VPC Demo

Samuel P Callisto, PhD

27 August 2021

Packages you may not be familiar with: - vpc - glue - here - data.table (preferred over read_csv, also works with NONMEM table files)

```
## Required packages
suppressPackageStartupMessages({
  library(tidyverse)
  library(mrgsolve)
  library(vpc)
  library(glue)
  library(here)
  library(data.table)
})

## making plots pretty
mrg_vpc_theme = new_vpc_theme(list(
  sim_pi_fill = "steelblue3", sim_pi_alpha = 0.5,
  sim_median_fill = "grey60", sim_median_alpha = 0.5
))

## programmatically set run number
## easy to run for multiple models this way
runno <- "r2"

## control sequence
writeplots <- TRUE
```

Load in the NONMEM data set, which we will use as a template for our VPC simulations

Question: What are we trying to determine by doing a VPC?

```
data <- data.table::fread(here("data","dat1.csv"), na=".") %>%
  ## add columns necessary for mrgsim
  mutate(
    evid=mdv,
    DOSE=ifelse(amt==0,NA_real_,amt),
    cmt=1
  ) %>%
  ## fill DOSE within a subject
  group_by(ID) %>%
  fill(DOSE, .direction="downup") %>%
  ungroup()
head(data)
```

```
## # A tibble: 6 x 10
```

```
##      ID  time   amt    u1     dv   mdv   NUM  evid  DOSE   cmt
##   <int> <dbl> <int> <dbl>  <dbl> <int> <int> <int> <dbl> <dbl>
## 1     1   0     100  NA    NA         1     1     1   100     1
## 2     1   0.5     0  87.5  1.70       0     2     0   100     1
## 3     1   1       0  76.6  1.44       0     3     0   100     1
## 4     1   1.5     0  67.0  1.56       0     4     0   100     1
## 5     1   2       0  58.6  1.16       0     5     0   100     1
## 6     1   4       0  34.3  0.544      0     6     0   100     1
```

```
#' # Simulate the vpc
#'
#' ## Load the mrgsolve model
#'
#' This should reflect `../model/nonmem/r2.ctl`
mod <- mread(glue("../model/sim/{runno}.cpp"))
```

```
## Building r2_cpp ... done.
```

```
#' # Set up the simulation
#'
#' Create a function to simulate out one replicate
sim <- function(rep, data, model) {
  mrgsim(
    model,
    data = data,
    carry_out = "DOSE,evid,NUM",
    Req = "Y",
    output = "df",
    quiet  = TRUE
  ) %>%  mutate(irep = rep)
}

#' Simluate data

#' 200 replicates
isim <- seq(200)

set.seed(86486)
sims <- lapply(
  isim, sim,
  data = data,
  mod = mod
) %>%
  bind_rows()

sum(sims$Y)
```

```
## [1] 102732.8
```

