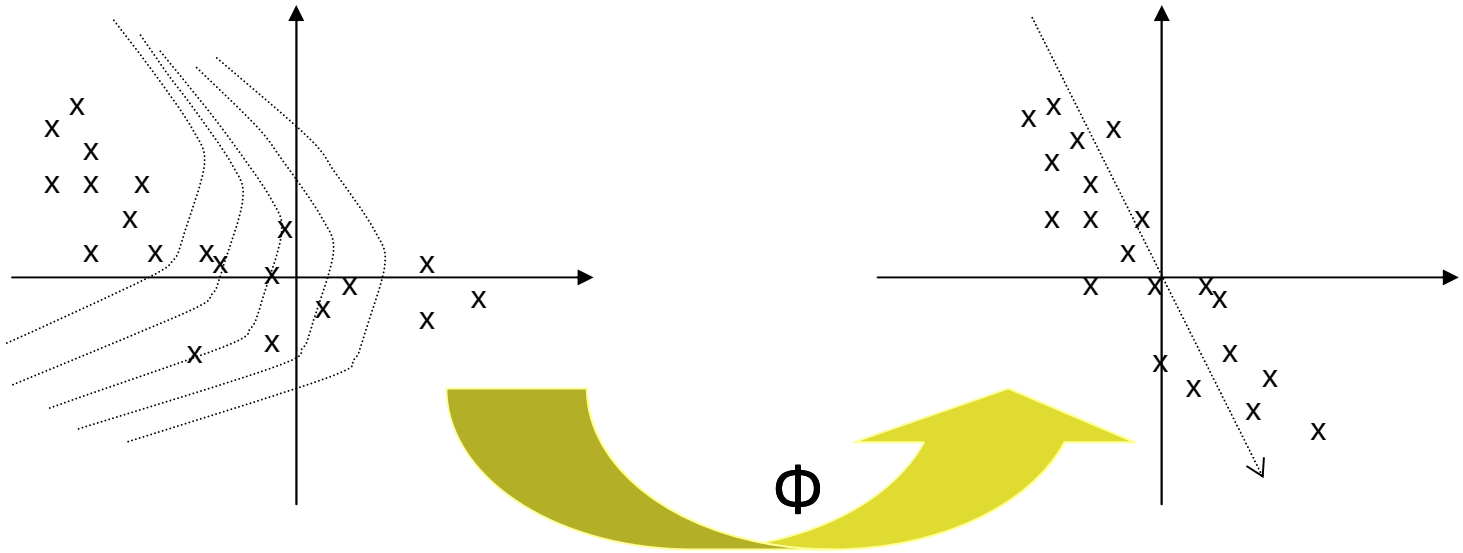




K-PLS et les noyaux (Kernel) dans les espaces auto reproduisant

- Passage dans un espace de données transformées («feature space») de grande dimension.
- Une droite dans Φ(E) donne **une courbe** dans E.
- Un **noyau de Mercer** *K* est une fonction qui réalise un produit scalaire dans cet espace.
- Soit *K* un noyau, **semi défini positif**, il existe alors :

Φ tel que : $K(x_i, x_j) = \langle \Phi(x_i), \Phi(x_j) \rangle$



B. Schölkopf & A.J. Smola (2002) *Learning with Kernels*, MITT Press, 626p.

- On peut définir un **grand nombre de noyaux**, les plus utilisés sont :

$$K(x,y) = \langle x, y \rangle^d \quad \text{noyau polynomial de degré } d$$

$$K(x,y) = \exp\left(-\frac{\|x-y\|^2}{\sigma} \right) \quad \text{noyau Gaussien (radial)}$$

$$K(x,y) = \tanh(\alpha \langle x, y \rangle + \beta) \quad \text{noyau sigmoïde}$$

$$K(x,y) = B_{2p+1}(\|x-y\|^2) \quad \text{noyau base Spline d'ordre pair}$$

- Les noyaux ont beaucoup de **propriétés mathématiques** …

- Sont très largement **utilisés en Analyse Multivariée** : SVM, K-PCA, K-LDA, K-PLS …

L'algorithme (NIPALS) K-PLS

Rosipal R., & Trejo L.J. (2001) Kernel Partial Least Squares Regression in reproducing kernel Hilbert Space, *J. of Mach. Learn. Res.*, **2**, 97-123.

NIPALS-PLS

1. u initialisé au hasard
2. $t = XX^t u$
3. $t = \frac{t}{\|t\|}$
4. $c = Y^t t$
5. $u = Yc$
6. $u = \frac{u}{\|u\|}$
7. répéter les étapes 2 à 6 jusqu'à CV
8. déflation $X = X - tt^t X$ et $Y = Y - tt^t Y$

NIPALS-K-PLS

1. u initialisé au hasard
2. $t = Ku$
3. $t = \frac{t}{\|t\|}$
4. $c = Y^t t$
5. $u = Yc$
6. $u = \frac{u}{\|u\|}$
7. répéter les étapes 2 à 6 jusqu'à CV
8. déflation $K = (I - tt^t)K(I - tt^t)$ et $Y = Y - tt^t Y$

- Pour K-PLS, détermination du **nombre de composantes** et du (ou des) **paramètre**(s) du noyau par VC.

Une petite simulation pour comparer K-PLS et PLSS1-GA

$$Y = 4.26(\exp(-X) - 4\exp(-2X) + 3\exp(-3X)) + \varepsilon$$

avec, n = 100, $X \sim U[0, 2.5]$ et $\varepsilon \sim N(0, 0.04)$

Paramètres de PLSS1-AG :

. N_{gene} = 100,
. T_{pop} = 100,
. π_c = 0.5,
. π_m = 0.9.

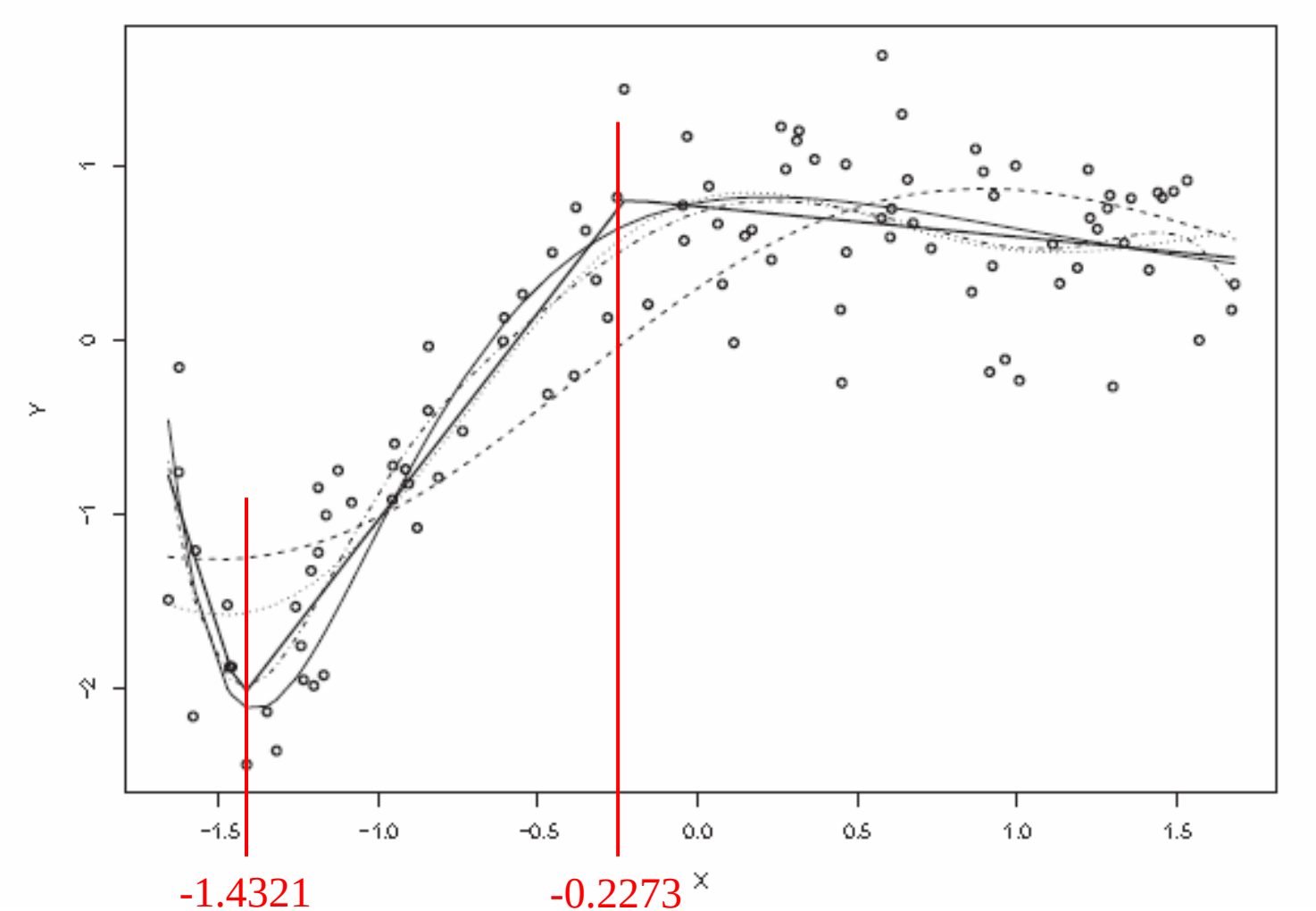
10 « runs », convergence chaque fois.

