R-script

```
sim.GxT.cross <- function(CV, theta0 = 0.95, theta1 = 0.80, theta2 = 1.25, target = 0.80,
                          groups = 2, capacity = 24, n, split = c(0.5, 0.5),
                          mue = c(0.95, 1 / 0.95), level = 0.05, setseed = TRUE, nsims = 1e5,
                          progr = TRUE, print = TRUE, details = FALSE) {
  require(PowerTOST)
  ##############################################################################################
  # Performs simulations of the G×T interaction test of 2×2×2 studies.                         #
  # Model 1: group, sequence, treatment, subject (nested within group × sequence),             #
  #          period (nested within group), group-by-sequence interaction,                      #
  #          group-by-treatment interaction                                                    #
  # linear model (all effects fixed) = ANOVA                                                   #
  # ------------------------------------------------------------------------------------------ #
  # This code is copyright © 2023 by Helmut Schütz and is open source; you can redistribute it #
  # and/or modify it under the terms of the GNU General Public License as published by the Free #
  # Software Foundation; either version 3, or (at your option) any later version. See the GNU GPL #
  # for more details Copies of the GPL-3 versions are available at:                            #
  # https://www.gnu.org/licenses/gpl-3.0.html                                                  #
  ##############################################################################################
  # Arguments:
  #   CV       : assumed intra-subject CV; can be a two-element vector - in this case the sample
  #              size is estimated based on the pooled CV;
  #              first element CV of group 1, second CV of group 2
  #   theta0   : assumed T/R-ratio for sample size estimation (default 0.95)
  #   theta1   : lower BE-limit (default 0.80)
  #   theta2   : upper BE-limit (default 1.25)
  #   target   : targetpower (default 0.80)
  #   groups   : number of groups (default 2)
  #   capacity : maximum capacity of the clinical site (default 24)
  #   n        : vector of group sizes, where length(n) == 2
  #              (optional; if given, the argument split is ignored)
  #   split    : group sizes / total sample size (default c(0.5, 0.5))
  #              must be a vector, where
  #              length(split) == groups & sum(split) == 1
  #              note: May lead to unbalanced sequences within groups!
  #   mue      : GMRs of groups
  #              must be a vector, where length(mue) == groups
  #              if all elements are equal: no true G×T interaction
  #   level    : level of the G×T test (default 0.05)
  #   setseed  : should a fixed seed issued? (default TRUE)
  #              if FALSE, random seeds are used
  #   nsims    : number of simulations (default 1e5)
  #   progr    : should a progress bar be shown? (default TRUE)
  #   print    : should summary of p(G×T) be shown? (default TRUE)
  #   details  : should the runtime be shown? (default FALSE)
  # Returns a results as a list with elements:
  #   GMR.w    : GMR weighted by number of subjects within group
  #   CVw      : CV weighted by number of subjects within group
  #   sig.pGxT : fraction of simulated studies with significant p(G×T)
  #   bin.level: significance level of the binomial test at 0.05
  #   ks       : Kolmogorov-Smirnov test of p(G×T) vs. uniform {0, 1}
  #   txt      : aggregated results
  #   summary  : descriptive statistics of p(G×T)
  #   runtime  : runtime in minutes
  # Cave: very long runtime
  ########################
  # Generate study data #
  ########################
  group.data <- function(CV = CV, mue = mue, n.group = n.group, capacity = capacity) {
    # generates simulated subject data
    if (length(n.group) < 2) stop("At least two groups required.")
    if (max(n.group) > capacity)
      warning("Largest group exceeds capacity of site!")
