



ROYAUME DU MAROC  
*Université Mohammed V - Rabat*  
*Faculté de Médecine et de Pharmacie*  
RABAT



Année 2021

N°: MS203/21

# Mémoire de fin d'études

Pour L'obtention du Diplôme National de  
Spécialité en : « Pharmacie industrielle »

Intitulé

**La Modélisation bayésienne hiérarchique  
appliquée aux données de dissolution des  
suppositoires à bases d'Eudragits L100 et S100**

Présenté par :

**Docteur** ABDELHAFID BENOMAR

Sous la direction du

**Professeur** AICHA FAHRY

---

# *Remerciements*

---

*Mes remerciements vont au Directeur du diplôme et à l'ensemble des professeurs des UPR encadrant la spécialité de « pharmacie industrielle » de la faculté de médecine et de pharmacie de RABAT pour leur encadrement, disponibilité et partage d'expériences.*

*Nous vous sommes très reconnaissants de votre engagement et des efforts fournis pour notre encadrement durant ce cursus de spécialité.*

*Je voudrais aussi témoigner toute ma reconnaissance à Aicha FAHRI professeure de pharmacie galénique pour son accompagnement durant la conduite de ce travail.*

*Je remercie également toute l'équipe du service de pharmacie galénique.*

## TABLE DES MATIERES

<b>LISTE DES FIGURES.....</b>	<b>4</b>
<b>LISTE DES ANNEXES .....</b>	<b>5</b>
<b>LISTE DES ABRIVIATIONS.....</b>	<b>6</b>
<b>I INTRODUCTION .....</b>	<b>7</b>
<b>II MATÉRIEL ET MÉTHODES.....</b>	<b>10</b>
<b>1. Matériel.....</b>	<b>10</b>
<b>2. Plan de mélange.....</b>	<b>10</b>
<b>3. Préparation de suppositoires par fusion- Moulage:.....</b>	<b>11</b>
<b>4. Essai chimiques et pharmaco-techniques des suppositoires préparés .....</b>	<b>11</b>
<b>5. Modélisation de la cinétique de dissolution .....</b>	<b>13</b>
<b>III RÉSULTATS ET DISCUSSION.....</b>	<b>18</b>
<b>IV CONCLUSION : .....</b>	<b>26</b>
<b>REFERENCES BIBLIOGRAPHIQUES .....</b>	<b>27</b>
<b>ANNEXES :.....</b>	<b>32</b>
<b>ANNEXE 1 : Données bruts de l'analyse statistiques.....</b>	<b>32</b>
<b>ANNEXE 2 : Selection du modèle (Paquet de Python) .....</b>	<b>34</b>
<b>1- importation des données .....</b>	<b>35</b>
<b>2- creation des fonction des modèles .....</b>	<b>35</b>
<b>3- indexation pour l'hierarchisation des modèles .....</b>	<b>36</b>
<b>4- creation des models baysien avec hierarchisation ou non .....</b>	<b>36</b>
<b>5- Relation entre Eud_S et EUD_L avec K1.....</b>	<b>45</b>
<b>Résumé .....</b>	<b>50</b>
<b>Abstract:.....</b>	<b>51</b>
<b>الملخص.....</b>	<b>52</b>
<b>Graphical Abstract : .....</b>	<b>53</b>

## LISTE DES TABLEAUX :

N°	Titre	Page
Tableau 1	Matrice des expériences avec des composantes fixes (PEG6000 et Barbital) et variables (EUD_L et S)	13
Tableau 2	Résultats des essais chimiques et pharmaco-techniques des 13 formules	21
Tableau 3	Résultats de l'essai de dissolution pour les 13 formules	22
Tableau 4	Le R sq (Adjust) et AIC pour la cinétique de libération de Barbital en utilisant différents modèles mathématiques obtenus par DD-Solver.	24
Tableau 5	Comparaison des modèles basée sur la WAIC, la pWAIC (Le nombre effectif de paramètres), la dWAIC (la différence entre les scores WAIC des autres modèles avec le meilleur modèle), le poids (la probabilité relative que le modèle explique les données).	25
Tableau 6	Statistiques de comparaison des modèles, p_loo, le nombre effectif de paramètres ; d_loo (la différence entre les scores loo des autres modèles avec le meilleur modèle), poids (la probabilité relative que le modèle explique les données).	25

## LISTE DES FIGURES

N°	Titre	Page
Figure 1	Courbe d'étalonnage de Barbitol sodique	15
Figure 2	Modèles cinétiques	17
Figure 3	Libération cinétique de premier ordre de Barbitol pour toutes les formulations.	23
Figure 4	Relation quadratique entre $K_1$ et la composition en Eudragits ( $p=0.05$ )	23
Figure 5	Courbe et distribution postérieure du paramètre du meilleur modèle ( $K_1$ ) et de son écart-type ( $\sigma$ )	26
Figure 6	$K_1$ estimé par la méthode bayésienne, comparé au $K_1$ estimé par la méthode standard. (La valeur $P$ et le coefficient de détermination ( $R^2$ ) ont été calculés dans Scipy en utilisant la technique de régression non linéaire des moindres carrés).	27
Figure 7	Résultats observés par rapport aux valeurs prévues par la méthode bayésienne (la valeur $P$ et le coefficient de détermination ( $R_{sq}$ ) ont été calculés dans Scipy en utilisant la technique de régression linéaire par les moindres carrés).	27
Figure 8	Taux de dissolution ( $K_1$ ) basé sur les quantités de deux Eudragits L et S. (La valeur $p$ et le coefficient de détermination ( $R_{sq}$ ) ont été calculés dans Scipy en utilisant la technique de régression linéaire des moindres carrés).	28

## **LISTE DES ANNEXES**

Annexe 1: Données bruts de l'analyse Bayésienne

Annexe 2 : Package Python

## LISTE DES ABRIVIATIONS

EUD\_L : EUDRAGIT L100

EUD\_S : EUDRAGIT S100

PA : Principe actif (Barbital)

Waic: widely applicable information  
criterion

LOO : Leave-one-out cross-validation  
(Validation croisée de l'exclusion)

Eu.Ph : pharmacopée Européenne

AIC : Critère d'information d'Akaike

SR : libération prolongée (sustained  
release)

DO : Densité optique

l'U.S.P. USA pharmacopée

PyMC3 : paquet python pour la  
modélisation probabiliste

MCMC : chaque chaîne de Markov Monte  
Carlo

pWAIC : le nombre effectif de paramètres

dWAIC : la différence entre les scores

WAIC des autres modèles avec le meilleur  
modèle

p\_loo : le nombre effectif de paramètres ;

d\_loo : Différence entre les scores loo des  
autres modèles avec le meilleur modèle

SE : Standard erreur

## I. INTRODUCTION

Malgré une longue histoire d'utilisation de médicaments pour l'administration par voie rectale et vaginale, le marché mondial actuel des suppositoires est limité, principalement en raison du manque d'acceptabilité par les utilisateurs. Par conséquent, le développement pharmaceutique rationnel dans le domaine de suppositoires reste faible. Pourtant, les suppositoires offrent plusieurs avantages par rapport aux autres formes pharmaceutiques. (T. V. Orlova et al .2010, Anthony S Ham et al .2017)

Les méthodes de formulation actuelles de suppositoires sont basées sur l'approche empirique de développement pharmaceutique, c'est à dire sans optimisation. Ainsi, le développement émergent de suppositoires s'est concentré sur l'amélioration des formulations classique existante afin d'améliorer l'administration des PA .( Anthony S Ham et al .2017)

Par conséquent, avec la poursuite de la recherche et le développement rationnels de suppositoires, il existe un potentiel important pour l'administration de médicaments de la plus par des familles thérapeutiques par suppositoires.

Les objectifs généraux de la formulation des suppositoires à libération prolongée sont d'obtenir une concentration sanguine souhaitable du principe actif (PA) , de maintenir cette concentration à un niveau à peu près constant pendant une période de temps appropriée , de réduire la fréquence d'administration du médicament et de ses effets secondaires, et de prolonger la vitesse d'absorption au niveau de rectum , en effet, les suppositoires à libération prolongée peuvent permettre obtenir une action clinique soutenue en soulageant par exemple la douleur pendant le sommeil, ce qui est souvent utile pour réduire l'anxiété (Ambikanandan Misra et al , 2014 . Ozgüney et al 2008).

Les polymères sont devenus des éléments incontournables dans le développement de nouveaux produits pharmaceutiques dont les suppositoires. Ainsi, les progrès en sciences des polymères ont révolutionné la conception, le développement et les performances des systèmes de délivrance de médicaments (Drug delivery système DDS). Parmi ces polymères (Tableau), les Eudragits ont fait l'objet de plusieurs études de formulation de suppositoires, surtout Eudragit RL100 et RS 100 (Ambikanandan Misra et al. 2014).

Plusieurs études ont utilisé les polymères Eudragits pour développer des suppositoires à libération prolongé.

Ozgüney et ces collaborateurs (2008) ont préparé des suppositoires de Kétoprofène en utilisant Eudragit RL100 et ont suivi la dissolution pendant 8 heures.



Cette étude a montré l'avantage de ces formes matricielles par rapport aux formes conventionnelles de ce principe actif. Dans une autre étude S.Maity (2001) ont formulés des suppositoires à deux couches à libération prolongée, ces derniers ont été préparés en utilisant du polyéthylène glycol 4000 et de l'Eudragit RS-100 et leurs caractéristiques chimiques et pharmaco-techniques ont été évaluées à la fois in vitro et in vivo.

Pour les polymères sujets de notre mémoire ; (i) Eudragit L100 et (ii) Eudragit S100 (Copolymère d'acide méthacrylique) et à notre connaissance la seule étude a été faite est par E.Baloglu et al. (2002), ils ont préparé et évalué la cinétique de libération à partir de suppositoires à libération prolongée formulés et composés de PEG 400, 4000, Witepsol et Eudragit L100.

Ces deux polymères présentent des sensibilités différentes au pH à savoir que l'EUD\_L se solubilise dans des  $\text{pH} \geq 6$  et l'EUD\_S se solubilise à des  $\text{pH} \geq 7$  (Joao A.C et al 2019) d'où l'intérêt d'associer les deux dans une même formulation. En optimisant la composition du mélange de ces deux polymères, la vitesse de dissolution désirée peut être obtenu en modifiant la diffusivité du liquide rectale à travers la gélification de la structure de la matrice (Di Colo, Falchi et al. 2002 ; Nikam, Kotade et al. 2011 ; Misra 2014 ; Surti, Mahajan et al. 2020).

Vu l'importance de la dissolution pour la biodisponibilité et les performances cliniques, le test de dissolution des formes pharmaceutiques solides est devenue l'un des tests les plus importants dans le développement et la fabrication des médicaments, ainsi que dans l'évaluation réglementaire de la qualité des produits pharmaceutiques (R . Panchagnula et al 2000). Non seulement les tests de dissolution peuvent fournir des informations concernant le taux et l'étendue de l'absorption des médicaments dans l'organisme, mais ils peuvent également évaluer les effets des propriétés biopharmaceutiques des substances médicamenteuses et des excipients sur les propriétés de libération du PA (Sau Lawrence Lee et al).

Les modèles mathématiques jouent un rôle crucial dans l'interprétation du mécanisme de libération des PA en fonction de la dose administré (Varma, Kaushal et al. 2004 ; Grassi, Grassi et al. 2006 ; Gouda, Baishya et al. 2017 ; Paarakh, Jose et al. 2018). La cinétique de libération des PA pour des formulations données a généralement été ajustée à des modèles semi-empiriques prédéfinis en utilisant une approche de modélisation fréquentiste non hiérarchique.

Le modèle approprié pour chaque formulation a été sélectionné sur la base du R carré (Adjust) le plus élevé et du critère d'information d'Akaike (AIC) le plus bas (Gouda, Baishya et al. 2017 ; Victor et Francis 2017 ; Paarakh, Jose et al. 2018).

Une telle situation peut conduire à une variation du modèle le mieux adapté parmi les formulations, en particulier lorsque le nombre de points de la courbe était limité. Ceci est peu significatif d'un point de vue biopharmaceutique étant donné l'interdépendance entre les données. De plus, la convergence des modèles peut être problématique (Piray, Dezfouli et al. 2019).

La modélisation bayésienne hiérarchique peut résoudre ces problèmes en exprimant le modèle sur toutes les formulations et par conséquent en traitant à la fois l'ajustement et la comparaison des modèles dans le même cadre (Piray, Dezfouli et al. 2019). L'approche de la modélisation bayésienne hiérarchique suppose que les formulations partagent le même meilleur modèle et ne diffèrent que par les valeurs de certains ou de tous les paramètres du modèle (Andrews, Ansari et al. 2002 ; Millar 2009 ; Wiecki, Sofer et al. 2013)

Les différences entre les deux concepts statistiques ont leur origine dans des conceptions différentes de la probabilité. Dans la pratique, l'approche fréquentiste quantifie les propriétés des données dérivées, tandis que l'approche bayésienne quantifie l'incertitude de la mesure (VanderPlas 2014).

Le premier objectif de ce mémoire est de formuler des suppositoires à bases des Eudragits en incorporant un PA (Barbital sodique) puis de suivre le profil de dissolution de ces derniers. Le deuxième objectif était d'utiliser l'approche hiérarchique de la modélisation bayésienne pour sélectionner le modèle cinétique empirique approprié de la libération prolongée puis d'optimiser la vitesse de dissolution in vitro du suppositoire de la matrice en contrôlant la composition d'EUD\_L100 et EUD\_S100 dans le mélange expérimental ( $K_{1(\text{libération})} = f(\text{Eud-L100}, \text{Eudr-S100})$ ).

## II. MATÉRIEL ET MÉTHODES

### 1. Matériel

Tous les matériaux utilisés dans cette étude, c'est-à-dire le Sodium Barbital", le PEG 6000, l'Eudragit® \_L100, l'Eudragit® \_S100 (Rohm Pharma, GmbH, Allemagne), le NaH<sub>2</sub>PO<sub>4</sub> (monobasique) et Na<sub>2</sub>HPO<sub>4</sub> (dibasique) ont été acquis par le laboratoire de pharmacie galénique de la faculté de médecine et de pharmacie de Rabat-Maroc. Tous les ingrédients utilisés dans cette étude étaient de qualité analytique.

### 2. Plan de mélange

Pour la présente étude, nous avons utilisé un plan d'expérience personnalisé avec contrainte et avec facteur de processus, à savoir le temps.

Les quantités des deux Eudragits variaient de 0 à 2 g chacun, avec une contrainte supérieure de 2 g pour la somme des deux Eudragits L et S, sur la base de tests préliminaires. Au-delà de cette limite, il y aurait peu d'écoulement, un mauvais remplissage des moules métalliques et un aspect inacceptable des suppositoires préparés. La quantité de Macrogol 6000 et de Barbital a été fixée à 17,4 g et 0,6 g respectivement. Pour chacune des 13 formulations, trois expériences ont été réalisées et cinq temps d'échantillonnage ont été effectués : 20, 50, 80, 160 et 235 min (**Tableau 1**).