Paramètres de K-PLS :

. Noyau Gaussien,
. σ = 1.8.

Solution : A = 1, d = 1 et 2 noeuds

Sorties graphiques pour la simulation et les solutions

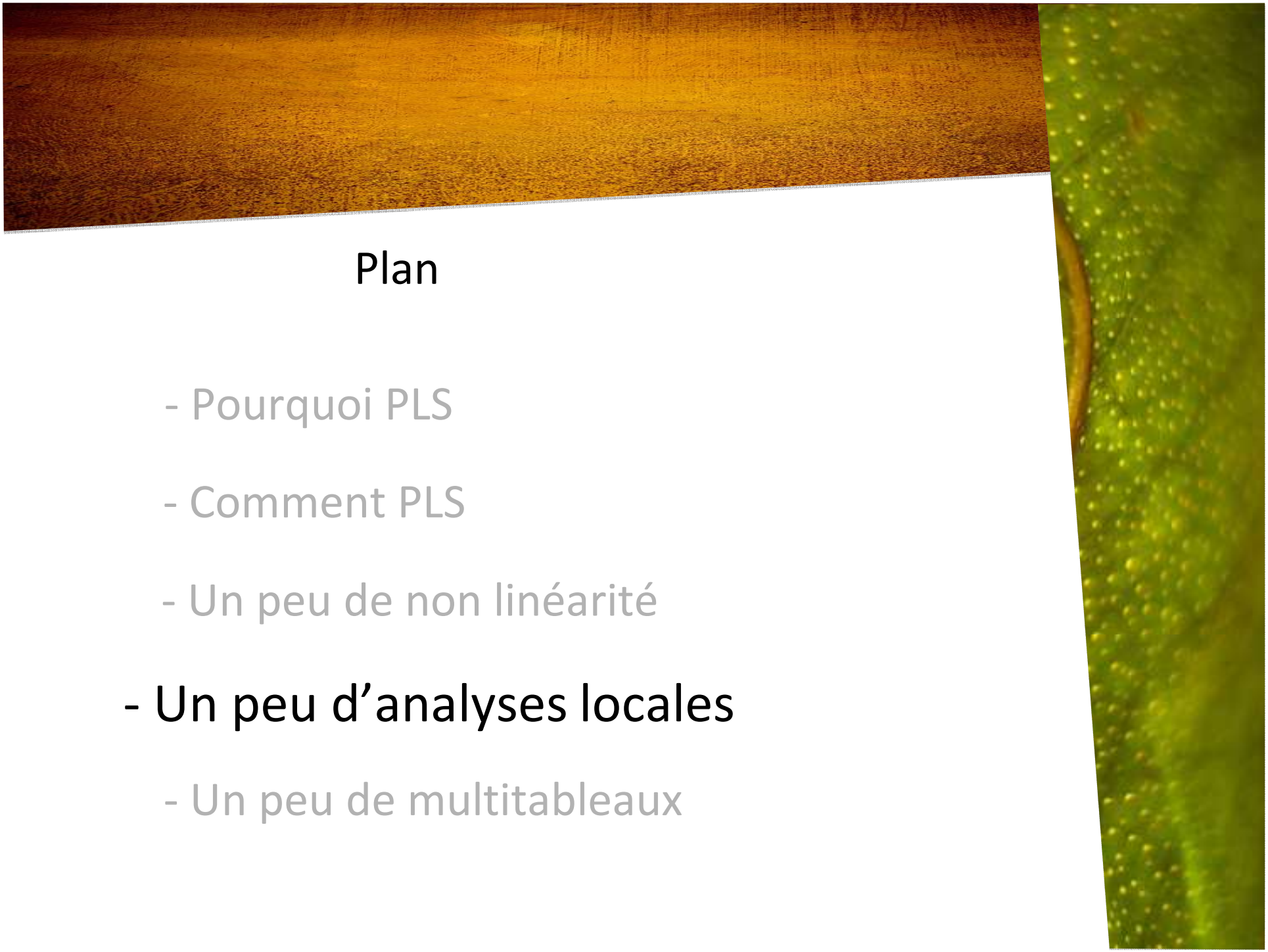


Rv PLSS1-GA = 0.8323

Rv K-PLS 1 = 0.7033

Rv K-PLS 4 = 0.8054

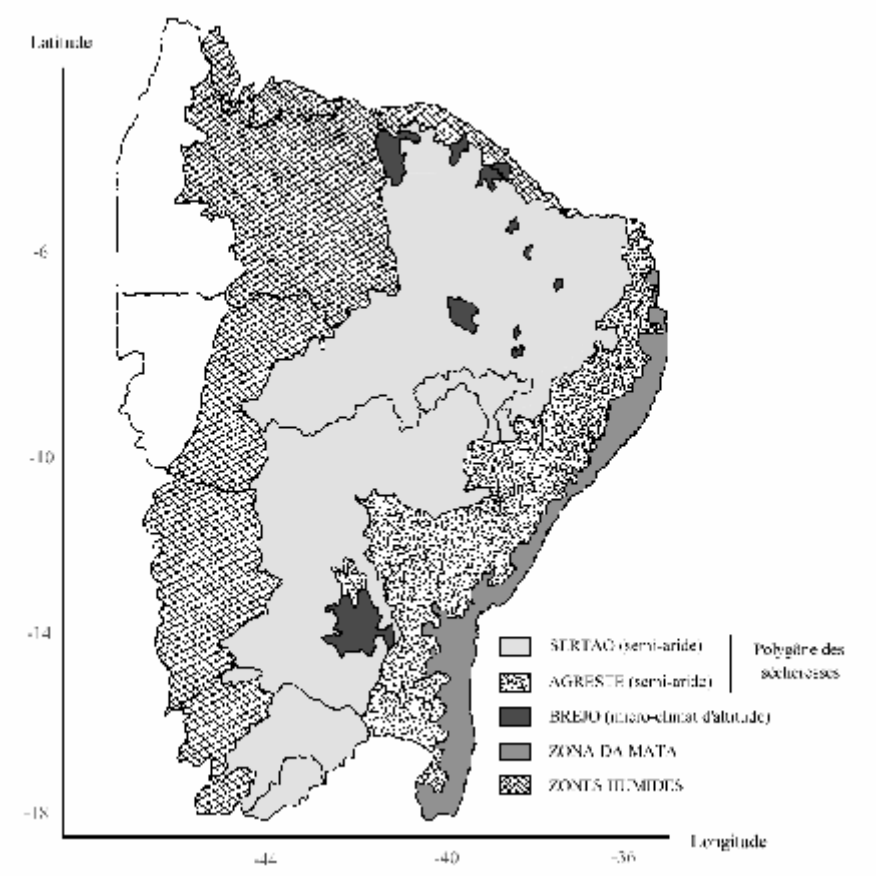
Rv K-PLS 8 = 0.8367

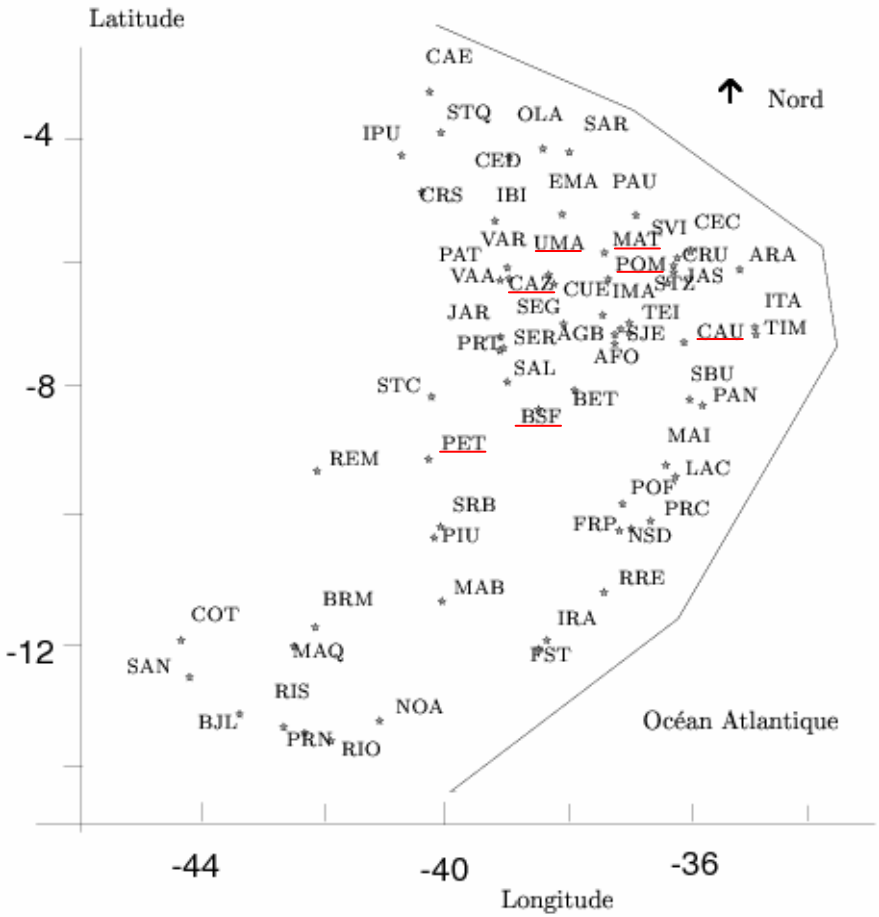


Plan

- Pourquoi PLS
- Comment PLS
- Un peu de non linéarité
- Un peu d’analyses locales
- Un peu de multitableaux

