

# Estimation de paramètres dans les modèles joints paramétriques en étendant le code du package saemix en R

Emmanuelle Comets<sup>1 2</sup> ([emmanuelle.comets@inserm.fr](mailto:emmanuelle.comets@inserm.fr))

Alexandra Lavalley-Morelle, France Mentré, Jimmy Mullaert<sup>1</sup>

<sup>1</sup>Université Paris Cité and Université Sorbonne Paris Nord, Inserm, IAME, F-75018 Paris, France

<sup>2</sup>Université de Rennes, Inserm, EHESP, Irset - UMRs\_1085, F-35000 Rennes, France

Journées R, Vannes

12-14 juin 2024



# Plan

---

## ① Introduction

## ② Le package saemix

Modèles et méthodes

Pharmacocinétique de la théophylline

Toenail infection

## ③ Extension aux modèles joints

## ④ Conclusion

# Overview

---

- ① Introduction
- ② Le package saemix
- ③ Extension aux modèles joints
- ④ Conclusion

# Modèles non-linéaires à effets mixtes pour les données longitudinales

- Données longitudinales recueillies dans de nombreuses études
  - études pharmacocinétiques et pharmacodynamiques dans le développement de médicaments
  - courbes de croissance, évolution des systèmes en agronomie
  - imagerie médicale, surveillance géographique,...
- Modélisation utile pour :
  - quantifier des phénomènes (évolution moyenne, variabilité)
  - faire de l'inférence (sélection de modèles, test d'hypothèses)
  - prédire (diagnostic, prise de décision)
- Utilisation de modèles à effets mixtes pour prendre en compte la nature hiérarchique des données
- Quelques uns des outils disponibles mettant en œuvre ces approches
  - logiciels commerciaux: NONMEM<sup>1</sup>, Monolix<sup>2</sup>
  - approches Bayésiennes: Stan<sup>3</sup>, appelé par rstan en R, BUGS<sup>4</sup>, JAGS, appelé par rjags en R<sup>5</sup>
  - packages R: nlme<sup>6</sup> dans R-base, saemix<sup>7 8</sup> étendu aux modèles pour données discrètes<sup>9</sup>

---

<sup>1</sup>Beal et al. (1989-2022)

<sup>2</sup>Lavielle (2014)

<sup>3</sup>Carpenter et al. (2017) *Journal of Statistical Software*

<sup>4</sup>Lunn et al. (2012)

<sup>5</sup>Plummer, Stukalov, and Denwood (2023)

<sup>6</sup>Pinheiro and Bates (2000)

<sup>7</sup>Comets, Lavenu, and Lavielle (2011) *20<sup>th</sup> meeting of the Population Approach Group in Europe, Athens, Greece*

<sup>8</sup>Comets, Lavenu, and Lavielle (2017) *Journal of Statistical Software*

<sup>9</sup>Karimi, Lavielle, and Moulines (2020) *Computational Statistics & Data Analysis*

# Overview

---

## ① Introduction

## ② Le package saemix

Modèles et méthodes

Pharmacocinétique de la théophylline

Toenail infection

## ③ Extension aux modèles joints

## ④ Conclusion

## Méthodes mises en œuvre dans saemix

$$\begin{aligned} y_{ij} &\sim \mathcal{D}(x_{ij}, \psi_i, \sigma) , \quad 1 \leq i \leq N , \quad 1 \leq j \leq n_i \\ \psi_i &= h^{-1}(\mu, \beta, c_i, \eta_i), \quad \eta_i \sim_{i.i.d.} \mathcal{N}(0, \Omega) \end{aligned} \tag{1}$$

⇒ Estimation par maximum de vraisemblance des paramètres de population :  $\theta = \{\mu, \beta, \Omega, \sigma\}$

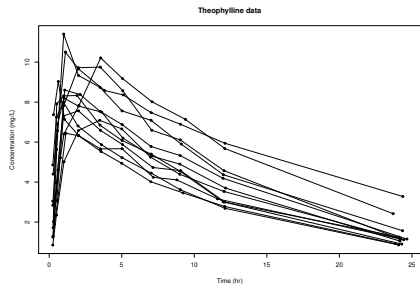
- Algorithme SAEM<sup>10</sup>
  - variante de l'algorithme EM (itératif)
  - étape E : simulation des paramètres individuels par un algorithme de Metropolis-Hastings et approximation stochastique de la vraisemblance conditionnelle
  - étape M : maximisation
- Estimation de l'incertitude sur les paramètres estimés
  - matrice d'information de Fisher (FIM), le plus usuellement calculée par linéarisation du modèle
  - autres approches : bootstrap, sampling-importance resampling (SIR)
- Estimation de la vraisemblance
  - différentes méthodes (linéarisation, importance sampling, quadrature)
- Estimation des paramètres individuels
  - moyenne (et déviation standard) des distributions conditionnelles
- Evaluation du modèle : graphes diagnostiques (package npde<sup>11</sup>)

<sup>10</sup>Kuhn and Lavielle (2005) *Computational Statistics and Data Analysis*

<sup>11</sup>Comets and Mentré (May 2021) *AAPS Journal*

## Exemple de données continues: la PK de la théophylline

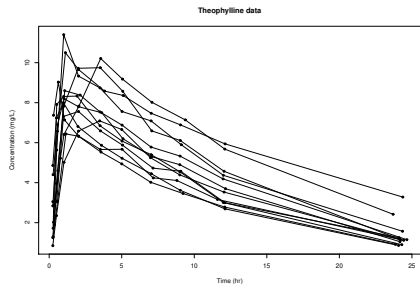
### Données: concentrations de théophylline



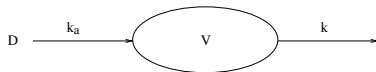
12 sujets ayant reçu 1 dose de théophylline  
10 mesures en 24 heures

## Exemple de données continues: la PK de la théophylline

### Données: concentrations de théophylline



### Modèle pharmacocinétique



$k_a$ : constante d'absorption;  $V$ : volume de distribution

$k$ : constante d'élimination

Equation d'un modèle à un compartiment avec absorption d'ordre 1

$$C(t, k_a, V, Cl) = \frac{D_i k_{ai}}{V_i(k_{ai} - k_{ei})} (e^{-k_{ai}t_{ij}} - e^{-k_{ei}t_{ij}})$$

où  $D_i$ : dose reçue par le sujet  $i$  et  $k_e = \frac{Cl}{V}$  où  $Cl$ : clairance de la molécule



