INFUSION TESTS

kyleb

Fri Sep 15 20:22:29 2017

```
Sys.setenv(RSTUDIO_PANDOC = "/usr/lib/rstudio-server/bin/pandoc")


.libPaths("/data/Rlibs")
library(mrgsolve)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(readr)
library(ggplot2)

carry <- c("cmt", "amt","ii", "addl", "rate", "evid", "ss")

to_data_set <- function(x,id = NULL) {
  x <- as.data.frame(x)
  x <- mutate(x, C = '.', DV = '.', cmt = if_else(cmt==0, 2, cmt))
  x <- dplyr::select(x, "C", everything())
  if(is.numeric(id)) x <- mutate(x,ID = id)
  x
}
sv <- function(x,file) {
  write.csv(file = file, row.names = FALSE, quote = FALSE, x)
}
nonr <- metrumrg::NONR
run <- function(number) {
  nonr(number, project = "model", command = "/opt/NONMEM/nm73/nmqual/autolog.pl", checkrunno=FALSE)
  return(tabread(number))
}
tabread <- function(number) {
  tab <- file.path("model", number, "TAB")
  if(file.exists(tab)) return(read_table(tab, skip=1))
  stop("the run failed")
}
sim <- function(x, e,...) {
  mrgsim(x, events = e, carry.out = carry, digits = 5, ...)
}


code <- '
```

```
$SET req = ""
$PARAM CL = 1, V = 30, KA = 1.5
LAGT = 0, MODE = 0, DUR2 = 2, RAT2 = 10, BIOAV = 1

$PKMODEL cmt = "GUT CENT", depot = TRUE

$MAIN

F_CENT = BIOAV;
ALAG_CENT = LAGT;

if(MODE==1) R_CENT = RAT2;
if(MODE==2) D_CENT = DUR2;


$TABLE
capture DV = (CENT/(V/1000));
capture CP = DV;

$CAPTURE LAGT MODE DUR2 RAT2 BIOAV
'

mod <- mcode_cache("tests1", code)
```

```
## Compiling tests1 ...
```

```
## done.
```

```
mod <- update(mod, end=72)
```

Bolus

```
ev <- ev(amt = 100, ii = 12, addl = 3)
out1 <- sim(mod,ev)
data1 <- to_data_set(out1, 1)
```

Infusion

```
ev <- ev(amt = 100, ii = 12, addl =3, rate = 100/10, cmt = 2)
out2 <- sim(mod,ev)
data2 <- to_data_set(out2, 2)
```

Infusion, lag

```
ev <- ev(amt = 100, ii = 12, addl=3, rate = 100/10, LAGT = 5, cmt = 2)
out3 <- sim(mod,ev)
data3 <- to_data_set(out3, 3)
```

Infusion, lag, bioav

```
ev <- ev(amt = 100, ii = 12, addl=3, rate = 100/10, LAGT = 5, BIOAV = 0.412, cmt = 2)
out4 <- sim(mod,ev)
data4 <- to_data_set(out4, 4)
```

Infusion, bioav, ss

```
ev <- ev(amt = 100, ii = 12, addl=3, rate = 100/10, LAGT = 0, BIOAV = 0.412, ss = 1, cmt = 2)
out5 <- sim(mod,ev)
data5 <- to_data_set(out5, 5)
```

Infusion, bioav, ss, II < DUR

```
ev <- ev(amt = 100, ii = 6, addl = 3, rate = 100/10, BIOAV = 0.812, ss = 1, cmt = 2)
out6 <- sim(mod,ev)
data6 <- to_data_set(out6, 6)
```

Infusion, ss, II < DUR

```
ev <- ev(amt = 100, ii = 6, addl = 3, rate = 100/10, ss = 1, cmt = 2)
out6.1 <- sim(mod,ev)
data6.1 <- to_data_set(out6.1, 6.1)
```

Infusion, ss, II multiple of DUR

```
ev <- ev(amt = 100, ii = 6, addl = 3, rate = signif(100/12,5), ss = 1, cmt = 2)
out6.2 <- sim(mod,ev)
data6.2 <- to_data_set(out6.2, 6.2)
```

Infusion, bioav, ss, II == DUR

```
ev <- ev(amt = 100, ii = 10, addl=3, rate = 100/10, LAGT = 0, BIOAV = 0.412, ss = 1, cmt = 2)
out7 <- sim(mod,ev)
data7 <- to_data_set(out7, 7)
```