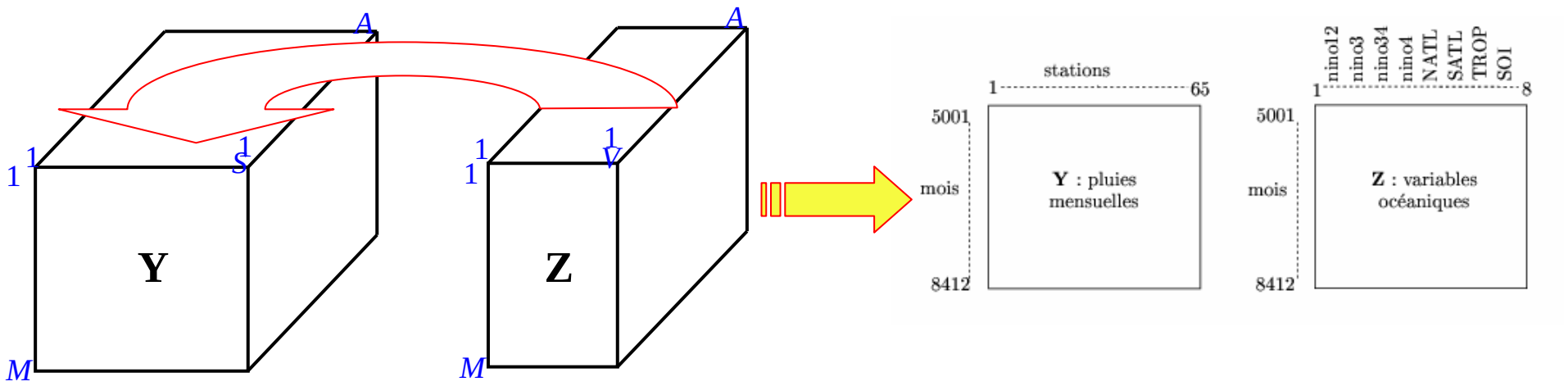
Prédire les précipitations du Nordeste Brésilien





Prédire les précipitations, entre 01/1950 et 12/1984, mesurées mensuellement sur 65 stations du Nordeste Brésilien, en fonction des Températures Superficielles de Mer (pour 7 zones) et de l'Indice d'Oscillation Australe (SOI) (donc 8 variables océaniques au total).
Prédiction complexe, à cause d'une grande irrégularité spatio-temporelle, zone semi-aride, soumis à périodiquement à des épisodes de sécheresse grave (20% de la superficie du Brésil).

Les données



choisies parmi les 65

a prédire $Y(408 \times 7)$

34 ans × 12 mois

$Y_c(89 \times 7)$ calibration

$Y_v(15 \times 7)$ validation

Q1 : février à mai

Q2 : juin à septembre

Q3 : octobre à janvier

} 89 + 15 « saisons »

8 prédicteurs $Z(408 \times 8)$

$Z_c(89 \times 8)$ calibration

$Z_v(15 \times 8)$ validation

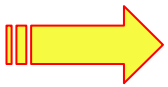
Matrice des corrélations entre les huit variables explicatives pour l'échantillon d'apprentissage

	SOI	nino12	nino3	nino4	nino34	nATL	SATL	TROP
SOI	1.00	−0.29	−0.53	−0.83	−0.78	−0.14	−0.01	−0.44
nino12	−0.29	1.00	0.88	0.20	0.56	−0.78	0.88	0.91
nino3	−0.53	0.88	1.00	0.51	0.86	−0.55	0.68	0.91
nino4	−0.83	0.20	0.51	1.00	0.86	0.27	−0.06	0.45
nino34	−0.78	0.56	0.86	0.86	1.00	−0.11	0.29	0.72
nATL	−0.14	−0.78	−0.55	0.27	−0.11	1.00	−0.89	−0.64
SATL	−0.01	0.88	0.68	−0.06	0.29	−0.89	1.00	0.85
TROP	−0.44	0.91	0.91	0.45	0.72	−0.64	0.85	1.00

MAIS

- Quelques corrélations très (trop !?) élevées : nino12 et 3 avec TROP = 0.91 et nATL avec SATL = - 0.89 …

- Forte dépendance temporelle (la spatiale a été « éliminée » ?!)



PLS s'impose, en tenant compte de la dépendance locale !

La régression locale

Mise au point par Cleveland (79) puis Cleveland et Devlin (88). La méthode LOcally Weighted Scatterplot Smoothing (LOWESS ou LOESS), est une généralisation de la régression polynomiale de degré r, en x, qui, en chaque point expérimental x_0 utilise une pondération, fonction de la distance des points voisins, dans le but de prédire $y(x_0)$.

La prédiction en x_0 est de la forme : $\hat{y}(x_0) = \hat{\beta}_0(x_0) + \sum_{k=1}^{r} \hat{\beta}_k(x_0) x_0^k$

où $\left\{\hat{\beta}_k(x_0)\right\}_{k=0,1,\ldots,r}$ est solution du problème :

$$\min_{\{\beta_k(x_0)\}_{k=0,1,\ldots,r}} \left(\sum_{i=1}^{n} w_i(x_0) \left\| y_i - \beta_0(x_0) - \sum_{k=1}^{r} \beta_k(x_0) x_i^k \right\|^2 \right)$$

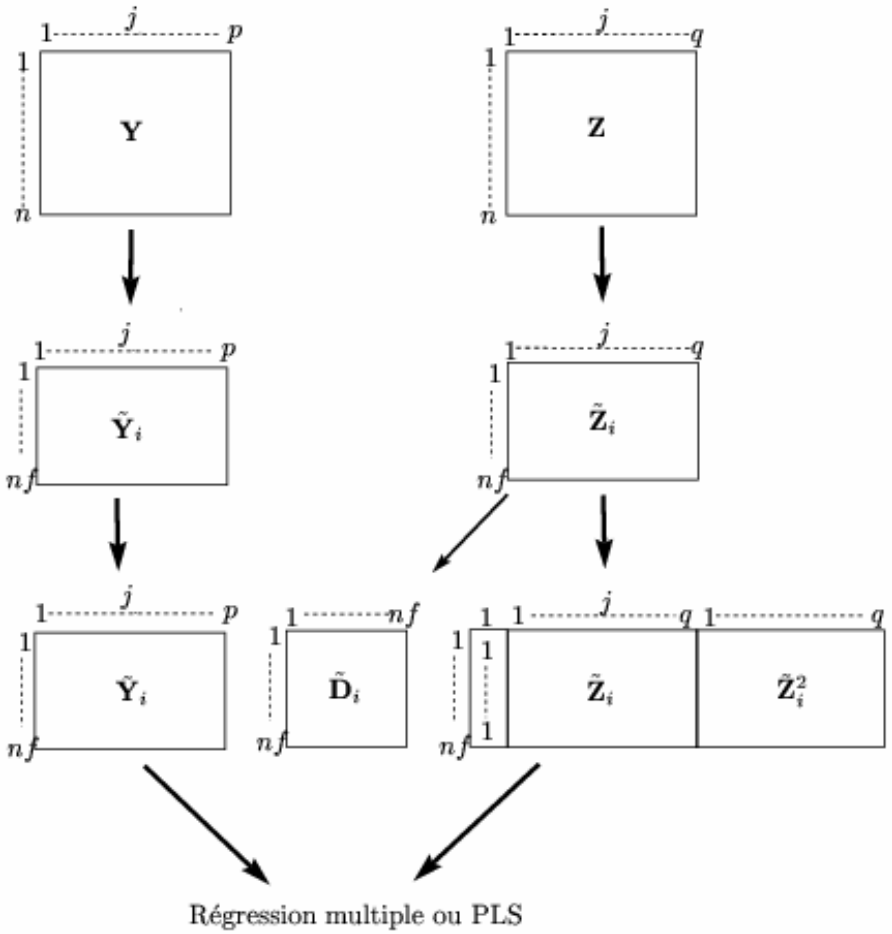
dans cette équation la pondération de l'observation i, $w_i(x_0)$, est calculée avec la fonction tricube (ou autre : Epanechnikov, Gaussien …) définie par :

$$w_i(x_0) = \chi_{\{\|x_i - x_0\| \le d_f\}} \times \left(1 - \left\| \frac{x_i - x_0}{d_f} \right\|^3 \right)^3$$

Remarque : souvent (et cela sera notre façon de procéder), on prends une proportion d'observations f (en général $0.25 \le f \le 0.50$) comme voisins, au lieu de prendre une borne d_f pour les distances.