## Définition du modèle en saemix

Données continues : décomposition en un modèle structurel  $f$  et un modèle d'erreur résiduelle  $g$

$$\begin{aligned} y_{ij} &= f(x_{ij}, \psi_i) + g(x_{ij}, \psi_i, \sigma) \epsilon_{ij} \quad , \quad 1 \leq i \leq N \quad , \quad 1 \leq j \leq n_i \\ \psi_i &= h^{-1}(\mu, \beta, c_i, \eta_i), \quad \eta_i \sim_{i.i.d.} \mathcal{N}(0, \Omega) \\ \epsilon_{ij} &\sim_{i.i.d.} \mathcal{N}(0, Id_{n_i}) \end{aligned} \quad (2)$$

### Définition du modèle en saemix

- modèle structurel défini préalablement par une fonction nommée
- modèle statistique défini via **saemixModel()**

# Définition du modèle en saemix

## Modèle structurel

- 3 arguments pré-définis
- renvoie : vecteur des prédictions du modèle  $f$  en chaque observation

```
model1cpt<-function(psi id xidep) {  
  dose<-xidep[,1]  
  tim<-xidep[,2]  
  ka<-psi[id,1]  
  V<-psi[id,2]  
  CL<-psi[id,3]  
  k<-CL/V  
  ypred<-dose*ka/(V*(ka-k))*(exp(-k*tim)-exp(-ka*tim))  
  return(ypred)  
}
```

Parameter table (dataframe  
 $N \times \text{par}$ )

Vector connecting the lines  
in xidep to the lines in psi

Predictors (dataframe  
 $\text{nobs} \times \text{npar}$ )

Created  
by  
saemix

Model equations

# Définition du modèle en saemix

## Modèle statistique

- paramètres du modèle et CI

```
saemix.model<-saemixModel(model=model1cpt,  
  description="One-compartment model with first-order absorption",  
  psi0=matrix(c(1.,20,0.5,0.1,0,-0.01),ncol=3,byrow=TRUE,  
    dimnames=list(NULL, c("ka","V","CL"))),  
  transform.par=c(1,1,1), fixed.estim=c(1,1,1),  
  covariate.model=matrix(c(0,0,1,0,0,0),ncol=3,byrow=TRUE),  
  covariance.model=matrix(c(1,0,0,0,1,1,0,1,1),ncol=3,byrow=TRUE),  
  omega.init=matrix(c(1,0,0,0,1,0,0,0,1),ncol=3,byrow=TRUE),  
  error.model="combined")
```

Nb of parameters: 3  
parameter names: ka V CL  
distribution:

Parameter	Distribution	Estimated
[1,] ka	log-normal	Estimated
[2,] V	log-normal	Estimated
[3,] CL	log-normal	Estimated

Variance-covariance matrix:

	ka	V	CL
ka	1	0	0
V	0	1	1
CL	0	1	1

Error model: combined , initial values: a.1=1 b.1=1

Covariate model:

	ka	V	CL
[1,]	0	0	1
[2,]	0	0	0

Initial values

	ka	V	CL
Pop.CondInit	1.0	20	0.50
Cov.CondInit	0.1	0	-0.01

# Définition du modèle en saemix

## Modèle statistique

- modèle de variabilité inter-individuelle (IIV) : distribution

```
saemix.model<-saemixModel(model=model1cpt,  
  description="One-compartment model with first-order absorption",  
  psi0=matrix(c(1,.20,0.5,0.1,0,-0.01),ncol=3,byrow=TRUE,  
    dimnames=list(NULL, c("ka","V","CL"))),  
  transform.par=c(1,1,1), fixed.estim=c(1,1,1),  
  covariate.model=matrix(c(0,0,1,0,0,0),ncol=3,byrow=TRUE),  
  covariance.model=matrix(c(1,0,0,0,1,1,0,1,1),ncol=3,byrow=TRUE),  
  omega.init=matrix(c(1,0,0,0,1,0,0,0,1),ncol=3,byrow=TRUE),  
  error.model="combined")
```

```
Nb of parameters: 3  
parameter names: ka V CL  
distribution:  
Parameter Distribution Estimated  
[1, ka log-normal Estimated  
[2, V log-normal Estimated  
[3, CL log-normal Estimated  
Variance-covariance matrix:  
ka V CL  
ka 1 0 0  
V 0 1 1  
CL 0 1 1
```

Options in saemix: 0=normal,  
1=log-normal, 2=probit,  
3=logit

# Définition du modèle en saemix

## Modèle statistique

- modèle d'IIV : structure de covariance

```
saemix.model<-saemixModel(model=model1cpt,  
  description="One-compartment model with first-order absorption",  
  psi0=matrix(c(1.,20,0.5,0.1,0,-0.01),ncol=3,byrow=TRUE,  
    dimnames=list(NULL, c("ka","V","CL"))),  
  transform.par=c(1,1,1), fixed.estim=c(1,1,1),  
  covariate.model=matrix(c(0,0,1,0,0,0),ncol=3,byrow=TRUE),  
  covariance.model=matrix(c(1,0,0,0,1,1,0,1,1),ncol=3,byrow=TRUE),  
  omega.init=matrix(c(1,0,0,0,1,0,0,0,1),ncol=3,byrow=TRUE),  
  error.model="combined")
```

```
Nb of parameters: 3  
parameter names: ka V CL  
distribution:  
Parameter Distribution Estimated  
[1,] ka      log-normal Estimated  
[2,] V      log-normal Estimated  
[3,] CL      log-normal Estimated
```

```
Variance-covariance matrix:  
ka V CL  
ka 1 0 0  
V 0 1 1  
CL 0 1 1
```