Essais	PEG6000 (g)	BARBITAL (g)	EUD_L (X1) (g)	EUD_S (X2) (g)	Temps de prélèvement en minute :20,50, 80,160,235min
1	17,4	0,6	0.7	1.3	20,50, 80,160,235min
2	17,4	0,6	2	0	20,50, 80,160,235min
3	17,4	0,6	1.8	0.2	20,50, 80,160,235min
4	17,4	0,6	0.3	1.7	20,50, 80,160,235min
5	17,4	0,6	1.5	0.5	20,50, 80,160,235min
6	17,4	0,6	1.2	0.8	20,50, 80,160,235min
7	17,4	0,6	1	1	20,50, 80,160,235min
8	17,4	0,6	1.3	0.7	20,50, 80,160,235min
9	17,4	0,6	0.2	1.8	20,50, 80,160,235min
10	17,4	0,6	0.5	1.5	20,50, 80,160,235min
11	17,4	0,6	0.8	1.2	20,50, 80,160,235min
12	17,4	0,6	1.7	0.3	20,50, 80,160,235min
13	17.4	0.6	0	0	20,50, 80,160,235min

**Tableau 1** : Matrice des expériences avec des composantes fixes (PEG6000 et Barbital) et variables (EUD\_L et

S)

### 3. Préparation de suppositoires par fusion- Moulage:

Il est réalisé par moulage par fusion, le système matriciel consiste à dissoudre ou à disperser de manière homogène le principe actif dans toute la matrice polymère. La préparation des suppositoires de Barbitol à libération prolongée (SR) a été conçue selon les contraintes de la conception du mélange (somme des deux Eudragits inférieure ou égale à 10%).

La quantité de PEG 6000 est portée à 46 C° et maintenue sous agitation manuelle dans la capsule en acier inoxydable jusqu'à l'obtention d'une masse homogène sans incorporation de bulles d'air, puis les autres éléments de la formule sont ajoutés en fonction de la matrice d'expérience. Après remplissage, le moule métallique est placé au réfrigérateur pendant quelques minutes ; le temps nécessaire à la solidification des suppositoires.

A la sortie de réfrigérateur, on procède au raclage et au contrôle visuel (Coloration, présence ou non de fissures) avant de stocker les unités formulées dans des flacons en verre fermés hermétiquement. La dernière étape, c'est d'aller contrôler ces formulations selon la monographie générale de la Pharmacopée européenne (Eur.Ph 10).

### 4. Essai chimiques et pharmaco-techniques des suppositoires préparés

Les suppositoires préparés ont été évalués en fonction des paramètres de la monographie générale Eur.Ph, à savoir l'uniformité de masse, l'uniformité du teneur, la dureté, la désagrégation et le test de dissolution.

#### *a. Caractère organoleptique (Aspect) : Couleur, Odorat, Fissuration*

Les suppositoires doivent être acceptables par le patient (plus ou moins blanchâtre).

#### *b. Uniformité de Masse :*

La norme est inférieure à 5% (le poids moyen des suppositoires adulte est de 2.5g)

#### *c. Teneur en principe actif :*

Ce test est effectué sur un suppositoire qui doit être fondu et dissous dans 250 ml de tampon phosphate pH=7,2. La teneur en Barbitol a été déterminée à l'aide d'un spectrophotomètre UV/Visible en mesurant l'absorbance de l'échantillon à 245 nm. Pour la conversion des données de densité optique (DO) en concentration puis en % F de dissolution, un étalonnage externe a été effectué comme suit : à partir d'une solution mère de 400 µg/ml de barbitol de sodium, 3 tests d'une solution de 40 µg/ ml ont été préparés, la

lecture de la DO à 245 nm et la courbe d'étalonnage externe a été tracée par Excel comme suit (Figure 1).

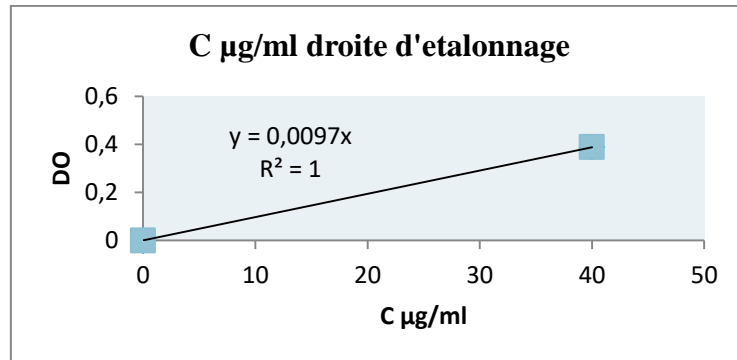


Figure 1: Courbe d'étalonnage de Barbital sodique

#### *d. Dureté*

Ce test a été effectué avec l'appareil type Erweka AR 400 (Erweka Apparatebau-GmbH Allemagne). Le suppositoire a été placé dans le dispositif de maintien avec la pointe vers le haut et la chambre d'essai a ensuite été fermée avec une plaque de verre. La température à l'intérieur de la chambre d'essai a été maintenue à 25°C au moyen d'une circulation d'eau provenant du thermostat connecté au testeur. Une charge initiale (600 g) a été appliquée et à intervalles réguliers d'une minute, un disque de 200 g a été ajouté jusqu'à ce que le suppositoire soit écrasé. La masse nécessaire pour écraser le suppositoire a ensuite été calculée comme la somme de la charge initiale et des masses ajoutées jusqu'à ce que le suppositoire s'effondre.

#### *e. Temps de désagrégation*

Le test a été effectué dans une solution tampon de pH 7.2 à 37°C (+/- 0,5) en utilisant l'appareil de désagrégation de comprimés de l'U.S.P. (SOTAX DT 3, Heusenstamm, Allemagne). Le temps de désagrégation a été enregistré dès que les suppositoires placés dans le panier étaient complètement dissous. Il s'agit d'un test optionnel pour les suppositoires à libération modifiée.

#### *f. Essais de dissolution*

Le pourcentage % F de dissolution a été mesuré par l'appareil de dissolution type Panier (Type 1 USP), le choix de cet appareil a été basé sur des essais préliminaires : voir annexes.

(PHARMA TEST PTWS 1220, 63512 HAINBURG ALLEMAGNE). Les conditions de test sont les suivantes : (1) volume de liquide de dissolution de 500 ml (2) tampon de phosphate 7,2 (3) la température d'essai est réglée à 37°C (± 0,5°C) et (4) la vitesse de rotation des paniers est réglée à 50 r/min. Des échantillons de 3 ml ont été prélevés aux intervalles de temps déjà mentionnés. Le volume initial du milieu de dissolution a été maintenu par l'ajout d'une quantité équivalente de solution tampon fraîche après chaque prélèvement. La concentration de Barbitol libéré a été déterminée à 245 nm<sup>39</sup> à l'aide du spectrophotomètre UV/Vis (Spectrophotometre 6305 UV-VIS 198 AT 1000 nm JENWAY, Bibby Scientific France SAS).

## 5. Modélisation de la cinétique de dissolution

Pour les suppositoires à libération modifiée, la libération du principe actif est due à différents paramètres tels que la désintégration, la diffusion du principe actif à travers une membrane, le gonflement et la diffusion à travers un polymère, l'érosion d'une matrice (Costa et Lobo 2001 ; Faisant, Siepmann et al. 2002 ; Varma, Kaushal et al. 2004). Différents modèles cinétiques ont été utilisés pour décrire la cinétique de libération (**Figure 2**). (Faisant, Siepmann et al. 2002 ; Peppas et Narasimhan 2014). Le taux d'ordre zéro (T. Güneri 2004) Eq. (1) explique les systèmes où le taux de libération de PA ne dépend pas de sa concentration. Le modèle du premier ordre (T. Güneri 2004). Eq. (2) explique les systèmes où le taux de libération de la drogue dépend de sa concentration.

Le modèle de Higuchi décrit la libération de drogues à partir d'une matrice insoluble comme une racine carrée de processus dépendant du temps, basé sur le modèle de diffusion de Fickienne Eq (3). Korsmeyer et al. ont dérivé une relation mathématique simple qui décrit la libération de drogue à partir d'un système polymère Eq. (4) et du modèle de Hixson-Crowell Eq(5) (Costa et Lobo 2001 ; Varma, Kaushal et al. 2004 ; Peppas et Narasimhan 2014).

Le modèle cinétique d'ordre dit "zéro" :

$$Q_0 = K_0 t \quad (1)$$

Le modèle cinétique de premier ordre :

$$Q_1 = (1 - e^{-kt}) \quad (2)$$

Le modèle cinétique de Higuchi :

$$Q_{KP} = K_{KP} t^n \quad (3)$$

Le modèle cinétique de Korsmeyer-Peppas :

$$Q_H = K_H t^{0.5} \quad (4)$$

Le modèle Hixson Crowell

$$Q_{HC} = \{1 - [(1 - K_{HC} t)]^3\} \quad (5)$$

Où Q est le pourcentage cumulé de libération de Barbital, Q0, QH, QKP et QHC sont les interceptes des modèles, k0, K1, KHC et KKP sont le taux de libération, n'est l'exposant de diffusion indiquant le mécanisme de libération de la drogue (Costa et Lobo 2001 ; Zhang, Huo et al. 2010).

Lorsque la valeur de n est égale à 0,5, la libération de PA suit la loi de diffusion de Fickienne. Lorsque 0,5 < n < 1, la libération de PA suit une diffusion non fickienne. Lorsque = 1,0, la libération de PA est causée par le gonflement du polymère (Costa et Lobo 2001 ; Zhang, Huo et al. 2010).

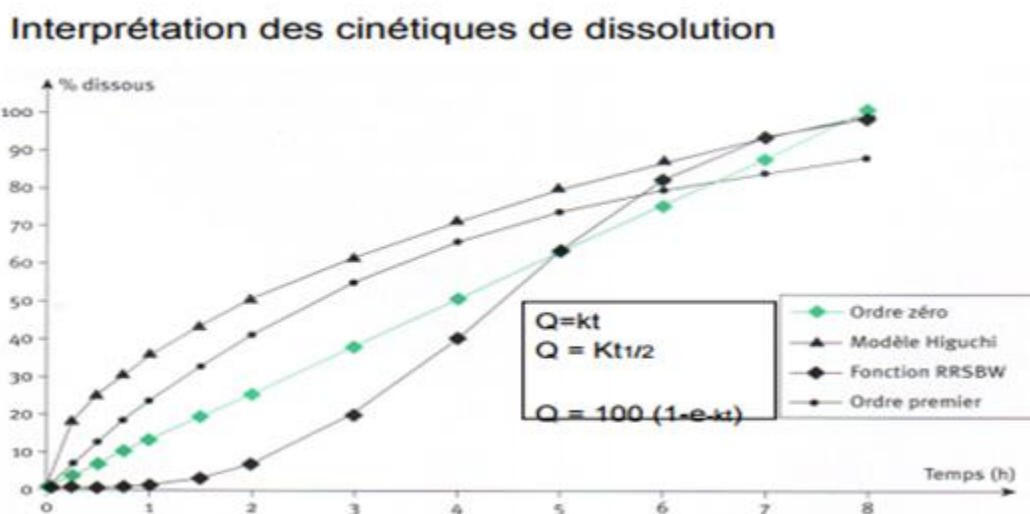


Figure 2 : Les Modèles cinétiques les plus utilisés en biopharmacie (P.Wehrlé. 2012).

Les différences entre les approches fréquentistes et bayésiennes en matière d'analyse statistique trouvent leur origine dans des conceptions différentes de la probabilité : les fréquentistes définissent la probabilité comme étant liée aux fréquences d'événements répétés, tandis que les bayésiens la définissent comme une mesure de l'incertitude (Hong, Carlin et al. 2013 ; Vander Plas 2014). En pratique, cela signifie que les fréquentistes quantifient généralement les propriétés des quantités dérivées des données à la lumière des paramètres fixes du modèle, tandis que les Bayésiens quantifient généralement les propriétés des paramètres inconnus du modèle à la lumière des données observées.

### *5.1 Approche fréquentiste*

Les données de chaque formulation ont été ajustées individuellement aux cinq modèles mentionnés ci-dessus à l'aide de DD -Solver, un programme complémentaire de Microsoft Excel (Zhang, Huo et al. 2010 ; Zuo, Gao et al. 2014) .

Le DD -Solver utilise la technique d'ajustement non linéaire par les moindres carrés, et les paramètres des modèles ont été déterminés en minimisant la somme des erreurs quadratiques (SS) (Nakata 2010 ; Zuo, Gao et al. 2014).

Les valeurs initiales du modèle ont été fixées individuellement avant l'optimisation itérative. Le meilleur modèle pour chaque formulation est sélectionné sur la base de plusieurs critères tels que R-sq ajusté et AIC. Tous deux présentent une limitation dans le processus de sélection et d'inférence du modèle lorsque le nombre de paramètres est différent entre les modèles candidats (Nakata 2010). De plus, le DD-Solveur ne permet pas de hiérarchiser un ou tous les paramètres du modèle. Étant donné que le modèle sélectionné (modèle le mieux adapté) peut varier d'une formulation à l'autre, en particulier lorsque le nombre de points de la courbe a été limité (Nakata 2010).

### *5.2 Approche bayésienne*

Les données de toutes les formulations ont été ajustées aux cinq modèles mentionnés ci-dessus à l'aide d'un modèle bayésien hiérarchique utilisant les paquets PyMC3 de Python (Patil, Huard et al. 2010 ; Salvatier, Wiecki et al. 2016). Cette approche suppose que les formulations partagent le même meilleur modèle et ne diffèrent que sur les valeurs de certains ou de tous les paramètres du modèle, pour chacun des cinq modèles. Les paramètres antérieurs de chaque modèle étaient les suivants : la variation était supposée être nulle et les paramètres du modèle étaient fixés à la valeur initiale obtenue lorsque le modèle était ajusté à toutes les données.

Le tracé de la distribution postérieure des paramètres sur chaque chaîne de Markov Monte-Carlo (MCMC) et la convergence des chaînes ont été utilisés pour examiner la performance des modèles (Cowles et Carlin 1996 ; Bates et Campbell 2001). Les MCMC sont une classe d'algorithmes qui peuvent caractériser efficacement les distributions postérieures, même de grande dimension, en tirant des échantillons randomisés de telle sorte que les points soient distribués en fonction de la partie postérieure (VanderPlas 2014). Pour chaque modèle, des MCMC en chaîne de 2000 cycles ont été considérés avec une période de rodage de 200 premiers cycles. Le programme python (comme le carnet Jupyter) est disponible comme données supplémentaires.