Régression PLS locale (LPLS1 ou LPLS2)

Etape 0	Initialisation : choix du paramètre de lissage f, du degré du polynôme r, de la fonction de poids w (et du nombre a de composantes) ; $i = 1$.
Etape 1	Détermination des voisins de la ligne $\mathbf{Z}_i$ (en utilisant le paramètre de lissage f). On calcule pour cela toutes les distances euclidiennes entre $\mathbf{Z}_i$ et les autres lignes, et on conserve les nf lignes les plus proches.
Etape 2	Construction de $\tilde{\mathbf{Z}}_i$ regroupant les nf lignes voisines de $\mathbf{Z}_i$, et de $\tilde{\mathbf{Y}}_i$ regroupant les nf lignes de $\mathbf{Y}$ correspondantes.
Etape 3	Calcul de la matrice $nf \times nf$ de poids $\tilde{\mathbf{D}}_i$ en utilisant la fonction tricube appliquée sur les distances normalisées entre $\mathbf{Z}_i$ et toutes les lignes voisines.
Etape 4	Régression multivariée (PLS) de $\tilde{\mathbf{Y}}_i$ par $\tilde{\mathbf{Z}}_i$ en utilisant la métrique $\tilde{\mathbf{D}}_i$ (et a composantes). Obtention de la matrice des coefficients de régression $\widehat{\beta}_i$, de dimensions $q \times p$.
Etape 5	$\widehat{Y}_i = \mathbf{Z}_i \widehat{\beta}_i$.
Etape 6	Si $i = n$ fin, sinon $i = i + 1$ et revenir à l'étape 1.



Remarque : ici, en pratique, *r* est égal à 1.

Comparaison PLS1, LOWESS et LPLS1 pour l'échantillon d'apprentissage avec le PRESS

Station	PLS (A)	LOWESS ($\nu = nf$)	LPLS1 ($\nu = nf, A$)
MAT	0.138 ($a = 5$)	0.154 ($\nu = 57$)	0.133 ($\nu = 44, a = 2$)
CAU	0.459 ($a = 5$)	0.507 ($\nu = 56$)	0.458 ($\nu = 36, a = 1$)
BSF	0.617 ($a = 3$)	0.616 ($\nu = 63$)	0.609 ($\nu = 63, a = 3$)
PET	0.514 ($a = 5$)	0.545 ($\nu = 81$)	0.541 ($\nu = 84, a = 4$)
POM	0.203 ($a = 4$)	0.196 ($\nu = 37$)	0.166 ($\nu = 39, a = 2$)
UMA	0.156 ($a = 4$)	0.168 ($\nu = 61$)	0.151 ($\nu = 43, a = 3$)
CAZ	0.179 ($a = 4$)	0.186 ($\nu = 51$)	0.168 ($\nu = 49, a = 4$)

Pour PLS1 : A est déterminé à l'aide du PRESS par validation croisée.
Pour LOWESS et LPLS1 : $\nu = nf$ varie entre 9 et 89 et on choisi le meilleur résultat du PRESS.

Rq : de nombreux autres essais ont été réalisés (LPLS2, $r = 2$, pas de regroupement par saison …)

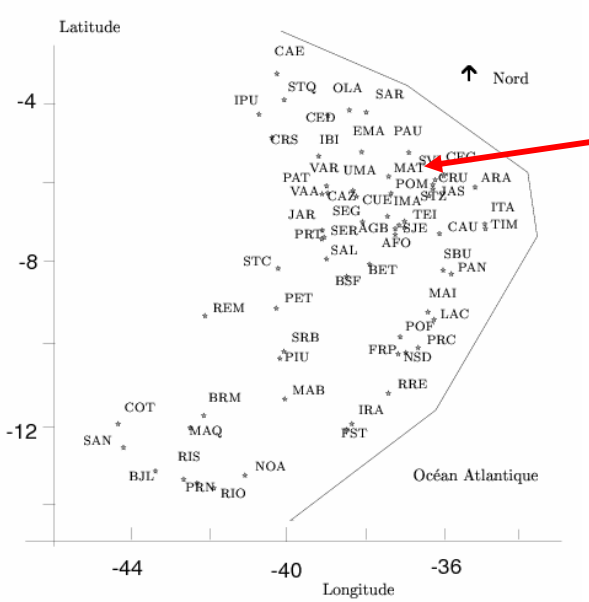
Comparaison PLS1, LOWESS et LPLS1 pour l'échantillon test avec le MSEP

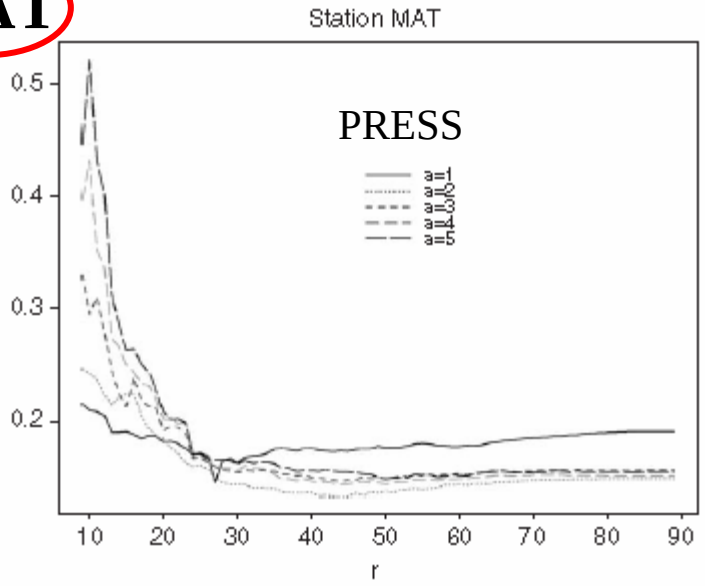
Station	PLS (A)	LOWESS ($\nu = nf$)	LPLS1 ($\nu = nf, A$)
MAT	0.126 ($a = 5$)	0.096 ($\nu = 33$)	0.071 ($\nu = 35, a = 2$)
CAU	0.459 ($a = 2$)	0.240 ($\nu = 24$)	0.178 ($\nu = 13, a = 2$)
BSF	0.225 ($a = 2$)	0.247 ($\nu = 74$)	0.161 ($\nu = 17, a = 1$)
PET	0.313 ($a = 3$)	0.611 ($\nu = 80$)	0.412 ($\nu = 49, a = 1$)
POM	0.124 ($a = 5$)	0.100 ($\nu = 51$)	0.068 ($\nu = 32, a = 2$)
UMA	0.162 ($a = 4$)	0.189 ($\nu = 82$)	0.112 ($\nu = 20, a = 2$)
CAZ	0.110 ($a = 4$)	0.064 ($\nu = 39$)	0.041 ($\nu = 15, a = 1$)

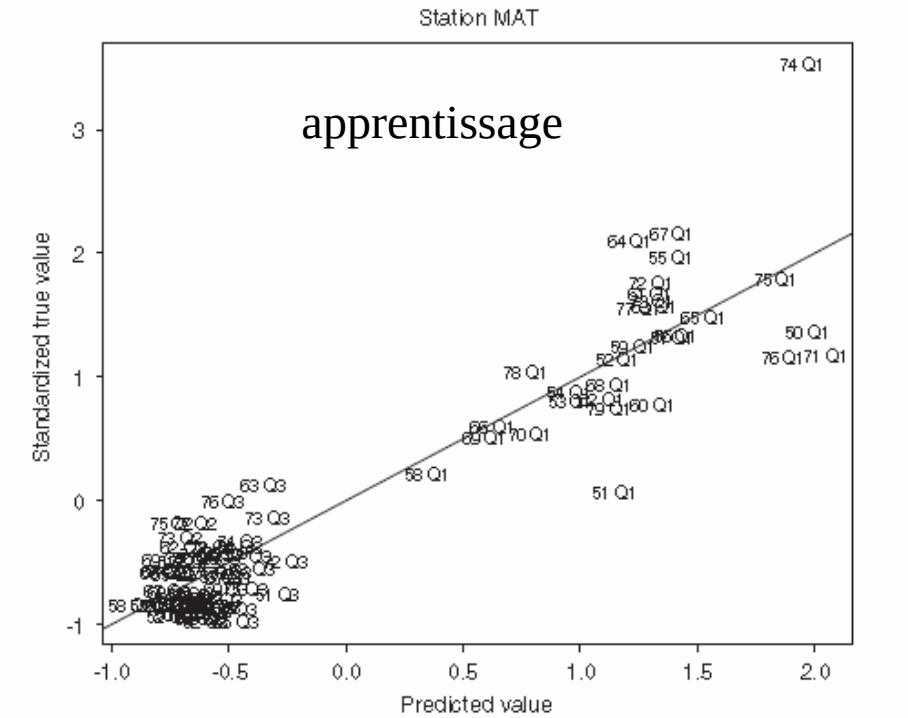
$$MSEP = \sum_{i=1}^{n} \left(y_i - \hat{y}_i \right)^2$$