Infusion,, ss, II == DUR

```
ev <- ev(amt = 100, ii = 10, addl=3, rate = 100/10, LAGT = 0, ss = 1, cmt = 2)
out7.1 <- sim(mod,ev)
data7.1 <- to_data_set(out7.1, 7.1)
```

Bolus, bioav, ss

```
ev <- ev(amt = 100, ii = 12, addl=3,  LAGT = 0, BIOAV = 0.412, ss = 1)
out8 <- sim(mod,ev)
data8 <- to_data_set(out8, 8)
```

Bolus, lag, bioav

```
ev <- ev(amt = 100, ii = 12, addl=3,  LAGT = 5, BIOAV = 0.412)
out9 <- sim(mod,ev)
data9 <- to_data_set(out9, 9)
```

Infusion / bolus

```
ev <- ev(amt = 100, rate = 10) + ev(time = 12, amt = 50)
out10 <- sim(mod,ev)
data10 <- to_data_set(out10, 10)
```

Infusion (D__) lag, BIOAV

```
ev <- ev(amt = 100, rate = -2, DUR2 = 9, MODE = 2, cmt = 2, ii = 12, addl = 5, LAGT = 5, BIOAV = 0.61)
out11 <- sim(mod,ev)
data11 <- to_data_set(out11,11)
```

Simulate

```
sims <- list(out1,out2,out3,out4,out5,out6,out6.1,out6.2,out7,out7.1,out8,out9,out10,out11)
sims <- lapply(sims, as.data.frame)
sims <- bind_rows(sims)
data <- bind_rows(data1,data2,data3,data4,data5,data6,data6.1,data6.2,data7,data7.1,data8,data9,data10,
sv(data, "data/101.csv")
out <- run(101)
```

```
## Run 101 complete.
```

```
## NONR complete.
```

```
## Parsed with column specification:
## cols(
##   TIME = col_double(),
##   EVID = col_double(),
##   CP = col_double(),
##   IPRED = col_double(),
##   PRED = col_double(),
##   DV = col_double()
## )
```

Overall Summary

```
dim(out)
```

```
## [1] 1037    6
```

```
dim(sims)
```

```
## [1] 1037   16
```

```
summary(out$CP - sims$CP)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##       0       0       0       0       0       0
```

```
data$NM <- out$CP
data$MRGSIM <- sims$CP
```

