Rapport de stage

Marion DEFFARGES

Université de Poitiers

Tuteur de stage : Romain Yvinec
Enseignant référent : Hermine Biermé

Remerciments

Je voudrais remercié mes deux encadrants de stage Romain Yvinec et Hermine Biermé qui ont sus ce montrer attentifs et disponibles.

Je tiens à remercier également toutes l'équipe Bios qui m'a accueilli au sein de son équipe : Anne Poupon (DR CNRS, co-responsable), Eric Reiter (CR INRA), Pascale Crépieux (DR CNRS, co-responsable), Gilles Bruneau (CR INRA), Yves Combarnous (DR Em CNRS), Danièle Klett (IR CNRS), Véronique Bozon (MCU), Yannick Corde (IE), Olivier Galland (Assistant Ingénieur), Manon Chevalier (Assistant Ingénieur), Pauline Raynaud (Assistant Ingénieur), Ophélie Travert (Assistant Ingénieur), Thomas Boulo (Technicien), Joël Fontaine (Technicien), Christophe Gauthier (Technicien), Océane Vaugrente (Techinicienne), Shifa Tahir (Doctorante), Aurélie Tréfier (Doctorante), Francesco de Pascali (Doctorant), Linh Tran (Post-doc), Flavie Landomiel (Post-doc), Aude Lemettre (stagiaire M2), Laetitia Mathias (stagiaire M2) et Khalid Aouiss (stagiaire M2).

Mais aussi à Lucie Pellissier et Jorge Gandia-Sanchez qui ont mené les expériences sur lesquelles j'ai étudié les données.

Table des matières

Introduction **9**

1 Structure **11**

2 La démarche **13**

3 Données à traiter **15**

4 Statistiques descriptives **19**

5 Traitement **23**
5.1 Création d'un modèle de réseaux de réaction biochimique 23
5.2 Construction du modèle statistique 28
5.2.1 Beta-arrestine 28
5.2.2 Calcium 29
5.3 Optimisation 31

A Complément **33**
A.1 Analyse descriptive des données brutes 33
A.1.1 beta-arrestine 33
A.1.2 Calcium 36
A.2 Test normalité des données brutes 36
A.3 Expériences 36
A.3.1 Transfection 37
A.3.2 pharmacologie 38
A.4 logiciel utilisé 38

A.4.1 D2D 39
A.4.2 HYPE 44

B Théorie **47**
B.1 Modèles de réseaux de réactions biochimiques 47
B.1.1 Utilité des modèles mathématiques en biologie 47
B.1.2 Construction d'un modèle mathématique 49
B.1.3 Équations différentielles ordinaires 51
B.1.4 Exemples 52
B.2 Inférence paramétrique- modèle statistique 55
B.2.1 Construction du système d'EDO 56
B.2.2 Construction du modèle statistique 61
B.2.3 Les estimateurs 61
B.3 Algorithme d'optimisation 66
B.3.1 Gradient 72
B.3.2 Hessienne 76
B.4 Qualité du modèle 80
B.4.1 Identifiabilité 80
B.4.2 intervalle de confiance 80
B.4.3 valeurs aberrante 81
B.5 Selection du meilleur modèle 82
B.6 Biologie 84
B.7 Test de normalité 86
B.8 Résolution équation différentielle ordinaire 86
B.8.1 équation différentielle linéaire 87
B.8.2 équation différentielle non linéaire : méthode numérique 87

C Programmation **93**
C.1 Importation des données 93
C.1.1 données normalisés 93
C.1.2 données brutes calcium 96
C.2 graphique 100
C.2.1 dose réponse 100
C.2.2 courbe cinétique 103
C.2.3 Graphique données brutes 107
C.3 Noramlité 114

C.3.1 code général 114
C.3.2 beta-arrestine 117
C.3.3 Calcium 118
C.4 D2D 125
C.4.1 Création tableau de donnée 125
C.4.2 Models 145
C.4.3 data 148
C.4.4 programmation 150

D Figure **155**
D.1 Dose réponse 155
D.2 Courbe cinétique 158
D.3 Donées brutes 165
D.4 Test de normalité 176
D.4.1 beta-arrestine 176
D.4.2 Calcium 176
D.4.3 beta-arrestine brute 176
D.4.4 Calcium brut 183

Bibliographie **185**

Introduction

Dans ce stage nous nous intéressons à l'estimation paramétrique de modèles de signalisation cellulaire appliqué au calcul de biais des RCPG portant ainsi sur la modélisation dynamique de réseaux de réactions moléculaires et l'inférence statistique des paramètres de ces modèles. Il s'est déroulé au sein de l'équipe BIOS de l'INRA qui est spécialisée dans les RCPG.

Dans un premier temps, mon stage s'est porté sur les démarches théoriques pour nous concentrer ensuite sur le traitements de données expérimentales. L'objectif de ce mémoire étant de réaliser une étude statistique sur celle-ci.

Pour ce faire, et après une courte présentation de la structure d'accueil, les démarches derrières le traitement de donné sont présentées brièvement. Par la suite, quelques notions de biologie moléculaire sont décrites afin d'introduire les données. Dans un second temps, une analyse descriptive des données est effectuée avant d'appliquer les démarches spécifiques à notre étude. Dans la dernière partie, les résultats statistiques sont analysés. La théorie derrière les démarches, le déroulement des expérience et les connaissances biologiques nécessaire à la compréhension des données sont expliqués plus en détail en annexe.

Pendant ce stage j'ai pu assister à des conférences sur les mathématiques appliqués à la biologie ce qui a été fort instructifs.

Chapitre 1

Structure

L'Institut national de la recherche agronomique (INRA) est un organisme français de recherche en agronomie fondé en 1946. Premier institut de recherche agronomique en Europe et deuxième dans le monde, l'INRA mène des recherches dans le but d'obtenir une alimentation saine et de qualité, une agriculture durable, et pour un environnement préservé et valorisé.

Il existe dix-sept centres de recherche régionaux et un centre-siège à Paris. Le stage s'est déroulé au sein du centre INRA Val de Loire, basée à Nouzilly (20 km de Tours), qui comporte plusieurs unités de recherche notamment l'unité Physiologie de la Reproduction et des Comportements (PRC). L'unité PRC est constitué de 11 équipes de recherche dont les thématiques sont organisées autour de l'axe de contrôle de la reproduction et des comportements, allant de l'analyse comportementale et neurobiologique à l'analyse des mécanismes moléculaires et génétiques liés à la fonction de reproduction et sa régulation.

Au sein de l'unité, l'équipe Biologie et Bioinformatique des Systèmes de Signalisation (BIOS), au sein de laquelle j'ai été admise, développe une stratégie pluridisciplinaire de biologie systémique combinant des approches expérimentales (aux échelles moléculaire, cellulaire, physiologique), de modélisation mathématique et informatique, afin d'acquérir un savoir exhaustif et quantitatif des phénomènes biologiques médiés par les récepteurs couplés aux protéines G (RCPG), en particulier ceux liés à la reproduction (récepteurs des hormones gonadotropes et récepteurs des adipocytokines). Les objectifs de l'équipe BIOS sont de comprendre la dynamique des réseaux de signalisation cellulaire induits par la stimulation de ces récepteurs par un ligand donné, en définissant précisément les effets intracellulaires induits et leurs conséquences biologiques. Cette

compréhension permet à l'équipe BIOS, en parallèle, de développer de nouvelles approches pharmacologiques afin de contrôler finement l'activité de récepteurs lié à la reproduction et pour réduire les effets néfastes associés à l'utilisation de leur ligand (naturel ou de synthèse) chez les animaux d'élevages et chez l'humain.

Chapitre 2

La démarche

Notre étude se porte sur l'évolution de la concentration de certaines molécules sous l'action de ligand [1] (cf.Annexe B.6 page 84). Notre objectif est de parvenir à une compréhension des systèmes et de prédire /modifier leur comportement.
A partir d'une liste de réseaux de réactions biochimiques [2] de ligands, on transforme la modélisation de la réaction en systèmes d'équations diérentielles ordinaires (EDO) afin d'étudier cette réaction (cf.Annexe B.1 page 47). Cependant, ces modèles contiennent de nombreux paramètres dont les valeurs sont en général inconnus : concentrations initiales des molécules, constantes cinétiques de réaction, paramètres d'échelles reliant les concentrations aux mesures effectuées, etc...Ils doivent alors être estimés à partir de données expérimentales.

Ces données expérimentales, correspondant aux concentration des molécules souhaités, sont obtenues via des expériences in vitro sur des lignés cellulaires généralement cancéreuses réalisés au sein des laboratoires de l'INRA (cf.Annexe B.1 page 47). Elles sont effectuées pour la reproduction animal, mais peuvent-être transposable pour l'homme.

Un modèle statistique représentant les erreurs de mesures est ensuite créé en fonction des résultats (cf.Annexe B.2 page 55). Deux modèles gaussiens sont possibles :

- modèle additif gaussien (cas général) :

$$Y_{obs} = Y_{obs} + \beta \text{ où } \beta \text{ : bruit } (\sim \mathcal{N}(\mu, \sigma^2))$$

[1] molécule qui se lie de manière réversible à une macromolécule ciblée, dans notre étude c'est une molécule utile dans le processus de reproduction comme des hormones

[2] ensemble de relations qui décrivent les interactions entre certaines molécules et donnent lieu à des transformations (dégradation, production, formation de complexe etc...) de certaines molécules en de nouvelles molécules.

- modèle multiplicatif gaussien :

$$Y_{obs} = Y_{obs} \times \beta \text{ où } \beta \text{ : bruit } (\sim \mathcal{N}(\mu, \sigma^2))$$

A l'aide d'un estimateur une fonction objective est créée. L'optimisation de cette fonction (cf.Annexe B.3 page 66) permet ensuite d'estimer les paramètres, à l'aide de l'un d'un modèle statistiques. Ce problème d'optimisation peut être résolu en utilisant des méthodes d'optimisation hybrifes mélangeant des mathodes locales, globales, basé sur le gradient ou des méthodes sans dérivation. Les méthodes les plus connus et utilisés dans ces algorithmes hybrides sont répertoriées dans le tableau suivant : Le tableau des principales méthodes utilisés est construit :

	méthode sans dérivation	méthode basée sur le gradient
local	CMA-ES	Newton line search trust-region
global	Algorithme de génétique particle Swarm	scatter search clustering search

De plus, pour les algorithmes basé sur le gradient, le gradient et le hessien (ou des approximations) de la fonction objective doivent être calculés. Pour les évaluer, des analyses de sensibilité adjointe ont été développées puisque les approches standards (différences finies ou analyse de sensibilité directe) sont exigeantes en termes de calcul pour les modèles d'ODE de grande dimension.

Une fois les paramètres optimaux estimés, les intervalles de confiance de chaque paramètres sont calculés en utilisant des outils d'identifiabilité comme le profil de vraisemblance (cf.Annexe B.4 page 80).
Des problèmes de non identifiabilité peuvent se produire dans ce cas deux solutions sont alors possibles :
-redéfinition du modèle
-reparamètrisation (αA_0 non identifiabilité : $\alpha A_0 = \beta$)

Chapitre 3

Données à traiter

BIOS travaille sur les récepteurs couplés aux protéines G [1] (RCPG ou GPCRs) qui sont des protéines membranaires.

L'étude actuel se porte sur l'ocytocine et les AVP peptides qui sont 2 ligands endogènes[2] ce qui permet de mesurer leurs actions à travers la dose de calcium et la proximité des beta-arrestine avec le récepteur. Le but est de connaître l'action de ses ligands sur les récepteurs d'ocytocine (OXTR) et les récepteurs d'AVP (AV1A AV1B et AV2). Pour cela, 7 concentrations (10^{-4} à 10^{-10} pour le Calcium, 10^{-5} à 10^{-11} pour la Beta-arrestine) et 1 tampon sont testés pour chaque couple ligand-récepteur et pour chaque type de mesure (calcium ou beta-arrestine). De plus, pour chaque mesure entre 4 réplicats sont réalisé. On a alors 520 jeux de données.

Le recrutement de beta-arrestine et la concentration de calcium sont issu d'expériences in-vitro sur des cellules cancéreuses humaines (cf.Annexe A.3 page 36) réalisé au sein des laboratoires de l'INRA. Il est à noté qu'un problème d'échelle de temps existe puisque les expériences sur la beta-arrestine durent environ 30 minutes tandis que celle sur le Calcium durent seulement 15 secondes.

Beta-arrestine

Le recrutement de beta-arrestine n'est pas une mesure direct puisqu'elle est le résultat du ratio de l'indice de fluorescence de deux autres molécules : Rluc et Venus. En effet, lorsqu'on injecte un ligand après le passage du récepteur cela augmente l'excitation du Rlucs (Renilla Luciferase) (qui émet dans les 480nm) et excitant alors les molécule Venus (émettant dans les 530nm).

[1] se caractérisent par une structure commune de sept hélices hydrophobes transmembranaires, connectées par trois boucles extracellulaires et trois boucles intracellulaires

[2] qui prend naissance à l'intérieur d'un corps, d'un organisme, d'une société, qui est dû à une cause interne

Le bret (un logiciel de la machine nommée MITRAS) mesure alors l'indice de fluorescence de ces 2 molécules :

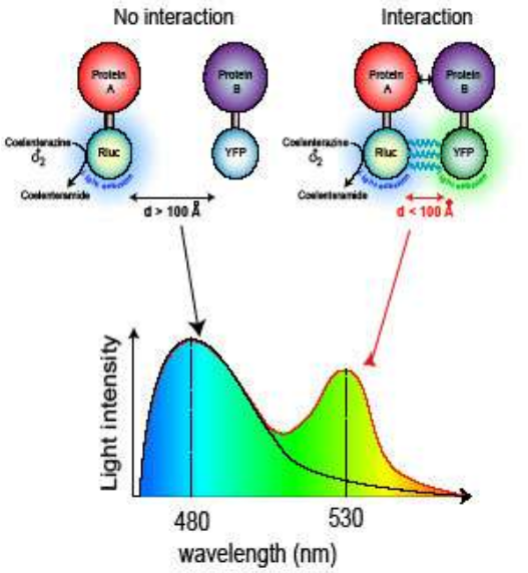

FIGURE 3.1 : Interaction entre Venus et Rluc

Pour avoir le taux de recrutement de beta-arrestine normalisé la formule suivante est utilisé :

$$\frac{\text{Sign Venus}}{\text{Sign Rluc}} - \text{basal}$$

Ainsi pour étudier seulement l'effet des ligands, la valeur moyenne du basal est retranchés aux ratio des deux indices de fluorescence.
Cependant, Il faut faire attention aux données obtenues puisque le Rlucs émet de la fluorescence sans présence de ligands (d'où l'importance de retrancher les données basales) et il émet aussi une petite partie en 530 nm par conséquent le Bret mesure une présence de Venus même sans ligand.

Calcium

Le calcuim est quand à lui une mesure direct de son indice de fluorescence à l'aide du MITRAS. Lorsqu'on injecte un ligand, du calcium est libéré dans la cellule via le cycle des protéines G et le

reticulum endoplasimque (cf.5.1), on mesure alors la concentration du calcium dans les cellules par luminescence. Cependant le reticulum endoplasmique (RE) contient encore du calcium après le passage du ligand, afin d'éviter un biais on mesure cette quantité de calcium présent dans le RE. Pour cela, un inomycin est injecté après l'action du ligand (environ 20 secondes après) qui vas détruire le reticulum endoplasmique et ainsi libérer le calcium contenu dans ce-dernier.
Ainsi on obtiens 2 types données : la concentration de calcium issu du cycle G et la concentration de calcium issu de la destruction du reticulum endoplasmique.

Pour avoir la concentration de calcium normalisé la formule suivante est utilisé :

$$\frac{x_{i,j}}{AUC[\text{Ligand}]_j + AUC[\text{Ionomycin}]_j}$$

où $x_{i,j}$: Chaque point de donnée

$AUC[\text{Ligand}]$: Aire sous la courbe des données ligands

$AUC[\text{Ionomycin}]$: Aire sous la courbe des données Ionomycins

Il s'agit d'une normalisation standard d'autres normalisations sont peut-être possibles.

Dans la suite de ce rapport seules les données normalisées seront étudiées (certaines études sur les données brutes sont effectuées dans l'Annexe Complément).

Chapitre 4

Statistiques descriptives

Une étude préliminaires des données normalisé est réalisé afin de voir le comportement général des données au cours du temps (courbes cinétiques) et en fonction des doses (courbes doses réponses).
Pour étudier les courbes doses réponses on se place à $t = 100$. Dans un soucie de place seule le récepteur OXTR est présenté (programme : (cf.Annexe C.2.1 page 100), autre graphique : (cf.Annexe D.1 page 155)).

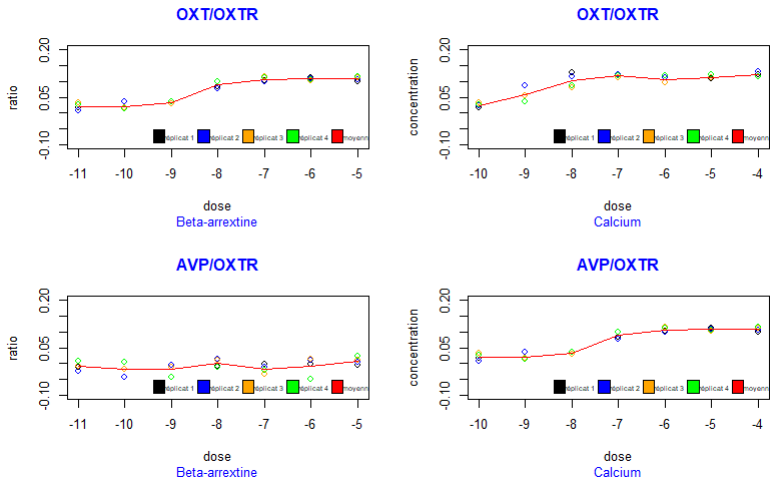

FIGURE 4.1 : Courbe dose réponse du récepteur OXTR

D'après le graphique, le recrutement de beta-arrestine du couple AVP/OXTR reste nulle pour n'importe quels dose. Le recrutement de beta-arestine de l'autre couple et la concentrations de calcium pour les deux couples, semblent avoir une réponse croissante avec la concentration du

ligand injecté.

Pour étudier les courbes cinétiques on se place aux différentes doses. Dans un soucie de place seule le récepteur OXTR est présenté (programme : (cf.Annexe C.2.2 page 103), autre graphique : (cf.Annexe D.2 page 158)).

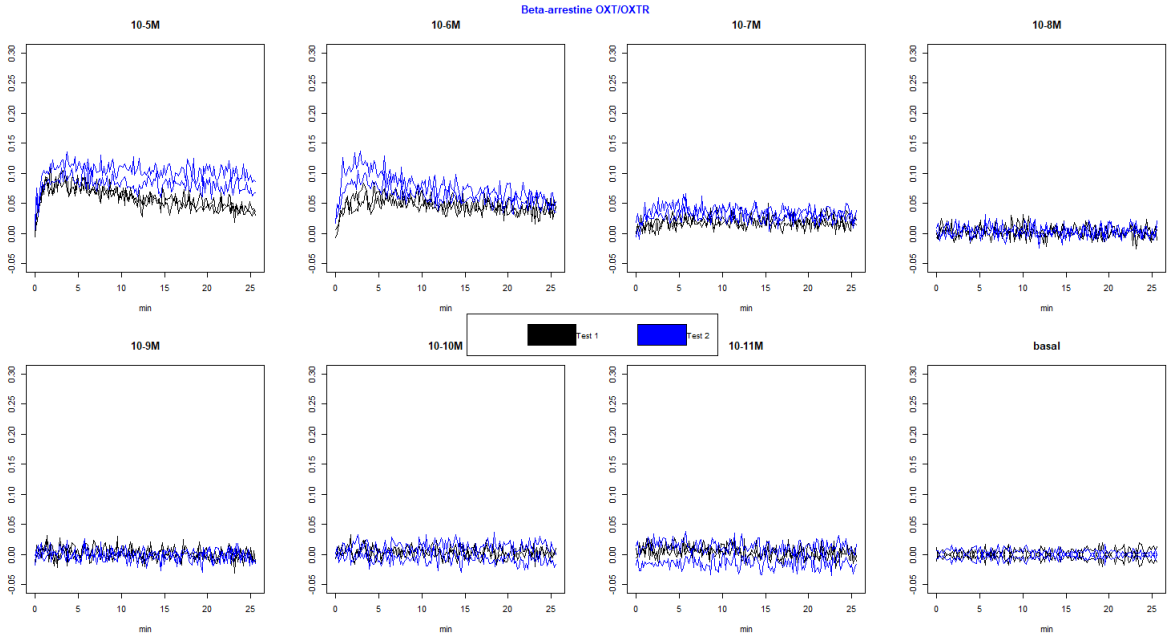

FIGURE 4.2 : Courbe cinétique beta-arrestine du couple OXT/OXTR

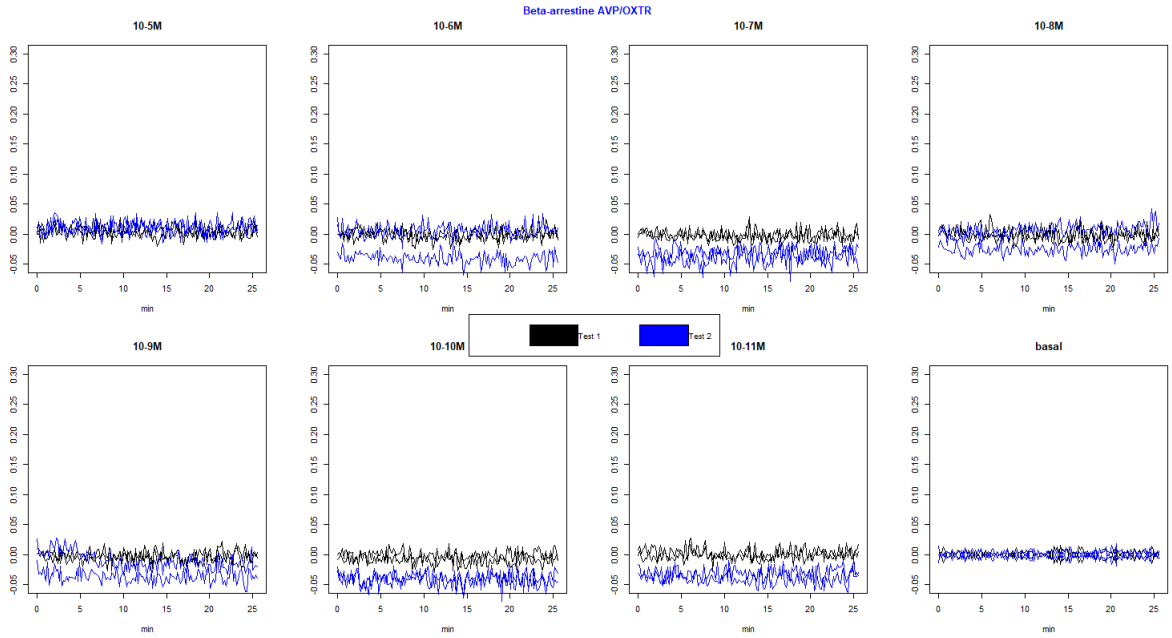

FIGURE 4.3 : Courbe cinétique beta-arrestine du couple AVP/OXTR

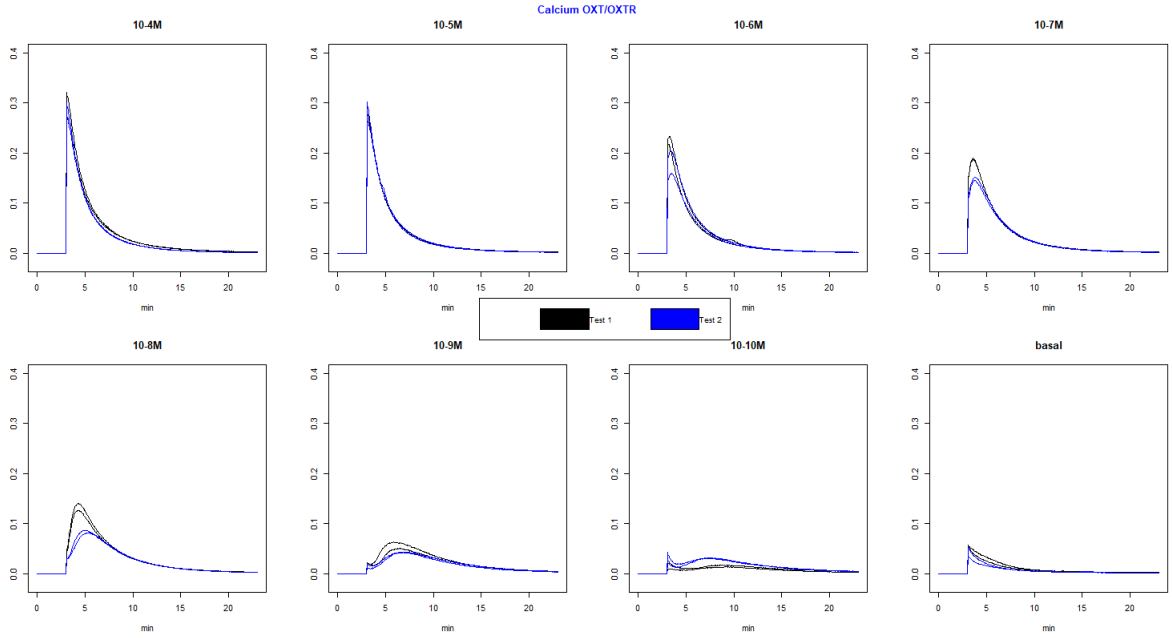

FIGURE 4.4 : Courbe cinétique calcium du couple OXT/OXTR

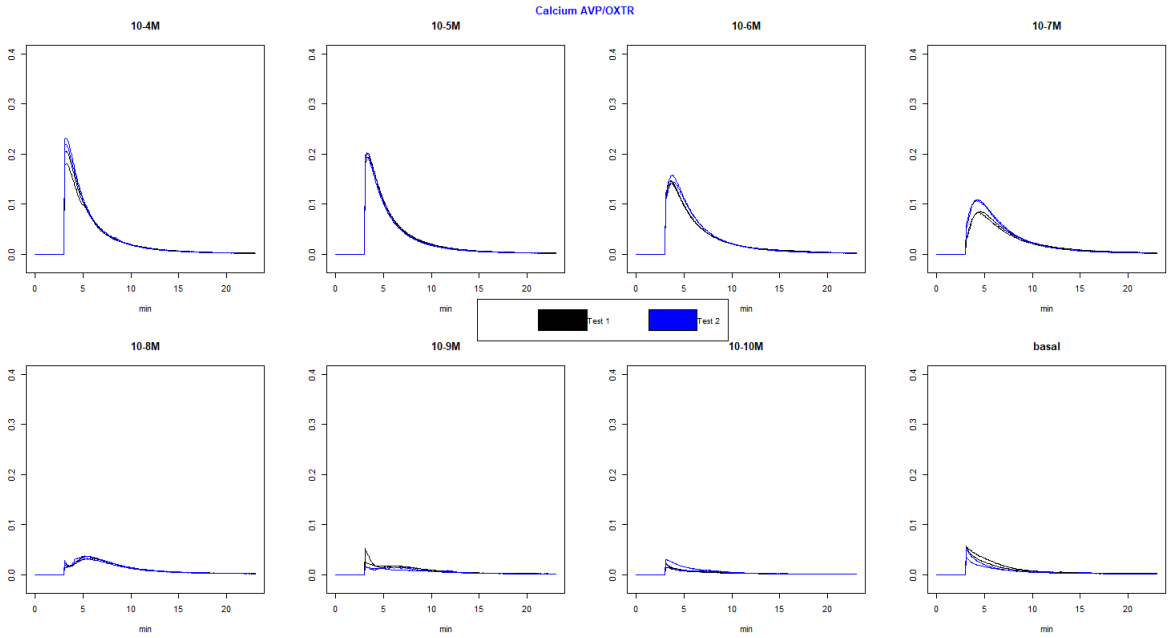

FIGURE 4.5 : Courbe cinétique calcium du couple AVP/OXTR

D'après les graphiques un premier tableau récapitulatif permettant de voir quels ligands agit sur quels récepteurs est construit. Par la suite, l'objectif est de savoir à quels vitesse les ligands agissent sur chaque récepteur.

ligand + récepteur	Ca	$\beta arrestine$
OXT + OXTR	+	+
OXT + AV1A	+	-
OXT + AV1B	+	-
OXT + AV2	+	-
AVP + OXTR	+	-
AVP + AV1A	+	+
AVP + AV1B	+	+
AVP + AV2	+	+

L'analyse des données brutes est placée en annexe ((cf.Annexe A.1 page 33).

Chapitre 5

Traitement

Les données vont maintenant êtres traités comme expliqueé dans la section démarches :

- Création d'un modèle de réseaux de réaction
- Construction d'un modèle Statistiques
- Optimisation des paramètres
- identifiabilité et intervalle de confiance

5.1 Création d'un modèle de réseaux de réaction biochimique

Le réseaux de réaction biochimique dont proviennent nos données a pour but d'expliquer la voie activée par la protéine G_q dans le cycle de protéines G. Cela conduit à une libération de calcium à partir des réserves intracellulaires et à un recrutement de beta-arrestine.
Le fonctionnement du cycle de protéines G est étudié plus en détails en expliquant chaque élément du réseau de réaction biochimique final.

1. Réception du ligand et activation de la protéine G.

$$L + R \leftrightarrows LR$$
$$LR + G \leftrightarrows LRG$$

2. la protéine G se dissocient en sous-unités lors de l'activation.

$$LRG \rightarrow LR + G_{\alpha}.GTP + G_{\beta\gamma}$$

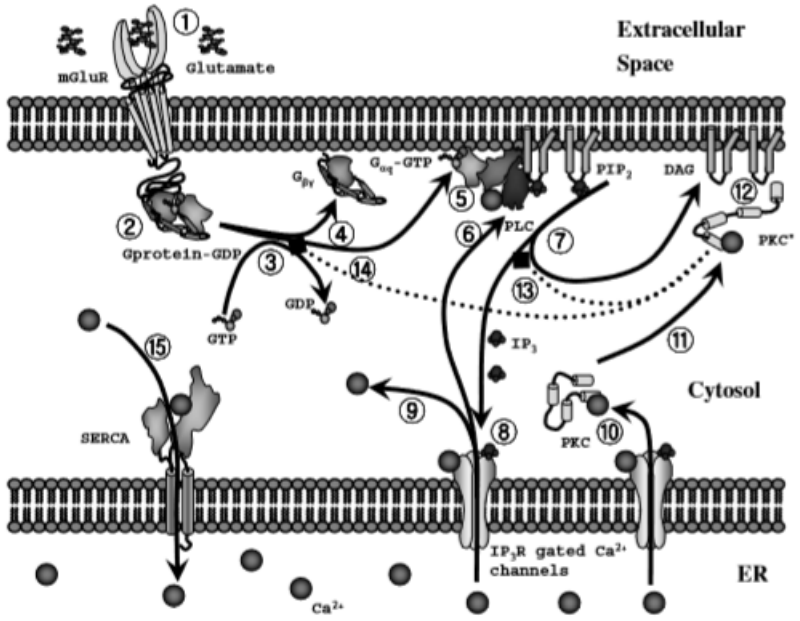

FIGURE 5.1 : Cycle de protéines G

3. $G_\alpha.GTP$ se désactive en hydrolysant GTP en GDP son (GTP lié perd une molécule de phosphate, grâce à une capacité intrinsèque de la sous-unité α, appelée activité GTPase).

$$G_\alpha.GTP \rightarrow G_\alpha.GDP$$
$$G_\alpha.GDP + G_{\beta\gamma} \rightarrow G$$

4. Translocation de la protéine G_q active vers le PLC (PhospholiPase C).

$$G_\alpha.GTP + PLC \leftrightarrows G_\alpha.GTPPLC$$
$$Ca^{2+} + G_\alpha.GTPPLC \leftrightarrows G_\alpha.GTPPLC^*$$

5. P_IP_2 (Phosphatidylinositol biphosphate) est séparé par le PLC en IP_3 (Inositol tri phosphate) et DAG (DiAcylGlycérol).

$$G_\alpha.GTPPLC^* + P_IP_2 \leftrightarrows G_\alpha.GTPPLC^*P_IP_2$$
$$G_\alpha.GTPPLC^*P_IP_2 \rightarrow G_\alpha.GTPPLC^* + IP_3 + DAG$$

6. IP_3 s'associe à un canal du réticulum endoplasmique[1] (ER).

$$IP3 + IP3R \leftrightarrows IP3IP3R$$
$$IP3IP3R + Ca^{2+} \leftrightarrows IP3IP3RCa^2$$
$$IP3IP3RCa^{2+} + Ca^{2+} \leftrightarrows IP3IP3RCa^{2+}Ca^2$$

7. Libération de Ca^{2+} (ion calcium)

$$\frac{dCa^{2+}}{dt} = K(1 + \alpha[IP3IP3RCa^{2+}])(Ca^{2+}_{tot} - [Ca^{2+}]) - \frac{P_1[Ca^{2+}]^2}{[Ca^{2+}]^2 + P_2^2}$$

8. DAG reste dans la membrane active PKC (Protéine Kinase Calcium dépendante).

$$DAG + PKC \leftrightarrows PKC^*$$

9. PKC engendre alors une cascade de phosphorylation.

$$LR + PKC^* \rightarrow LR_p + PKC$$

10. Mais aussi cascade de phosphorylation par réaction basal :

$$LR + GRK \leftrightarrows LR_p$$

a) Désensibilisation du RCPG :
Une augmentation de la fixation de protéines de beta-arrestines au récepteur qui empêchent ainsi le couplage du RCPG avec sa protéine G (beta-arrestine prend la place de la protéine G sous le récepteur).

b) Endocytose du RCPG :
Une fois à la membrane, la beta-arrestine fixent le RCPG aux éléments d'internalisation (la clathrine et l'adaptateur de la clathrine AP2).

c) Le récepteur internalisé peut alors suivre deux voies :
Soit il est dirigé vers des endosomes de recyclage vers la membrane plasmique (resensibilisation) où il est de nouveau apte à percevoir le signal (complexe [RCPG / -arrestine] faible)
Soit il est orienté vers le compartiment lysosomial pour y être dégradé (complexe [RCPG / -arrestine] fort).

[1] réservoir de Ca^{2+} qui échange du Ca^{2+} avec le cytosol via des canaux sensibles à l'IP_3, des pompes à calcium et des fuites.

$$LR_p + \beta \rightarrow LR_p\beta$$
$$LR_p\beta \rightarrow \emptyset$$

Par conséquent, un modèle détaillé peut être utilisé pour faire la lumière sur les mécanismes importants dans la voie :

$$\begin{aligned}
L + R &\leftrightarrows LR \\
LR + G &\leftrightarrows LRG \\
LRG &\rightarrow LR + G_\alpha.GTP + G_{\beta\gamma} \\
G_\alpha.GTP &\rightarrow G_\alpha.GDP \\
G_\alpha.GDP + G_{\beta\gamma} &\rightarrow G \\
G_\alpha.GTP + PLC &\leftrightarrows G_\alpha.GTPPLC \\
Ca^{2+} + G_\alpha.GTPPLC &\leftrightarrows G_\alpha.GTPPLC^* \\
G_\alpha.GTPPLC^* + P_IP_2 &\leftrightarrows G_\alpha.GTPPLC^*P_IP_2 \\
G_\alpha.GTPPLC^*P_IP_2 &\rightarrow G_\alpha.GTPPLC^* + IP_3 + DAG \\
IP_3 &\rightarrow \emptyset \\
DAG &\rightarrow \emptyset \\
IP3 + IP3R &\leftrightarrows IP3IP3R \\
IP3IP3R + Ca^{2+} &\leftrightarrows IP3IP3RCa^{2+} \\
IP3IP3RCa^{2+} + Ca^{2+} &\leftrightarrows IP3IP3RCa^{2+}Ca^2 \\
\frac{dCa^{2+}}{dt} &= K(1 + \alpha[IP3IP3RCa^{2+}])(Ca^{2+}_{tot} - [Ca^{2+}]) - \frac{P_1[Ca^{2+}]^2}{[Ca^{2+}]^2 + P_2^2} \\
DAG + PKC &\leftrightarrows PKC^* \\
LR + PKC^* &\rightarrow LR_p + PKC \\
LR + GRK &\leftrightarrows LR_p \\
LR_p + \beta &\rightarrow LR_p\beta \\
LR_p\beta &\rightarrow \emptyset
\end{aligned}$$

Le modèle est simplifié (Modèle 1) :

$$
\begin{aligned}
L + R &\leftrightarrows LR \\
LR + G &\leftrightarrows LRG \\
Ca^{2+} + LRG &\rightarrow LR + G + Ca^{2+} + IP3 + DAG \\
IP_3 &\rightarrow \emptyset \\
DAG &\rightarrow \emptyset \\
IP3 + IP3R &\leftrightarrows IP3IP3R \\
IP3IP3R + Ca^{2+} &\leftrightarrows IP3IP3RCa^{2+} \\
IP3IP3RCa^{2+} + Ca^{2+} &\leftrightarrows IP3IP3RCa^{2+}Ca^{2} \\
Ca^{2+} &\rightarrow \emptyset (\text{vitesse de réaction : } \frac{P_1[Ca^{2+}]^2}{[Ca^{2+}]^2 + P_2^2}) \\
\emptyset &\rightarrow Ca^{2+} (\text{vitesse de réaction : } K(1 + \alpha[IP3IP3RCa^{2+}])(Ca_{tot}^{2+} - [Ca^{2+}])) \\
RL + DAG &\rightarrow LR_p \\
LR &\leftrightarrows LR_p \\
LR_p + \beta &\rightarrow LR_p\beta \\
LR_p\beta &\rightarrow \emptyset
\end{aligned}
$$

Le modèle est simplifié de nouveau (Modèle 2) :

$$
\begin{aligned}
L + R &\leftrightarrows LR \\
Ca^{2+} + LR &\rightarrow LR + Ca^{2+} + IP3 \\
IP_3 &\rightarrow \emptyset \\
Ca^{2+} &\rightarrow \emptyset (\text{vitesse de réaction : } \frac{P_1[Ca^{2+}]^2}{[Ca^{2+}]^2 + P_2^2}) \\
\emptyset &\rightarrow Ca^{2+} (\text{vitesse de réaction : } K(1 + \alpha[IP3] * [Ca^{2+}])(Ca_{tot}^{2+} - [Ca^{2+}])) \\
LR + \beta &\rightarrow LR\beta \\
LR\beta &\rightarrow \emptyset
\end{aligned}
$$

Le modèle est simplifié une dernière fois (Modèle 3) :

$$
\begin{aligned}
L + R &\leftrightarrows LR \\
Ca^{2+} &\rightarrow \emptyset (\text{vitesse de réaction : } \frac{P_1[Ca^{2+}]^2}{[Ca^{2+}]^2 + P_2^2}) \\
\emptyset &\rightarrow Ca^{2+} (\text{vitesse de réaction : } K(1 + \alpha[LR] * [Ca^{2+}])(Ca_{tot}^{2+} - [Ca^{2+}])) \\
LR + \beta &\rightarrow LR\beta \\
LR\beta &\rightarrow \emptyset
\end{aligned}
$$

5.2 Construction du modèle statistique

Le logiciel D2D (cf.Annexe A.4.1 page 39) servant à déterminer les paramètres θ du modèle de réseaux de réactions est basé sut un modèle statistiques gaussien. Ainsi pour pouvoir utiliser ce logiciel, les données de beta-arrestine et de calcium doivent suivre ce modèle. La normalité des données est alors testé sur les données de beta-arrestine et de calcium (mais aussi sur les données brutes : (cf.Annexe A.2 page 36)).

Pour cela, le test de Lillefort est utilisé (il semble être le test de normalisé le plus adéquat (cf.Annexe B.7 page 86)).

Les données actuelle dépende du facteur temps. Afin de supprimer ce facteur, pour chaque test et à chaque point de temps, la moyenne des deux réplicats et retranché à la valeur de chaque point de temps :

$$x_{i,j} = x_{i,j} - M_{i,j}$$

où M est la moyenne des deux réplicats issus de la même série pour chaque point de temps.

Une fois le facteur temps enlever on obtiens le bruit de chaque mesures notre but étant d'en trouver la loi de distribution (un programme général est construit (cf.Annexe C.3.1 page 114)).

5.2.1 Beta-arrestine

Ainsi pour chaque réplicat de chaque données, le test de Lillefort est appliqué. Puis le nuage de point l'histogramme et le qqplot sont tracés (programme :(cf.Annexe C.3.2 page 117)).
Ici seules sont présenté les résultat sur le récepteur OXT pour les test statistiques (autre test (cf.Annexe C.3.2 page 117)) :

```
>                                  #OXTR OXT
> X=ba.OO
> #les tests renvoie le nom des colonnes qui ont une p-value<0.5
> #(ne suivent pas une loi normal)
> which(LL.bruit(e(X))<0.05) #Test de Lilliefort
integer(0)
>                               #OXTR AVP
> X=ba_oa
> which(LL.bruit(e(X))<0.05) #Test de Lilliefort
integer(0)
```

Ici seules les courbes du récepteur OXT à une dose 10^{-5} sont tracés (autres graphiques :(cf.Annexe D.4.1 page 176)) :

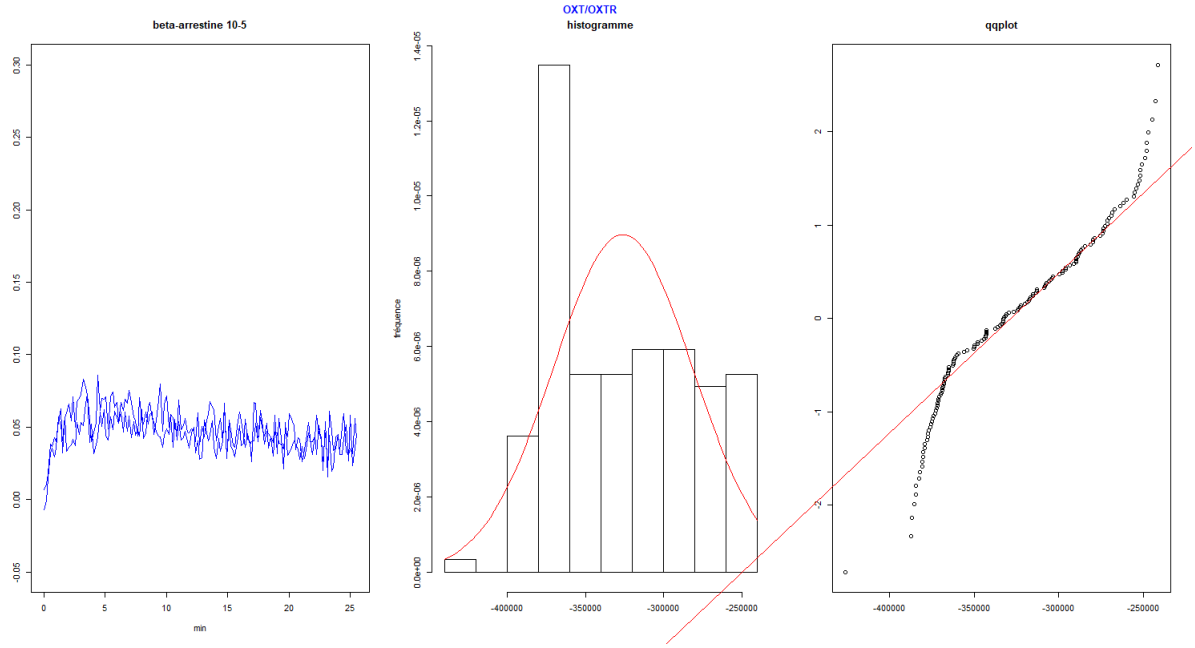

FIGURE 5.2 : Test de normalité beta-arrestine du couple OXT/OXTR à une dose 10^{-5}

Par conséquent les données de beta-arrestine normalisé suivent une loi normale.

5.2.2 Calcium

Dans le cas des données calcium, le test de Lillefort est également appliqué. Cependant les résultats semblent indiquer un rejet de l'hypothèse de normalité pour un seuil de 95%. Le test est

alors appliqué en passant les données au logarithme. Les mesures de formes(comparaison de la forme de l'échantillon avec la loi normal) sont calculé pour plus d'information. Ici les tests ne s'applique que pour le couple OXT/OXTR (autre test (cf.Annexe C.3.3 page 118)) :

```
> library(e1071) #utile pour  les mesures de formes
> #OXTR OXT
> X=c_oo
> skewness(e(X))
[1] 7.266606
#asymtrique
> kurtosis(e(X))
[1] 54.68088
#moins aplatit donc plus concentre
> which(LL.bruit(e(X))<0.05)
 [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19
 20 21 22 23 24 25 26 27 28 29
[30] 30 31 32

>                    #Calcium suit loi log-normal?
> X=X[which(X[,1]==15):which(X[,1]==3),]
> a=log(abs(e(X))+0.000000001)
> skewness(a)
[1] 1.375581
#plus proche de la loi normale mais asymtrique
> kurtosis(a)
[1] 7.646124
#plus proche de la loi normale mais toujours moins aplatit
#et plus concentre
> which(LL.bruit(a)<0.05)
 [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19
 20 21 22 23 24 25 26 27 28 29 30 31 32
```

Pour une meilleure visualisation le nuage de point, l'histogramme et le qqplot sont tracés (programme :(cf.Annexe C.3.3 page 118)). Ici seules les courbes du couple OXT/OXTR à une dose 10^{-5} sont tracé (autres graphiques :(cf.Annexe D.4.2 page 176)) :

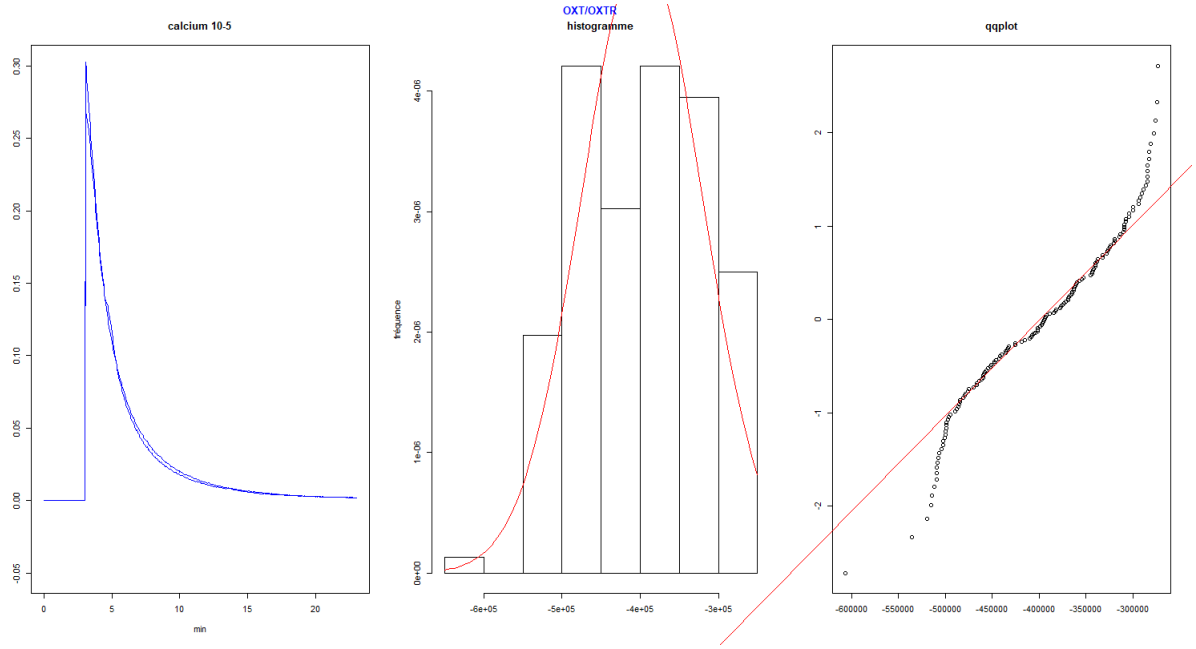

FIGURE 5.3 : Test de normalité calcium du couple OXT/OXTR à une dose 10^{-5}

Ainsi, les données calciums ne semblent pas suivre une loi normal ou log-normal (même en variant le seuil jusqu'à 90%). Une normalisation différentes des données permettraient peut-être d'obtenir une distribution gaussienne du jeux de données mais dans un soucis de temps, ces données seront supposer normalement distribuées par la suite.
Le modèle statistique gaussien additif est ainsi obtenue :

$$Y_{obs} = Y_{obs} + \beta \text{ où } \beta \text{ : bruit } (\sim \mathcal{N}(\mu, \sigma^2))$$

5.3 Optimisation

Comme le modèle statistique est un modèle gaussien additif, l'utilisation du logiciel D2D est alors possible. Il est constitué d'un fichier Models reprenant le modèle du réseau de réaction du chapitre précedent et d'un fichier Data reprenant les jeux de données issus des expériences (pour plus d'information : (cf.Annexe A.4.1 page 39) et pour voir les codes : (cf.Annexe A.4.1 page 39)).
Ce logiciel permet de faire une optimisation (issue d'algorithme hybride basé sur le gradient) à l'aide de l'estimateur du maximum de vraisemblance.

Le modèle 1 est fitté 100 fois. Les paramètre θ obtenue

Annexe A

Complément

A.1 Analyse descriptive des données brutes

(Retour au rapport : 4.Statistiques descriptives) Pour l'instant seules les données normalisés ont été traité, mais nous avons pu avoir les données brutes. Notre étude se porte alors sur celles-ci et les étapes de la normalisation.