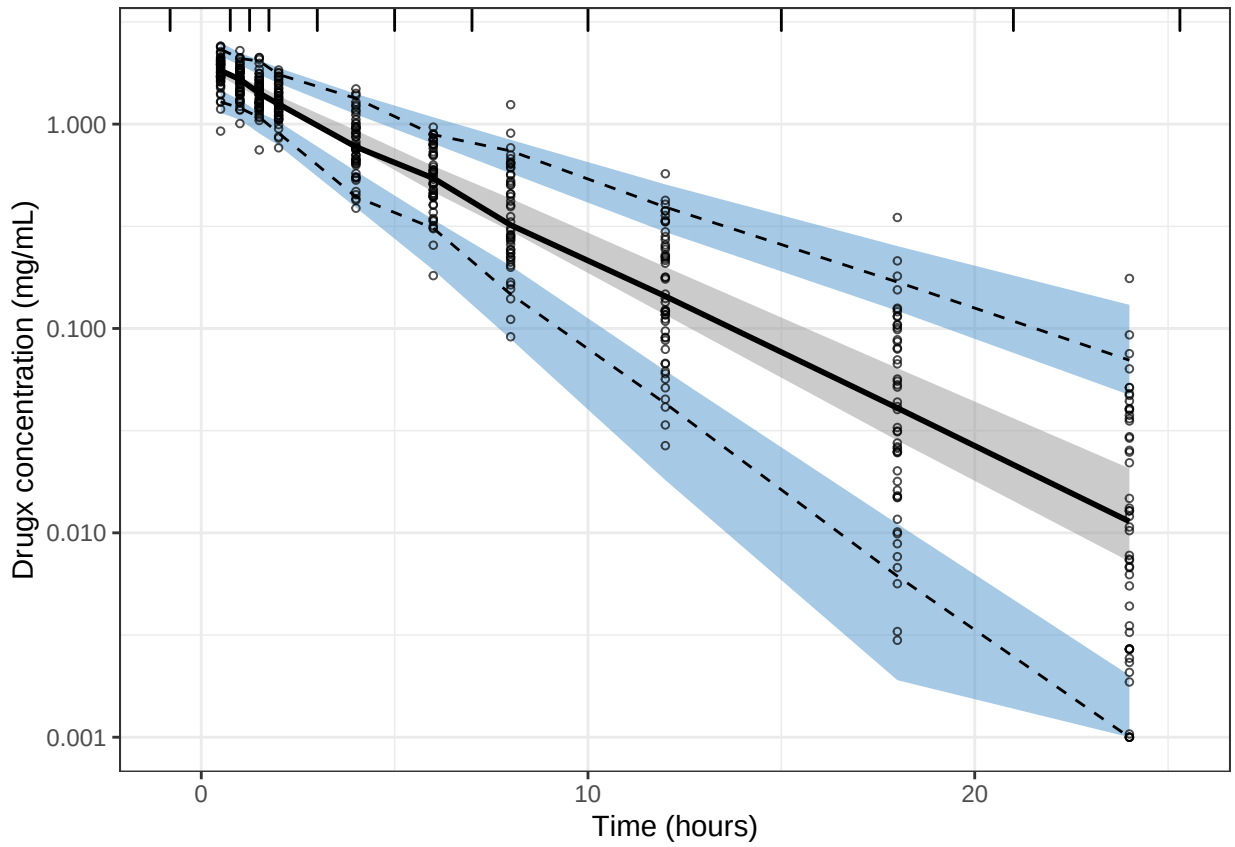
```
#' Filter both the observed and simulated data
fdata <-  rename(data, id=ID) %>%
  ## remove dosing rows
  filter(evid == 0)
fsims <- rename(sims, id=ID) %>%
  ## remove dosing rows
  filter(evid == 0)
```

Creating the VPC plot

The vpc function from Ron Keizer's vpc package does most of the heavy lifting here. The documentation and vignettes on this package are sparse, but it is beneficial to spend some time reading the help file for the vpc function to see all the different argument options.

```
#' # Create the plot
#'
#' Pass observed and simulated data into vpc function
p1 <- vpc(
  obs = fdata,
  sim = fsims,
  obs_cols = list(dv = "dv"),
  sim_cols=list(dv="Y", sim="irep"),
  log_y = TRUE,
  pi = c(0.05, 0.95),
  ci = c(0.025, 0.975),
  show = list(obs_dv = TRUE),
  vpc_theme = mrg_vpc_theme
) +
  ## use ggplot to alter aesthetics
  theme_bw() +
  xlab("Time (hours)") +
  ylab("Drugx concentration (mg/mL)")

p1
```



```
## saves to png/pdf
if(writeplots){
  mrggsave::mrggsave(p1, stem = "pk-vpc-{runno}", height = 7.5,
                     dir = here("deliv","figure"), script = "pk-vpc-final.R")
}
```

As the analyst we have a couple of options with our VPC: - stratify by dose (or other relevant covariate!) - dose-normalized (e.g. `DVN = DV / DOSE`) - prediction-corrected VPC

Question: What would be a case when we would need to consider these options?

Question: How could we determine if the model will accurately predict future data?

Pred-corrected VPC

Typically we only need to do this if we have different doses in the data set. But I will just do one here so you can see the required input

Need to get PREDs. Two options: - can pull from NONMEM table file - can simulate from model using NONMEM

```
tab <- data.table::fread(
  here("model","nonmem",runno,glue("{runno}.tab")), skip=1
)
head(tab)
```

```
##    ID NUM TIME AMT      DV MDV   IPRED    PRED    CWRES     NPDE
## 1:  1   1  0.0 100 0.00000   1 2.01670 2.01640  0.00000  0.00000
## 2:  1   2  0.5   0 1.70170   0 1.76850 1.81330 -0.28321 -0.33185
## 3:  1   3  1.0   0 1.43610   0 1.55080 1.63070 -0.56012 -0.53401
## 4:  1   4  1.5   0 1.56280   0 1.35990 1.46650  0.76619  0.80642
## 5:  1   5  2.0   0 1.15740   0 1.19250 1.31890 -0.36709 -0.33185
## 6:  1   6  4.0   0 0.54394   0 0.70519 0.86262 -1.66840 -1.42790
```

