

# Identification et Quantification de métabolites dans un spectre RMN

---

R. Servien, P. Tardivel, C. Canlet, L. Debrauwer, M. Tremblay-Franco, D. Concordet

UMR 1331 Toxalim,  
INRA - ENVT,  
Toulouse



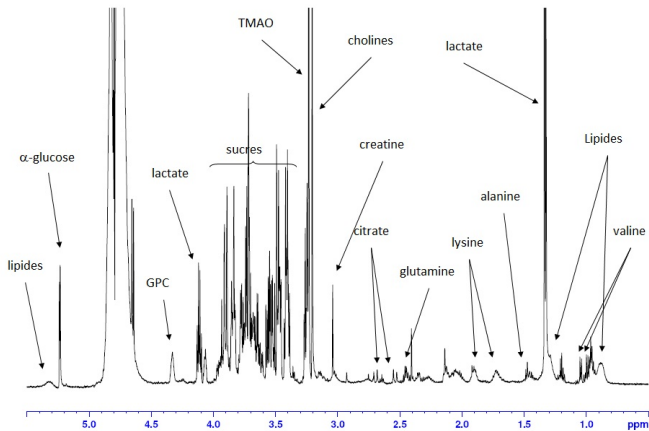
---

Séminaire de l'unité MIAT  
13 Mars 2015

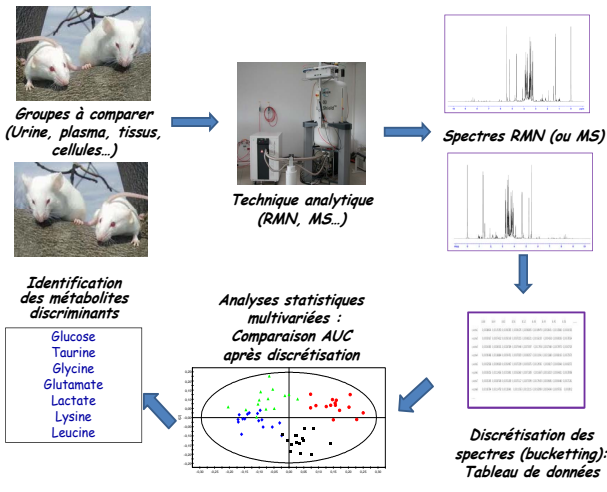
# Contexte

- Métabolomique : Domaine en plein essor.
- Applications en biologie, en médecine ...
- Deux groupes d'étude (traité/non traité) et on procède à l'étude des métabolites participant aux réactions métaboliques d'un organisme.

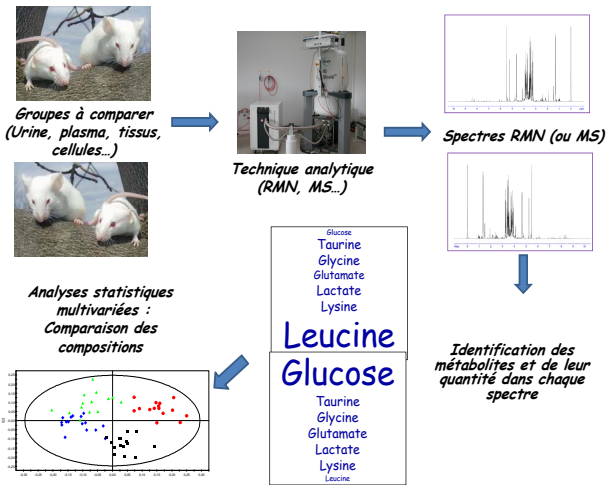
# Spectre RMN



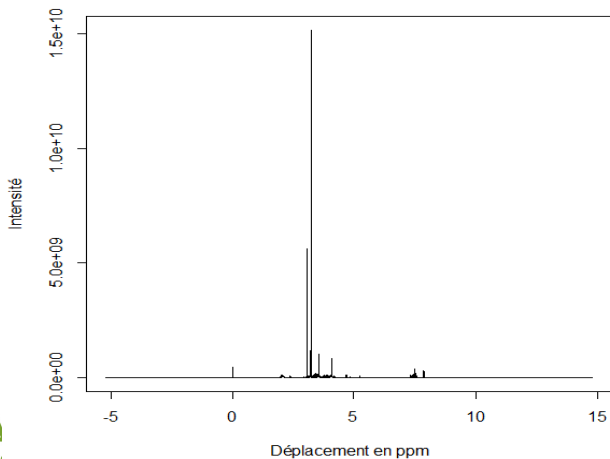
# Empreintes Métaboliques : Stratégie actuelle