# Définition du modèle en saemix

## Modèle statistique

- modèle de covariables

```
saemix.model<-saemixModel(model=model1cpt,  
  description="One-compartment model with first-order absorption",  
  psi0=matrix(c(1.,20,0.5,0.1,0,-0.01),ncol=3,byrow=TRUE,  
    dimnames=list(NULL, c("ka","V","CL"))),  
  transform.par=c(1,1,1), fixed.estim=c(1,1,1),  
  covariate.model=matrix(c(0,0,1,0,0,0),ncol=3,byrow=TRUE),  
  covariance.model=matrix(c(1,0,0,0,1,0,1,0,1,1),ncol=3,byrow=TRUE),  
  omega.init=matrix(c(1,0,0,0,1,0,0,0,1),ncol=3,byrow=TRUE),  
  error.model="combined")
```

```
Nb of parameters: 3  
parameter names: ka V CL  
distribution:  
Parameter Distribution Estimated  
[1,] ka      log-normal Estimated  
[2,] V      log-normal Estimated  
[3,] CL      log-normal Estimated  
Variance-covariance matrix:  
ka V CL  
ka 1 0 0  
V 0 1 1  
CL 0 1 1  
Error model: combined , initial values: a.1=1 b.1=1  
Covariate model:  
ka V CL  
[1,] 0 0 1  
[2,] 0 0 0  
Initial values  
ka V CL  
Pop.CondInit 1.0 20 0.50  
Cov.CondInit 0.1 0 -0.01
```

# Définition du modèle en saemix

## Modèle statistique

- modèle d'erreur résiduelle

```
saemix.model<-saemixModel(model=model1cpt,  
  description="One-compartment model with first-order absorption",  
  psi0=matrix(c(1.,20,0.5,0.1,0,-0.01),ncol=3,byrow=TRUE,  
    dimnames=list(NULL, c("ka","V","CL"))),  
  transform.par=c(1,1,1), fixed.estim=c(1,1,1),  
  covariate.model=matrix(c(0,0,1,0,0,0),ncol=3,byrow=TRUE),  
  covariance.model=matrix(c(1,0,0,0,1,1,0,1,1),ncol=3,byrow=TRUE),  
  omega.init=matrix(c(1.0,0.0,1,0,0,0,1),ncol=3,byrow=TRUE),  
  error.model="combined")
```

```
Nb of parameters: 3  
parameter names: ka V CL  
distribution:  
Parameter Distribution Estimated  
[1,] ka      log-normal Estimated  
[2,] V      log-normal Estimated  
[3,] CL      log-normal Estimated  
Variance-covariance matrix:  
ka V CL  
ka 1 0 0  
V 0 1 1  
CL 0 1 1  
Error model: combined , initial values: a.1=1 b.1=1  
Covariate model:  
ka V CL  
[1,] 0 0 1  
[2,] 0 0 0  
Initial values  
ka V CL  
Pop.CondInit 1.0 20 0.50  
Cov.CondInit 0.1 0 -0.01
```

# Définition de l'objet données

## Données structurées

- défini via **saemixData()**
- prédicteurs dans le même ordre que la fonction modèle

```
saemix.data<-saemixData(name.data=theo.saemix.header=TRUE,sep=" ",na=NA,  
  name_group=c("Id"),name.predictors=c("Dose","Time"),  
  name.response=c("Concentration"),name.covariates=c("Weight","Sex"),  
  units=list(x="hr",y="mg/L",covariates=c("kg","-")), name.X="Time")
```

Grouping variable (Subject)

Outcome (Y variable)

Predictors (X variable)

The following SaemixData object was successfully created:

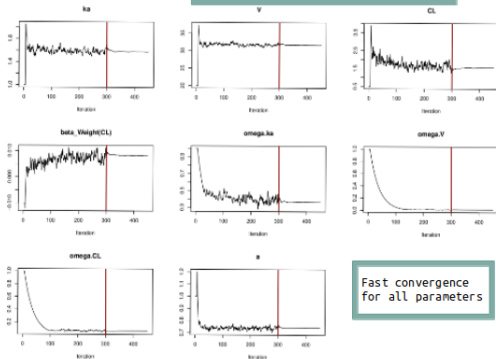
```
Object of class SaemixData  
longitudinal data for use with the SAEM algorithm  
Dataset: theo.saemix  
Structured data: Concentration ~ Dose + Time | Id  
X variable for graphs: Time (hr)  
covariates: Weight (kg), Sex (-)  
reference class for covariate Sex : 0
```



# Ajustement

```
saemix.fit<-saemix(saemix.model,saemix.data,saemix.options)
```

List of settings (number of chains, of iterations) and options (save graphs, algorithms)



Fast convergence  
for all parameters

## Results

### Fixed effects

Parameter	Estimate	SE	CV(%)	p-value
[1,] ka	1.567	0.2998	19.1	-
[2,] V	31.475	1.3838	4.4	-
[3,] CL	1.581	1.0155	64.2	-
[4,] beta_Weight(CL)	0.008	0.0092	113.5	0.19
[5,] a	0.743	0.0569	7.7	-

### Variance of random effects

Parameter	Estimate	SE	CV(%)
ka omega.ka	0.388	0.175	45
V omega.V	0.015	0.009	59
CL omega.CL	0.070	0.034	49

### Correlation matrix of random effects

	omega.ka	omega.V	omega.CL
omega.ka	1	0	0
omega.V	0	1	0
omega.CL	0	0	1

### Statistical criteria

Likelihood computed by linearisation

-2LL= 343.4919

AIC = 359.4919

BIC = 363.3712

Likelihood computed by importance sampling

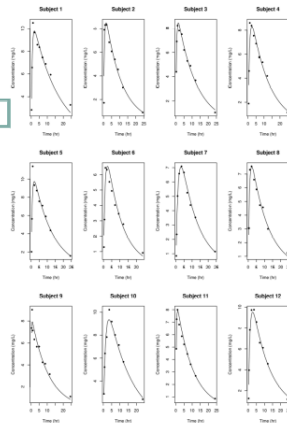
# Quelques graphes diagnostiques <sup>12</sup>

```
plot(saemix.fit)
```

```
ynpde<-npdeSaemix(saemix.fit)
```