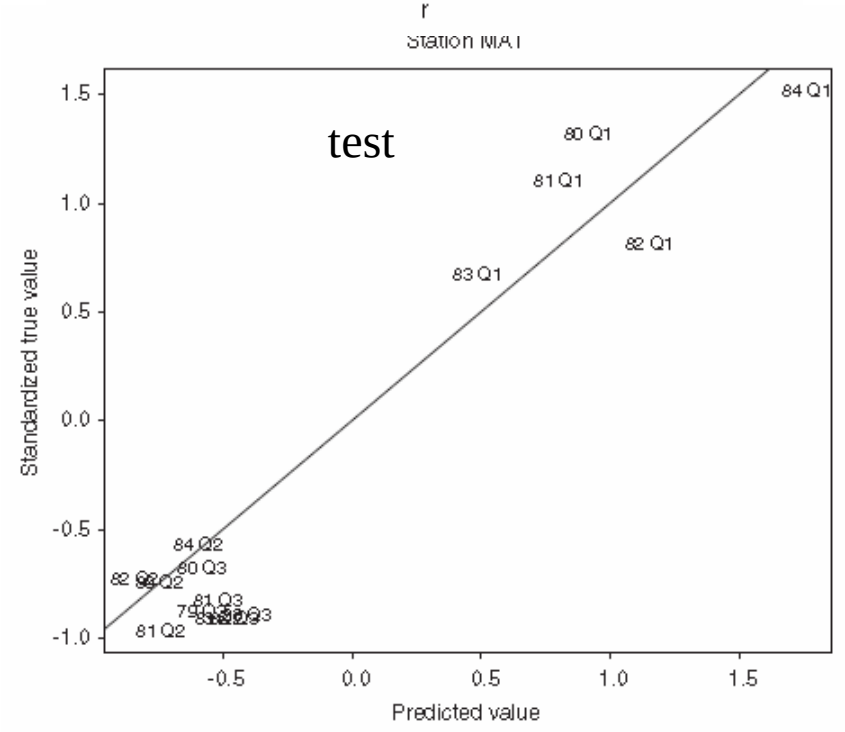
où y_i est la valeur de la variable à prédire donnée par l'échantillon test et $\hat{y}_i$ la valeur prédite par le modèle à l'aide des variables explicatives.

Quelques sorties de LPLS1 pour la station : MAT









Plan

- Pourquoi PLS
- Comment PLS
- Un peu de non linéarité
- Un peu d’analyses locales
- Un peu de multitableaux

Chimiométrie et tabac

- Pour 24 tabacs, issus de 9 **origines géographiques différentes**, dans le but de prédire la **qualité des tabacs**, on a mesuré les **blocs de variables** suivants :

  Les **3 blocs « prédicteurs »**

  50 variables chimiques : bloc X_1,
  29 variables de thermolyse : bloc X_2,
  1500 variables des spectres proche infrarouge : bloc X_3.

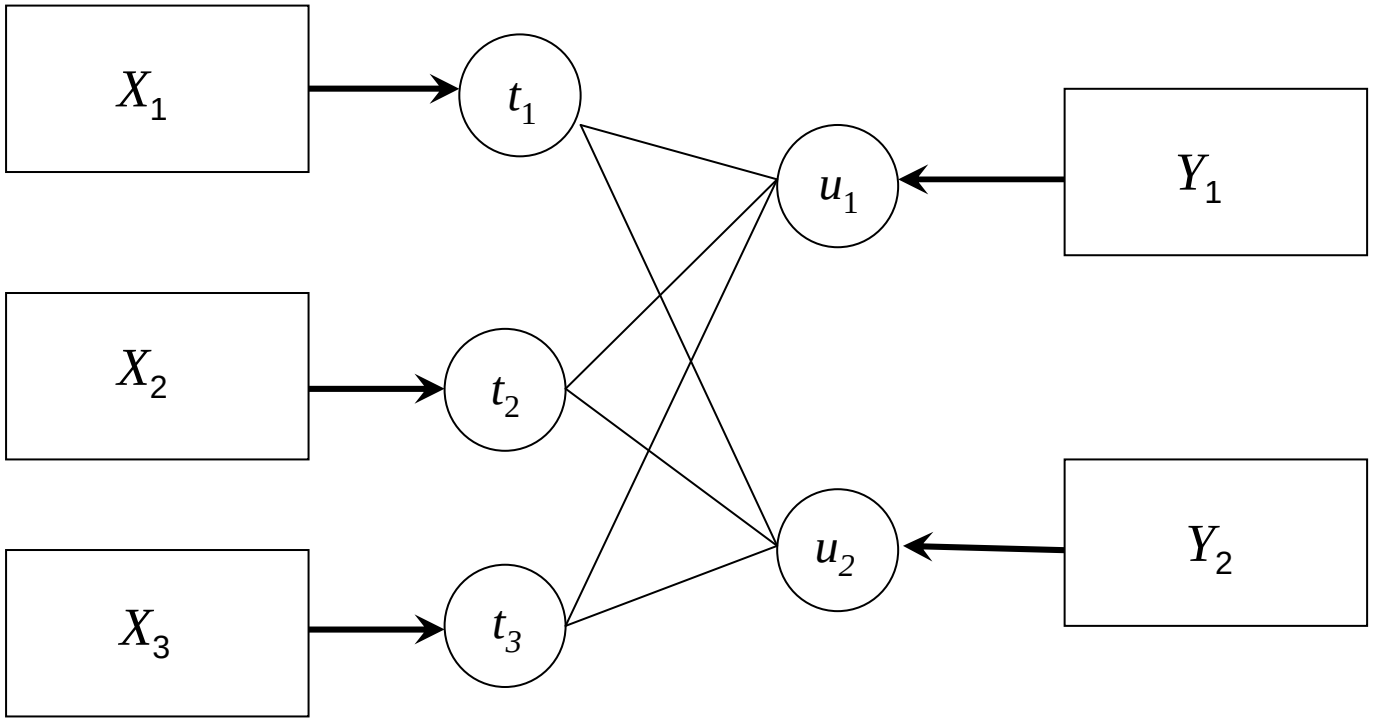
  Pour prédire les **2 blocs « réponse »**

  2 variables chimiques : bloc Y_1,
  4 autres variables chimiques : bloc Y_2.

- Données confidentielles issues de *Altadis*©, Orléans (maintenant *Impérial Tobacco*©).

- Prétraitements : centrées et normées (var usuelles)
  et débruitées par OSC pour les spectres.

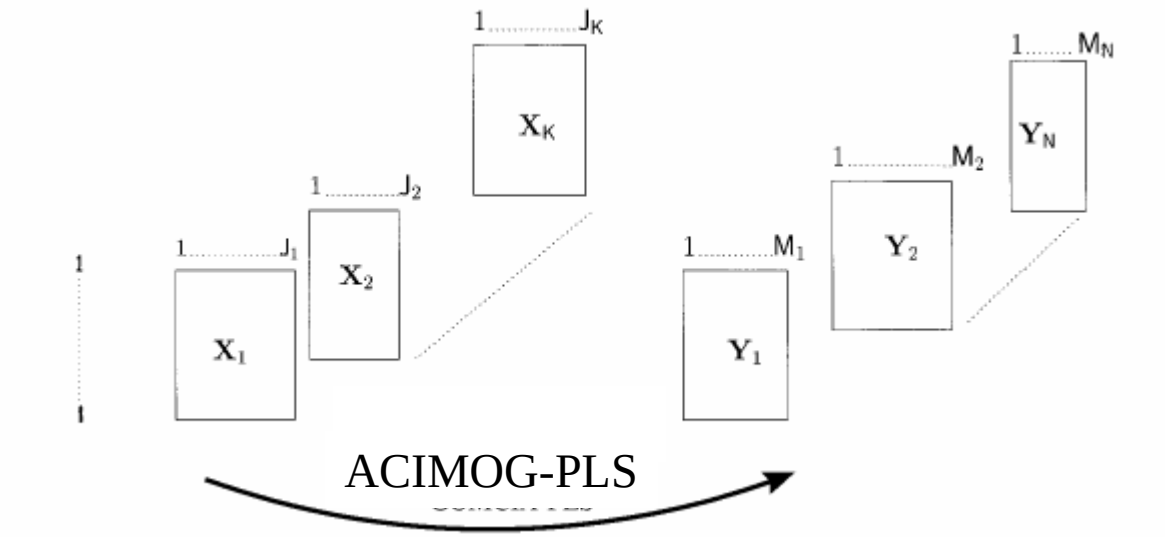
Graphe utilisant l'écriture en variables latentes du « modèle PLS » de H. Wold (1975)



On cherche à **maximiser** : $\operatorname{cov}\left(\sum_{k=1}^{3} t_k, \sum_{n=1}^{2} u_n\right) = \sum_{k=1}^{3}\sum_{n=1}^{2} \operatorname{cov}\left(t_k, u_n\right)$

Puis, avec les *t*, **modéliser** Y_1 et Y_2

La méthode ACIMOG-PLS



Le problème est donc :

$$\max_{\{a_k\},\{b_n\}} \left\{ \sum_{k=1}^{K} \sum_{n=1}^{N} \operatorname{cov}\left(t_k = X_k a_k, u_n = Y_n b_n\right) \right\} = \max_{\{a_k\},\{b_n\}} \left\{ \operatorname{cov}\left(\sum_{k=1}^{K} t_k, \sum_{n=1}^{N} u_n \right) \right\}$$

sous les contraintes :

$$a_k^t a_k = \left\| a_k \right\|^2 = 1 \text{ et } b_n^t b_n = \left\| b_n \right\|^2 = 1$$

Rq : dans l'exemple, $K = 3$ et $N = 2$.