    subject <- rep(1L:sum(n.group), each = 2)
    group   <- period <- treatment <- sequence <- NULL
    for (i in seq_along(n.group)) { # half of the subjects in sequence TR, the other in RT
      sequence  <- c(sequence, c(rep("TR", n.group[i]),
                                 c(rep("RT", n.group[i]))))
      treatment <- c(treatment, rep(c("T", "R"), ceiling(n.group[i] / 2)),
                                 rep(c("R", "T"), floor(n.group[i] / 2)))
      period    <- c(period, rep(c(1:2), ceiling(n.group[i] / 2)),
                             rep(c(1:2), floor(n.group[i] / 2)))
      group     <- c(group, rep(i, ceiling(n.group[i])),
                            rep(i, floor(n.group[i])))
    }
    data    <- data.frame(subject, group, sequence, treatment, period, Y = NA_real_)
    for (i in seq_along(n.group)) {
      # lognormal distribution
      if (length(CV) == 1 ) { # equal variances of groups
        data$Y[data$group == i & data$treatment == "T"] <-
          exp(mue[i] + rnorm(n = n.group[i], mean = 0, sd = CV2se(CV)))
        data$Y[data$group == i & data$treatment == "R"] <-
          exp(1 + rnorm(n = n.group[i], mean = 0, sd = CV2se(CV)))
```

```
      } else {               # unequal variances of groups
        if (i == 1) { # first group
          data$Y[data$group == i & data$treatment == "T"] <-
            exp(mue[i] + rnorm(n = n.group[i], mean = 0, sd = CV2se(CV[1])))
          data$Y[data$group == i & data$treatment == "R"] <-
            exp(1 + rnorm(n = n.group[i], mean = 0, sd = CV2se(CV[1])))
        } else {      # second group
          data$Y[data$group == i & data$treatment == "T"] <-
            exp(mue[i] + rnorm(n = n.group[i], mean = 0, sd = CV2se(CV[2])))
          data$Y[data$group == i & data$treatment == "R"] <-
            exp(1 + rnorm(n = n.group[i], mean = 0, sd = CV2se(CV[2])))
        }
      }
    }
    facs       <- c("subject", "group", "sequence", "treatment", "period")
    data[facs] <- lapply(data[facs], factor) # factorize for lm()
    return(data)
  }
  ########################
  # Initial computations #
  ########################
  CV.p <- CV
  if (!length(CV) == 1) CV.p <- mse2CV(mean(CV2mse(CV)))
  if (missing(n)) { # sample size estimation
    tmp     <- sampleN.TOST(CV = CV.p, theta0 = theta0, theta1 = theta1,
                            theta2 = theta2, design = "2x2x2",
                            targetpower = target, print = FALSE)
    n       <- tmp[["Sample size"]]
    pwr     <- tmp[["Achieved power"]]
    n.group <- as.integer(n * split)
  } else {
    pwr     <- power.TOST(CV = CV.p, theta0 = theta0, theta1 = theta1,
                          theta2 = theta2, design = "2x2x2", n = sum(n))
    n.group <- n
  }
  sig     <- 0L           # counter of significant GxT interactions
  p.GxT   <- numeric(length = nsims)
  if (sum(n.group) < sum(n)) { # increase size of last group if necessary
    n.group[groups] <- n.group[groups] + n - sum(n.group)
  }                      # TODO: Check & correct (add another group?)