Different plot types available  
through the plot.type argument

Individual plots



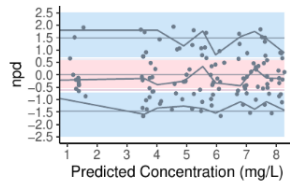
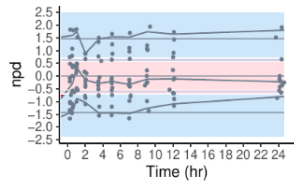
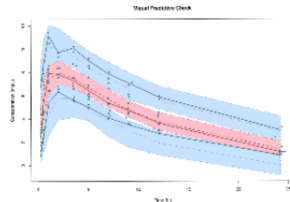
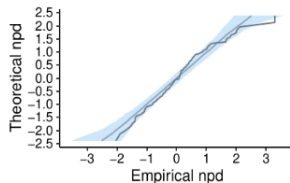
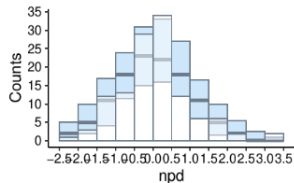
# Quelques graphes diagnostiques <sup>12</sup>

```
plot(saemix.fit)  
ynpde<-npdeSaemix(saemix.fit)
```

Diagnostic plots from the npde package

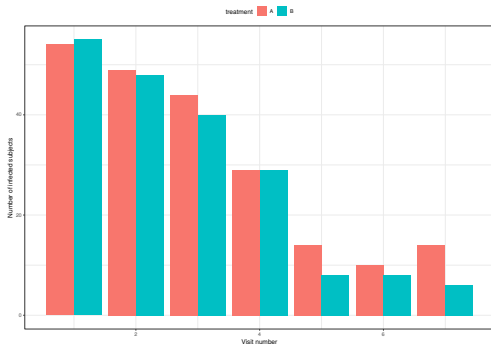
Normalised prediction discrepancies

VPC



## Exemple de données discrètes

Dataset Toenail (package prLogistic de R)

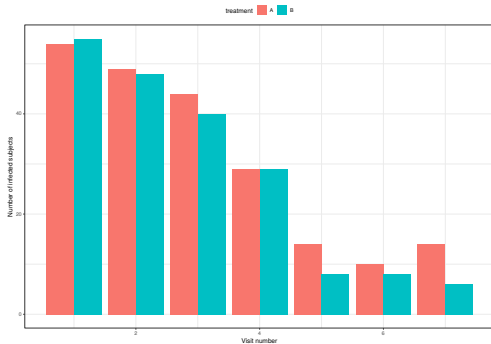


Essai clinique randomisé comparant 2 traitements pour une infection fongique de l'ongle<sup>13</sup>  
294 patients avec des mesures à 7 visites (n=1908)  
Diminution de la proportion de patients infectés au cours du temps

<sup>13</sup>De Backer et al. (1998) *Journal of the American Academy of Dermatology*

## Exemple de données discrètes

Dataset Toenail (package prLogistic de R)



Régression logistique pour  $P(Y = 1)$  ( $Y=1$  infection sévère ou modérée, 0 pour légère ou nulle), avec effet traitement sur la pente

$$\begin{aligned} \text{logit}(P(y_{ij} = 1)) &= \theta_{1,i} + (\theta_2 + \beta \mathbb{1}_{trt=B}) t_{ij} \\ \theta_1 &\sim N(\mu_1, \omega_1^2) \end{aligned} \quad (2)$$

# Définition du modèle en saemix

## Définition du modèle pour données discrètes

- même structure pour la fonction modèle
- renvoie : log-vraisemblance en chaque observation

## Fonction supplémentaire pour simuler sous le modèle

- non requise pour l'estimation
- utilisée pour les diagnostics
- renvoie : simulations sous le modèle

```
binary.model<-function(psi,id,xidep) {  
  tim<-xidep[,1]  
  y<-xidep[,2]  
  inter<-psi[id,1]  
  slope<-psi[id,2]  
  logit<-inter+slope*tim  
  pevent<-exp(logit)/(1+exp(logit))  
  logpdf<-rep(0,length(tim))  
  P.obs = (y==0)*(1-pevent)+(y==1)*pevent  
  logpdf <- log(P.obs)  
  return(logpdf)  
}
```

Need to pass observed outcome as a predictor for this model to compute the corresponding probability

Equation for the logit

Simulation with a binomial distribution

```
simulBinary<-function(psi,id,xidep) {  
  tim<-xidep[,1]  
  y<-xidep[,2]  
  inter<-psi[id,1]  
  slope<-psi[id,2]  
  logit<-inter+slope*tim  
  pevent<-1/(1+exp(-logit))  
  ysim<-rbinom(length(tim),size=1, prob=pevent)  
  return(ysim)  
}
```

# Définition du modèle en saemix

## Définition du modèle statistique

- changement du type de modèle

```
saemix.model<-saemixModel(model=binary.model,description="Binary model",  
  simulate.function=simulBinary, modeltype="likelihood",  
  psi0=matrix(c(-0.5,-.15,0,0),ncol=2,byrow=TRUE,  
    dimnames=list(NULL,c("theta1","theta2"))),  
  transform.par=c(0,0), covariate.model=c(0,1),  
  covariance.model=matrix(c(1,0,0,0),ncol=2),  
  omega.init=diag(c(0.5,0.3)), verbose=FALSE)
```

Model defined through the likelihood of the observations

```
saemix.options<-list(seed=1234567,save=FALSE,save.graphs=FALSE,  
  displayProgress=FALSE, nb.chains=10, fim=FALSE, print=FALSE)
```

FIM not computed for appropriate for discrete models  
Increase number of chains for discrete data

# Résultats

## Calcul des SE en modèle discret

- approches bootstrap<sup>13</sup>
- ici deux approches utilisées<sup>14</sup>

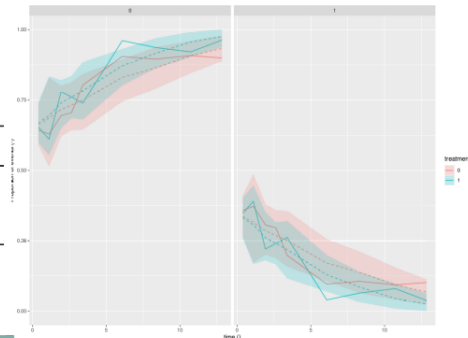
```
binary.fit <- saemix(saemix.model, saemix.data, saemix.options)
summary(binary.fit)
```