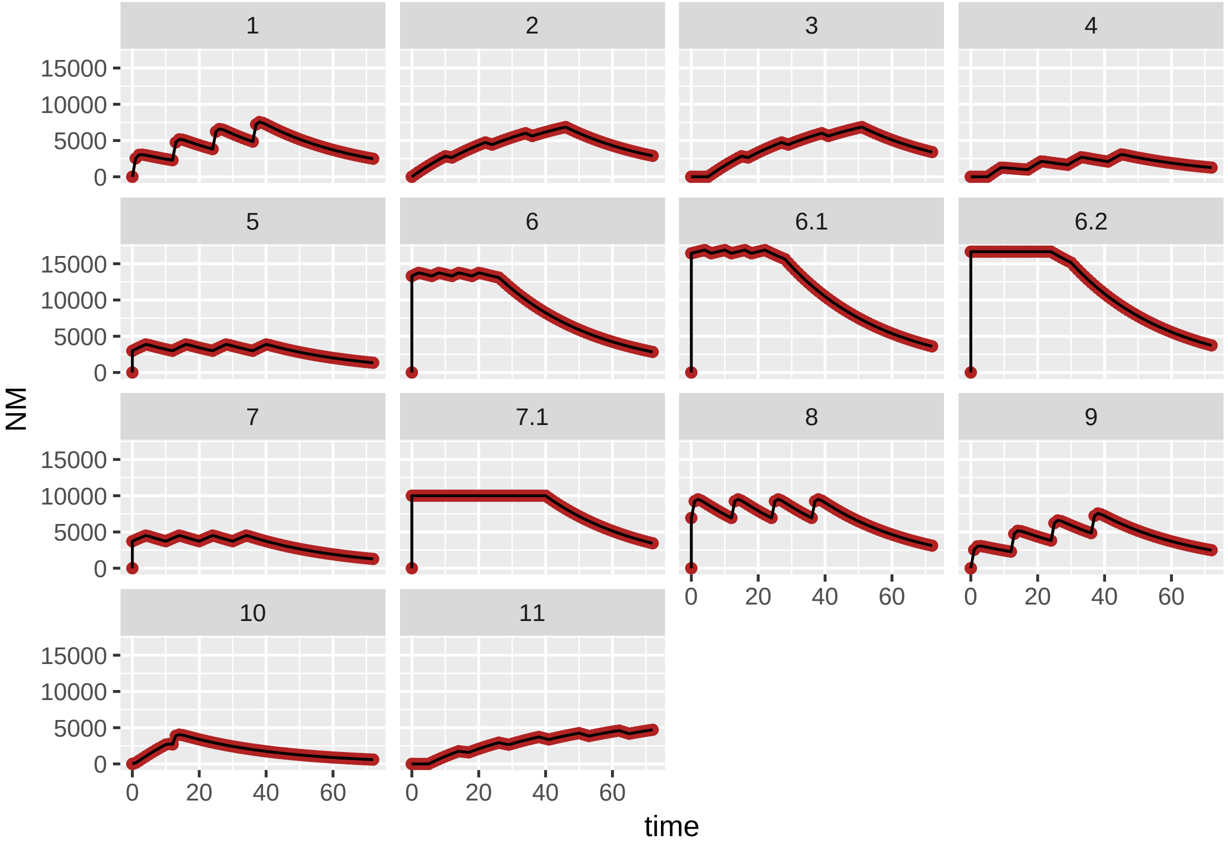
Summary by RUN

```
group_by(data,ID) %>%
  mutate(diff = NM - MRGSIM) %>%
  summarise(mean = mean(diff), max = max(diff), min = min(diff))
```

```
## # A tibble: 14 x 4
##       ID  mean   max   min
##    <dbl> <dbl> <dbl> <dbl>
##  1   1.0     0     0     0
##  2   2.0     0     0     0
##  3   3.0     0     0     0
##  4   4.0     0     0     0
##  5   5.0     0     0     0
##  6   6.0     0     0     0
##  7   6.1     0     0     0
##  8   6.2     0     0     0
##  9   7.0     0     0     0
## 10   7.1     0     0     0
## 11   8.0     0     0     0
## 12   9.0     0     0     0
## 13  10.0     0     0     0
## 14  11.0     0     0     0
```

Plot

```
ggplot(data = data) +
  geom_point(aes(time,NM),color = "firebrick") +
  geom_line(aes(time,MRGSIM,group = ID)) +
  facet_wrap(~ID)
```