  GMR.w   <- prod(mue^n.group)^(1 / sum(n.group)) # weighted GMR
  CV.w    <- mse2CV(mean(CV2mse(CV)))             # weighted CV
  rt      <- proc.time()[[3]]                     # start timer
  if (setseed) set.seed(123456)                   # for reproducibility
  if (progr) pb <- txtProgressBar(style = 3)
  ###############
  # Simulations #
  ###############
  ow      <- options()    # safe defaults
  options(contrasts = c("contr.treatment", "contr.poly"), digits = 12)
  for (sim in 1:nsims) {  # the workhorse
    data       <- group.data(CV = CV, mue = mue, n.group = n.group,
                             capacity = capacity) # get simulated data
    # nested interaction model 1 (all effects fixed)
    model1     <- lm(log(Y) ~ group +
                             sequence +
                             treatment +
                             subject %in% (group*sequence) +
                             period %in% group +
                             group:sequence +
                             group:treatment,
                             data = data)
    p.GxT[sim] <- anova(model1)[["group:treatment", "Pr(>F)"]]
    if (p.GxT[sim] < level) { # significant GxT interaction
      sig <- sig + 1L
    }
    if (progr) setTxtProgressBar(pb, sim / nsims)
  }
  options(ow)             # restore defaults
  sig.pGxT <- sig / nsims # fraction of significant p(G×T)
  rt       <- signif((proc.time()[[3]] - rt) / 60, 3) # runtime (minutes)
  if (progr) close(pb)
  # Kolmogorov-Smirnov test of p(G×T) vs. standard uniform {0, 1}
  # exact if x < 100 and no ties, approximate otherwise
  ks       <- ks.test(x = p.GxT, y = "punif", 0, 1)
  ################
  # Prepare plot #
  ################
  unif <- qqplot(x = qunif(ppoints(nsims)), y = p.GxT, plot.it = FALSE)
  main <- paste0(groups, " groups (", paste(n.group, collapse=", "), ")")
  if (length(unique(mue)) == 1) {
    if (groups == 2) {
      main <- paste0(main, "\nGMR of both groups ", sprintf("%.4f", mue[1]))
      if (isTRUE(all.equal(CV[1], CV[2]))) {
        main <- paste0(main, "\nCV of both groups ", paste0(sprintf("%.4f", CV[1]), "\n"))
      } else {
        main <- paste0(main, "\nCVs (", paste(sprintf("%.4f", CV), collapse=", "), ")")
        main <- paste0(main, sprintf(", weighted %.4f", CV.w))
      }
```

```
    } else {
      main <- paste0(main, "\nGMR of all groups ",
                     paste(sprintf("%.4f", mue[1]), collapse=", "), "\n")
    }
  } else {
    main <- paste0(main, "\nGMRs (", paste(sprintf("%.4f", mue), collapse=", "), ")")