No difference between treatments

Parameter	saemix	Case bootstrap		Conditional bootstrap	
	Estimate	Mean (SD)	CI	Mean (SD)	CI
$\theta_1$	-1.71	-1.74 (0.360)	[-2.44, -1.08]	-1.53 (0.31)	[-2.20, -0.98]
$\theta_2$	-0.39	-0.40 (0.073)	[-0.56, -0.28]	-0.40 (0.044)	[-0.49, -0.32]
$\beta_{\text{treatment}, \theta_2}$	-0.15	-0.16 (0.13)	[-0.44, 0.066]	-0.15 (0.07)	[-0.29, -0.02]
$\omega_{\theta_1}^2$	16.11	17.06 (4.50)	[10.36, 28.23]	14.83 (2.82)	[10.01, 21.50]

```
binary.fit <- simulateDiscreteSaemix(binary.fit, nsim=nsim)
discreteVPC(binary.fit, outcome="binary", which.cov="treatment")
```

VPC stratified by treatment





# Overview

---

- ① Introduction
- ② Le package saemix
- ③ Extension aux modèles joints
- ④ Conclusion

## Extension du code dans le package saemix

- Package actuel
  - modèles pour différents types de réponses (continues, discrètes, survie)
  - une seule réponse prise en charge
  - calcul de la FIM linéarisée inadaptée aux modèles non gaussiens
- Cas d'étude<sup>15</sup>
  - modèle joint multivarié développé pour relier l'évolution de biomarqueurs à la mortalité chez des patients hospitalisés pour la Covid-19
  - sélection des biomarqueurs dans Monolix<sup>16</sup>
  - évènements décès et sortie d'hôpital modélisés comme des risques compétitifs

### Objectifs<sup>17</sup>

- étendre le code du package aux modèles multi-réponses
- mettre en œuvre un calcul de la FIM par une approximation stochastique<sup>18</sup>
- évaluer les performances du code modifié

<sup>15</sup> Lavalley-Morelle et al. (2023) *Biom J*

<sup>16</sup> Lavielle (2014)

<sup>17</sup> Lavalley-Morelle et al. (2024) *Comput Progs Meth Biomed*

<sup>18</sup> Delattre and Kuhn (2023) *HAL sciences*

## Changements pour l'utilisateur

### Extension de l'objet saemixData

Colonne spécifiant à quelle réponse correspond une observation

```
saemix.data<-saemixData(name.data=data_joint, name.group=c("id"),  
                        name.predictors=c("time","obs"),  
                        name.response="obs", name.ytype = "ytype")
```

Argument name.ytype pour spécifier la colonne attribuant les types de réponses



# Changements pour l'utilisateur

## Définition de la fonction modèle

Renvoi des prédictions pour les données continues et de  $\ell$  pour les données non gaussiennes

```
JMmodel<-function(psi,id,xidep) {
  ytype<-xidep$ytype # type of response (1: continuous, 2: event)
  b0 <- psi[id,1]
  b1 <- psi[id,2]
  h0 <- psi[id,3]
  alpha <- psi[id,4]

  ypred <- b0+b1*xidep$time # predictions for the longitudinal part

  T<-xidep$time[ytype==2] # vector of times (survival response)
  Nj <- length(T)
  ev = xidep$obs[ytype==2] # vector of observations (survival response)
  cens<-which(ev==0) # with censored ones
  ind <- which(ev==1) # and event ones

  # Creating vectors of the same length of T to compute likelihood of the survival part
  #(so removing duplicates)
  b0b = b0[ytype==2] # to have vectors of the same length as T
  b1b = b1[ytype==2]
  h0b = h0[ytype==2]
  alphab = alpha[ytype==2]

  haz <- h0b*exp(alphab*(b0b+b1b*T)) # instantaneous hazard
  # cumulative hazard (explicit expression in that case)
  H <- (h0b/(alphab*b1b))*exp((b0b+b1b*T)*alphab)-(h0b/(alphab*b1b))*exp(alphab*b0b)

  logpdf <- rep(0,Nj)
  # likelihood contributions for censored observations
```

ytype utilisé pour repérer les types de réponse  
(passé automatiquement dans xidep)

sortie du modèle pour les 2 (ici) types de  
réponse

Ici même temps de censure pour tous les sujets

Calcul des probabilités selon que l'évènement  
est observé ou non

# Changements pour l'utilisateur

## Définition du modèle statistique

- spécification du type de réponse
- un seul modèle d'erreur ici (code ad hoc)

dans l'extension, vecteur listant les types de réponses

```
saemix.model<-saemixModel(model=JMmodel,description="JM LMEM-TTE constant baseline hazard",
  modeltype=c("structural","likelihood"),
  psi0=matrix(param,ncol=4,byrow=TRUE,
    dimnames=list(NULL, c("b0","b1","h0","alpha"))),
  transform.par=c(0,0,1,0), covariance.model=diag(c(1,1,0,0)),
  fixed.estim = c(1,1,1,1),error.model = "constant",
```

# Application au cas d'étude

- Association avec le risque de décès chez les patients hospitalisés : taux de neutrophiles augmenté, pH artériel bas et CRP élevée, ainsi qu'un score 4C élevé à l'admission
- Résultats similaires à ceux trouvés dans l'étude initiale<sup>19</sup>