A.1.1 beta-arrestine

(programme : (cf.Annexe C.2.3 page 107), autre graphique : (cf.Annexe D.3 page 165)).

Comme vu précédemment la mesure du recrutement de beta arretine dépend de deux mesures : mesure du Rluc et de la Venus.

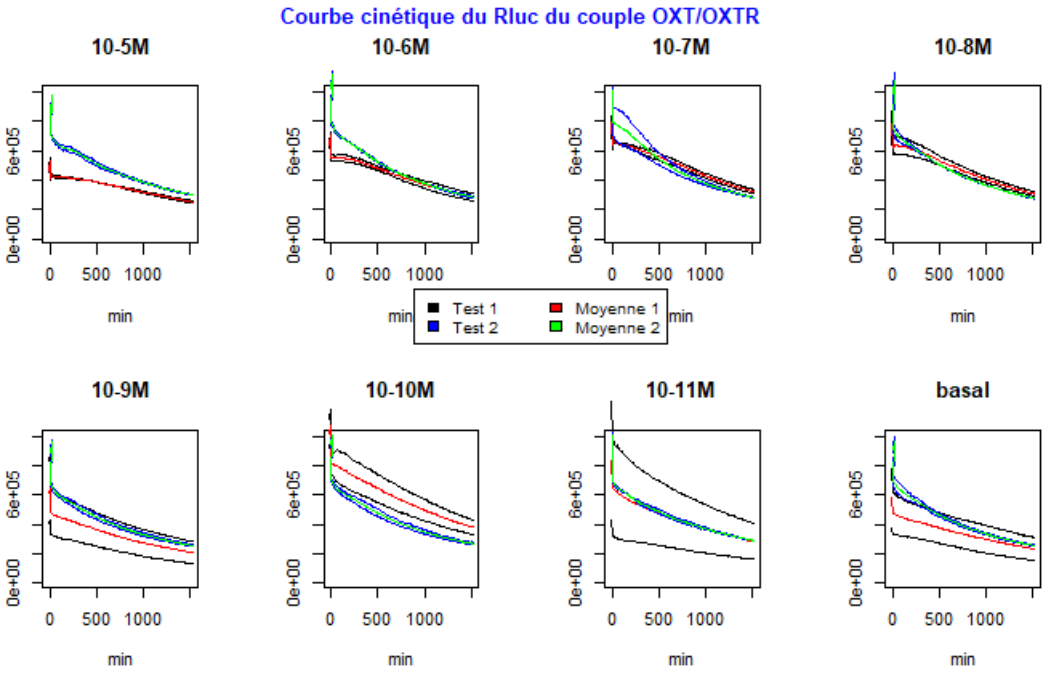

FIGURE A.1 : Courbe cinétique Rluc du couple OXT/OXTR

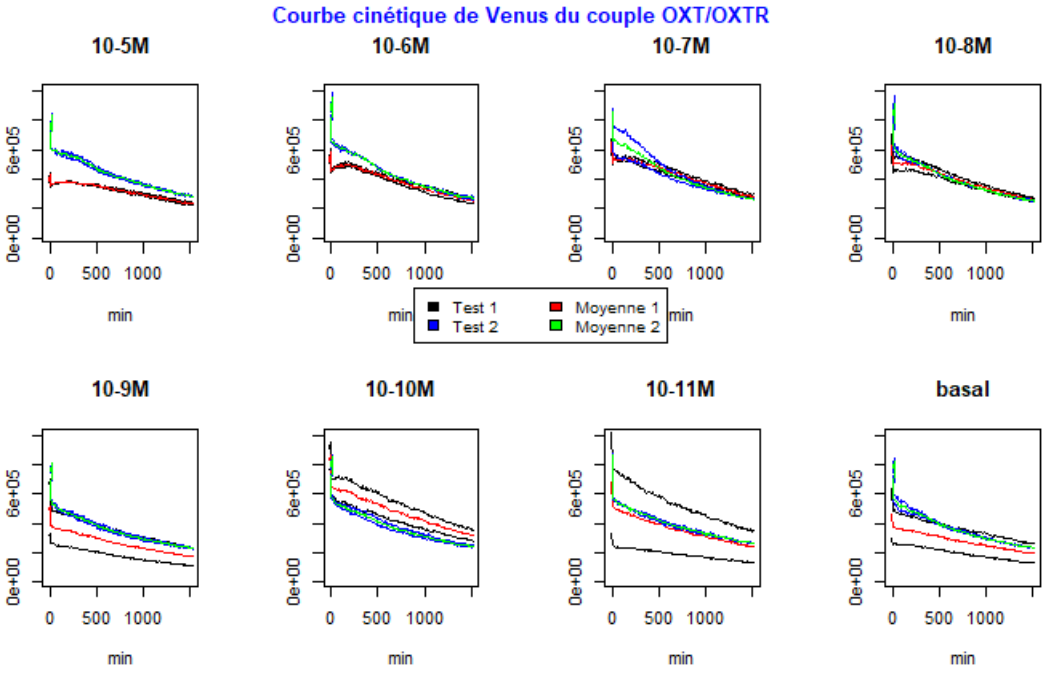

FIGURE A.2 : Courbe cinétique Venus du couple OXT/OXTR

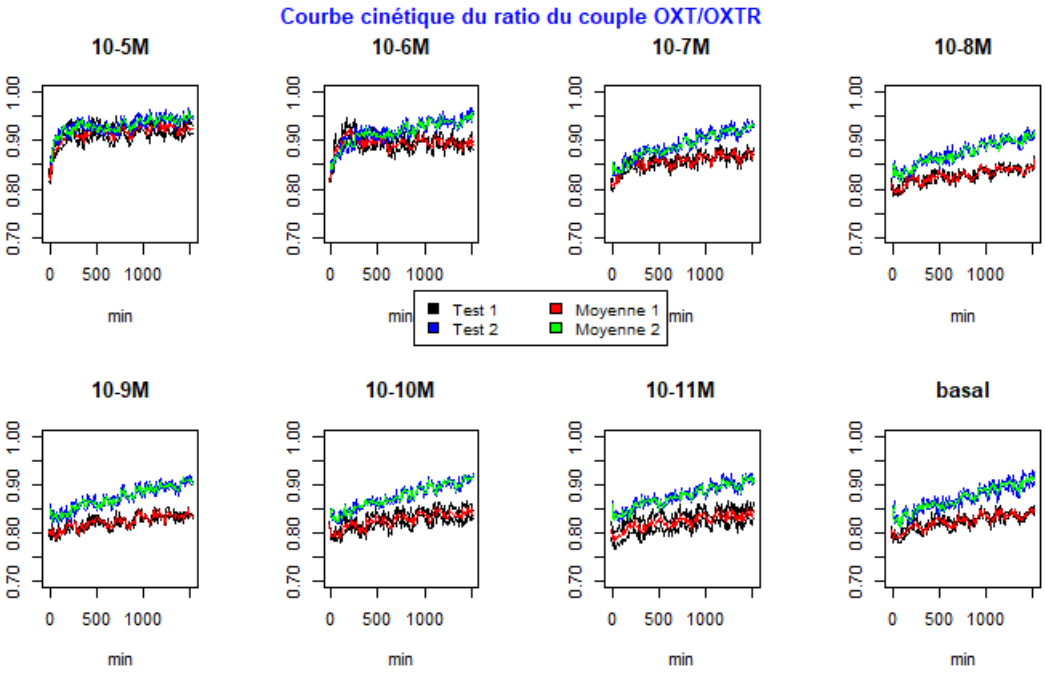

FIGURE A.3 : Courbe cinétique du ratio du couple OXT/OXTR

A.1.2 Calcium

(programme : (cf.Annexe C.2.3 page 111), autre graphique : (cf.Annexe D.3 page 176)).

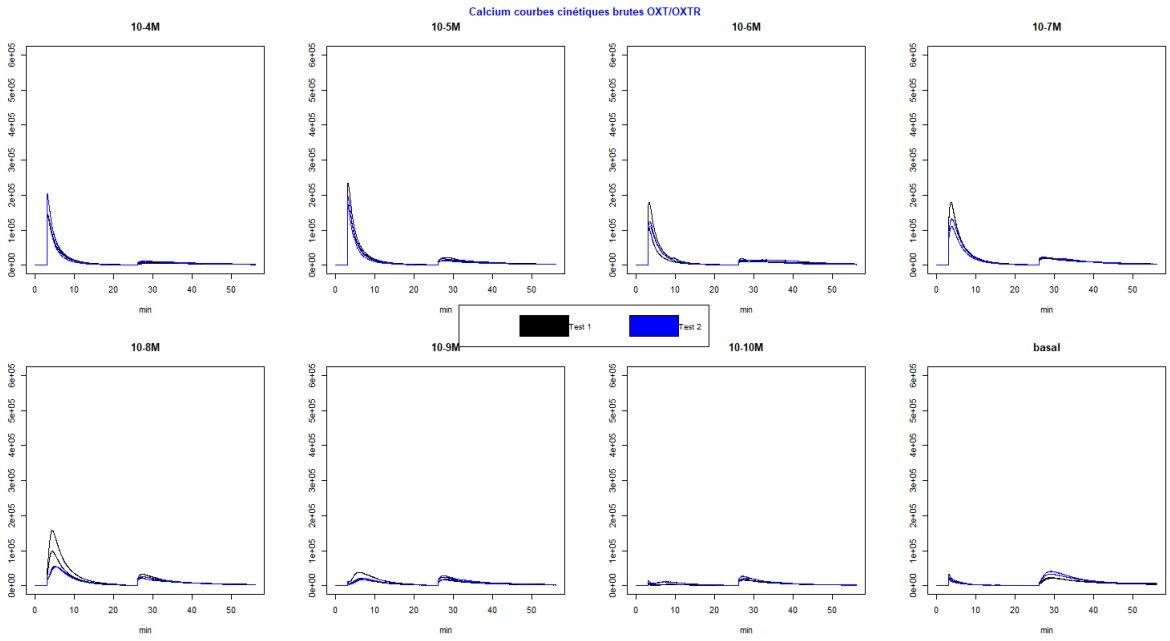

FIGURE A.4 : Courbe cinétique des données calcium brut du couple OXT/OXTR

A.2 Test normalité des données brutes

(Retour au rapport : 5.2.Construction du modèle statistique)

A.3 Expériences

(Retour au rapport : 2.La démarche)

Pour le calicum, on ajoute une enzyme qui agit avec la colotérazine qui en présence de calium émet de la lumière. Ces enzymes sont transfecté dans les cellules.

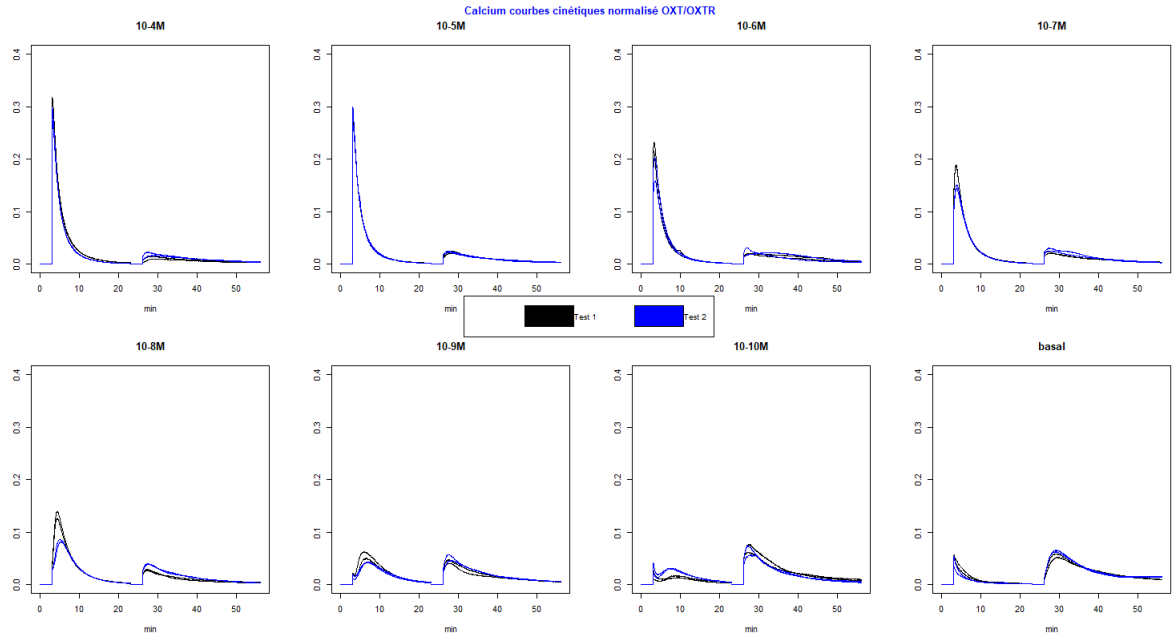

FIGURE A.5 : Courbe cinétique des données calcium normalisée du couple OXT/OXTR

A.3.1 Transfection

La transfection est un processus de transfert de gènes, c'est-à-dire l'introduction de matériel génétique exogène dans des cellules eucaryotes (n'utilisant pas comme vecteur un virus).
Au sein des laboratoires de l'INRA, cette manipulation se fait via des cellule rénale humaine de type cancéreuse permettant une vitesse de division cellulaire rapide et une tansfection simple en terme de manipulation.
Ayant assisté à une transfection, les étapes de cette manpulation sont donc détaillé par la suite.
Un manipulation est répété pour chaque ADN de récepteur (chacune dans une fiole différentes : OXTR,AV1A,AV1B,AV2) :

1. ajout du milieu + l'ADN de la beta-arrestine fluo (=Venus).
2. ajout du métafectan et incubation pendant 15 min (permettant aux cellules d'assimiler les ADN)
3. mise dans les puits
4. ajout des cellules
5. mise en incubation

Pendant le temps d'incubation (en général), le traitement des cellules est effectué :

1. Premier lavage des cellules : enlever le serum (attention une cellules sans serum ne vie pas plus de 48h)
2. Ajout du tampon
3. ajout de l'enzyme
4. Ajout du milieu avec serum
5. Dilution du milieu avec serum

Une fois la transfection terminé vient la deuxième étape : la pharmacologie. Pour ce faire, la plaque de 96 puits (en incubation depuis la fin de la transfection) est vidé du sérum la veille de la partie pharmaco.

A.3.2 pharmacologie

Une nouvelle plaque (blanche cette fois-ci) est utilisé pour laquelle, dans chaque puits, le substrat est pipeté ensuite les solutions sont ajouté avant de mettre la boite dans la MITHRAS.
Le MITRAS permet d'utiliser la technique du BRET ((Bioluminescence Resonance Energy Transfer) qui mesure la luminescence sans excitation des cellules : Le donneur est une molécule luminescente, la coelenterazine, dont l'excitation est déclenchée par une enzyme, la Renilla Luciferase (Rluc), tandis que l'accepteur peut être une protéine fluorescente comme la Venus. Le BRET dépend de la distance séparant le donneur de l'accepteur. Lorsque la Rluc et la Venus se trouvent à proximité (d < 10 nm), un transfert d'énergie peut avoir lieu entre les deux molécules avec l'émission d'une fluorescence caractéristique à 530 nm. L'absence de toute interaction se traduit par un spectre d'émission avec un pic centré à 480 nm. Toutefois, si les deux protéines sont en interaction un deuxième pic caractéristique du BRET apparaît aux environs de 530 nm. Ce pic correspond à l'émission de la Venus excitée par le transfert d'énergie provenant de la réaction enzymatique.
Afin de vérifié la présence de Venus dans les cellules, le Tritar est utilisé. Il excite les venus et mesure leur indice de fluorescence. Cette indice doit rester à peu prés constant au cours du temps. Il existe un biais dans la mesure des bata-arrestine puisque'il en existe déjà dans la cellule (endogène) qui on un effet sur l'expérience mais sont immesurables.

A.4 logiciel utilisé

(Retour au rapport : 5.2.Construction du modèle statistique)

A.4.1 D2D

Data2Dynamics est un logiciel de modélisation adapté à l'estimation des paramètres dans les systèmes dynamiques développés pour les réseaux de réactions biochimiques, mais n'est pas limité à cela.
Le logiciel est un ensemble de méthodes numériques pour la modélisation dynamique quantitative et un langage complet de description de modèle et de données. Pour l'étalonnage du modèle, c'est-à-dire l'estimation des paramètres, des algorithmes d'optimisation numérique stochastique et déterministe peuvent être utilisés.Actuellement, sept optimiseurs sont disponibles comprennant des méthodes stochastiques (par exemple différentes réalisations de l'optimisation de l'essaim de particules, des stratégies évolutives et des stratégies d'escalade). Pour l'estimation des paramètres, un algorithme d'optimisation déterministe (LSQNONLIN) basé sur des sensibilités évaluées, qui sont intégrées avec le système ODE est utilisé par défaut. Cette approche est la plus efficace et fiable pour les problèmes typiques de biologie des systèmes (http ://journals.plos.org/plosone/article ?id=10.1371/journal.pone.0074335). L'identifiabilité améliore la prévisibilité d'un modèle et a des conséquences sur les prédictions du modèle en termes d' observabilité. Pour l'analyse d'incertitude du paramètre et des prédictions du modèle, l'approche de vraisemblance de profil et les approches d'échantillonnage de Monte Carlo de chaîne de Markov sont disponibles.

Diverses approches sont proposées :

1. Pour le bruit de mesure des données expérimentales, il peut être estimé avec la dynamique du modèle.
2. Pour les entrées de modèle, elles peuvent être implémentées sous forme de fonctions paramétrées et peuvent être estimées avec la dynamique du modèle.
3. Pour l'étalonnage de modèle (estimation de paramètre), des algorithmes d'optimisation numérique stochastique et déterministe peuvent être utilisés.
4. Pour l'analyse d'incertitude du paramètre et des prédictions du modèle, l'approche de vraisemblance de profil et les approches d'échantillonnage de Monte Carlo de chaîne de Markov sont utilisé.
5. Pour une solution numérique et des calculs de dérivation, des solveurs numériques parallélisés sont implémentés.

Notez que les paramètres estimés et le test de qualité d'ajustement dépendent des erreurs de mesure. Ce test évalue principalement si les erreurs correspondent et n'est donc pas recommandé

pour évaluer si la dynamique du modèle correspond aux données.Une méthode pour comparer plusieurs modèles est donnée par le test du rapport de vraisemblance.

Pour lancer le logiciel, il faut un dossier "data" contenant, un document .xls et un document .def. Le premier contient les données sous formes de tableau (chaque espèce ne possède que une colonne et les variances sont dans une autres colonnes). Le deuxième document, contient les condition spécifique au jeu de donnée.

Models

Il faut un autre dossier s'appelant "Models" et contenant un fichier .def . Celui-ci se compose en plusieurs parties (pouvant être vide) :

1. DESCRIPTION : contient le nom du model (Peut être utilisé pour stocker d'autres informations comme la date, la version, les auteus...).

2. PREDICTOR : définit la variable indépendante (généralement le temps).

```
          PREDICTOR
t                 T    "min"      "time"     0    100
```

   t : identificateurs de variables indépendante qui sera utilisé dans les expressions mathématiques plus tard.
   T : type d'unité
   "min" : l'unité
   "temps" : étiquette utilisée pour les graphiques.
   0 100 : plage de la variable indépendante.

3. COMPARTMENTS : utilisé pour définir des compartiments dans un modèle (en général pas utilisé).

4. STATES : décrit les variables qui évoluent dans le temps.

```
pSTAT5    C    "nM"       "conc."   cyt   1   "phospho STAT5"          1
```

pSTAT5 : identificateurs de variables dynamiques qui seront utilisés dans les expressions mathématiques plus tard (les conditions initiales pour chaque variable dynamique seront générées en tant que paramètres libres en utilisant le préfixe init_).
C : types d'unités (concentration)
"nM" : l'unité
"conc." : étiquette utilisée pour les graphiques.
cyt : compartiment où vit l'état (laissé vide si aucun compartiment).
1 (par défaut) : l'état doit être montré dans les tracés (0 = non, 1 = oui).
"phospho STAT5" : étiquette pour les graphiques.
1 (par défaut) : l'état est strictement positif (0 = non, 1 = oui).

5. INPUTS : la fonction d'entrée (généralement dépendant de la variable indépendante) est spécifié spécifiée. (vide)

6. REACTIONS : Le système ODE est spécifié comme réseau de réaction biochimique.

```
Epo + EpoR      ->  Epo_EpoR    CUSTOM "k1 * Epo * EpoR"    "label1"
```

po + EpoR -> Epo_EpoR : réaction dans une notation simple.
CUSTOM :mot-clé qui spécifie le type de réaction(ou MASSACTION peu recommandée mais pour lequel on peut utiliser <->).
"k1 * Epo * EpoR" : expression mathématique de l'équation de la vitesse de réaction (uniqement dans le cas CUSTOM sinon juste un nom de paramètre correspondant à la constante de vitesse).
"label1" (facultatif) : étiquette.
Il est possible de remplacer REACTIONS par ODES. Dans ce cas, les équations différentielles ordinaires sont directement spécifiées et définissent les côtés droits des ODE (attention il faut autant d'ODE que de variables dynamiques dans le modèle).

7. DERIVED : les variables dérivées des variables dynamiques et d'entrée sont définis définies (généralement vide).

8. OBSERVABLES : spécifie les observables de modèle par défaut et leur modèle d'erreur correspondant (ERRORS).

```
tSTAT5_au  C   "au"  "conc."   1   1   "scale_tSTAT5 * pSTAT5"
```

tSTAT5_au : identificateur observable correspondant à l'en-tête de colonne correspondant dans la feuille de données(et différent des identificateurs de variables dynamiques).
C : types d'unités (concentration).
"au" : l'unité.
"conc." : étiquette.
1 : la valeur maximale dans la feuille de données de l'observable doit être redimensionnée (0 = non, 1 = oui).
1 : les données brutes du tableur et les observables du modèle doivent être comparés sur une échelle log-10 (0 = non, 1 = oui).
"scale_tSTAT5 * pSTAT5" : expression mathématique de la fonction observable.

9. ERRORS : L'amplitude du bruit de mesure (l'écart-type de la distribution normale) est implémenté comme fonction paramétrée (Il est obligatoire de spécifier le bruit de mesure pour chaque observable dans le fichier de définition de données).

```
    tSTAT5_au       "sd_STAT5_au"
```

tSTAT5_au : identificateur donné dans la section observables.
"sd_STAT5_au" : 'expression mathématique de la fonction de bruit (sd indique que les mesures ont le même bruit de mesure).

10. CONDITIONS : conditions des paramètres du modèle peuvent sont spécifiées.

```
 kD                "koff/kon"
init_Epo_EpoR     "0"
```

kD : paramètre cible.
"koff / kon" : expression mathématique qui remplacera le paramètre cible dans le modèle.
La deuxième ligne initialise le paramètre Epo_EpoR à 0.

Toutes les expressions qui ne sont pas définies en tant que compartiments, états ou entrées sont traitées comme des paramètres libres.

Commandes La première étapes consite à initailiser la conole avec **arInit** pour pouvoir charger les fichiers de modèle et de données. Ensuite il existe de nombreuses commandes :

1. **arCompileAll** : lie tous les fichiers de modèle et de données dans les diverses conditions et crée du code C.

2. **arPrint** : liste de tous les paramètres du modèle, leurs types (= free, C = constant, D = dynamic, I = initial value, E = error model), leurs noms, leurs valeurs actuelles, limite supérieure (ub) et limite inférieure (lb), ...

```
                      name                         lb        value        ub
#   1|DI  | CISEqc                    |       -3        +2.6          +4 | 1    +
#   2|DI  | CISEqcOE                  |       -3       -0.28          +3 | 1
#   3|D   | CISInh                    |       -3        +8.9         +12 | 1    +
```

3. **arPlot** : trace les trajectoires du modèle et des données expérimentales pour les valeurs de paramètres actuelles.

4. **arFit** : Exécution de l'algorithme d'optimisation sur les données expérimentales à l'aide du maximum de vraisemblance.

5. **arFitLHS (n)** Exécution d'une séquence de n Fit. Les résultats des n Fit sont affichés par arPlotFitstri par qualité d'ajustement et les valeurs des paramètres du meilleur ajustement remplacent les valeurs précédentes.

6. **arsave()** : sauvegarde des résultats dans un document workspace.matfichier.

7. **ar.** : stocke les informations importante :

   - **arFit** : fonction objectif
   - **ar.res** : union des résidus provenant des ensembles de données individuels (ar.model.data.res) et des pénalités des paramètres individuels.
   - **ar.ple** : calculs de vraisemblance de profil
   - **arChi2** : évalue la discordance entre le modèle et les données. Cettes fonction intègre l'ODE et met à jour les prédictions du modèle en fonction des paramètres actuels.
   - **ar.ndata** : le nombre de points de données utilisés pour l'ajustement.
   - **ar.config.fiterrors_correction** : la correction de Bessel
   - **ar.chi2errest** : somme de toutes les erreurs de mesure dépendantes des paramètres.
   - **ar.chi2priorest** : somme des pénalités provenant des a priori pour les paramètres.

- **ar.chi2constrest** : pénalité utilisée lorsque des contraintes d'état stable sont introduites.
- **ar.chi2fit** : pénlité utilisé lorsque les paramètres d'erreur sont adaptés. (ar.chi2fit = ar.chi2+ar.chi2err).

ar : les équations du modèle ?

L'optimiseur appelle la fonction objective qui, à son tour, appelle arChi2 pour mettre à jour le modèle et les résidus. Par défaut, l'estimateur lsqnonlin est utilisé pour l'optimisation. Il travaille avec des residus, ie minimise sum(res^2) où res = [ar.res ar.constr]. Si des contraintes d'état stable sont utilisées, la fonction objective devient sum ([ar.res, ar.constr]. $\hat{2}$). Dans le cas de paramètres d'erreur ajustés, il existe une discordance entre la fonction d'objectif utilisée pour l'optimisation (basée sur ar.res et ar.model.data.reserr) et tous les champs chi2 dans l'architecture globale (basés sur ar.model.data.chi2err). L'amplitude de ar.model.data.reserr est proportionnelle à la racine carrée de la contribution des paramètres d'écart-type dans la log-vraisemblance.

L'estimation du maximum de vraisemblance donne des estimations biaisées pour les paramètres d'erreur dans l'échantillon fini. La correction dite de Bessel a été suggérée pour modifier la probabilité de réduire le biais des variances estimées.

Si les erreurs de mesure sont connues, alors la statistique de l'ajustement du khi-deux devrait être du même ordre de grandeur que les degrés de liberté, c'est-à-dire ar.chi2fit devrait être similaire à ar.ndata-sum (ar.qFit == 1). Le test statistique correspondant est le test de qualité d'ajustement fréquemment appliqué qui est implémenté dans d2d sous la forme arChi2Test.

A.4.2 HYPE

HYPE est un logiciel d'optimisation développé par Anne Poupon sur C+ au sein de l'INRA. Cette algorithme utilise une méthode évolutionnaire, c'est à dire un algorithme stochastique s'inspirant de la théorie de l'évolution. L'idée est de faire évoluer un ensemble de solutions à un problème donné, en utilisant itérativement des processus aléatoires, dans l'optique de trouver les meilleurs résultat.
Pour cela, la résolutions des EDO est basé uniquement sur la loi d'action de masse.

De plus, Hype utilise une nouvelle fonction objective non symétrique :

$$f'(P) = \frac{1}{n_m} \sum_{m \in M} \left[\frac{1}{nt^m} \sum_{t \leq nt^m} \frac{|d_m(t) - s_m^p(t)|}{d_m(t)} \left(1 - exp\left(-\frac{1}{2} \frac{\frac{|d_m(t) - s_m^p(t)|^2}{d_m(t)^2}}{\sigma_m(t)^2} \right) \right) \right]$$

nt^m : normalisation car les molécules n'ont pas le même nombre de points.

$\sigma(t)^2$: prend en compte l'écart-type sur les données.

$\frac{|d_m(t)-s_m^p(t)|}{d_m(t)}$: écart entre les valeurs estimées et observées.

En comparaison, la fonction objective de D2D est faite en fonction de l'erreur au carré suivant une grille avec différent point de départs aléatoires, tandis que HYPE, pour avoir une meilleur optimisation que le carré, utilise la valeur absolue.
Comme l'algorithme génétique converge toujours, mais ne se rapproche pas suffisamment de la solution contrairement à CMA-ES qui ne converge pas souvent, mais quand cela arrive, il donne une meilleur approximation de la solution. Ainsi un algorithme hybride combinant ces deux méthodes est utilisé.
Une méthode de Skew-Cauchy ayant pour but de supprimer les paramètre non-identifiable pour augmenter l'identifiabilité des autres est pris en copte dans le programme.
Attention cependant, puisque c'est un logiciel surtout utilisé pour la prédiction. Et il ne faut pas oublier que plus un modèle est complexe, plus les erreurs augmentent.

Annexe B

Théorie

B.1 Modèles de réseaux de réactions biochimiques

(2) (3) (1)

B.1.1 Utilité des modèles mathématiques en biologie

En biologie, les équations différentielles ordinaires (ODE) sont utilisés pour acquérir une compréhension des processus complexes. Les modèles de phénomènes cellulaires prennent souvent la forme de diagrammes d'interactions schématiques.

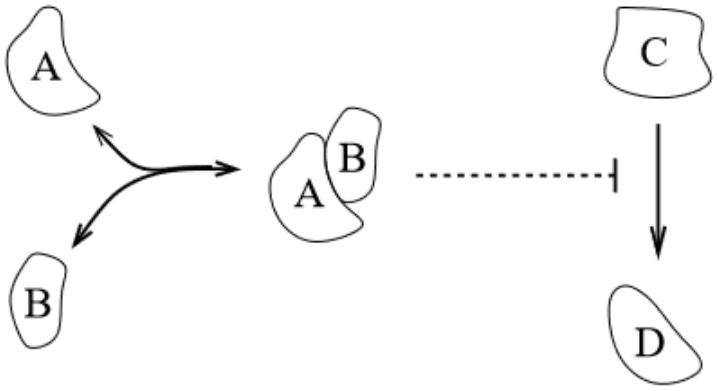

FIGURE B.1 : Exemple de diagramme d'interactions

Un diagramme d'interaction représente les espèces dans un système et indique leurs interac-

tions les unes avec les autres. Les composants de ces diagrammes sont des espèces moléculaires (ex : ions, molécules, macromolécules, complexes moléculaires). Les interactions (flèches) dans le diagramme peuvent représenter une gamme de processus (ex : liaison, déliaison chimique, catalyse de réaction...). La vitesse du processus dépend de la quantité de certaines espèces moléculaires.

Les diagrammes d'interactions laissent généralement des ambiguïtés en ce qui concerne le comportement du système. En utilisant une description mathématique, nous pouvons éliminer l'incertitude dans le comportement du modèle en exigeant une représentation quantitative de chacune des interactions.
Cependant pour un grand nombre de processus cellulaires, seule une compréhension qualitative des interactions moléculaires est possible par contre pour les processus ayant suffisamment de données recueillies le diagramme d'interactions peut être utilisé pour formuler un modèle mathématique dynamique (EDO).

Attention, les modèles résument seulement de façon concise toutes les données et offrent ainsi une vision limitée du comportement du système. De plus, les modèles sont des hypothèses ainsi leurs résultats sont aussi des hypothèses.
Par conséquent, un modèle mathématique a pour avantage de permettre une étude du comportement dynamique du modèle d'une manière et dans des conditions qui ne seraient pas possible en laboratoire (de rapide simulations, pas de coût réel, une observation à tous temps du modèle, fournir des liens entre la structure et le comportement du réseau).

Un modèle mathématique peut servir de deux façon : pour prédire le comportement du système (dans des conditions données) ou pour donner un aperçu général de leurs comportements potentiels. Ainis, bien que les modèles de réseaux cellulaires n'aient qu'un pouvoir prédictif limité, ils sont utiles pour guider le choix des composants et suggérer les expériences les plus efficaces pour tester les performances du système.

Un modèle peut-être stable ce qui signifie que ; ou instable. Les modèles de processus biologiques arrivent presque toujours, à long terme, à des comportements stables (les composants du système inhibent leur propre activité).
En biologie moléculaire, les systèmes étudiés seront complexe (interactions non linéaires). Ces modèles non-linéaires qui apparaissent le plus souvent dans les interactions biochimiques et génétiques sont des saturations(ex : saturation hyperbolique et saturation sigmoïdale).

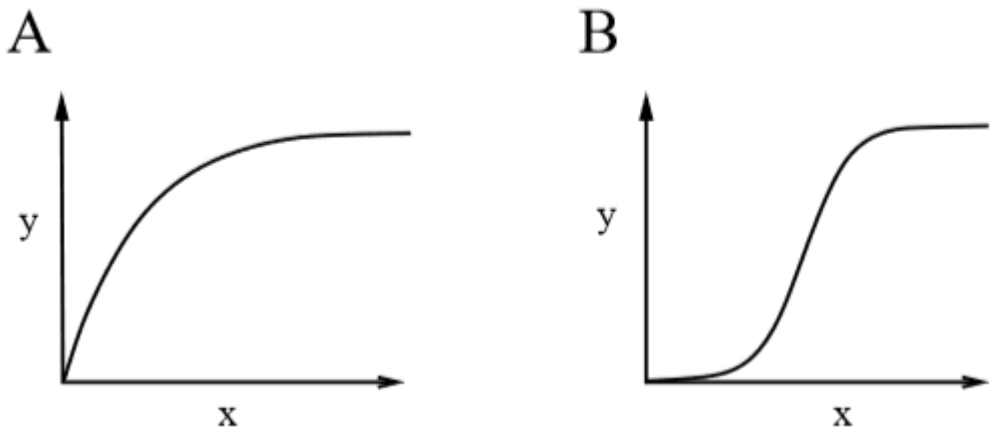

FIGURE B.2 : A : saturation sigmoïdale, B : saturation hyperbolique

B.1.2 Construction d'un modèle mathématique

La construction du modèle mathématique s'effectue en partant de réseaux de réactions biochimiques.

réseaux de réactions biochimiques

Un réseau de réaction biochimique est un langage général pour définir la structure des modèles en biologie des systèmes. Il a l'avantage d'être un langage commun entre les diverses communautés, biologistes, mathématiciens, informaticiens, chimistes qui travaillent en biologie de systèmes.
A partir d'un graphique de réaction (cf Figure B.3),

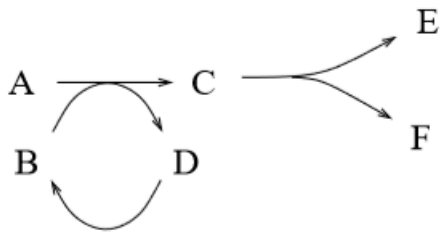

FIGURE B.3 : Exemple de graphique de réaction

on peux en déduire son réseau de réaction chimique :

$$\begin{aligned} A + B &\longrightarrow C \\ D &\longrightarrow B \\ C &\longrightarrow E + F \end{aligned}$$

Les lois de thermodynamique exigent que toutes les réactions chimiques soient réversibles. Cependant, une réaction inverse est possible et se déroule à un rythme négligeable ainsi elle est supposée irréversible. L'hypothèse de l'irréversibilité est donc importante à poser.

Il existe deux types de réseau de réaction :

- réseau fermé :
  pas de réactions avec les produits (ou réactifs) se trouvant en dehors du réseau. Cet état est appelé un équilibre thermique.

- réseau ouvert :
  échange du matériel avec l'environnement extérieur et atteint un état stable (flux constant à travers le réseau). Cet état est appelé un équilibre dynamique.
  Le réseau de réaction chimique fermé précédent peut-être complété pour en obtenir un ouvert :

$$\begin{aligned} A + B &\longrightarrow C \\ D &\longrightarrow B \\ C &\longrightarrow E + F \\ \emptyset &\longrightarrow A \\ E &\longrightarrow \emptyset \\ F &\longrightarrow \emptyset \end{aligned}$$

constante cinétique

Pour prédire les variations de concentration au cours du temps, nous devons connaître la vitesse des réactions qui dépend des concentrations des réactifs.
La vitesse de réaction dépend de deux hypothèses :

- Hypothèse 1 :homogénéité spatiale.
  Le volume de réaction est bien agité (réactifs sont repartirent équitablement dans le volume) rendant la vitesse de réaction indépendante de la position. Cela se produit dans des cuves de réaction agitées en laboratoire.

- Hypothèse 2 :hypothèse de continuité.
  Il y a un grand nombre de molécules de chaque espèce, donc la quantité moléculaire peut-être décrite par une concentration qui varie continuellement. Cette hypothèse convient aux espèces cellulaires dont le nombre de molécules est supérieur ou égal à des milliers.

Pour notre cadre d'étude, nous supposerons donc ces deux hypothèse vrai.

B.1.3 Équations différentielles ordinaires

Sous ses hypothèses, il est possible de faire un lien entre les réseau de réaction chimique et les EDO grâce à la loi de l'action de masse : la vitesse d'une réaction est proportionnelle aux concentrations de chacune des substances réactantes. Montrons quelques exemples simple où k_0, k_1 et k_2 sont les constantes cinétiques de réaction :
conversion irréversible :

$$X \to P \Longrightarrow \frac{d[P]}{dt} = k_1[X]$$
$$A + B \to C \Longrightarrow \frac{d[C]}{dt} = k_2[A][B]$$

Désintégration :

$$A \to \emptyset \Longrightarrow \frac{d[A]}{dt} = -k_1[A]$$

Production et désintégration :

$$\emptyset \to A \to \emptyset \Longrightarrow \frac{d[A]}{dt} = -k_1[A] + k_0$$

Conversion réversible :

$$\begin{cases} A \to B \\ B \to A \end{cases} \Longrightarrow \begin{cases} \frac{d[A]}{dt} = -k_1[A] + k_2[B] \\ \frac{d[B]}{dt} = k_1[A] - k_2[B] \end{cases}$$

Autres exemples :

$$\begin{cases} C \to A \\ A + B \to C \end{cases} \Longrightarrow \begin{cases} \frac{d[A]}{dt} = -k_1[A][B] + k_2[C] \\ \frac{d[B]}{dt} = -k_1[A][B] \\ \frac{d[C]}{dt} = k_1[A][B] - k_2[C] \end{cases}$$

Cependant il faut compléter l'équation différentielle à l'aide de la concentration au début de l'expérience, soit à $t = 0$ ce qui correspond à la condition initiale.

échelle de temps

Les données issues des processus biochimiques n'ont pas toujours la même vitesse et s'étendent sur plusieurs ordres de grandeur. En effet, les processus plus lents sont considérés comme figés dans le temps tandis que les processus plus rapides sont supposés se produire instantanément. Lors de la construction d'un modèle dynamique, il faut dont choisir une échelle de temps primaire.

Cela influe sur le choix des espèces et des processus à inclure dans le modèle. D'après une règle largement utilisé, une différence dans des échelles de temps d'au moins un ordre de grandeur (c'est-à-dire au moins un facteur de dix) peut être utilisée pour la réduction de modèle.
Les techniques de réduction de modèles par séparation des échelles de temps sont :

- l'élimination des variables lentes : une valeur constante remplace chaque variable lente et est traitée comme un paramètre fixe.

- le traitement des variables rapides, deux approches : l'hypothèse d'équilibre rapide et l'hypothèse d'état quasi-stationnaire.

B.1.4 Exemples

Quelques exemples de modèles de réactions biochimiques sont présentés par la suite.

Cas simple de désintégration

Soit le cas de *Désintégration* :

$$A \to \emptyset \Longrightarrow \frac{d[A]}{dt} = -k_1[A]$$

Le modèle peut être traduite par l'équation suivante (où $N(t) =$ *la concentration au cours du temps* et a est la vitesse de réaction et avec $t_0 = 0$) :

$$\begin{cases} N'(t) &= -a\, N(t) \\ N(t_0) &= N_0 \end{cases}$$

La résolution de l'équation donne :

$$\begin{aligned} N'(t) + aN(t) = 0 &\iff (N(t)e^{at})' = 0 \\ &\iff N(t)e^{at} - N(t_0)e^{at_0} = 0 \\ &\iff N(t) = N_0 e^{-at} \end{aligned}$$

Article Goulet

Dans cette article, l'intérêt se porte sur les récepteurs d'oestrogènes ERa qui sont corrélés au cancer du sein. L'étude s'effectue en plusieurs étapes.

1. La première étape est l'acquisition de données expérimentales.
   D'après celle-ci les ERa contiennent des ligand LBD et des monomères libres. Le LBD semble réagir avec le MBP et forment une fusion de monomère. On cherche à modéliser les monomères LBD (M_1) se transformant en homodimère LBD, les monomère issus de la fusion de lBD et MBP (M_2) se transformant en homodimère de fusion et enfin la création d'hétérodimère. Les molécules étudiées sont donc les suivantes :

FIGURE B.4 : Les cinq protéines du test d'échange de dimères

Comme les monomères se transforment naturellement en dimère (se formant à partir de deux monomères), les équations suivantes sont établies :

$$\left\{ M + M \leftrightarrows D \right. \quad \Longrightarrow \quad \left\{ \begin{array}{rcl} \frac{d[M]}{dt} & = & -2k_1[M]^2 + 2k_2[D] \\ \frac{d[D]}{dt} & = & k_1[M]^2 - k_2[D] \end{array} \right.$$

où D est un dimère et M un monomère.

Supposons que le système soit fermé et posons D_{11} homodimère du LBD et D_{22} homodimère de la fusion de dimère. L'équation suivante est alors obtenue :

$$\left\{ \begin{array}{l} M_1 + M_1 \leftrightarrows D_{11} \\ M_2 + M_2 \leftrightarrows D_{22} \\ M_1 + M_2 \leftrightarrows D_{12} \end{array} \right. \quad \Longrightarrow \quad \left\{ \begin{array}{rcl} \frac{d[M_1]}{dt} & = & -2k_{11}[M_1]^2 + 2k_{12}[D_{11}] - k_{31}[M_1][M_2] + k_{32}[D_{12}] \\ \frac{d[M_2]}{dt} & = & -2k_{21}[M_2]^2 + 2k_{22}[D_{22}] - k_{31}[M_1][M_2] + k_{32}[D_{12}] \\ \frac{d[D_{11}]}{dt} & = & k_{11}[M_1]^2 - k_{12}[D_{11}] \\ \frac{d[D_{22}]}{dt} & = & k_{21}[M_2]^2 - k_{22}[D_{22}] \\ \frac{d[D_{12}]}{dt} & = & k_{31}[M_1][M_2] - k_{32}[D_{12}] \end{array} \right.$$

2. A l'aide de connaissances biologiques et de techniques analytiques, le modèle est réduit. Pour l'instant le modèle dépend de six paramètres et du bruit. Or, d'après des hypothèses de biologies, $M_1 + 2D_{11} + D_{12}$ et $M_2 + 2D_{22} + D_{12}$ sont conservés durant le système. Ce qui est vérifiable mathématiquement :

$$\begin{aligned} \frac{dM_1}{dt} + 2\frac{dD_{11}}{dt} + \frac{dD_{12}}{dt} = & -2k_{11}[M_1]^2 + 2k_{12}[D_{11}] - k_{31}[M_1][M_2] + k_{32}[D_{12}] + 2k_{11}[M_1]^2 \\ & - 2k_{12}[D_{11}] + k_{31}[M_1][M_2] - k_{32}[D_{12}] \\ = & \, 0 \end{aligned}$$

Grâce à cette hypothèse, le modèle est simplifié :

$$\begin{cases} M_1(t_0) + 2D_{11}(t_0) + D_{12}(t_0) &= \alpha_1 \\ M_2(t_0) + 2D_{22}(t_0) + D_{12}(t_0) &= \alpha_2 \end{cases} \Longrightarrow \begin{cases} M_1 &= \alpha_1 - 2D_{11} - D_{12} \\ M_2 &= \alpha_2 - 2D_{22} - D_{12} \end{cases}$$

$$\Longrightarrow \begin{cases} \frac{d[D_{11}]}{dt} &= k_{11}(\alpha_1 - 2D_{11} - D_{12})^2 - k_{12}[D_{11}] \\ \frac{d[D_{22}]}{dt} &= k_{21}(\alpha_2 - 2D_{22} - D_{12})^2 - k_{22}[D_{22}] \\ \frac{d[D_{12}]}{dt} &= k_{31}(\alpha_1 - 2D_{11} - D_{12})(\alpha_2 - 2D_{22} - D_{12}) \\ & \quad -k_{32}[D_{12}] \end{cases}$$

De plus, d'après les observations expérimentales, $k_{11} = k_{21} = \frac{1}{2}k_{31} = k_1$, $k_{12} = k_{22} = k_{32} = k_2$ et $M_1(t_0) = M_2(t_0) = 0$, une nouvelle simplification est possible en posant :

$$\begin{cases} x = [D_{11}] \\ y = [D_{22}] \\ z = [D_{12}] \end{cases} \quad et \quad \begin{cases} x(t_0) &= d_{11} \\ y(t_0) &= d_{22} \\ z(t_0) &= d_{12} \end{cases}$$

Le modèle devient alors :

$$\begin{cases} M_1(t_0) + 2D_{11}(t_0) + D_{12}(t_0) &= \alpha_1 \\ M_2(t_0) + 2D_{22}(t_0) + D_{12}(t_0) &= \alpha_2 \end{cases} \Longrightarrow \begin{cases} \alpha_1 &= 2d_{11} + d_{12} \\ \alpha_2 &= 2d_{22} + d_{12} \end{cases}$$

$$\Longrightarrow \begin{cases} \frac{dx}{dt} &= k_1(2d_{11} + d_{12} - 2x - z)^2 - k_2x \\ \frac{dy}{dt} &= k_1(2d_{22} + d_{12} - 2y - z)^2 - k_2y \\ \frac{dz}{dt} &= 2k_1(2d_{11} + d_{12} - 2x - z)(2d_{22} + d_{12} - 2y - z \\ & \quad -k_2z \end{cases}$$

B.2 Inférence paramétrique- modèle statistique

Une fois le système d'équation construit, celui-ci doit être résolu afin de pouvoir construire un modèle statistique basé sur les résultats de l'EDO. Ce modèle statistique consiste à induire les caractéristiques inconnues de données à partir d'un échantillon, il s'agit donc de faire de l'inférence paramétrique[1]. Les caractéristiques de l'échantillon, une fois connues, reflètent avec une certaine marge d'erreur celles de la population.

[1]L'inférence est une opération logique par laquelle on admet une proposition en vertu de sa liaison avec d'autres propositions déjà tenues pour vraies

B.2.1 Construction du système d'EDO

La réaction entre des produits($mu_{i,j}$) et des réactifs ($mu_{i,j}$) est données par :

$$\sum_{i=1}^{n} \mu_{i,j} X_i \leftrightarrows \sum_{i=1}^{n} \eta_{i,j} X_i$$

Le système d'équation différentielle peut ainsi être simplifier et s'écrire : $\bar{x} = Sv(x)$ où v est un vecteur du flux de réaction et S est une matrice stoechiométrique. Pour cela, le flux de réaction suivant la cinétique d'action de masse est calculé :

$$v_j = k_{+j} \prod_{i=1}^{n} x_i^{\mu_{i,j}} - k_{-j} \prod_{i=1}^{n} x_i^{\eta_{i,j}}$$

où x_i correspond à la concentration de X_i.

Et la matrice stoechiométrique est donnée par :

$$S_{i,j} = \eta_{i,j} - \mu_{i,j}$$

exemple : Réaction d'enzyme

Le système étudié est :

$$\begin{cases} A + E \leftrightarrows AE \\ AE \rightarrow B + E \end{cases} \Longrightarrow \begin{cases} \frac{d[A]}{dt} &= k_{+1}[A][E] - k_{-1}[AE] \\ \frac{d[E]}{dt} &= k_{+1}[A][E] - k_{-1}[AE] + k_2[AE] \\ \frac{d[AE]}{dt} &= -k_{+1}[A][E] + k_{-1}[AE] - k_2[AE] \\ \frac{d[B]}{dt} &= k_2[AE] \end{cases}$$

Par la suite on définit :

$$\begin{cases} x_1 = [A] \\ x_2 = [E] \\ x_3 = [AE] \\ x_4 = [B] \end{cases}$$

Afin de simplifier l'équation, plusieurs étapes sont effectuées :

1. Calcul de la cinétique des réactions qui permet une réécriture du système d'équations différentielles :

$$\begin{cases} v_1 &= k_{+1}x_1x_2 - k_{-1}x_3 \\ v_2 &= k_2x_3 \end{cases} \Longrightarrow \begin{cases} \frac{dx_1}{dt} &= -v_1 \\ \frac{dx_2}{dt} &= -v_1 + v_2 \\ \frac{dx_3}{dt} &= v_1 - v_2 \\ \frac{dx_4}{dt} &= -v_2 \end{cases}$$

$$\Longrightarrow \quad \frac{d}{dt}\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} -1 & 0 \\ -1 & 1 \\ 1 & -1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}$$

$$\Longrightarrow \quad \frac{d}{dt}\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = Sv(x)$$

   Cependant la réécriture du système peut-être plus rapide en calculant directement S.