Control stream

```
writeLines(readLines("model/101.ctl"))
```

```
$PROB RUN# 101

$INPUT C ID TIME EVID AMT CMT SS II ADDL RATE LAGT MODE DUR2 RAT2 BIOAV DV

$DATA ../../data/101.csv IGNORE=C

$SUBROUTINES ADVAN2 TRANS2

$PK

TVCL=THETA(1)
CL=TVCL*EXP(ETA(1))

TVV2=THETA(2)
V=TVV2*EXP(ETA(2))

TVKA=THETA(3)
KA=TVKA*EXP(ETA(3))

ALAG2 = LAGT
F2 = BIOAV
```

```
   IF(MODE.EQ.1) R2 = RAT2
   IF(MODE.EQ.2) D2 = DUR2

   $ERROR
   IPRED=A(2)/(V/1000)
   Y=IPRED*EXP(ERR(1))

   CP = IPRED

   $THETA
   (1,   FIX) ;; CL
   (30,  FIX) ;; V
   (1.5, FIX) ;; KA

   $OMEGA
   0.0 FIX
   0.0 FIX
   0.0 FIX

   $SIGMA
   0.00 FIX

   $TABLE FILE=TAB TIME EVID CP IPRED PRED DV NOPRINT ONEHEADER NOAPPEND

   $SIMULATION (2674474) ONLYSIMULATION
```

```
devtools::session_info()
```

```
## Session info -------------------------------------------------------------
```

```
##  setting  value
##  version  R version 3.3.3 (2017-03-06)
##  system   x86_64, linux-gnu
##  ui       X11
##  language (EN)
##  collate  en_US.UTF-8
##  tz       Etc/UTC
##  date     2017-09-15
```

```
## Packages -----------------------------------------------------------------
```

```
##  package       * version     date
##  assertthat      0.2.0       2017-04-11
##  backports       1.1.0       2017-05-22
##  base          * 3.3.3       2017-03-06
##  bindr           0.1         2016-11-13
##  bindrcpp      * 0.2         2017-06-17
##  colorspace      1.3-2       2016-12-14
##  datasets      * 3.3.3       2017-03-06
##  devtools        1.13.2      2017-06-02
##  digest          0.6.12      2017-01-27
##  dplyr         * 0.7.3       2017-09-09
##  evaluate        0.10.1      2017-06-24
##  fork            1.2.5       2017-07-26
##  ggplot2       * 2.2.1       2016-12-30
##  glue             1.1.1       2017-06-21
##  graphics       * 3.3.3       2017-03-06
##  grDevices      * 3.3.3       2017-03-06
##  grid             3.3.3       2017-03-06
##  gtable           0.2.0       2016-02-26
##  hms              0.3         2016-11-22
##  htmltools        0.3.6       2017-04-28
##  knitr            1.16        2017-05-18
##  labeling         0.3         2014-08-23
##  lattice          0.20-35     2017-03-25
##  lazyeval         0.2.0       2016-06-12
##  magrittr         1.5         2014-11-22
##  MASS             7.3-45      2016-04-21
##  memoise          1.0.0       2016-01-29
##  methods        * 3.3.3       2017-03-06
##  metrumrg         5.57        2015-10-08
##  mrgsolve       * 0.8.9.9001  2017-09-15
##  munsell          0.4.3       2016-02-13
##  pkgconfig        2.0.1       2017-03-21
##  plyr             1.8.4       2016-06-08
##  R6               2.2.2       2017-06-17
##  Rcpp             0.12.12     2017-07-15
##  RcppArmadillo    0.7.960.1.2 2017-08-29
##  readr          * 1.1.1       2017-05-16
##  reshape          0.8.6       2016-10-21
##  rlang            0.1.2       2017-08-09
##  rmarkdown        1.6         2017-06-15
##  rprojroot        1.2         2017-01-16
##  scales           0.5.0       2017-08-24
##  stats          * 3.3.3       2017-03-06
##  stringi          1.1.5       2017-04-07
##  stringr          1.2.0       2017-02-18
##  tibble           1.3.4       2017-08-22
##  tools            3.3.3       2017-03-06
##  utils          * 3.3.3       2017-03-06
##  withr            1.0.2       2016-06-20
##  XML              3.98-1.9    2017-06-19
##  yaml             2.1.14      2016-11-12
##  source
##  CRAN (R 3.3.3)
##  CRAN (R 3.3.3)
##  local
##  CRAN (R 3.3.3)
##  CRAN (R 3.3.3)
##  CRAN (R 3.3.2)
##  local
##  CRAN (R 3.3.3)
##  CRAN (R 3.3.3)
##  cran (@0.7.3)
##  CRAN (R 3.3.3)
##  local
##  CRAN (R 3.3.3)
##  CRAN (R 3.3.3)
##  local
##  local
##  local
##  CRAN (R 3.3.2)
##  CRAN (R 3.2.3)
##  CRAN (R 3.3.3)
##  CRAN (R 3.3.3)
##  CRAN (R 3.3.2)
##  CRAN (R 3.3.3)
##  CRAN (R 3.2.3)
##  CRAN (R 3.3.2)
##  CRAN (R 3.3.2)
##  CRAN (R 3.3.2)
##  local
##  R-Forge (R 3.2.3)
##  Github (metrumresearchgroup/mrgsolve@8cbf215)
##  CRAN (R 3.3.2)
##  CRAN (R 3.3.3)
##  CRAN (R 3.3.2)
##  CRAN (R 3.3.3)
##  CRAN (R 3.3.3)
##  cran (@0.7.960)
##  CRAN (R 3.3.3)
##  CRAN (R 3.2.3)
##  CRAN (R 3.3.3)
##  CRAN (R 3.3.3)
##  CRAN (R 3.3.3)
##  CRAN (R 3.3.3)
##  local
##  CRAN (R 3.3.3)
##  CRAN (R 3.3.3)
##  CRAN (R 3.3.3)
##  local
##  local
##  CRAN (R 3.3.2)
##  CRAN (R 3.3.3)
##  CRAN (R 3.3.2)
```