Parameter (unit)	Value	SE	RSE (%)
<b>Longitudinal submodel</b>			
<b>Blood neutrophil counts</b>			
$\mu_{0n}$ ( $10^9 \cdot L^{-1}$ )	4.58	0.18	3.9
$\mu_{1n}$ ( $10^9 \cdot L^{-1} \cdot d^{-1}$ )	-0.15	0.0019	1.3
$\mu_{2n}$ ( $10^9 \cdot L^{-1} \cdot d^{-1}$ )	-0.17	0.012	6.9
$\mu_{3n}$ ( $10^9 \cdot L^{-1}$ )	7.1	0.97	13.7
$\omega_{0n}$ ( $10^9 \cdot L^{-1}$ )	4.3	0.41	9.4
$\omega_{1n}$ ( $10^9 \cdot L^{-1} \cdot d^{-1}$ )	0.011	0.0027	24.3
$\omega_{2n}$ ( $10^9 \cdot L^{-1} \cdot d^{-1}$ )	0.003	0.0011	36.4
$\omega_{3n}$ ( $10^9 \cdot L^{-1}$ )	0.55	0.19	35.7
$\sigma_{1n}$	0.32	0.003	0.9
<b>arterial pH</b>			
$\mu_{0p}$	7.44	0.0042	0.06
$\mu_{1p}$ ( $d^{-1}$ )	0.001	0.00078	0.08
$\omega_{0p}$	0.0015	0.00016	10.9
$\omega_{1p}$ ( $d^{-1}$ )	$1.5 \times 10^{-5}$	$4.5 \times 10^{-6}$	29.3
$\sigma_{ap}$	0.055	0.00031	0.6
<b>C-reactive protein</b>			
$\mu_{0c}$ ( $\log(mg \cdot L^{-1})$ )	4.17	0.082	2.0
$\mu_{1c}$ ( $\log(mg \cdot L^{-1}) \cdot d^{-1}$ )	-0.14	0.011	7.4
$\omega_{0c}$ ( $\log(mg \cdot L^{-1})$ )	0.85	0.10	12.1
$\omega_{1c}$ ( $\log(mg \cdot L^{-1}) \cdot d^{-1}$ )	0.014	0.0025	18.1
$\sigma_{wc}$ ( $\log(mg \cdot L^{-1})$ )	0.72	0.0075	1.0
<b>Survival submodel</b>			
<b>Death</b>			
$h_0$	0.0003	0.79	$2.7 \times 10^5$
$\alpha_{1n}$ ( $L \cdot 10^{-9}$ )	0.14	0.033	23.7
$\alpha_{1p}$	-6.48	2.78	42.9
$\alpha_{1c}$ ( $-log(mg \cdot L^{-1})$ )	0.55	0.12	21.4
$\beta_1$	0.31	0.070	22.5
<b>Discharge</b>			
$b$	12.1	1.08	8.9

<sup>19</sup> Lavalley-Morelle et al. (2023) *Biom J*

# Etude de simulation

- Scénarios

- 1 biomarqueur modélisé par un modèle linéaire (LMEM) ou non-linéaire (NLMEM, somme de 2 exponentielles)
- évènement unique (TTE) ou risques compétitifs (CR)

	Single event	2 competing risks
Linear mixed-effects model	LMEM – TTE	LMEM – CR
Nonlinear mixed-effects model	NLMEM – TTE	NLMEM – CR

- Protocole expérimental

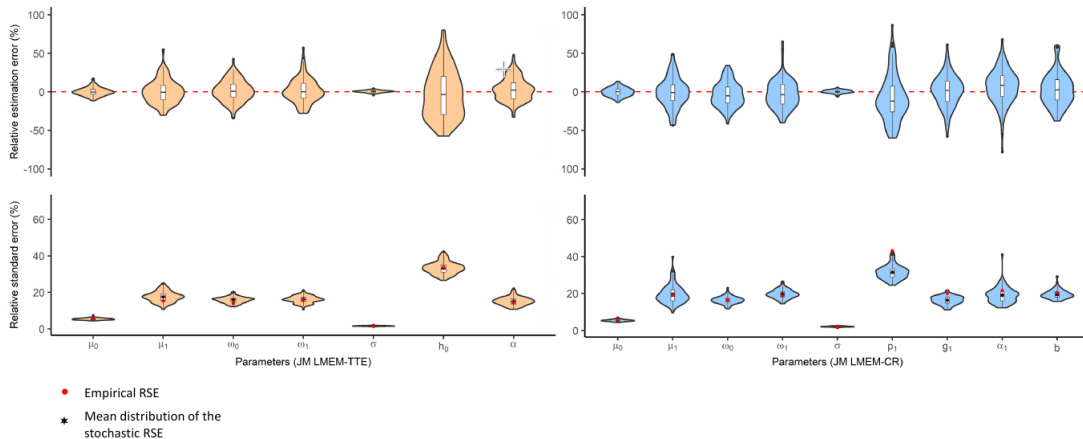
- 200 jeux de données de 100 patients
- biomarqueurs mesurés tous les jours jusqu'à l'évènement (censure à J30)

- Métriques

- erreurs d'estimation sur les paramètres dans tous les scénarios
- erreurs standards comparées à l'erreur empirique

# Résultats

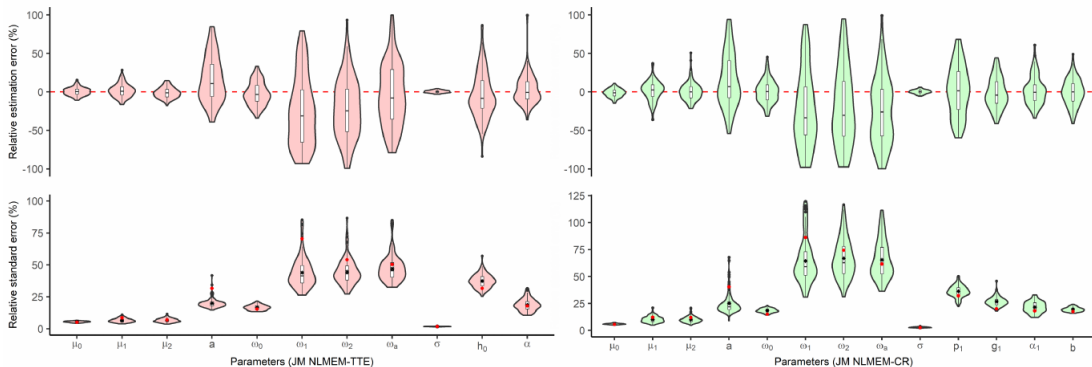
## Distribution of REE (top) and stochastic RSE versus empirical RSE (bottom) - JM LMEM-TTE and JM LMEM-CR





# Résultats

Distribution of REE (top) and stochastic RSE versus empirical RSE (bottom) - **JM NLMEM-TTE** and **JM NLMEM-CR**



- Empirical RSE
- ★ Mean distribution of the stochastic RSE

# Overview

---

- ① Introduction
- ② Le package saemix
- ③ Extension aux modèles joints
- ④ Conclusion

# Conclusion

- Package saemix
  - outil efficace, simple d'utilisation et ouvert pour ajuster des NLMEM dans un contexte fréquentiste
  - diagnostics graphiques pour évaluer la convergence et l'adéquation du modèle
  - aide en ligne ("?) et guide d'utilisateur détaillé (prière de RTM) à télécharger sur le github
- Extension de saemix aux modèles joints
  - bonnes propriétés pour l'estimation des paramètres et le calcul des erreurs standards (dans un cas riche)
  - possibilité pour les utilisateurs de définir directement la vraisemblance du modèle
  - code ad hoc disponible sur le git pour utilisateurs avertis (et joueurs)
  - prochaines étapes
    - intégrer le code pour réponses multiples dans le package
    - développer des outils diagnostiques spécifiques
    - optimiser l'intégration numérique lorsque la vraisemblance complète n'a pas de forme explicite (temps de calculs longs)
- Perspectives
  - beaucoup de développements encore à intégrer (LOQ, réponses multiples, IOV, HMM, ODE,...)
  - améliorer la déclaration des réponses et la définition des modèles en utilisant un langage possédant une 'grammaire des modèles mixtes'<sup>20</sup>

saemix est un package collaboratif

Join the fun on github: <https://github.com/saemixdevelopment/saemixextension/tree/master> !