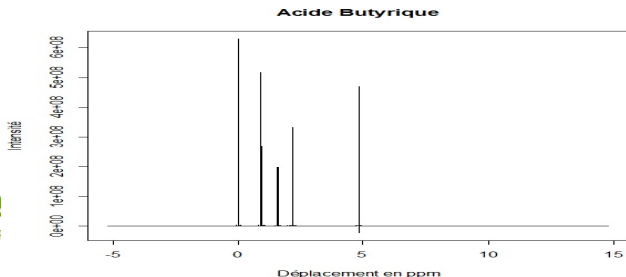
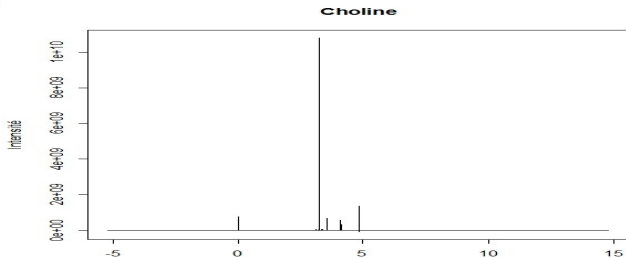
# Stratégie proposée



# Exemple de mélange à analyser



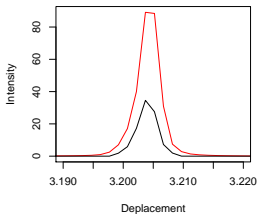
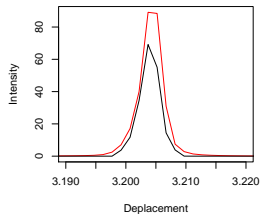
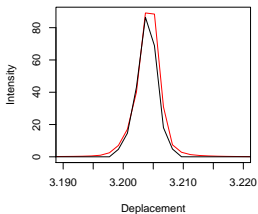
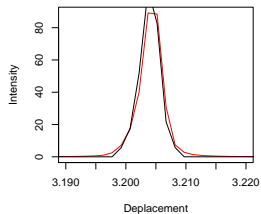
# Exemples de métabolites



Identification et Quantification de métabolites dans un spectre RMN - R. Servien (INRA Toulouse)