    if (length(unique(n.group)) == 1) {
      main <- paste0(main, ", pooled ", sprintf("%.4f", GMR.w))
      if (isTRUE(all.equal(CV[1], CV[2]))) {
        main <- paste0(main, "\nCV of both groups ", paste0(sprintf("%.4f", CV[1]), "\n"))
      } else {
         main <- paste0(main, "\nCVs (", paste(sprintf("%.4f", CV), collapse=", "), ")")
         main <- paste0(main, sprintf(", weighted %.4f", CV.w))
      }
    } else {
      main  <- paste0(main, ", weighted ", sprintf("%.4f", GMR.w))
      if (isTRUE(all.equal(CV[1], CV[2]))) {
        main <- paste0(main, "\nCV of both groups ", paste0(sprintf("%.4f", CV[1]), "\n"))
      } else {
         main <- paste0(main, "\nCVs (", paste(sprintf("%.4f", CV), collapse=", "), ")")
         main <- paste0(main, sprintf(", weighted %.4f", CV.w))
      }
    }
  }
  if (ks$p.value == 0) {
    sub <- paste0(ks$method, sprintf(": p <%.2g", .Machine$double.eps))
  } else {
    sub <- paste0(ks$method, sprintf(": p %.4f", ks$p.value))
  }
  ########
  # Plot #
  ########
  dev.new(width = 4.6, height = 4.6)
  op <- par(no.readonly = TRUE) # save defaults
  par(pty = "s", font.main = 1, cex.main = 0.9, cex.lab = 1, cex.axis = 0.8, cex.sub = 0.9,
      xaxs = "i", yaxs = "i", las = 1)
  plot(x = c(0, 1), y = c(0, 1), type = "n", axes = FALSE, main = main, sub = sub,
       xlab = "uniform {0, 1} quantiles", ylab = expression(italic(p(G%*%T))))
  axis(1)
  axis(1, at = seq(0, 1, 0.05), tcl = -0.25, labels = FALSE)
  axis(2)
  axis(2, at = seq(0, 1, 0.05), tcl = -0.25, labels = FALSE)
  grid()
  abline(a = 0, b = 1, col="lightgray") # unity line
  abline(h = level, lty = 2)            # at the level of the GxT test
  points(unif$x[unif$y < level], unif$y[unif$y < level], col = "red", pch = 19, cex = 0.05)
  points(unif$x[unif$y >= level], unif$y[unif$y >= level], col = "blue", pch = 19, cex = 0.05)
  x <- max(unif$x[unif$y < level])
  y <- max(unif$y[unif$y < level])
  lines(x = rep(x, 2), y = c(-0.05, level), col = "red")
  axis(1, at = x, tcl = -0.25, labels = FALSE, col.ticks = "red")
  mtext(sprintf("%.4f", sig / nsims), side = 1, line = 0.1, at = sig / nsims, cex = 0.8)
  box()
  par(op)                         # restore defaults
  ####################
  # Aggregate results #
  ####################
  bin.level <- binom.test(0.05 * nsims, nsims, alternative = "less")$conf.int[2]
  sumry     <- summary(p.GxT)
  if (details) cat("Runtime for", formatC(nsims, format = "d", big.mark = ","),
                   "simulations:", rt, "minutes\n")
  if (print) {