2. Calcul de S :

$$S = \eta_{i,j} - \mu_{i,j} = \begin{pmatrix} -1 & 0 \\ -1 & 1 \\ 1 & -1 \\ 0 & 1 \end{pmatrix}$$

3. Afin de voir l'évolution temporelle, on utilise R.

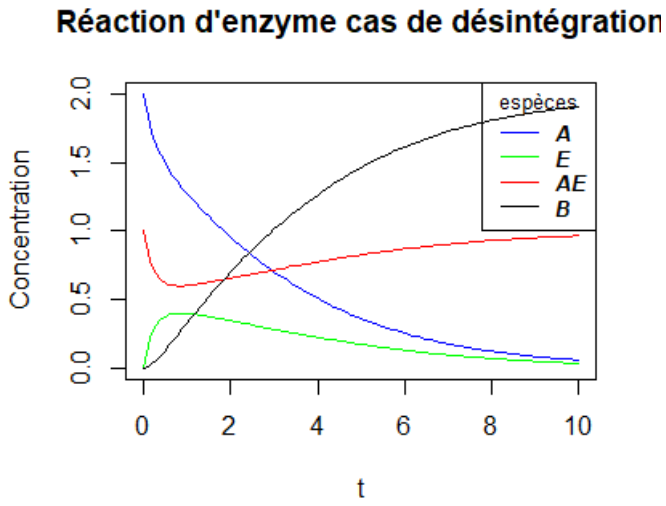

FIGURE B.5 : modélisation concentrations

```
V=function(x,k){
  V=matrix(0,2)
  V[1]=k[1]*x[1]*x[2]-k[1]*x[3]
  V[2]=k[1]*x[3]
  V
}
S=function(t,x,k){
  M=matrix(data = c(-1,0,-1,1,1,-1,0,1),nrow = 4,ncol=2,byrow = T)
  dx=M%*%V(x,k)
  list (dx)
}
out <- ode(func = S, y =c(A=2,E=1,AE=0,B=0), times = (0:100)/10,parms = c(1
plot(out[,1],out[,2],type="l",xlim = c(0,10),ylim = c(0,2),col='blue',main
legend("topright",c("A", "E","AE","B"),
       col=c( "blue","green","red","black"), lty=1, cex=0.8,title="espèces"
lines(out[,1],out[,3],type="l",col='red')
lines(out[,1],out[,4],type="l",col='green')
lines(out[,1],out[,5],type="l",col='black')
```

exemple : Métabolisme en réseau Le système étudié est :

$$\begin{cases} \rightarrow X_1 \\ X_1 + X_3 \rightarrow X_2 + X_4 \\ X_2 \rightarrow \\ X_4 \rightarrow X_3 \end{cases} \Longrightarrow \begin{cases} v_1 = k_1 \\ v_2 = k_2 x_1 x_3 \\ v_3 = k_3 x_2 \\ v_4 = k_4 x_4 \end{cases}$$

$$\Longrightarrow \quad \frac{d}{dt}\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 1 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & -1 & 0 & 1 \\ 0 & 1 & 0 & -1 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{pmatrix}$$

$$\Longrightarrow \quad \frac{d}{dt}\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = Sv(x)$$

Afin de voir l'évolution temporelle, on utilise R (en posant k=1 et X(0)=(0,4,0,4).

Démonstration de l'équation de Michaelis-Menten

L'équation de Michaelis-Menten permet de décrire la cinétique d'une réaction catalysée par une enzyme agissant sur un substrat unique pour donner un produit. Elle relie la vitesse stationnaire initiale de la réaction à la concentration en substrat et à des paramètres caractéristiques de l'enzyme.

L'équation décrit un comportement cinétique très classique, observé avec de nombreuses enzymes. Cependant, elle ne permet pas de prendre compte certains comportements complexes, ni de comprendre les propriétés cinétiques d'enzymes qui catalysent une réaction faisant intervenir plusieurs substrats.

On part de l'hypothèse de mécanisme ci dessous :

$$E + S \underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}} ES \underset{k_{-2}}{\overset{k_2}{\rightleftharpoons}} E + P$$

avec k_1, k_{-1}, k_2, k_{-2} les constantes de vitesse des différentes étapes. On s'intéresse à la vitesse de réaction de P :

$$v = k_2 \times [ES] - k_{-2}[E][P]$$

L'analyse de Michaelis-Menten se fait à l'aide de deux hypothèses simplificatrices :

- Pas de réaction inverse. On se place dans des conditions initiales où il n'y a pas de produit P dans le test enzymatique.

$$E + S \underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}} ES \overset{k_2}{\rightarrow} E + P$$
$$\text{et } v = k_2 \times [ES]$$

- Approximation des états quasi stationnaires. On suppose que la concentration du complexe enzyme-substrat est constante ($\frac{d[ES]}{dt} = 0$) puisque k_1 et k_{-1} sont très rapide par rapport à k_2.

D'où :

$$\begin{aligned}\frac{d[ES]}{dt} = 0 \Rightarrow k_1[E][S] - k_{-1}[ES] - k_2[ES] = 0 \\ \Rightarrow k_1[E][S] = (k_{-1} + k_2)[ES] \\ \Rightarrow [E] = \frac{(k_{-1} + k_2)[ES]}{k_1[S]}\end{aligned}$$

Or d'après la conservation de la quantité totale d'enzyme :

$$[E_T] = [E] + [ES]$$

D'où :

$$\begin{aligned}[E_T] = [E] + [ES] \Rightarrow [E_T] = \frac{(k_{-1} + k_2)[ES]}{k_1[S]} + [ES] \\ \Rightarrow \frac{[E_T]}{[ES]} = \frac{(k_{-1} + k_2)}{k_1[S]} + 1 \\ \Rightarrow [ES] = \frac{[E_T]}{\frac{(k_{-1}+k_2)}{k_1[S]} + 1}\end{aligned}$$

On alors :

$$v = k_2 \frac{[E_T]}{\frac{(k_{-1}+k_2)}{k_1[S]} + 1}$$

Soit $v_{\max} = k_2[E_T]$ et $K_M = \frac{k_2+k_{-1}}{k_1}$, on trouve alors l'équation de Michaelis et Menten :

$$v = \frac{v_{\max}}{1+\frac{K_M}{[S]}} = \frac{v_{\max} \times [S]}{[S]+K_M}$$

B.2.2 Construction du modèle statistique

indépendants ?
Pour construire des modèles prédictifs, les paramètres du modèle θ sont déduits à partir de données expérimentales impliquant ainsi une simulation numérique répétée du modèle.
Le modèle est composé de la solution numérique des équation différentielle dépendant des paramètres ($y_i^{th}(t_j, \theta)$) et de la mesure (provenant des données de mesures corrompues par le bruit). De plus, à chacun des T points temporels t_j différentes mesures $y_{i,j}^{obs}$ peuvent être enregistrées dans les données expérimentales D. Le modèle suivant est alors obtenu dans lequel le bruit est supposé normalement distribué :

$$y_{i,j}^{obs} = y_i^{th}(t_j, \theta) + \epsilon_{i,j}$$
$$\text{où } \epsilon_{i,j} \sim \mathcal{N}(0, \sigma_{i,j}(\theta)) \text{ donc } y_{i,j}^{obs} \sim \mathcal{N}(y_i^{th}(t_j, \theta), \sigma_{i,j}(\theta))$$

[2]

Remarquons que d'autres hypothèses de bruit sont possibles comme des distributions log-normales, la distribution de Laplace.

B.2.3 Les estimateurs

Les paramètres θ sont ensuite ajustés en minimisant les erreurs à l'aide d'un estimateur (maximum de vraisemblance en général). L'estimateur des moindres carrés (pénalisant les résidus de façon quadratique) peut-être utilisé mais dans notre cas, nous utiliserons l'estimateur du maximum de vraisemblance.

Estimateur des moindres carrés

Le résidu r est défini comme la différence entre la valeur prédite et la valeur observée ($r_i = y_i^{obs} - y_i(\theta)$). Ainsi la somme des moindres carrées est définit par :

$$S(\theta) = \sum_{i=1}^{n} (y_i^{obs} - y_i(\theta))^2$$

[2] équation différentielle dont la sensibilité aux paramètres va rendre difficile la résolution par des méthodes numériques explicites

exemple
Soit L'EDO définit par :

$$\frac{dx}{dt} = \theta \Rightarrow x = \theta t$$

soit $x(0) = 0$ la condition initial et y les valeurs observées. Les données expérimentales suivantes sont obtenues :

$$y(1) = 2, y(2) = 4 \text{ et } y(3) = 5$$

1. calcule de $S(\theta)$:

$$\begin{aligned} S(\theta) &= \sum_{i=1}^{3} (y(i) - \theta t)^2 \\ &= (2-\theta)^2 + (4-2\theta)^2 + (5-3\theta)^2 \\ &= 4 - 4\theta + \theta^2 + 16 - 16\theta + 4\theta^2 + 25 - 30\theta + 90\theta^2 \\ &= 14\theta^2 - 50\theta + 45 \end{aligned}$$

2. Calcul du gradient pour en déduire $\hat{\theta}$:

$$\begin{aligned} \frac{dS(\hat{\theta})}{d\theta} = 0 \Rightarrow 28\hat{\theta} - 50 = 0 \\ \Rightarrow \hat{\theta} = \frac{50}{28} = \frac{25}{14} \end{aligned}$$

Estimateur du maximum de vraisemblance

Comme $y_{i,j}^{obs} \sim \mathcal{N}(y_i^{th}(t_j, \theta), \sigma_{i,j}(\theta))$ (en supposant le bruit normalement distribué), l'estimateur du maximum de vraisemblance est utilisés :

$$\theta^{ML} = argmax\_L(\theta) = argmax\_p(D|\theta) \ .$$

1. Calcul de $p(D|\theta)$:

$$
\begin{aligned}
p(D|\theta) &= \prod_{j=1}^{T}\prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma_{i,j}^2(\theta)}} \exp\left(-\frac{1}{2}\left(\frac{y_{i,j}^{obs} - y_i(t_j,\theta)}{\sigma_{i,j}(\theta)}\right)^2\right) \\
&= \prod_{j=1}^{T}\prod_{i=1}^{n} (2\pi\sigma_{i,j}(\theta)^2)^{-1/2} \exp\left(-\frac{1}{2}\left(\frac{y_{i,j}^{obs} - y_i(t_j,\theta)}{\sigma_{i,j}(\theta)}\right)^2\right)
\end{aligned}
$$

2. En passant au log négatif on obtient :

$$
\begin{aligned}
-ln(p(D|\theta)) &= \sum_{j=1}^{T}\sum_{i=1}^{n} -ln((2\pi\sigma_{i,j}^2(\theta))^{-1/2}) + \frac{1}{2}\left(\frac{y_{i,j}^{obs} - y_i(t_j,\theta)}{\sigma_{i,j}(\theta)^2}\right)^2 \\
&= \frac{1}{2}\sum_{j=1}^{T}\sum_{i=1}^{n} ln(2\pi\sigma_{i,j}^2(\theta)) + \left(\frac{y_{i,j}^{obs} - y_i(t_j,\theta)}{\sigma_{i,j}(\theta)}\right)^2 \\
&= J(\theta)
\end{aligned}
$$

exemple

Soit L'EDO définit par :

$$\frac{dx}{dt} = \theta \Rightarrow x = \theta t$$

soit $x(0) = 0$ la condition initial et y les valeurs observées. Les données expérimentales suivantes sont obtenues :

$$y(1) = 1, y(1) = 2, y(1) = 3$$
$$y(2) = 4, y(2) = 2, y(2) = 6$$
$$y(3) = 5, y(3) = 4, y(3) = 6$$

a) Calcul de la variance : $\sigma^2 = \frac{\sum(x-\bar{x})}{n-1}$

$$\sigma_1^2 = \frac{1}{2}((1-2)^2 + (2-2)^2 + (3-2)^2) = 1$$
$$\sigma_2^2 = \frac{1}{2}((4-4)^2 + (2-4)^2 + (6-4)^2) = 4$$
$$\sigma_3^2 = \frac{1}{2}((5-5)^2 + (4-5)^2 + (6-5)^2) = 1$$

b) calcule de $J(\theta)$:

$$\begin{aligned} J(\theta) &= \frac{\sum_{i=1}^{3}(y(i) - \theta t)^2}{\sigma_i^2} \\ &= (1-\theta)^2 + (2-\theta)^2 + (3-\theta)^2 + \frac{(4-2\theta)^2 + (2-2\theta)^2 + (6-2\theta)^2}{4} \\ &+ (5-3\theta)^2 + (4-3\theta)^2 + (6-3\theta)^2 \\ &= 33\theta^2 - 114\theta + 105 \end{aligned}$$

c) Calcul du gradient pour en déduire $\hat{\theta}$:

$$\frac{dJ(\hat{\theta})}{d\theta} = 0 \Rightarrow 66\hat{\theta} - 114 = 0$$
$$\Rightarrow \hat{\theta} = \frac{114}{66} = \frac{57}{33}$$

L'exemple est testé sur R :

```
    #données observés
D=matrix(c(1,2,3,4,2,6,5,4,6),3,3,byrow = T)
x0=0

des=function(t,x,k){
  des=-k
  return(des)
}



#sigma
S=function(D){
  S= matrix(0,1,3)
  for (i in 1:3){
      S[,i]=1/2*((D[i,1]-mean(D[i,]))^2+(D[i,2]-mean(D[i,]))^2+(D[i,3]-mean(
  }
  return(S)
}



#J(theta) en supposant loi normal du bruit

J=function(x,t,s,yobs,f){
  J=matrix(data=0,3,3)
  for (i in 1:3){
    for (j in 1:3)
      J[i,j]=1/2*(log(2*pi*s[,j]^2)+((yobs[i,j]-f(t[j],x0,x))/(s[,j]))^2)
  }
  return(sum(J))
}
#fonction ne dépend plus que du paramétre
J2=function(o){
 a= J(o,t,S(D),D,des)
  return(a)
```

}

3. Calcule du gradient de $J(\theta)$:

$$\begin{aligned}
\frac{dJ(\theta)}{d\theta} &= \frac{1}{2}\sum_{j=1}^{T}\sum_{i=1}^{n}\left[\left(\frac{1}{2\pi\sigma_i^2(\theta)}2\pi\frac{d\sigma_i^2(\theta)}{d\theta}\right)+\frac{1}{(\sigma_i^2(\theta))^2}\left(2.(y_{i,j}^{obs}-y_i(t_j,\theta)).\frac{-dy_i(t_j,\theta)}{d\theta}\sigma_i^2(\theta)\right.\right.\\
&\left.\left.-(y_{i,j}^{obs}-y_i(t_j,\theta))^2\frac{d\sigma_i^2(\theta)}{d\theta}\right)\right]\\
&= \frac{1}{2}\sum_{j=1}^{T}\sum_{i=1}^{n}\left[\left(\frac{1}{\sigma_i^2(\theta)}\frac{d\sigma_i^2(\theta)}{d\theta}\right)-\left(2.(y_{i,j}^{obs}-y_i(t_j,\theta)).\frac{dy_i(t_j,\theta)}{d\theta}\frac{1}{\sigma_i^2(\theta)}\right.\right.\\
&\left.\left.+\frac{1}{(\sigma_i^2(\theta))^2}(y_{i,j}^{obs}-y_i(t_j,\theta))^2\frac{d\sigma_i^2(\theta)}{d\theta}\right)\right]\\
&= \frac{1}{2}\sum_{j=1}^{T}\sum_{i=1}^{n}\left[\frac{1}{\sigma_i^2(\theta)}\frac{d\sigma_i^2(\theta)}{d\theta}-\frac{(y_{i,j}^{obs}-y_i(t_j,\theta))^2}{(\sigma_i^2(\theta))^2}\frac{d\sigma_i^2(\theta)}{d\theta}-2.\frac{(y_{i,j}^{obs}-y_i(t_j,\theta))}{\sigma_i^2(\theta)}.\frac{dy_i(t_j,\theta)}{d\theta}\right]\\
&= \frac{1}{2}\sum_{j=1}^{T}\sum_{i=1}^{n}\left[\frac{1}{\sigma_i^2(\theta)}\left(1-\frac{(y_{i,j}^{obs}-y_i(t_j,\theta))^2}{\sigma_i^2(\theta)}\right)\frac{d\sigma_i^2(\theta)}{d\theta}-\left(2.\frac{(y_{i,j}^{obs}-y_i(t_j,\theta))}{\sigma_i^2(\theta)}\right)\frac{dy_i(t_j,\theta)}{d\theta}\right]
\end{aligned}$$

Attention, plusieurs sources d'erreurs sont possibles :

- données expérimentales non adéquates
- données expérimentales impossible à reproduire exactement
- solution numérique approximatives

Comme le logarithme est une fonction strictement monotone, la minimisation de $-ln(p(D|\theta)) = J(\theta)$ est équivalente à la maximisation de $p(D|\theta)$. Par conséquent, le problème de minimisation suivant est obtenu :

$$\hat{\theta} = argmin\, J(\theta)$$

B.3 Algorithme d'optimisation

(Retour au rapport : 2.La démarche)
Cross-over : exploration globale Mutation : exploration locale Après un nombre suffisant de générations, tous les individus sont proches du minimum global.

L'algorithme génétique est une heuristique, il arrive qu'il ne converge pas du tout, il faut donc faire l'estimation plusieurs fois.

Nous avons divisé le problème d'inférence des paramètres en deux parties : l'optimisation, la simulation/calcul de gradient et la simulation/calcul de la hessienne.
Dans ce problème d'optimisation, on cherche à minimiser la fonction objective $J(\theta)$.
La fonction $J(\theta)$ peut posséder des minima locaux et des points de selle [3] ce qui est parfois problématique. En effet, les algorithmes d'optimisation peuvent rester bloqués dans les minima locaux(donnant une sous-optimisation entre les données expérimentales et la simulation du modèle). De plus, les points de selle affectent la performance des méthodes d'optimisation locales. Cependant, il n'existe pas de méthode d'optimisation unique pour ce cas, souvent un algorithme hybride est plus performant que les algorithmes usuels. Par exemple, les boîtes à outils pour la biologie computationnelle présentent un mélange de méthodes locales et globales incluant derivative-free et gradient-based methods.

Un algorithme peut-être issus de :

méthodes locales Elles construisent une séquence de points qui diminuent successivement les valeurs de $J(\theta)$. Cette procédure est généralement rapides, mais peut être bloquée dans les minima locaux.
La recherche locale consiste à passer d'une solution à une autre solution proche dans l'espace des solutions candidates (l'espace de recherche) jusqu'à ce qu'une solution considérée comme optimale soit trouvée, ou que le temps imparti soit dépassé. Un algorithme de recherche locale part d'une solution candidate et la déplace de façon itérative vers une solution voisine1. Cette méthode est applicable seulement si une notion de voisinage est définie sur l'espace de recherche. Le même problème peut avoir plusieurs définitions différentes de voisinage ; l'optimisation locale avec des voisinages qui limitent les changements à k composantes de la solution est souvent appelée k-optimale.

Chaque solution candidate a plus d'une solution voisine, le choix de celle vers laquelle se déplacer est pris en utilisant seulement l'information sur les solutions voisines de la solution courante, d'où le terme de recherche locale. Quand le choix de la solution voisine est fait uniquement en prenant celle qui maximise le critère.

Le critère d'arrêt de la recherche locale peut être une limite en durée. Une autre possibilité est de s'arrêter quand la meilleure solution trouvée par l'algorithme n'a pas été améliorée

[3] deux points stationnaires où le gradient disparaît ($\nabla J(\hat{\theta}) = 0$)

depuis un nombre donné d'itérations. Les algorithmes de recherche locale sont des algorithmes sous-optimaux puisque la recherche peut s'arrêter alors que la meilleure solution trouvée par l'algorithme n'est pas la meilleure de l'espace de recherche. Cette situation peut se produire même si l'arrêt est provoqué par l'impossibilité d'améliorer la solution courante car la solution optimale peut se trouver loin du voisinage des solutions parcourues par l'algorithme.
A chaque pas, on ne choisit pas forcement la pente la plus descendante mais dans une cercle la valeur la plus petite.

méthodes globales Elles reposent souvent sur une population de points qui sont itérativement affinés augmentant ainsi les chances d'atteindre le minimum global mais plus lentement .L'optimisation globale se distingue de l'optimisation locale en se concentrant sur la recherche du maximum ou du minimum sur toutes les valeurs d'entrée, par opposition à la recherche de minima ou maxima locaux.
A chaque pas, on choisit la pente la plus descendante.

algorithme mutli-start : Cependant, si le point de départ est dans un puits, le minimum local ne sera pas forcement le global. Pour éviter cela, la méthode du multi-start est utilisé, il s'agit de quadriller l'espace et de lancer l'algorithme locale pour chaque case ce qui permet de trouver tous les minimum locaux. Cette méthode a un temps de calcule très élevé.

Un algorithme peut utiliser :

méthodes basées sur les gradients Elles utilisent les dérivées de la fonction $J(\theta)$ et effectuent des optimisations locales répétées qui sont démarrées à des points (aléatoires) dans l'espace des paramètres. Il faut spécifié un nombre maximum d'optimisations locales.
Elles sont applicables lorsque le gradient (et le Hessian) de $J(\theta)$ est continu et peut être calculé avec précision (valable pour la distribution normale et log-normale du bruit mais pas pour la distribution de Laplace par exemple) .

méthodes sans dérivation Elles utilisent principalement des schémas stochastiques utilisant uniquement la fonction $J(\theta)$.

Le tableau des principales méthodes utilisés est construit :

	méthode sans dérivation	méthode basée sur le gradient
local	CMA-ES	Newton line search trust-region
global	Algorithme de génétique particle Swarm	scatter search clustering search

CMA-ES

New parent : weighted average Best direction New generation Quand CMA-ES converge, il est très bon, mais cela n'arrive pas toujours

L'algorithme CMA-ES (Covariance Matrix Adaptation Evolution Strategies) repose sur l'adaptation, au cours des itérations, de la matrice de variance-covariance de la distribution multi-normale utilisée pour la mutation.

La moyenne de la distribution est mise à jour de sorte que la probabilité de solutions candidates réussies est maximisée. La matrice de covariance de la distribution est mise à jour de telle sorte que la probabilité d'étapes de recherche réussies est augmentée. Les deux mises à jour peuvent être interprétées comme une descente de gradient naturelle. En conséquence, le CMA effectue une analyse itérée des principales composantes des étapes de recherche réussies tout en conservant tous les axes principaux.

Un contrôle de taille de pas empêche efficacement la convergence prématurée tout en permettant une convergence rapide vers un optimum.

à chaque étape d'itération, une combinaison pondérée des meilleures solutions candidates de est utilisée pour mettre à jour les paramètres de distribution. La boucle principale se compose de trois parties principales : 1) échantillonnage de nouvelles solutions, 2) réorganisation des solutions échantillonnées en fonction de leur condition physique, 3) mise à jour des variables d'état internes sur la base des échantillons réorganisés.

Line-search methods

Cette méthode diminue successivement les valeurs de $J(\theta)$ en construisant une séquence de points itérativement.

$$\theta_k = \theta_{k-1} + \gamma.s$$

où γ est la taille du pas et s la direction. Les directions sont fixées en premier, puis un γ approprié est déterminé.

Pour la distribution normale et log-normale, on choisit :

$$s = -\nabla J(\theta_{k-1}) \Rightarrow \theta_k = \theta_{k-1} - \gamma.\nabla J(\theta_{k-1})$$

Ainsi, le gradient fournit la pente la plus raide ce qui donne la plus forte diminution de $J(\theta)$ dans un voisinage autour de θ.
Elle est généralement rapide, mais peut être bloquée dans les minima locaux et un problème lié aux petites tailles de pas est possible.

Méthode de Newton

Elle part du même principe :

$$\theta_k = \theta_{k-1} + \gamma.s \Rightarrow Y^{th}(\theta_{k-1} + \gamma.s) = Y^{th}(\theta_{k-1}) + \gamma.s\nabla_\theta Y^{th}(\theta_{k-1}) + o(\gamma.s)$$

Mais cette méthode permet d'éviter le problème lié aux petites tailles de pas (dans ce cas $\gamma = 1$) à l'aide du pas de newton :

$$s_{newt} = -(\nabla^2 J(\theta_{k-1}))^{-1}\nabla J(\theta_{k-1}) \Rightarrow \theta_k = \theta_{k-1} - (\nabla^2 J(\theta_{k-1}))^{-1}\nabla J(\theta_{k-1})$$

Cependant, au voisinage d'un point selle, le pas de Newton peut pointer dans la direction du point selle (attirant ainsi l'optimiseur vers lui). Plusieurs modifications de la méthode de Newton existent comme : la méthode de Gauss-Newton,la méthode de Quasi-Newton et la méthode de Levenberg-Marquardt.

Approximation de Gauss-Newton

Ces algorithmes sont utilisées pour un bruit de mesure normalement distribué et de paramètres connus. Ils ignorent les dérivées partielles de second ordre dans la Hessienne qui est approximé par la matrice d'information de Fisher (FIM) (cf 4.2.2). Le principal avantage de cette approche est que la FIM peut être calculée au même coût qu'un gradient en utilisant l'analyse de sensibilité directe.

Quasi-Newton Approximation

Les méthodes de quasi-Newton sont utilisés pour les problèmes n'étant pas du type des moindres carrés. En utilisant des algorithme comme Broyden-Fletcher Goldfarb-Shannoz (cf . 4.2.4) et l'algorithme de rang symétrique 1 (SR1) (cf. 4.2.5).
Les versions Quasi-Newton sont généralement peu coûteuses à calculer, car elles ne nécessitent que des manipulations algébriques simples du gradient.

optimisation locale multi-start

L'optimisation locale multi-start utilise un algorithme d'optimisation locale qui est démarré à différents points. L'optimisation locale est effectuée en utilisant des algorithmes de moindres carrés (méthodes de Gauss-Newton combinées avec des algorithmes de région de confiance, des méthodes de type (quasi-) Newton). Dans chaque itération, la fonction objectif et son gradient sont calculés. Si la fonction objectif satisfait certains critères, par exemple si la norme du gradient est inférieure à un certain seuil, l'optimisation est arrêtée, sinon le paramètre est mis à jour et la procédure continue.
Les points initiaux sont tirés aléatoirement dans l'espace des paramètres. Le nombre de points de départ est fixé et pour cela on évalue le meilleur nombre en fonction de la meilleur valeur de fonction objective trouvée sur toutes les exécutions (trouver en étudiant les courbes de probabilité de cascade).
Les résultats sont ensuite triés par leur valeur de fonction objective finale. La meilleure des solutions est retenue comme solution globale. Bien sûr, il n'y a aucune garantie qu'un minimum global a été trouvé. Pour augmenter la confiance, on peux exécuter le programme avec plus de points de départ différents.

Approche hiérarchique

L'approche hiérarchique converge fréquemment puisque lorsque la dimension du problème d'optimisation est réduit de quelques paramètres, il y a une amélioration de la convergence.
L'approche hiérarchique produit des valeurs de paramètres de même qualité que la méthode standard, tout en obtenant une meilleure convergence et une réduction du temps de calcul. Elle évite l'augmentation de la dimensionnalité et est applicable à de nombreuse distributions de bruit.

Algorithme à régions de confiance (trust-region methods)

Une longueur de pas maximum est définit puis une direction appropriée est identifié (l'inverse de Line-search methods). Ce qui permet de faire de grands pas (ne se limitent pas aux pas de petites tailles) près des points de selle (ne sont pas attirées par eux) et de toujours produire des directions descendantes. Elles résolvent le sous-problème :

$$\min_s \frac{1}{2} s^T \nabla^2 J(\theta) s + s^T \nabla J(\theta)$$

Malheureusement ce problème est résolu de manière approximative et est difficile à résoudre explicitement.

B.3.1 Gradient

Fournir un gradient est essentiel pour les méthodes basées sur le gradient. Le gradient peut être calculé en dérivant la solution à l'ODE (souvent appelés les sensibilités du modèle) dont il existe approches de calcul.

Différence fini

$$\frac{dJ(\theta)}{d\theta} = \frac{J(\theta + \epsilon) + J(\theta)}{\epsilon}$$

Souvent non fiable en raison d'erreurs numériques.

Calcul direct

Le bruit peut dépendre de plusieurs loi (cf 3.4), on a donc différents valeurs de $J(\theta)$ possible en fonction du bruit.

normale $\frac{dJ(\theta)}{d\theta} = \frac{1}{2} \sum_{j=1}^{T} \sum_{i=1}^{n} \left[\frac{1}{\sigma_i^2(\theta)} \left(1 - \frac{(y_{i,j}^{obs} - y_i(t_j, \theta))^2}{\sigma_i^2(\theta)} \right) \frac{d\sigma_i^2(\theta)}{d\theta} - \left(2. \frac{(y_{i,j}^{obs} - y_i(t_j, \theta))}{\sigma_i^2(\theta)} \right) \frac{dy_i(t_j, \theta)}{d\theta} \right]$

Laplace Comme $\epsilon_{i,j} \sim Laplace(0, b_i(\theta))$ donc $y_{i,j}^{obs} \sim Laplace(y_i^{th}(t_j, \theta), b_i(\theta))$, l'estimateur du maximum de vraisemblance est calculé :

$$\theta^{ML} = argmax L(\theta) = argmax p(D|\theta) \ .$$

1. Calcul de $L(\theta)$:

$$L(\theta) = \prod_{j=1}^{T}\prod_{i=1}^{n} \frac{1}{2b_i(\theta)} \exp\left(\frac{-|y_{i,j}^{obs} - y_i(t_j,\theta)|}{b_i(\theta)}\right)$$

2. En passant au log on obtiens :

$$\begin{aligned} -ln(L(\theta)) &= \prod_{j=1}^{T}\prod_{i=1}^{n}\left(-ln(2b_i(\theta) + \frac{-|y_{i,j}^{obs} - y_i(t_j,\theta)|}{b_i(\theta)}\right) \\ &= -\prod_{j=1}^{T}\prod_{i=1}^{n}\left(ln(2b_i(\theta)) + \frac{|y_{i,j}^{obs} - y_i(t_j,\theta)|}{b_i(\theta)}\right) \\ &= J(\theta) \end{aligned}$$

3. Calcule du gradient de $J(\theta)$:

$$\begin{aligned} \frac{dJ(\theta)}{d\theta} &= -\sum_{j=1}^{T}\sum_{i=1}^{n}\left[\left(\frac{1}{2b_i(\theta)}2\frac{db_i(\theta)}{d\theta}\right) + \frac{1}{b_i^2(\theta)}\left(b_i(\theta)sign(y_{i,j}^{obs} - y_i(t_j,\theta))\frac{-dy_i(t_j,\theta)}{d\theta}\right.\right. \\ &\left.\left. - (|y_{i,j}^{obs} - y_i(t_j,\theta)|)\frac{db_i(\theta)}{d\theta}\right)\right] \\ &= -\sum_{j=1}^{T}\sum_{i=1}^{n}\left[\frac{1}{b_i(\theta)}\frac{db_i(\theta)}{d\theta} - \frac{1}{b_i(\theta)}sign(y_{i,j}^{obs} - y_i(t_j,\theta))\frac{dy_i(t_j,\theta)}{d\theta}\right. \\ &\left. - (|y_{i,j}^{obs} - y_i(t_j,\theta)|)\frac{db_i(\theta)}{d\theta}\frac{1}{b_i^2(\theta)}\right] \\ &= \sum_{j=1}^{T}\sum_{i=1}^{n}\left[-\frac{1}{b_i(\theta)}\left(1 + \frac{(|y_{i,j}^{obs} - y_i(t_j,\theta)|)}{b_i(\theta)}\right)\frac{db_i(\theta)}{d\theta} + \right. \\ &\left.\left(\frac{sign(y_{i,j}^{obs} - y_i(t_j,\theta))}{b_i(\theta)}\right)\frac{dy_i(t_j,\theta)}{d\theta}\right] \end{aligned}$$

D'où :

$$\boxed{\frac{dJ(\theta)}{d\theta} = \sum_{j=1}^{T}\sum_{i=1}^{n}\left[-\frac{1}{b_i(\theta)}\left(1 + \frac{(|y_{i,j}^{obs} - y_i(t_j,\theta)|)}{b_i(\theta)}\right)\frac{db_i(\theta)}{d\theta} + \left(\frac{sign(y_{i,j}^{obs} - y_i(t_j,\theta))}{b_i(\theta)}\right)\frac{dy_i(t_j,\theta)}{d\theta}\right]}$$

Cauchy Comme $\epsilon_{i,j} \sim Cauchy(0, \gamma_i(\theta))$ donc $y_{i,j}^{obs} \sim Cauchy(y_i^{th}(t_j, \theta), \gamma_i(\theta))$, l'estimateur du maximum de vraisemblance est calculé :

1. Calcul de $L(\theta)$:

$$\begin{aligned} L(\theta) &= \prod_{j=1}^{T}\prod_{i=1}^{n} \frac{1}{\pi\gamma_i(\theta)\left(1+\frac{(y_{i,j}^{obs}-y_i(t_j,\theta))^2}{\gamma_i^2(\theta)}\right)} \\ &= \prod_{j=1}^{T}\prod_{i=1}^{n} \frac{1}{\pi\gamma_i(\theta)} \frac{\gamma_i^2(\theta)}{\gamma_i^2(\theta)+(y_{i,j}^{obs}-y_i(t_j,\theta))^2} \\ &= \prod_{j=1}^{T}\prod_{i=1}^{n} \frac{1}{\pi} \frac{\gamma_i(\theta)}{\gamma_i^2(\theta)+(y_{i,j}^{obs}-y_i(t_j,\theta))^2} \end{aligned}$$

2. En passant au log on obtiens :

$$\begin{aligned} -ln(L(\theta)) &= \prod_{j=1}^{T}\prod_{i=1}^{n} \left(-ln(\pi) + ln(\gamma_i(\theta)) - ln(\gamma_i^2(\theta)+(y_{i,j}^{obs}-y_i(t_j,\theta))^2\right) \\ &= J(\theta) \end{aligned}$$

3. Calcule du gradient de $J(\theta)$:

$$\begin{aligned} \frac{dJ(\theta)}{d\theta} &= \sum_{j=1}^{T}\sum_{i=1}^{n} \left[\left(\frac{1}{\gamma_i(\theta)}\frac{d\gamma_i(\theta)}{d\theta}\right) - \frac{1}{\gamma_i^2(\theta)+(y_{i,j}^{obs}-y_i(t_j,\theta))^2}\left(2\gamma_i(\theta)\frac{d\gamma_i(\theta)}{d\theta}\right.\right. \\ &\quad \left.\left. + 2(y_{i,j}^{obs}-y_i(t_j,\theta))\frac{-dy_i(t_j,\theta)}{d\theta}\right)\right] \\ &= \sum_{j=1}^{T}\sum_{i=1}^{n} \left[\left(\frac{1}{\gamma_i(\theta)} - \frac{2\gamma_i(\theta)}{\gamma_i^2(\theta)+(y_{i,j}^{obs}-y_i(t_j,\theta))^2}\right)\frac{d\gamma_i(\theta)}{d\theta} + \right. \\ &\quad \left. 2\frac{(y_{i,j}^{obs}-y_i(t_j,\theta))}{\gamma_i^2(\theta)+(y_{i,j}^{obs}-y_i(t_j,\theta))^2}\frac{dy_i(t_j,\theta)}{d\theta}\right] \end{aligned}$$

d'où :

$$\frac{dJ(\theta)}{d\theta} = \sum_{j=1}^{T}\sum_{i=1}^{n} \left[\left(\frac{1}{\gamma_i(\theta)} - \frac{2\gamma_i(\theta)}{\gamma_i^2(\theta)+(y_{i,j}^{obs}-y_i(t_j,\theta))^2}\right)\frac{d\gamma_i(\theta)}{d\theta} + 2\frac{(y_{i,j}^{obs}-y_i(t_j,\theta))}{\gamma_i^2(\theta)+(y_{i,j}^{obs}-y_i(t_j,\theta))^2}\frac{dy_i(t_j}{d\theta}\right.$$

Analyse de sensibilité directe

Soit notre fonction objective :

$$\begin{aligned} J(\theta) &= \sum_{j=1}^{T}\sum_{i=1}^{n}\left(\frac{y_{i,j}^{obs} - y_i(x(t_j,\theta),\theta)}{\sigma_{i,j}(\theta)}\right)^2 \\ &= \tilde{J}(Y^{th}(x(t_j,\theta)) \end{aligned}$$

On a donc :

$$\begin{aligned} \frac{dJ}{d\theta_k} &= \sum_{j=1}^{T}\sum_{i=1}^{n}\left[\frac{d\tilde{J}}{dY^{th}}\cdot\underbrace{\frac{dy_i(x(t_j,\theta)}{d\theta}}_{=S^y} + \frac{dJ}{d\theta}\right] \\ S^y &= \frac{dy_i(x(t_j,\theta)}{d\theta} \\ &= \frac{dY^{th}}{dx}\cdot\underbrace{\frac{dx(t_j,\theta)}{d\theta}}_{=S^x} + \frac{dY^{th}}{d\theta} \\ \frac{dS^x(t_j,\theta)}{dt} &= \frac{d}{dt}\frac{dx(t_j,\theta)}{d\theta} \\ &= \frac{d}{d\theta}\frac{dx(t_j,\theta)}{dt} \\ &= \frac{f}{dx}\frac{dx(t_j,\theta)}{d\theta} + \frac{df}{d\theta} \\ et\ \frac{dS^x(0,\theta)}{dt} &= \frac{dS_0^x}{dt} \end{aligned}$$

où f et la solution de l'EDO.

Analyse de sensibilité adjointe

Soit notre fonction objective :

$$J(\theta) = \frac{1}{2}\sum_{j=1}^{T}\sum_{i=1}^{n}\left(\frac{y_{i,j}^{obs} - y_i(x(t_j,\theta),\theta)}{\sigma_{i,j}(\theta)}\right)^2$$

On a $t \in (t_{nt}, t_{nt+1})$:

$$\frac{dJ}{d\theta} = \int_0^T p^T\frac{df}{d\theta}dt - p(t_0)^T\frac{dx_0}{d\theta} + \frac{dJ}{d\theta}$$

où :

$$\begin{aligned}
\frac{dp(t)}{dt} &= -\left(\frac{df(x(t),\theta)}{dx}\right)^T .p(t) \\
p(t_j) &= \lim_{t\to t_{j+}} p(t) + \frac{dJ}{dx} \\
&= \lim_{t\to t_{j+}} p(t) + \frac{1}{2}\sum_{i=1}^{n}\frac{1}{\sigma_{i,j}^2}\left(2(y_{i,j}^{obs} - y_i(x(t_j,\theta),\theta))\frac{dy_i(x(t_j,\theta),\theta)^T}{d\theta}\right) \\
&= \lim_{t\to t_{j+}} p(t) + \sum_{i=1}^{n}\frac{1}{\sigma_{i,j}^2}(y_{i,j}^{obs} - y_i(x(t_j,\theta),\theta))\frac{dy_i(x(t_j,\theta),\theta)^T}{d\theta}
\end{aligned}$$

L'approche adjointe alternative a longtemps été jugée plus efficace sur le plan informatique pour le calcul des gradients pour les grands systèmes. En revanche, en biologie des systèmes, elle n'as que de rares applications.

B.3.2 Hessienne

Calcul direct

- **Loi normale** :
  On a :

$$\frac{dJ(\theta)}{d\theta_t} = \frac{1}{2}\sum_{j=1}^{T}\sum_{i=1}^{n}\left[\frac{1}{\sigma_i^2(\theta)}\left(1 - \frac{(y_{i,j}^{obs} - y_i(t_j,\theta))^2}{\sigma_i^2(\theta)}\right)\frac{d\sigma_i^2(\theta)}{d\theta_t} - \left(2.\frac{(y_{i,j}^{obs} - y_i(t_j,\theta))}{\sigma_i^2(\theta)}\right)\frac{dy_i(t_j,\theta)}{d\theta_t}\right]$$

  d'où :

$$
\begin{aligned}
\frac{dJ^2(\theta)}{d\theta_t d\theta_t} &= \frac{1}{2}\sum_{j=1}^{T}\sum_{i=1}^{n}\left[-\frac{1}{\sigma^4(\theta)}\frac{d\sigma_i^2(\theta)}{d\theta_m}\left(1-\frac{(y_{i,j}^{obs}-y_i(t_j,\theta))^2}{\sigma_i^2(\theta)}\right)\frac{d\sigma_i^2(\theta)}{d\theta_t}\right.\\
&+\frac{1}{\sigma_i^2(\theta)}\left(2.\frac{(y_{i,j}^{obs}-y_i(t_j,\theta))}{\sigma_i^2(\theta)}\frac{dy_i(t_j,\theta)}{d\theta_m}+\frac{(y_{i,j}^{obs}-y_i(t_j,\theta))^2}{\sigma_i^4(\theta)}\frac{d\sigma_i^2(\theta)}{d\theta_m}\right)\frac{d\sigma_i^2(\theta)}{d\theta_t}\\
&+\frac{1}{\sigma^2(\theta)}\left(1-\frac{(y_{i,j}^{obs}-y_i(t_j,\theta))^2}{\sigma_i^2(\theta)}\right)\frac{d\sigma_i^2(\theta)}{d\theta_t d\theta_m}\\
&-2.\frac{(y_{i,j}^{obs}-y_i(t_j,\theta))}{-\sigma_i^4(\theta)}\frac{d\sigma_i^2(\theta)}{d\theta_m}\frac{dy_i(t_j,\theta)}{d\theta_t}\\
&-2.\frac{1}{\sigma_i^2(\theta)}\frac{-dy_i(t_j,\theta)}{d\theta_m}\frac{dy_i(t_j,\theta)}{d\theta_t}\\
&-2.\frac{(y_{i,j}^{obs}-y_i(t_j,\theta))}{\sigma_i^2(\theta)}\frac{dy_i(t_j,\theta)}{d\theta_t d\theta_m}\\
&=\frac{1}{2}\sum_{j=1}^{T}\sum_{i=1}^{n}\left[\left(-\frac{1}{\sigma^4(\theta)}\left(1-\frac{(y_{i,j}^{obs}-y_i(t_j,\theta))^2}{\sigma_i^2(\theta)}\right)+\frac{1}{\sigma_i^2(\theta)}\frac{(y_{i,j}^{obs}-y_i(t_j,\theta))^2}{\sigma_i^4(\theta)}\right)\frac{d\sigma_i^2(\theta)}{d\theta_m}\frac{d\sigma_i^2(\theta)}{d\theta_t}\right.\\
&+\frac{1}{\sigma^2(\theta)}\left(1-\frac{(y_{i,j}^{obs}-y_i(t_j,\theta))^2}{\sigma_i^2(\theta)}\right)\frac{d\sigma_i^2(\theta)}{d\theta_t d\theta_m}\\
&+2.\frac{(y_{i,j}^{obs}-y_i(t_j,\theta))}{\sigma_i^4(\theta)}\frac{dy_i(t_j,\theta)}{d\theta_m}\frac{d\sigma_i^2(\theta)}{d\theta_t}\\
&+2.\frac{(y_{i,j}^{obs}-y_i(t_j,\theta))}{\sigma_i^4(\theta)}\frac{d\sigma_i^2(\theta)}{d\theta_m}\frac{dy_i(t_j,\theta)}{d\theta_t}\\
&+2.\frac{1}{\sigma_i^2(\theta)}\frac{dy_i(t_j,\theta)}{d\theta_m}\frac{dy_i(t_j,\theta)}{d\theta_t}\\
&-2.\frac{(y_{i,j}^{obs}-y_i(t_j,\theta))}{\sigma_i^2(\theta)}\frac{dy_i(t_j,\theta)}{d\theta_t d\theta_m}\\
&=\frac{1}{2}\sum_{j=1}^{T}\sum_{i=1}^{n}\left[-\frac{1}{\sigma^4(\theta)}\left(1-2\frac{(y_{i,j}^{obs}-y_i(t_j,\theta))^2}{\sigma_i^2(\theta)}\right)\frac{d\sigma_i^2(\theta)}{d\theta_m}\frac{d\sigma_i^2(\theta)}{d\theta_t}\right.\\
&+\frac{1}{\sigma^2(\theta)}\left(1-\frac{(y_{i,j}^{obs}-y_i(t_j,\theta))^2}{\sigma_i^2(\theta)}\right)\frac{d\sigma_i^2(\theta)}{d\theta_t d\theta_m}\\
&+2.\frac{(y_{i,j}^{obs}-y_i(t_j,\theta))}{\sigma_i^4(\theta)}\left(\frac{dy_i(t_j,\theta)}{d\theta_m}\frac{d\sigma_i^2(\theta)}{d\theta_t}+\frac{d\sigma_i^2(\theta)}{d\theta_m}\frac{dy_i(t_j,\theta)}{d\theta_t}\right)\\
&+2.\frac{1}{\sigma_i^2(\theta)}\frac{dy_i(t_j,\theta)}{d\theta_m}\frac{dy_i(t_j,\theta)}{d\theta_t}\\
&-2.\frac{(y_{i,j}^{obs}-y_i(t_j,\theta))}{\sigma_i^2(\theta)}\frac{dy_i(t_j,\theta)}{d\theta_t d\theta_m}
\end{aligned}
$$

- **Loi Laplace**

$$\frac{dJ^2(\theta)}{d\theta_t d\theta_m} = \sum_{j=1}^{T}\sum_{i=1}^{n}\left[\left(-\frac{1}{b_i(\theta)}\left(1+\frac{(|y_{i,j}^{obs}-y_i(t_j,\theta)|)}{b_i(\theta)}\right)\frac{d^2b_i(\theta)}{d\theta_t d\theta_m} - \frac{1}{b_i(\theta)}\left(1+\frac{(|y_{i,j}^{obs}-y_i(t_j,\theta)|)}{b_i(\theta)}\right)\frac{d}{}\right.\right.$$
$$\left.\left. - \frac{1}{b_i(\theta)}\left(1+\frac{(|y_{i,j}^{obs}-y_i(t_j,\theta)|)}{b_i(\theta)}\right)\frac{db_i(\theta)}{d\theta} + \left(\frac{sign(y_{i,j}^{obs}-y_i(t_j,\theta))}{b_i(\theta)}\right)\frac{dy_i(t_j,\theta)}{d\theta}\right]$$

Fisher information matrix : FIM

La FIM est liée à la covariance des estimations du maximum de vraisemblance et fournit une approximation de la Hessiennne de la fonction du log-vraisemblance négative. Ainsi, la Hessienne est approximé en négligeant les termes qui dépendent de la dérivée de second ordre et de la dérivé de sigma lorsque que le bruit suit une loi normal, ce qui donne la FIM (Fisher Information Matrix)

$$\frac{d^2J}{d\theta_t d\theta_m} \simeq \sum_{j=1}^{T}\sum_{i=1}^{n}\frac{1}{\sigma_{i,j}^2(\theta)}\frac{dy_i(t_j,\theta)}{d\theta_t}\frac{dy_i(t_j,\theta)}{d\theta_m}$$
$$= FIM$$

Analyse de sensibilité adjoint de second ordre

Cette analyse inclue les dérivées de premier ordre et de second ordre des variables d'état. La complexité de calcul du problème évolue de manière quadratique pour le nombre de paramètres et linéairement pour le nombre de variables d'état, ainsi cette méthode ne peut-être utilisé que pour les applications de faible dimension. Cependant, elle possède de meilleures propriétés de mise à l'échelle que les méthodes courantes pour calculer la Hessienne tout en ayant des résultats précis. La boîte à outils AMICI permet une implémentation prête à l'emploi de cette méthode.

algorithme de Broyden-Fletcher Goldfarb-Shannoz

Le schéma BFGS est un algorithme qui calcule séquentiellement une approximation du Hessian au cours d'un processus d'optimisation calculant à chaque étape les gradients. Il s'agit d'un algorithme à direction de descente [4] dont le pas de descente est fonction inverse de la courbure. Si

[4] à partir d'un point de départ, on se déplace sur le plan tangent à la surface, vers le bas.La direction de descente est l'opposé du gradient de la fonction.

la courbure est importante, alors la surface s'écarte vite du plan tangent ainsi il impportant d'avoir un petit pas. L'algorithme calcule donc une approximation de la matrice hessienne (contenant les dérivées partielles d'ordre 2) qui est calculé sur la base des perturbations dans chaque direction de paramètre par un petit pas δ :

$$\frac{dJ^2(\theta)}{d\theta_k d\theta_m} \approx \frac{\frac{dJ(\theta+\delta\epsilon_m)}{d\theta_k} - \frac{dJ(\theta-\delta\epsilon_m)}{d\theta_k d\theta_m}}{2\delta}$$

Le caractère positif assure des directions de descente, mais conduit généralement à de petites tailles au voisinage des points de selle.

algorithme de rang symétrique 1

L'algorithme SR1 permet des approximations négatives et indéfinies ce qui évite les points de selle en permettant des directions de courbure négative.

Approximation séquentielle

L'algorithme calcul une approximation du Hessien H en $\theta(k)$ via une mise à jour continue de $\theta(k-1)$. On définit B comme l'inverse de H, on obtiens :

$$\begin{gathered}\Delta_\theta = \theta(k) - \theta(k-1) \text{ :mise à jour de l'optimiseur} \\ \alpha_k = \nabla_\theta J(\theta(k)) - \nabla_\theta J(\theta(k-1)) \text{ : changement de gradient} \\ \gamma_k = \frac{1}{\alpha_k^T \Delta_\theta} \text{ : normalisation}\end{gathered}$$

D'où :

$$\begin{aligned}H_{k+1} &= (I - \gamma_k \alpha_k \Delta_\theta^T) H_k (I - \gamma_k \alpha_k \Delta_\theta^T) + \gamma_k \alpha_k \alpha_k^T \\ B_{k+1} &= B_k \frac{B_k \alpha_k \alpha_k^T B_k}{\alpha_k^T B_k \alpha_k} + \gamma_k \Delta_\theta \Delta_\theta^T\end{aligned}$$

B.4 Qualité du modèle

B.4.1 Identifiabilité

définition

Un paramètre θ_i est structurellement (ou uniquement) **identifiable globalement** si :

$$\forall \theta' \in \omega, y(t,\theta) = y(t,\theta') \Rightarrow \theta_i = \theta'_i$$

Un paramètre θ_i est structurellement **identifiable localement** si :

$$\forall \theta' \in \omega, \exists \text{ un voisinage } V(\theta')/ \text{ si } \theta \in V(\theta') \text{ et } y(t,\theta) = y(t,\theta') \Rightarrow \theta_i = \theta'_i$$

Un paramètre θ_i est structurellement **non-identifiable** si :

$$\forall \theta' \in \omega, \exists! \text{ un voisinage } V(\theta')/\theta \in V(\theta') \text{ et } y(t,\theta) = y(t,\theta') \Rightarrow \theta_i = \theta'_i$$

Des méthodes d'identification existe mais sont très complexes en calcul direct et peu réalisable dans la pratique en approche basé sur la transformation.