Nous avons utilisé deux méthodes pour la sélection des modèles bayésiens, telles que le critère d'information largement applicable (WAIC) et la validation croisée (LOO) (Watanabe et Opper 2010 ; Vehtari, Gelman et al. 2017).

Ces méthodes évaluent les modèles sur la base de leur validité prédictive. La validité prédictive est une évaluation de la capacité de chaque modèle à prédire les résultats d'expériences futures.

Cependant, les mesures de la validité prédictive basées sur un nouvel ensemble de données peuvent ne pas être réalisables, ou être coûteuses. La méthode Loo examine chaque donnée omise la performance du modèle (Gelman, Hwang et al. 2014 ; Vehtari, Gelman et al. 2015 ; Luo et Al-Harbi 2017). La WAIC utilise l'ensemble de la distribution postérieure des modèles pour évaluer la performance du modèle (Gelman, Hwang et al. 2014 ; Vehtari, Gelman et al. 2015 ; Luo et Al-Harbi 2017).

Le critère d'information largement disponible est calculé d'après Gelman et al. (2014) comme :

$$WAIC = -2 \log p_{ppd} + \frac{2}{n} \sum_{i=1}^n p_{WAICn} \quad (6)$$

Où  $p_{ppd}$  est la densité prédictive par point logarithmique et  $p_{WAIC}$  est un terme qui pénalise les modèles ayant un grand nombre de paramètres. Le facteur de -2 amène la WAIC à être sur la même échelle que les autres critères d'information.  $p_{ppd}$  est approximé en utilisant la sortie postérieure d'une chaîne MCMC par :

$$p_{ppd} = \sum \log (E_{postp} (y_i | \theta)) \quad (7)$$

Où la somme extérieure est sur des points de données individuels, et le terme intérieur est l'espérance de la probabilité sur l'ensemble de l'échantillon postérieur. Cette valeur est ensuite corrigée pour tenir compte du nombre estimé de paramètres du modèle par :

$$p_{WAICn} = \sum \text{var}_{post} \log(p(y_i | \theta)) \quad (8)$$

Où le terme interne est la variance de la probabilité sur l'ensemble de l'échantillon postérieur pour chaque point de données. Ce terme restant pénalise les modèles qui ont un ajustement inégal (c'est-à-dire une variance élevée) sur différents points de données, ce qui peut être une indication de surajustement.

La méthode la plus simple pour sélectionner un modèle est de classer les modèles en fonction de leur valeur WAIC ou de leur valeur loo et de choisir le modèle ayant la valeur la plus faible. Cependant, cette méthode simple ne tient pas compte de l'incertitude de la WAIC de chaque modèle compte tenu de l'erreur type de la WAIC. Nous avons donc procédé à une

sélection de modèles en calculant un poids moyen pour chaque modèle par rapport au meilleur modèle (le WAIC ou loo le plus bas) :

$$w_i = \frac{\exp(-1/2dIC_i)}{\sum_j \exp(-1/2dIC_j)} \quad (9)$$

Étaient  $w_i$ , le poids du modèle  $i$ , qui lui correspond la probabilité relative par rapport à tous les modèles testés.

Pour la sélection des modèles, nous avons 5 modèles hiérarchiques cités ci-dessus, en plus nous ajoutons les modèles communs correspondants pour chaque catégorie du modèle, qui visent à tester l'hypothèse de l'effet significatif des quantités des deux Eudragit sur le taux de rejet. En d'autres termes, la sélection du modèle hiérarchique correspond à un effet significatif des quantités d'Eudragit sur le taux de rejet, inversement la sélection du modèle commun correspond à l'absence d'effet des quantités d'Eudragit sur la cinétique de rejet.

L'effet d'Eudragits sur le taux de libération a été évalué en étudiant la relation entre les paramètres du modèle sélectionné et le poids de Eud\_L ou S dans la formulation à l'aide de la technique de régression non linéaire dans le paquet Scipy (Bressert 2012 ; Virtanen, Gommers et al. 2020).

### III. RÉSULTATS ET DISCUSSION

Tous les suppositoires satisfont aux exigences de la pharmacopée européenne en matière d'uniformité du contenu, de variation de poids, de dureté et de temps de désintégration (tableau 2). Cependant, l'uniformité du contenu varie de 96,1 à 101,5, et la variation de poids de 2,23g à 2,67g. La dureté de toutes les formulations était supérieure à 4,8 Kg et le temps de désintégration se situait entre 30 et 203 min (**Table 2**).

Essais	Uniformité de teneur%	Variation de masse (g)	Dureté (kg)	Désagrégation (min)	Aspect
F1	98,355	2.5	4.8	148	Acceptable
F2	98,478	2.67	4.8	55	Acceptable
F3	101,50	2.56	4.8	95	Acceptable
F4	100,04	2.51	4.8	197.5	Acceptable
F5	99,416	2.53	4.8	70	Acceptable
F6	98,009	2.46	4.8	80	Acceptable
F7	99,133	2.35	4.8	85	Acceptable
F8	97,990	2.53	4.8	75	Acceptable
F9	96,133	2.5	4.8	200	Acceptable
F10	99,207	2.53	4.8	203	Acceptable
F11	97,572	2.5	4.8	90	Acceptable
F12	98,630	2.57	4.8	55	Acceptable
13	100.5	2.56	4.8	30	Acceptable

Tableau 2 : Résultats des essais chimiques et pharmaco-techniques des 13 formules.

#### 1. Modélisation de la libération in vitro

Des tests de dissolution in vitro ont montré que les suppositoires préparés ont un effet SR, même s'ils ne contiennent pas d'Eudragit (formule n° 13 avec PEG 6000 uniquement). Le pourcentage de Barbital dissous en 4 heures (235 minutes) était supérieur à 90% pour 4 formules (F3, F5, F12 et F13), supérieur à 80% pour les formules F2, F6, F7 et F8 et inférieur à 80% pour 5 formules (F1, F4, F9, F10 et F11).(**Table 3**).

Formulations	20 min	50 min	80 min	160 min	235 min
F1 (%)	13.87	27.89	37.17	66.8	78.25
F2(%)	28.4	63.09	85.98	88.61	89.23
F3(%)	27.42	60.77	82.27	91.08	93.71
F4(%)	15	31.5	38.3	74.7	75.6
F5(%)	18.6	41.2	56.4	85.3	90.5
F6(%)	14	26.9	38.7	74.2	87.4
F7(%)	13.4	30.41	45.26	81.44	88.66
F8(%)	15.05	38.1	64.95	86.6	87.89
F9(%)	19.2	27.7	39.6	61.1	70.9
F10(%)	13.87	26.19	36.03	50.62	75.26
F11(%)	11.31	23.56	36.13	52.01	75.39
F12(%)	20.9	67.7	87	88.3	92.1
F13(%)	39.59	60.67	84.66	94.35	101.8

Tableau 3 : Résultats de l'essai de dissolution pour les 13 formules

### 1.1 Approche fréquentiste de la sélection des modèles

Les caractéristiques de libération in vitro des suppositoires SR ont été évaluées pour déterminer le modèle cinétique le mieux adapté. Les taux de libération de toutes les formulations ont été décrits au mieux avec le modèle de premier ordre qui présentait un  $Rsq(Adjust) > 0,90$  et  $AIC < 16$  (**Tableau 4**).

Sur la base du  $Rsq(Adjust)$ , le modèle du premier ordre était le meilleur modèle pour les formules 1, 2, 3, 4, 5, 8 et 13, le modèle de Higuchi est le meilleur pour la formule 9, le modèle de Korsmeyer-Peppas était le meilleur pour les formules 6, 10 et 11 et le modèle de Hixson-Krowell était le meilleur pour les formules 7 et 12.

Sur la base de l'AIC, le modèle du premier ordre était le meilleur pour les formules 1, 2, 3, 4, 5, 8 et 13, le modèle de Korsmeyer-Peppas est le meilleur pour les formules 9, 10 et 11 et enfin le meilleur modèle pour les formules 6 et 7 était celui de Hixson-Krowell (Zhang, Huo et al. 2010 ; Zuo, Gao et al. 2014). Cela a montré que la libération pour les formulations issus de même mélange ; formules interdépendantes, peut avoir différents meilleurs modèles in vitro, ce qui est l'une des faiblesses de l'approche fréquentiste (VanderPlas 2014).

En fait, les résultats de cette approche n'ont pas eu de signification biopharmaceutique claire.

La relation entre la constante K1 est la composition en Eudragits est quadratique comme montre la **figure 3**. La **figure 4** montre les profils de dissolution des 13 formules selon le modèle de premier ordre qu'on suppose le plus adapté selon l'approche fréquentiste).

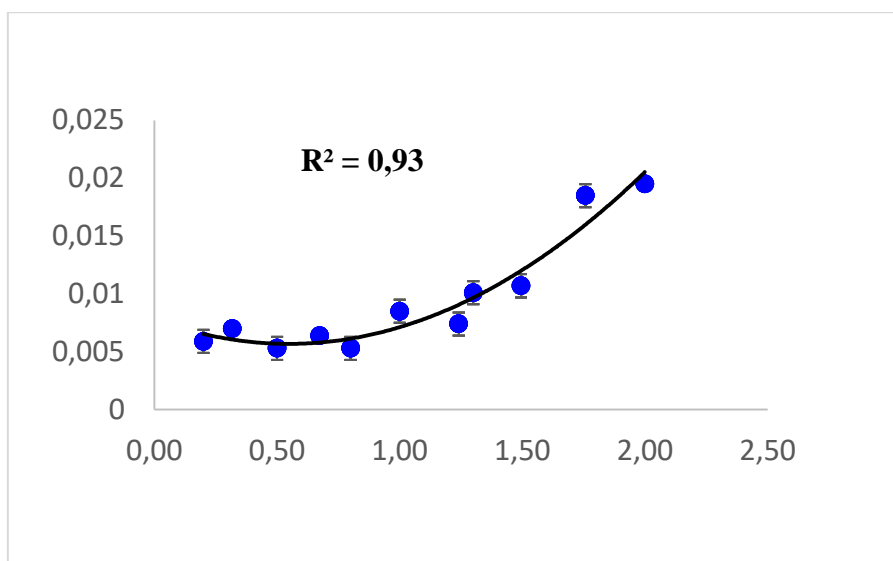


Figure 3: Relation quadratique entre  $K_1$  et la composition en Eudragits ( $p=0.005$ ) (Pour les valeurs de  $k_{1f}$  voir l'annexe 1)

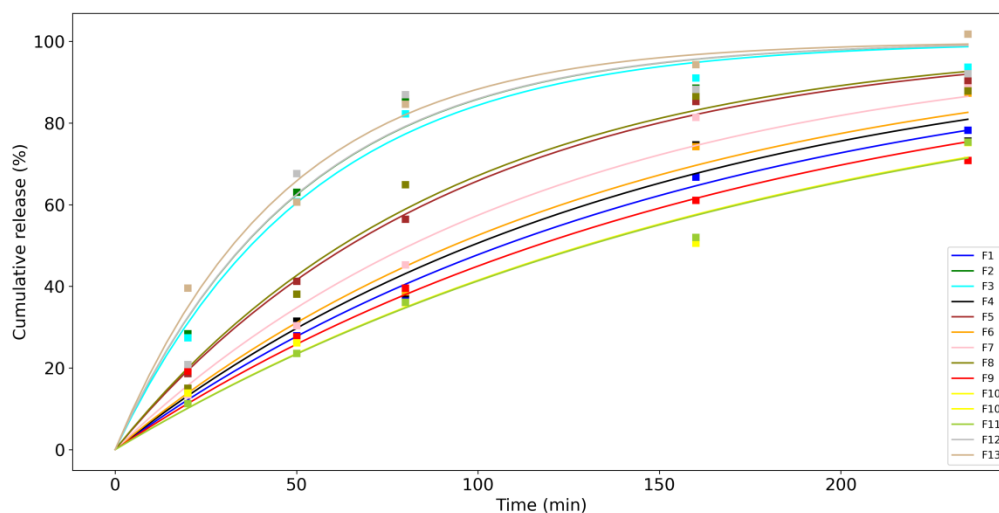


Figure 4: La cinétique de libération de premier ordre de Barbitol pour toutes les formulations (Tracé de la courbe théorique par rapport aux points expérimentaux).

PARAMETERS		F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13
ZERO ORDER MODEL	R <sup>2</sup> Ajust	0.89	-0.74	-0.32	0.78	0.66	0.93	0.87	0.59	0.72	0.89	0.93	-0.37	-0.65
	K0	0.37	0.51	0.52	0.39	0.46	0.41	0.44	0.47	0.35	0.34	0.34	0.52	0.55
	AIC	30.82	44.4	43.5	34.4	37.66	29.87	33.37	38.98	33.47	29.53	27.62	44.41	43.9
FIRST ORDER MODEL	R <sup>2</sup> Ajust	0.99	0.92	0.97	0.96	1	0.97	0.98	0.97	0.95	0.96	0.98	0.91	0.97
	K1	0.01	0.02	0.02	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.02	0.02
	AIC	16.68	28.7	23.7	25.4	15.51	25.02	24.75	25.76	24.52	24.15	21.35	30.9	23.0
HIGUCHI MODEL	R <sup>2</sup> Ajust	0.94	0.61	0.77	0.92	0.96	0.91	0.91	0.9	0.98	0.93	0.92	0.64	0.78
	KH	4.84	7.03	7.11	5.02	6.14	5.28	5.63	6.17	4.58	4.34	4.32	7.12	7.56
	AIC	28.11	36.9	34.84	29.5	27.28	31.59	31.47	31.97	19.91	27.15	28.74	37.67	33.8
KORSMEYER-PEPPAS	R <sup>2</sup> Ajust	0.99	0.7	0.8	0.93	0.95	0.98	0.96	0.87	0.99	0.98	0.99	0.64	0.91
	K	1.93	16.5	13.72	2.74	4.84	1.53	2.19	5.04	3.34	1.72	1.32	14.44	16.5
	n	0.68	0.33	0.37	0.62	0.55	0.75	0.69	0.54	0.56	0.69	0.74	0.36	0.34
	AIC	21.31	36.2	34.66	29.5	28.6	24.08	28.47	33.77	18.48	22.03	19.79	38.25	30.2
HIXSON-CROWOLL	R <sup>2</sup> Ajust	0.99	0.89	0.96	0.94	0.98	0.89	0.99	0.96	0.91	0.95	0.98	0.9	0.95
	KHC	0	0.05	0.05	0.02	0.03	0.02	0.02	0.03	0.02	0.02	0.02	0.05	0.06
	AIC	19.68	30.6	26.41	27.5	22.37	19.3	20.26	27.57	27.92	25.22	21.72	31.5	26.2

Tableau 4. Le R sq (Ajust) et AIC pour la cinétique de libération de Barbitol en utilisant différents modèles mathématiques obtenus par DD-Solver.

