AUC-Compare Resuls

2023-10-12

```
# packages
library(tcpl)
```

```
## tcpl (v3.0.1) loaded with the following settings:
##   TCPL_DB:    C:/Program Files/R/R-4.2.2/library/tcpl/csv
##   TCPL_USER:  NA
##   TCPL_HOST:  NA
##   TCPL_DRVR:  tcplLite
## Default settings stored in tcpl config file. See ?tcplConf for more information.
```

```
library(tcplfit2)
library(DBI)
```

Load Sample Data

```
# Connect to DB
con <- dbConnect(drv = RMySQL::MySQL(), user="_dataminer", password = "pass", host = "ccte-mysql
-res.epa.gov") # this user and pass is a read-only connection
#dbGetQuery(con, "SHOW DATABASES") # show all available databases

tcplConf(user="_dataminer", pass="pass",
         db="invitrodb_test", drvr="MySQL", host="ccte-mysql-res.epa.gov")
#tcplConfList()

neuro.asids <- c(20) #IUF, UKN, Shafer, Mundy
neuro.aeids <- tcplLoadAeid(fld='asid',val=neuro.asids, add.fld=c('acid','acnm'))

# load sample data for mean firing rate
mc0 <- tcplPrepOtpt(tcplLoadData(lvl = 0L, fld = 'acid', val = 2471, type = 'mc'))
mc5 <- tcplPrepOtpt(tcplLoadData(lvl = 5L, fld = 'aeid', val = 2494, type = 'mc'))
```

Compare Hill Model Results (two chemicals)

AUC from the `get_AUC` function for the first chemical:

```
# find winning hill model
hill <- mc5[modl %in% "hill",]

# select DTXSID6020226 out of 22 other options
meth.mc0 <- mc0[dsstox_substance_id %in% 'DTXSID6020226',]
meth.mc5 <- mc5[dsstox_substance_id %in% 'DTXSID6020226',]

# prepare input for get_AUC
conc <- unique(meth.mc0$conc) # use concentration in mc0 which is not in log
fit_method <- "hill"

# test function
modpars <- list(tp = meth.mc5$tp, ga = meth.mc5$ga, p = meth.mc5$p,
                la = meth.mc5$la, q = meth.mc5$q, er = meth.mc5$er)
get_AUC(fit_method, min(conc), max(conc), ps = modpars)
```

```
## [1] 186.6553
```

```
# 186.6553
```

Kelly’s code for calculating AUC for the first chemical:

```
# create parameters for old function using updated tcplv3 parameters
mc5.mea.dev <- meth.mc5
mc5.mea.dev[, hill_tp := tp]
mc5.mea.dev[, hill_ga := ga]
mc5.mea.dev[, hill_gw := p] # per sarah explanation that 'hill_gw' and 'p' are the same
mc5.mea.dev$use.me <- 1

# function needed to fit, this is the same as loghill function in tcplfit2
hill_curve <- function(hill_tp, hill_ga, hill_gw, lconc){
  return(hill_tp/(1+10^((hill_ga - lconc)*hill_gw)))
}

mc5.mea.dev[use.me ==1L & modl == "hill",
            auc := mapply(function(lower,
                                   upper,
                                   hill_tp,
                                   hill_ga,
                                   hill_gw) integrate(hill_curve,
                                                      lower,
                                                      upper,
                                                      hill_tp=hill_tp,
                                                      hill_ga=hill_ga,
                                                      hill_gw=hill_gw)$value,
                          lower = mc5.mea.dev[use.me ==1L & modl == "hill", logc_min],
                          upper = mc5.mea.dev[use.me ==1L & modl == "hill", logc_max],
                          hill_tp = mc5.mea.dev[use.me ==1L & modl == "hill", hill_tp],
                          hill_ga = mc5.mea.dev[use.me ==1L & modl == "hill", hill_ga],
                          hill_gw = mc5.mea.dev[use.me ==1L & modl == "hill", hill_gw])]

mc5.mea.dev$auc
```

```
## [1] 109.3081
```

Discrepancy between results is due to where log transformation happens. `get_AUC` expect all inputs (parameters and concentration) to be in normal units. The function will do necessary log conversion internally and then take the integral. For hill model, `ga` is converted to `log10(ga)` and is used for estimating AUC in log scale.

I can alter Kelly's code to get the same result from my function:

```
mc5.mea.dev[use.me ==1L & modl == "hill",
            auc := mapply(function(lower,
                                   upper,
                                   hill_tp,
                                   hill_ga,
                                   hill_gw) integrate(hill_curve,
                                                      lower,
                                                      upper,
                                                      hill_tp=hill_tp,
                                                      hill_ga=hill_ga,
                                                      hill_gw=hill_gw)$value,
                          lower = mc5.mea.dev[use.me ==1L & modl == "hill", logc_min],
                          upper = mc5.mea.dev[use.me ==1L & modl == "hill", logc_max],
                          hill_tp = mc5.mea.dev[use.me ==1L & modl == "hill", hill_tp],
                          hill_ga = log10(mc5.mea.dev[use.me ==1L & modl == "hill", hill_ga]),
                          hill_gw = mc5.mea.dev[use.me ==1L & modl == "hill", hill_gw])]

mc5.mea.dev$auc
```

```
## [1] 186.6553
```