B.4.2 intervalle de confiance

définition

Un **intervalle** (ou région) **de confiance** $(1-\alpha)\%$ est un intervalle calculé à partir des observations qui inclut le paramètre intéressant. Il reflète un niveau de signification de α.

Un paramètre θ_i est pratiquement identifiable à partir d'un ensemble de données D si son intervalle de confiance est fin. La région de confiance basée sur la vraisemblance du paramètres θ pour le niveau de signification de α est :

$$\begin{aligned}
IC^{\alpha} =& \{\theta | \frac{L(\theta)}{L(\theta^{ML})} < exp(-\frac{\Delta_\theta}{2})\} \\
=& \{\theta | ln(\frac{L(\theta)}{L(\theta^{ML})}) < -\frac{\Delta_\theta}{2}\} \\
=& \{\theta | ln(L(\theta)) - ln(L(\theta^{ML})) < -\frac{\Delta_\theta}{2}\} \\
=& \{\theta | -2(ln(L(\theta)) - ln(L(\theta^{ML}))) > \Delta_\theta\} \\
=& \{\theta | 2(J(\theta) - J(\theta^{ML})) > \Delta_\theta\}
\end{aligned}$$

où $\Delta_\theta = \mathcal{X}^2(\alpha, n_\theta)$.

D'après la série de Taylor de log-vraisemblance négative, on a :

$$J(\theta) \approx J(\theta^{ML}) + \tfrac{1}{2}(\theta - \theta^{ML})^T H(\theta - \theta^{ML})$$

Ainsi pour une distribution multi-variée non normalisée avec matrice de covariance $\Sigma = H^{-1}$, comme $2(J(\theta) - J(\theta^{ML}))$, on as :

$$\begin{aligned} IC_i^{\alpha} =& \{\theta | J(\theta) - J(\theta^{ML}) < \frac{\Delta_\theta}{2}\} \\ =& \{\theta | \frac{1}{2}(\theta - \theta^{ML})^T H(\theta - \theta^{ML}) < \frac{\Delta_\theta}{2}\} \\ =& \{\theta | (\theta - \theta^{ML})^T H(\theta - \theta^{ML}) < \Delta_\theta\} \end{aligned}$$

La qualité de l'approximation locale dépend de la taille de l'échantillon, de la propriété d'identification et du système.

B.4.3 valeurs aberrante

Les valeurs aberrantes sont des points de données corrompus par de grandes erreurs. Elles ne peuvent souvent pas être évitée et faussent les estimations des paramètres entraînant des prédictions de modèle incorrecte. Il existe de nombreuses sources potentielles d'erreurs de mesure pendant la collecte et le traitement des données venant des limitations techniques et des erreurs humaines (erreurs de pipetage ou un étiquetage incorrect) entraînant des erreurs potentiellement importantes.
L'utilisation de fonctions objectives qui exploitent la distance au carré, comme la distribution normale, donne un grand poids aux valeurs aberrantes. Les distributions plus lourdes réduisent l'effet des valeurs aberrantes.
Une exclusion manuelle des valeurs aberrantes prend beaucoup de temps et sont difficile à identifier puisqu'il n'y as pas de définition universelle pour les valeurs aberrantes (les points de données extrêmes réellement générés par les mécanismes sous-jacents ne doivent pas être supprimés car ils contiennent des informations importantes). Les points de données qui ont une grande distance par rapport à la trajectoire simulée correspondante peuvent être supprimés selon, par exemple la règle des trois sigmas.

De nombreuses méthodes ont été développées pour la détection et l'élimination des valeurs aberrantes. La plupart des algorithmes attribuent un score au degré d'anomalie ou un label binaire à un point de données. Cet étiquetage est généralement basé sur une mesure de distribution ou de distance (par ex. la distance du plus grand voisin). L'objectif est de savoir si un point de données est suffisamment anormal pour être retiré. Les mesures bruitées, la taille et la complexité des

données rendent l'élimination des valeurs aberrantes difficile.

Les estimateurs de type de maximum de vraisemblance se sont révélés robustes aux valeurs aberrantes (par exemple, l'estimateur d'écart le moins absolu et l'estimateur M de Huber).

Pour la loi normale, s'il y a de fortes valeurs aberrantes, la trajectoire ajustée et la trajectoire vraie diffèrent, ce qui implique des erreurs d'estimation. Cependant, pour la distribution de Laplace, Huber, Cauchy et Student, l'ajustement a donné des trajectoires de systèmes proches de la trajectoire utilisée pour simuler les données. En présence de valeurs aberrantes, les distributions plus lourdes sont préférées au modèle de bruit normal. Ainsi les distributions plus lourdes peuvent être utilisées comme outils de diagnostic pour tester la présence de valeurs aberrantes.

Par ailleurs le temps de calcul est aussi important et pour les distributions de Laplace et de Huber, il est élevé (aucune approximation du Hessien basée sur les sensibilités de premier ordre n'a pu être trouvée) tandis qu'il est faible pour les distributions normale, de Cauchy et de Student.

Pour la distribution normale, la distribution de Laplace et Huber, les IC calculés s'élargissent en présence de de valeurs aberrantes. Alors que, pour la distribution de Cauchy et de Student, les IC deviennent encore plus serrés (contre-intuitif car le contenu d'information dans les données devrait être diminué). Ainsi, la distribution de Laplace et Huber offre le meilleur ratio de couverture en présence de valeurs aberrantes. Il a été démontré que les valeurs aberrantes ont une plus grande influence sur les intervalles de confiance lors de l'utilisation de l'hypothèse de distribution normale.

La performance des estimateurs dépend souvent de la taille de l'échantillon. En général, la distribution de Cauchy et de Student doit être appliquée avec soin si le modèle est trop flexible et si un surapprentissage est à prévoir.

Pour les grands modèles avec de nombreux paramètres et variables d'état, la distribution de Laplace pourrait être avantageuse.

B.5 Selection du meilleur modèle

Comme les données sont ajustées seulement sur une seule expérience, les constantes de vitesse décrivent uniquement cette expérience. Il est donc nécessire de refaire l'étude sur un ensemble

de n expériences, on utilise alors une analyse par clustering. Si les résultats expérimentaux sont cohérents et les simulations de modèles précises, alors les constantes de vitesse calculées doivent être proches dans toutes les expériences. Malgré tout, même avec des données cohérentes, le modèle mathématique peut être incomplet et incapable de prendre en compte tous les phénomènes présents dans les expériences.

Pour choisir le meilleur modèle dans un ensemble de modèles candidats, un critère qui minimise l'erreur quadratique moyenne de la prédiction est utilisé. Un critère populaire est le facteur de Bayes cependant, dans le cas des modèles généraux d'EDO, il n'y a aucune preuve d'efficacité.

Un critère de sélection de modèle est le critère d'information bayésien (BIC) . La valeur BIC pour le modèle M_m est :

$$\begin{aligned} BIC_m &= -2ln(p(D|\hat{\theta}_m)) + ln(|D|)n_{\hat{\theta}_m} \\ &= 2J(\hat{\theta}_m) + ln(|D|)n_{\hat{\theta}_m} \end{aligned}$$

Le critère d'information Akaike (AIC) :

$$\begin{aligned} AIC_m &= -2ln(p(D|\hat{\theta}_m)) + 2n_{\hat{\theta}_m} \\ &= 2J(\hat{\theta}_m) + 2n_{\hat{\theta}_m} \end{aligned}$$

Les valeurs BIC et AIC faibles sont préférables et les différences supérieures à 10 sont supposées être importantes.

Le critère de choix du modèle doit être calculé pour chaque modèle afin de déterminer le modèle optimal. En statistique, une approche souvent utilisée pour réduire le nombre de modèles à tester est la régression par étapes (comprenant la sélection vers l'avant et l'élimination vers l'arrière et les combinaisons des deux).

Une approche alternative consiste à pénaliser le nombre de paramètres dans la fonction objectif.

Pour de grands ensembles de modèles candidats et de données limitées, il arrive fréquemment qu'aucun modèle ne soit choisi par le critère de sélection du modèle. Au contraire, un modèle qui est plausible, ne peut pas être rejeté et devrait être pris en compte dans l'analyse suivante. Dans ce cas, la moyenne du modèle peut être utilisée pour prédire le comportement du processus.

B.6 Biologie

(Retour au rapport)

Tous les organismes vivants sont composés d'une ou plusieurs cellules (cf.B.6). Un corps humains est composé d'environ 50 billions de cellules. Ces cellules se développent à partir d'un seul ovule fécondé. Il existe environ 200 types de cellules différentes. Elles sont organisées en tissus qui sont eux-même organisés en organes. Les cellules elles-mêmes contiennent des organites, qui sont analogues à nos propres organes.

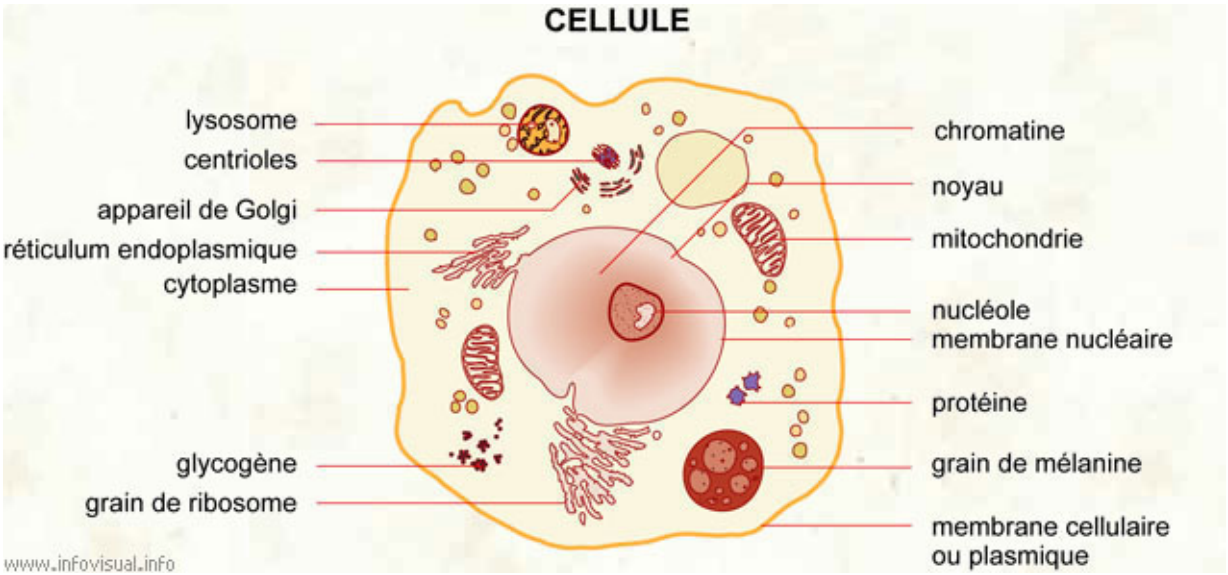

FIGURE B.6 : Schéma cellule

Une cellule est constitué d'un noyau et d'une membrane, son intérieur est un environnement aqueux . Les bactéries, qui sont des organismes unicellulaires, sont procaryotes, tandis que les animaux, les plantes, les champignons et les protistes unicellulaires sont composés de cellules eucaryotes.

Le bon fonctionnement d'un organisme dépend de la capacité de ses cellules à interagir correctement avec leur environnement et entre elles. Par exemple, une réponse immunitaire à des composés nocifs (comme les piqûres d'insectes) est la libération d'histamine, qui interagit ensuite avec les cellules de la zone affectée pour induire des mécanismes protecteurs tels que le gonflement et la contraction musculaire. Le traitement d'une variété de tels signaux par diverses

cellules est ce qui permet le bon fonctionnement des organismes, et les mécanismes sous-jacents peuvent être collectivement appelés la signalisation cellulaire. Comprendre ces mécanismes peut aider à expliquer et à corriger les défaillances du fonctionnement physiologique. Par exemple, chez les personnes allergiques, la libération de l'histamine se produit en réponse à des substances inoffensives.
L'une des façons dont les cellules répondent aux signaux est via des protéines appelées récepteurs, qui agissent comme des capteurs. Ils sont activés par des ligands, molécules qui induisent une réponse en se liant aux récepteurs et en formant un complexe. Les récepteurs qui résident dans la membrane cellulaire sont appelés récepteurs de surface cellulaire ; ceux-ci possèdent une partie extracellulaire à laquelle les ligands peuvent se lier, et une partie à l'intérieur de la membrane où d'autres molécules peuvent interagir. De cette manière, le signal extracellulaire peut être détecté et relayé dans la cellule. Les molécules qui peuvent agir comme ligands comprennent les hormones, les neurotransmetteurs et les médicaments. Les hormones sont des messagers chimiques qui, à partir des organes où elles sont produites, traversent la circulation sanguine vers d'autres organes, où elles peuvent induire une réponse cellulaire en interagissant avec les récepteurs de la surface cellulaire ; Des fonctions physiologiques aussi variées que la croissance, le métabolisme, la digestion et la régulation de la fréquence cardiaque sont déclenchées par les hormones. Les neurotransmetteurs sont des messagers chimiques qui transmettent des impulsions entre les nerfs, ou entre les nerfs et les muscles. De nombreux médicaments existants sont des composés synthétiques, conçus pour induire des réponses similaires à des ligands endogènes ou pour inhiber leur action. Par exemple, les antihistaminiques agissent en bloquant la liaison de l'histamine à son récepteur.
. Dans notre cas d'études, on s'intéresse à une famille de récepteurs de surface cellulaire : les récepteurs couplés aux protéines G (GPCR). Les RCPG sont la plus grande famille de récepteurs membranaires, à travers lesquels la majorité des signaux extracellulaires sont traités. Les GPCR sont la cible d'environ 50% des médicaments utilisés en clinique (car ils assurent la médiation de nombreuses fonctions physiologiques, y compris le métabolisme, la reproduction, le développement, l'homéostasie hormonale et le comportement). Par conséquent, la compréhension des mécanismes sous-jacents de signalisation permettrait l'identification de nouvelles cibles médicamenteuses.

Les protéines G sont ainsi appelées en raison du rôle joué par les nucléotides de la guanine dans la régulation de leur activité. La protéine G est un hétérotrimère, c'est-à-dire qu'elle est composée de trois sous-unités : les unités α, β et γ. Dans l'état inactif de la protéine G, le site de liaison de la sous-unité G_α est occupé par le nucléotide guanosine diphosphate (GDP). Le récepteur peut être activé avec ou sans ligand lié, et lorsque (dans son état actif) il se lie à la

protéine G, le GDP se dissocie et laisse place au guanine nucleotide, le guanosine triphosphate (GTP), pour se lier. Le GTP active la protéine G, provoquant un changement de conformation et provoquant la dissociation de la protéine G du complexe ligand-récepteur, qui peut ensuite se lier à d'autres protéines G inactives.
Les récepteurs couplés aux protéines $GPCR$ sont diverses protéines membranaires qui transfèrent des informations environnementales à l'intérieur de la cellule en générant des signaux intracellulaires.
Notre étude se porte sur la voie activée par la protéine G_q, qui conduit à la libération de calcium à partir des réserves intracellulaires.

les molécules qui apparaissent dans les organismes vivants (biomolécules) sont composées presque exclusivement de six atomes : carbone (C), hydrogène (H), azote (N), oxygène (O), le phosphore (P) et le soufre (S).
Dans la cellule, des liaisons covalentes sont formées et rompues à l'aide de catalyseurs enzymatiques.
Les biomolécules qui sont maintenues ensemble par une ou plusieurs liaisons hydrogène sont associées.

tous utilise la protéine Gq mais AV2 utilise aussi Gs (augmente AMPC mais pas étudié ici). autre couplage beta-arrestine (traduction locale).

basal= étude au repos

B.7 Test de normalité

(Retour au rapport : 5.2.Construction du modèle statistique)

B.8 Résolution équation différentielle ordinaire

Une équation ordinaire tel que $y'(t) = f(t, y(t))$ peut-être :

- **autonome** Si $f(t, y(t)) = g(y(t))$, c'est-à-dire si f ne dépend pas de son premier paramètre.
- **linéaire** Si $\exists A(t) \in \mathcal{C}(\mathbb{R}, \mathcal{M}_d(\mathbb{R}))$ et $B(t) \in \mathcal{C}(\mathbb{R}, \mathbb{R}^d)$ tel que $y'(t) = A(t)y(t) + B(t)$

B.8.1 équation différentielle linéaire

Un équation linéaire peut-être :

1. **homogène** $y'(t) = Ay(t)$ avec $A \in \mathcal{M}_d(\mathbb{R})$
   L'ensemble des solutions est :

$$S_0(x) = \lambda e^{-\int A(x)dx}$$

2. **stationnaire** $y'(t) = Ay(t) + B$ avec $A \in \mathcal{M}_d(\mathbb{R})$ et $B \in \mathbb{R}$
   Pas de méthode universelle, mais des méthodes au cas par cas !

B.8.2 équation différentielle non linéaire : méthode numérique

Les modèles non linéaires présentent un large éventail de comportements et n'admettent généralement pas de solutions explicites bien que les équations différentielles linéaires apparaissent naturellement en perturbant une équation différentielle non linéaire autour d'une de ses solutions. Nous recourrons à la simulation numérique pour étudier le comportement de ces systèmes. Elle est basé sur des procédés de résolution numérique : les méthodes numériques.
Elles permettent une résolution du problème de Cauchy. D'après le théorème de Cauchy-lipschitz :

Soit$f : I \times \mathbb{R}^n \to \mathbb{R}$ continue et localement lipschitz par rapport à sa seconde variable.
Alors le problème de Cauchy-lipschitz on a :

$$\begin{cases} y'(t) &= f(t, y(t)) \\ y(t_0) &= y_0 \end{cases} \tag{B.1}$$

admet une unique solution sur un voisinage de t_0

L'existence et l'unicité d'une solution maximale de (B.1) est ainsi garantit.
Deux méthodes de résolution numérique sont présentés : la méthode d'Euler et la méthode de Runge-Kutta.

Méthode d'Euler

On cherche à calculer numériquement le problème de Cauchy. Pour cela, on va se donner une grille sur un intervalle [t_0,t_0+T], c'est-à-dire des points $0 = t_0 < t_1 < t_2 < ... < t_N = t_0 + T$. On va essayer de calculer une approximation des valeurs de la solution en tous ces points, c'est-à-dire des valeurs $y_n \approx y(t_n)$. Ainsi la méthode d'Euler (ou méthode de la Tangente) consiste à construire une solution approchée affine par morceaux.
Il s'agit de la méthode la plus intuitive. Pour une équation différentielle $y'(t) = f(t, y(t))$ où f

est une fonction suffisamment régulière (nous la supposerons toujours de classe C^2). On cherche à approximer la solution y(t) valant y_0 en t_0. Nous ne connaissons pas y(t), mais sachant que sa dérivée en t_0 vaut f(t_0, y_0), si t augmente d'un petit pas ΔT, le graphe de la solution reste proche de sa tangente. En suivant cette tangente, le point suivant est obtenue :

$$(t_1, y_1) = \Big(t_0 + \Delta T, y_0 + \Delta T f(t_0, y_0)\Big)$$

avec un pas $\Delta T << 1$. De plus, $N\Delta T = T \iff \Delta T = \frac{T}{N+1}$ = h

Ensuite, à partir du point (t_1,y_1) l'itération est renouvelé pour aboutir au point :

$$(t_2, y_2) = \Big(t_1 + \Delta T, y_1 + \Delta T f(t_1, y_1)\Big)$$

et ainsi de suite.

Méthode d'Euler explicite

Sachant que :

$$\begin{cases} y'(t_n) &= f(t_n, y(t_n)) \\ t_{n+1} &= t_n + \Delta T \end{cases}$$

En utilisant la formule de Taylor, on a :

$$\begin{aligned} y(t_{n+1}) = y(t_n) + y'(t_n)\Delta T + o(\Delta T) &\iff \frac{y(t_{n+1}) - y(t_n)}{\Delta T} = y'(t_n) + o(1) \\ &\iff y'(t_n) \approx \frac{y(t_{n+1}) - y(t_n)}{t_{n+1} - t_n} \end{aligned}$$

De même,

$$\begin{aligned} & y(t_{n+1}) = y(t_n) + y'(t_n)\Delta T + \frac{\Delta T^2}{2} y"(t_n) + o(\Delta T^2) \\ \iff & \frac{y(t_{n+1}) - y(t_n)}{\Delta T} = y'(t_n) + \frac{\Delta T}{2} y"(t_n) + o(\Delta T) \end{aligned}$$

Ainsi :

$$\frac{y(t_{n+1}) - y(t_n)}{\Delta T} = f(t_n, y(t_n)) + \underbrace{\frac{\Delta T}{2} y"(t_n) + o(\Delta T)}_{\to 0 \text{ quand } \Delta T \to 0}$$

Or, comme on espère avoir $y_n \approx y(t_n)$, le schéma explicite suivant est proposé :

$$\boxed{y_{n+1} = y_n + \Delta T f(t_n, y_n)}$$

Étant donné un pas de temps ΔT et une suite $(t_n = t_0 + n\Delta T)_{n\in\mathbb{N}}$. Le schéma d'Euler explicite associé à l'équation différentielle ordinaire est donné par la relation de récurrence $y_{n+1} = y_n + \Delta T f(t_n, y_n)$. Une "solution" de ce schéma est un $(N+1)$-uplet $(y_n)_{n\in[\![0,N+1]\!]}$ de $\mathbb{R}^n$.

Méthode d'Euler implicite

Le schéma d'Euler implicite consiste à calculer un pas "après".

$$\begin{aligned} \frac{y(t_{n+1})-y(t_n)}{t_{n+1}-t_n} &= \frac{y(t_n)-y(t_{n+1})}{t_n-t_{n+1}} \\ &= y'(t_{n+1}) + \frac{\Delta T}{2} y"(t_{n+1}) + o(\Delta T) \\ &= f(t_{n+1}, y(t_{n+1})) + \underbrace{\frac{\Delta T}{2} y"(t_{n+1}) + o(\Delta T)}_{\to\ 0 \text{ quand } \Delta T\to 0} \end{aligned}$$

Le schéma implicite suivant est proposé :

$$\boxed{y_{n+1} = y_n + \Delta T f(t_{n+1}, y_{n+1})}$$

Méthode de Runge-Kutta

Les méthodes de Runge-Kutta[5] sont des méthodes plus précises que la méthode d'Euler car elles convergent considérablement plus vite. Ce sont les méthodes les plus employées dans la pratique. Ces méthodes reposent sur le principe de l'itération, c'est-à-dire qu'une première estimation de la solution est utilisée pour calculer une seconde estimation, plus précise, et ainsi de suite.
On a :

$$y(t_{n+1}) = y(t_n) + \int_{t_i}^{t_{i+1}} f(s, y(s))ds$$

Pour calculer une intégrale, une formule de quadrature[6] est utilisée :

$$\int_0^1 \psi(s)ds \simeq \sum_{j=0}^{q} b_j \psi(c_j)$$

où $c_0 < c_1 < ... < c_q$ et $\psi(s) = f(t_n + s\Delta T, y(t_n + s\Delta T))$.
Ainsi :

[5] elle fut développée à la fin du 19eme siècle par les mathématiciens allemands C.Runge et W.Kutta

[6] d'après la formule des trapèzes qui consiste à introduire des points équidistants a_i dans l'intervalle $0 = a_0 < a_1 < ..... < a_{N+1} = 1$ et à remplacer l'intégrale par la somme $I = h/2(f(a_0) + f(a_1)) + h/2(f(a_1) + f(a_2)) + ... + h/2(f(a_N) + f(a_{N+1}))$

$$y(t_{n+1}) \simeq y(t_n) + \Delta T \sum_{j=0}^{q} b_j f(t_{n,j}, y(t_{n,j}))$$

où $t_{n,j} = t_n + c_j \Delta T$.
Ceci suggère une méthode que l'on peut écrire :

$$y(t_{n+1}) = y(t_n) + \Delta T \sum_{j=0}^{q} b_j k_{n,j}$$

où $k_{n,j} \simeq f(t_{n,j}, y(t_{n,j}))$.

Cette méthode semble difficile à résoudre et pour éviter ce cas, on suppose que les $y_{n,j}$ ne dépendent que des points déjà calculés, c'est-à-dire des $y_{n,k}$ pour k<j. Ainsi :

$$y(t_{n,j}) = y(t_n) + \Delta T \sum_{k=0}^{j-1} a_{n,k} f(t_{n,k}, y(t_{n,k}))$$

et les $y_{n,j}$ et $k_{n,j}$ sont calculés de proche en proche.

La méthode est souvent résumée par le tableau des différents poids de quadrature, appelé tableau de Butcher :

c_1	$a_{1,1}$	...	...	...	$a_{1,q}$
.	.				.
.	.				.
.	.				.
c_q	$a_{q,1}$	...	...	...	$a_{q,q}$
	b_1	...	...	...	b_q

où les méthodes d'intégrations approchées correspondent aux lignes. On pose par convention $a_{n,j} = 0$ pour $j \geq 1$.

Méthode de Runge-Kutta d'ordre 1 (q=1)

C'est la méthode d'Euler basée sur la méthode des rectangles. On a :

c_1 =0	0 = $a_{1,1}$
	1 = b_1

Donc :

$$f(t_n, y_n) \text{ comme pente}$$
$$t_{n+1} = t_n + \Delta T$$
$$y_{n+1} = y_n + \Delta T f(t_n, y_n)$$

Méthode de Runge-Kutta d'ordre 2 (q=2)

Cette méthode est basée sur la méthode des points milieu. On a :

$$\begin{array}{c|cc} 0 & 0 & 0 \\ \alpha & \alpha & 0 \\ \hline & 1-\frac{1}{2\alpha} & \frac{1}{2\alpha} \end{array}$$

où $\alpha \in]0,1[$.
Donc :

$$y_{n+1} = y_n + \Delta T\left[(1-\frac{1}{2\alpha})\underbrace{f(t_n,y_n)}_{k_1} + \frac{1}{2\alpha}\underbrace{f(t_n+\alpha\Delta T, y_n+\alpha\Delta T\overbrace{f(t_n,y_n)}^{k_1})}_{k_2}\right]$$

$$\Rightarrow y_{n+1} = y_n + \Delta T\underbrace{(1-\frac{1}{2\alpha})k_1 + \frac{1}{2\alpha}k_2}_{\phi(t,y,\Delta T)}$$

Méthode de Runge-Kutta d'ordre 4 (q=4)

C'est la méthode "classique" basée sur la formule de Simpson. On a :

$$\begin{array}{c|cccc} 0 & 0 & 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ \hline & \frac{1}{6} & \frac{2}{6} & \frac{2}{6} & \frac{1}{6} \end{array}$$

Donc :

$$\boxed{y_{n+1} = y_n + \tfrac{\Delta T}{6}(k_1+2k_2+2k_3+k_4)}$$

avec :

$$\left\{\begin{array}{rcc} k_1 & = & f(t_n,y_n) \\ k_2 & = & f(t_n+\frac{\Delta T}{2}, y_n+\frac{\Delta T}{2}k_1) \\ k_3 & = & f(t_n+\frac{\Delta T}{2}, y_n+\frac{\Delta T}{2}k_2) \\ k_4 & = & f(t_n+\Delta T, y_n+\Delta T k_3) \end{array}\right\}$$

et $\phi(t,y,\Delta T) = \frac{1}{6}(k_1+2k_2+2k_3+k_4)$

L'idée est que la valeur suivante (y_{n+1}) est approchée par la somme de la valeur actuelle (y_n) et du produit de la taille de l'intervalle (ΔT) par la pente estimée. La pente est obtenue par une moyenne pondérée des pentes :
- k_1 est la pente au début de l'intervalle ;

- k_2 est la pente au milieu de l'intervalle, en utilisant la pente k_1 pour calculer la valeur de y au point $t_n + \frac{\Delta T}{2}$ par le biais de la méthode d'Euler ;
- k_3 est de nouveau la pente au milieu de l'intervalle, mais obtenue cette fois en utilisant la pente k_2 pour calculer y ;
- k_4 est la pente à la fin de l'intervalle, avec la valeur de y calculée en utilisant k_3.

Dans la moyenne des quatre pentes, un poids plus grand est donné aux pentes au point milieu.

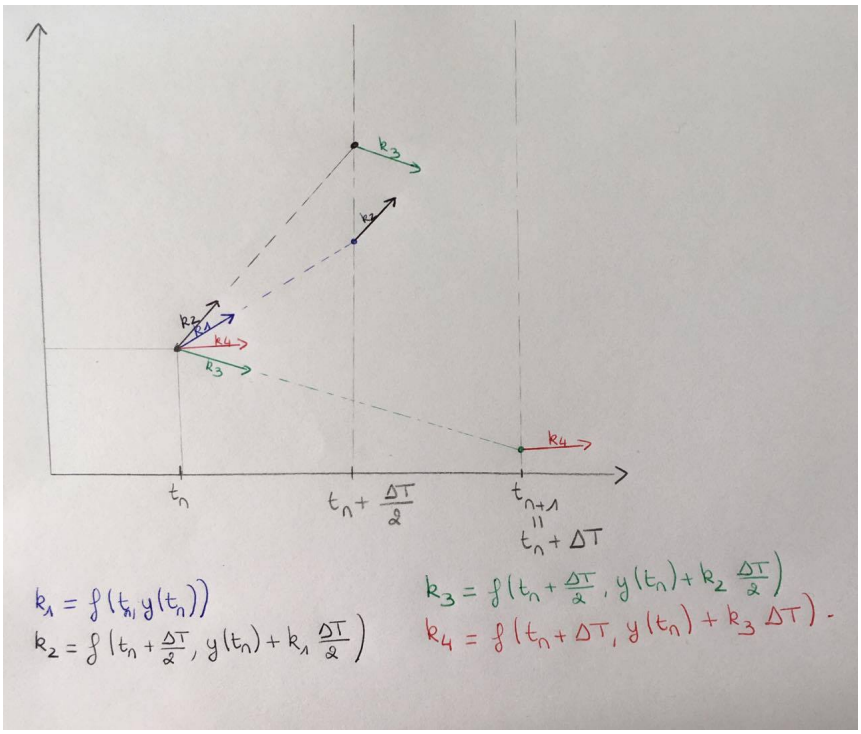

FIGURE B.7 : Schéma méthode de Runge-Kutta à l'ordre 4

Or, les échelles de temps des processus biochimiques s'étendent sur plusieurs ordres de grandeur entraînant la raideur des EDO correspondantes qui dépend du choix des paramètres. Par conséquent, il est toujours conseillé d'utiliser des méthodes numériques pouvant gérer correctement la rigidité, pour l'inférence des paramètres comme la méthode de Runge-Kutta.

Annexe C

Programmation

C.1 Importation des données

C.1.1 données normalisés

Beta-arrestine

```
library(readxl)
                    #OXTR OXT
ba.OO <- data.frame(read_excel("~/stageM1/donn es/Kinetics &
Dose data-OXTR AVR OXT AVP beta arrestine 2 v3.xlsx", skip = 2))
                    #OXTR AVP
ba_oa <- data.frame(read_excel("~/stageM1/donn es/Kinetics &
Dose data-OXTR AVR OXT AVP beta arrestine 2 v3.xlsx", skip = 2,
sheet=2))
                    #AV1A OXT
ba_aao <- data.frame(read_excel("~/stageM1/donn es/Kinetics &
Dose data-OXTR AVR OXT AVP beta arrestine 2 v3.xlsx", skip = 2,
sheet=3))
                    #AV1A AVP
ba_aaa <- data.frame(read_excel("~/stageM1/donn es/Kinetics &
Dose data-OXTR AVR OXT AVP beta arrestine 2 v3.xlsx", skip = 2,
sheet=4))
                    #AV1B OXT
ba_abo <- data.frame(read_excel("~/stageM1/donn es/Kinetics &
Dose data-OXTR AVR OXT AVP beta arrestine 2 v3.xlsx", skip = 2,
```

```
sheet=5))
                    #AV1B AVP
ba_aba <- data.frame(read_excel("~/stageM1/donnees/Kinetics &
Dose data-OXTR AVR OXT AVP beta arrestine 2 v3.xlsx", skip = 2,
sheet=6))
#probleme dans le tableau excel
ba_aba=ba_aba[,-c(6,7,8,9,14,15,20,21,26,27,32,33,38,39,44,45)]
                    #AV2 OXT
ba_a2o <- data.frame(read_excel("~/stageM1/donnees/Kinetics &
Dose data-OXTR AVR OXT AVP beta arrestine 2 v3.xlsx", skip = 2,
sheet=7))
                    #AV2 AVP
ba_a2a <- data.frame(read_excel("~/stageM1/donnees/Kinetics &
Dose data-OXTR AVR OXT AVP beta arrestine 2 v3.xlsx", skip = 2,
sheet=8))
```

Calcium

```
library(readxl)
                    #OXTR OXT
c_oo <- data.frame(read_excel("~/stageM1/donnees/Kinetics &
Dose data-OXTR AVR OXT AVP Calcium v2.xlsx",skip = 2))
                    #OXTR AVP
c_oa <- data.frame(read_excel("~/stageM1/donnees/Kinetics &
Dose data-OXTR AVR OXT AVP Calcium v2.xlsx",sheet=2,skip = 2))
                    #AV1A OXT
c_aao <- data.frame(read_excel("~/stageM1/donnees/Kinetics &
Dose data-OXTR AVR OXT AVP Calcium v2.xlsx",sheet=3,skip = 2))
                    #AV1A AVP
c_aaa <- data.frame(read_excel("~/stageM1/donnees/Kinetics &
Dose data-OXTR AVR OXT AVP Calcium v2.xlsx",sheet=4,skip = 2))
                    #AV1B OXT
c_abo <- data.frame(read_excel("~/stageM1/donnees/Kinetics &
Dose data-OXTR AVR OXT AVP Calcium v2.xlsx",sheet=5,skip = 2))
                    #AV1B AVP
c_aba <- data.frame(read_excel("~/stageM1/donnees/Kinetics &
```

```
Dose data-OXTR AVR OXT AVP Calcium v2.xlsx",sheet=6,skip = 2))
                    #AV2 OXT
c_a2o <- data.frame(read_excel("~/stageM1/donn es/Kinetics &
Dose data-OXTR AVR OXT AVP Calcium v2.xlsx",sheet=7,skip = 2))
                    #AV2 AVP
c_a2a <- data.frame(read_excel("~/stageM1/donn es/Kinetics &
Dose data-OXTR AVR OXT AVP Calcium v2.xlsx",sheet=8,skip = 2))

\end{verbatim}
\subsection{donn es brutes beta-arrestine}
\begin{verbatim}
    #s paration: V=indice venus/ B=indice rlcuc
                #1=exp rience 1/ 2=exp rience 2/r=ratio
separation=function(X){
  X=X[-which(is.na(X)),]
  X[1,2]=-20
  X[2,2]=-10
  X[which(X[,1] == "BASAL")[3],2]=-20
  X[which(X[,1] == "BASAL")[3]+1,2]=-10
  X1=X[,2:(which(is.na(X[1,]))-1)]
  X2=X[,(which(is.na(X[1,]))+1):ncol(X)]
  X1B=X1[-(which(X[,1] == "BASAL")[3]:nrow(X1)),]
  X1V=X1[which(X[,1] == "BASAL")[3]:nrow(X1),]
  X2B=X2[-(which(X[,1] == "BASAL")[3]:nrow(X2)),]
  X2V=X2[which(X[,1] == "BASAL")[3]:nrow(X2),]
  X1r=X1V/X1B
  X1r[,1]=X1V[,1]
  X2r=X2V/X2B
  X2r[,1]=X2V[,1]
  return(list(X1=X1, X2=X2 , B1=X1B, V1=X1V, B2=X2B, V2=X2V, r1=X1r, r2=X2r))
}


                            #importation donn es brut
library(readxl)
                            #OXTR AVP
```

```
O_o<- data.frame(read_excel("~/stageM1/donn es/OXTR DR oxt AVP.xlsx",
col_names = FALSE,col_types = c("text",rep("numeric",35)), skip =4))
OO=separation(O_o)
                              #OXTR AVP
O_A <- data.frame(read_excel("~/stageM1/donn es/OXTR DR oxt AVP.xlsx",
col_names = FALSE,sheet=2,col_types = c("text",rep("numeric",35)), skip =4)
OA=separation(O_A)
                              #AV1A OXT
A1A_o <- data.frame(read_excel("~/stageM1/donn es/V1AR DR oxt AVP.xlsx",
col_names = FALSE,col_types = c("text",rep("numeric",35)),
skip =4))
A1Ao=separation(A1A_o)
                              #AV1A AVP
A1A_A <- data.frame(read_excel("stageM1/donn es/V1AR DR oxt AVP.xlsx",
col_names = FALSE,sheet=2,col_types = c("text",rep("numeric",35)), skip =4)
A1AA=separation(A1A_A)
                              #AV1B OXT
A1B_o <- data.frame(read_excel("~/stageM1/donn es/V1BR DR oxt AVP.xlsx",
col_names = FALSE,col_types = c("text",rep("numeric",35)),
skip =4))
A1Bo=separation(A1B_o)
                              #AV1B AVP
A1B_A <- data.frame(read_excel("~/stageM1/donn es/V1AR DR oxt AVP.xlsx",
col_names = FALSE,sheet=2,col_types = c("text",rep("numeric",35)), skip =4)
A1BA=separation(A1B_A)
                              #AV2 OXT
A2_o <- data.frame(read_excel("~/stageM1/donn es/V2R DR oxt AVP.xlsx",
col_names = FALSE,col_types = c("text",rep("numeric",35)), skip =4))
A2o=separation(A2_o)
                              #AV2 AVP
A2_a <- data.frame(read_excel("~/stageM1/donn es/V2R DR oxt AVP.xlsx",
col_names = FALSE,sheet=2,col_types = c("text",rep("numeric",35)), skip =4)
A2a=separation(A2_a)
```

C.1.2 données brutes calcium

```
library(readxl)
library(tis)
separationc=function(x1,x2){
  #AUCl= aire sous la courbe ligand
  #AUCi= aire sous la courbe reste de calcium
  AUCl=c()
  AUCi=c()
  for (i in 2:ncol(x1)){
    AUCl=c(AUCl,sum(lintegrate(x1[,1], x1[,i],xint=x1[,1])))
    AUCi=c(AUCi,sum(lintegrate(x2[,1], x2[,i],xint=x2[,1])))
  }
  #f= normalisation (sous ligand puis sous sous le reste de calcium)
  xf1=x1
  for (i in 2:ncol(x1)){
    xf1[,i]=x1[,i]/(AUCl[i-1]+AUCi[i-1])
  }
  xf2=x2
  for (i in 2:ncol(x1)){
    xf2[,i]=x2[,i]/(AUCl[i-1]+AUCi[i-1])
  }
  #t= prend en compte ligand et lib ration
  x3=x2
  x3[,1]=23.05+x2[,1]
  xt=rbind(x1,x3)
  #normalisation de t
  xft=xt
  for (i in 2:ncol(x1)){
    xft[,i]=xt[,i]/(AUCl[i-1]+AUCi[i-1])
  }

  return(list(x1=x1,x2=x2,n1=xf1, n2=xf2 , AUCl=AUCl, AUCi=AUCi,t=xt,nt=xft))
}

#x1=donn es  r ponse ligand
#x2=donn es  lib ration calcium
```

```
#n1=normalisation de x1
#n2= normalisation de x2
#AUCl= Aire sous la courbe de x1
#AUCi= Aire sous la courbe de x2
#t= donn es  r ponse ligand+lib ration calcium
#nt=normalisation de t


                              #OXTR OXT
#1=donn es sous ligand
o_o1<- data.frame(read_excel("~/stageM1/donn es/CALCIUM KINETICS.XLSX",
col_names =c("times",rep("-4",4),rep("-5",4),rep("-6",4),rep("-7",4),
rep("-8",4),rep("-9",4),rep("-10",4),rep("tampon",4)),
col_types = c(rep("numeric",33)),skip = 2))
# 2 = lib ration du reste du calcium
o_o2<- data.frame(read_excel("~/stageM1/donn es/CALCIUM KINETICS.XLSX",
sheet=2, col_names =c("times",rep("-4",4),rep("-5",4),rep("-6",4),
rep("-7",4),rep("-8",4),rep("-9",4),rep("-10",4),rep("tampon",4)),
col_types = c(rep("numeric",33)),skip = 2))
oo=separationc(o_o1,o_o2)


                              #OXTR AVP
#-5 que 3 replicats pour o_a1 donc on double un replicat
o_a1_1<- data.frame(read_excel("~/stageM1/donn es/CALCIUM KINETICS.XLSX",
sheet=3, col_names =c("times",rep("-4",4),rep("-5",3),rep("-6",4),
rep("-7",4),rep("-8",4),rep("-9",4),rep("-10",4),rep("tampon",4)),
col_types = c(rep("numeric",32)),skip = 2))
o_a1=cbind(o_a1_1[,c(1,2,3,4,5,6,7,8)],o_a1_1[,8],o_a1_1[,c(9:32)])
colnames(o_a1)=c("times",rep("-4",4),rep("-5",4),rep("-6",4),rep("-7",4),
rep("-8",4),rep("-9",4),rep("-10",4),rep("tampon",4))
o_a2<- data.frame(read_excel("~/stageM1/donn es/CALCIUM KINETICS.XLSX",
sheet=4, col_names =c("times",rep("-4",4),rep("-5",4),rep("-6",4),
rep("-7",4),rep("-8",4),rep("-9",4),rep("-10",4),rep("tampon",4)),
col_types = c(rep("numeric",33)),skip = 2))
colnames(o_a2)=c("times",rep("-4",4),rep("-5",4),rep("-6",4),
```

```
rep("-7",4),rep("-8",4),rep("-9",4),rep("-10",4),rep("tampon",4))
oa=separationc(o_a1,o_a2)

                              #AV1A OXT
#que 3 r plicats pour tous (sauf  tampon)
a1a_o1<- data.frame(read_excel("~/stageM1/donn es/CALCIUM KINETICS.XLSX",
sheet=5, col_names =c("times",rep("-4",3),rep("-5",3),rep("-6",3),
rep("-7",3),rep("-8",3),rep("-9",3),rep("-10",3),rep("tampon",4)),
col_types = c(rep("numeric",26)),skip = 2))
a1a_o2<- data.frame(read_excel("~/stageM1/donn es/CALCIUM KINETICS.XLSX",
sheet=6, col_names =c("times",rep("-4",3),rep("-5",3),rep("-6",3),rep("-7",3),
rep("-8",3),rep("-9",3),rep("-10",3),rep("tampon",4)),
col_types = c(rep("numeric",26)),skip = 2))
a1ao=separationc(a1a_o1,a1a_o2)
                              #AV1A AVP
a1a_a1<- data.frame(read_excel("~/stageM1/donn es/CALCIUM KINETICS.XLSX",
sheet=7, col_names =c("times",rep("-4",4),rep("-5",4),rep("-6",4),rep("-7",4),
rep("-8",4),rep("-9",4),rep("-10",4),rep("tampon",4)),
col_types = c(rep("numeric",33)),skip = 2))
a1a_a2<- data.frame(read_excel("~/stageM1/donn es/CALCIUM KINETICS.XLSX",
sheet=8, col_names =c("times",rep("-4",4),rep("-5",4),rep("-6",4),rep("-7",4),
rep("-8",4),rep("-9",4),rep("-10",4),rep("tampon",4)),
col_types = c(rep("numeric",33)),skip = 2))
a1aa=separationc(a1a_a1,a1a_a2)
                              #AV1B OXT
a1b_o1<- data.frame(read_excel("~/stageM1/donn es/CALCIUM KINETICS.XLSX",
sheet=9, col_names =c("times",rep("-4",4),rep("-5",4),rep("-6",4),rep("-7",4),
rep("-8",4),rep("-9",4),rep("-10",4),rep("tampon",4)),
col_types = c(rep("numeric",33)),skip = 2))
a1b_o2<- data.frame(read_excel("~/stageM1/donn es/CALCIUM KINETICS.XLSX",
sheet=10, col_names =c("times",rep("-4",4),rep("-5",4),rep("-6",4),rep("-7",4),
rep("-8",4),rep("-9",4),rep("-10",4),rep("tampon",4)),
col_types = c(rep("numeric",33)),skip = 2))
a1bo=separationc(a1b_o1,a1b_o2)
                              #AV1B AVP
a1b_a1<- data.frame(read_excel("~/stageM1/donn es/CALCIUM KINETICS.XLSX",
```

```
sheet=11, col_names =c("times",rep("-4",4),rep("-5",4),rep("-6",4),rep("-7"
rep("-8",4),rep("-9",4),rep("-10",4),rep("tampon",4)),
col_types = c(rep("numeric",33)),skip = 2))
a1b_a2<- data.frame(read_excel("~/stageM1/donn es/CALCIUM KINETICS.XLSX",
sheet=12, col_names =c("times",rep("-4",4),rep("-5",4),rep("-6",4),rep("-7"
rep("-8",4),rep("-9",4),rep("-10",4),rep("tampon",4)),
col_types = c(rep("numeric",33)),skip = 2))
a1ba=separationc(a1b_a1,a1b_a2)
                              #AV2 OXT
a2_o1<- data.frame(read_excel("~/stageM1/donn es/CALCIUM KINETICS.XLSX",
sheet=13, col_names =c("times",rep("-4",4),rep("-5",4),rep("-6",4),
rep("-7",4),rep("-8",4),rep("-9",4),rep("-10",4),rep("tampon",4)),
col_types = c(rep("numeric",33)),skip = 2))
a2_o2<- data.frame(read_excel("~/stageM1/donn es/CALCIUM KINETICS.XLSX",
sheet=14, col_names =c("times",rep("-4",4),rep("-5",4),rep("-6",4),
rep("-7",4),rep("-8",4),rep("-9",4),rep("-10",4),rep("tampon",4)),
col_types = c(rep("numeric",33)),skip = 2))
a2_o2=a2_o2[-(which(is.na(a2_o2[,1]==TRUE))),]
a2o=separationc(a2_o1,a2_o2)
                              #AV2 AVP
a2_a1<- data.frame(read_excel("~/stageM1/donn es/CALCIUM KINETICS.XLSX",
sheet=15, col_names =c("times",rep("-4",4),rep("-5",4),rep("-6",4),
rep("-7",4),rep("-8",4),rep("-9",4),rep("-10",4),rep("tampon",4)),
col_types = c(rep("numeric",33)),skip = 2))
a2_a2<- data.frame(read_excel("~/stageM1/donn es/CALCIUM KINETICS.XLSX",
sheet=16, col_names =c("times",rep("-4",4),rep("-5",4),rep("-6",4),
rep("-7",4),rep("-8",4),rep("-9",4),rep("-10",4),rep("tampon",4)),
col_types = c(rep("numeric",33)),skip = 2))
a2a=separationc(a2_a1,a2_a2)
```

