

mapbayr :: MAP – Bayesian Estimation in R



MODEL: *mrgmod*



1. Coder le modèle “classiquement”
<https://mrgsolve.org/>
2. Le modifier pour la compatibilité **mapbayr**
\$PARAM: Ajouter les ETA1=0, ETA2=0 etc...
\$MAIN: Ajouter ces ETA1, ETA2 au code
\$SIGMA: Err proportionnelle 1^{er}, additive 2^{ème}
\$CAPTURE: Capturer DV
\$CAPTURE: Capturer variable d'intérêt (CL...)
(\$CMT: Définir [ADM] et [OBS])
3. Importer: `model <- mread("model.cpp")`
4. Vérifier:
 - `check_mapbayr_model(model)`
 - Simulations rapides: `ev()`, `mrgsim()`
 - (Comparaison NONMEM. `?vs_nonmem`)

DATA: *data.frame*, *tibble*

Format **NM-TRAN** (tel NONMEM, MONOLIX...)
Variables: *ID*, *time*, *evid*, *cmt*, *amt*, *DV*, *mdv* etc...

- Données déjà existantes et formatées:
`data <- read.csv("path/to/file.csv")`
- Données à formater: `?data_helpers`
 - à l'aide d'un modèle: retourne un modèle
« avec les données à l'intérieur »

```
model %>%  
  adm_lines(amt = 100) %>%  
  obs_lines(time = 12, DV = 3.4) %>%  
  add_covariates(BW = 40) %>% get_data()  
- sans modèle: retourne les données  
adm_lines(amt = 100, cmt = 1) %>%  
obs_lines(time=12, DV=3.4, cmt=2) %>%  
add_covariates(BW = 40)
```

ESTIMATION : *mapbayests*

```
est <- mapbayest(model, data, ...)  
... : autres options ?mapbayest
```

<code>est\$model</code>	modèle original
<code>est\$arg.optim</code>	arguments pour usage interne
<code>est\$arg.ofv.fix</code>	
<code>est\$arg.ofv.id</code>	
<code>est\$opt.value</code>	résultat brut de l'algorithme
<code>est\$final_eta</code>	valeurs finales de eta
<code>est\$covariance</code>	matrice de covariance
<code>est\$information</code>	temps analyse, version package
<code>est\$mapbay_tab</code>	table finale, avec:
<ul style="list-style-type: none">- variables des données originales (ID, time...)- IPRED: Individual Prediction- PRED : Population Prediction (Patient typique)- Autres covariables éventuelles- Estimations de ETA1, ETA2...	

EXTRAIRE ÉLÉMENTS

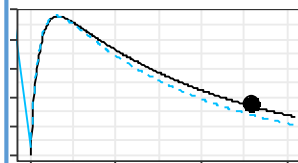
- La table finale (`est$mapbay_tab`):
`as.data.frame(est)` # `as_tibble(est)`
- Temps d'analyses et versions des packages
`est$information`
- `?get_x`
- Les valeurs de eta:
`get_eta(est)`
- Les variables capturées dans \$CAPTURE
`get_param(est, "CL")`
- Les données originales
`get_data(est)`
- Matrice de covariance (approximative)
`get_cov(est)`
- Une table semblable au .phi de NONMEM
`get_phi(est)`

SIMULATIONS A POSTERIORI

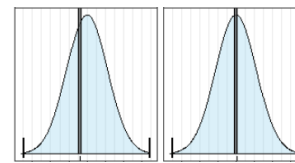
1. Mettre à jour le modèle avec valeurs patient
(ex: ETA1 = 0.123, BW = 40, OMEGA = 0, etc...)
`?use_posterior`
2. Simulations classiques... <https://mrgsolve.org/>
 - Vraies conditions d'adm et obs: `data_set()`
`est %>%
 use_posterior() %>%
 data_set(data) %>%
 mrgsim(observ=TRUE)`
 - Conditions d'adm et obs hypothétiques: `ev()`
`scenarios <- expand.ev(...)`
`est %>%
 use_posterior() %>%
 ev(scenarios) %>%
 mrgsim()`

METHODES PLOT : *ggplot*

- Conc vs time:
`?plot.mapbayests`
`plot(est)`



- Distrib param:
`?hist.mapbayests`
`hist(est)`



Objets ggplot modifiables avec ``+``