AIC: The Akaike Information Criterion

### 1.2 Approche de sélection du modèle bayésien

Le modèle d'ordre 1 présentait la valeur la plus basse de WAIC et de loo. Les erreurs-types (SE) variaient de 3 à 15. Le poids du modèle était de 0,97 et de 0,99 sur la base de la WAIC et des toilettes respectivement. La probabilité relative (poids) du modèle sélectionné était supérieure à 0,5, ce qui indique qu'aucun modèle concurrent n'était possible.

Loo et la WAIC estiment les mêmes critères de performance prédictive. La sélection du modèle commun représentait l'hypothèse H0, c'est-à-dire que K1 n'est pas affecté par les quantités des deux Eudragits, tandis que la sélection du modèle hiérarchique représentait l'hypothèse alternative, c'est-à-dire que K1 est affecté par les quantités des deux Eudragits. Selon les résultats, l'hypothèse alternative H1 a été acceptée, les quantités des deux Eudragits ont un effet significatif sur K1. (**Tableau 5 et 6**).

	<b>rang</b>	<b>WAIC</b>	<b>SE</b>	<b>p_waic</b>	<b>d_waic</b>	<b>Poids</b>
Premier_ordre_hierarchique	0	-169.64	6.15	11.92	0.00	0.97
Higuchi_hierarchique	1	-191.57	17.93	17.44	21.93	0.01
Korsemyer_hierarchique	2	-196.96	15.17	23.02	27.32	0.02
Première_ordre_partagée	3	-203.46	8.95	5.13	33.83	0.00
Zéro_ordre_hiérarchique	4	-240.27	12.16	28.97	70.64	0.00
Hixson_C. hierarchique	5	-272.72	2.97	0.34	103.08	0.00

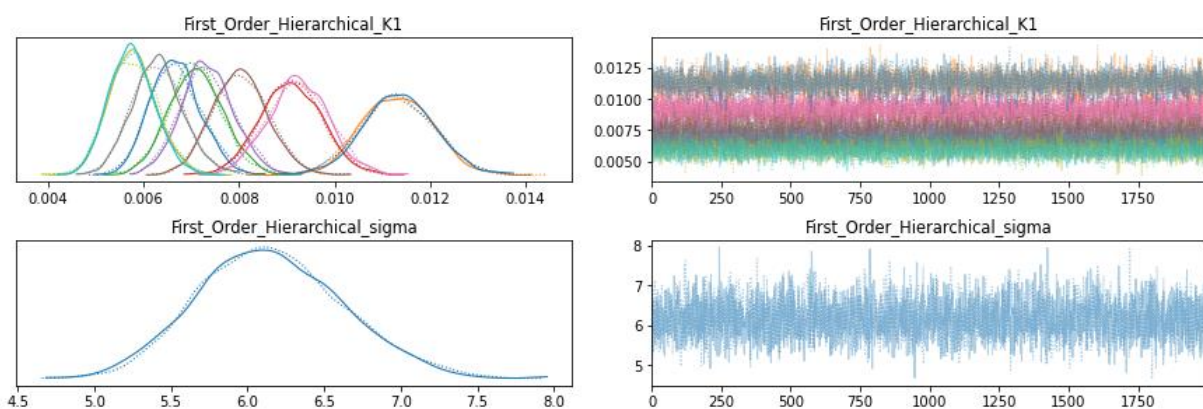
**Tableau 5** : Comparaison des modèles basée sur la WAIC, la pWAIC (Le nombre effectif de paramètres), la dWAIC (la différence entre les scores WAIC des autres modèles avec le meilleur modèle), le poids (la probabilité relative que le modèle explique les données).

	<b>rang</b>	<b>Loo</b>	<b>SE</b>	<b>p_loo</b>	<b>d_loo</b>	<b>poids</b>
Premier_ordre_hierarchique	0	-170.08	6.41	12.36	0.00	0.99
Higuchi_hierarchique	1	-192.76	16.97	18.63	22.68	0.01
Korsemyer_hierarchique	2	-199.53	13.48	25.59	29.45	0.00
Première_ordre_partagée	3	-203.46	8.66	5.13	33.38	0.00
Zéro_ordre_hiérarchique	4	-244.76	10.43	33.45	74.68	0.00
Hixson_C. hierarchique	5	-272.72	3.03	0.34	102.64	0.00

**Tableau 6** : Statistiques de comparaison des modèles, p\_loo, le nombre effectif de paramètres ; d\_loo (la différence entre les scores loo des autres modèles avec le meilleur modèle), poids (la probabilité relative que le modèle explique les données).

L'estimation bayésienne des paramètres du modèle sélectionné a été réalisée sur la base de la distribution normale postérieure de l'estimation. La chaîne de Markov s'est stabilisée comme l'indique le tracé (**Figure 5**). La distribution postérieure du paramètre du meilleur modèle ( $K_1$ ) était normale pour toutes les formulations. Le  $K_1$  estimé variait de 0,006 à 0,012 et son SE de 0,0001 à 0,001 (Données supplémentaires). Le SE du modèle était de 6,154 sur la base de la WAIC et de 6,4 sur la base de la LOO, ce qui confirme la solidité des résultats.

La LOO et la WAIC ont estimé les mêmes critères de performance prédictive et étaient asymptotiquement égales (Vehtari et Ojanen 2012 ; Gelman, Hwang et al. 2014 ; Vehtari, Mononen et al. 2016).



*Figure 5* : Tracé de la distribution postérieure du paramètre du meilleur modèle ( $K_1$ ) et de son écart-type (sigma).

**A gauche, sont tracées les distributions postérieures pour chaque paramètre et sa vraie valeur (ligne noire), et à droite, nous avons les valeurs individuelles échantillonnées à chaque pas de temps pendant l'échantillonnage. Ce sont nos distributions postérieures des paramètres que nous estimons, et les lignes verticales représentent les vraies valeurs. Nous pouvons voir que nous sommes assez proches de ces valeurs réelles.**

La méthode hiérarchique donne un seul modèle stable (modèle du premier ordre) pour une même série de formules alors que la méthode standard (Approche fréquentiste) donne plusieurs modèles. Les  $k_1$  du modèle du premier ordre estimé par la méthode fréquentiste et la méthode bayésienne sont fortement liés de manière significative (**Figure 6**). La méthode bayésienne a donné une bonne estimation des données observées (**Figure 7**) et a été utilisée pour une analyse plus approfondie. (Vander Plas 2014).



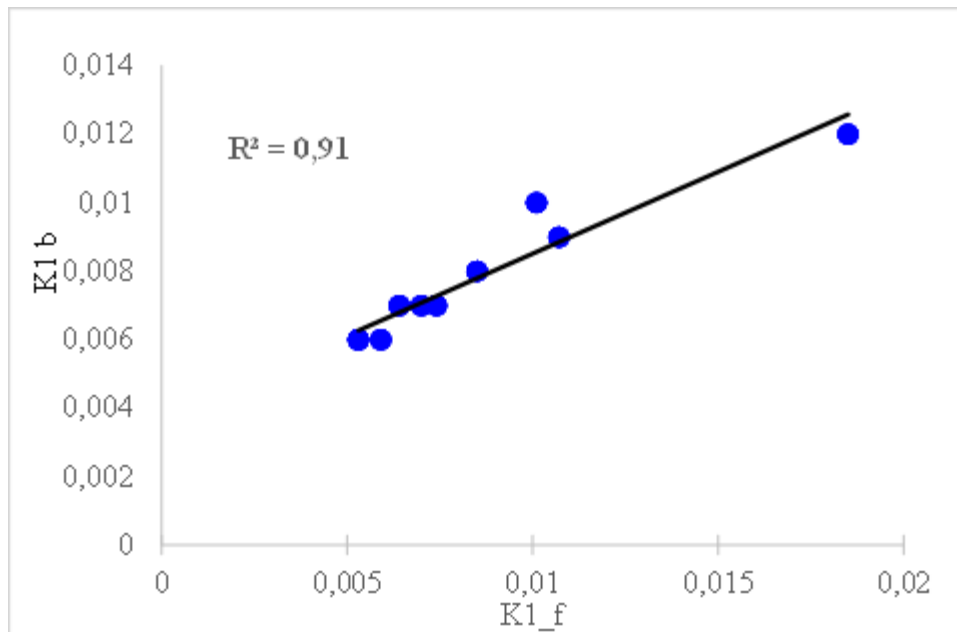


Figure 6 : K1 estimé par la méthode bayésienne, comparé au K<sub>1</sub> estimé par la méthode standard. (La valeur p et le coefficient de détermination (Rsq) ont été calculés dans Scipy en utilisant la technique de régression non linéaire des moindres carrés).

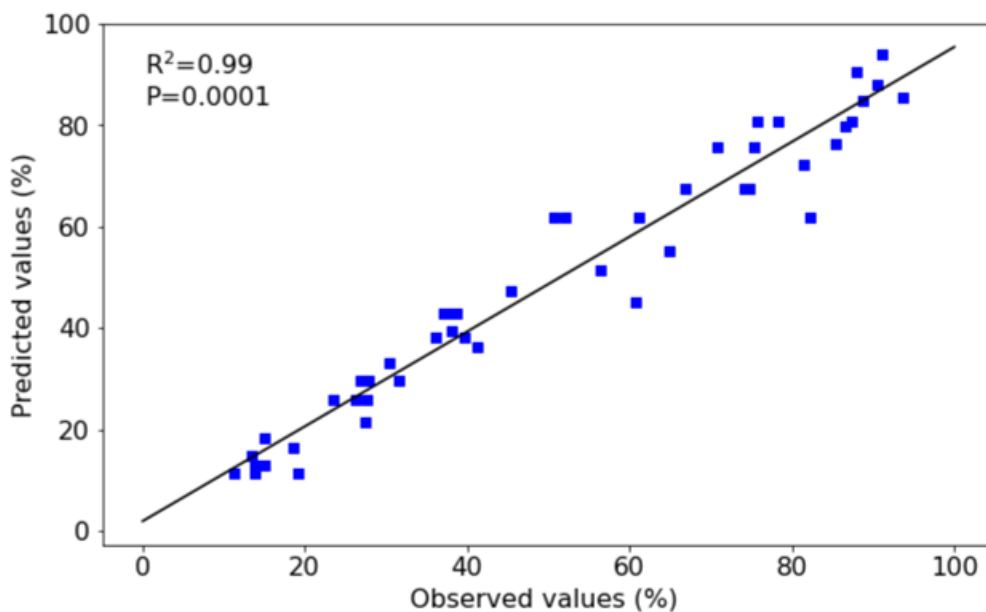


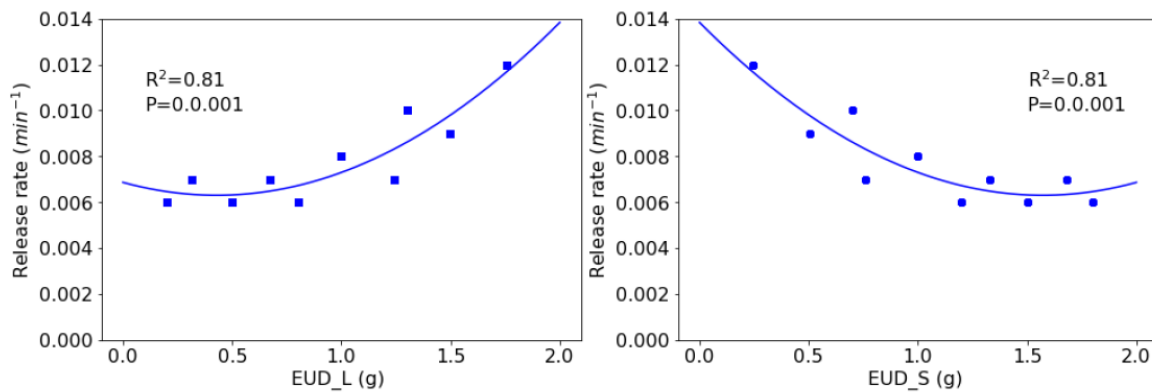
Figure 7: Résultats observés par rapport aux valeurs prévues par la méthode bayésienne (la valeur P et le coefficient de détermination (Rsq) ont été calculés dans Scipy en utilisant la technique de régression linéaire par les moindres carrés).

La relation entre K<sub>1</sub> et les quantités des deux Eudragits était quadratique (Equation 10), cette équation peut s'écrire pour EUD\_L comme suit :

$$Y (\% F) = 0,0031 X^2 - 0,0026 X + 0,0069 \quad (Rsq= 0,8214) \text{ (Eq.10)}$$

Le taux de libération ( $k_1$ ) était faible pour les valeurs d'Eudragit L jusqu'à 0,5 et augmente jusqu'à atteindre son maximum à 2 g de cet Eudragit, au contraire pour l'Eudragit S,  $K_1$  diminue de 0 (g) pour atteindre son minimum à 1,5 (g) (**Figure 8**).

Par cette équation qui met en relation le taux de dissolution avec la quantité de Eud\_L et donc le mélange des deux Eudragits (Total =10%) avec  $K = -2b/a = 0,0026/0,0031 = 0,41 \text{ min}^{-1}$ .



*Figure 8* : vitesse de libération ( $K_1$ ) basé sur les quantités de deux Eudragits L et S. (La valeur P et le coefficient de détermination ( $R_{sq}$ ) ont été calculés dans Scipy en utilisant la technique de régression linéaire des moindres carrés).

La libération de médicaments à partir de bases de suppositoires dépend généralement de la solubilité du PA dans la base et de la composition chimique de la base. Le Barbitol est un PA sous forme de sel de sodium ayant une grande affinité pour les bases hydrophiles, c'est-à-dire le PEG 6000. L'avantage du mélange des deux Eudragits réside dans la différence de solubilité des deux polymères dans la plage de  $\text{pH} = 7-8$  (systèmes  $\text{pH}$  dépendants), c'est-à-dire avec l'Eudragit L 100, le liquide intestinal soluble à partir de ( $\text{pH} 6$ ) et le S 100, le liquide intestinal soluble à partir de ( $\text{pH} 7$ ) (Di Colo, Falchi et al. 2002 ; Hong, Carlin et al. 2013).

Le modèle cinétique du premier ordre validé par la méthode bayésienne comme étant le modèle le plus adéquat pour décrire et prédire la libération de la matière active, ce résultat est dû à la matrice dissoute en fonction du  $\text{pH}$  de la région rectale.

Gibaldi et Feldman ont été les premiers à proposer des modèles cinétiques de libération de premier ordre en 1967, suivis par Wagner en 1969, pour ce modèle la vitesse de libération dépend de la concentration et  $\text{pH}$  dépendant.

#### IV CONCLUSION :

La méthode hiérarchique nous a permis de déduire que le modèle d'ordre 1 est le seul modèle qui nous permettra d'expliquer le mécanisme de libération des barbituriques de la matrice formée par les deux Eudragits 100 L et S, mais qu'il s'agit en même temps d'un outil permettant de prédire la composition optimale d'un suppositoire à effet prolongé.