C.2 graphique

C.2.1 dose réponse

(Retour au rapport : 4) Les graphiques sont mis dans un autre annexe (cf.Annexe D.1 page 155)

```
#fonction qui renvoie le graphique de dose reponse de
#beta-arrestine pour un temps t
cdrb=function(X,t,i){
  n=c("blue","orange","green")
  R=matrix(0,nrow = 7,ncol=6)
  R[,1]=c(1e-05,1e-06,1e-07,1e-08,1e-09,1e-10,1e-11)
  R[,1]=log10(R[,1])
  R[,2]=c(X[t,2],X[t,6],X[t,10],X[t,14],X[t,18],X[t,22],X[t,28])
  plot(R[,1],R[,2],xlab="dose",ylim = c(-0.1,0.2),ylab ="ratio",
  main = i,sub="Beta-arrextine",cex.sub = 1, col.sub = "blue",
  col.main = "blue")
 legend("bottomright",c("r plicat 1", "r plicat 2",
 "r plicat 3","r plicat 4","moyenne"),x.intersp=0,y.intersp=0,
 box.lty=0,fill=c("black", "blue","orange","green" ,"red"),
 horiz=TRUE, cex=0.5,pt.cex=0.3,text.width      = 0.4)
  for (i in 1:3){
    R[,i+2]=c(X[t,2+i],X[t,6+i],X[t,10+i],X[t,14+i],X[t,18+i],
            X[t,20+i],X[t,28+i])
    points(R[,1],R[,i+2],col=n[i])
  }
  for (i in 1:7){
    R[i,6]=mean(R[i,-c(1,6)],na.rm=TRUE)
  }
  lines(R[,1],R[,6],col="red")
}
#fonction qui renvoie le graphique de dose reponse du calcium
#pour un temps t
cdrc=function(X,t,i){
  n=c("blue","orange","green")
  R=matrix(0,nrow = 7,ncol=6)
  R[,1]=c(1e-04,1e-05,1e-06,1e-07,1e-08,1e-09,1e-10)
  R[,1]=log10(R[,1])
  R[,2]=c(X[t,2],X[t,6],X[t,10],X[t,14],X[t,18],X[t,22],X[t,28])
  plot(R[,1],R[,2],xlab="dose",ylim = c(-0.1,0.2),
  ylab = "concentration",main = i,sub="Calcium",cex.sub = 1,
  col.sub = "blue",col.main = "blue")
```

```
  legend("bottomright",c("r plicat 1", "r plicat 2",
  "r plicat 3","r plicat 4","moyenne"),x.intersp=0,y.intersp=0,
  box.lty=0,fill=c("black", "blue","orange","green" ,"red") ,
  horiz=TRUE, cex=0.5,pt.cex=0.3,text.width     = 0.4)
  for (i in 1:3){
    R[,i+2]=c(X[t,2+i],X[t,6+i],X[t,10+i],X[t,14+i],X[t,18+i],
            X[t,20+i],X[t,28+i])
    points(R[,1],R[,i+2],col=n[i])
  }
  for (i in 1:7){
    R[i,6]=mean(R[i,-c(1,6)],na.rm=TRUE)
  }
  lines(R[,1],R[,6],col="red")
}


#Il faut IMPORTER les donn es normalis  avant

par(mfrow = c(2,2))
                                    #OXT/OXTR

cdrb(ba.OO,100,"OXT/OXTR")
cdrc(c_oo,100,"OXT/OXTR")

                                    #AVP/OXTR
cdrb(ba_oa,100,"AVP/OXTR")
cdrc(c_oa,100,"AVP/OXTR")

                                      #OXT/AV1A

cdrb(ba_aao,100,"OXT/AV1A")
cdrc(c_aao,100,"OXT/AV1A")

                                    #AVP/AV1A
cdrb(ba_aaa,100,"AVP/AV1A")
cdrc(c_aaa,100,"AVP/AV1A")
```

```
73
74                                    #OXT/AV1B
75 cdrb(ba_abo,100,"OXT/AV1B")
76 cdrc(c_abo,100,"OXT/AV1B")
77
78                                    #AVP/AV1B
79 cdrb(ba_aba,100,"AVP/AV1B")
80 cdrc(c_aba,100,"AVP/AV1B")
81
82                                      #OXT/AV2
83 cdrb(ba_a2o,100,"OXT/AV2")
84 cdrc(c_a2o,100,"OXT/AV2")
85
86                                      #AVP/AV2
87 cdrb(ba_a2a,100,"AVP/AV2")
88 cdrc(c_a2a,100,"AVP/AV2")
```

C.2.2 courbe cinétique

Les graphiques sont mis dans un autre annexe (cf.Annexe D.2 page 158).Retour au rapport : (cf.Annexe 4 page 20)

beta-arrestine

```
1  #fonction pour tracer le graphique
2  ccb=function(X,q){
3    c=c("black","blue","blue","orange","orange")
4    a=seq(2,30,by=4)
5    n=c("10-5M","10-6M","10-7M","10-8M","10-9M","10-10M","10-11M",
6    "basal")
7    for (i in 1:8){
8    plot(X[,1],X[,a[i]],type = "l",xlab = "min",ylab = "",
9    ylim = c(-0.05,0.3))
10   title(n[i])
11   for (j in (a[i]+1):(a[i]+3)){
12     lines(X[,1],X[,j],col=c[j-a[i]])
```

```
13    }
14    }
15    par(xpd=NA)
16    legend(-40,0.4,c("Test 1","Test 2"),fill = c("black","blue"),
17    cex=0.9,x.intersp=0,y.intersp=0, horiz=TRUE,
18    text.width= 0.1,xjust = 0.5)
19    title( main=q,outer = T,col.main="blue",line = -1)
20  }
21  #variante pour donnee contenant des valeur manquante
22  ccb2=function(X,q){
23    c=c("black","blue","blue","orange","orange")
24    a=seq(2,30,by=4)
25    n=c("10-5M","10-6M","10-7M","10-8M","10-9M","10-10M","10-11M",
26    "basal")
27    for (i in 1:8){
28      plot(X[,1],X[,a[i]],type = "l",xlab = "min",ylab = "",
29      ylim = c(-0.05,0.3))
30      title(n[i])
31      lines(X[,1],X[,a[i]+1],col=c[1])
32      points(X[,1],X[,a[i]+2],col=c[2])
33      points(X[,1],X[,a[i]+3],col=c[3])
34    }
35    par(xpd=NA)
36    legend(-40,0.4,c("Test 1","Test 2"),fill = c("black","blue"),
37    cex=0.9,x.intersp=0,y.intersp=0, horiz=TRUE,text.width= 0.1,
38    xjust = 0.5)
39    title( main=q,outer = T,col.main="blue",line = -1)
40  }
41  ccb3=function(X,q){
42    c=c("black","blue","blue","orange","orange")
43    a=seq(2,30,by=4)
44    n=c("10-5M","10-6M","10-7M","10-8M","10-9M","10-10M","10-11M",
45    "basal")
46    plot(X[,1],X[,a[1]],xlab = "min",ylab = "",
47    ylim =c(-0.05,0.3))
48    title(n[1])
```

```
  points(X[,1],X[,a[1]+1],col=c[1])
  for (j in (a[1]+2):(a[1]+3)){
    lines(X[,1],X[,j],col=c[2])
  }
  for (i in 2:8){
    plot(X[,1],X[,a[i]],type = "l",xlab = "min",ylab = "",
    ylim = c(-0.05,0.3))
    title(n[i])
    for (j in (a[i]+1):(a[i]+3)){
      lines(X[,1],X[,j],col=c[j-a[i]])
    }
  }
  par(xpd=NA)
  legend(-40,0.4,c("Test 1","Test 2"),fill = c("black","blue"),
  cex=0.9,x.intersp=0,y.intersp=0, horiz=TRUE,text.width= 0.1,
  xjust = 0.5)
  title( main=q,outer = T,col.main="blue",line = -1)
}

par(mfrow = c(2,4))

                              #OXTR OXT et OXTR AVP

ccb(ba.OO,"Beta-arrestine OXT/OXTR")
ccb(ba_oa,"Beta-arrestine AVP/OXTR")

                              #AV1A OXT et AVP
ccb(ba_aao,"Beta-arrestine OXT/AV1A")
ccb(ba_aaa,"Beta-arrestine AVP/AV1A")

                              #AV1B OXT et AVP
ccb(ba_abo,"Beta-arrestine OXT/AV1B")
ccb3(ba_aba,"Beta-arrestine AVP/AV1B")

                                #AV2 OXT et AVP
ccb2(ba_a2o,"Beta-arrestine OXT/AV2")
```

```
85 ccb2(ba_a2a,"Beta-arrestine AVP/AV2")
```

Calcium

```
1  #fonction pour tracer le graphique
2  ccc=function(X,q){
3    a=seq(2,30,by=4)
4    c=c("black","blue","blue")
5    n=c("10-4M","10-5M","10-6M","10-7M","10-8M","10-9M","10-10M",
6    "basal")
7    for (i in 1:8){
8      plot(X[,1],X[,a[i]],type = "l",xlab = "min",ylab = "",
9      ylim = c(-0.02,0.4))
10     title(n[i])
11     for (j in (a[i]+1):(a[i]+3)){
12       lines(X[,1],X[,j],col=c[j-a[i]])
13     }
14   }
15   par(xpd=NA)
16   legend(-35,0.55,c("Test 1","Test 2"),fill = c("black","blue"),
17   cex=0.9,x.intersp=0,y.intersp=0, horiz=TRUE,text.width= 0.1,
18   xjust = 0.5)
19   title( main=q,outer = T,col.main="blue",line = -1)
20 }
21 #variante pour donnees contenant des valeurs manquantes
22 ccc2=function(X,q){
23   a=seq(2,30,by=4)
24   c=c("black","blue","blue")
25   n=c("10-4M","10-5M","10-6M","10-7M","10-8M","10-9M","10-10M",
26   "basal")
27   for (i in 1:8){
28     plot(X[,1],X[,a[i]],type = "l",xlab = "min",ylab = "",
29     ylim = c(-0.02,0.4))
30     title(n[i])
31     for (j in (a[i]+1):(a[i]+2)){
32       lines(X[,1],X[,j],col=c[j-a[i]])
```

```
    }
  }
  par(xpd=NA)
  legend(-35,0.55,c("Test 1","Test 2"),fill = c("black","blue"),
  cex=0.9,x.intersp=0,y.intersp=0, horiz=TRUE,text.width= 0.1,
  xjust = 0.5)
  title( main=q,outer = T,col.main="blue",line = -1)
}
par(mfrow = c(2,4))

#courbecine_ca_OXT_AV1A
                                    #OXTR
ccc(c_oo,"Calcium OXT/OXTR")
ccc(c_oa,"Calcium AVP/OXTR")


                                    #AV1A
ccc2(c_aao,"Calcium OXT/AV1A") #prob
ccc(c_aaa,"Calcium AVP/AV1A")

                                    #AV1B
ccc(c_abo,"Calcium OXT/AV1B")
ccc(c_aba,"Calcium AVP/AV1B")

                                    #AV2
ccc(c_a2o,"Calcium OXT/AV2")
ccc(c_a2a,"Calcium AVP/AV2")
```

C.2.3 Graphique données brutes

beta-arrestine

Graphiques : (cf.Annexe D.3 page 165). Retour au rapport : (cf.Annexe A.1.1 page 33)

```
#Calcul valeur moyenne pour 2 replicat (utile pour les fonctions)
moy=function(X){
  m=matrix(0,ncol=ncol(X)/2+1,nrow = nrow(X))
```

```
  m[,1]=X[,1]
  a=seq(0,100,by=1)
  for (j in 2:(ncol(X)/2+1)){
    for (i in 1:nrow(X)){
      m[i,j]=(X[i,(j+a[j-1])]+X[i,(j+a[j])])/2
    }
  }
  return(m)
}

#fonction pour tracer le ratio
graph.ratio=function(X,q){
  n=c("10-5M","10-6M","10-7M","10-8M","10-9M","10-10M","10-11M",
  "basal")
  a=seq(2,18,by=2)
  M=moy(X[["r1"]])
  M2=moy(X[["r2"]])
  for (i in 1:8){
    plot(X[["r1"]][,1],X[["r1"]][,a[i]],type = "l",xlab = "min",
    ylab = "",ylim = c(0.7,1.3))
    title(n[i])
    lines(X[["r1"]][,1],X[["r1"]][,a[i]+1],col="black")
    lines(M[,1],M[,i+1],col="red")
    lines(X[["r2"]][,1],X[["r2"]][,a[i]],col="blue")
    lines(X[["r2"]][,1],X[["r2"]][,a[i]+1],col="blue")
    lines(M2[,1],M2[,i+1],col="green")
  }
  par(xpd=NA)
  legend(-3400,1.6,c("Test 1","Test 2","Moyenne 1","Moyenne 2"),
 fill = c("black","blue","red","green"),ncol=2,cex=0.8)
  title( main=q,outer = T,col.main="blue",line = -1)

}
#fonction pour tracer les donnees Venus
graph.venus=function(X,q){
  n=c("10-5M","10-6M","10-7M","10-8M","10-9M","10-10M","10-11M",
```

```
  "basal")
  a=seq(2,17,by=2)
  M=moy(X[["V1"]])
  M2=moy(X[["V2"]])
  for (i in 1:8){
    plot(X[["V1"]][,1],X[["V1"]][,a[i]],type = "l",xlab = "min",
    ylab = "",col="black",ylim = c(0,1000000))
    title(n[i])
    lines(X[["V1"]][,1],X[["V1"]][,a[i]+1],col="black")
    lines(M[,1],M[,i+1],col="red")
    lines(X[["V2"]][,1],X[["V2"]][,a[i]],col="blue")
    lines(X[["V2"]][,1],X[["V2"]][,a[i]+1],col="blue")
    lines(M2[,1],M2[,i+1],col="green")
  }
  par(xpd=NA)
  legend(-5150,2000000,c("Test 1","Test 2","Moyenne 1",
  "Moyenne 2"), fill = c("black","blue","red","green"),
  ncol=2,cex=0.9)
  title( main=q,outer = T,col.main="blue",line = -1)
}
#fonction pour tracer les donnees Rluc
graph.rluc=function(X,q){
  n=c("10-5M","10-6M","10-7M","10-8M","10-9M","10-10M","10-11M",
  "basal")
  a=seq(2,17,by=2)
  M=moy(X[["B1"]])
  M2=moy(X[["B2"]])
  for (i in 1:8){
    plot(X[["B1"]][,1],X[["B1"]][,a[i]],type = "l",xlab = "min",
    ylab = "",col="black",ylim = c(10000,1000000))
    title(n[i])
    lines(X[["B1"]][,1],X[["B1"]][,a[i]+1],col="black")
    lines(M[,1],M[,i+1],col="red")
    lines(X[["B2"]][,1],X[["B2"]][,a[i]],col="blue")
    lines(X[["B2"]][,1],X[["B2"]][,a[i]+1],col="blue")
    lines(M2[,1],M2[,i+1],col="green")
```

```
76    }
77    par(xpd=NA)
78    legend(-5150,2000000,c("Test 1","Test 2","Moyenne 1",
79    "Moyenne 2"),fill = c("black","blue","red","green"),ncol=2,
80    cex=0.9)
81    title( main=q,outer = T,col.main="blue",line = -1)
82  }
83
84  par(mfrow = c(2,4))
85
86
87                #OXTR
88  #OXT
89  graph.ratio(OO,"Courbe cin tique du ratio du couple OXT/OXTR")
90  graph.venus(OO,"Courbe cin tique de Venus du couple OXT/OXTR")
91  graph.rluc(OO,"Courbe cin tique du Rluc du couple OXT/OXTR")
92  #AVP
93  graph.ratio(OA,"Courbe cin tique du ratio du couple AVP/OXTR")
94  graph.venus(OA,"Courbe cin tique de Venus du couple AVP/OXTR")
95  graph.rluc(OA,"Courbe cin tique du Rluc du couple AVP/OXTR")
96
97                #AV1A
98  #OXT
99  graph.ratio(A1Ao,"Courbe cin tique du ratio du couple OXT/AV1A")
100 graph.venus(A1Ao,"Courbe cin tique de Venus du couple OXT/AV1A")
101 graph.rluc(A1Ao,"Courbe cin tique du Rluc du couple OXT/AV1A")
102 #AVP
103 graph.ratio(A1AA,"Courbe cin tique du ratio du couple AVP/AV1A")
104 graph.venus(A1AA,"Courbe cin tique de Venus du couple AVP/AV1A")
105 graph.rluc(A1AA,"Courbe cin tique du Rluc du couple AVP/AV1A")
106
107               #AV1B
108 #OXT
109 graph.ratio(A1Bo,"Courbe cin tique du ratio du couple OXT/AV1B")
110 graph.venus(A1Bo,"Courbe cin tique de Venus du couple OXT/AV1B")
111 graph.rluc(A1Bo,"Courbe cin tique du Rluc du couple OXT/AV1B")
```

```
#AVP
graph.ratio(A1BA,"Courbe cin tique du ratio du couple AVP/AV1B")
graph.venus(A1BA,"Courbe cin tique de Venus du couple AVP/AV1B")
graph.rluc(A1BA,"Courbe cin tique du Rluc du couple AVP/AV1B")

                #AV2
#OXT
graph.ratio(A2o,"Courbe cin tique du ratio du couple OXT/AV2")
graph.venus(A2o,"Courbe cin tique de Venus du couple OXT/AV2")
graph.rluc(A2o,"Courbe cin tique du Rluc du couple OXT/AV2")
#AVP
graph.ratio(A2a,"Courbe cin tique du ratio du couple AVP/AV2")
graph.venus(A2a,"Courbe cin tique de Venus du couple AVP/AV2")
graph.rluc(A2a,"Courbe cin tique du Rluc du couple AVP/AV2")
```

Calcium

Graphiques : (cf.Annexe D.3 page 176). Retour au rapport : (cf.Annexe A.1.2 page 36)

```
#graphique des donees brutes
graph.tot=function(X,q){
  n=c("10-4M","10-5M","10-6M","10-7M","10-8M","10-9M","10-10M",
  "basal")
  a=seq(2,33,by=4)
  for (i in 1:8){
    plot(X[["t"]][,1],X[["t"]][,a[i]],type = "l",xlab = "min",
    ylab = "",ylim = c(0,600000))
    title(n[i])
    lines(X[["t"]][,1],X[["t"]][,a[i]+1],col="black")
    lines(X[["t"]][,1],X[["t"]][,a[i]+2],col="blue")
    lines(X[["t"]][,1],X[["t"]][,a[i]+3],col="blue")
  }
  par(xpd=NA)
  legend(-90,800000,c("Test 1","Test 2"),fill = c("black","blue"),
  cex=0.9,x.intersp=0,y.intersp=0, horiz=TRUE,text.width= 0.1,
  xjust = 0.5)
  title( main=q,outer = T,col.main="blue",line = -1)
```

```
}
#graphiques des donees normalise
graph.ntot=function(X,q){
  n=c("10-4M","10-5M","10-6M","10-7M","10-8M","10-9M","10-10M",
  "basal")
  a=seq(2,33,by=4)
  for (i in 1:8){
    plot(X[["nt"]][,1],X[["nt"]][,a[i]],type = "l",xlab = "min",
    ylab = "",ylim = c(0,0.4))
    title(n[i])
    lines(X[["nt"]][,1],X[["nt"]][,a[i]+1],col="black")
    lines(X[["nt"]][,1],X[["nt"]][,a[i]+2],col="blue")
    lines(X[["nt"]][,1],X[["nt"]][,a[i]+3],col="blue")
  }
  par(xpd=NA)
  legend(-90,0.55,c("Test 1","Test 2"),fill =c("black","blue"),
  cex=0.9,x.intersp=0,y.intersp=0, horiz=TRUE,
  text.width= 0.1,xjust = 0.5)
  title( main=q,outer = T,col.main="blue",line = -1)
}
#variante quand 3 replicat
graph.tot2=function(X,q){
  n=c("10-4M","10-5M","10-6M","10-7M","10-8M","10-9M","10-10M",
  "basal")
  a=seq(2,25,by=3)
  for (i in 1:8){
    plot(X[["t"]][,1],X[["t"]][,a[i]],type = "l",xlab = "min",
    ylab = "",ylim = c(0,600000))
    title(n[i])
    lines(X[["t"]][,1],X[["t"]][,a[i]+1],col="black")
    lines(X[["t"]][,1],X[["t"]][,a[i]+2],col="blue")
  }
  par(xpd=NA)
  legend(-90,800000,c("Test 1","Test 2"),fill = c("black","blue"),
  cex=0.9,x.intersp=0,y.intersp=0, horiz=TRUE,
  text.width    = 0.1,xjust = 0.5)
```

```
  title( main=q,outer = T,col.main="blue",line = -1)
}
graph.ntot2=function(X,q){
  n=c("10-4M","10-5M","10-6M","10-7M","10-8M","10-9M","10-10M",
  "basal")
  a=seq(2,25,by=3)
  for (i in 1:8){
    plot(X[["nt"]][,1],X[["nt"]][,a[i]],type = "l",xlab = "min",
    ylab = "",ylim = c(0,0.4))
    title(n[i])
    lines(X[["nt"]][,1],X[["nt"]][,a[i]+1],col="black")
    lines(X[["nt"]][,1],X[["nt"]][,a[i]+2],col="blue")
  }
  par(xpd=NA)
  legend(-90,0.5,c("Test 1","Test 2"),fill = c("black","blue"),
  cex=0.9,x.intersp=0,y.intersp=0, horiz=TRUE,
  text.width    = 0.1,xjust = 0.5)
  title( main=q,outer = T,col.main="blue",line = -1)
}


par(mfrow = c(2,4))
                          #OXTR
graph.tot(oo,"Calcium courbes cin tiques brutes OXT/OXTR")
graph.ntot(oo,"Calcium courbes cin tiques normalis  OXT/OXTR")
graph.tot(oa,"Calcium courbes cin tiques brutes AVP/OXTR")
graph.ntot(oa,"Calcium courbes cin tiques normalis  AVP/OXTR")
                          #AV1A
graph.tot(a1aa,"Calcium courbes cin tiques brutes AVP/AV1A")
graph.ntot(a1aa,"Calcium courbes cin tiques normalis  AVP/AV1A")
graph.tot2(a1ao,"Calcium courbes cin tiques brutes OXT/AV1A")
graph.ntot2(a1ao,"Calcium courbes cin tiques normalis  OXT/AV1A")

                          #AV1B
graph.tot(a1bo,"Calcium courbes cin tiques brutes OXT/AV1B")
graph.ntot(a1bo,"Calcium courbes cin tiques normalis  OXT/AV1B")
```

```
graph.tot(a1ba,"Calcium courbes cin tiques brutes AVP/AV1B")
graph.ntot(a1ba,"Calcium courbes cin tiques normalis  AVP/AV1B")

                          #AV2
graph.tot(a2o,"Calcium courbes cin tiques brutes OXT/AV2")
graph.ntot(a2o,"Calcium courbes cin tiques normalis  OXT/AV2")
graph.tot(a2a,"Calcium courbes cin tiques brutes AVP/AV2")
graph.ntot(a2a,"Calcium courbes cin tiques normalis  AVP/AV2")
```

C.3 Noramlité

C.3.1 code général

(Retour au rapport : 5.2.Construction du modèle statistique)

```
#Calcul ratio - moyenne de chaque exp rience pour avoir bruit
#Calcul valeur moyenne pour 2  r plicat
moy=function(X){
  m=matrix(0,ncol=ncol(X)/2+1,nrow = nrow(X))
  m[,1]=X[,1]
  a=seq(0,100,by=1)
  for (j in 2:(ncol(X)/2+1)){
    for (i in 1:nrow(X)){
      m[i,j]=(X[i,(j+a[j-1])]+X[i,(j+a[j])])/2
    }
  }
  return(m)
}
#Calcul valeur moyenne pour 3  r plicats
moy3=function(X){
  m=matrix(0,ncol=((ncol(X)-1)/3+1),nrow = nrow(X))
  m[,1]=X[,1]
  a=seq(2,100,by=3)
  for (j in 2:((ncol(X)-1)/3+1)){
    for (i in 1:nrow(X)){
      m[i,j]=(X[i,(a[j-1])]+X[i,(a[j-1]+1)]+X[i,(a[j-1]+2)])/3
    }
```

```
  }
  return(m)
}

#Calcul r sidues : donn es-moyenne (bruit)
e=function(X){
  M=moy(X)
  e=matrix(nrow = nrow(X),ncol = ncol(X))
  e[,1]=X[,1]
  a=seq(0,100,by=1)
  for (i in 1:nrow(X)){
    for (j in seq(2,ncol(X),by=2)){
      e[i,j]=X[i,j]-M[i,j/2+1]
      e[i,j+1]=X[i,j+1]-M[i,j/2+1]
    }
  }
  return(e)
}
#Calcul r sidues pour 3 r plicats
e3=function(X){
  M=moy3(X)
  e=matrix(nrow = nrow(X),ncol = ncol(X))
  e[,1]=X[,1]
  for (i in 1:nrow(X)){
    for (j in seq(2,ncol(X),by=3)){
      e[i,j]=X[i,j]-M[i,which(seq(2,ncol(X),by=3)==j)+1]
      e[i,j+1]=X[i,j+1]-M[i,which(seq(2,ncol(X),by=3)==j)+1]
      e[i,j+2]=X[i,j+2]-M[i,which(seq(2,ncol(X),by=3)==j)+1]
    }
  }
  return(e)
}
#bruit suit loi normal?
hist.bruit=function(X){
  n=c("10-5M","10-5M","10-6M","10-6M","10-7M","10-7M","10-8M","10-8M","10-9M","1
  a=e(X)
```

```
  par(mfrow = c(1,2))
  for (i in 2:ncol(a)){
    hist(a[,i],main=n[i-1], prob=T)
    curve(dnorm(x,mean(a[,i]),sd(a[,i])),add=T,col="red")
  }
}
qq.bruit=function(X){
  n=c("10-5M","10-5M","10-6M","10-6M","10-7M","10-7M","10-8M","10-8M","10-9
  a=e(X)
  par(mfrow = c(1,2))
  for (i in 2:ncol(a)){
    qqnorm(X[,i], datax=TRUE, main=n[i-1])
    qqline(X[,i],datax=TRUE,col="red")
  }
}

#test stat
ks.bruit=function(X){
  A=c()
  for (i in 2:ncol(X)){
    A=c(A,ks.test(X[,i],"pnorm",mean(X[,i]),sd(X[,i]))$p.value)
  }
  return(A)
}
ST.bruit=function(X){
  A=c()
  for (i in 2:ncol(X)){
    A=c(A,shapiro.test(X[,i])$p.value)
  }
  return(A)
}

library(nortest)
LL.bruit=function(X){
  A=c()
  for (i in 2:ncol(X)){
```

```
    A=c(A,lillie.test(X[,i])$p.value)
  }
  return(A)
}
```

C.3.2 beta-arrestine

(Retour au rapport : 5.2.1.Beta-arrestine, graphique :(cf.Annexe D.4.1 page 176))

test

```
>                                #AV1A OXT
> X=ba_aao
> which(ST.bruit(e(X))<0.05)
[1] 11 12 15 16 17 18
> which(ks.bruit(e(X))<0.05)
integer(0)
> which(LL.bruit(e(X))<0.05)
integer(0)
>                                 #AV1A AVP
> X=ba_aaa
> which(ST.bruit(e(X))<0.05)
[1] 11 12
> which(ks.bruit(e(X))<0.05)
integer(0)
> which(LL.bruit(e(X))<0.05)
integer(0)
                                 #AV1B OXT
> X=ba_abo
> which(ST.bruit(e(X))<0.05)
[1] 31 32
> which(ks.bruit(e(X))<0.05)
integer(0)
> which(LL.bruit(e(X))<0.05)
integer(0)
>                                  #AV1B AVP
```

```
> X=ba_aba
> which(ST.bruit(e(X))<0.05)
[1] 29 30 31 32
> which(ks.bruit(e(X))<0.05)
integer(0)
> which(LL.bruit(e(X))<0.05)
integer(0)
>                                  #AV2 OXT
> X=ba_a2o
> which(ST.bruit(e(X))<0.05)
[1] 11 12
> which(ks.bruit(e(X))<0.05)
integer(0)
> which(LL.bruit(e(X))<0.05)
integer(0)
>                                  #AV2 AVP
> X=ba_a2a
> which(ST.bruit(e(X))<0.05)
[1] 11 12 15 16 17 18
> which(ks.bruit(e(X))<0.05)
integer(0)
> which(LL.bruit(e(X))<0.05)
integer(0)
```

graphique

C.3.3 Calcium

(Retour au rapport : 5.2.2.Calcium, graphique :FTNC)

test

```
>                               #OXTR AVP
> X=c_oa
> skewness(e(X))
```

```
[1] 7.266615
> kurtosis(e(X))
[1] 54.68096
> which(ST.bruit(e(X))<0.05)
 [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19
 20 21 22 23 24 25 26 27 28 29 30 31 32
> which(ks.bruit(e(X))<0.05)
 [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19
 20 21 22 23 24 25 26 27 28 29 30 31 32
> which(LL.bruit(e(X))<0.05)
 [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19
 20 21 22 23 24 25 26 27 28 29 30 31 32
> #Calcium suit loi log-normal?
> X=X[which(X[,1]==15):which(X[,1]==3),]
> a=log(abs(e(X))+0.000000001)
> skewness(a)
[1] 1.609895
> kurtosis(a)
[1] 8.790671
> which(ST.bruit(a)<0.05)
 [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19
 20 21 22 23 24 25 26 27 28 29
[30] 30 31 32
> which(ks.bruit(a)<0.05)
 [1]  7  8  9 10 11 12 13 14 15 16 19 20 23 24 25 26 27 28 29
 30 31 32
> which(LL.bruit(a)<0.05)
 [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 19 20 21
 22 23 24 25 26 27 28 29 30 31 32
>
>                          #AV1A OXT
> X=c_aao
> skewness(e(X))
[1] 7.266615
> kurtosis(e(X))
[1] 54.68096
```

```
> which(ST.bruit(e(X))<0.05)
 [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19
 20 21 22 23 24 25 26 27 28 29 30 31 32
> which(ks.bruit(e(X))<0.05)
 [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19
 20 21 22 23 24 25 26 27 28 29 30 31 32
> which(LL.bruit(e(X))<0.05)
 [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19
 20 21 22 23 24 25 26 27 28 29 30 31 32
#Calcium suit loi log-normal?
> X=X[which(X[,1]==15):which(X[,1]==3),]
> a=log(abs(e(X))+0.000000001)
> skewness(a)
[1] 1.609895
> kurtosis(a)
[1] 8.790671
> which(ST.bruit(a)<0.05)
 [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19
 20 21 22 23 24 25 26 27 28 29
[30] 30 31 32
> which(ks.bruit(a)<0.05)
 [1]  7  8  9 10 11 12 13 14 15 16 19 20 23 24 25 26 27 28 29
 30 31 32
> which(LL.bruit(a)<0.05)
 [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 19 20 21
 22 23 24 25 26 27 28 29 30 31 32

>                          #AV1A AVP
> X=c_aaa
> skewness(e(X))
[1] 7.266615
> kurtosis(e(X))
[1] 54.68096
> which(ST.bruit(e(X))<0.05)
 [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19
 20 21 22 23 24 25 26 27 28 29 30 31 32
```

```
> which(ks.bruit(e(X))<0.05)
 [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19
 20 21 22 23 24 25 26 27 28 29 30 31 32
> which(LL.bruit(e(X))<0.05)
 [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19
 20 21 22 23 24 25 26 27 28 29 30 31 32
 #Calcium suit loi log-normal?
> X=X[which(X[,1]==15):which(X[,1]==3),]
> a=log(abs(e(X))+0.000000001)
> skewness(a)
[1] 1.609895
> kurtosis(a)
[1] 8.790671
> which(ST.bruit(a)<0.05)
 [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19
 20 21 22 23 24 25 26 27 28 29
[30] 30 31 32
> which(ks.bruit(a)<0.05)
 [1]  7  8  9 10 11 12 13 14 15 16 19 20 23 24 25 26 27 28 29
 30 31 32
There were 12 warnings (use warnings() to see them)
> which(LL.bruit(a)<0.05)
 [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 19 20 21
 22 23 24 25 26 27 28 29 30 31 32

>                        #AV1B OXT
> X=c_abo
> skewness(e(X))
[1] 7.266603
> kurtosis(e(X))
[1] 54.68084
> which(ST.bruit(e(X))<0.05)
 [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19
 20 21 22 23 24 25 26 27 28 29 30 31 32
> which(ks.bruit(e(X))<0.05)
 [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19
```

```
 20 21 22 23 24 25 26 27 28 29 30 31 32
> which(LL.bruit(e(X))<0.05)
 [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19
 20 21 22 23 24 25 26 27 28 29 30 31 32
 #Calcium suit loi log-normal?
> X=X[which(X[,1]==15):which(X[,1]==3),]
> a=log(abs(e(X))+0.000000001)
> skewness(a)
[1] 1.609895
> kurtosis(a)
[1] 8.790671
> which(ST.bruit(a)<0.05)
 [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19
 20 21 22 23 24 25 26 27 28 29
[30] 30 31 32
> which(ks.bruit(a)<0.05)
 [1]  7  8  9 10 11 12 13 14 15 16 19 20 23 24 25 26 27 28 29
 30 31 32
> which(LL.bruit(a)<0.05)
 [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 19 20 21
 22 23 24 25 26 27 28 29 30 31 32

>                              #AV1B AVP
> X=c_aba
> skewness(e(X))
[1] 7.266603
> kurtosis(e(X))
[1] 54.68084
> which(ST.bruit(e(X))<0.05)
 [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19
 20 21 22 23 24 25 26 27 28 29 30 31 32
> which(ks.bruit(e(X))<0.05)
 [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19
 20 21 22 23 24 25 26 27 28 29 30 31 32
> which(LL.bruit(e(X))<0.05)
 [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19
```

```
 20 21 22 23 24 25 26 27 28 29 30 31 32
> #Calcium suit loi log-normal?
> X=X[which(X[,1]==15):which(X[,1]==3),]
> a=log(abs(e(X))+0.000000001)
> skewness(a)
[1] 1.278035
> kurtosis(a)
[1] 9.82389
> which(ST.bruit(a)<0.05)
 [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19
 20 21 22 23 24 25 26 27 28 29 30 31 32
> which(ks.bruit(a)<0.05)
 [1]  5  6  9 10 13 14 17 18 21 22 23 24 25 26 27 28 29 30 31
 32
> which(LL.bruit(a)<0.05)
 [1]  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21
 22 23 24 25 26 27 28 29 30 31 32

>                          #AV2 OXT
> X=c_a2o
> skewness(e(X))
[1] 7.266605
> kurtosis(e(X))
[1] 54.68087
> which(ST.bruit(e(X))<0.05)
 [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19
 20 21 22 23 24 25 26 27 28 29
[30] 30 31 32
> which(ks.bruit(e(X))<0.05)
 [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19
 20 21 22 23 24 25 26 27 28 29 30 31 32
> which(LL.bruit(e(X))<0.05)
 [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19
 20 21 22 23 24 25 26 27 28 29
[30] 30 31 32
> #Calcium suit loi log-normal?
```

```
> X=X[which(X[,1]==15):which(X[,1]==3),]
> a=log(abs(e(X))+0.000000001)
> skewness(a)
[1] 1.25783
> kurtosis(a)
[1] 8.766921
> which(ST.bruit(a)<0.05)
 [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19
 20 21 22 23 24 25 26 27 28 29 30 31 32
> which(ks.bruit(a)<0.05)
 [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 21
 22 25 26 27 28 29 30 31 32
> which(LL.bruit(a)<0.05)
 [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19
 20 21 22 25 26 27 28 29 30 31 32
> #AV2 AVP
> X=c_a2a
> skewness(e(X))
[1] 7.266599
> kurtosis(e(X))
[1] 54.6808
> which(ST.bruit(e(X))<0.05)
 [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19
 20 21 22 23 24 25 26 27 28 29 30 31 32
> which(ks.bruit(e(X))<0.05)
 [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19
 20 21 22 23 24 25 26 27 28 29 30 31 32
> which(LL.bruit(e(X))<0.05)
 [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19
 20 21 22 23 24 25 26 27 28 29 30 31 32
> #Calcium suit loi log-normal?
> X=X[which(X[,1]==15):which(X[,1]==3),]
> a=log(abs(e(X))+0.000000001)
> skewness(a)
[1] 1.693752
> kurtosis(a)
```

```
[1] 9.553009
> which(ST.bruit(a)<0.05)
 [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19
 20 21 22 23 24 25 26 27 28 29 30 31 32
> which(ks.bruit(a)<0.05)
 [1]  1  2  3  4  7  8 11 12 13 14 15 16 19 20 21 22 23 24 25
 26 27 28 29 30 31 32
> which(LL.bruit(a)<0.05)
 [1]  1  2  3  4  5  6  7  8 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27
```