    txt <- paste0("p(G\u00D7T) <", level, sprintf(" in %4g%% of ", 100 * sig.pGxT),
                  formatC(nsims, format = "d", big.mark = ","),
                  " simulated studies.")
    if (length(unique(mue)) == 1) {
      txt <- paste0(txt, "\nG\u00D7T correctly not detected",
                    sprintf(" in %.3g%% of cases", 100 * (1 - sig.pGxT)))
    } else {
      txt <- paste0(txt, "\nG\u00D7T falsely not detected",
                    sprintf(" in %.3g%% of cases", 100 * (1 - sig.pGxT)))
    }
    if (sig.pGxT > bin.level) {
      txt <- paste0(txt, "\nBinomial test significant at the 0.05 level.")
    } else {
      txt <- paste0(txt, "\nBinomial test not significant at the 0.05 level.")
    }
    txt <- paste0(txt, "\nSummary of p(G\u00D7T):\n")
    cat(txt)
    print(round(sumry, 6))
  }
  eval <- data.frame(pGxT = sig.pGxT, limit = bin.level, binomial = "not significant")
  if (sig.pGxT > bin.level) eval$binomial <- "significant"
  res  <- list(n = n.group, mue = mue, CV = CV, GMR.w = GMR.w, CV.w = CV.w, eval = eval,
               summary = sumry, ks = ks, runtime = rt)
  return(res)
} # EOF sim.GxT.cross()
```

```
#############
# Scenarios #
#############
CV <- 0.335 # for 1–9
# note: argument ratio in the function CVp2CV() is the ratio of variances
# for a quick check set nsims = 1e3
# to show the results in full precision use print(x)

# 1. no G×T, equal group sizes, and GMRs
r <- 1 # equal variances of groups
x <- sim.GxT.cross(CV = CVp2CV(CV, ratio = r), theta0 = 1, target = 0.90,
                   mue = rep(1, 2))

# 2. no G×T, equal group sizes, and GMRs
r <- 1 / 1.5 # unequal variances of groups (CV1 < CV2)
x <- sim.GxT.cross(CV = CVp2CV(CV, ratio = r), theta0 = 1, target = 0.90,
                   mue = rep(1, 2))

# 3. no G×T, equal group sizes, and GMRs
r <- 1.5 # unequal variances of groups (CV1 > CV2)
x <- sim.GxT.cross(CV = CVp2CV(CV, ratio = 1.5), theta0 = 1, target = 0.90,
                   mue = rep(1, 2))

# 4. no G×T, unequal group sizes, equal GMRs
r <- 1 # equal variances of groups
x <- sim.GxT.cross(CV = CVp2CV(CV, ratio = r), theta0 = 1, target = 0.90,
                   split = c(1 - 10 / 48, 10 / 48), mue = rep(1, 2))

# 5. true G×T, equal groups sizes, GMR2 = 1 / GMR1
r <- 1 # equal variances of groups
x <- sim.GxT.cross(CV = CVp2CV(CV, ratio = r), theta0 = 1, target = 0.90,
                   mue = c(0.95, 1 / 0.95))

# 6. true G×T, equal groups sizes, GMR2 = 1 / GMR1
r <- 1 / 1.5 # unequal variances of groups (CV1 < CV2)
x <- sim.GxT.cross(CV = CVp2CV(CV, ratio = r), theta0 = 1, target = 0.90,