Key reference: Bergstrand, 2011 https://pubmed.ncbi.nlm.nih.gov/21302010/

Update data sets to include PREDs

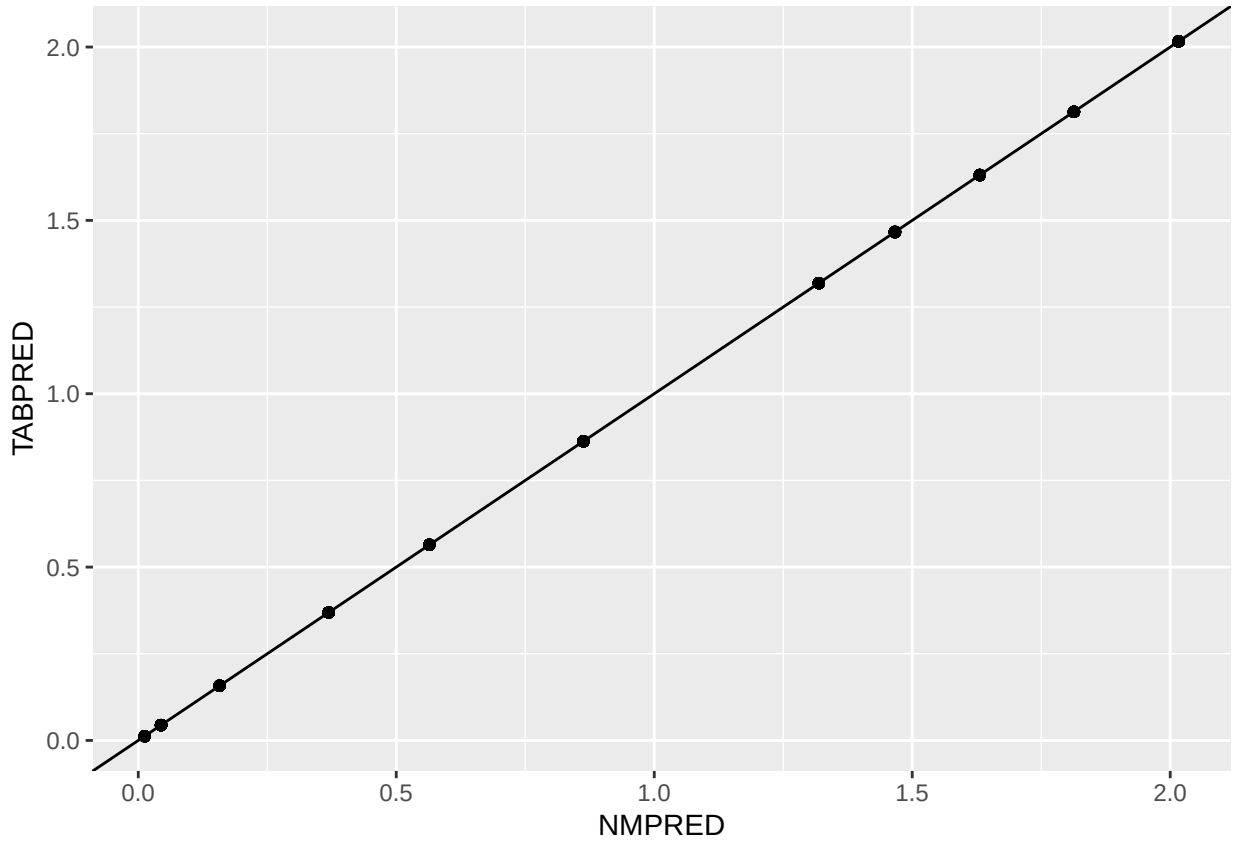
```
#' Now, remove random effects and simulate PREDs
pred <- sim(
  rep=1,
  data = data,
  mod = zero_re(mod)
)
```

compare NONMEM and mrgsolve PREDs

They match entirely, so it doesn't matter which method you use. PREDs are deterministic since they zero-out the random effects.

```
full_join(
  rename(tab, TABPRED=PRED),
  rename(pred, NMPRED=Y)
) %>%
  ggplot() +
  geom_point(aes(x=NMPRED,y=TABPRED)) +
  geom_abline(intercept=0, slope=1)
```

```
## Joining, by = c("ID", "NUM")
```




```
## modify data to include obs only and append PREDs
fdata <- data %>%
  rename(id=ID) %>%
  full_join(
    select(pred, NUM, PRED=Y)
  ) %>% filter(evid == 0)
```

```
## Joining, by = "NUM"
```