graphique

C.4 D2D

C.4.1 Création tableau de donnée

```
    tab=function(x,g){
  X=matrix(nrow = (nrow(x)*4), ncol = 2)
  #r p tition de la colonne temps
  a=seq(1,nrow(x)*4,4)
  for (j in 1:nrow(x)){
    for (i in 1:4){
      X[i+a[j]-1,1]=x[j,1]
    }
  }
  #donn es
  for (j in 1:nrow(x)){
    for (i in 1:4){
      X[i+a[j]-1,2]=x[j,a[g]+i]
    }
  }
  X[which(is.na(X[,2])==TRUE),2]=""
  return(X)
}
```

```
19
20 library(readxl)
21             #OXTR/AVP
22
23 #beta
24 ba_oa <- data.frame(read_excel("stageM1/donn es/Kinetics & Dose data-OXTR
25 x=ba_oa
26
27 X2=tab(x,1)
28 colnames(X2)=c("temps","B_au")
29 write.table(X2, file="stageM1/donn es/tab/AVP_OXTR_10-5_b.xls",, quote=TRU
30             dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c
31 X2=tab(x,2)
32 colnames(X2)=c("temps","B_au")
33 write.table(X2, file="stageM1/donn es/tab/AVP_OXTR_10-6_b.xls",, quote=TRU
34             dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c
35 X2=tab(x,3)
36 colnames(X2)=c("temps","B_au")
37 write.table(X2, file="stageM1/donn es/tab/AVP_OXTR_10-7_b.xls",, quote=TRU
38             dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c
39 X2=tab(x,4)
40 colnames(X2)=c("temps","B_au")
41 write.table(X2, file="stageM1/donn es/tab/AVP_OXTR_10-8_b.xls",, quote=TRU
42             dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c
43
44 X2=tab(x,5)
45 colnames(X2)=c("temps","B_au")
46 write.table(X2, file="stageM1/donn es/tab/AVP_OXTR_10-9_b.xls",, quote=TRU
47             dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c
48 X2=tab(x,6)
49 colnames(X2)=c("temps","B_au")
50 write.table(X2, file="stageM1/donn es/tab/AVP_OXTR_10-10_b.xls",, quote=TR
51             dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c
52 X2=tab(x,7)
53 colnames(X2)=c("temps","B_au")
54 write.table(X2, file="stageM1/donn es/tab/AVP_OXTR_10-11_b.xls",, quote=TR
```

```
            dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c("esc
X2=tab(x,8)
colnames(X2)=c("temps","B_au")
write.table(X2, file="stageM1/donn es/tab/AVP_OXTR_basal_b.xls",, quote=TRUE,
            dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c("esc
```

```
        #calcium

x=c_oa
x=x[61:nrow(x),]
x[,1]=x[,1]-3
X2=tab(x,1)
colnames(X2)=c("temps","Ca_au")
write.table(X2, file="~/stageM1/donn es/tab/AVP_OXTR_10-4_ca.csv",
quote=TRUE, dec=",", row.names=FALSE, col.names=TRUE, sep =" ",
qmethod = c("escape"))
X2=tab(x,2)
colnames(X2)=c("temps","Ca_au")
write.table(X2, file="~/stageM1/donn es/tab/AVP_OXTR_10-5_ca.csv",
quote=TRUE,dec=",", row.names=FALSE, col.names=TRUE, sep =" ",
qmethod = c("escape"))
X2=tab(x,3)
colnames(X2)=c("temps","Ca_au")
write.table(X2, file="~/stageM1/donn es/tab/AVP_OXTR_10-6_ca.csv",
quote=TRUE, dec=",", row.names=FALSE, col.names=TRUE, sep =" ",
qmethod = c("escape"))
X2=tab(x,4)
colnames(X2)=c("temps","Ca_au")
write.table(X2, file="~/stageM1/donn es/tab/AVP_OXTR_10-7_ca.csv", quote=TRUE,
dec=",", row.names=FALSE, col.names=TRUE, sep =" ",
qmethod = c("escape"))
```

```
X2=tab(x,5)
colnames(X2)=c("temps","Ca_au")
write.table(X2, file="~/stageM1/donnees/tab/AVP_OXTR_10-8_ca.csv", quote=T
            dec=",", row.names=FALSE, col.names=TRUE, sep =" ",
            qmethod = c("escape"))
X2=tab(x,6)
colnames(X2)=c("temps","Ca_au")
write.table(X2, file="~/stageM1/donnees/tab/AVP_OXTR_10-9_ca.csv", quote=T
            dec=",", row.names=FALSE, col.names=TRUE, sep =" ",
            qmethod = c("escape"))
X2=tab(x,7)
colnames(X2)=c("temps","Ca_au")
write.table(X2, file="~/stageM1/donnees/tab/AVP_OXTR_10-10_ca.csv", quote=
            dec=",", row.names=FALSE, col.names=TRUE, sep =" ",
            qmethod = c("escape"))
X2=tab(x,8)
colnames(X2)=c("temps","Ca_au")
write.table(X2, file="~/stageM1/donnees/tab/AVP_OXTR_basal_ca.csv", quote=
            dec=",", row.names=FALSE, col.names=TRUE, sep =" ",
            qmethod = c("escape"))


# OXTR/OXT

#beta
x=ba.OO
X2=tab(x,1)
colnames(X2)=c("temps","B_au")
write.table(X2, file="stageM1/donnees/tab/OXT_OXTR_10-5_b.xls", quote=TRUE
            dec=",", row.names=FALSE, col.names=TRUE, sep =" ",
            qmethod = c("escape"))
X2=tab(x,2)
colnames(X2)=c("temps","B_au")
write.table(X2, file="stageM1/donnees/tab/OXT_OXTR_10-6_b.xls",quote=TRUE,
            dec=",", row.names=FALSE, col.names=TRUE, sep =" ",
            qmethod = c("escape"))
```

```
X2=tab(x,3)
colnames(X2)=c("temps","B_au")
write.table(X2, file="stageM1/donn es/tab/OXT_OXTR_10-7_b.xls",quote=TRUE,
            dec=",", row.names=FALSE, col.names=TRUE, sep =" ",
            qmethod = c("escape"))
X2=tab(x,4)
colnames(X2)=c("temps","B_au")
write.table(X2, file="stageM1/donn es/tab/OXT_OXTR_10-8_b.xls",, quote=TRUE,
            dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c("esc
X2=tab(x,5)
colnames(X2)=c("temps","B_au")
write.table(X2, file="stageM1/donn es/tab/OXT_OXTR_10-9_b.xls",, quote=TRUE,
            dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c("esc
X2=tab(x,6)
colnames(X2)=c("temps","B_au")
write.table(X2, file="stageM1/donn es/tab/OXT_OXTR_10-10_b.xls",, quote=TRUE,
            dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c("esc
X2=tab(x,7)
colnames(X2)=c("temps","B_au")
write.table(X2, file="stageM1/donn es/tab/OXT_OXTR_10-11_b.xls",, quote=TRUE,
            dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c("esc
X2=tab(x,8)
colnames(X2)=c("temps","B_au")
write.table(X2, file="stageM1/donn es/tab/OXT_OXTR_basal_b.xls",, quote=TRUE,
            dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c("esc



#calcium
c_oo <- data.frame(read_excel("stageM1/donn es/Kinetics & Dose data-OXTR AVR OX
x=c_oo
x=x[61:nrow(x),]
x[,1]=x[,1]-3
X2=tab(x,1)
colnames(X2)=c("temps","Ca_au")
```

```
163 write.table(X2, file="stageM1/donnes/tab/OXT_OXTR_10-4_ca.csv",, quote=TR
164             dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c
165 X2=tab(x,2)
166 colnames(X2)=c("temps","Ca_au")
167 write.table(X2, file="stageM1/donnes/tab/OXT_OXTR_10-5_ca.csv",, quote=TR
168             dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c
169 X2=tab(x,3)
170 colnames(X2)=c("temps","Ca_au")
171 write.table(X2, file="stageM1/donnes/tab/OXT_OXTR_10-6_ca.csv",, quote=TR
172             dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c
173 X2=tab(x,4)
174 colnames(X2)=c("temps","Ca_au")
175 write.table(X2, file="stageM1/donnes/tab/OXT_OXTR_10-7_ca.csv",, quote=TR
176             dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c
177 X2=tab(x,5)
178 colnames(X2)=c("temps","Ca_au")
179 write.table(X2, file="stageM1/donnes/tab/OXT_OXTR_10-8_ca.csv",, quote=TR
180             dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c
181 X2=tab(x,6)
182 colnames(X2)=c("temps","Ca_au")
183 write.table(X2, file="stageM1/donnes/tab/OXT_OXTR_10-9_ca.csv",, quote=TR
184             dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c
185 X2=tab(x,7)
186 colnames(X2)=c("temps","Ca_au")
187 write.table(X2, file="stageM1/donnes/tab/OXT_OXTR_10-10_ca.csv",, quote=T
188             dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c
189 X2=tab(x,8)
190 colnames(X2)=c("temps","Ca_au")
191 write.table(X2, file="stageM1/donnes/tab/OXT_OXTR_basal_ca.csv",, quote=T
192             dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c
193
194
195 #assemblage des deux tableaux
196 # A=merge(X,X2,by="temps",all=T)
197 # A[which(is.na(A[,2])==TRUE),2]=""
198 # A[which(is.na(A[,4])==TRUE),4]=""
```

```
# A[,3]=1
# A[,5]=1
# write.table(A, file="stageM1/donn es/OXTR_OXT5.xls",, quote=TRUE,
#             dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c("e
                          # AV1A OXT
#beta
ba_aao <- data.frame(read_excel("stageM1/donn es/Kinetics & Dose data-OXTR AVR
x=ba_aao
X2=tab(x,1)
colnames(X2)=c("temps","B_au")
write.table(X2, file="stageM1/donn es/tab/OXT_AV1A_10-5_b.xls",, quote=TRUE,
            dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c("esc
X2=tab(x,2)
colnames(X2)=c("temps","B_au")
write.table(X2, file="stageM1/donn es/tab/OXT_AV1A_10-6_b.xls",, quote=TRUE,
            dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c("esc
X2=tab(x,3)
colnames(X2)=c("temps","B_au")
write.table(X2, file="stageM1/donn es/tab/OXT_AV1A_10-7_b.xls",, quote=TRUE,
            dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c("esc
X2=tab(x,4)
colnames(X2)=c("temps","B_au")
write.table(X2, file="stageM1/donn es/tab/OXT_AV1A_10-8_b.xls",, quote=TRUE,
            dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c("esc
X2=tab(x,5)
colnames(X2)=c("temps","B_au")
```

```
write.table(X2, file="stageM1/donnees/tab/OXT_AV1A_10-9_b.xls",, quote=TRU
            dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c
X2=tab(x,6)
colnames(X2)=c("temps","B_au")
write.table(X2, file="stageM1/donnees/tab/OXT_AV1A_10-10_b.xls",, quote=TR
            dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c
X2=tab(x,7)
colnames(X2)=c("temps","B_au")
write.table(X2, file="stageM1/donnees/tab/OXT_AV1A_10-11_b.xls",, quote=TR
            dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c
X2=tab(x,8)
colnames(X2)=c("temps","B_au")
write.table(X2, file="stageM1/donnees/tab/OXT_AV1A_basal_b.xls",, quote=TR
            dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c
#calcium
c_aao <- data.frame(read_excel("stageM1/donnees/Kinetics & Dose data-OXTR
x=c_aao
x=x[61:nrow(x),]
x[,1]=x[,1]-3
X2=tab(x,1)
colnames(X2)=c("temps","Ca_au")
write.table(X2, file="stageM1/donnees/tab/OXT_AV1A_10-4_ca.csv",, quote=TR
            dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c
X2=tab(x,2)
colnames(X2)=c("temps","Ca_au")
write.table(X2, file="stageM1/donnees/tab/OXT_AV1A_10-5_ca.csv",, quote=TR
            dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c
X2=tab(x,3)
colnames(X2)=c("temps","Ca_au")
write.table(X2, file="stageM1/donnees/tab/OXT_AV1A_10-6_ca.csv",, quote=TR
            dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c
X2=tab(x,4)
colnames(X2)=c("temps","Ca_au")
write.table(X2, file="stageM1/donnees/tab/OXT_AV1A_10-7_ca.csv",, quote=TR
            dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c
X2=tab(x,5)
```

```
colnames(X2)=c("temps","Ca_au")
write.table(X2, file="stageM1/donn es/tab/OXT_AV1A_10-8_ca.csv",, quote=TRUE,
            dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c("esc
X2=tab(x,6)
colnames(X2)=c("temps","Ca_au")
write.table(X2, file="stageM1/donn es/tab/OXT_AV1A_10-9_ca.csv",, quote=TRUE,
            dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c("esc
X2=tab(x,7)
colnames(X2)=c("temps","Ca_au")
write.table(X2, file="stageM1/donn es/tab/OXT_AV1A_10-10_ca.csv",, quote=TRUE,
            dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c("esc
X2=tab(x,8)
colnames(X2)=c("temps","Ca_au")
write.table(X2, file="stageM1/donn es/tab/OXT_AV1A_basal_ca.csv",, quote=TRUE,
            dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c("esc
```

```
                # AV1A AVP
#beta
ba_aaa <- data.frame(read_excel("stageM1/donn es/Kinetics & Dose data-OXTR AVR
x=ba_aaa
X2=tab(x,1)
colnames(X2)=c("temps","B_au")
write.table(X2, file="stageM1/donn es/tab/AVP_AV1A_10-5_b.xls",, quote=TRUE,
            dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c("esc
X2=tab(x,2)
colnames(X2)=c("temps","B_au")
write.table(X2, file="stageM1/donn es/tab/AVP_AV1A_10-6_b.xls",, quote=TRUE,
            dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c("esc
```

```
X2=tab(x,3)
colnames(X2)=c("temps","B_au")
write.table(X2, file="stageM1/donnees/tab/AVP_AV1A_10-7_b.xls",, quote=TRU
            dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c
X2=tab(x,4)
colnames(X2)=c("temps","B_au")
write.table(X2, file="stageM1/donnees/tab/AVP_AV1A_10-8_b.xls",, quote=TRU
            dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c
X2=tab(x,5)
colnames(X2)=c("temps","B_au")
write.table(X2, file="stageM1/donnees/tab/AVP_AV1A_10-9_b.xls",, quote=TRU
            dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c
X2=tab(x,6)
colnames(X2)=c("temps","B_au")
write.table(X2, file="stageM1/donnees/tab/AVP_AV1A_10-10_b.xls",, quote=TR
            dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c
X2=tab(x,7)
colnames(X2)=c("temps","B_au")
write.table(X2, file="stageM1/donnees/tab/AVP_AV1A_10-11_b.xls",, quote=TR
            dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c
X2=tab(x,8)
colnames(X2)=c("temps","B_au")
write.table(X2, file="stageM1/donnees/tab/AVP_AV1A_basal_b.xls",, quote=TR
            dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c
#calcium
c_aaa <- data.frame(read_excel("stageM1/donnees/Kinetics & Dose data-OXTR
x=c_aaa
x=x[61:nrow(x),]
x[,1]=x[,1]-3
X2=tab(x,1)
colnames(X2)=c("temps","Ca_au")
write.table(X2, file="stageM1/donnees/tab/AVP_AV1A_10-4_ca.csv",, quote=TR
            dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c
X2=tab(x,2)
colnames(X2)=c("temps","Ca_au")
write.table(X2, file="stageM1/donnees/tab/AVP_AV1A_10-5_ca.csv",, quote=TR
```

```
343             dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c("esc
344 X2=tab(x,3)
345 colnames(X2)=c("temps","Ca_au")
346 write.table(X2, file="stageM1/donn es/tab/AVP_AV1A_10-6_ca.csv",, quote=TRUE,
347             dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c("esc
348 X2=tab(x,4)
349 colnames(X2)=c("temps","Ca_au")
350 write.table(X2, file="stageM1/donn es/tab/AVP_AV1A_10-7_ca.csv",, quote=TRUE,
351             dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c("esc
352 X2=tab(x,5)
353 colnames(X2)=c("temps","Ca_au")
354 write.table(X2, file="stageM1/donn es/tab/AVP_AV1A_10-8_ca.csv",, quote=TRUE,
355             dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c("esc
356 X2=tab(x,6)
357 colnames(X2)=c("temps","Ca_au")
358 write.table(X2, file="stageM1/donn es/tab/AVP_AV1A_10-9_ca.csv",, quote=TRUE,
359             dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c("esc
360 X2=tab(x,7)
361 colnames(X2)=c("temps","Ca_au")
362 write.table(X2, file="stageM1/donn es/tab/AVP_AV1A_10-10_ca.csv",, quote=TRUE,
363             dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c("esc
364 X2=tab(x,8)
365 colnames(X2)=c("temps","Ca_au")
366 write.table(X2, file="stageM1/donn es/tab/AVP_AV1A_basal_ca.csv",, quote=TRUE,
367             dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c("esc
368
369
370
371
372
373
374
375
376                                 # AV1B OXT
377 #beta
378 ba_abo <- data.frame(read_excel("stageM1/donn es/Kinetics & Dose data-OXTR AVR
```

```
379 x=ba_abo
380 X2=tab(x,1)
381 colnames(X2)=c("temps","B_au")
382 write.table(X2, file="stageM1/donnees/tab/OXT_AV1B_10-5_b.xls",, quote=TRU
383             dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c
384 X2=tab(x,2)
385 colnames(X2)=c("temps","B_au")
386 write.table(X2, file="stageM1/donnees/tab/OXT_AV1B_10-6_b.xls",, quote=TRU
387             dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c
388 X2=tab(x,3)
389 colnames(X2)=c("temps","B_au")
390 write.table(X2, file="stageM1/donnees/tab/OXT_AV1B_10-7_b.xls",, quote=TRU
391             dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c
392 X2=tab(x,4)
393 colnames(X2)=c("temps","B_au")
394 write.table(X2, file="stageM1/donnees/tab/OXT_AV1B_10-8_b.xls",, quote=TRU
395             dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c
396 X2=tab(x,5)
397 colnames(X2)=c("temps","B_au")
398 write.table(X2, file="stageM1/donnees/tab/OXT_AV1B_10-9_b.xls",, quote=TRU
399             dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c
400 X2=tab(x,6)
401 colnames(X2)=c("temps","B_au")
402 write.table(X2, file="stageM1/donnees/tab/OXT_AV1B_10-10_b.xls",, quote=TR
403             dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c
404 X2=tab(x,7)
405 colnames(X2)=c("temps","B_au")
406 write.table(X2, file="stageM1/donnees/tab/OXT_AV1B_10-11_b.xls",, quote=TR
407             dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c
408 X2=tab(x,8)
409 colnames(X2)=c("temps","B_au")
410 write.table(X2, file="stageM1/donnees/tab/OXT_AV1B_basal_b.xls",, quote=TR
411             dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c
412 #calcium
413 c_abo <- data.frame(read_excel("stageM1/donnees/Kinetics & Dose data-OXTR
414 x=c_abo
```

```
415 x=x[61:nrow(x),]
416 x[,1]=x[,1]-3
417 X2=tab(x,1)
418 colnames(X2)=c("temps","Ca_au")
419 write.table(X2, file="stageM1/donn es/tab/OXT_AV1B_10-4_ca.csv",, quote=TRUE,
420             dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c("esc
421 X2=tab(x,2)
422 colnames(X2)=c("temps","Ca_au")
423 write.table(X2, file="stageM1/donn es/tab/OXT_AV1B_10-5_ca.csv",, quote=TRUE,
424             dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c("esc
425 X2=tab(x,3)
426 colnames(X2)=c("temps","Ca_au")
427 write.table(X2, file="stageM1/donn es/tab/OXT_AV1B_10-6_ca.csv",, quote=TRUE,
428             dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c("esc
429 X2=tab(x,4)
430 colnames(X2)=c("temps","Ca_au")
431 write.table(X2, file="stageM1/donn es/tab/OXT_AV1B_10-7_ca.csv",, quote=TRUE,
432             dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c("esc
433 X2=tab(x,5)
434 colnames(X2)=c("temps","Ca_au")
435 write.table(X2, file="stageM1/donn es/tab/OXT_AV1B_10-8_ca.csv",, quote=TRUE,
436             dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c("esc
437 X2=tab(x,6)
438 colnames(X2)=c("temps","Ca_au")
439 write.table(X2, file="stageM1/donn es/tab/OXT_AV1B_10-9_ca.csv",, quote=TRUE,
440             dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c("esc
441 X2=tab(x,7)
442 colnames(X2)=c("temps","Ca_au")
443 write.table(X2, file="stageM1/donn es/tab/OXT_AV1B_10-10_ca.csv",, quote=TRUE,
444             dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c("esc
445 X2=tab(x,8)
446 colnames(X2)=c("temps","Ca_au")
447 write.table(X2, file="stageM1/donn es/tab/OXT_AV1B_basal_ca.csv",, quote=TRUE,
448             dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c("esc
449
450
```

```
451
452
453
454
455
456
457
458
459
460
461                                  # AV1B AVP
462 #beta
463 ba_aba <- data.frame(read_excel("stageM1/donn es/Kinetics & Dose data-OXTR
464 x=ba_aba
465 X2=tab(x,1)
466 colnames(X2)=c("temps","B_au")
467 write.table(X2, file="stageM1/donn es/tab/AVP_AV1B_10-5_b.xls",, quote=TRU
468             dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c
469 X2=tab(x,2)
470 colnames(X2)=c("temps","B_au")
471 write.table(X2, file="stageM1/donn es/tab/AVP_AV1B_10-6_b.xls",, quote=TRU
472             dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c
473 X2=tab(x,3)
474 colnames(X2)=c("temps","B_au")
475 write.table(X2, file="stageM1/donn es/tab/AVP_AV1B_10-7_b.xls",, quote=TRU
476             dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c
477 X2=tab(x,4)
478 colnames(X2)=c("temps","B_au")
479 write.table(X2, file="stageM1/donn es/tab/AVP_AV1B_10-8_b.xls",, quote=TRU
480             dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c
481 X2=tab(x,5)
482 colnames(X2)=c("temps","B_au")
483 write.table(X2, file="stageM1/donn es/tab/AVP_AV1B_10-9_b.xls",, quote=TRU
484             dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c
485 X2=tab(x,6)
486 colnames(X2)=c("temps","B_au")
```

```
write.table(X2, file="stageM1/donn es/tab/AVP_AV1B_10-10_b.xls",, quote=TRUE,
            dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c("esc
X2=tab(x,7)
colnames(X2)=c("temps","B_au")
write.table(X2, file="stageM1/donn es/tab/AVP_AV1B_10-11_b.xls",, quote=TRUE,
            dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c("esc
X2=tab(x,8)
colnames(X2)=c("temps","B_au")
write.table(X2, file="stageM1/donn es/tab/AVP_AV1B_basal_b.xls",, quote=TRUE,
            dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c("esc
#calcium
c_aba <- data.frame(read_excel("stageM1/donn es/Kinetics & Dose data-OXTR AVR O
x=c_aba
x=x[61:nrow(x),]
x[,1]=x[,1]-3
X2=tab(x,1)
colnames(X2)=c("temps","Ca_au")
write.table(X2, file="stageM1/donn es/tab/AVP_AV1B_10-4_ca.csv",, quote=TRUE,
            dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c("esc
X2=tab(x,2)
colnames(X2)=c("temps","Ca_au")
write.table(X2, file="stageM1/donn es/tab/AVP_AV1B_10-5_ca.csv",, quote=TRUE,
            dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c("esc
X2=tab(x,3)
colnames(X2)=c("temps","Ca_au")
write.table(X2, file="stageM1/donn es/tab/AVP_AV1B_10-6_ca.csv",, quote=TRUE,
            dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c("esc
X2=tab(x,4)
colnames(X2)=c("temps","Ca_au")
write.table(X2, file="stageM1/donn es/tab/AVP_AV1B_10-7_ca.csv",, quote=TRUE,
            dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c("esc
X2=tab(x,5)
colnames(X2)=c("temps","Ca_au")
write.table(X2, file="stageM1/donn es/tab/AVP_AV1B_10-8_ca.csv",, quote=TRUE,
            dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c("esc
X2=tab(x,6)
```

```
colnames(X2)=c("temps","Ca_au")
write.table(X2, file="stageM1/donnees/tab/AVP_AV1B_10-9_ca.csv",, quote=TR
            dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c
X2=tab(x,7)
colnames(X2)=c("temps","Ca_au")
write.table(X2, file="stageM1/donnees/tab/AVP_AV1B_10-10_ca.csv",, quote=T
            dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c
X2=tab(x,8)
colnames(X2)=c("temps","Ca_au")
write.table(X2, file="stageM1/donnees/tab/AVP_AV1B_basal_ca.csv",, quote=T
            dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c
```

```
                                # AV2 OXT
#beta
ba_a2o <- data.frame(read_excel("stageM1/donnees/Kinetics & Dose data-OXTR
x=ba_a2o
X2=tab(x,1)
colnames(X2)=c("temps","B_au")
write.table(X2, file="stageM1/donnees/tab/OXT_AV2_10-5_b.xls",, quote=TRUE
            dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c
X2=tab(x,2)
colnames(X2)=c("temps","B_au")
write.table(X2, file="stageM1/donnees/tab/OXT_AV2_10-6_b.xls",, quote=TRUE
            dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c
X2=tab(x,3)
```

```
559 colnames(X2)=c("temps","B_au")
560 write.table(X2, file="stageM1/donn es/tab/OXT_AV2_10-7_b.xls",, quote=TRUE,
561             dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c("esc
562 X2=tab(x,4)
563 colnames(X2)=c("temps","B_au")
564 write.table(X2, file="stageM1/donn es/tab/OXT_AV2_10-8_b.xls",, quote=TRUE,
565             dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c("esc
566 X2=tab(x,5)
567 colnames(X2)=c("temps","B_au")
568 write.table(X2, file="stageM1/donn es/tab/OXT_AV2_10-9_b.xls",, quote=TRUE,
569             dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c("esc
570 X2=tab(x,6)
571 colnames(X2)=c("temps","B_au")
572 write.table(X2, file="stageM1/donn es/tab/OXT_AV2_10-10_b.xls",, quote=TRUE,
573             dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c("esc
574 X2=tab(x,7)
575 colnames(X2)=c("temps","B_au")
576 write.table(X2, file="stageM1/donn es/tab/OXT_AV2_10-11_b.xls",, quote=TRUE,
577             dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c("esc
578 X2=tab(x,8)
579 colnames(X2)=c("temps","B_au")
580 write.table(X2, file="stageM1/donn es/tab/OXT_AV2_basal_b.xls",, quote=TRUE,
581             dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c("esc
582 #calcium
583 c_a2o <- data.frame(read_excel("stageM1/donn es/Kinetics & Dose data-OXTR AVR O
584 x=c_a2o
585 x=x[61:nrow(x),]
586 x[,1]=x[,1]-3
587 X2=tab(x,1)
588 colnames(X2)=c("temps","Ca_au")
589 write.table(X2, file="stageM1/donn es/tab/OXT_AV2_10-4_ca.csv",, quote=TRUE,
590             dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c("esc
591 X2=tab(x,2)
592 colnames(X2)=c("temps","Ca_au")
593 write.table(X2, file="stageM1/donn es/tab/OXT_AV2_10-5_ca.csv",, quote=TRUE,
594             dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c("esc
```

```
595 X2=tab(x,3)
596 colnames(X2)=c("temps","Ca_au")
597 write.table(X2, file="stageM1/donn es/tab/OXT_AV2_10-6_ca.csv",, quote=TRU
598             dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c
599 X2=tab(x,4)
600 colnames(X2)=c("temps","Ca_au")
601 write.table(X2, file="stageM1/donn es/tab/OXT_AV2_10-7_ca.csv",, quote=TRU
602             dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c
603 X2=tab(x,5)
604 colnames(X2)=c("temps","Ca_au")
605 write.table(X2, file="stageM1/donn es/tab/OXT_AV2_10-8_ca.csv",, quote=TRU
606             dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c
607 X2=tab(x,6)
608 colnames(X2)=c("temps","Ca_au")
609 write.table(X2, file="stageM1/donn es/tab/OXT_AV2_10-9_ca.csv",, quote=TRU
610             dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c
611 X2=tab(x,7)
612 colnames(X2)=c("temps","Ca_au")
613 write.table(X2, file="stageM1/donn es/tab/OXT_AV2_10-10_ca.csv",, quote=TR
614             dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c
615 X2=tab(x,8)
616 colnames(X2)=c("temps","Ca_au")
617 write.table(X2, file="stageM1/donn es/tab/OXT_AV2_basal_ca.csv",, quote=TR
618             dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c
619
620
621
622
623
624
625
626
627
628
629
630
```

```
631                                  # AV2 AVP
632 #beta
633 ba_a2a <- data.frame(read_excel("stageM1/donn es/Kinetics & Dose data-OXTR AVR
634 x=ba_a2a
635 X2=tab(x,1)
636 colnames(X2)=c("temps","B_au")
637 write.table(X2, file="stageM1/donn es/tab/AVP_AV2_10-5_b.xls",, quote=TRUE,
638             dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c("esc
639 X2=tab(x,2)
640 colnames(X2)=c("temps","B_au")
641 write.table(X2, file="stageM1/donn es/tab/AVP_AV2_10-6_b.xls",, quote=TRUE,
642             dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c("esc
643 X2=tab(x,3)
644 colnames(X2)=c("temps","B_au")
645 write.table(X2, file="stageM1/donn es/tab/AVP_AV2_10-7_b.xls",, quote=TRUE,
646             dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c("esc
647 X2=tab(x,4)
648 colnames(X2)=c("temps","B_au")
649 write.table(X2, file="stageM1/donn es/tab/AVP_AV2_10-8_b.xls",, quote=TRUE,
650             dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c("esc
651 X2=tab(x,5)
652 colnames(X2)=c("temps","B_au")
653 write.table(X2, file="stageM1/donn es/tab/AVP_AV2_10-9_b.xls",, quote=TRUE,
654             dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c("esc
655 X2=tab(x,6)
656 colnames(X2)=c("temps","B_au")
657 write.table(X2, file="stageM1/donn es/tab/AVP_AV2_10-10_b.xls",, quote=TRUE,
658             dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c("esc
659 X2=tab(x,7)
660 colnames(X2)=c("temps","B_au")
661 write.table(X2, file="stageM1/donn es/tab/AVP_AV2_10-11_b.xls",, quote=TRUE,
662             dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c("esc
663 X2=tab(x,8)
664 colnames(X2)=c("temps","B_au")
665 write.table(X2, file="stageM1/donn es/tab/AVP_AV2_basal_b.xls",, quote=TRUE,
666             dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c("esc
```

```
667 #calcium
668 c_a2a <- data.frame(read_excel("stageM1/donn es/Kinetics & Dose data-OXTR
669 x=c_a2a
670 x=x[61:nrow(x),]
671 x[,1]=x[,1]-3
672 X2=tab(x,1)
673 colnames(X2)=c("temps","Ca_au")
674 write.table(X2, file="stageM1/donn es/tab/AVP_AV2_10-4_ca.csv",, quote=TRU
675             dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c
676 X2=tab(x,2)
677 colnames(X2)=c("temps","Ca_au")
678 write.table(X2, file="stageM1/donn es/tab/AVP_AV2_10-5_ca.csv",, quote=TRU
679             dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c
680 X2=tab(x,3)
681 colnames(X2)=c("temps","Ca_au")
682 write.table(X2, file="stageM1/donn es/tab/AVP_AV2_10-6_ca.csv",, quote=TRU
683             dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c
684 X2=tab(x,4)
685 colnames(X2)=c("temps","Ca_au")
686 write.table(X2, file="stageM1/donn es/tab/AVP_AV2_10-7_ca.csv",, quote=TRU
687             dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c
688 X2=tab(x,5)
689 colnames(X2)=c("temps","Ca_au")
690 write.table(X2, file="stageM1/donn es/tab/AVP_AV2_10-8_ca.csv",, quote=TRU
691             dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c
692 X2=tab(x,6)
693 colnames(X2)=c("temps","Ca_au")
694 write.table(X2, file="stageM1/donn es/tab/AVP_AV2_10-9_ca.csv",, quote=TRU
695             dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c
696 X2=tab(x,7)
697 colnames(X2)=c("temps","Ca_au")
698 write.table(X2, file="stageM1/donn es/tab/AVP_AV2_10-10_ca.csv",, quote=TR
699             dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c
700 X2=tab(x,8)
701 colnames(X2)=c("temps","Ca_au")
702 write.table(X2, file="stageM1/donn es/tab/AVP_AV2_basal_ca.csv",, quote=TR
```

```
703             dec=",", row.names=FALSE, col.names=TRUE, sep =" ", qmethod = c("esc
```

C.4.2 Models

retour texte

Modèle jouet

```
 DESCRIPTION
"Test Model"

PREDICTOR
t               T   min         time 0 100

COMPARTMENTS

STATES
X_state             X   "x"      "x"

INPUTS

REACTIONS
            ->      X_state        CUSTOM "p"

DERIVED

OBSERVABLES
X_au    X   "au"   "x"  0 0 "X_state"

ERRORS
X_au    "X_au_sd"

CONDITIONS
init_X_state      "0"
```

Modèle 1

```
 DESCRIPTION
"Premier Model"

PREDICTOR
t               T   min         time 0 100

COMPARTMENTS

STATES
Ca_state               C   "nM"      "conc."
B_state                C   "nM"      "conc."
L_state                C   "nM"      "conc."
R_state               C   "nM"      "conc."
LR_state               C   "nM"      "conc."
G_state                C   "nM"      "conc."
LRG_state              C   "nM"      "conc."
DAG_state              C   "nM"      "conc."
IP3_state              C   "nM"      "conc."
IP3R_state              C   "nM"      "conc."
IP3IP3R_state          C   "nM"      "conc."
IP3IP3RCa_state        C   "nM"      "conc."
IP3IP3RCaCa_state      C   "nM"      "conc."
LRp_state              C   "nM"      "conc."
LRpB_state             C   "nM"      "conc."

INPUTS

REACTIONS
L_state + R_state     ->   LR_state                      CUSTOM "k_1*L_state
LR_state              ->  L_state + R_state              CUSTOM "k_2*LR_stat
LR_state + G_state    ->   LRG_state                     CUSTOM "k_3*LR_stat
LRG_state             ->  LR_state + G_state             CUSTOM "k_4*LRG_sta
Ca_state + LRG_state  ->  LR_state + G_state + Ca_state + IP3_state + DAG_s
IP3_state             ->                                 CUSTOM "k_6*IP3_sta
```

```
DAG_state             ->                                  CUSTOM "k_7*DAG_state"
IP3_state + IP3R_state ->  IP3IP3R_state                   CUSTOM "k_8*IP3_state*I
IP3IP3R_state         ->  IP3_state + IP3R_state          CUSTOM "k_9*IP3IP3R_sta
IP3IP3R_state + Ca_state ->  IP3IP3RCa_state              CUSTOM "k_10*IP3IP3R_sta
IP3IP3RCa_state      -> IP3IP3R_state + Ca_state          CUSTOM "k_11*IP3IP3RCa_s
IP3IP3RCa_state + Ca_state ->  IP3IP3RCaCa_state         CUSTOM "k_12*IP3IP3RCa_s
IP3IP3RCaCa_state       -> IP3IP3RCa_state + Ca_state    CUSTOM "k_13*IP3IP3RCaCa
Ca_state                ->                               CUSTOM "(p_1^2*Ca_state^
                        ->  Ca_state                     CUSTOM "K*(1+a*IP3IP3RCa
LR_state + DAG_state    ->   LRp_state                   CUSTOM "k_14*LR_state*DA
LR_state                ->   LRp_state                   CUSTOM "k_15*LR_state"
LRp_state               ->   LR_state                    CUSTOM "k_16*LRp_state"
LRp_state + B_state     ->   LRpB_state                  CUSTOM "k_17*LRp_state*B
LRpB_state              ->                               CUSTOM "k_18*LRpB_state"

DERIVED

OBSERVABLES
Ca_au        C   "au"  "conc."   0   0   "Ca_state"
B_au        C   "au"    "conc." 0   0   "B_state"

ERRORS
Ca_au    "sd_Ca_au"
B_au     "sd_B_au"

CONDITIONS
init_LR_state        "0"
init_LRG_state      "0"
init_IP3IP3R_state      "0"
init_IP3IP3RCa_state    "0"
init_IP3IP3RCaCa_state  "0"
init_LRp_state        "0"
init_LRpB_state       "0"
```

C.4.3 data

```
DESCRIPTION

PREDICTOR
t               T   min         time 0 100

INPUTS

OBSERVABLES

ERRORS

CONDITIONS
```

Pour AVP_AV1 à 1

```
DESCRIPTION

PREDICTOR
t               T   min         time 0 100

INPUTS

OBSERVABLES
Ca_au        C   "au"  "conc."   0   0   "Ca_state"

ERRORS
Ca_au    "sd_Ca_au"

CONDITIONS
init_L_state        "10^(-4)"
init_R_state        "init_AV1A"
k_1                 "k_1_AVP_AV1A"
k_2                 "k_2_AVP_AV1A"
k_3                 "k_3_AVP_AV1A"
k_4                 "k_4_AVP_AV1A"
```

```
k_5                 "k_5_AVP_AV1A"
k_14                 "k_14_AVP_AV1A"
k_15                 "k_15_AVP_AV1A"
k_16                 "k_16_AVP_AV1A"
k_17                 "k_17_AVP_AV1A"
k_18                 "k_18_AVP_AV1A"

DESCRIPTION

PREDICTOR
t               T   min         time 0 100

INPUTS

OBSERVABLES
B_au        C   "au"    "conc." 0   0   "B_state"

ERRORS
B_au     "sd_B_au"

CONDITIONS
init_L_state        "10^(-5)"
init_R_state        "init_AV1B"
k_1                 "k_1_AVP_AV1B"
k_2                 "k_2_AVP_AV1B"
k_3                 "k_3_AVP_AV1B"
k_4                 "k_4_AVP_AV1B"
k_5                 "k_5_AVP_AV1B"
k_14                 "k_14_AVP_AV1B"
k_15                 "k_15_AVP_AV1B"
k_16                 "k_16_AVP_AV1B"
k_17                 "k_17_AVP_AV1B"
k_18                 "k_18_AVP_AV1B"
```

Modèle 1

C.4.4 programmation

Modèle Jouet

```
 clear all;
arInit

arLoadModel('ABC_model');
arLoadData('ABC_data_test');

arCompileAll();

ar.config.ploterrors = -1;
arFitLHS(10);
```

Modèle Test

```
  clear all;
arInit

arLoadModel('p_model');
arLoadData('AVP_OXTR_10-5_b');
arLoadData('AVP_OXTR_10-5_ca');
arLoadData('AVP_OXTR_10-6_b');
arLoadData('AVP_OXTR_10-6_ca');

arLoadData('OXT_OXTR_10-5_b');
arLoadData('OXT_OXTR_10-5_ca');
arLoadData('OXT_OXTR_10-6_b');
arLoadData('OXT_OXTR_10-6_ca');
arCompileAll();

arFitLHS(100);
```

Modèle 1

```
clear all;
arInit
```

```
arLoadModel('p_model');
arLoadData('AVP_OXTR_10-5_b'); %Data with equidistant observation
arLoadData('AVP_OXTR_10-6_b');
arLoadData('AVP_OXTR_10-7_b');
arLoadData('AVP_OXTR_10-8_b');
arLoadData('AVP_OXTR_10-9_b');
arLoadData('AVP_OXTR_10-10_b');
arLoadData('AVP_OXTR_10-11_b');
arLoadData('AVP_OXTR_basal_b');
arLoadData('AVP_OXTR_10-5_ca');
arLoadData('AVP_OXTR_10-6_ca');
arLoadData('AVP_OXTR_10-7_ca');
arLoadData('AVP_OXTR_10-8_ca');
arLoadData('AVP_OXTR_10-9_ca');
arLoadData('AVP_OXTR_10-10_ca');
arLoadData('AVP_OXTR_10-11_ca');
arLoadData('AVP_OXTR_basal_ca');
arLoadData('OXT_OXTR_10-5_b'); %Data with equidistant observation
arLoadData('OXT_OXTR_10-6_b');
arLoadData('OXT_OXTR_10-7_b');
arLoadData('OXT_OXTR_10-8_b');
arLoadData('OXT_OXTR_10-9_b');
arLoadData('OXT_OXTR_10-10_b');
arLoadData('OXT_OXTR_10-11_b');
arLoadData('OXT_OXTR_basal_b');
arLoadData('OXT_OXTR_10-5_ca');
arLoadData('OXT_OXTR_10-6_ca');
arLoadData('OXT_OXTR_10-7_ca');
arLoadData('OXT_OXTR_10-8_ca');
arLoadData('OXT_OXTR_10-9_ca');
arLoadData('OXT_OXTR_10-10_ca');
arLoadData('OXT_OXTR_10-11_ca');
arLoadData('OXT_OXTR_basal_ca');
arLoadData('AVP_AV1A_10-5_b'); %Data with equidistant observation
arLoadData('AVP_AV1A_10-6_b');
```

```
arLoadData('AVP_AV1A_10-7_b');
arLoadData('AVP_AV1A_10-8_b');
arLoadData('AVP_AV1A_10-9_b');
arLoadData('AVP_AV1A_10-10_b');
arLoadData('AVP_AV1A_10-11_b');
arLoadData('AVP_AV1A_basal_b');
arLoadData('AVP_AV1A_10-5_ca');
arLoadData('AVP_AV1A_10-6_ca');
arLoadData('AVP_AV1A_10-7_ca');
arLoadData('AVP_AV1A_10-8_ca');
arLoadData('AVP_AV1A_10-9_ca');
arLoadData('AVP_AV1A_10-10_ca');
arLoadData('AVP_AV1A_10-11_ca');
arLoadData('AVP_AV1A_basal_ca');
arLoadData('OXT_AV1A_10-5_b');
arLoadData('OXT_AV1A_10-6_b');
arLoadData('OXT_AV1A_10-7_b');
arLoadData('OXT_AV1A_10-8_b');
arLoadData('OXT_AV1A_10-9_b');
arLoadData('OXT_AV1A_10-10_b');
arLoadData('OXT_AV1A_10-11_b');
arLoadData('OXT_AV1A_basal_b');
arLoadData('OXT_AV1A_10-5_ca');
arLoadData('OXT_AV1A_10-6_ca');
arLoadData('OXT_AV1A_10-7_ca');
arLoadData('OXT_AV1A_10-8_ca');
arLoadData('OXT_AV1A_10-9_ca');
arLoadData('OXT_AV1A_10-10_ca');
arLoadData('OXT_AV1A_10-11_ca');
arLoadData('OXT_AV1A_basal_ca');
arLoadData('AVP_AV1B_10-5_b'); %Data with equidistant observation
arLoadData('AVP_AV1B_10-6_b');
arLoadData('AVP_AV1B_10-7_b');
arLoadData('AVP_AV1B_10-8_b');
arLoadData('AVP_AV1B_10-9_b');
arLoadData('AVP_AV1B_10-10_b');
```

```
arLoadData('AVP_AV1B_10-11_b');
arLoadData('AVP_AV1B_basal_b');
arLoadData('AVP_AV1B_10-5_ca');
arLoadData('AVP_AV1B_10-6_ca');
arLoadData('AVP_AV1B_10-7_ca');
arLoadData('AVP_AV1B_10-8_ca');
arLoadData('AVP_AV1B_10-9_ca');
arLoadData('AVP_AV1B_10-10_ca');
arLoadData('AVP_AV1B_basal_ca');
arLoadData('AVP_AV1B_10-11_ca');
arLoadData('OXT_AV1B_10-5_b');
arLoadData('OXT_AV1B_10-6_b');
arLoadData('OXT_AV1B_10-7_b');
arLoadData('OXT_AV1B_10-8_b');
arLoadData('OXT_AV1B_10-9_b');
arLoadData('OXT_AV1B_10-10_b');
arLoadData('OXT_AV1B_10-11_b');
arLoadData('OXT_AV1B_basal_b');
arLoadData('OXT_AV1B_10-5_ca');
arLoadData('OXT_AV1B_10-6_ca');
arLoadData('OXT_AV1B_10-7_ca');
arLoadData('OXT_AV1B_10-8_ca');
arLoadData('OXT_AV1B_10-9_ca');
arLoadData('OXT_AV1B_10-10_ca');
arLoadData('OXT_AV1B_10-11_ca');
arLoadData('OXT_AV1B_basal_ca');
arLoadData('AVP_AV2_10-5_b'); %Data with equidistant observation
arLoadData('AVP_AV2_10-6_b');
arLoadData('AVP_AV2_10-7_b');
arLoadData('AVP_AV2_10-8_b');
arLoadData('AVP_AV2_10-9_b');
arLoadData('AVP_AV2_10-10_b');
arLoadData('AVP_AV2_10-11_b');
arLoadData('AVP_AV2_basal_b');
arLoadData('AVP_AV2_10-5_ca');
arLoadData('AVP_AV2_10-6_ca');
```

```
arLoadData('AVP_AV2_10-7_ca');
arLoadData('AVP_AV2_10-8_ca');
arLoadData('AVP_AV2_10-9_ca');
arLoadData('AVP_AV2_10-10_ca');
arLoadData('AVP_AV2_10-11_ca');
arLoadData('AVP_AV2_basal_ca');
arLoadData('OXT_AV2_10-5_b');
arLoadData('OXT_AV2_10-6_b');
arLoadData('OXT_AV2_10-7_b');
arLoadData('OXT_AV2_10-8_b');
arLoadData('OXT_AV2_10-9_b');
arLoadData('OXT_AV2_10-10_b');
arLoadData('OXT_AV2_10-11_b');
arLoadData('OXT_AV2_basal_b');
arLoadData('OXT_AV2_10-5_ca');
arLoadData('OXT_AV2_10-6_ca');
arLoadData('OXT_AV2_10-7_ca');
arLoadData('OXT_AV2_10-8_ca');
arLoadData('OXT_AV2_10-9_ca');
arLoadData('OXT_AV2_10-10_ca');
arLoadData('OXT_AV2_10-11_ca');
arLoadData('OXT_AV2_basal_ca');

arCompileAll();

arFitLHS(100);
```

Annexe D

Figure

D.1 Dose réponse

Programmation : (cf.Annexe C.2.1 page 100)(Retour au rapport :4)

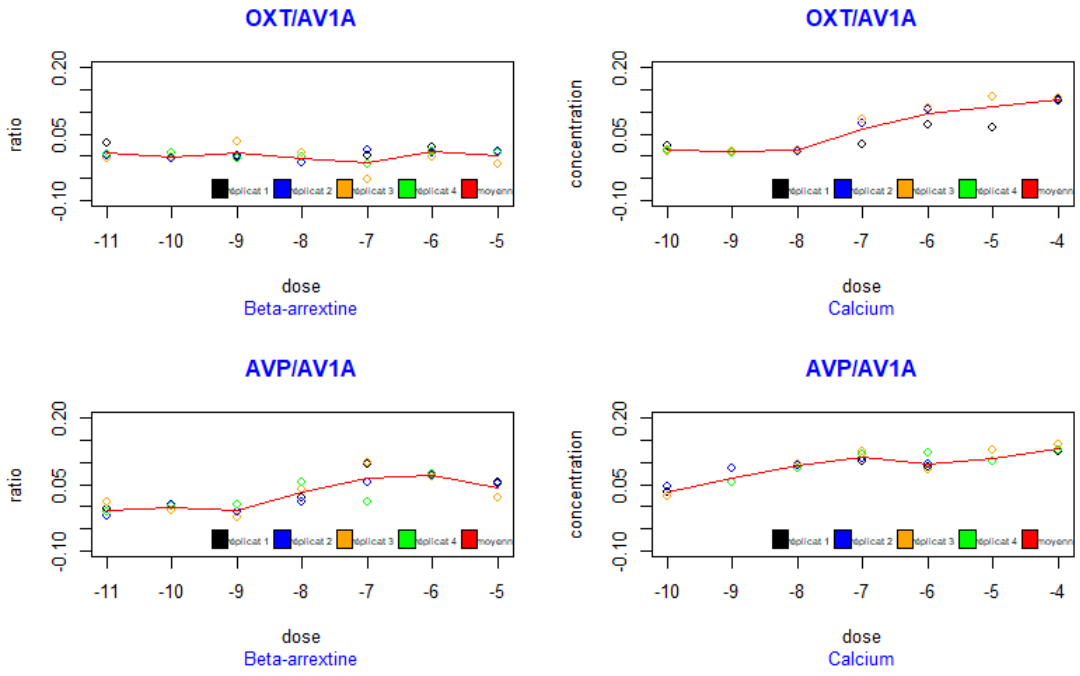

FIGURE D.1 : Courbe dose réponse du récepteur AV1A

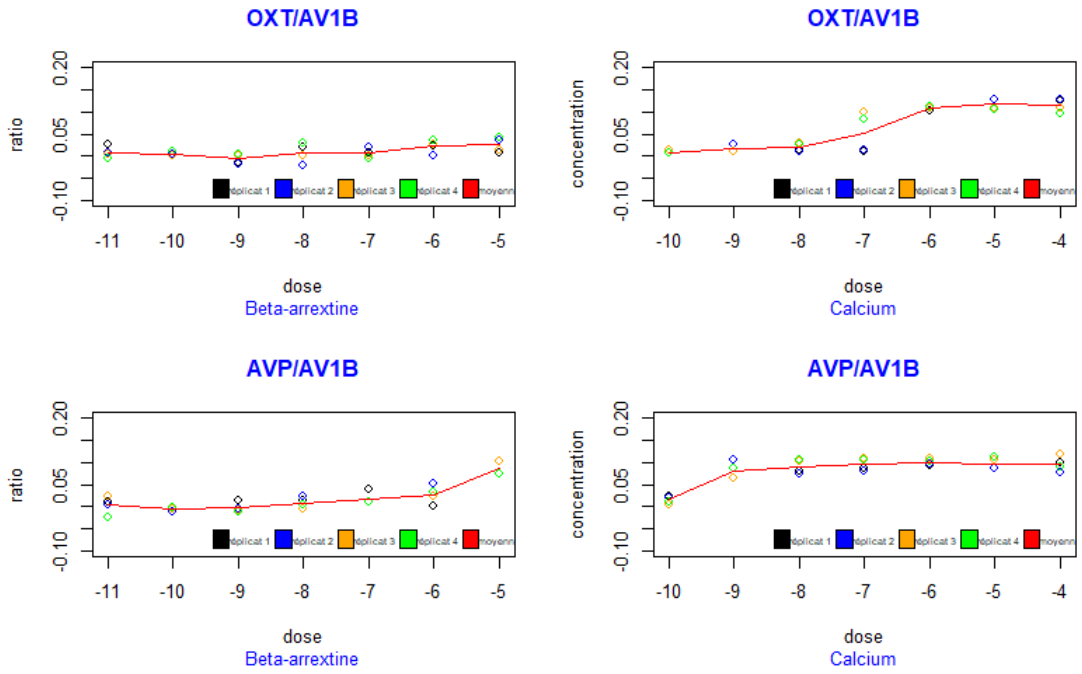

FIGURE D.2 : Courbe dose réponse du récepteur AV1B

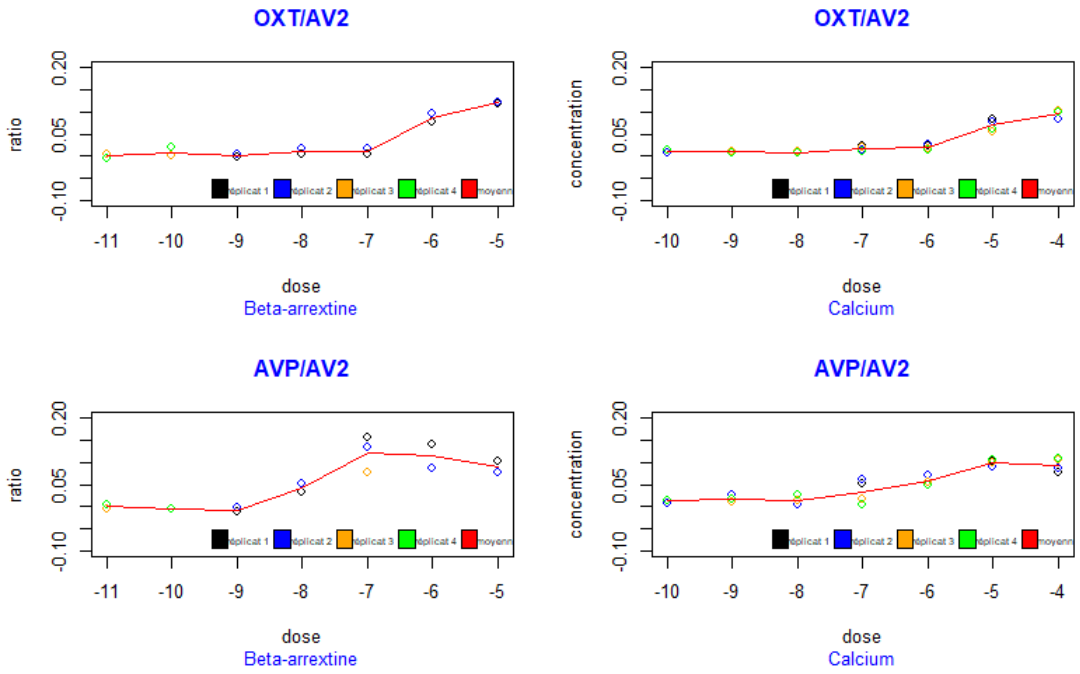

FIGURE D.3 : Courbe dose réponse du récepteur AV2

D.2 Courbe cinétique

Programmation : (cf.Annexe C.2.2 page 103). Retour au rapport : (cf.Annexe 4 page 20)