L'algorithme ACIMOG-PLS

- obtenu en écrivant le Lagrangien, en annulant les dérivées partielles et en « arrangeant ».

- ce n'est pas une diagonalisation, ni MBPLS (attention 2 différentes) de Wold (1987) ni HPLS

- sa convergence a été montrée (maximum local).

- si $K = N = 1$, on retrouve PLS ordinaire.

- une fois déterminé les t et les u, on réalise une déflation, mais il y a plusieurs choix :

	GOMCIA-PLS1	GOMCIA-PLS2	GOMCIA-PLS3
$\mathbf{X}_k$	$\mathbf{t}_a$	$\mathbf{t}_a = \sum_k \mathbf{t}_{k,a}$	$\sum_k \mu_{k,a}\mathbf{t}_{k,a}$
$\mathbf{Y}_n$	$\mathbf{t}_a$	$\mathbf{t}_a = \sum_k \mathbf{t}_{k,a}$	$\sum_k \mu_{k,a}\mathbf{t}_{k,a}$

- on peut écrire les modèles pour les Y, réaliser la validation croisée et calculer des BIP (généralisation des VIP de PLS).

Step 0: initialization

*$\mathbf{X}_k$, $k = 1,\ldots,K$, and $\mathbf{Y}_n$, $n = 1,\ldots,N$, are supposed correctly preprocessed.

Fix $\mathbf{b}_n$ as the first left singular vector of unit length of $\mathbf{Y}_n'\mathbf{DX}$.

(It is also possible to fix it to $(1/\sqrt{N_m})\,\mathbf{1}_{N_m}$.)

$\mathbf{a}_k = \mathbf{X}_k'\mathbf{DYb}$, where $\mathbf{b}' = [\mathbf{b}_1'|\ldots|\mathbf{b}_N']$.

$\mathbf{a}_k = \mathbf{a}_k/\|\mathbf{a}_k\|$.

$\mu_k = \mathbf{b}'\mathbf{Y}'\mathbf{DX}_k\mathbf{a}_k = \Sigma_{n=1}^{N}\mathbf{u}_n'\mathbf{Dt}_k$.

$\lambda_n = \mathbf{a}'\mathbf{X}'\mathbf{DY}_n\mathbf{b}_n = \Sigma_{k=1}^{K}\mathbf{u}_n'\mathbf{Dt}_k$, where $\mathbf{a}' = [\mathbf{a}_1'|\ldots|\mathbf{a}_K']$.

$C_{opt} = \Sigma_k \mu_k$.

$iter = 0$, index of iterations.

Step 1: current step

$iter = iter + 1$.

$\mathbf{b}_n = \mathbf{Y}_n'\mathbf{DXa}$.

$\mathbf{b}_n = \mathbf{b}_n/\|\mathbf{b}_n\|$.

$\mathbf{a}_k = \mathbf{X}_k'\mathbf{DYb}$, where $\mathbf{b}' = [\mathbf{b}_1'|\ldots|\mathbf{b}_N']$.

$\mathbf{a}_k = \mathbf{a}_k/\|\mathbf{a}_k\|$.

$\mu_k = \mathbf{b}'\mathbf{Y}'\mathbf{DX}_k\mathbf{a}_k = \Sigma_{n=1}^{N}\mathbf{u}_n'\mathbf{Dt}_k$.

$\lambda_n = \mathbf{a}'\mathbf{X}'\mathbf{DY}_n\mathbf{b}_n = \Sigma_{k=1}^{K}\mathbf{u}_n'\mathbf{Dt}_k$, where $\mathbf{a}' = [\mathbf{a}_1'|\ldots|\mathbf{a}_K']$.

$C_{opt}^{new} = \Sigma_k \mu_k$.

Step 2: convergence test

If $C_{opt}^{new} - C_{opt} < 10^{-6}$ or if $iter = 50$, go to Step 3.

Else set $C_{opt} = C_{opt}^{new}$ and go to Step 1.

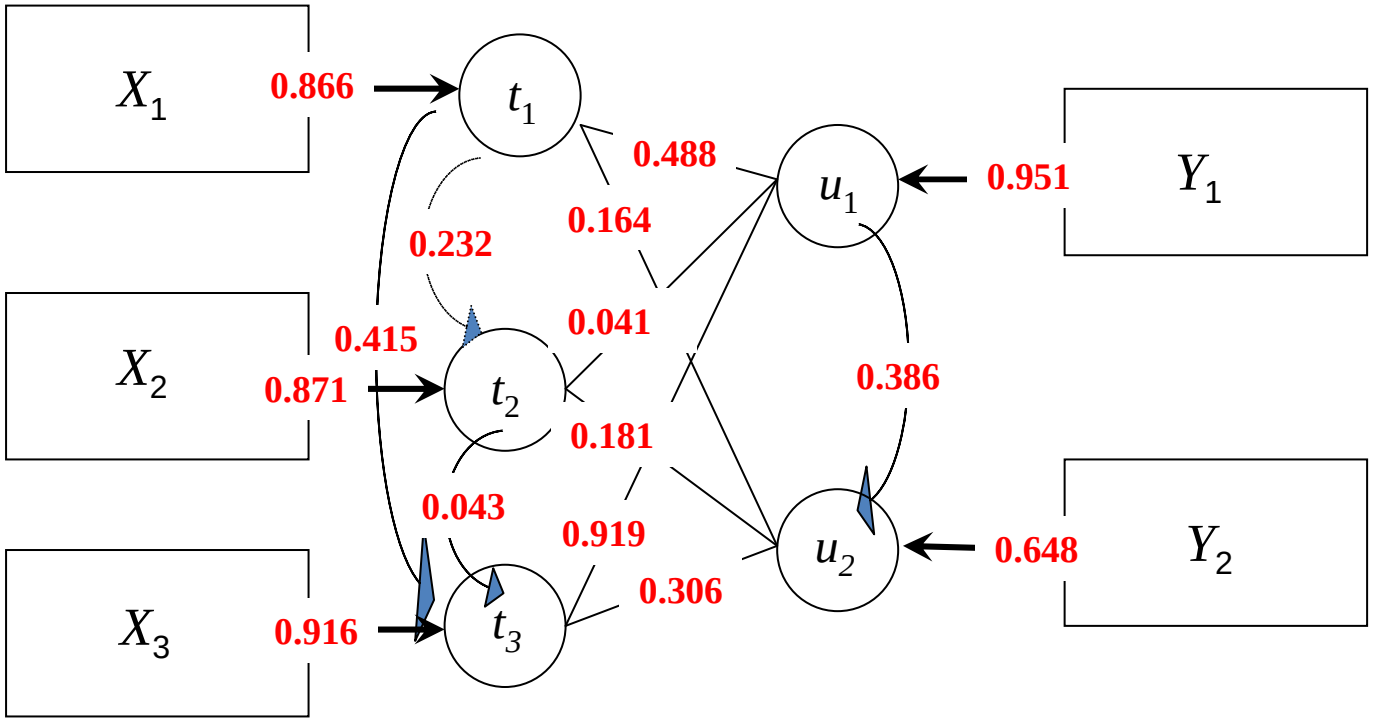
Step 3: end

$n = 1,\ldots,N$, $\mathbf{u}_n = \mathbf{Y}_n\mathbf{b}_n$,

$k = 1,\ldots,K$, $\mathbf{t}_k = \mathbf{X}_k\mathbf{a}_k$.