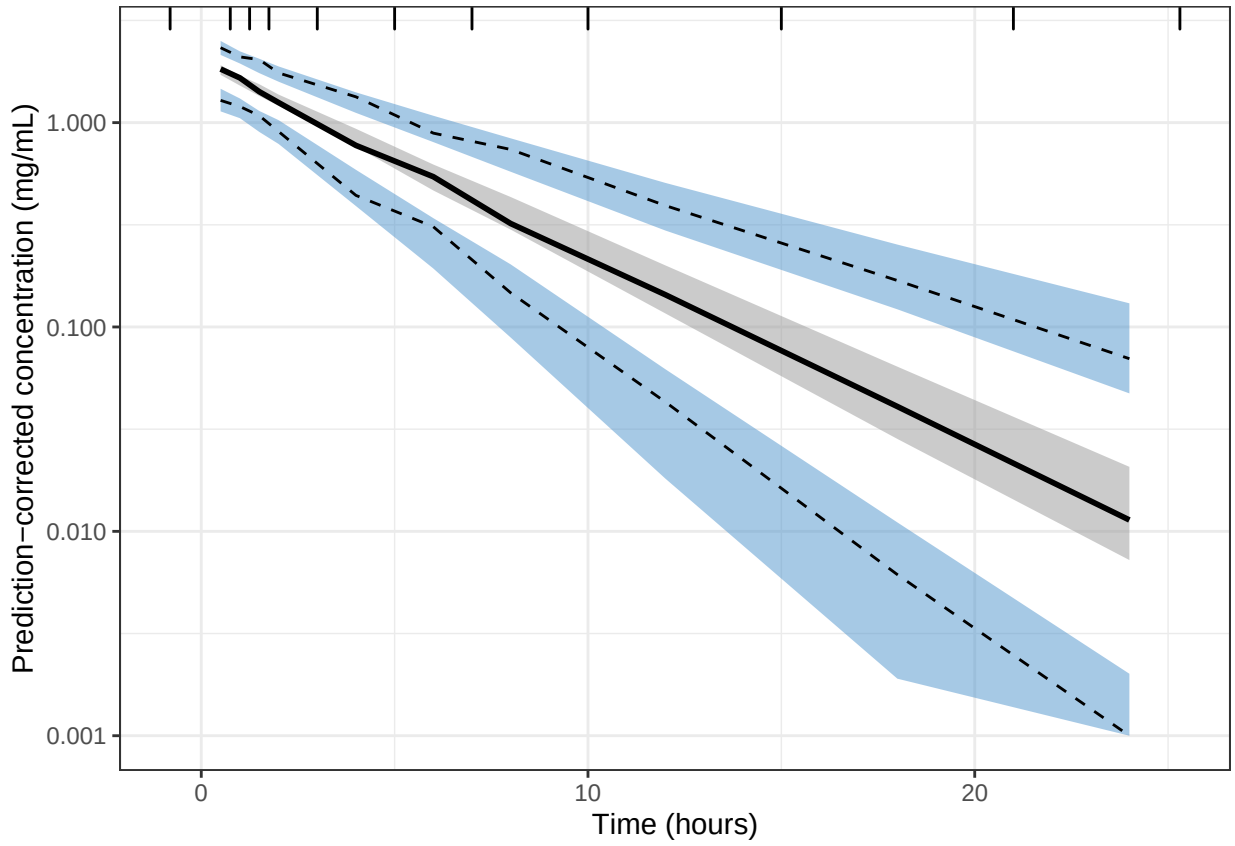
```
fsims <-  sims %>%
  rename(id=ID) %>%
  full_join(
    select(pred, NUM, PRED=Y)
  ) %>% filter(evid == 0)
```

```
## Joining, by = "NUM"
```

```
#' # Create the plot
#'
#' Pass observed and simulated data into vpc function
pc1 <- vpc(
  obs = fdata,
  sim = fsims,
  pred_corr = TRUE,
  obs_cols = list(dv = "DV", idv="time", pred="PRED"),
  sim_cols=list(dv="Y", idv="time", pred="PRED", sim="irep"),
  log_y = TRUE,
  pi = c(0.05, 0.95),
```

```
  ci = c(0.025, 0.975),
  vpc_theme = mrg_vpc_theme
) +
    ## use ggplot to alter aesthetics
  theme_bw() +
  xlab("Time (hours)") +
  ylab("Prediction-corrected concentration (mg/mL)")

pc1
```




```
## write plots
if(writeplots){
  mrggsave::mrggsave(p1, stem = "pk-vpc-{runno}-pred-corr", height = 7.5,
                     dir = here("deliv","figure"), script = "pk-vpc-final.R")
}
```