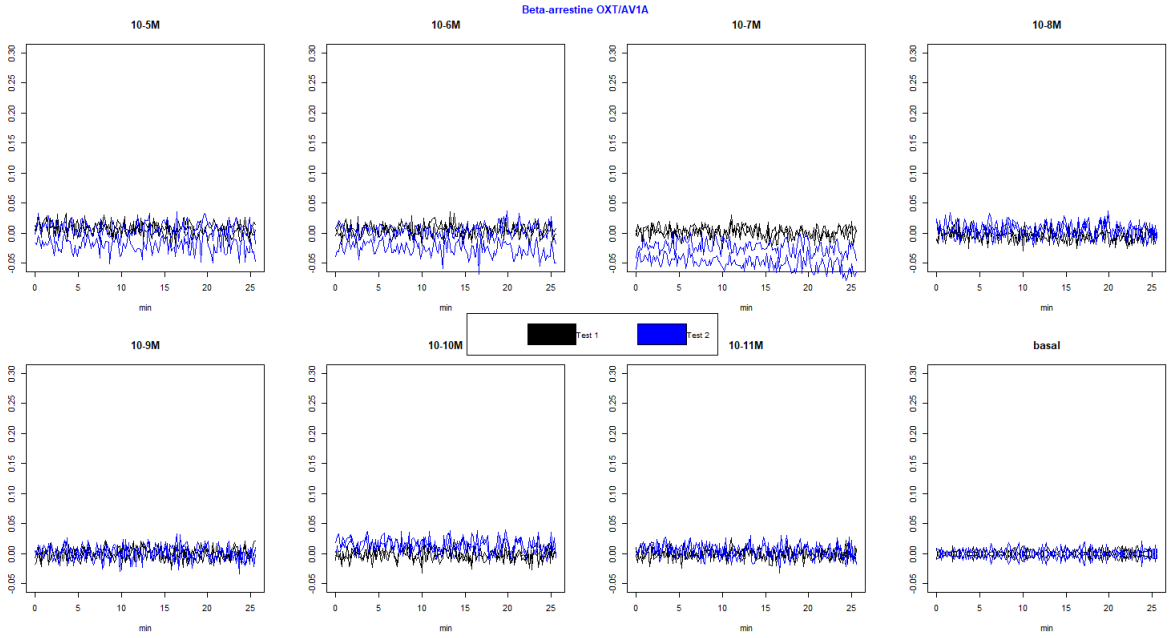

FIGURE D.4 : Courbe cinétique beta-arrestine du couple OXT/AV1A

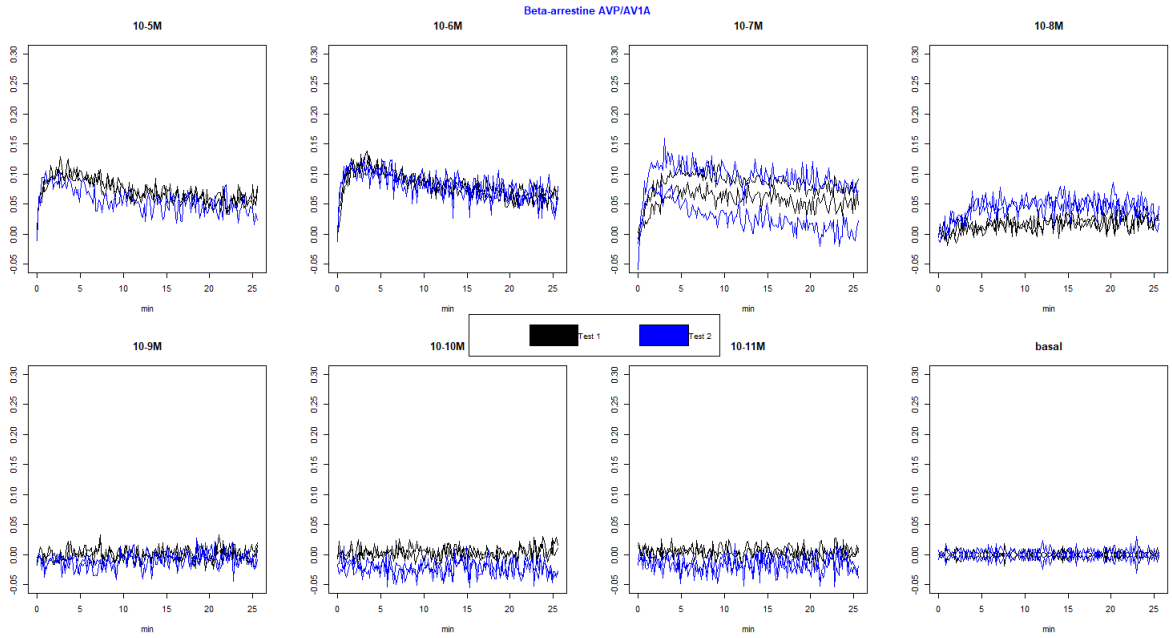

FIGURE D.5 : Courbe cinétique beta-arrestine du couple AVP/AV1A

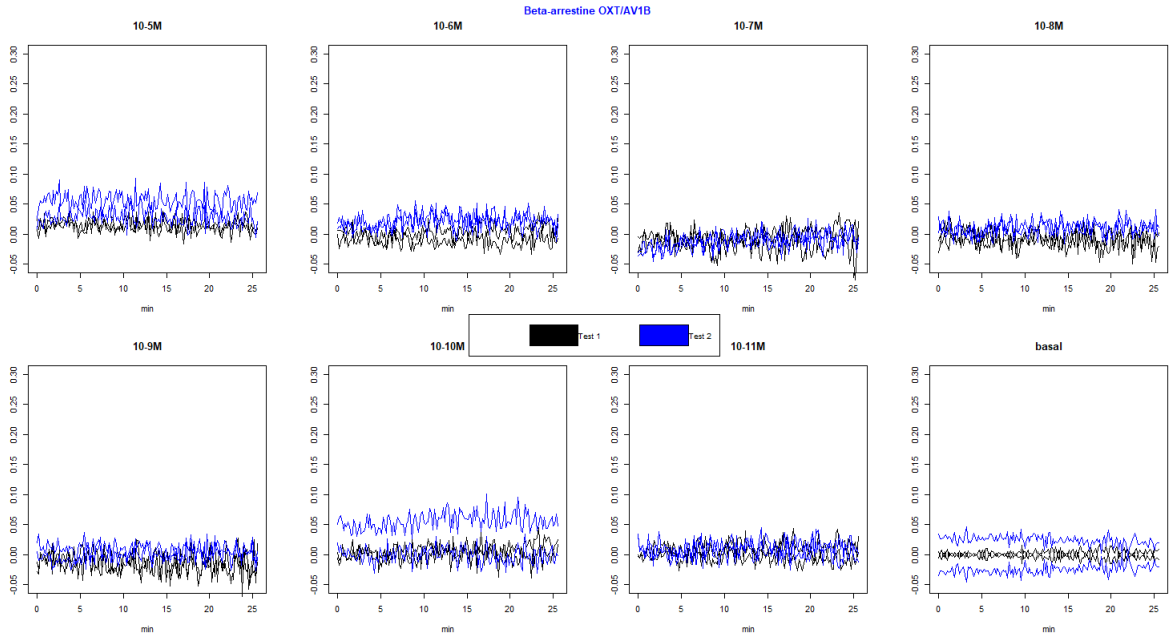

FIGURE D.6 : Courbe cinétique beta-arrestine du couple OXT/AV1B

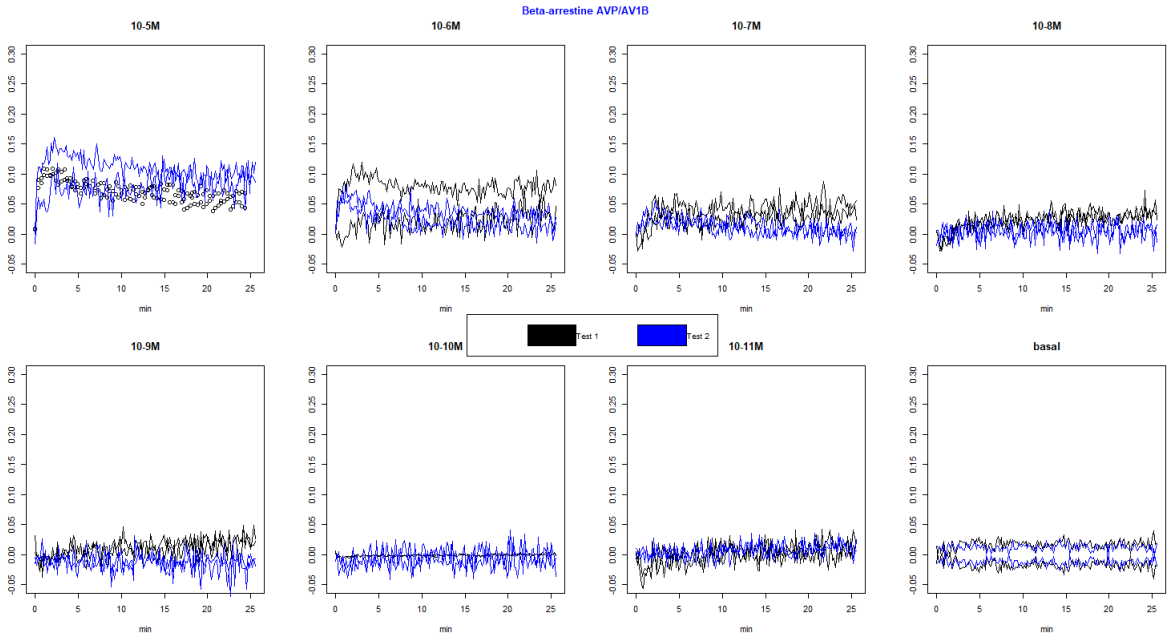

FIGURE D.7 : Courbe cinétique beta-arrestine du couple AVP/AV1B

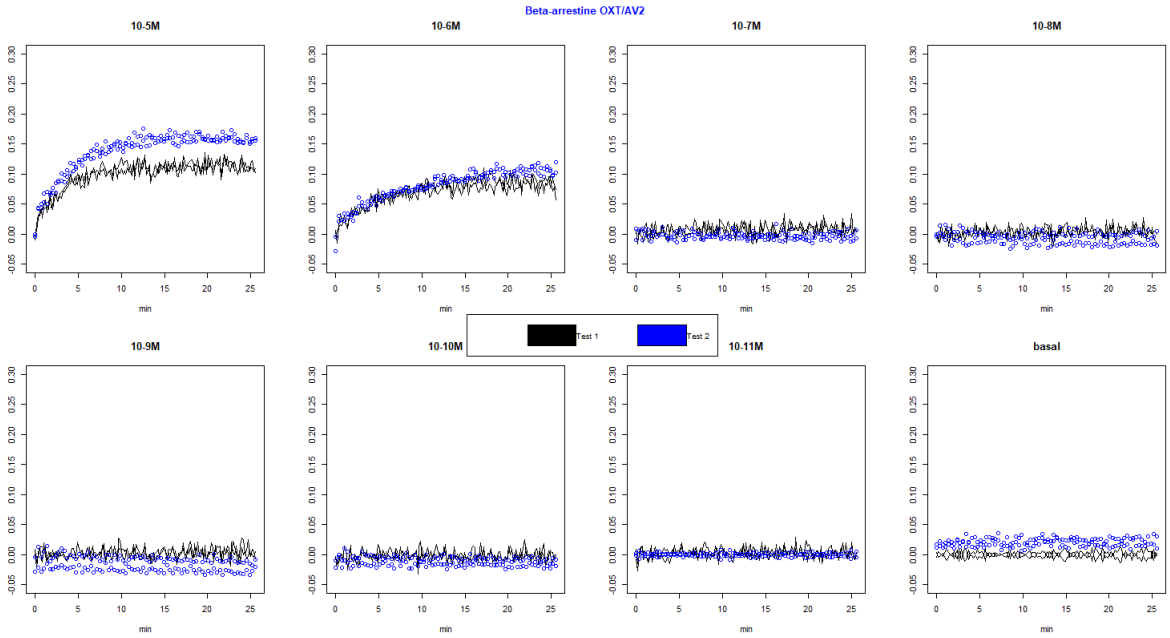

FIGURE D.8 : Courbe cinétique beta-arrestine du couple OXT/AV2

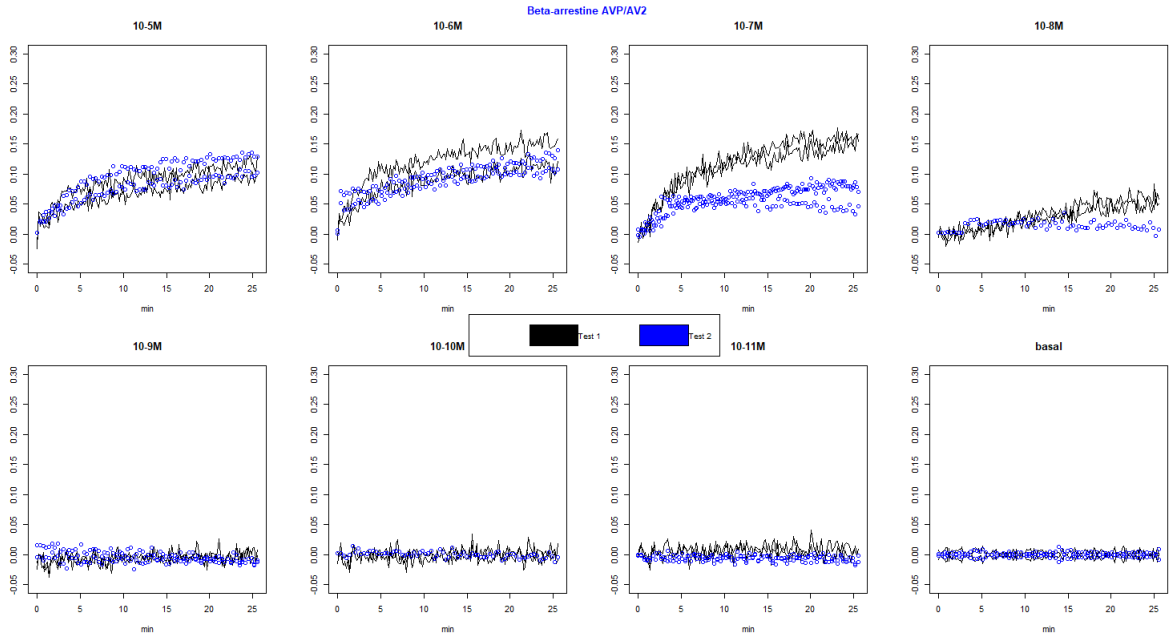

FIGURE D.9 : Courbe cinétique beta-arrestine du couple AVP/AV2

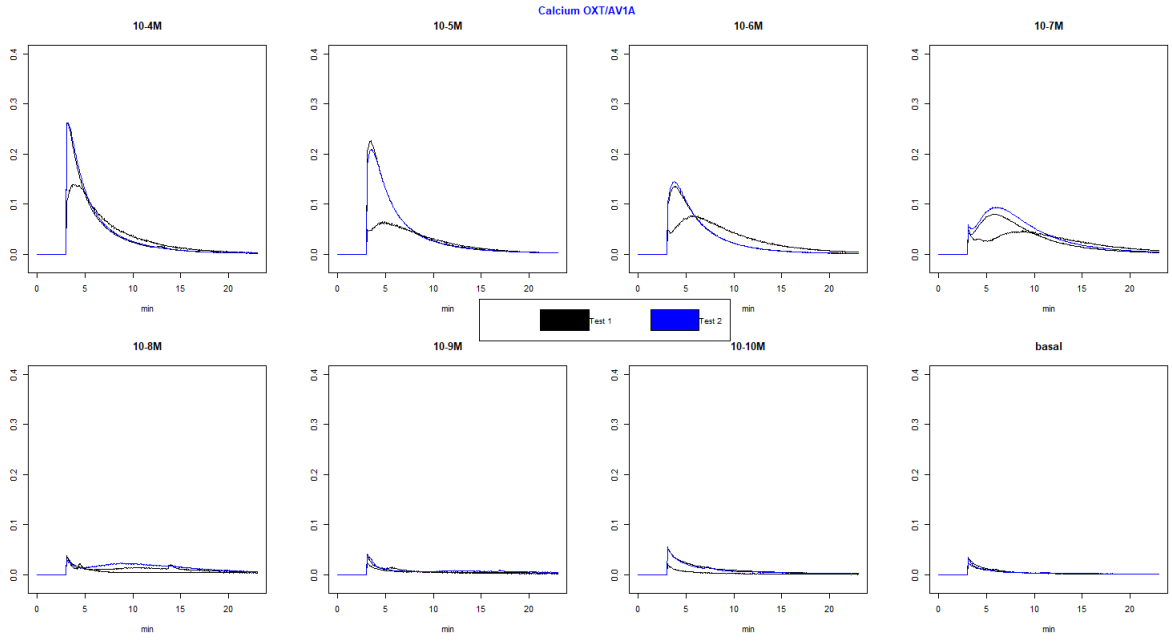

FIGURE D.10 : Courbe cinétique calcium du couple OXT/AV1A

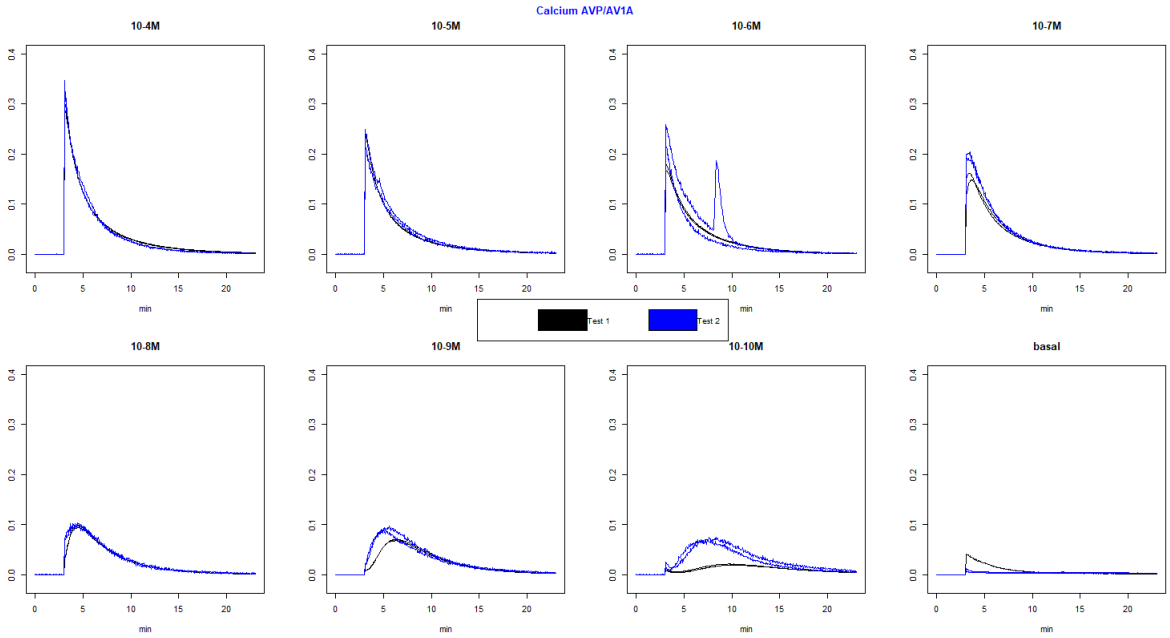

FIGURE D.11 : Courbe cinétique calcium du couple AVP/AV1A

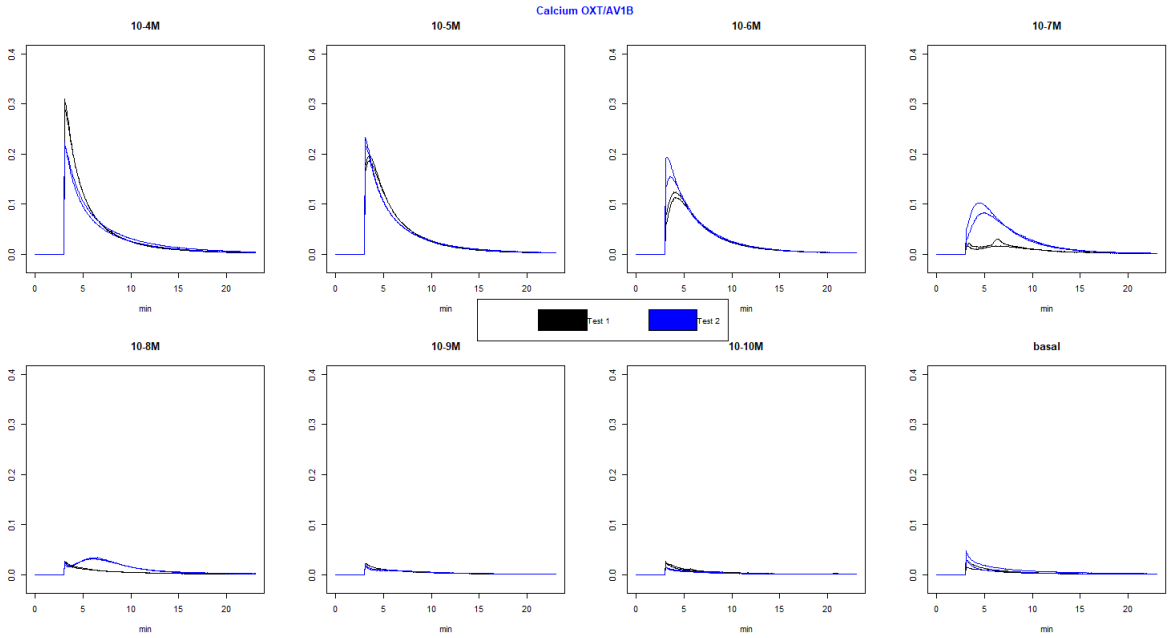

FIGURE D.12 : Courbe cinétique calcium du couple OXT/AV1B

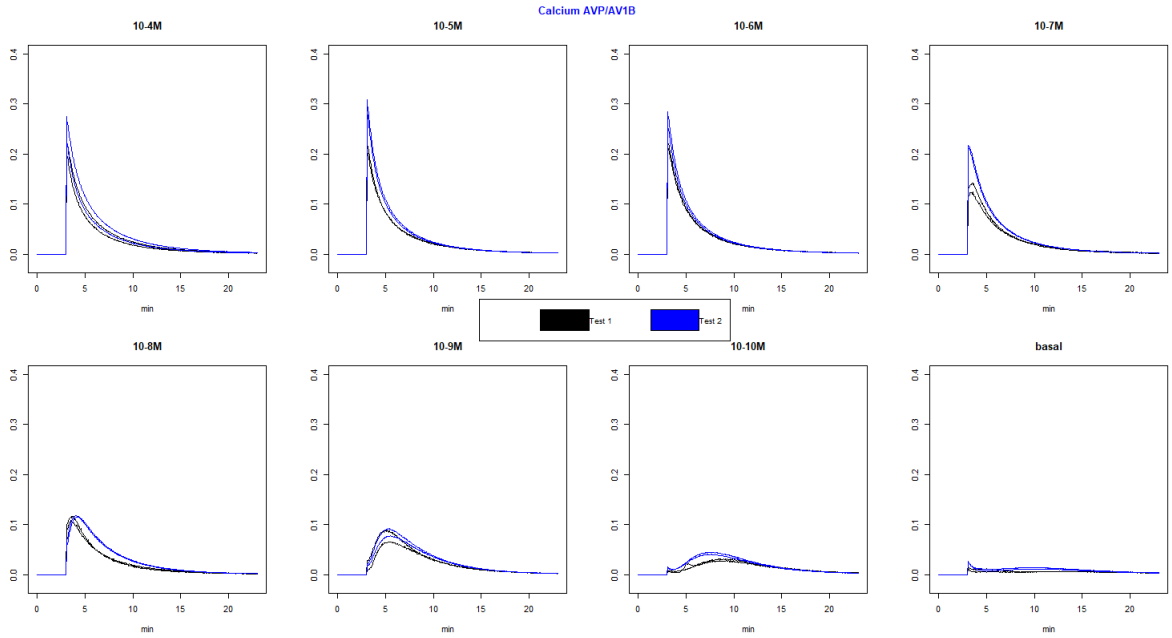

FIGURE D.13 : Courbe cinétique calcium du couple AVP/AV1B

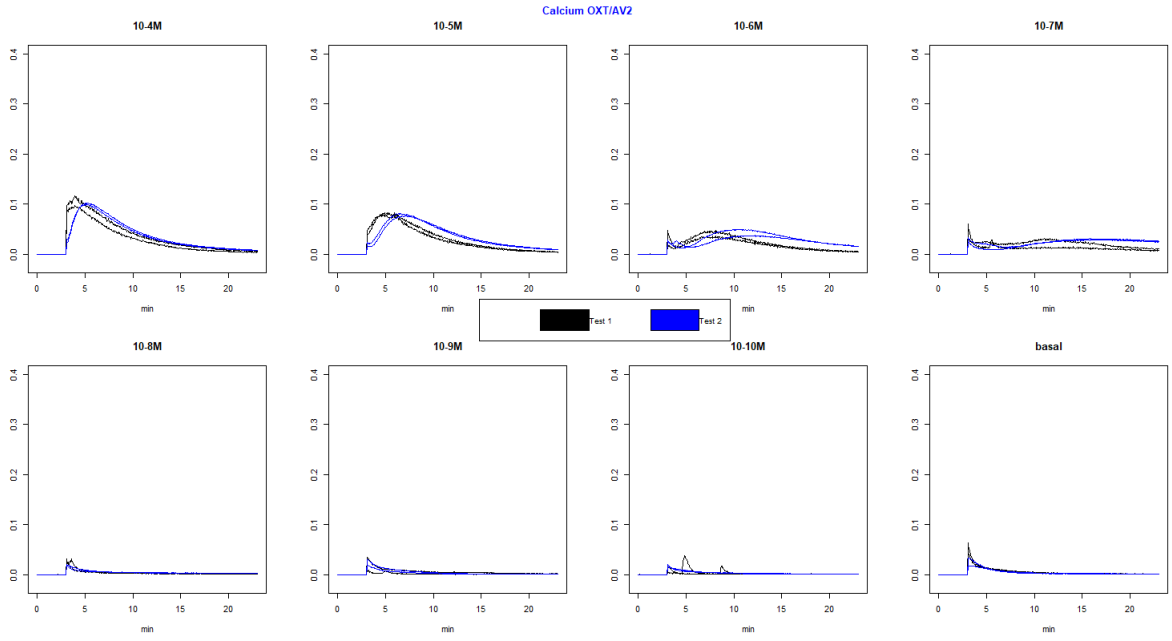

FIGURE D.14 : Courbe cinétique calcium du couple OXT/AV2

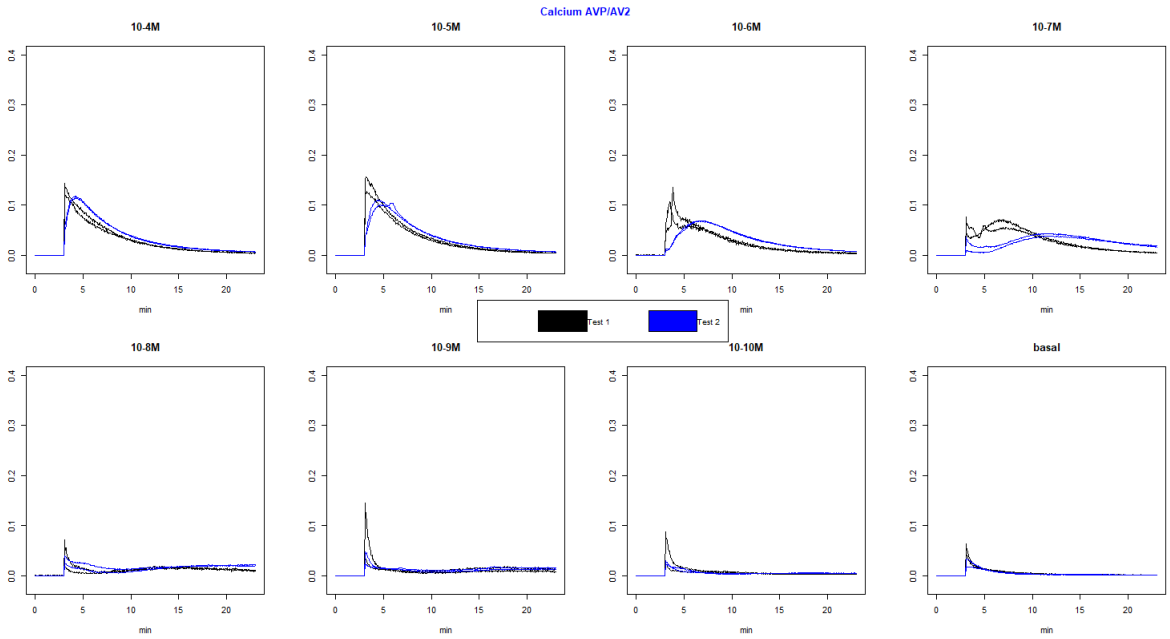

FIGURE D.15 : Courbe cinétique calcium du couple AVP/AV2

D.3 Donées brutes

beta-arrestine

Programme : (cf.Annexe C.2.3 page 107). Retour au rapport : (cf.Annexe A.1.1 page 33)