Or I can altar the input to `get_AUC` to get the same result from Kelly's code:

```
modpars <- list(tp = meth.mc5$tp, ga = 10^meth.mc5$ga, p = meth.mc5$p,
                la = meth.mc5$la, q = meth.mc5$q, er = meth.mc5$er)
get_AUC(fit_method, min(conc), max(conc), ps = modpars)
```

```
## [1] 109.3081
```

Check with one more chemical: Convert `ga` from mc5 table to `log10(ga)` :

```
# compare row 2 (DTXSID5024891) result to get_AUC function
# log10 the ga
mc5.mea.dev <- hill
mc5.mea.dev[, hill_tp := tp]
mc5.mea.dev[, hill_ga := ga]
mc5.mea.dev[, hill_gw := p]
mc5.mea.dev$use.me <- 1
mc5.mea.dev[use.me ==1L & modl == "hill",
            auc := mapply(function(lower,
                                   upper,
                                   hill_tp,
                                   hill_ga,
                                   hill_gw) integrate(hill_curve,
                                                      lower,
                                                      upper,
                                                      hill_tp=hill_tp,
                                                      hill_ga=hill_ga,
                                                      hill_gw=hill_gw)$value,
                          lower = mc5.mea.dev[use.me ==1L & modl == "hill", logc_min],
                          upper = mc5.mea.dev[use.me ==1L & modl == "hill", logc_max],
                          hill_tp = mc5.mea.dev[use.me ==1L & modl == "hill", hill_tp],
                          hill_ga = log10(mc5.mea.dev[use.me ==1L & modl == "hill", hill_ga]),
                          hill_gw = mc5.mea.dev[use.me ==1L & modl == "hill", hill_gw])]
round(mc5.mea.dev$auc,2)[2]
```

```
## [1] 126.21
```

```
# compare row 2 (DTXSID5024891) result to get_AUC function
meth.mc0 <- mc0[dsstox_substance_id %in% 'DTXSID5024891' & spid %in% "EPAPLT0167E09",]
meth.mc5 <- mc5[dsstox_substance_id %in% 'DTXSID5024891' & spid %in% "EPAPLT0167E09",]

conc <- unique(meth.mc0$conc)
fit_method <- "hill"

modpars <- list(tp = meth.mc5$tp, ga = meth.mc5$ga, p = meth.mc5$p,
                la = meth.mc5$la, q = meth.mc5$q, er = meth.mc5$er)
get_AUC(fit_method, min(conc), max(conc), ps = modpars)
```

```
## [1] 126.2112
```

```
#126.2112
```

Compare Gain-Loss Model results (one chemical)

For Gain-Loss model, `ga` and `la` are being converted to log10-scale internally by `get_AUC`.

```
# this is equivalent to loggnls function in tcplfit2
 gnls_curve <- function(top, ga, gw, la, lw, lconc){
   gain <- 1/(1+10^((ga - lconc)*gw))
   loss <- 1/(1+10^((lconc - la)*lw))
   return(top*gain*loss)
 }

# find winning gain-loss model, there are 13 options
gnls <- mc5[modl %in% "gnls",]

# select DTXSID5026259
meth.mc0 <- mc0[dsstox_substance_id %in% 'DTXSID5026259' & spid %in% "EX000364",]
meth.mc5 <- mc5[dsstox_substance_id %in% 'DTXSID5026259'& spid %in% "EX000364",]

conc <- unique(meth.mc0$conc)
fit_method <- "gnls"

# test function
modpars <- list(tp = meth.mc5$tp, ga = meth.mc5$ga, p = meth.mc5$p,
                la = meth.mc5$la, q = meth.mc5$q, er = meth.mc5$er)
get_AUC(fit_method, min(conc), max(conc), ps = modpars)
```

```
## [1] -47.90211
```

```
# -47.90211

mc5.mea.dev <- meth.mc5
mc5.mea.dev$use.me <- 1

mc5.mea.dev[modl == "gnls",
            auc := mapply(function(lower,
                                   upper,
                                   top,
                                   ga,
                                   gw,
                                   la,
                                   lw) integrate(gnls_curve,
                                                 lower,
                                                 upper,
                                                 top=top,
                                                 ga=ga,
                                                 gw=gw,
                                                 la=la,
                                                 lw=lw)$value,
                          lower = mc5.mea.dev[use.me ==1L & modl == "gnls", logc_min],
                          upper = mc5.mea.dev[use.me ==1L & modl == "gnls", logc_max],
                          top = mc5.mea.dev[use.me ==1L & modl == "gnls", tp],
                          ga = log10(mc5.mea.dev[use.me ==1L & modl == "gnls", ga]),
                          gw = mc5.mea.dev[use.me ==1L & modl == "gnls", p],
                          la = log10(mc5.mea.dev[use.me ==1L & modl == "gnls", la]),
                          lw = mc5.mea.dev[use.me ==1L & modl == "gnls", q])]


mc5.mea.dev$auc
```

```
## [1] -47.90211
```