*Xalim*

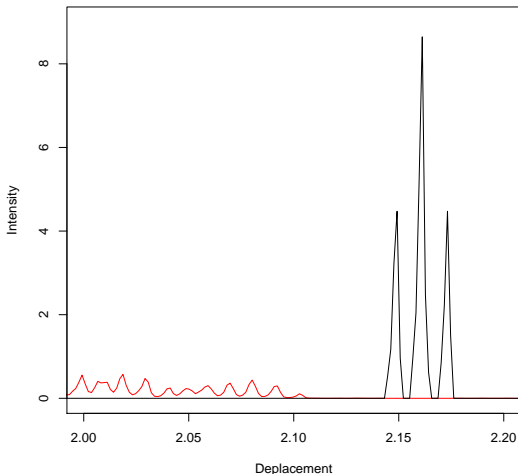
# Choline

**Proportion = 0.2****Proportion = 0.4****Proportion = 0.5****Proportion = 0.6**

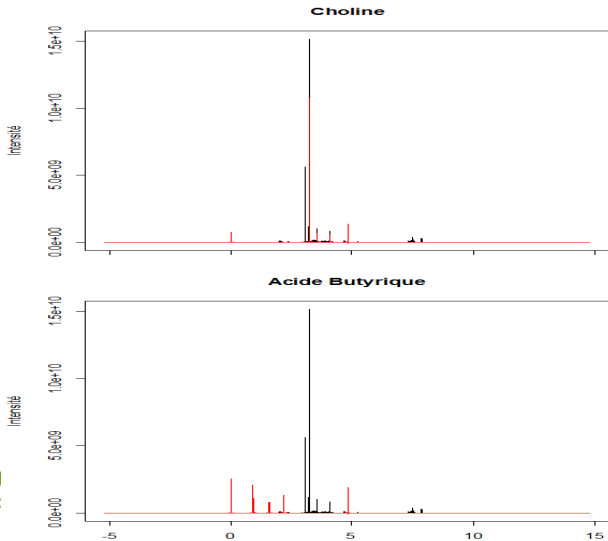


# Acide Butyrique

Proportion = 0.2



# Tests préliminaires



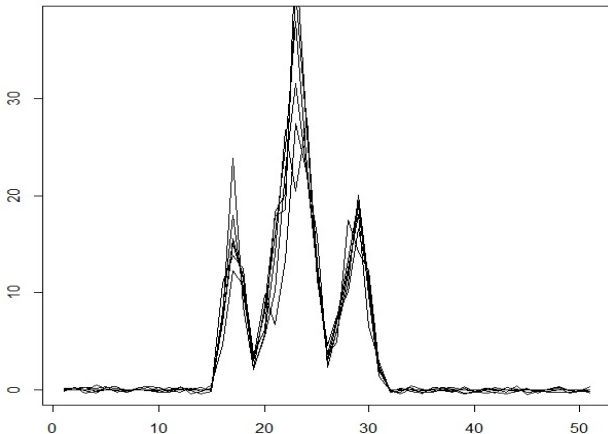
## Modélisation naïve

- Un mélange est modélisé comme une fonction  $f(x) \geq 0$ .
- On dispose d'une bibliothèque de spectres de  $K$  métabolites purs représentables par des fonctions  $(U_i(x))_{i=1,\dots,K}$  positives.
- Nous allons décomposer la fonction  $f$  de la façon suivante :

$$f(x) = \sum_{i=1}^K \alpha_i U_i(x) + R(x)$$

- $f$  et les  $(U_i(x))_{i=1,\dots,K}$  sont normalisés i.e. pour tout  $i$ ,  $\int U_i(x) dx = 1$
- $\alpha_i \in [0, 1]$  représente l'abondance du  $i^{\text{ème}}$  métabolite dans le mélange et on a  $\sum_{i=1}^K \alpha_i \leq 1$ ,
- $R(x) \geq 0$  est la part du spectre non identifiable à l'aide des spectres des métabolites de la bibliothèque.

# Les signaux sont bruités



# Modélisation statistique

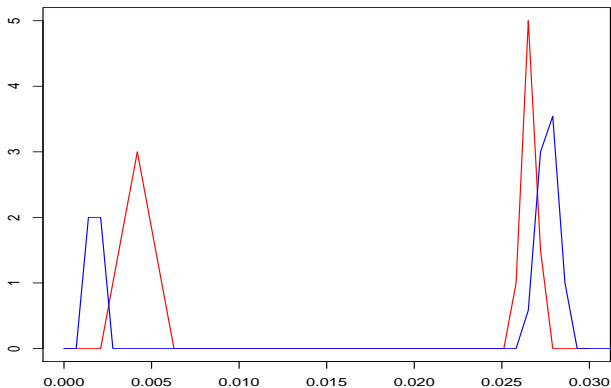
- Un mélange est modélisé comme une fonction **aléatoire**  
 $Y(x) \geq 0$ .
- On dispose d'une bibliothèque de spectres de  $K$  métabolites purs représentables par des fonctions **aléatoires**  $(U_i(x))_{i=1,\dots,K}$  positives.
- Les bruits blancs  $\varepsilon_x$  et  $(\varepsilon_x^i)_{i=1,\dots,K}$  ont un écart-type constant et sont indépendants.
- Nous allons décomposer la fonction  $Y$  de la façon suivante :

$$Y(x) = \left( \sum_{i=1}^K \alpha_i U_i(x) + R(x) \right) (1 + \varepsilon_x)$$
$$\begin{cases} V_1(x) = U_1(x)(1 + \varepsilon_x^1) \\ \vdots \\ V_K(x) = U_K(x)(1 + \varepsilon_x^K) \end{cases}$$

# Problèmes de déformation

- Il n'y a pas la même concentration du métabolite entre le métabolite pur et le mélange : les pics n'ont pas forcément la même forme.
- Il n'y a pas les mêmes conditions expérimentales (pH, produits, expérimentateur ...) : les pics peuvent être décalés.

# Exemple



# Déformation

- Ce n'est pas simplement une translation uniforme sur le spectre.
- Les déformations sont locales.
- Le spectre peut être translaté mais également dilaté ou contracté.
- Déformation maximale fixée par l'expertise (0.01 ppm).