Cette étude propose une manière originale et nouvelle de maîtriser la modélisation de la cinétique de la libération prolongée afin de minimiser le nombre d'expériences, de gagner du temps et d'économiser des produits chimiques coûteux.

## REFERENCES BIBLIOGRAPHIQUES :

1. Andrews, R. L., A. Ansari, et al. (2002). "Hierarchical Bayes versus finite mixture conjoint analysis models: A comparison of fit, prediction, and partworth recovery." *Journal of Marketing Research* 39(1): 87-98
2. Bates, B. C. and E. P. Campbell (2001). "A Markov chain Monte Carlo scheme for parameter estimation and inference in conceptual rainfall-runoff modeling." *Water resources research* 37(4): 937-947
3. Benomar, A., S. Yanisse, et al. (2020). "Application of design mixture and desirability function in the optimization of pharmaco-technical parameters of macrogols-based suppositories." *International Journal of Research in Pharmaceutical Sciences* 11(2): 1270-1281.
4. Bressert, E. (2012). *SciPy and NumPy: an overview for developers*, " O'Reilly Media, Inc."
5. Costa, P. and J. M. S. Lobo (2001). "Modeling and comparison of dissolution profiles." *European Journal of Pharmaceutical Sciences* 13(2): 123-133.
6. Cowles, M. K. and B. P. Carlin (1996). "Markov chain Monte Carlo convergence diagnostics: a comparative review." *Journal of the American Statistical Association* 91(434): 883-904.
7. Di Colo, G., S. Falchi, et al. (2002). "In vitro evaluation of a system for pH-controlled peroral delivery of metformin." *Journal of Controlled Release* 80(1-3): 119-128.
8. Faisant, N., J. Siepmann, et al. (2002). "PLGA-based microparticles: elucidation of mechanisms and a new, simple mathematical model quantifying drug release." *European Journal of Pharmaceutical Sciences* 15(4): 355-366.
9. Gelman, A., J. Hwang, et al. (2014). "Understanding predictive information criteria for Bayesian models." *Statistics and computing* 24(6): 997-1016.

10. Gouda, R., H. Baishya, et al. (2017). "Application of mathematical models in drug release kinetics of carbidopa and levodopa ER tablets." *J. Dev. Drugs* 6(02).
11. Grassi, M., G. Grassi, et al. (2006). *Understanding drug release and absorption mechanisms: a physical and mathematical approach*, CRC press.
12. Hong, H., B. P. Carlin, et al. (2013). "Comparing Bayesian and frequentist approaches for multiple outcome mixed treatment comparisons." *Medical Decision Making* 33(5): 702-714.
13. Joshi, M. (2013). "Role of eudragit in targeted drug delivery." *Int J Curr Pharm Res* 5(2): 58-62.
14. Kauss, T., K. Gaudin, et al. (2012). "Screening paediatric rectal forms of azithromycin as an alternative to oral or injectable treatment." *International journal of pharmaceutics* 436(1-2): 624-630.
15. Luo, Y. and K. Al-Harbi (2017). "Performances of LOO and WAIC as IRT model selection methods." *Psychological Test and Assessment Modeling* 59(2): 183.
16. Malekmohammadi, S., H. Hadadzadeh, et al. (2019). "Design and synthesis of gatekeeper coated dendritic silica/titania mesoporous nanoparticles with sustained and controlled drug release properties for targeted synergetic chemo-sonodynamic therapy." *ACS Biomaterials Science & Engineering* 5(9): 4405-4415.
17. Millar, R. B. (2009). "Comparison of hierarchical Bayesian models for overdispersed count data using DIC and Bayes' factors." *Biometrics* 65(3): 962-969.
18. Misra, A. (2014). *Applications of polymers in drug delivery*, Smithers Rapra.
19. Nakata, M. (2010). A numerical evaluation of highly accurate multiple-precision arithmetic version of semidefinite programming solver: SDPA-GMP,-QD and-DD. 2010 IEEE International Symposium on Computer-Aided Control System Design, IEEE.
20. Nikam, V. K., K. Kotade, et al. (2011). "Eudragit a versatile polymer: a review." *Pharmacologyonline* 1(5): 152-164.

21. Onyeji, C. O., A. S. Adebayo, et al. (1999). "Effects of absorption enhancers in chloroquine suppository formulations: I: In vitro release characteristics." *European Journal of Pharmaceutical Sciences* 9(2): 131-136.
22. Özgüney, I., İ. Özcan, et al. (2007). "The preparation and evaluation of sustained release suppositories containing ketoprofen and Eudragit RL 100 by using factorial design." *Pharmaceutical Development and Technology* 12(1): 97-107.
23. Paarakh, M. P., P. A. Jose, et al. (2018). "Release kinetics–concepts and applications." *Int. J. Pharm. Res. Tech* 8(1): 12-20.
24. Patil, A., D. Huard, et al. (2010). "PyMC: Bayesian stochastic modelling in Python." *Journal of statistical software* 35(4): 1.
25. Peppas, N. A. and B. Narasimhan (2014). "Mathematical models in drug delivery: How modeling has shaped the way we design new drug delivery systems." *Journal of Controlled Release* 190: 75-81.
26. Piray, P., A. Dezfouli, et al. (2019). "Hierarchical Bayesian inference for concurrent model fitting and comparison for group studies." *PLoS computational biology* 15(6): e1007043.
27. Ranjita, S. and S. Kamalinder (2010). "In-vitro release of paracetamol from suppository suppositories: role of additives." *IN-VITRO* 8(1): 57-71.
28. Salvatier, J., T. V. Wiecki, et al. (2016). "Probabilistic programming in Python using PyMC3." *PeerJ Computer Science* 2: e55.
29. Surti, N., A. Mahajan, et al. (2020). *Polymers in Rectal Drug Delivery. Applications of Polymers in Drug Delivery*, Elsevier: 263-280.
30. T. Güneri, M. A. a. G. E. (2004). "Sustained release suppositories of ibuprofen microspheres " *FABAD Journal of Pharmaceutical Sciences*.
31. Thakral, S., N. K. Thakral, et al. (2013). "Eudragit®: a technology evaluation." *Expert opinion on drug delivery* 10(1): 131-149.

32. VanderPlas, J. (2014). "Frequentism and bayesianism: a python-driven primer." arXiv preprint arXiv:1411.5018.
33. Varma, M. V., A. M. Kaushal, et al. (2004). "Factors affecting mechanism and kinetics of drug release from matrix-based oral controlled drug delivery systems." *American Journal of drug delivery* 2(1): 43-57.
34. Vehtari, A., A. Gelman, et al. (2015). "Efficient implementation of leave-one-out cross-validation and WAIC for evaluating fitted Bayesian models." arXiv preprint arXiv:1507.04544.
35. Vehtari, A., A. Gelman, et al. (2017). "Practical Bayesian model evaluation using leave-one-out cross-validation and WAIC." *Statistics and computing* 27(5): 1413-1432.
36. Vehtari, A., T. Mononen, et al. (2016). "Bayesian leave-one-out cross-validation approximations for Gaussian latent variable models." *The Journal of Machine Learning Research* 17(1): 3581-3618.
37. Vehtari, A. and J. Ojanen (2012). "A survey of Bayesian predictive methods for model assessment, selection and comparison." *Statistics Surveys* 6: 142-228.
38. Victor, O. and O. Francis (2017). "Evaluation of the kinetics and mechanism of piroxicam release from lipophilic and hydrophilic suppository bases." *Int J ChemTech Res* 10: 189-198.
39. Virtanen, P., R. Gommers, et al. (2020). "SciPy 1.0: fundamental algorithms for scientific computing in Python." *Nature methods* 17(3): 261-272.
40. Watanabe, S. and M. Opper (2010). "Asymptotic equivalence of Bayes cross validation and widely applicable information criterion in singular learning theory." *Journal of machine learning research* 11(12).
41. Wiecki, T. V., I. Sofer, et al. (2013). "HDDM: Hierarchical Bayesian estimation of the drift-diffusion model in Python." *Frontiers in neuroinformatics* 7: 14.
42. Zhang, Y., M. Huo, et al. (2010). "DDSolver: an add-in program for modeling and comparison of drug dissolution profiles." *The AAPS journal* 12(3): 263-271.

43. Zuo, J., Y. Gao, et al. (2014). "Evaluation of the DDSolver software applications."  
BioMed research international 2014.



## ANNEXES :

### ANNEXE 1 : Données bruts de l'analyse statistiques

Run	EUD_L	EUD_S	Ratio	Temps	yf	k1_f	K1_b	se	y pred_f	y pred_b
1	0.67	1.33	0.51	20	13.87	0.006	0.007	0.0001	12.015	13.064
1	0.67	1.33	0.51	50	27.89	0.006	0.007	0.0001	27.385	29.531
1	0.67	1.33	0.51	80	37.17	0.006	0.007	0.0001	40.070	42.879
1	0.67	1.33	0.51	160	66.8	0.006	0.007	0.0001	64.084	67.372
1	0.67	1.33	0.51	235	78.25	0.006	0.007	0.0001	77.776	80.699
3	1.76	0.24	7.24	20	27.42	0.019	0.012	0.001	30.927	21.337
3	1.76	0.24	7.24	50	60.77	0.019	0.012	0.001	60.347	45.119
3	1.76	0.24	7.24	80	82.27	0.019	0.012	0.001	77.236	61.711
3	1.76	0.24	7.24	160	93.71	0.019	0.012	0.001	94.818	85.339
3	1.76	0.24	7.24	235	91.08	0.019	0.012	0.001	98.706	94.039
4	0.32	1.68	0.19	20	14.95	0.007	0.007	0.0001	13.064	13.064
4	0.32	1.68	0.19	50	31.5	0.007	0.007	0.0001	29.531	29.531
4	0.32	1.68	0.19	80	38.25	0.007	0.007	0.0001	42.879	42.879
4	0.32	1.68	0.19	160	74.69	0.007	0.007	0.0001	67.372	67.372
4	0.32	1.68	0.19	235	75.62	0.007	0.007	0.0001	80.699	80.699
5	1.50	0.50	2.96	20	18.61	0.011	0.009	0.001	19.265	16.473
5	1.50	0.50	2.96	50	41.24	0.011	0.009	0.001	41.433	36.237
5	1.50	0.50	2.96	80	56.44	0.011	0.009	0.001	57.514	51.325
5	1.50	0.50	2.96	160	85.31	0.011	0.009	0.001	81.950	76.307
5	1.50	0.50	2.96	235	90.46	0.011	0.009	0.001	91.910	87.937
6	1.24	0.76	1.63	20	13.97	0.007	0.007	0.0001	13.757	13.064
6	1.24	0.76	1.63	50	26.91	0.007	0.007	0.001	30.927	29.531
6	1.24	0.76	1.63	80	38.66	0.007	0.007	0.001	44.678	42.879
6	1.24	0.76	1.63	160	74.23	0.007	0.007	0.001	69.395	67.372
6	1.24	0.76	1.63	235	87.37	0.007	0.007	0.001	82.430	80.699
7	1.00	1.00	1.00	20	13.4	0.009	0.008	0.001	15.634	14.786
7	1.00	1.00	1.00	50	30.41	0.009	0.008	0.001	34.623	32.968
7	1.00	1.00	1.00	80	45.26	0.009	0.008	0.001	49.338	47.271
7	1.00	1.00	1.00	160	81.44	0.009	0.008	0.001	74.334	72.196
7	1.00	1.00	1.00	235	88.66	0.009	0.008	0.001	86.433	84.741
8	1.30	0.70	1.86	20	15.05	0.01	0.01	0.001	18.291	18.127
8	1.30	0.70	1.86	50	38.1	0.01	0.01	0.001	39.649	39.347
8	1.30	0.70	1.86	80	64.95	0.01	0.01	0.001	55.425	55.067
8	1.30	0.70	1.86	160	86.6	0.01	0.01	0.001	80.131	79.810
8	1.30	0.70	1.86	235	87.89	0.01	0.01	0.001	90.685	90.463
9	0.20	1.80	0.11	20	19.23	0.006	0.006	0.0001	11.130	11.308
9	0.20	1.80	0.11	50	27.68	0.006	0.006	0.001	25.547	25.918
9	0.20	1.80	0.11	80	39.59	0.006	0.006	0.001	37.625	38.122

9	0.20	1.80	0.11	160	61.08	0.006	0.006	0.001	61.093	61.711
9	0.20	1.80	0.11	235	70.88	0.006	0.006	0.001	75.005	75.586
10	0.50	1.50	0.33	20	13.87	0.005	0.006	0.001	10.058	11.308
10	0.50	1.50	0.33	50	26.19	0.005	0.006	0.001	23.279	25.918
10	0.50	1.50	0.33	80	36.03	0.005	0.006	0.001	34.558	38.122
10	0.50	1.50	0.33	160	50.62	0.005	0.006	0.001	57.173	61.711
10	0.50	1.50	0.33	235	75.26	0.005	0.006	0.001	71.220	75.586
11	0.80	1.20	0.67	20	11.31	0.005	0.006	0.001	10.058	11.308
11	0.80	1.20	0.67	50	23.56	0.005	0.006	0.001	23.279	25.918
11	0.80	1.20	0.67	80	36.13	0.005	0.006	0.001	34.558	38.122
11	0.80	1.20	0.67	160	52.01	0.005	0.006	0.001	57.173	61.711
11	0.80	1.20	0.67	235	75.39	0.005	0.006	0.001	71.220	75.586