<sup>20</sup>Swat et al. (2015) *Clinical Pharmacology and Therapeutics: Pharmacometrics & Systems Pharmacology*

# Plus de détails pour les curieux (aka un peu d'auto-promotion)



## *Journal of Statistical Software*

August 2017, Volume 80, Issue 3.

doi: 10.18637/jss.v080.i03

+

### Parameter Estimation in Nonlinear Mixed Effect Models Using saemix, an R Implementation of the SAEM Algorithm

Emmanuelle Comets  
INSERM CIC 1414  
U. Rennes-I  
INSERM, IAME, UMR 1137  
U. Paris Diderot

Audrey Lavenu  
U. Rennes-I  
INSERM CIC 1414

Marc Lavielle  
INRIA Saclay, Popix  
U. Paris Sud



## Journal de la Société Française de Statistique

Vol. 151 No. 1 (2010)

### Model evaluation in nonlinear mixed effect models, with applications to pharmacokinetics

**Titre:** Evaluation des modèles non-linéaires à effets mixtes, avec une application en pharmacocinétique

Emmanuelle Comets <sup>1</sup>, Karl Brendel <sup>2</sup> and France Mentré <sup>1</sup>

Computer Methods and Programs in Biomedicine 247 (2024) 108095



Contents lists available at ScienceDirect

### Computer Methods and Programs in Biomedicine

journal homepage: [www.elsevier.com/locate/cmpb](http://www.elsevier.com/locate/cmpb)



Extending the code in the open-source *saemix* package to fit joint models of longitudinal and time-to-event data

Alexandra Lavalley-Morelle <sup>a</sup>, France Mentré <sup>a,b</sup>, Emmanuelle Comets <sup>a,c,e,l</sup>, Jimmy Mullaert <sup>a,b,d,l</sup>



# Notations

## Modèle hiérarchique

$$\begin{aligned} y_{ij} &\sim \mathcal{D}(x_{ij}, \psi_i, \sigma) , \quad 1 \leq i \leq N , \quad 1 \leq j \leq n_i \\ \psi_i &= h^{-1}(\mu, \beta, c_i, \eta_i), \quad \eta_i \sim_{i.i.d.} \mathcal{N}(0, \Omega) \end{aligned} \tag{3}$$

Paramètres de population :  $\theta = \{\mu, \beta, \Omega, \sigma\}$

Dans le cadre de ce topo, paramètres estimés par maximisation de la vraisemblance (MLE), avec pour principaux algorithmes :

- algorithmes de minimisation (gradient, Hessien)
- variantes d'algorithmes EM

Log-vraisemblance (LL) obtenue en intégrant sur la distribution des paramètres individuels (inconnus)

$$LL(\theta|y) = \sum_{i=1}^N \log(p(y_i|\theta)) = \sum_{i=1}^N \log \left( \int p(y_i|\theta, \psi_i) p(\psi_i|\theta) d\psi_i \right) \tag{4}$$

# L'algorithme SAEM<sup>21 22</sup>

Initialisation :  $\theta = \theta_0$ .

## A l'itération $k$

- étape S (simulation) : tirer  $\psi^{(k)}$  dans la distribution conditionnelle  $p(\cdot|y; \theta_k)$ 
  - utilisation d'un algorithme de Metropolis-Hastings
- approximation stochastique de  $Q_k(\theta) = \mathbb{E} \left( \log p(y, \psi; \theta) | y, \theta_{k-1} \right)$  avec une suite de pas décroissants  $\gamma_k (> 0$  tels que  $\gamma_1 = 1$ )

$$Q_k(\theta) = Q_{k-1}(\theta) + \gamma_k (\log p(y; \psi^{(k)}; \theta_k) - Q_{k-1}(\theta))$$

- Étape M : mettre à jour l'estimation de  $\theta$

$$\theta_k = \text{Arg max}_{\theta} Q_k(\theta)$$

<sup>21</sup>Delyon, Lavielle, and Moulines (1999) *Annals of Statistics*

<sup>22</sup>Kuhn and Lavielle (2005) *Computational Statistics and Data Analysis*

# Calcul de la vraisemblance

---

- Calcul effectué après la fin de l'algorithme
- Différentes méthodes mises en œuvre dans saemix
  - linéarisation du modèle à l'ordre 1 (forme explicite pour LL)
  - quadrature gaussienne
  - échantillonnage préférentiel (importance sampling)
- Log-vraisemblance utilisée pour l'inférence
  - comparaison de modèles emboîtés par le test du rapport de vraisemblance (LRT, Likelihood Ratio Test)
  - comparaison de modèles selon des critères d'information comme l'AIC, le BIC et le BICc<sup>23</sup>

---

<sup>23</sup>Delattre, Lavielle, and Poursat (2014) *Electronic Journal of Statistics*



## Estimation de l'incertitude sur les paramètres de population

- Estimateur du maximum de vraisemblance asymptotiquement distribué selon une loi normale
  - borne inférieure de la variance d'estimation : inverse de la matrice d'information de Fisher (FIM)
  - ⇒ façon la plus usuelle de calculer l'incertitude
- Calcul de la FIM par linéarisation à l'ordre 1 du modèle en négligeant les corrélations d'estimation entre effets fixes et aléatoires
  - approximation acceptable dans le cas de modèles à données continues
  - méthode par défaut dans saemix
- Calcul sans linéarisation (non mises en œuvre dans saemix)
  - méthode de Louis par simulation MC <sup>24</sup>
  - calcul exact avec des intégrations numériques ou stochastiques <sup>25</sup>
- Approches bootstrap aussi mises en œuvre dans le package <sup>26</sup>
- Sampling Importance Resampling (SIR) <sup>27</sup> aussi disponible dans la version github (en développement)