Résultat pour les données de tabac avec ACIMOG-PLS1

(en cor^2 pas le critère)



Résultats des ACIMOG-PLS pour les données de tabac

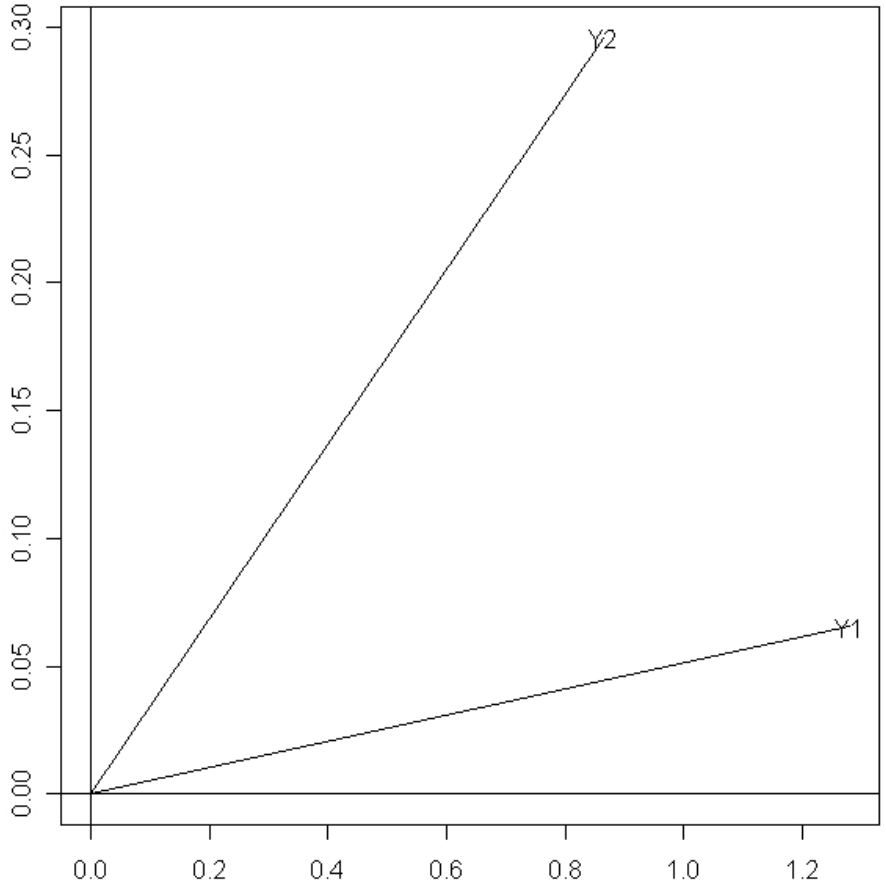
	% inertie expliquée			
bloc	Y_1		Y_2	
n° comp.	1	2	1	2
ACIMOG-PLS1	90.45	92.52	41.92	62.11
ACIMOG-PLS2	62.66	90.16	28.55	36.42
ACIMOG-PLS3	81.20	91.08	28.94	37.65
HPLS	0.40	9.76	14.05	16.74
MPLS	86.55	89.83	30.72	37.46
PLS	86.05	90.65	29.95	37.30

meilleur résultat par VC

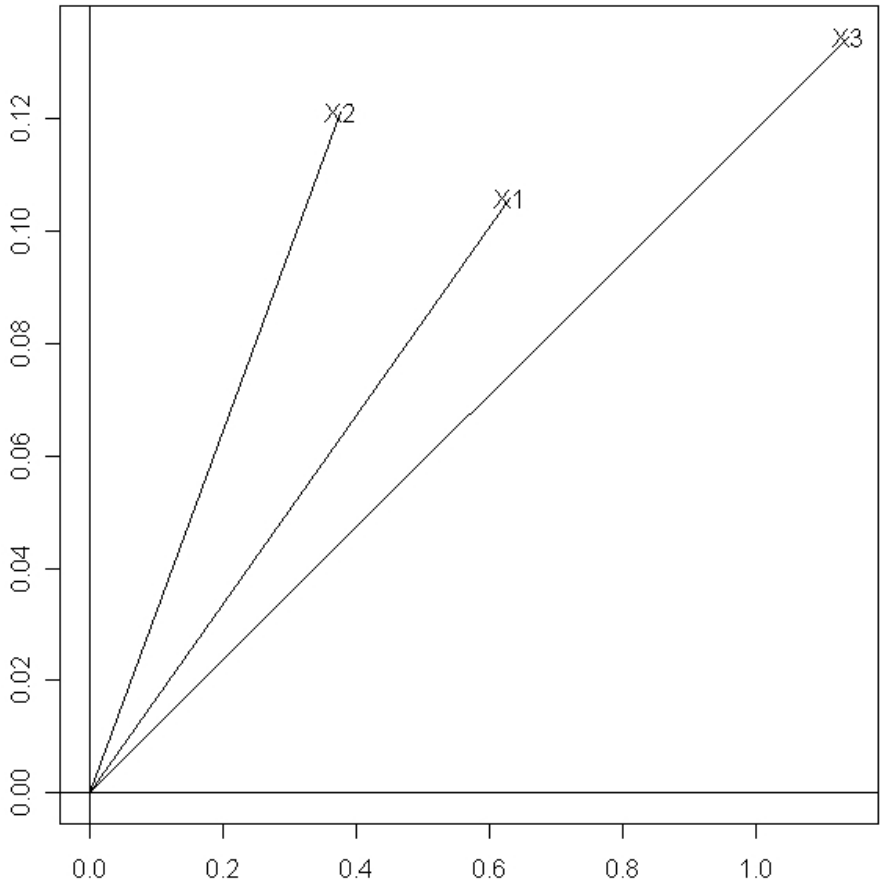
Représentations graphique pour l'ACIMOG-PLS1 (1)

- Représentation de chaque bloc :

Y_n représenté pour l'axe a par $\text{cov}\left(\sum_{k=1}^{K} t_{k,a}, u_{n,a}\right)$

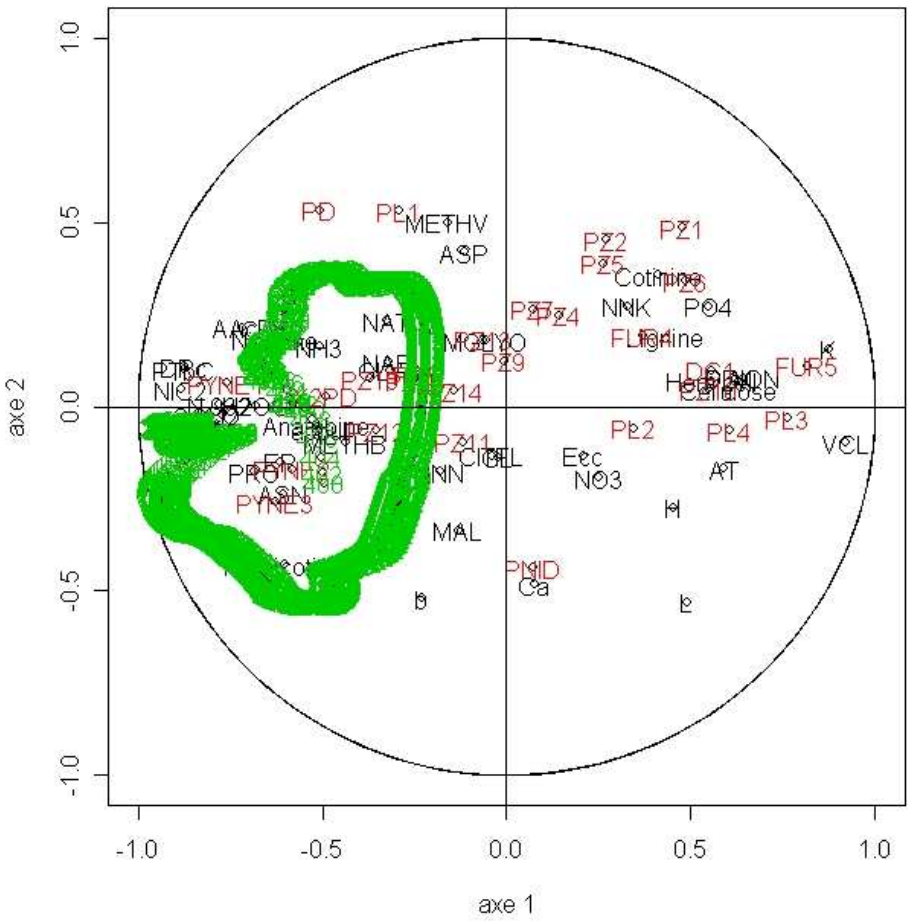


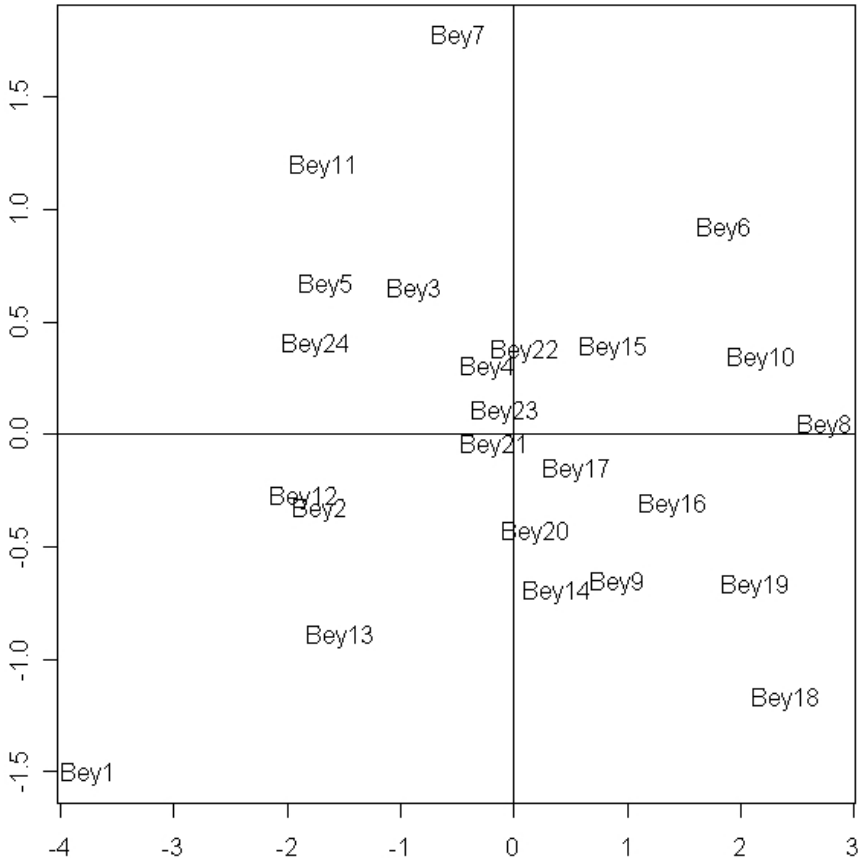
X_k représenté pour l'axe a par $\text{cov}\left(t_{k,a}, \sum_{n=1}^{N} u_{n,a}\right)$