## ANNEXE 2 : Selection du modèle (Package Python)

In [1]:

```
#Importation des bibliothèques
```

```
!pip install arviz pymc3==3.8
import numpy as np
import scipy as sc
import math as mt
import pandas as pd
import matplotlib.pyplot as plt
import pymc3 as pm
import arviz as az
import theano as tt
#from statsmodels.formula.api import glm as glm_sm
#import statsmodels.api as sm
#from pymc3.glm import GLM
np.random.seed(42)
pm.set_tt_rng(42)
Requirement already satisfied: arviz in c:\programdata\anaconda3\lib\site-
packages (0.10.0)
Requirement already satisfied: pymc3==3.8 in
c:\programdata\anaconda3\lib\site-packages (3.8)
Requirement already satisfied: scipy>=0.19 in
c:\programdata\anaconda3\lib\site-packages (from arviz) (1.5.0)
Requirement already satisfied: matplotlib>=3.0 in
c:\programdata\anaconda3\lib\site-packages (from arviz) (3.2.2)
Requirement already satisfied: setuptools>=38.4 in
c:\programdata\anaconda3\lib\site-packages (from arviz)
(49.2.0.post20200714)
Requirement already satisfied: numpy>=1.12 in
c:\programdata\anaconda3\lib\site-packages (from arviz) (1.18.5)
Requirement already satisfied: netcdf4 in
c:\programdata\anaconda3\lib\site-packages (from arviz) (1.5.4)
Requirement already satisfied: packaging in
c:\programdata\anaconda3\lib\site-packages (from arviz) (20.4)
Requirement already satisfied: pandas>=0.23 in
c:\programdata\anaconda3\lib\site-packages (from arviz) (1.0.5)
Requirement already satisfied: xarray>=0.16.1 in
c:\programdata\anaconda3\lib\site-packages (from arviz) (0.16.1)
Requirement already satisfied: tqdm>=4.8.4 in
c:\programdata\anaconda3\lib\site-packages (from pymc3==3.8) (4.47.0)
Requirement already satisfied: h5py>=2.7.0 in
c:\programdata\anaconda3\lib\site-packages (from pymc3==3.8) (2.10.0)
Requirement already satisfied: patsy>=0.4.0 in
c:\programdata\anaconda3\lib\site-packages (from pymc3==3.8) (0.5.1)
Requirement already satisfied: theano>=1.0.4 in
c:\programdata\anaconda3\lib\site-packages (from pymc3==3.8) (1.0.5)
Requirement already satisfied: kiwisolver>=1.0.1 in
c:\programdata\anaconda3\lib\site-packages (from matplotlib>=3.0->arviz)
(1.2.0)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in
c:\programdata\anaconda3\lib\site-packages (from matplotlib>=3.0->arviz)
(2.4.7)
Requirement already satisfied: python-dateutil>=2.1 in
c:\programdata\anaconda3\lib\site-packages (from matplotlib>=3.0->arviz)
(2.8.1)
Requirement already satisfied: cyclor>=0.10 in
c:\programdata\anaconda3\lib\site-packages (from matplotlib>=3.0->arviz)
(0.10.0)
```

Requirement already satisfied: cftime in c:\programdata\anaconda3\lib\site-packages (from netcdf4->arviz) (1.2.1)  
 Requirement already satisfied: six in c:\programdata\anaconda3\lib\site-packages (from packaging->arviz) (1.15.0)  
 Requirement already satisfied: pytz>=2017.2 in c:\programdata\anaconda3\lib\site-packages (from pandas>=0.23->arviz) (2020.1)  
 WARNING (theano.configdefaults): g++ not available, if using conda: `conda install m2w64-toolchain`  
 C:\ProgramData\Anaconda3\lib\site-packages\theano\configdefaults.py:560: UserWarning: DeprecationWarning: there is no c++ compiler.This is deprecated and with Theano 0.11 a c++ compiler will be mandatory  
 warnings.warn("DeprecationWarning: there is no c++ compiler."  
 WARNING (theano.configdefaults): g++ not detected ! Theano will be unable to execute optimized C-implementations (for both CPU and GPU) and will default to Python implementations. Performance will be severely degraded. To remove this warning, set Theano flags cxx to an empty string.  
 WARNING (theano.tensor.blas): Using NumPy C-API based implementation for BLAS functions.

In [2]:

```
import pymc3 as pm
print("Running on PyMC3 v{}".format(pm.__version__))

from platform import python_version

print(python_version())
Running on PyMC3 v3.8
3.8.3
```

## 1- importation des données

In [77]:

```
result_1 = pd.read_excel("results.xlsx")
result_1[["run", "EUD_L", "EUD_S", "ratio", "TIME", "yf"]].head()
```

Out[77]:

	run	EUD_L	EUD_S	ratio	TIME	yf
0	1	0.673877	1.326123	0.508156	20	13.87
1	1	0.673877	1.326123	0.508156	50	27.89
2	1	0.673877	1.326123	0.508156	80	37.17
3	1	0.673877	1.326123	0.508156	160	66.80
4	1	0.673877	1.326123	0.508156	235	78.25

## 2- creation des fonction des modèles

In [78]:

```
def firs_order (K1, x):
    cum=100*(1-np.exp(-K1*x))
    return cum
```

```

def Korsemyer_fun(a, b, x):
    cum=a*(x**b)
    return cum

def Higuchi_fun (a, x):
    cum=a*(x**0.5)
    return cum

def Hixson_fun (a, x):
    cum=1-((1-a*x)**3)
    return cum

```

### 3- indexation pour l'hierarchisation des modèles¶

In [80]:

```

runs = result_1["run"].unique()
n_run = len(runs)
idx_runs = pd.Index(runs).get_indexer(result_1.run)

```

### 4- creation des models bayisien avec hierarchisation ou non¶

In [82]:

```

#### zero model
zero_pooled = pm.Model("zero_pooled")
with zero_pooled:
    x= pm.Data("TIME", result_1['TIME'])
    yf = pm.Data("y_data", result_1['yf'])
    a = pm.Normal("a", mu=1, sigma=5)
    sigma = pm.HalfNormal("sigma")
    mu = a*x
    yf=pm.Normal('yf', mu=mu, sigma=sigma, observed=yf)
zero_hierarchical = pm.Model("zero_hierarchical")
with zero_hierarchical:
    x= pm.Data("TIME", result_1['TIME'])
    yf = pm.Data("y_data", result_1['yf'])
    a = pm.Normal("a", mu=1, sigma=5, shape=n_run)
    sigma = pm.HalfNormal("sigma")
    mu = a[idx_runs]*x
    yf=pm.Normal('yf', mu=mu, sigma=sigma, observed=yf)
##### the firs_orde model
F_Order_pooled = pm.Model("F_Order_pooled")
with F_Order_pooled:
    x= pm.Data("x_data", result_1['TIME'])
    yf = pm.Data("y_data", result_1['yf'])
    BoundedNormal = pm.Bound(pm.Normal, lower=0.000, upper=0.03)
    a = BoundedNormal("a", mu=0.007, sigma=0.001)
    sigma = pm.HalfNormal("sigma")
    mu = firs_order(a, x)
    yf=pm.Normal('yf', mu=mu, sigma=sigma, observed=yf)

F_Order_hierarchical = pm.Model("F_Order_hierarchical")
with F_Order_hierarchical:
    x= pm.Data("x_data", result_1['TIME'])
    yf = pm.Data("y_data", result_1['yf'])
    BoundedNormal = pm.Bound(pm.Normal, lower=0.000, upper=0.028)
    a = BoundedNormal("a", mu=0.007, sigma=0.001, shape=n_run)
    sigma = pm.HalfNormal("sigma")
    mu = firs_order(a[idx_runs], x)
    yf=pm.Normal('yf', mu=mu, sigma=sigma, observed=yf)

```

```

#####KP model
#####

Korsemyer_pooled = pm.Model("Korsemyer_Pooled")
with Korsemyer_pooled:
    x= pm.Data("TIME", result_1['TIME'])
    yf = pm.Data("y_data", result_1['yf'])
    a = pm.Normal("a", mu=10, sigma=5)
    b = pm.Normal("b", mu=0.5, sigma=0.3)
    sigma = pm.HalfNormal("sigma")
    mu = Korsemyer_fun(a, b, x)
    yf=pm.Normal('yf', mu=mu, sigma=sigma, observed=yf)
Korsemyer_hierarchical = pm.Model("Korsemyer_hierarchical")
with Korsemyer_hierarchical:
    x= pm.Data("x_data", result_1['TIME'])
    yf = pm.Data("y_data", result_1['yf'])
    a = pm.Normal("a", mu=10, sigma=2, shape=n_run)
    b = pm.Normal("b", mu=0.7, sigma=0.2, shape=n_run)
    sigma = pm.HalfNormal("sigma")
    cumm_r = Korsemyer_fun(a[idx_runs], b[idx_runs], x)
    yf=pm.Normal('yf', mu=cumm_r, sigma=sigma, observed=yf)

#####Higuchi model
#####
Higuchi_pooled = pm.Model("Higuchi_Pooled")
with Higuchi_pooled:
    x= pm.Data("x_data", result_1['TIME'])
    yf = pm.Data("y_data", result_1['yf'])
    a = pm.Normal("a", mu=10, sigma=5)

    sigma = pm.HalfNormal("sigma")
    mu = Higuchi_fun(a, x)
    yf=pm.Normal('yf', mu=mu, sigma=sigma, observed=yf)

Higuchi_hierarchical = pm.Model("Higuchi_hierarchical")
with Higuchi_hierarchical:
    x= pm.Data("x_data", result_1['TIME'])
    yf = pm.Data("y_data", result_1['yf'])
    a = pm.Normal("a", mu=12, sigma=10, shape=n_run)

    sigma = pm.HalfNormal("sigma")
    test = Higuchi_fun( a[idx_runs], x)
    yf=pm.Normal('yf', mu=test, sigma=sigma, observed=yf)
#pm.model_to_graphviz(Higuchi1_per_run_model)
##### Hixson model
#####

def Hixson_fun (a, x):
    cumm2=1-((1-a*x)**3)
    return cumm2
Hixson_pooled = pm.Model("Hixson_pooled")
with Hixson_pooled:
    x= pm.Data("x_data", result_1['TIME'])
    yf = pm.Data("y_data", result_1['yf'])
    BoundedNormal = pm.Bound(pm.Normal, lower=0.004, upper=0.006)
    a = BoundedNormal("a", mu=0.005, sigma=0.001)
    sigma = pm.HalfNormal("sigma", sd=50)
    cumm_r = Hixson_fun(a, x)
    yf=pm.Normal('yf', mu=100-cumm_r, sigma=sigma, observed=yf)

```

```
Hixson_hierarchical = pm.Model("Hixson_hierarchical")
with Hixson_hierarchical:
    x = pm.Data("x_data", result_1['TIME'])
    yf = pm.Data("y_data", result_1['yf'])
    BoundedNormal = pm.Bound(pm.Normal, lower=0.004, upper=0.006)
    a = BoundedNormal("a", mu=0.005, sigma=0.001, shape=n_run)
    sigma = pm.HalfNormal("sigma", sd=50)
    cumm_r = Hixson_fun(a[idx_runs], x)
    yf=pm.Normal('yf', mu=100-cumm_r, sigma=sigma, observed=yf)
```

## In [16]:

```
with zero_hierarchical:
    zero_hierarchical1_trace = pm.sample(2000, init='advi',
nuts_kwargs={"target_accept":0.9,"max_treedepth": 15})

#####
with F_Order_pooled:
    F_Order_pooled_trace = pm.sample(2000, init='advi',
nuts_kwargs={"target_accept":0.9,"max_treedepth": 15})

with F_Order_hierarchical:
    F_Order_hierarchical_trace = pm.sample(2000, init='advi',
nuts_kwargs={"target_accept":0.9,"max_treedepth": 15})

#####
with Korsemeyer_hierarchical:
    Korsemeyer_hierarchical_trace = pm.sample(2000, init='advi',
nuts_kwargs={"target_accept":0.9,"max_treedepth": 15})
#####

with Higuchi_hierarchical:
    Higuchi_hierarchical1_trace = pm.sample(2000, init='advi',
nuts_kwargs={"target_accept":0.9,"max_treedepth": 15})
#####
with Hixson_hierarchical:
    Hixson_hierarchical_trace = pm.sample(2000, init='advi',
nuts_kwargs={"target_accept":0.9,"max_treedepth": 15})
Auto-assigning NUTS sampler...
Initializing NUTS using advi...
Average Loss = 384.06: 7%|██████████| 13299/200000 [03:15<45:41,
68.11it/s]
Convergence achieved at 13300
Interrupted at 13,299 [6%]: Average Loss = 1.417e+05
Multiprocess sampling (2 chains in 2 jobs)
NUTS: [linear_hierarchical1_sigma, linear_hierarchical1_a]
Sampling 2 chains, 0 divergences:
100%|██████████| 5000/5000
[03:01<00:00, 27.48draws/s]
Auto-assigning NUTS sampler...
Initializing NUTS using advi...
Average Loss = 272.35: 6%|██████████| 12699/200000 [02:30<37:02,
84.26it/s]
Convergence achieved at 12700
Interrupted at 12,699 [6%]: Average Loss = 6,267.7
Multiprocess sampling (2 chains in 2 jobs)
NUTS: [F_Order_pooled_sigma, F_Order_pooled_a]
Sampling 2 chains, 0 divergences:
100%|██████████| 5000/5000
[02:14<00:00, 37.12draws/s]
```