### Spoiler alert

Voir la prochaine présentation pour plus de détails et des méthodes alternatives de calcul de l'incertitude

<sup>24</sup>Louis (1982) *Journal of the Royal Statistical Society: Series B (Methodological)*

<sup>25</sup>Riviere, Ueckert, and Mentré (May 2016) *Biostatistics*

Ueckert and Mentré (2017) *Computational Statistics & Data Analysis*

<sup>26</sup>Efron (1979) *Annals of Statistics*

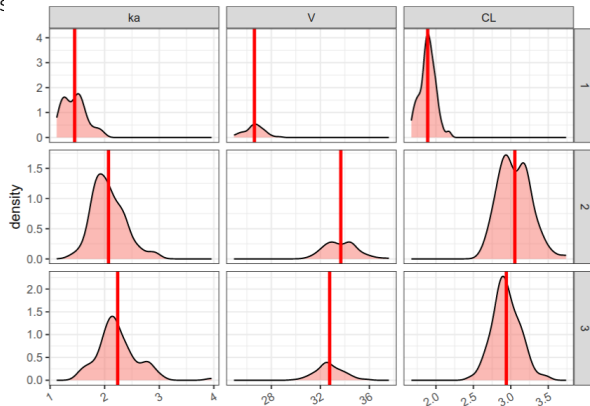
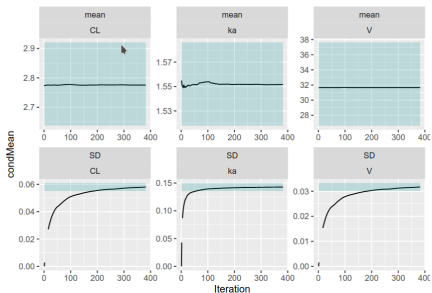
Thai et al. (2014) *Journal of Pharmacokinetics and Pharmacodynamics*

Comets et al. (2021) *Pharmaceutical Research*

<sup>27</sup>Dosne et al. (2017) *Journal of Pharmacokinetics and Pharmacodynamics*

## Estimation des paramètres individuels

- Paramètres individuels estimés comme le MAP ou la moyenne des distributions conditionnelles:
- Utilisation des paramètres individuels
  - graphes diagnostics
  - prédictions individuelles (choix de doses,...)
  - simulations
- Calcul des distributions conditionnelles
  - échantillonnage par MH
  - estimation de l'incertitude par la déviation standard

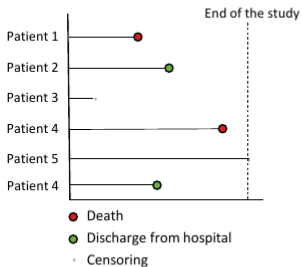


Conditional distributions for  $ka$ ,  $V$  and  $CL$  for the first three subjects in the theophylline example (100 samples)

# Modèle joint

Predict the death of hospitalized patients for severe infectious diseases

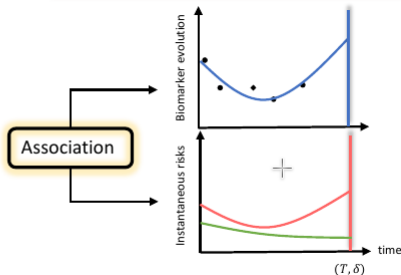
Hospitals equipped with laboratory information systems that routinely gather results of biological analyses



Competing risks



Consecutive biological observations can be used in a joint model to provide individual dynamic predictions<sup>1</sup> of patient prognosis



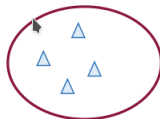
# Modèle pour les risques compétitifs

Risque instantané de décéder à l'hôpital :

$$h_1(t) = \lim_{\Delta t \rightarrow 0} \frac{P(t < T < t + \Delta t, \delta = 1 \vee E)}{\Delta t}$$

1

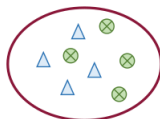
Approche cause-spécifique<sup>6</sup>



$$E = T > t$$

2

Approche sous-distribution<sup>7</sup>



$$E = \{T > t\} \cup \{T \leq t \cap \delta = 2\}$$



Patient toujours hospitalisés au temps  $t$  (= n'ayant pas fait d'évènement avant  $t$ )



Patient sortis d'hospitalisation au temps  $t$  (= ayant fait l'évènement 2 avant  $t$ )

# Scénarios de simulation

## LMEM – TTE

$$y_{ij} = m_l(\psi_i, t_{ij}) + \sigma \epsilon_{ij}$$

$$h_i(t, \psi_i; \theta) = h_0 \times \exp(\alpha \times m_l(\psi_i, t))$$

## LMEM-CR

$$y_{ij} = m_l(\psi_i, t_{ij}) + \sigma \epsilon_{ij}$$

$$h_{i1}(t, \psi_i; \theta) = \frac{p_1 g_1 \exp(-g_1 \times t)}{1 - p_1 (1 - \exp(-g_1 \times t))} \exp(\alpha_1 \times m_l(\psi_i, t))$$

$$h_{i2}(t, \psi_i; \theta) = \frac{1}{b} \times \frac{(1 - F_1(\infty)) \exp(-t/b)}{1 - (1 - F_1(\infty))(1 - \exp(-t/b))}$$

## NLMEM-TTE

$$y_{ij} = m_{nl}(\psi_i, t_{ij}) + \sigma \epsilon_{ij}$$

$$h_i(t, \psi_i; \theta) = h_0 \times \exp(\alpha \times m_{nl}(\psi_i, t))$$

## NLMEM-CR

$$y_{ij} = m_{nl}(\psi_i, t_{ij}) + \sigma \epsilon_{ij}$$

$$h_{i1}(t, \psi_i; \theta) = \frac{p_1 g_1 \exp(-g_1 \times t)}{1 - p_1 (1 - \exp(-g_1 \times t))} \exp(\alpha_1 \times m_{nl}(\psi_i, t))$$

$$h_{i2}(t, \psi_i; \theta) = \frac{1}{b} \times \frac{(1 - F_1(\infty)) \exp(-t/b)}{1 - (1 - F_1(\infty))(1 - \exp(-t/b))}$$