# Modélisation des déformations

- Un mélange est modélisé comme une fonction aléatoire  $Y(x) \geq 0$ .
- On dispose d'une bibliothèque de spectres de  $K$  métabolites purs représentables par des fonctions aléatoire  $(U_i(x))_{i=1,\dots,K}$  positives.
- Nous allons décomposer la fonction  $Y$  de la façon suivante :

$$Y(x) = \left( \sum_{i=1}^K \alpha_i U_i(\varphi_i^Y(x)) + R(x) \right) (1 + \varepsilon_x)$$
$$\begin{cases} V_1(x) = U_1(\varphi^1(x))(1 + \varepsilon_x^1) \\ \vdots \\ V_K(x) = U_K(\varphi^K(x))(1 + \varepsilon_x^K). \end{cases}$$

## Modèle final

$$Y(x) = \left( \sum_{i=1}^K \alpha_i U_i(\varphi_i^y(x)) + R(x) \right) (1 + \varepsilon_x)$$

$$\begin{cases} V_1(x) = U_1(\varphi^1(x))(1 + \varepsilon_x^1) \\ \vdots \\ V_K(x) = U_K(\varphi^K(x))(1 + \varepsilon_x^K) \end{cases}$$

Hypothèse implicite **d'identifiabilité** : il n'existe pas de métabolite dont le signal peut s'exprimer comme le mélange de signaux d'autres métabolites à une déformation près.

# Estimation des proportions $\alpha_j$

Contraintes :

- Il faut que tous les métabolites soient traités simultanément,
- Il faut une méthode parcimonieuse,
- Il faut que le calcul soit très rapide.

$$\sum_{i=1}^K \alpha_j \text{ est maximum sous } \begin{cases} \sum_{i=1}^K \alpha_j & \leq 1 \\ Y(t) & \geq \sum_{i=1}^K \alpha_j Z_i \circ \varphi_i(t). \end{cases} \quad (1)$$

# Problème

Pour identifier les métabolites présents dans le mélange, il faut donc estimer

- la variance des  $\varepsilon_X^K$ ,
- les déformations  $\varphi_i^Y$  et  $\varphi^i$  ou une composition des deux,
- les  $\alpha_i$  : si  $\alpha_i > 0$ , le  $i^{\text{e}}$  métabolite est présent dans le mélange.

Contraintes :

- Temps de calcul le plus faible possible,
- Doit permettre de gérer la "compétition" entre les métabolites : tout doit être estimé simultanément.

# Stratégie d'estimation

- Soit le couple  $(\hat{\varphi}_i, \hat{\alpha}_i)_{i=1, \dots, K}$  solution de (1).
- Soit le couple  $(\varphi_i^*, \alpha_i^*)_{i=1, \dots, K}$  obtenu en testant les métabolites un par un.
- Soit le couple  $(\varphi_i^*, \hat{\alpha}_i)_{i=1, \dots, K}$  obtenu en optimisant les déformations une par une, puis en optimisant avec compétition les  $\alpha_j$ .

## Proposition

Pour chaque  $i$ , on a

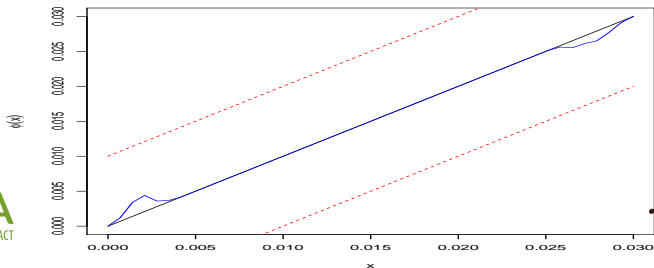
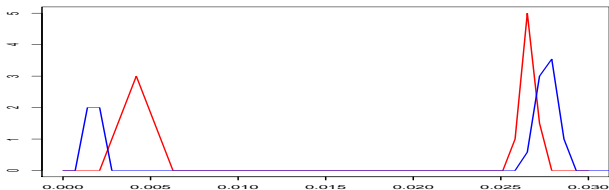
$$\alpha_j^* \geq \hat{\alpha}_i \geq \hat{\alpha}_j.$$

# Modélisation des déformations

Hypothèses minimales sur les fonctions de déformation  
 $(\varphi^i)_{i=1,\dots,K}$  et  $(\varphi_j^y)_{i=1,\dots,K}$  :

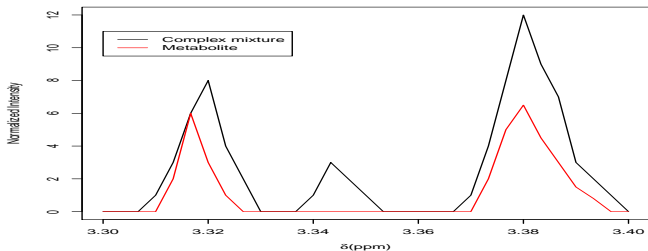
- les fonctions sont continues (les déformations ne sont pas trop violentes),
- les fonctions sont strictement monotones (elles ne peuvent pas intervertir l'ordre des pics),
- les déformations ont une amplitude maximale fixée par l'expertise.