Auto-assigning NUTS sampler...  
 Initializing NUTS using advi...  
 Average Loss = 248.48: 6%|██████ | 12299/200000 [03:28<52:58,  
 59.05it/s]  
 Convergence achieved at 12300  
 Interrupted at 12,299 [6%]: Average Loss = 4,683.8  
 Multiprocess sampling (2 chains in 2 jobs)  
 NUTS: [F\_Order\_hierarchical\_sigma, F\_Order\_hierarchical\_a]  
 Sampling 2 chains, 0 divergences:  
 100%|██| 5000/5000  
 [04:27<00:00, 18.67draws/s]  
 Auto-assigning NUTS sampler...  
 Initializing NUTS using advi...  
 Average Loss = 306.04: 17%|██████████ | 33899/200000 [10:36<51:57,  
 53.28it/s]  
 Convergence achieved at 33900  
 Interrupted at 33,899 [16%]: Average Loss = 2.9303e+13  
 Multiprocess sampling (2 chains in 2 jobs)  
 NUTS: [Korsemyer\_hierarchical2\_sigma, Korsemyer\_hierarchical2\_b,  
 Korsemyer\_hierarchical2\_a]  
 Sampling 2 chains, 16 divergences:  
 100%|██| 5000/5000  
 [10:36:05<00:00, 7.63s/draws]  
 There were 7 divergences after tuning. Increase `target\_accept` or  
 reparameterize.  
 There were 9 divergences after tuning. Increase `target\_accept` or  
 reparameterize.  
 Auto-assigning NUTS sampler...  
 Initializing NUTS using advi...  
 Average Loss = 264.89: 15%|██████████ | 29399/200000 [06:14<36:12,  
 78.51it/s]  
 Convergence achieved at 29400  
 Interrupted at 29,399 [14%]: Average Loss = 30,649  
 Multiprocess sampling (2 chains in 2 jobs)  
 NUTS: [Higuchi\_hierarchical1\_sigma, Higuchi\_hierarchical1\_a]  
 Sampling 2 chains, 0 divergences:  
 100%|██| 5000/5000  
 [02:59<00:00, 27.85draws/s]  
 Auto-assigning NUTS sampler...  
 Initializing NUTS using advi...  
 Average Loss = 304.75: 5%|██████ | 10999/200000 [02:59<51:26,  
 61.23it/s]  
 Convergence achieved at 11000  
 Interrupted at 10,999 [5%]: Average Loss = 317.43  
 Multiprocess sampling (2 chains in 2 jobs)  
 NUTS: [Hixson\_hierarchical\_sigma, Hixson\_hierarchical\_a]  
 Sampling 2 chains, 0 divergences:  
 100%|██| 5000/5000  
 [04:16<00:00, 19.47draws/s]  
 C:\ProgramData\Anaconda3\lib\site-packages\arviz\stats\stats.py:149:  
 UserWarning:  
 The scale is now log by default. Use 'scale' argument or 'stats.ic\_scale'  
 rcParam if you rely on a specific value.  
 A higher log-score (or a lower deviance) indicates a model with better  
 predictive accuracy.  
 warnings.warn(  
 C:\ProgramData\Anaconda3\lib\site-packages\arviz\data\io\_pymc3.py:87:  
 FutureWarning: Using `from\_pymc3` without the model will be deprecated in a  
 future release. Not using the model will return less accurate and less



```

useful results. Make sure you use the model argument or call from_pymc3
within a model context.
    warnings.warn(
C:\ProgramData\Anaconda3\lib\site-packages\arviz\stats\stats.py:683:
UserWarning: Estimated shape parameter of Pareto distribution is greater
than 0.7 for one or more samples. You should consider using a more robust
model, this is because importance sampling is less likely to work well if
the marginal posterior and LOO posterior are very different. This is more
likely to happen with a non-robust model and highly influential
observations.
    warnings.warn(
C:\ProgramData\Anaconda3\lib\site-packages\arviz\data\io_pymc3.py:87:
FutureWarning: Using `from_pymc3` without the model will be deprecated in a
future release. Not using the model will return less accurate and less
useful results. Make sure you use the model argument or call from_pymc3
within a model context.
    warnings.warn(
C:\ProgramData\Anaconda3\lib\site-packages\arviz\data\io_pymc3.py:87:
FutureWarning: Using `from_pymc3` without the model will be deprecated in a
future release. Not using the model will return less accurate and less
useful results. Make sure you use the model argument or call from_pymc3
within a model context.
    warnings.warn(
C:\ProgramData\Anaconda3\lib\site-packages\arviz\data\io_pymc3.py:87:
FutureWarning: Using `from_pymc3` without the model will be deprecated in a
future release. Not using the model will return less accurate and less
useful results. Make sure you use the model argument or call from_pymc3
within a model context.
    warnings.warn(
C:\ProgramData\Anaconda3\lib\site-packages\arviz\stats\stats.py:683:
UserWarning: Estimated shape parameter of Pareto distribution is greater
than 0.7 for one or more samples. You should consider using a more robust
model, this is because importance sampling is less likely to work well if
the marginal posterior and LOO posterior are very different. This is more
likely to happen with a non-robust model and highly influential
observations.
    warnings.warn(
C:\ProgramData\Anaconda3\lib\site-packages\arviz\data\io_pymc3.py:87:
FutureWarning: Using `from_pymc3` without the model will be deprecated in a
future release. Not using the model will return less accurate and less
useful results. Make sure you use the model argument or call from_pymc3
within a model context.
    warnings.warn(
C:\ProgramData\Anaconda3\lib\site-packages\arviz\stats\stats.py:683:
UserWarning: Estimated shape parameter of Pareto distribution is greater
than 0.7 for one or more samples. You should consider using a more robust
model, this is because importance sampling is less likely to work well if
the marginal posterior and LOO posterior are very different. This is more
likely to happen with a non-robust model and highly influential
observations.
    warnings.warn(
C:\ProgramData\Anaconda3\lib\site-packages\arviz\data\io_pymc3.py:87:
FutureWarning: Using `from_pymc3` without the model will be deprecated in a
future release. Not using the model will return less accurate and less
useful results. Make sure you use the model argument or call from_pymc3
within a model context.
    warnings.warn(
C:\ProgramData\Anaconda3\lib\site-packages\arviz\stats\stats.py:149:
UserWarning:
The scale is now log by default. Use 'scale' argument or 'stats.ic_scale'
rcParam if you rely on a specific value.

```

A higher log-score (or a lower deviance) indicates a model with better predictive accuracy.

```
warnings.warn(  
C:\ProgramData\Anaconda3\lib\site-packages\arviz\data\io_pymc3.py:87:  
FutureWarning: Using `from_pymc3` without the model will be deprecated in a  
future release. Not using the model will return less accurate and less  
useful results. Make sure you use the model argument or call from_pymc3  
within a model context.  
warnings.warn(  
C:\ProgramData\Anaconda3\lib\site-packages\arviz\stats\stats.py:1425:  
UserWarning: For one or more samples the posterior variance of the log  
predictive densities exceeds 0.4. This could be indication of WAIC starting  
to fail.  
See http://arxiv.org/abs/1507.04544 for details  
warnings.warn(  
C:\ProgramData\Anaconda3\lib\site-packages\arviz\data\io_pymc3.py:87:  
FutureWarning: Using `from_pymc3` without the model will be deprecated in a  
future release. Not using the model will return less accurate and less  
useful results. Make sure you use the model argument or call from_pymc3  
within a model context.  
warnings.warn(  
C:\ProgramData\Anaconda3\lib\site-packages\arviz\stats\stats.py:1425:  
UserWarning: For one or more samples the posterior variance of the log  
predictive densities exceeds 0.4. This could be indication of WAIC starting  
to fail.  
See http://arxiv.org/abs/1507.04544 for details  
warnings.warn(  
C:\ProgramData\Anaconda3\lib\site-packages\arviz\data\io_pymc3.py:87:  
FutureWarning: Using `from_pymc3` without the model will be deprecated in a  
future release. Not using the model will return less accurate and less  
useful results. Make sure you use the model argument or call from_pymc3  
within a model context.  
warnings.warn(  
C:\ProgramData\Anaconda3\lib\site-packages\arviz\stats\stats.py:1425:  
UserWarning: For one or more samples the posterior variance of the log  
predictive densities exceeds 0.4. This could be indication of WAIC starting  
to fail.  
See http://arxiv.org/abs/1507.04544 for details  
warnings.warn(  
C:\ProgramData\Anaconda3\lib\site-packages\arviz\data\io_pymc3.py:87:  
FutureWarning: Using `from_pymc3` without the model will be deprecated in a  
future release. Not using the model will return less accurate and less  
useful results. Make sure you use the model argument or call from_pymc3  
within a model context.  
warnings.warn(  
C:\ProgramData\Anaconda3\lib\site-packages\arviz\stats\stats.py:1425:  
UserWarning: For one or more samples the posterior variance of the log  
predictive densities exceeds 0.4. This could be indication of WAIC starting  
to fail.  
See http://arxiv.org/abs/1507.04544 for details  
warnings.warn(  
C:\ProgramData\Anaconda3\lib\site-packages\arviz\data\io_pymc3.py:87:  
FutureWarning: Using `from_pymc3` without the model will be deprecated in a  
future release. Not using the model will return less accurate and less  
useful results. Make sure you use the model argument or call from_pymc3  
within a model context.  
warnings.warn(  
C:\ProgramData\Anaconda3\lib\site-packages\arviz\stats\stats.py:1425:  
UserWarning: For one or more samples the posterior variance of the log  
predictive densities exceeds 0.4. This could be indication of WAIC starting  
to fail.
```

See <http://arxiv.org/abs/1507.04544> for details

```
warnings.warn(  
C:\ProgramData\Anaconda3\lib\site-packages\arviz\data\io_pymc3.py:87:  
FutureWarning: Using `from_pymc3` without the model will be deprecated in a  
future release. Not using the model will return less accurate and less  
useful results. Make sure you use the model argument or call from_pymc3  
within a model context.  
warnings.warn(  

```

	rank	waic	p_waic	d_waic	weight	\
F_Order_hierarchical	0	-203.7	12.6927	0	0.845682	
Higuchi_hierarchical1	1	-217.928	20.5306	14.2282	0.0855988	
Korsemyer_hierarchical2	2	-221.142	24.3939	17.4421	0.0687191	
F_Order_pooled	3	-232.817	4.7138	29.117	1.3923e-09	
linear_hierarchical1	4	-276.195	32.2572	72.4947	1.14243e-20	
Hixson_hierarchical	5	-298.535	0.393793	94.8353	1.83792e-24	

	se	dse	warning	waic_scale
F_Order_hierarchical	7.10545	0	True	log
Higuchi_hierarchical1	19.5698	12.2213	True	log
Korsemyer_hierarchical2	14.985	12.8329	True	log
F_Order_pooled	10.908	4.71199	True	log
linear_hierarchical1	14.7523	13.3313	True	log
Hixson_hierarchical	3.14894	16.2521	False	log

	rank	loo	p_loo	d_loo	weight	\
F_Order_hierarchical	0	-204.143	13.1359	0	0.905207	
Higuchi_hierarchical1	1	-219.511	22.1133	15.3676	0.0570819	
Korsemyer_hierarchical2	2	-224.177	27.4287	20.0337	0.0377111	
F_Order_pooled	3	-232.801	4.69829	28.6582	4.68916e-10	
linear_hierarchical1	4	-279.666	35.7291	75.5234	4.2792e-22	
Hixson_hierarchical	5	-298.535	0.394126	94.3925	7.75609e-23	

	se	dse	warning	loo_scale
F_Order_hierarchical	7.49999	0	False	log
Higuchi_hierarchical1	20.2158	12.6661	True	log
Korsemyer_hierarchical2	14.5566	13.2791	True	log
F_Order_pooled	11.4479	4.72151	False	log
linear_hierarchical1	14.5724	14.1223	True	log
Hixson_hierarchical	3.3436	16.3797	False	log

In [ ]:

```
comparison = pm.compare (  
  
    zero_hierarchical1.name:zero_hierarchical1_trace,  
  
    F_Order_pooled.name:F_Order_pooled_trace,  
    F_Order_hierarchical.name:F_Order_hierarchical_trace,  
  
    Korsemyer_hierarchical2.name: Korsemyer_hierarchical2_trace,  
  
    Higuchi_hierarchical1.name: Higuchi_hierarchical1_trace,  
  
    Hixson_hierarchical.name:Hixson_hierarchical_trace  
    }, ic='loo');  
  
comparison1 = pm.compare( {  
  
    zero_hierarchical1.name:zero_hierarchical1_trace,  
    F_Order_pooled.name:F_Order_pooled_trace,  
    F_Order_hierarchical.name:F_Order_hierarchical_trace,  
  
    Korsemyer_hierarchical2.name: Korsemyer_hierarchical2_trace,  

```

```
Higuchi_hierarchical1.name: Higuchi_hierarchical1_trace,

Hixson_hierarchical.name:Hixson_hierarchical_trace

}, ic='waic');
print(comparison1)

print(comparison)
In [ ]:

writer = pd.ExcelWriter('model_sel_loo_Dec.xlsx', engine='xlsxwriter')
# Convert the dataframe to an XlsxWriter Excel object.
comparison.to_excel(writer, sheet_name='Sheet1')
comparison1.to_excel(writer, sheet_name='Sheet2')
# Close the Pandas Excel writer and output the Excel file.
summary_fo.to_excel(writer, sheet_name='Sheet3')
writer.save()
```

In [1]:

```
#pm.compareplot(comparison1);
#fig = plt.gcf() # to get the current figure...
#fig.savefig('model_sel_2020.png', dpi=300)
```

In [19]:

```
def firs_order (K1, x):
    cumm2=100*(1-np.exp(-K1*x))
    return cumm2

First_Order_Hierarchical = pm.Model("First_Order_Hierarchical")
with First_Order_Hierarchical:
    x= pm.Data("x_data", result_1['TIME'])
    yf = pm.Data("y_data", result_1['yf'])
    BoundedNormal = pm.Bound(pm.Normal, lower=0.000, upper=0.028)
    K1 = BoundedNormal("K1", mu=0.007, sigma=0.001, shape=n_run)
    sigma = pm.HalfNormal("sigma")
    mu = firs_order(K1[idx_runs], x)
    yf=pm.Normal('yf', mu=mu, sigma=sigma, observed=yf)

with First_Order_Hierarchical:
    First_Order_Hierarchical_trace = pm.sample(2000, init='advi',
nuts_kwargs={"target_accept":0.9,"max_treedepth": 15})
    pm.traceplot(First_Order_Hierarchical_trace);
    fig = plt.gcf() # to get the current figure...
    fig.savefig('Trace_plot_VF.png', dpi=300)
    Auto-assigning NUTS sampler...
    Initializing NUTS using advi...
    Average Loss = 247.81: 6%|██████| 12399/200000 [03:49<57:52,
54.03it/s]
    Convergence achieved at 12400
    Interrupted at 12,399 [6%]: Average Loss = 4,396.9
    Multiprocess sampling (2 chains in 2 jobs)
    NUTS: [First_Order_Hierarchical_sigma, First_Order_Hierarchical_K1]
    Sampling 2 chains, 0 divergences:
    100%|██████████████████████████████████████████████████████████████| 5000/5000
[04:56<00:00, 16.86draws/s]
C:\ProgramData\Anaconda3\lib\site-packages\arviz\data\io_pymc3.py:87:
FutureWarning: Using `from_pymc3` without the model will be deprecated in a
future release. Not using the model will return less accurate and less
useful results. Make sure you use the model argument or call from_pymc3
within a model context.
```

```
warnings.warn(
```

In [7]:

```
pm.forestplot(First_Order_Hierarchical_trace,  
var_names=['First_Order_Hierarchical_K1']);  
fig = plt.gcf() # to get the current figure...  
fig.savefig('model_params_dec_2020.png', dpi=300)
```

In [20]:

```
# results sous form de dataframe  
df1=pm.summary(First_Order_Hierarchical_trace,  
var_names=['First_Order_Hierarchical_K1'])  
df1  
##### exporter les données df1 sous format excel  
writer = pd.ExcelWriter('First_Order_Hierarchical_K1_a.xlsx',  
engine='xlsxwriter')  
# Convert the dataframe to an XlsxWriter Excel object.  
df1.to_excel(writer, sheet_name='Sheet1')  
# Close the Pandas Excel writer and output the Excel file.  
writer.save()  
C:\ProgramData\Anaconda3\lib\site-packages\arviz\data\io_pymc3.py:87:  
FutureWarning: Using `from_pymc3` without the model will be deprecated in a  
future release. Not using the model will return less accurate and less  
useful results. Make sure you use the model argument or call from_pymc3  
within a model context.  
warnings.warn(
```

In [21] :