Représentations graphique pour l'ACIMOG-PLS1 (2)

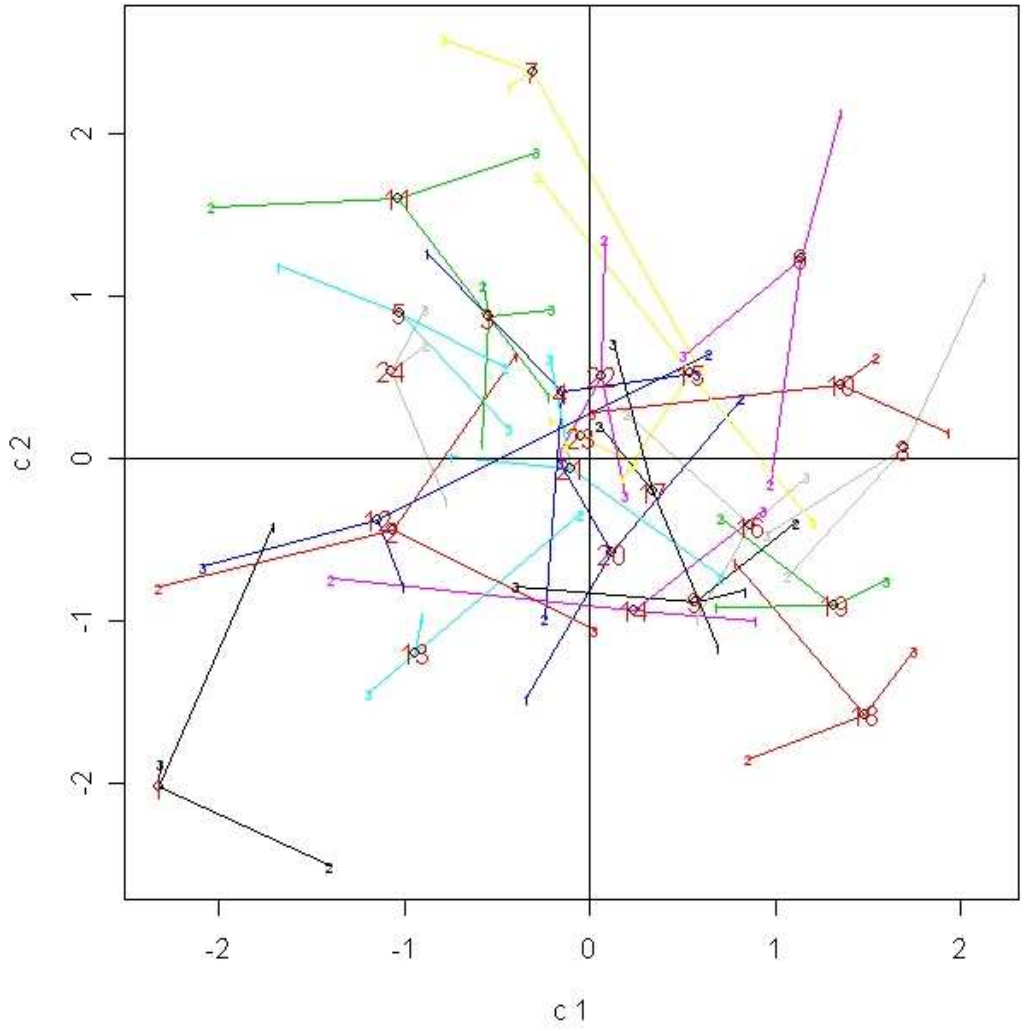
- Pour les deux premières « super composantes » (orthogonales): $t_a = \sum_{k=1}^{K} t_{k,a}$





Représentations graphique pour l'ACIMOG-PLS1 (3)

- Simultanée pour les deux premières composantes de chaque bloc :



Remerciements

Christelle REYNES : AG et Splines

Emeline SICARD : PLS local

Myrtille VIVIEN : PLS multitableaux

Conclusions

- Résout « simplement », dans ses premières formulations, un problème important de prédiction.

- Grande diffusion de PLS, dans beaucoup de domaines.

- Souplesse et adaptation des algorithmes de PLS, ce qui a généré beaucoup d'approximations dans des « nouveaux développements » : pas de critère explicite, donc pas de CV théorique etc...

- Possibilité d'optimiser de nouveaux critères, plus ou moins simples.

- Algorithmes pas toujours rapides (pas un gros problème).

- Procédures de validations toujours présentes.

Bibliographie succincte

Sur PLS « en général »

M. Tenenhaus (1998) *La régression PLS : théorie et pratique*, Editions Technip, 252p

S. Wold, M. Sjöström,& L. Eriksson (2001) PLS-regression: a basic tool of chemometrics, *Chem. Intel. Lab. Syst.*, **58**, 109-130

M. Andersson (2009) A comparison of nine PLS1 algorithms, *J. of Chem*, **23**, 518–529

Sur PLS spline

J.F. Durand & R. Sabatier (1997) Additive splines for partial least squares regression, *JASA*, **92**, 440, 1546-1554.

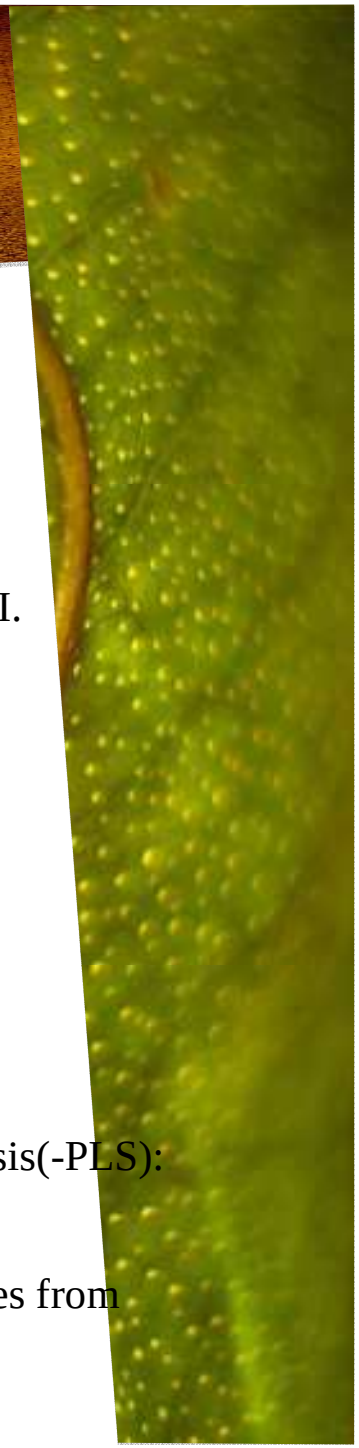
J.F. Durand (2001) Local polynomial additive regression through PLS and splines: PLSS, *Chem. Intel. Lab. Syst.*, **58**, 235-246

Sur les AG en Analyse des Données

C. Reynès (2007) *Etude des Algorithmes Génétiques et application aux données de protéomique*. Thèse, Université Montpellier I.

R. Sabatier & C. Reynès (2008) Extensions of simple component analysis and simple linear discriminant analysis using genetic algorithms, *CSDA*, **52**, 4779–4789

Bibliographie (suite)



Sur LPLS

E. Sicard (2004) *Choix de composantes pour l'analyse spatiale et la modélisation : application aux pluies du Nordeste brésilien*. Thèse, Université Montpellier II.

E. Sicard & R. Sabatier (2006) Theoretical framework for local PLS1 regression and application to a rainfall data set, *CSDA*, **51**, 1393 – 1410

Sur PLS multitableau (avec même du non-linéaire !)

M. Vivien (2002) *Approches PLS linéaires et non-linéaires pour la modélisation de multi-tableaux. Théorie et Applications*. Thèse, Université Montpellier I.

M. Vivien & R. Sabatier (2003) Generalized Orthogonal Multiple Co-Inertia Analysis(-PLS): new multiblock component and regression methods. *J. of Chem*, **17**, 287-301

M. Vivien, T. Verron &R. Sabatier (2005) Comparing and predicting sensory profiles from NIRS data: use of the GOMCIA and GOMCIA-PLS multiblock methods. *J. of Chem*, **19**, 162-170