# Déformations $\varphi$



# Algorithme itératif

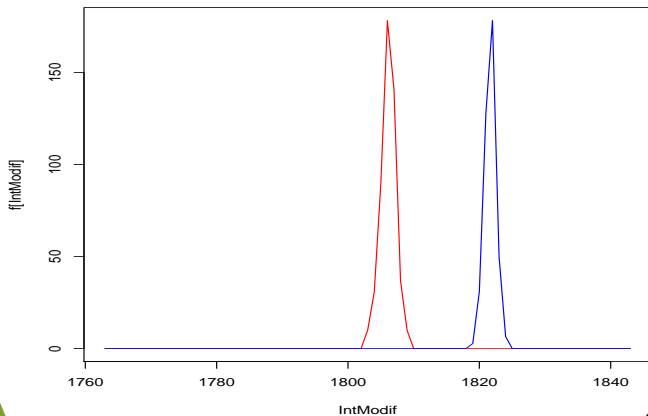
- On prend un métabolite.
- On estime  $\alpha_j$  en utilisant uniquement ce métabolite.
- On cherche le point de "contact".



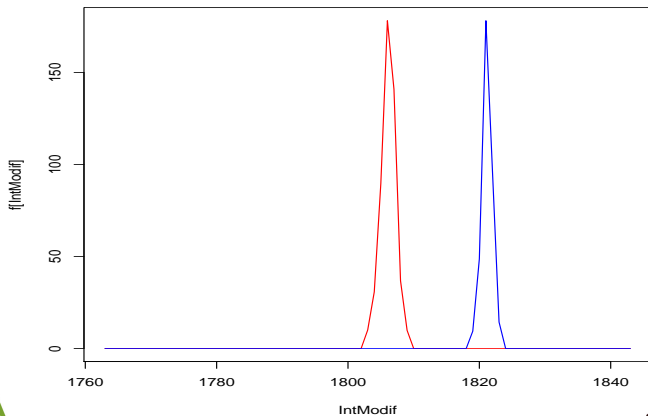
- On sélectionne sa composante connexe et on essaie de la recaler en utilisant la méthode de la page précédente.



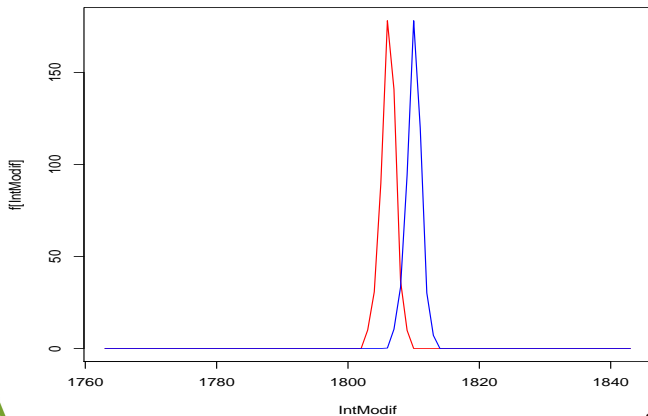
# Exemple



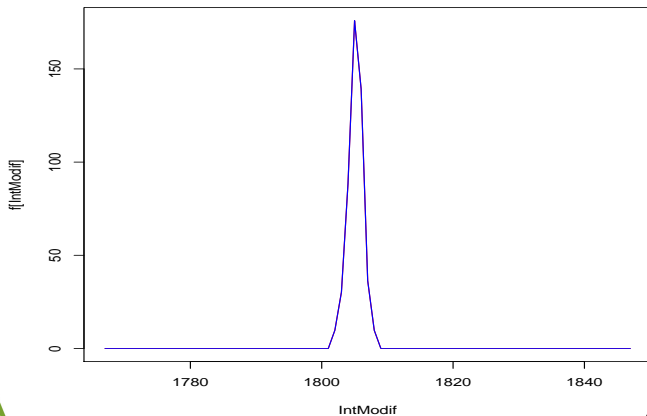
# Exemple



# Exemple



# Exemple



# Algorithme itératif

- A l'aide de l'étape précédente, on a fait augmenter le  $\alpha_j$  de ce métabolite.
- On réitère les étapes précédentes jusqu'à ce qu'aucune nouvelle déformation ne permette de faire augmenter le  $\alpha_j$ .
- On fait ça pour tous les métabolites.
- On obtient  $(\varphi_j^*, \alpha_j^*)_{j=1, \dots, K}$ .