Out[21] :

	mean	sd	hdi_3%	hdi_97%	mcse_mean	mcse_sd	ess_mean	ess_sd	ess_bulk	ess_tail	r_hat
First_Order_Hierarchical_K1[0]	0.007	0.001	0.006	0.008	0.0	0.0	6330.0	6177.0	6366.0	2754.0	1.0
First_Order_Hierarchical_K1[1]	0.011	0.001	0.010	0.013	0.0	0.0	4986.0	4986.0	4947.0	3126.0	1.0
First_Order_Hierarchical_K1[2]	0.007	0.001	0.006	0.008	0.0	0.0	5812.0	5652.0	5848.0	3327.0	1.0
First_Order_Hierarchical_K1[3]	0.009	0.001	0.008	0.010	0.0	0.0	5261.0	5261.0	5266.0	2926.0	1.0
First_Order_Hierarchical_K1[4]	0.007	0.001	0.006	0.008	0.0	0.0	6429.0	6429.0	6421.0	3437.0	1.0
First_Order_Hierarchical_K1[5]	0.008	0.001	0.007	0.009	0.0	0.0	6320.0	6103.0	6416.0	2693.0	1.0
First_Order_Hierarchical_K1[6]	0.009	0.001	0.008	0.010	0.0	0.0	4752.0	4710.0	4752.0	3205.0	1.0
First_Order_Hierarchical_K1[7]	0.006	0.001	0.005	0.007	0.0	0.0	7692.0	7390.0	7824.0	3169.0	1.0
First_Order_Hierarchical_K1[8]	0.006	0.001	0.005	0.007	0.0	0.0	6976.0	6810.0	7000.0	2832.0	1.0
First_Order_Hierarchical_K1[9]	0.006	0.000	0.005	0.007	0.0	0.0	7014.0	6803.0	7072.0	2820.0	1.0
First_Order_Hierarchical_K1[10]	0.011	0.001	0.010	0.013	0.0	0.0	5052.0	5052.0	5062.0	2892.0	1.0

## Relation entre Eud\_S et EUD\_L avec K1

In [45]:

```
result_1 = pd.read_excel("results_1.xlsx")
result_1.sort_values(['EUD_L', 'K1_HB'], inplace=True, ascending=True)
from scipy.optimize import curve_fit
def func(x, a, b, c):
    return a * x**2 - b * x + c
x1data2 = np.linspace(0, 2, 50)
x2data2 = np.linspace(0, 2, 50)
x1data=result_1['EUD_L']
x2data=result_1['EUD_S']
yerr=result_1['se']

#xdata=result_1['EUD_L']
ydata=result_1['K1_HB']
popt1, pcov1 = curve_fit(func, x1data, ydata)
a_opt, b_opt, c_opt = popt1
R2 = np.sum((func(x1data, a_opt, b_opt, c_opt) - ydata.mean())**2) /
np.sum((ydata - ydata.mean())**2)
print("r^2 = %10.6f" % R2)

popt2, pcov2 = curve_fit(func, x2data, ydata)
a2_opt, b2_opt, c2_opt = popt2

R2 = np.sum((func(x2data, a2_opt, b2_opt, c2_opt) - ydata.mean())**2) /
np.sum((ydata - ydata.mean())**2)
```

```

print("r^2 = %10.6f" % R2)
#####
fig, (ax0, ax1) = plt.subplots(nrows=1, ncols=2, sharex=True,
                               figsize=(16, 5))

ax0.tick_params(labelsize=16)
ax0.plot(x1data, ydata, 's', color='blue', label='data')
#ax0.errorbar(x1data, ydata, yerr=yerr)
#array([[14.69398827, 25.41747714, 11.75704008]])
ax0.plot(x1data2, func(x1data2, *popt1), 'b-')
ax0.set_ylim(0, 0.014)
ax0.set_xlabel("EUD_L (g) ", fontsize=16)
ax0.set_ylabel('Release rate $(\min^{-1})$', fontsize=16)
ax0.text(0.10, 0.0110, 'R$^{2}$=0.81', fontsize=16)
ax0.text(0.10, 0.010, 'P=0.0.001', fontsize=16)
plt.legend(fit)
#####
ax1.tick_params(labelsize=16)
ax1.plot(x2data, ydata, 'o', color='blue', label='data')
#array([[14.69398827, 25.41747714, 11.75704008]])
ax1.plot(x2data2, func(x2data2, *popt2), 'b-')
ax1.set_xlabel("EUD_S (g) ", fontsize=16)
ax1.set_ylim(0, 0.014)
ax1.set_ylabel('Release rate $(\min^{-1})$', fontsize=16)
ax1.text(1.5, 0.011, 'R$^{2}$=0.81', fontsize=16)
ax1.text(1.5, 0.010, 'P=0.0.001', fontsize=16)
plt.show()
fig.savefig('Fig_2.png', dpi=300)
r^2 = 0.821380
r^2 = 0.821380

```

In [82]:

```

result_1 = pd.read_excel("results_1.xlsx")
def func1(x, a, b, c):
    return a *x +b
xx = np.linspace(0, 100)
x=result_1['yf']
y=result_1['pred_y_baysien']
popt0, pcov0 = curve_fit(func1, x, y)
popt0
fig, (ax0) = plt.subplots(nrows=1, ncols=1, sharex=True,
                          figsize=(10, 6))

ax0.plot(x, y, 's', color='blue', label='data')
ax0.plot(xx, func1(xx, *popt0), 'b-', color='black')
ax0.tick_params(labelsize=16)
ax0.set_xlabel("Observed values (%) ", fontsize=16)
ax0.set_ylabel('Predicted values (%)', fontsize=16)
ax0.text(0.25, 90, 'R$^{2}$=0.99', fontsize=16)
ax0.text(0.25, 84, 'P=0.0001', fontsize=16)
plt.legend(fit)
plt.show()
fig.savefig('baa_Fig_11.png', dpi=300)
C:\ProgramData\Anaconda3\lib\site-packages\scipy\optimize\minpack.py:828:
OptimizeWarning: Covariance of the parameters could not be estimated
    warnings.warn('Covariance of the parameters could not be estimated',

```

In []:

```

result_1 = pd.read_excel("results_1.xlsx")
result_1.sort_values(['EUD_L', 'K1_HB'], inplace=True, ascending=True)
from scipy.optimize import curve_fit
def func(x, K1):

```

```

        return 100*(1-exp(k1*x))

xldata2 = np.linspace(0, 100, 50)
xldata=result_1['TIME']

se=result_1['se']
#xdata=result_1['EUD_L']
ydata=result_1['yf']
for run in runs:
    popt1, pcov1 = curve_fit(func, xldata, ydata)
    R2 = np.sum((func(xldata, a_opt, b_opt, c_opt) - ydata.mean())**2) /
np.sum((ydata - ydata.mean())**2)

popt1
#print("r^2 = %10.6f" % R2)
#####

```

In [84]:

```

BAA_curve_fig= pd.read_excel("BAA_curve_fig.xlsx")

fig, ax = plt.subplots(figsize=(16, 8))
ax.plot(BAA_curve_fig["x"], BAA_curve_fig["E1"], ',', color='blue')
ax.plot(BAA_curve_fig["px"], BAA_curve_fig["pE1"], 'b-', color='blue',
label='F1')

ax.plot(BAA_curve_fig["x"], BAA_curve_fig["E2"], ',', color='green')
ax.plot(BAA_curve_fig["px"], BAA_curve_fig["pE2"], 'b-', color='green',
label='F2')

ax.plot(BAA_curve_fig["x"], BAA_curve_fig["E3"], 's', color='aqua')
ax.plot(BAA_curve_fig["px"], BAA_curve_fig["pE3"], 'b-', color='aqua',
label='F3')
ax.plot(BAA_curve_fig["x"], BAA_curve_fig["E4"], 's', color='black')
ax.plot(BAA_curve_fig["px"], BAA_curve_fig["pE4"], 'b-', color='black',
label='F4')

ax.plot(BAA_curve_fig["x"], BAA_curve_fig["E5"], 's', color='brown')
ax.plot(BAA_curve_fig["px"], BAA_curve_fig["pE5"], 'b-', color='brown',
label='F5')

ax.plot(BAA_curve_fig["x"], BAA_curve_fig["E6"], 's', color='orange')
ax.plot(BAA_curve_fig["px"], BAA_curve_fig["pE6"], 'b-', color='orange',
label='F6')

ax.plot(BAA_curve_fig["x"], BAA_curve_fig["E7"], 's', color='pink')
ax.plot(BAA_curve_fig["px"], BAA_curve_fig["pE7"], 'b-', color='pink',
label='F7')

ax.plot(BAA_curve_fig["x"], BAA_curve_fig["E8"], 's', color='olive')
ax.plot(BAA_curve_fig["px"], BAA_curve_fig["pE8"], 'b-', color='olive',
label='F8')

ax.plot(BAA_curve_fig["x"], BAA_curve_fig["E9"], 's', color='red')
ax.plot(BAA_curve_fig["px"], BAA_curve_fig["pE9"], 'b-', color='red',
label='F9')

ax.plot(BAA_curve_fig["x"], BAA_curve_fig["E10"], 's', color='yellow')
ax.plot(BAA_curve_fig["px"], BAA_curve_fig["pE10"], 'b-', color='yellow',
label='F10')

ax.plot(BAA_curve_fig["x"], BAA_curve_fig["E10"], 's', color='yellow')

```



```

ax.plot(BAA_curve_fig["px"], BAA_curve_fig["pE10"], 'b-', color='yellow',
label='F10')

ax.plot(BAA_curve_fig["x"], BAA_curve_fig["E11"], 's', color='yellowgreen')
ax.plot(BAA_curve_fig["px"], BAA_curve_fig["pE11"], 'b-',
color='yellowgreen', label='F11')

ax.plot(BAA_curve_fig["x"], BAA_curve_fig["E12"], 's', color='silver')
ax.plot(BAA_curve_fig["px"], BAA_curve_fig["pE12"], 'b-', color='silver',
label='F12')

ax.plot(BAA_curve_fig["x"], BAA_curve_fig["E13"], ',', color='tan')
ax.plot(BAA_curve_fig["px"], BAA_curve_fig["pE13"], 'b-', color='tan',
label='F13')

#plt.plot(xx, func1(xx, *popt0), 'b-', color='black')
ax.tick_params(labelsize=16)
ax.set_xlabel("Time (min)", fontsize=16)
ax.set_ylabel('Cumulative release (%)', fontsize=16)
ax.legend(loc='lower right')
plt.text(0.25, 84, 'P=0.0001', fontsize=16)

plt.show()
fig.savefig('baa_curve.png', dpi=300)

```

## Résumé :

L'approche de modélisation bayésienne hiérarchique a été utilisée pour sélectionner le modèle cinétique empirique approprié pour la libération prolongée et pour optimiser la vitesse de dissolution in vitro du suppositoire de libération prolongée en contrôlant la composition d'Eudragit L-100 et d'Eudragit S-100 dans le mélange expérimental.

Treize formulations de suppositoires ont été préparées avec un mélange de 2g (10%) d'Eudragit R 100 et S100 selon un plan expérimental de mélange personnalisé. La libération cumulative de l'ingrédient actif a été mesurée à cinq reprises (20, 50, 80, 160 et 235 minutes).

Le meilleur modèle a été sélectionné en utilisant  $R^2$  (Ajustement) et AIC (Akaike Information Criterion) pour la méthode standard et en utilisant le poids de la validation croisée WAIC (critère d'information largement applicable) et LOO (Leave-one-out) pour l'approche bayésienne. L'approche fréquentiste a donné trois meilleurs modèles selon la formulation. Par rapport à cela, la méthode bayésienne a permis de définir un modèle unique, qui était le modèle du premier ordre. La probabilité relative de ce modèle était de 0,97, 0,99 sur la base de la WAIC et de la LOO respectivement.

La relation entre  $K_1$  (constante de taux de libération) et les quantités des deux Eudragits était quadratique, pour Eudragit\_L,  $Q_{\text{release}} (\%) = 0,0031X^2 - 0,0026 X + 0,0069$  et X est l'Eudragit L100 et  $K_1$  (taux de libération) = 0,41min<sup>-1</sup>.

La méthode bayésienne a permis de trouver le modèle le plus adéquat parmi plusieurs modèles qui peuvent être générés par l'approche fréquentiste standard.

**Abstract:**

The hierarchical Bayesian modeling approach was used to select the appropriate empirical kinetics model of sustained release and to optimize the in vitro dissolution rate of the sustained-release suppository by controlling the composition of Eudragit L-100 and Eudragit S-100 in the experimental mixture.

Thirteen formulations of suppositories were prepared with 2g (10%) mixture of Eudragit R 100 and S100 according to a personalized mixture experimental design. The cumulative release of active ingredient was measured at five time (20, 50, 80, 160, and 235min).

The best model was selected using Rsq (Adjust) and AIC (Akaike Information Criterion) for standard method and by using the weight of WAIC (widely applicable information criterion) and LOO (Leave-one-out) cross-validation for the Bayesian approach. Frequentist approach gave three best models depending on the formulation. Compared to this, the Bayesian method was able to define a single model, which was the first order model. The relative probability of this model was 0.97, 0.99 based on the WAIC, and LOO respectively.

The relationship between K1 (Release rate constant) and the quantities of the two Eudragits was quadratic, for Eudragit\_L,  $Q \text{ release (\%)} = 0.0031X^2 - 0.0026X + 0.0069$  and X is the Eudragit L100 and  $K1 \text{ (Rate release)} = 0.41\text{min}^{-1}$ . The Bayesian method allowed finding the most adequate model among several models that can be generated by the standard frequentist approach.

## الملخص:

الهرمي لاختيار نموذج الحركة التجريبية المناسبة للإطلاق المستمر وتحسين Bayesian تم استخدام نهج النمذجة Eudragit S-100 و Eudragit L-100 معدل الذوبان في المختبر لتحميل الإطلاق المستمر عن طريق التحكم في تكوين في الخليط التجريبي 100.

وفقاً للتصميم S100 و Eudragit R 100 من (10%) جم 2 تم تحضير ثلاثة عشر تركيبة من التحميل بمزيج (235، 160، 80، 50، 20) تم قياس الإصدار التراكمي للمكون النشط في خمس مرات. التجريبي لخليط شخصي (دقيقة).

للمطابقة AIC (Akaike Information Criterion) و Rsq (Adjust) تم اختيار أفضل نموذج باستخدام (ترك) LOO والتحقق من صحة (معيار المعلومات القابل للتطبيق على نطاق واسع) WAIC القياسية وباستخدام وزن بالمقارنة مع هذا ، كانت . أعطى النهج المتكرر ثلاثة أفضل النماذج اعتماداً على الصيغة. للنهج البايزي (مرة واحدة) كان الاحتمال النسبي لهذا النموذج . طريقة بايز قادرة على تحديد نموذج واحد ، والذي كان أول نموذج من الدرجة الأولى على التوالي LOO و WAIC بناءً على 0.99 و 0.97.

، Eudragit\_L من الدرجة الثانية ، بالنسبة لـ Eudragits وكميات (ثابت معدل الإصدار) K1 كانت العلاقة بين  $Q = 0.41$  (معدل الإصدار) K1 و Eudragit L100 هو  $X = 0.0069 + 0.0026X - 0.0031X^2$  (%) Q إصدار سمحت طريقة بايز بإيجاد أنسب نموذج من بين العديد من النماذج التي يمكن إنشاؤها بواسطة النهج التكراري .1-دقيقة القياسي.

# Graphical Abstract :