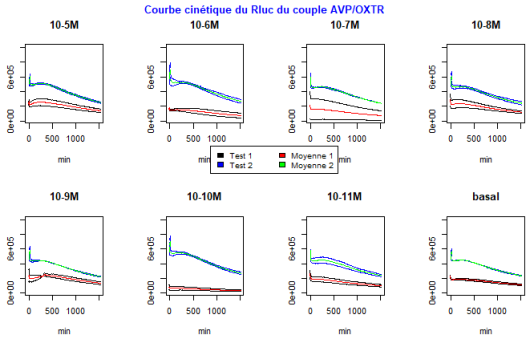

FIGURE D.16 : Courbe cinétique Rluc du couple AVP/OXTR

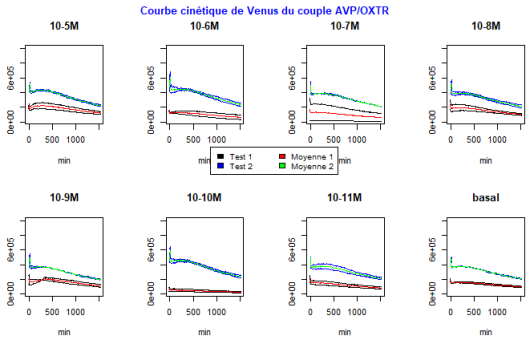

FIGURE D.17 : Courbe cinétique Venus du couple AVP/OXTR

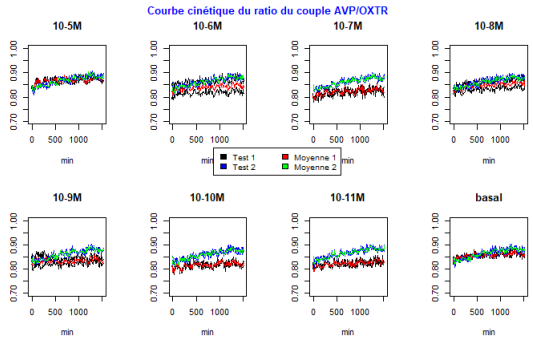

FIGURE D.18 : Courbe cinétique du ratio du couple AVP/OXTR

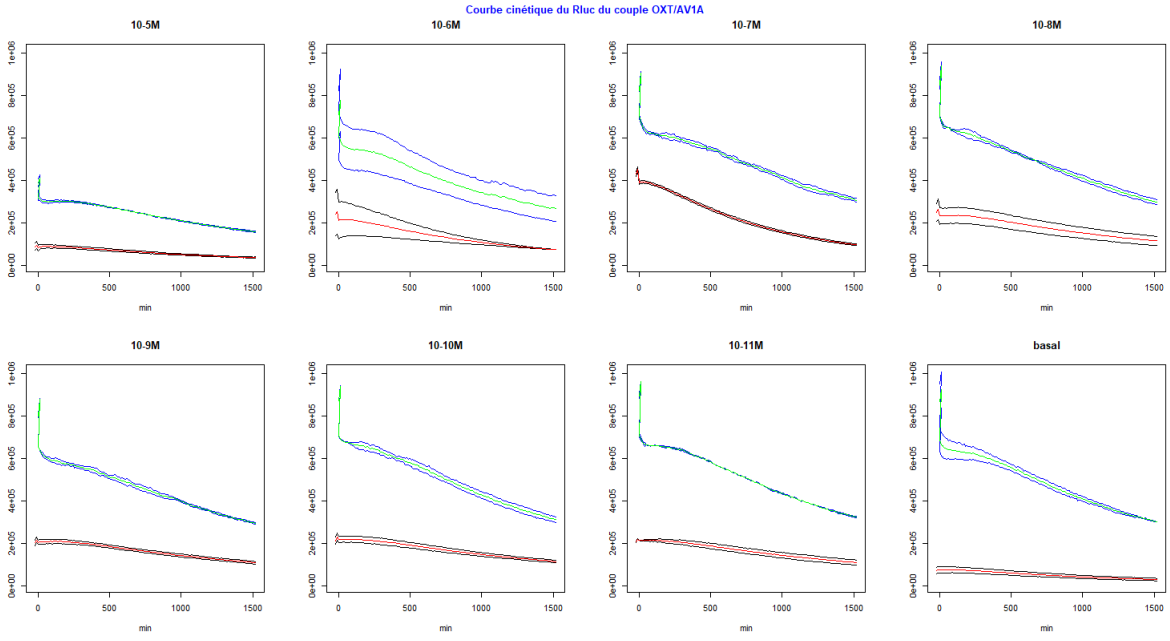

FIGURE D.19 : Courbe cinétique Rluc du couple OXT/AV1A

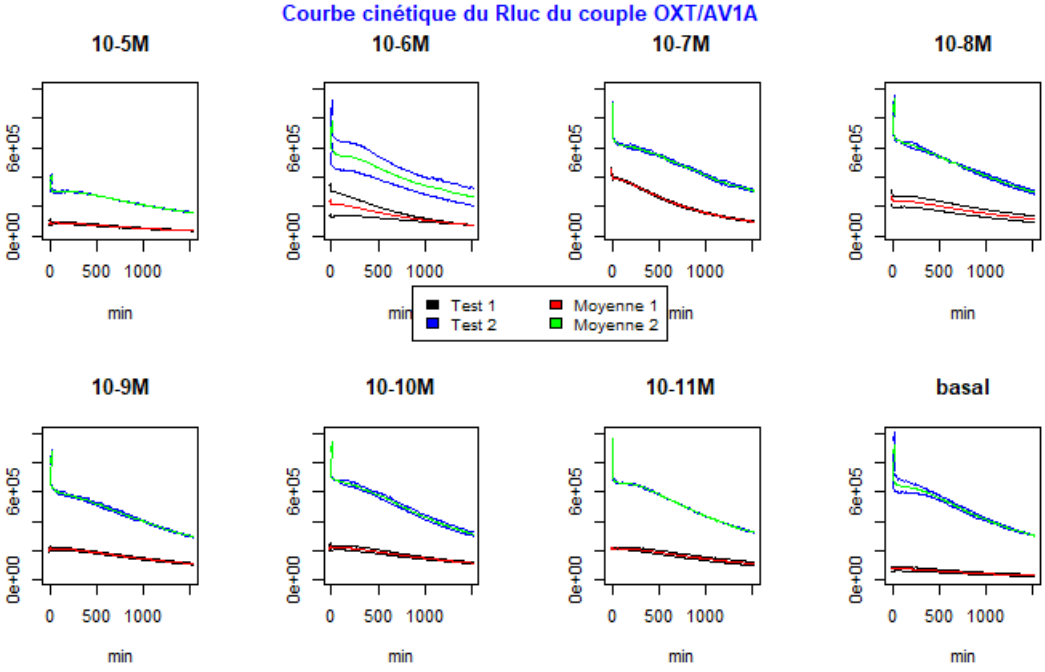

FIGURE D.20 : Courbe cinétique Venus du couple OXT/AV1A

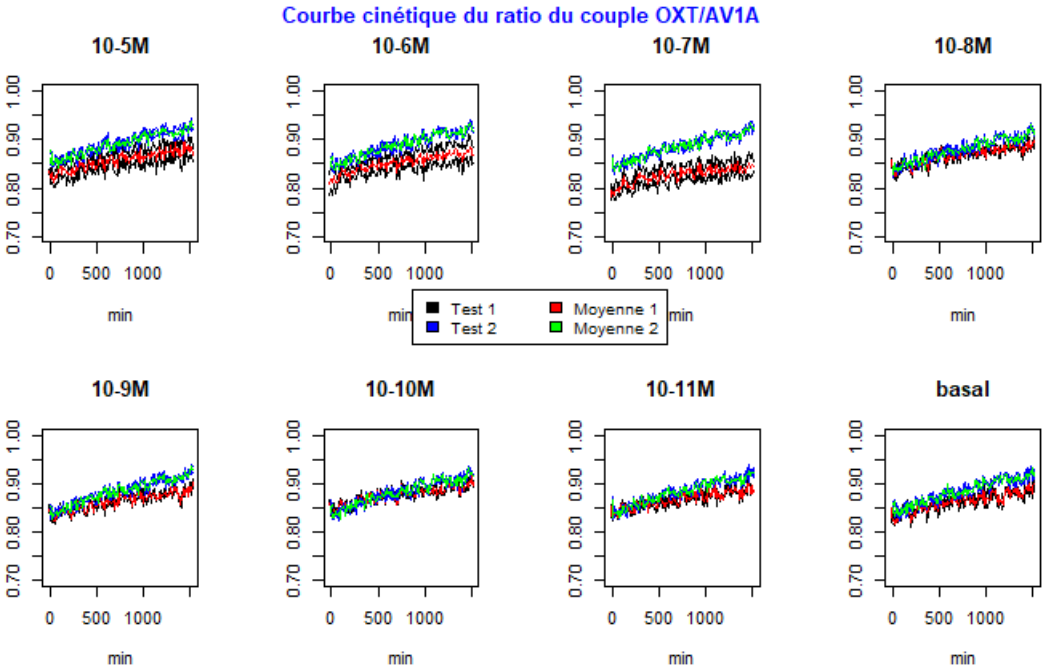

FIGURE D.21 : Courbe cinétique du ratio du couple OXT/AV1A

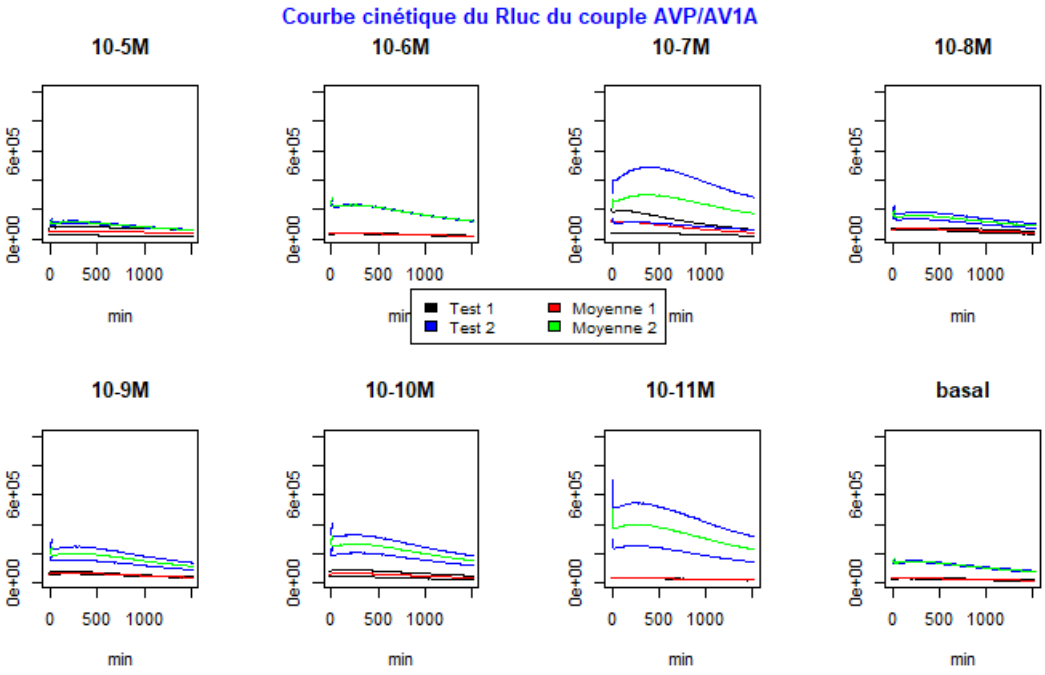

FIGURE D.22 : Courbe cinétique Rluc du couple AVP/AV1A

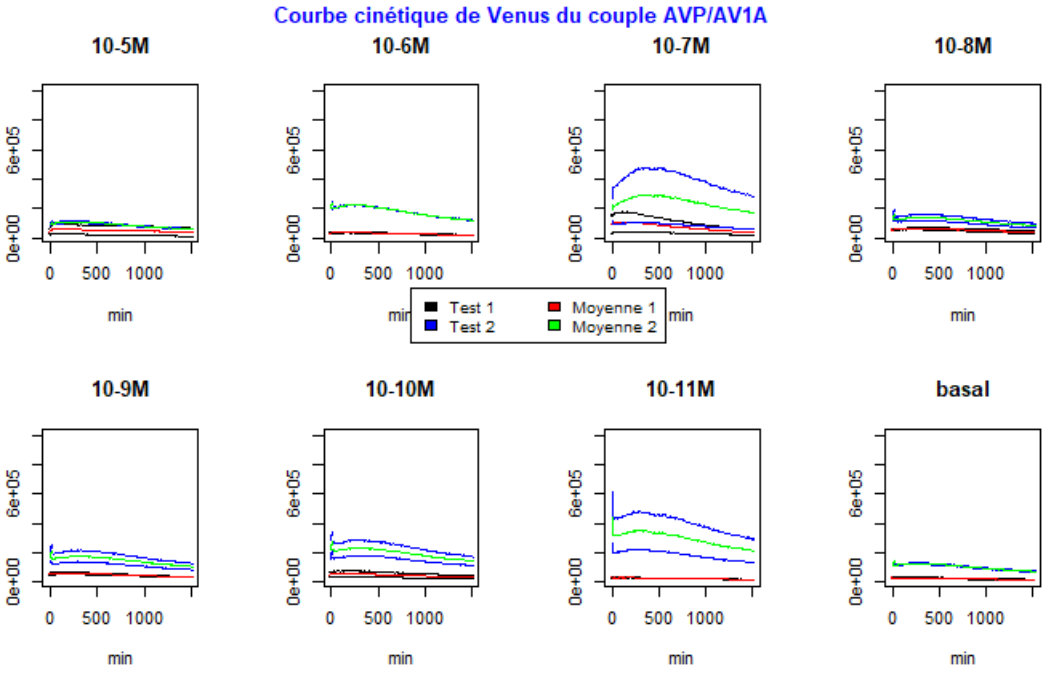

FIGURE D.23 : Courbe cinétique Venus du couple AVP/AV1A

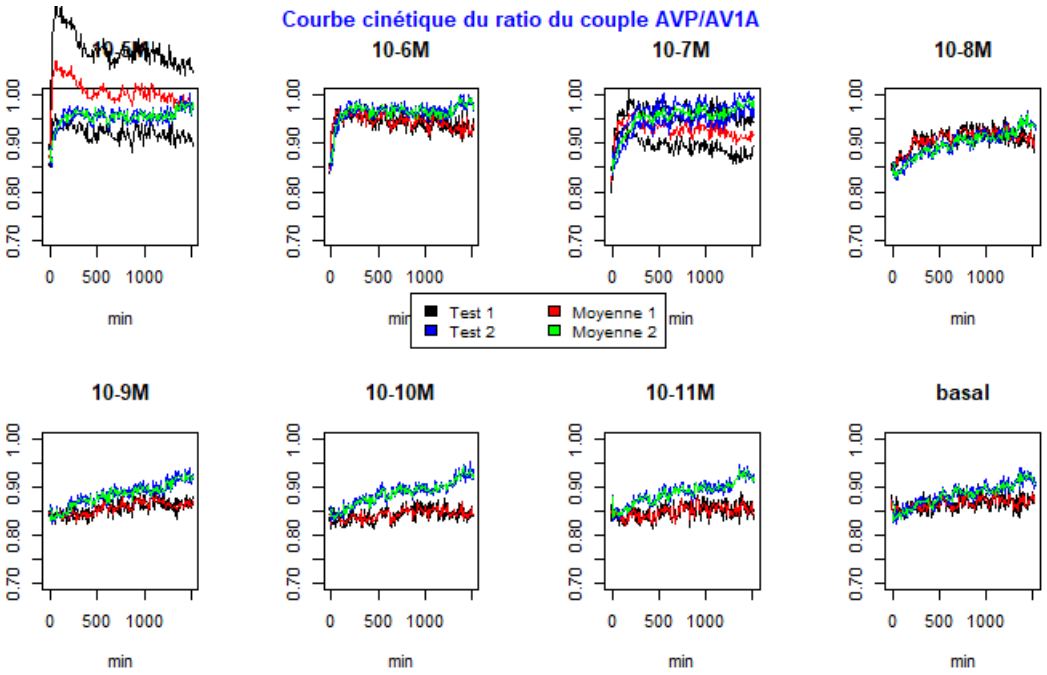

FIGURE D.24 : Courbe cinétique du ratio du couple AVP/AV1A

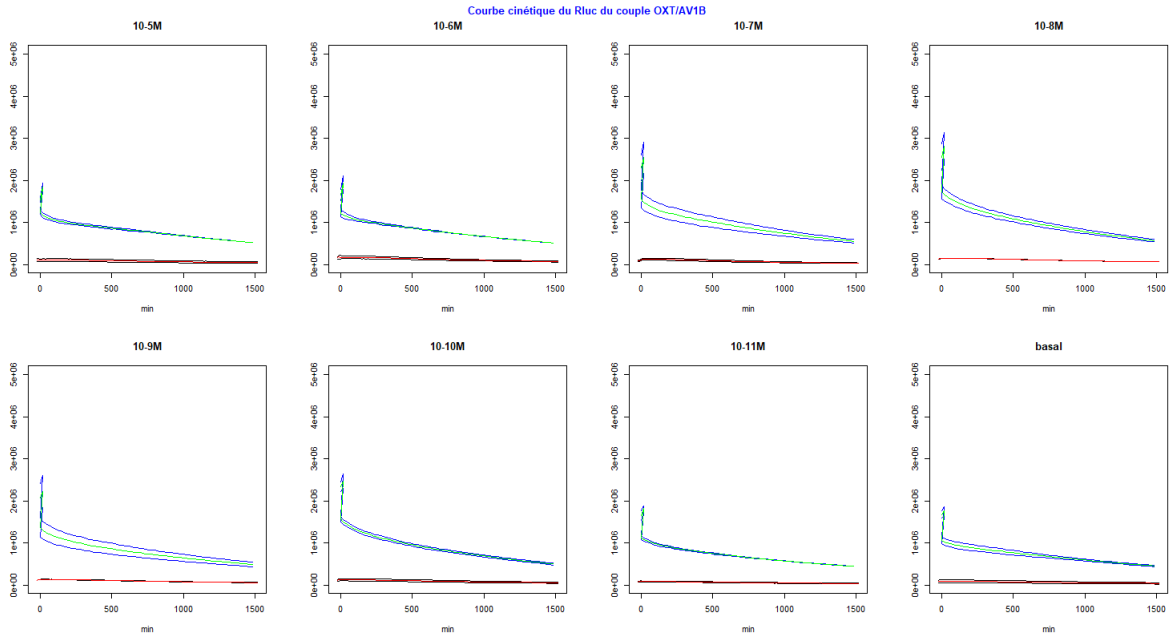

FIGURE D.25 : Courbe cinétique Rluc du couple OXT/AV1B

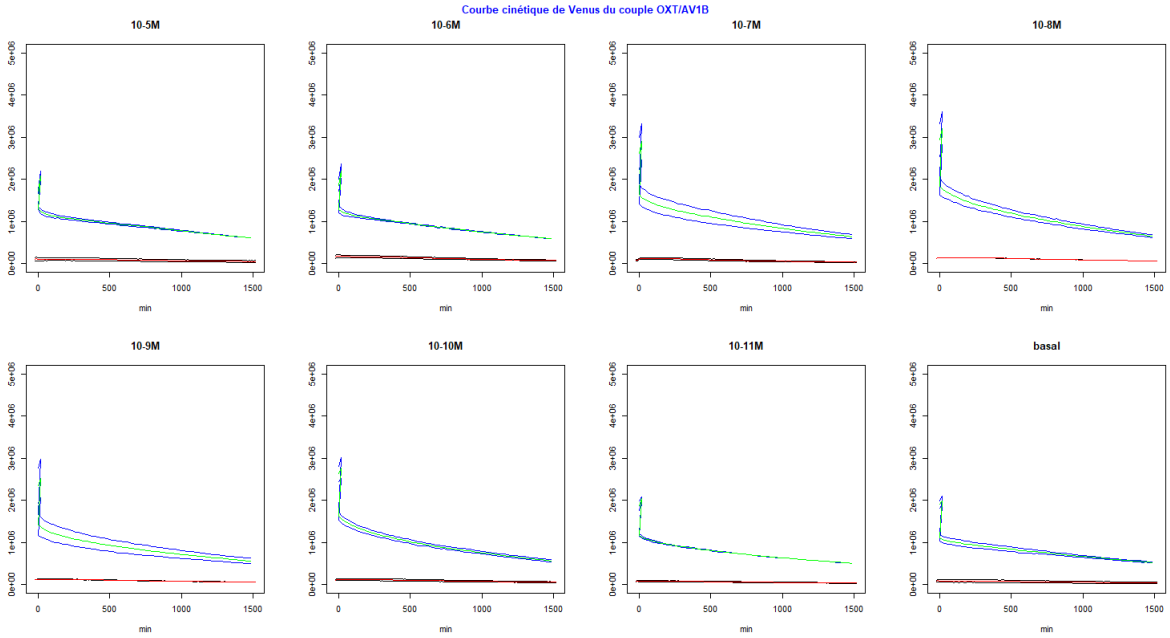

FIGURE D.26 : Courbe cinétique Venus du couple OXT/AV1B

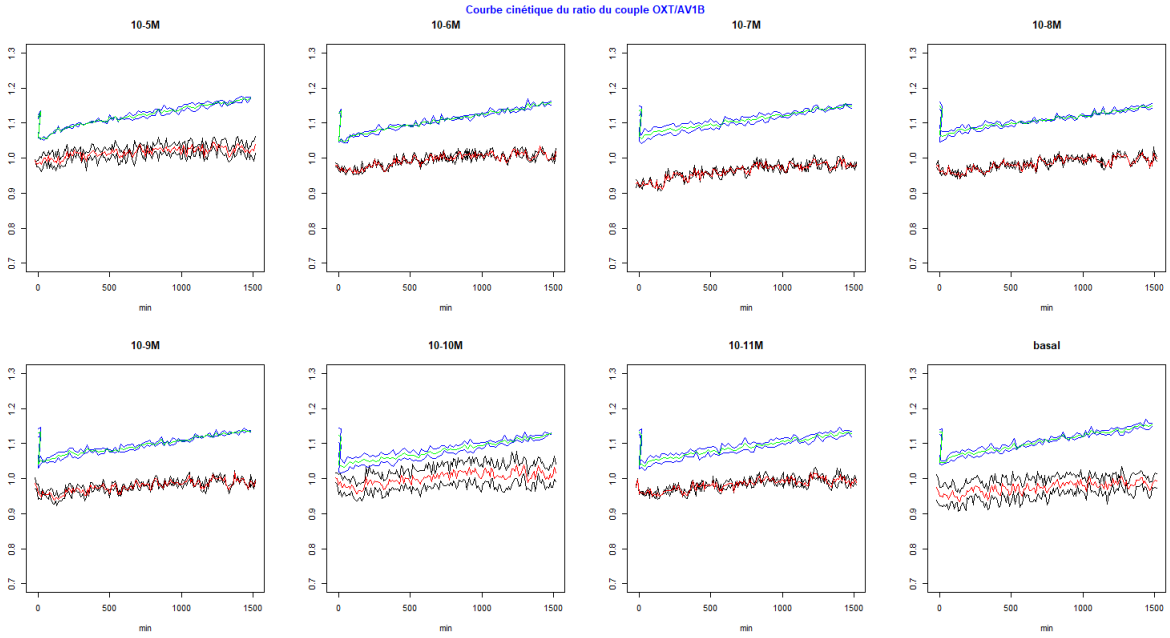

FIGURE D.27 : Courbe cinétique du ratio du couple OXT/AV1B

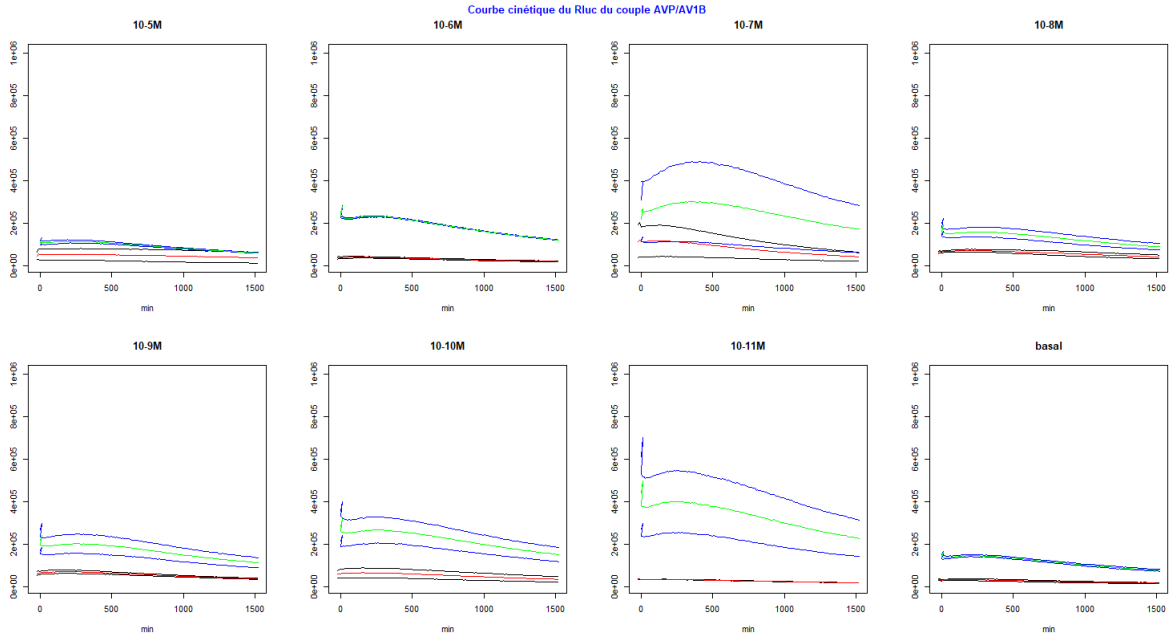

FIGURE D.28 : Courbe cinétique Rluc du couple AVP/AV1B

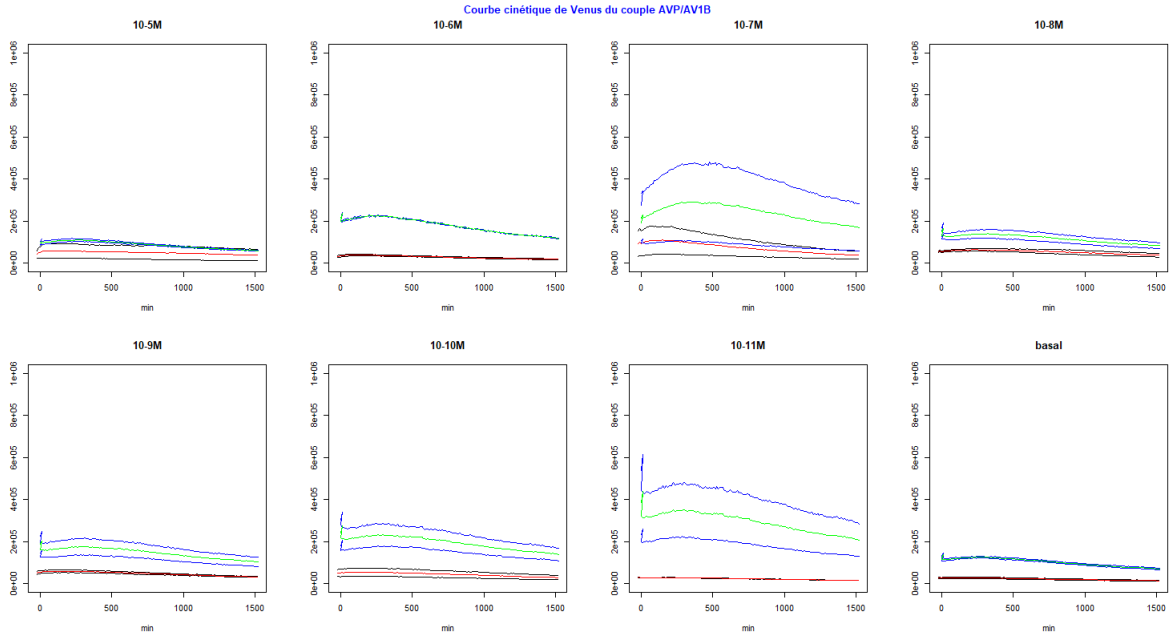

FIGURE D.29 : Courbe cinétique Venus du couple AVP/AV1B

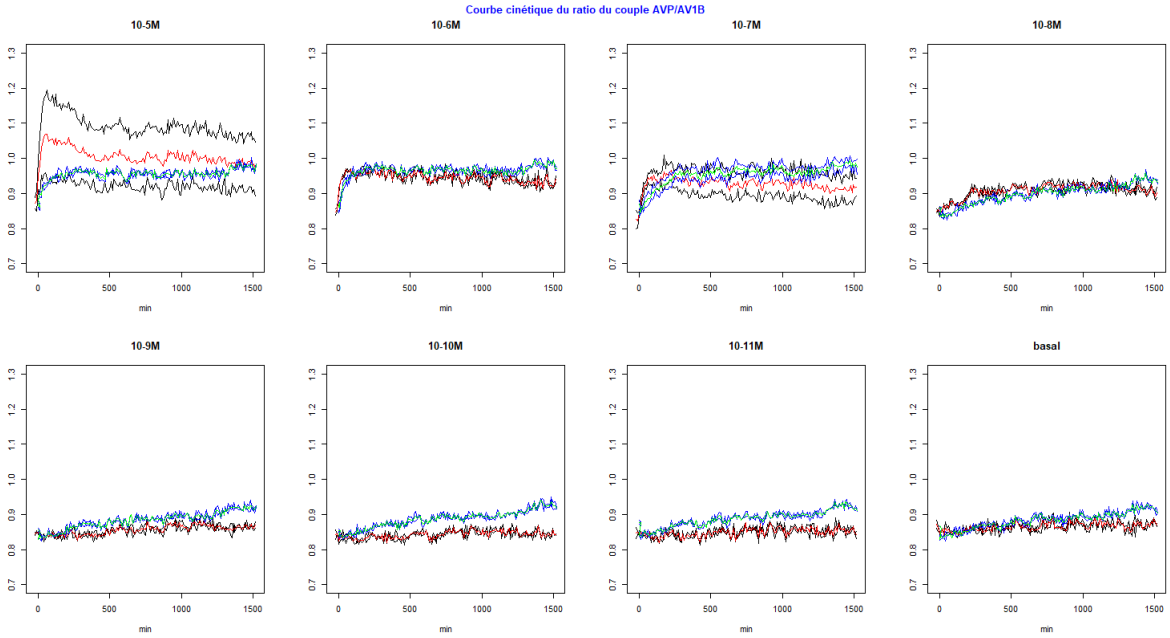

FIGURE D.30 : Courbe cinétique du ratio du couple AVP/AV1B

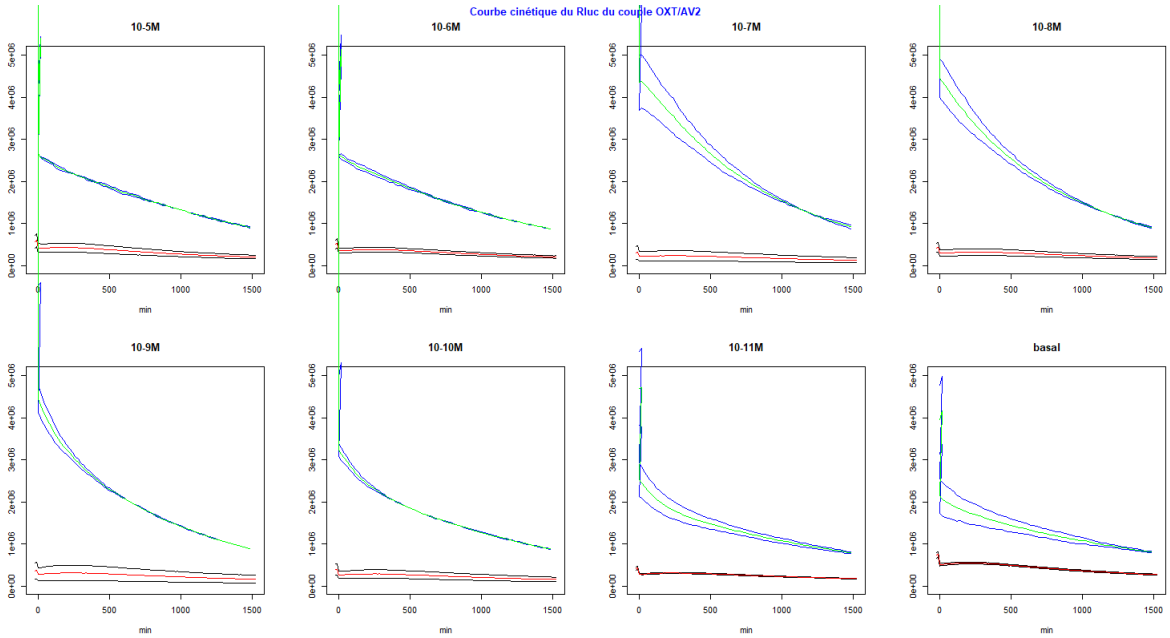

FIGURE D.31 : Courbe cinétique Rluc du couple OXT/AV2

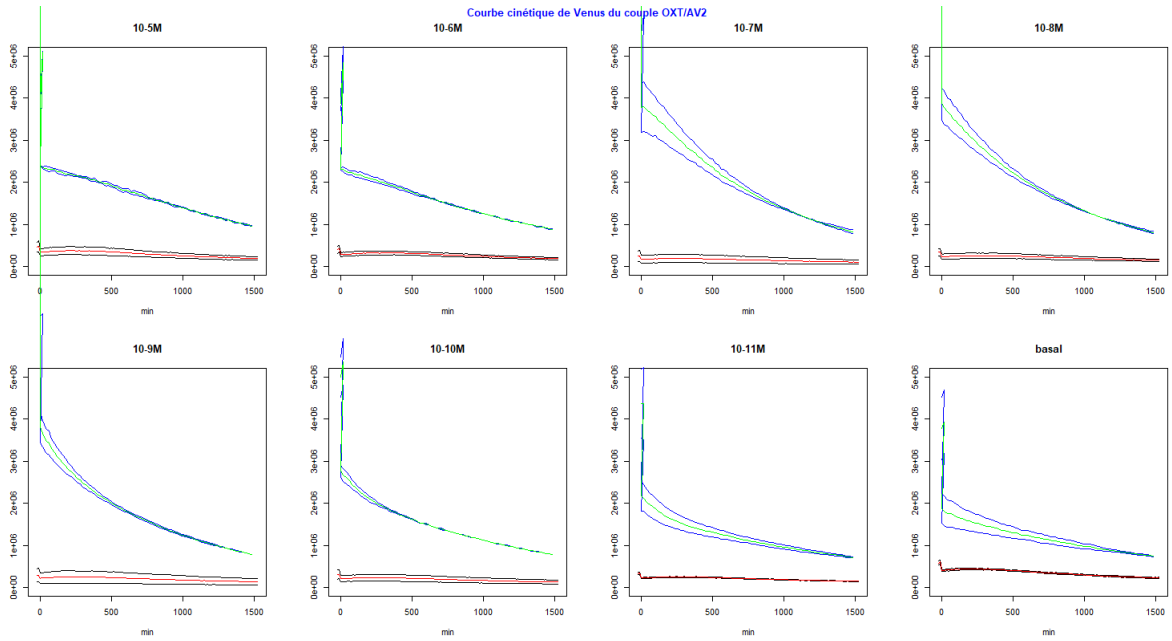

FIGURE D.32 : Courbe cinétique Venus du couple OXT/AV2

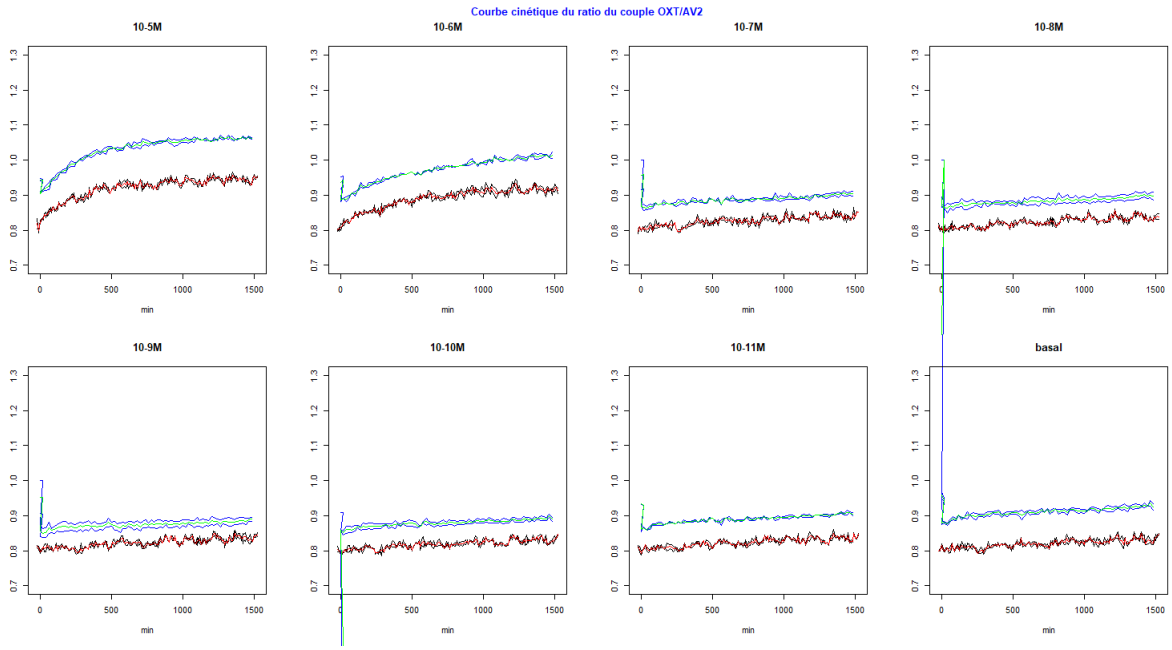

FIGURE D.33 : Courbe cinétique du ratio du couple OXT/AV2

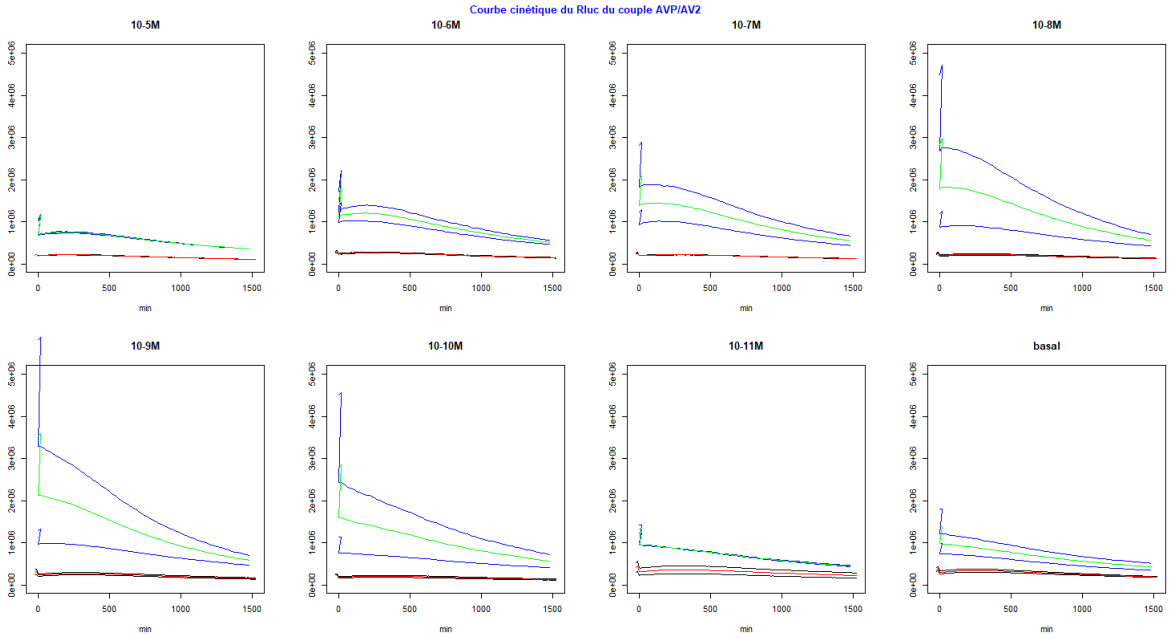

FIGURE D.34 : Courbe cinétique Rluc du couple AVP/AV2

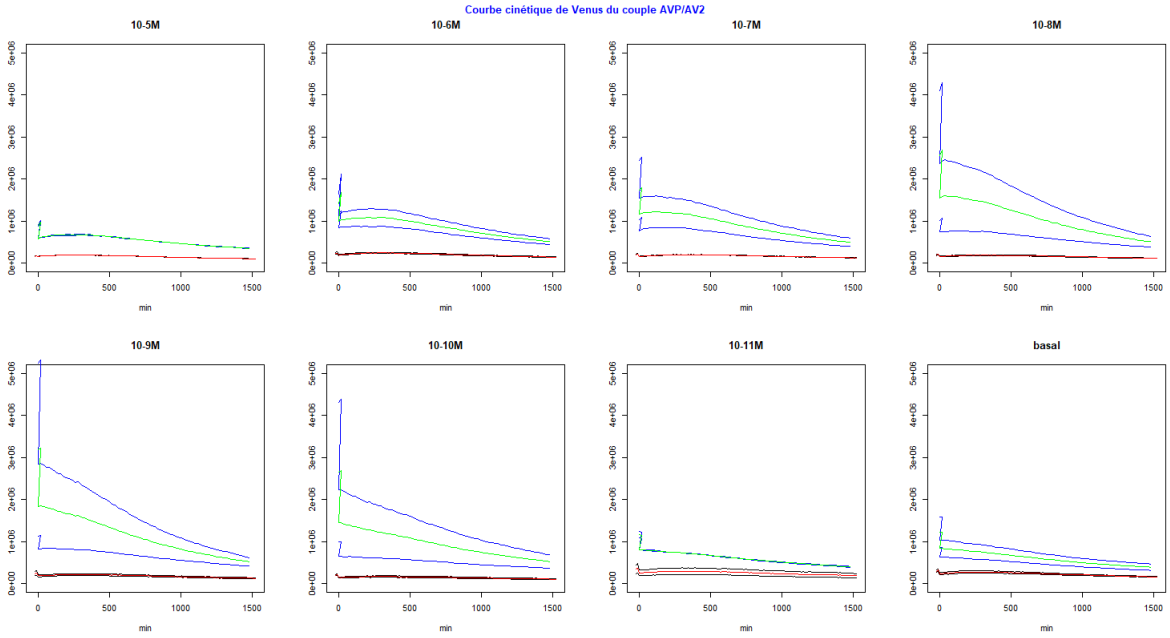

FIGURE D.35 : Courbe cinétique Venus du couple AVP/AV2

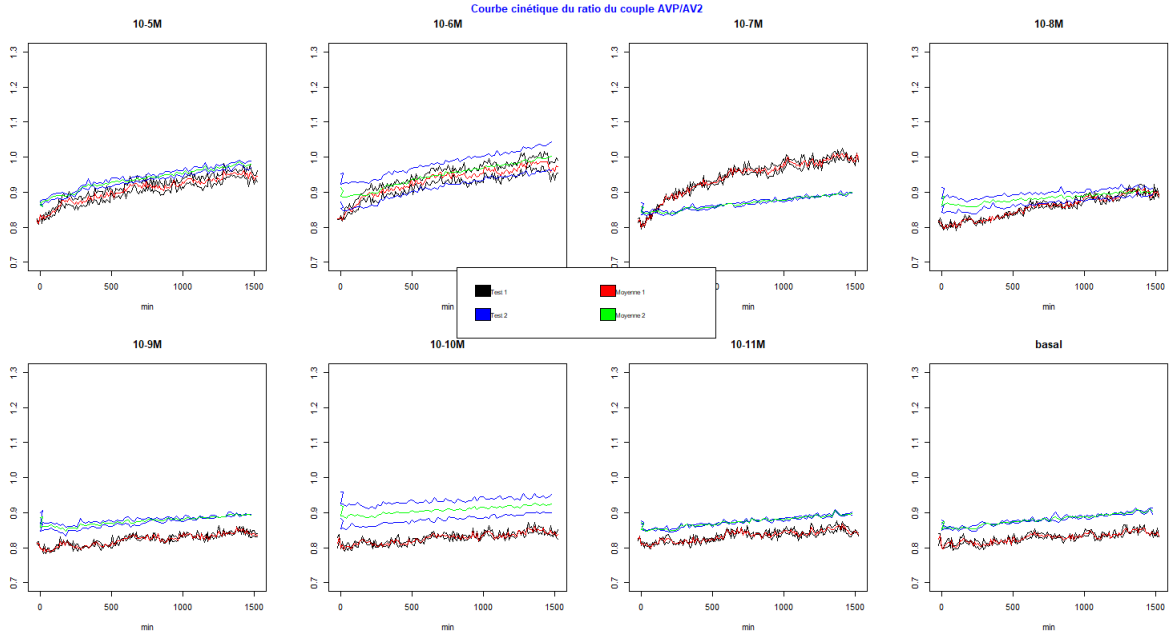

FIGURE D.36 : Courbe cinétique du ratio du couple AVP/AV2

Calcium

Programme : (cf.Annexe C.2.3 page 111). Retour au rapport : (cf.Annexe A.1.2 page 36)

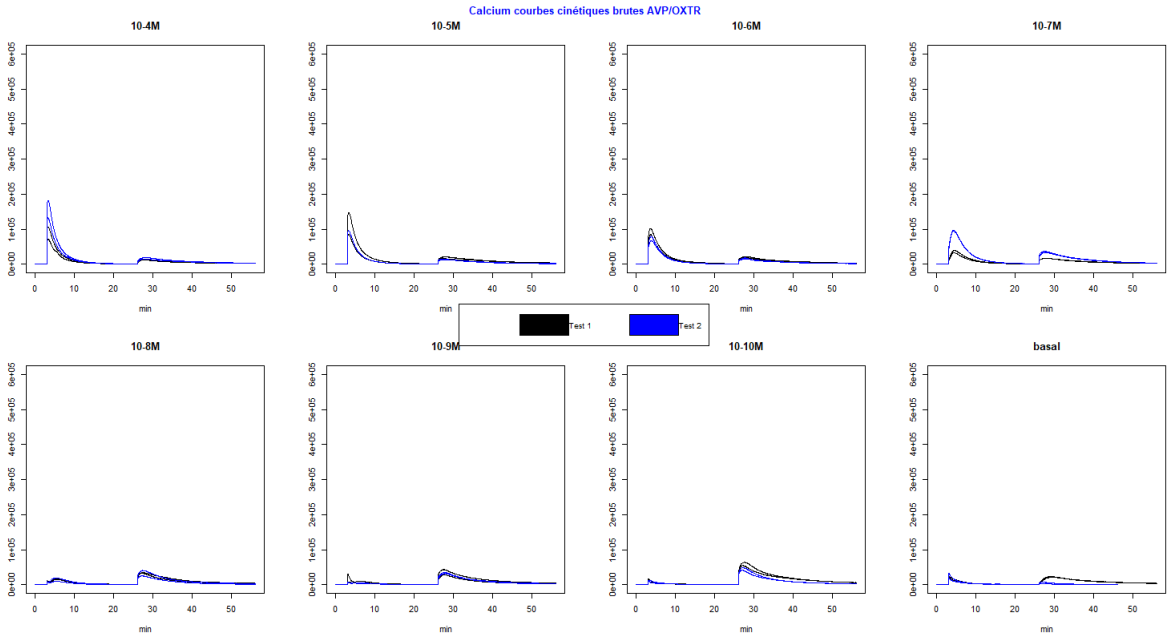

FIGURE D.37 : Courbe cinétique des données calcium brut du couple AVP/OXTR

D.4 Test de normalité

D.4.1 beta-arrestine

(Retour au rapport : 5.2.1.Beta-arrestine, programme :(cf.Annexe C.3.2 page 117))

D.4.2 Calcium

(Retour au rapport : 5.2.2.Calcium, programme :(cf.Annexe C.3.3 page 118))

D.4.3 beta-arrestine brute

(Retour au rapport : (cf.Annexe **??** page **??**), programme :(cf.Annexe **??** page **??**))

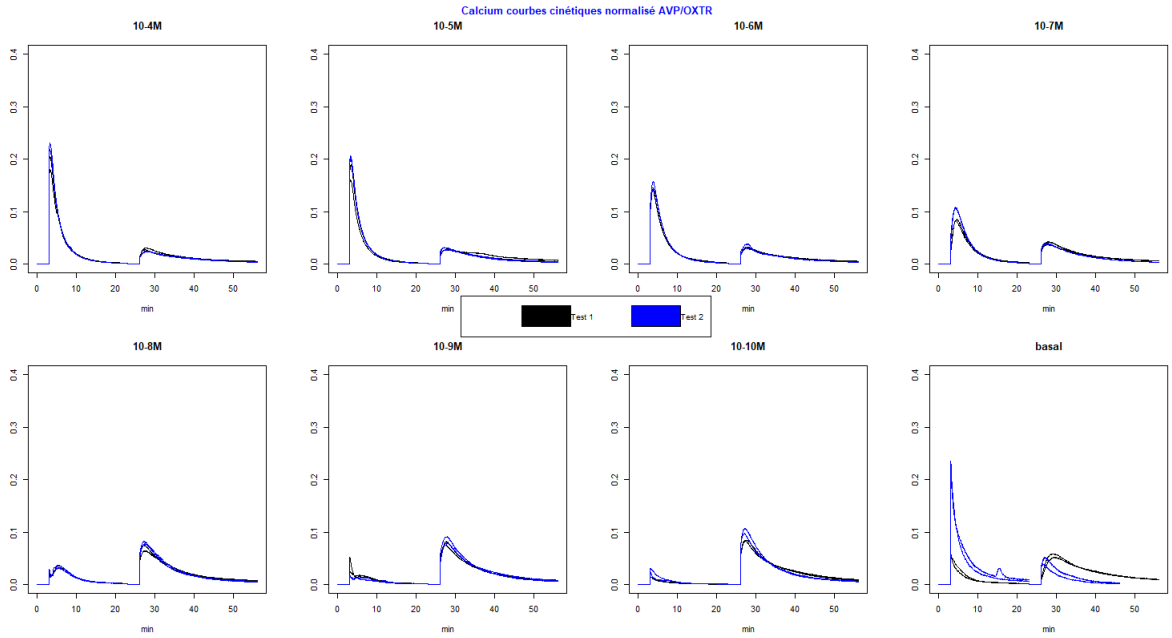

FIGURE D.38 : Courbe cinétique des données calcium normalisée du couple AVP/OXTR

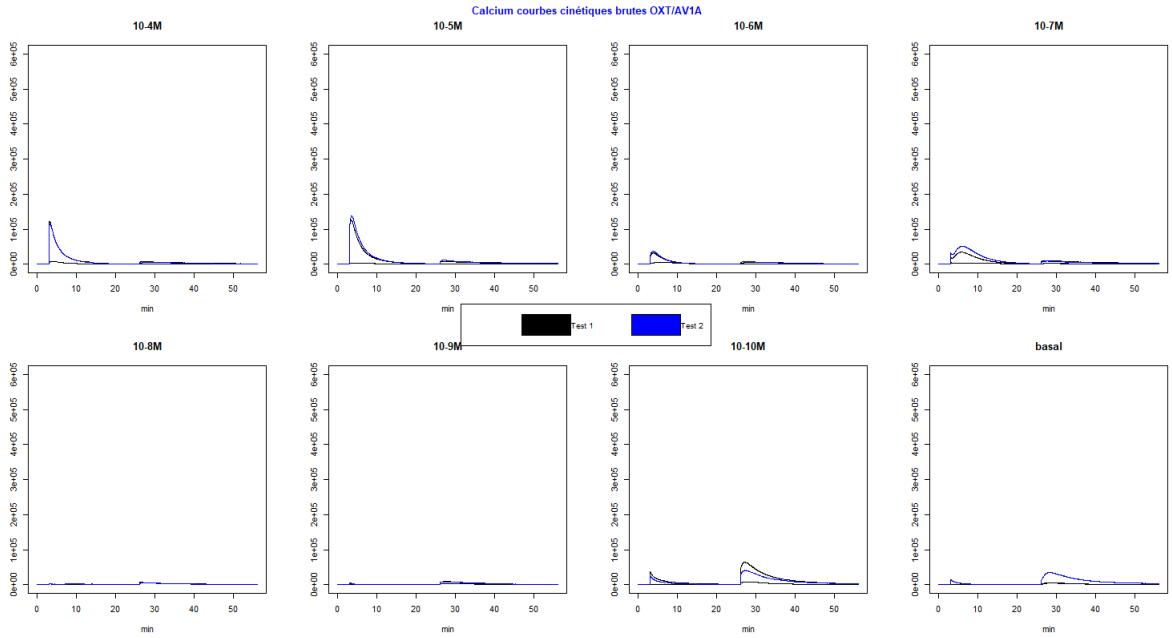

FIGURE D.39 : Courbe cinétique des données calcium brutes du couple OXT/AV1A

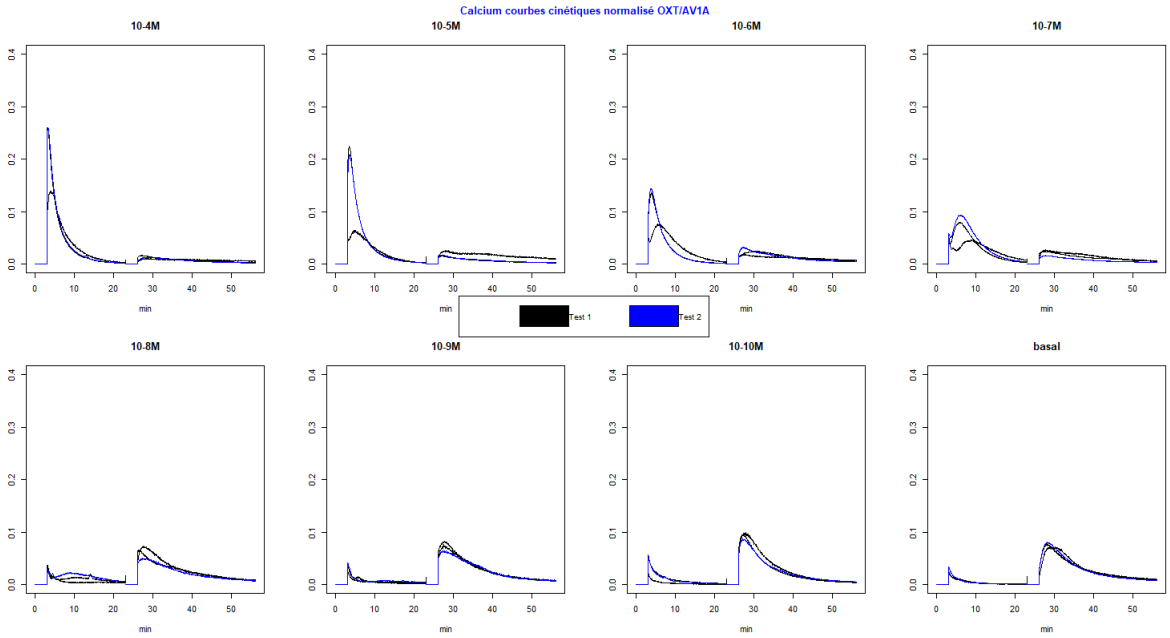

FIGURE D.40 : Courbe cinétique des données calcium normalisées du couple OXT/AV1A

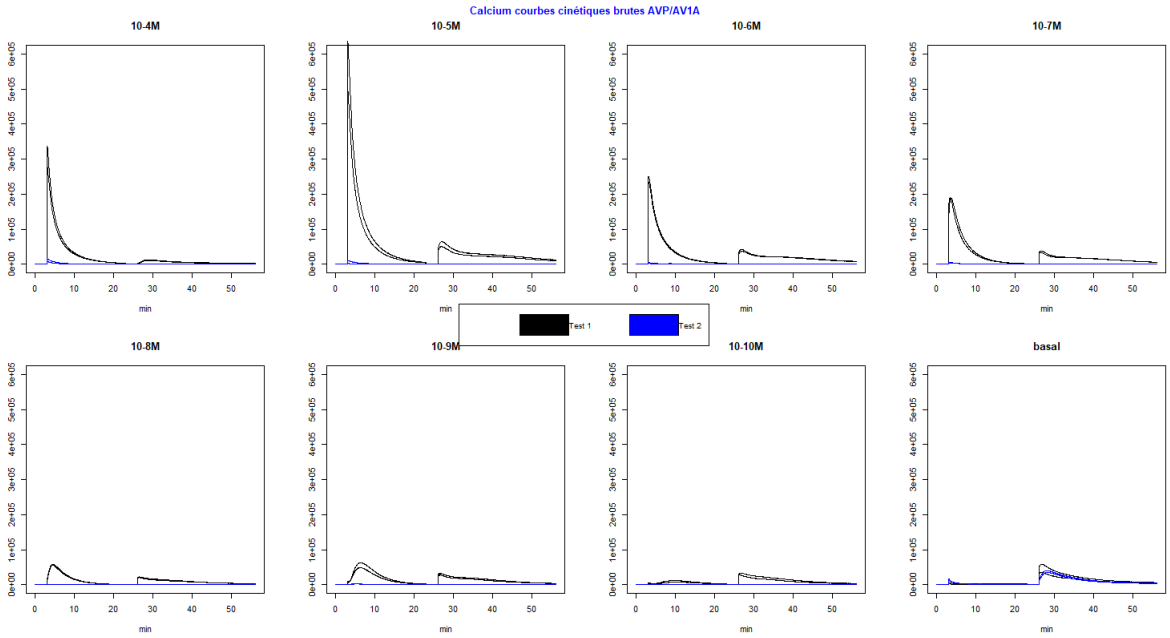

FIGURE D.41 : Courbe cinétique des données calcium brutes du couple AVP/AV1A

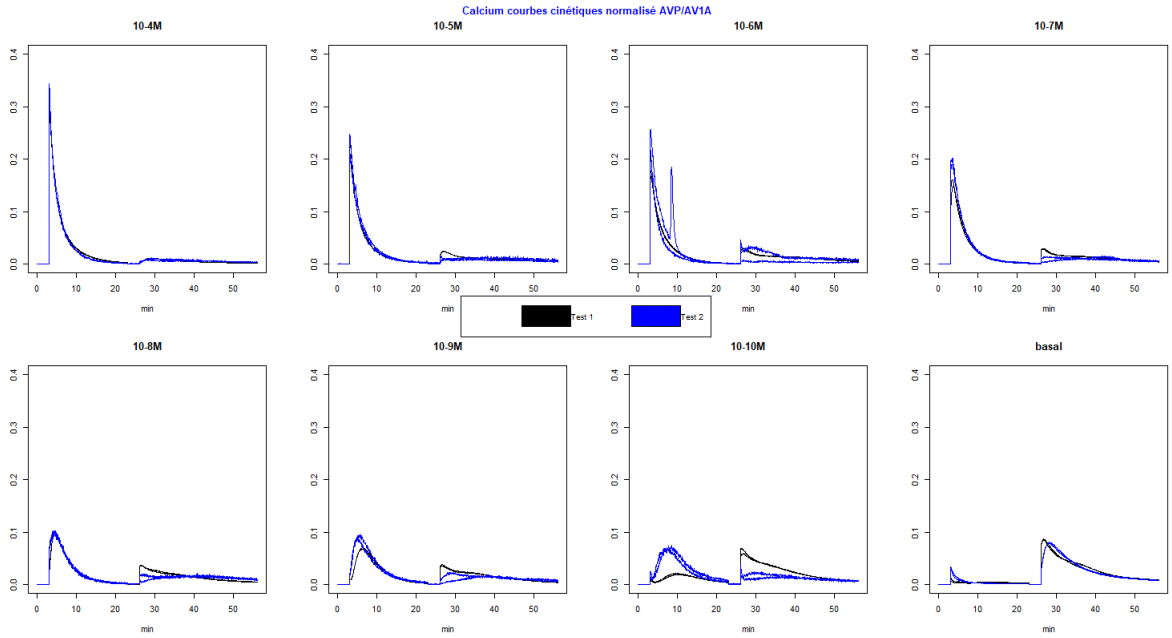

FIGURE D.42 : Courbe cinétique des données calcium normalisées du couple AVP/AV1A

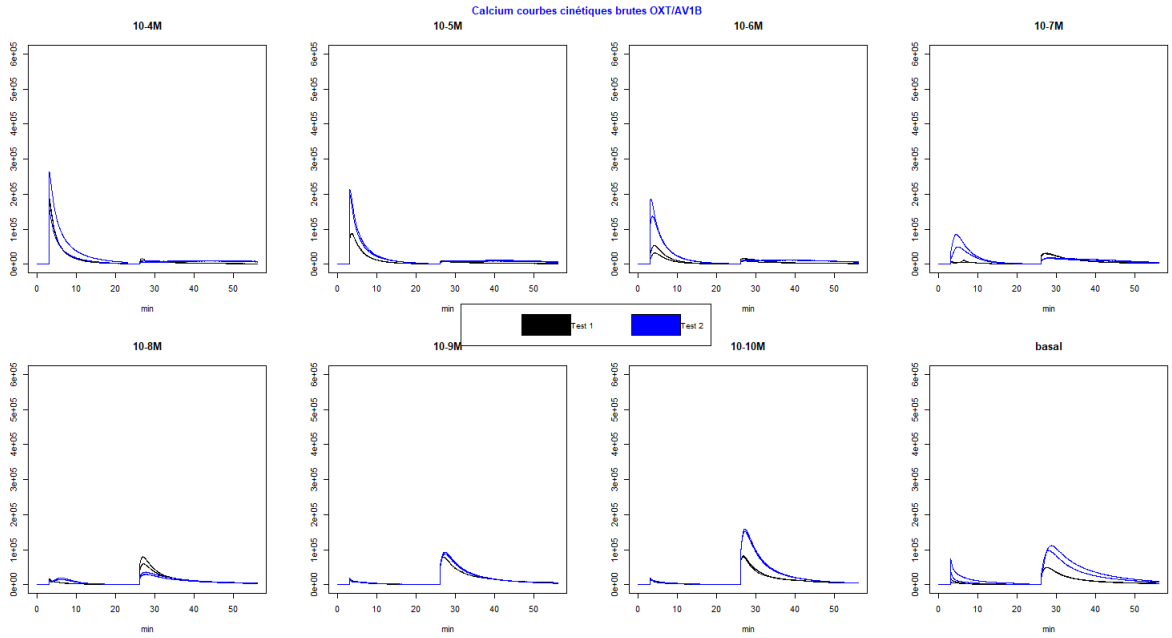

FIGURE D.43 : Courbe cinétique des données calcium brutes du couple OXT/AV1B

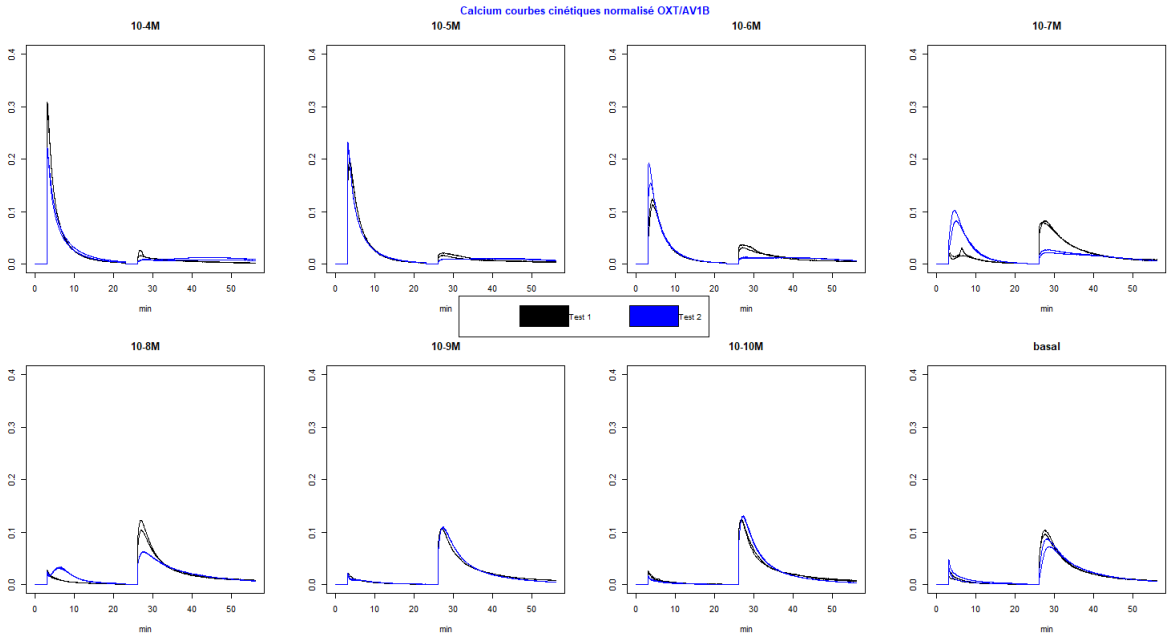

FIGURE D.44 : Courbe cinétique des données calcium normalisées du couple OXT/AV1B

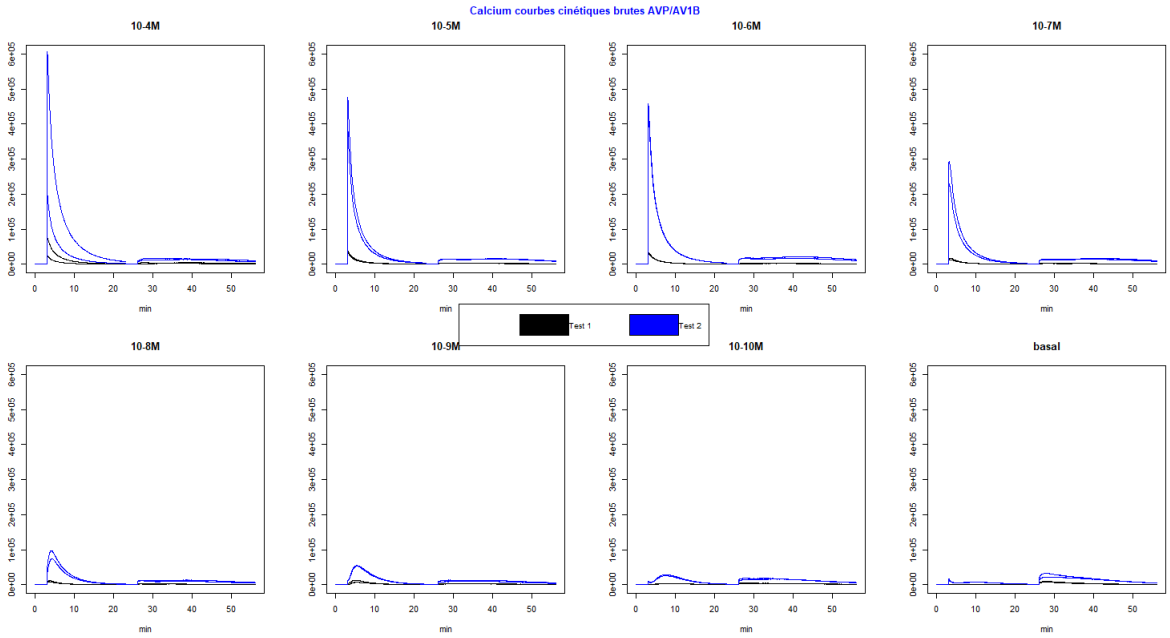

FIGURE D.45 : Courbe cinétique des données calcium brutes du couple AVP/AV1B

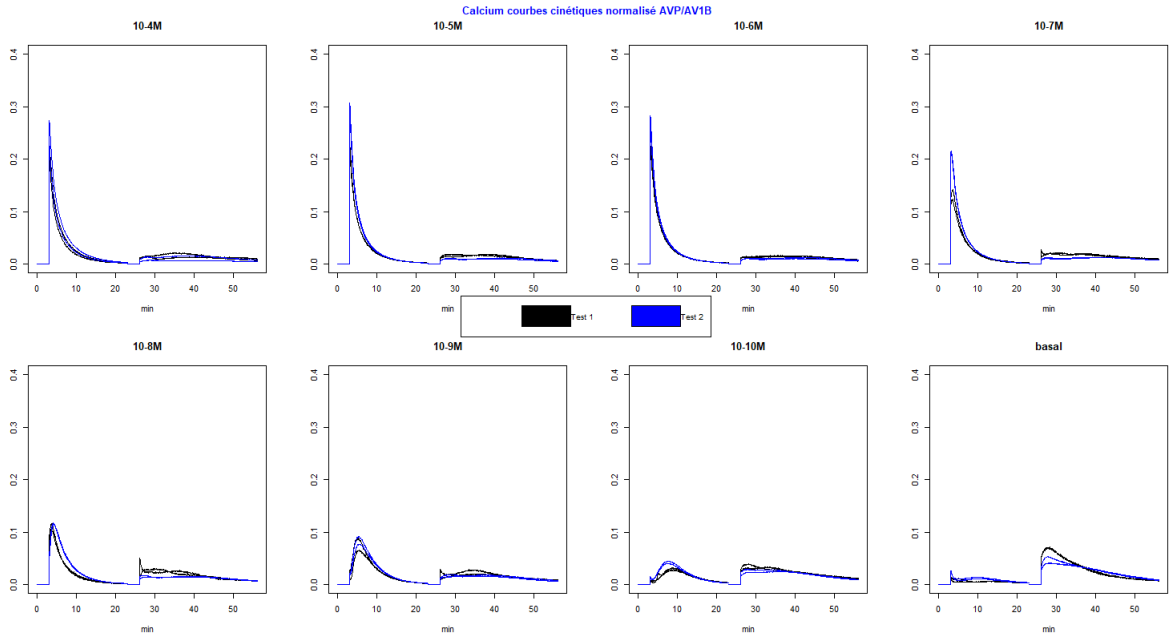

FIGURE D.46 : Courbe cinétique des données calcium normalisées du couple AVP/AV1B

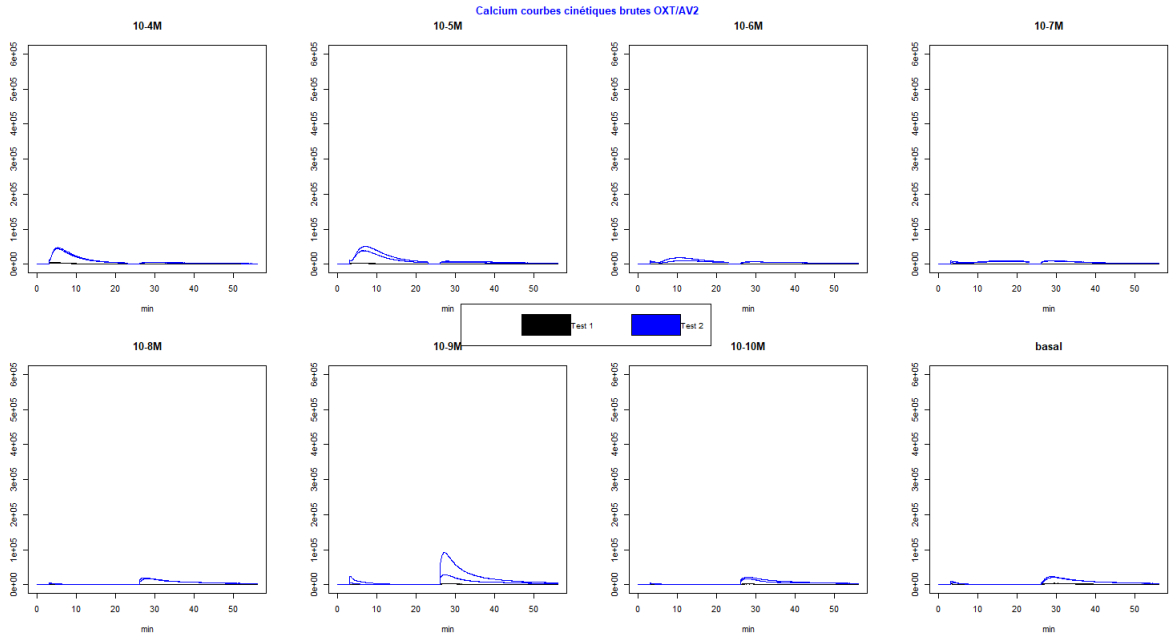

FIGURE D.47 : Courbe cinétique des données calcium brutes du couple OXT/AV2

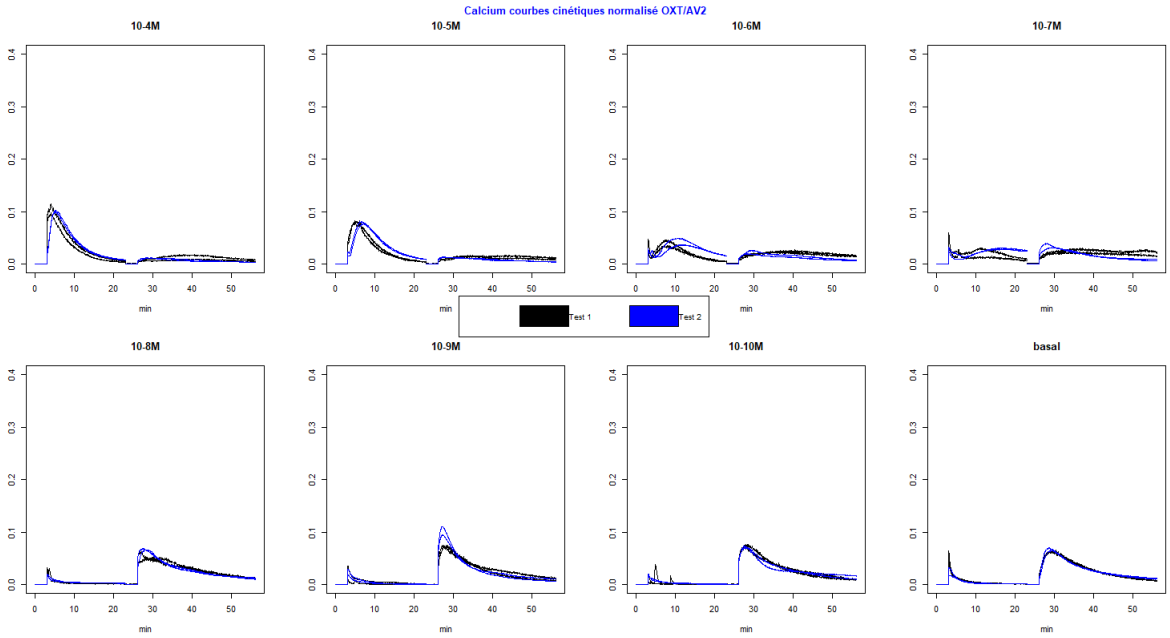

FIGURE D.48 : Courbe cinétique des données calcium normalisées du couple OXT/AV2

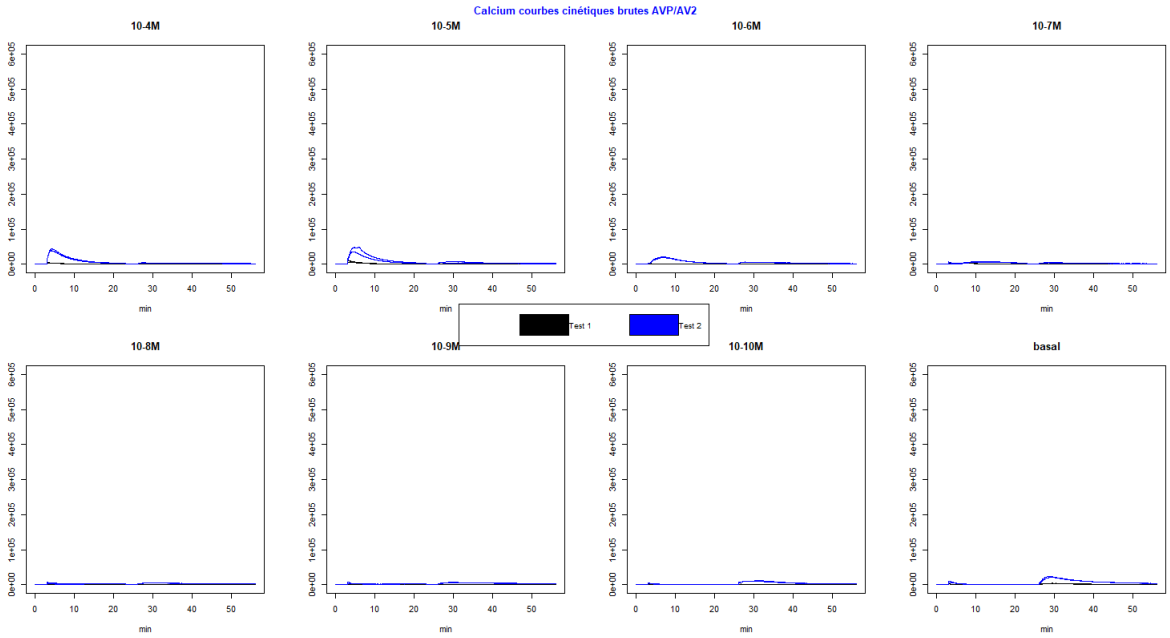

FIGURE D.49 : Courbe cinétique des données calcium brutes du couple AVP/AV2

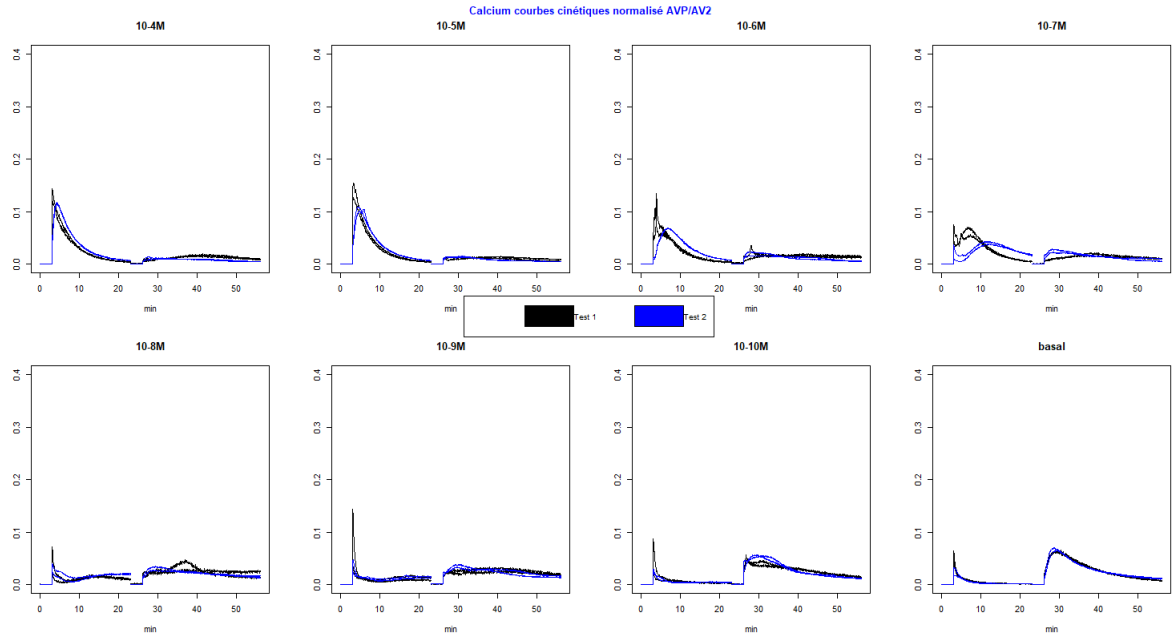

FIGURE D.50 : Courbe cinétique des données calcium normalisées du couple AVP/AV2

D.4.4 Calcium brut

(Retour au rapport : (cf.Annexe **??** page **??**), programme :(cf.Annexe **??** page **??**))

(2)

Bibliographie

[1] D.Goulet. Modeling, simulating, and parameter fitting of biochemical kinetic experiments. *SIAM Review*, 21 août 2015.

[2] Brian Ingalls. Mathematical modelling in systems biology : An introduction. *University of Waterloo*, chapitre 1 et 2, June 18, 2012.

[3] Boris N. Kholodenko. Cell-signalling dynamics in time and spac. *NATURE REVIEWS | MOLECULAR CELL BIOLOGY*, volume 7, march 2006.