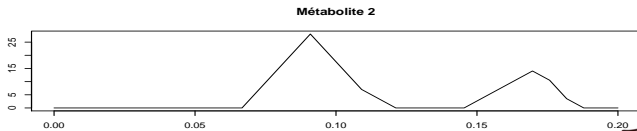
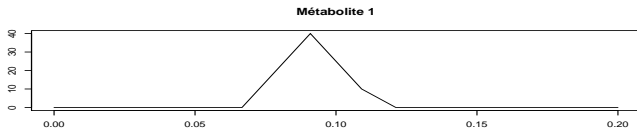
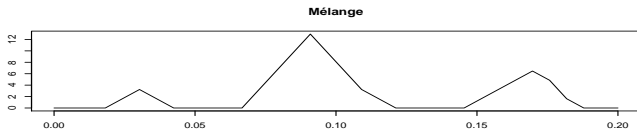
## Cadre formel

- $\varphi \in \mathcal{F}$  ensemble des fonctions continues strictement croissantes sur  $[a, b]$  qui valent  $a$  en  $a$  et  $b$  en  $b$ .
- On associe la fonction  $g_\zeta(x) = x + \zeta x(1 - x)$  au nombre  $\zeta \in [-1, 1]$ . On peut voir que  $g_\zeta \in \mathcal{F}$ .
- Considérons  
 $\mathcal{A}_n = \{\varphi_{\zeta^n} = g_{\zeta_n} \circ \dots \circ g_{\zeta_1}, \text{ avec } \zeta^n = (\zeta_1, \dots, \zeta_n) \in [-1, 1]^n\}$   
l'ensemble des polynômes obtenus en composant  $n$  fonctions  $g_{\zeta_1}, \dots, g_{\zeta_n}$ . On a :  $\mathcal{A}_1 \subset \dots \subset \mathcal{A}_n \subset \mathcal{F}$ .
- On estimera une fonction  $\varphi \in \mathcal{F}$  par une fonction  $\varphi_{\zeta^n} \in \mathcal{A}_n$ .

### Proposition

*L'ensemble des fonctions  $\mathcal{A}_n$  est dense pour  $(F, \|\cdot\|_\infty)$ .*

# Compétition



# Calcul des $\hat{\alpha}_j$

- On a obtenu  $(\varphi_i^*)_{i=1,\dots,K}$  à l'étape précédente.



$$\sum_{i=1}^K \alpha_i \text{ est maximum sous } \begin{cases} \sum_{i=1}^K \alpha_i & \leq 1 \\ Y(t) & \geq \sum_{i=1}^K \alpha_i Z_i \circ \varphi_i^*(t). \end{cases}$$

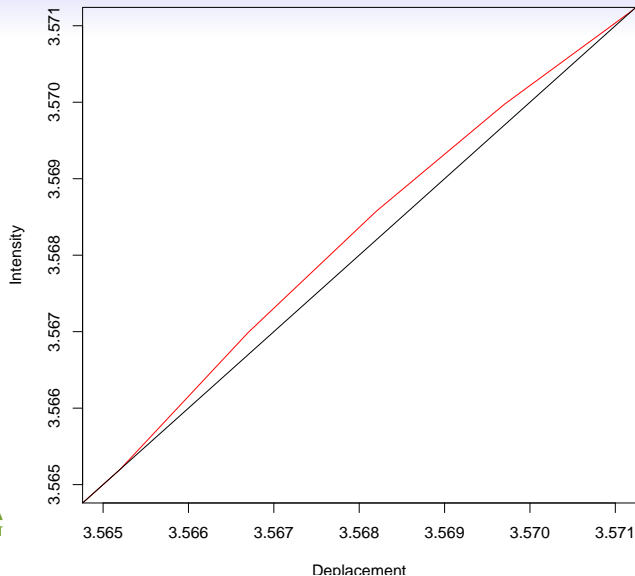
- Programmation linéaire.
- On obtient  $(\hat{\alpha}_i)_{i=1,\dots,K}$ .



# Programme

- Logiciel gratuit *R*.
- `source("ProgTotal.r")`
- `a < -Total("STD2Melange - 1.002.txt")`
- Bibliothèque de 36 métabolites à tester  $\approx$  15 secondes.

## Exemple de Déformation

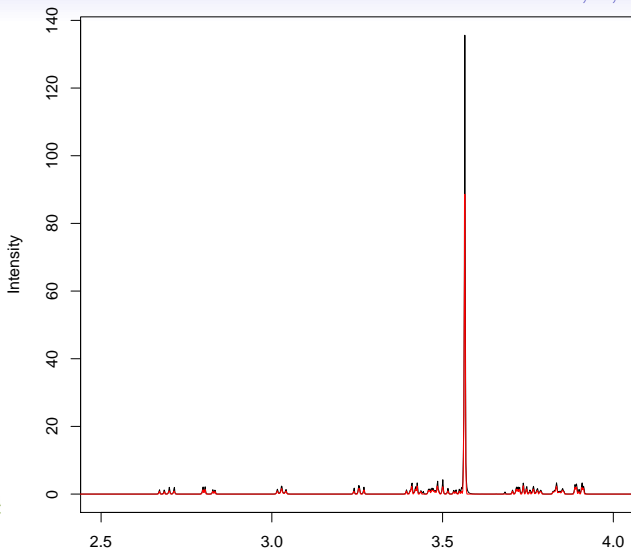


# Résultats

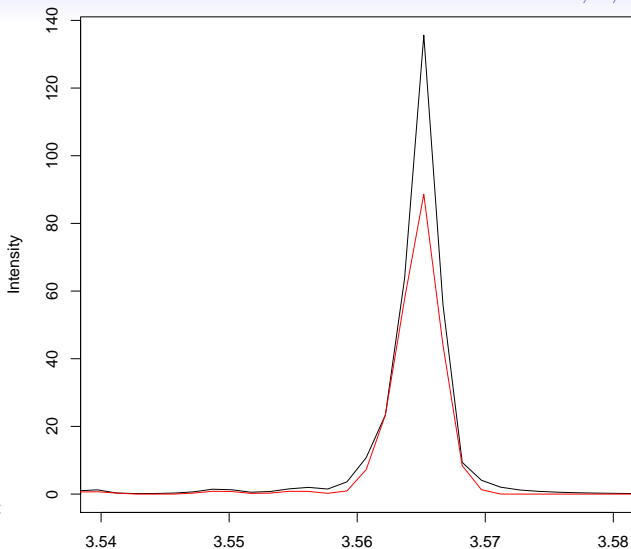
- 32 absents du mélange, 4 présents.

Nom du Métabolite	Proportion Minimum	Proportion Maximum	Vraie Proportion
Glycine	0.694	0.694	0.76
Glucose	0.103	0.103	0.15
Lysine	0.024	0.026	0.02
Acide Aspartic	0.039	0.040	0.08

# Reconstruction avec les $\left(\hat{\alpha}_i\right)_{i=1,\dots,K}$ .



# Reconstruction avec les $(\hat{\alpha}_i)_{i=1,\dots,K}$ .



# Travail actuel : Débiaisage avec bootstrap

- Approche bootstrap.
- On génère  $Z_i^B = Z_i (1 + \varepsilon_i)$ .
- et  $Y^B = \left( \sum_{i=1}^K \hat{\alpha}_i Z_i \right) (1 + \varepsilon)$ .
- Puis on calcule les  $\left( \hat{\alpha}_i^B \right)_{1 \leq i \leq K}$ .
- On itère 1000 fois  $\Rightarrow \forall i, \mathbb{E} \left( \hat{\alpha}_i^B \right)$ .
- Estimation du biais :  $\alpha_i - \hat{\alpha}_i \approx \hat{\alpha}_i - \mathbb{E} \left( \hat{\alpha}_i^B \right)$ .

# Perspectives

- Théorie :
  - Débiaiser les estimations (bootstrap ...)
  - Définir des intervalles de confiance (loi exacte des proportions ou simulée par bootstrap)
  - Introduction d'apprentissage.
- Application :
  - Package R et implémentation sous Galaxy.
  - Validation de la méthode (plus de mélanges, matrices différentes, données extérieures ...)