                   mue = c(0.95, 1 / 0.95))

# 7. true G×T, equal groups sizes, GMR2 = 1 / GMR1
r <- 1.5 # heteroscedasticity (CV1 > CV2)
x <- sim.GxT.cross(CV = CVp2CV(CV, ratio = 1.5), theta0 = 1, target = 0.90,
                   mue = c(0.95, 1 / 0.95))

# 8. true G×T, unequal groups sizes, GMR1 ~ 1.0605, GMR2 = 0.80
r <- 1 # equal variances of groups
x <- sim.GxT.cross(CV = CVp2CV(CV, ratio = r), theta0 = 1, target = 0.90,
                   split = c(1 - 10 / 48, 10 / 48),
                   mue = c(1.0604803726, 0.80))

# 9. true G×T, unequal groups sizes, GMR1 ~ 1.0605, GMR2 = 0.80
r <- 1 / 1.5 # unequal variances of groups (CV[1] < CV[2])
x <- sim.GxT.cross(CV = CVp2CV(CV, ratio = r), theta0 = 1, target = 0.90,
                   split = c(1 - 10 / 48, 10 / 48),
                   mue = c(1.0604803726, 0.80))

# 10. true G×T, unequal groups sizes, GMR1 ~ 1.0605, GMR2 = 0.80
r <- 1.5 # unequal variances of groups (CV1 > CV2)
x <- sim.GxT.cross(CV = CVp2CV(CV, ratio = r), theta0 = 1, target = 0.90,
                   split = c(1 - 10 / 48, 10 / 48),
                   mue = c(1.0604803726, 0.80))

# large sample sizes
CV <- 0.3 # for 11–12

# 11. no G×T, equal group sizes, and GMRs
r <- 1 # equal variances of groups
x <- sim.GxT.cross(CV = CVp2CV(CV, ratio = r), theta0 = 0.90, target = 0.80,
                   n = c(40, 40),
                   mue = rep(0.90, 2))

# 12. true G×T, unequal groups sizes, GMR1 ~ 0.8290, GMR2 = 1.25
r <- 1.5 # unequal variances of groups (CV1 > CV2)
x <- sim.GxT.cross(CV = CVp2CV(CV, ratio = r), theta0 = 0.90, target = 0.80,
                   n = c(64, 16),
                   mue = c(0.829040283289, 1.25))
```

Simulations

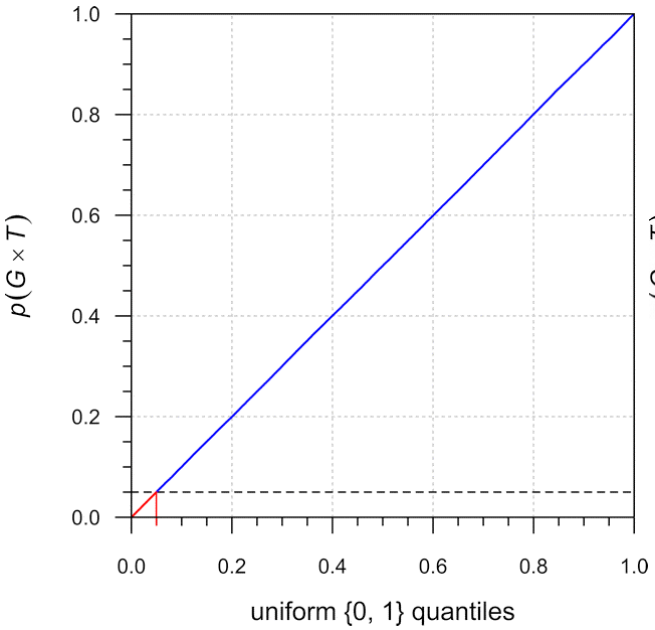

Fig. 1 Scenario 1: equal group sizes (n_1=n_2=24), homoscedasticity (CV_w=33.5%), no true Group-by-Treatment interaction (GMR_1=GMR_2=1.0)
p(G×T)=0.04967 (≤0.0511), p(unif.)=0.7563
G×T correctly not detected in 95.0% of cases

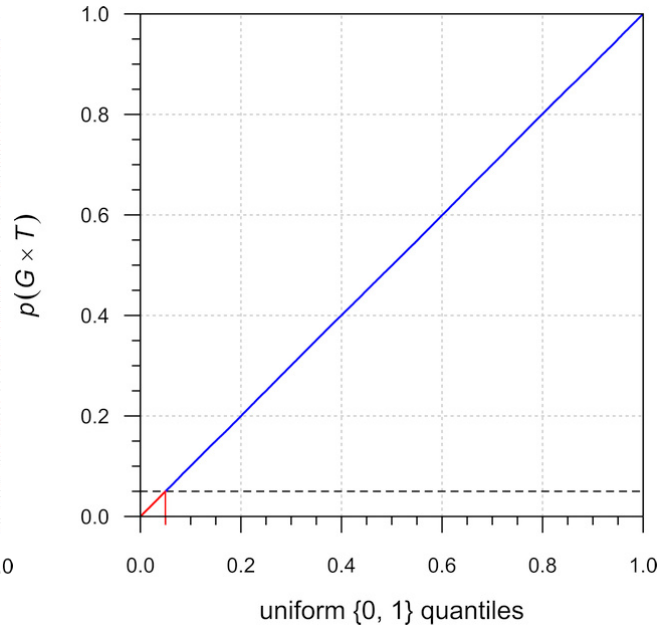

Fig. 2 Scenario 2: equal group sizes (n_1=n_2=24), heteroscedasticity (CV_w=29.8%, 36.9%), no true Group-by-Treatment interaction (GMR_1=GMR_2=1.0)
p(G×T)=0.04971 (≤0.0511), p(unif.)= 0.8944
G×T correctly not detected in 95.0% of cases

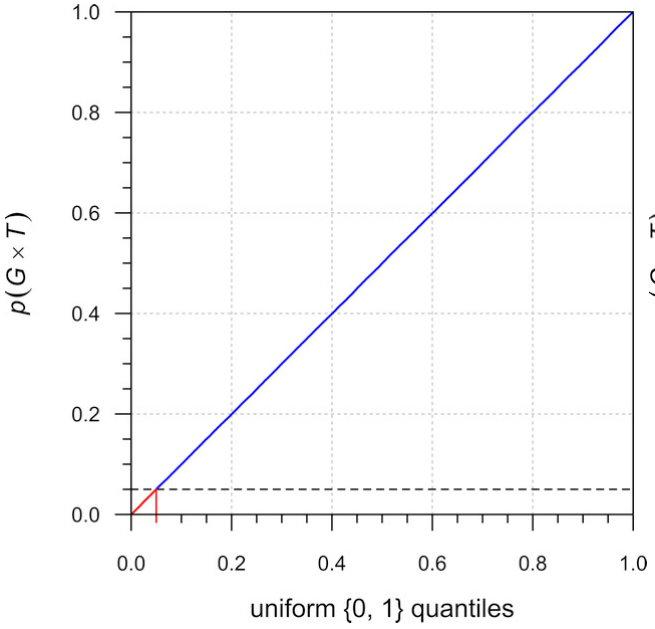

Fig. 3 Scenario 3: equal group sizes (n_1=n_2=24), heteroscedasticity (CV_w=36.9%, 29.8%), no true Group-by-Treatment interaction (GMR_1=GMR_2=1.0)
p(G×T)= 0.04994 (≤0.0511), p(unif.)= 0.9273
G×T correctly not detected in 95.1% of cases

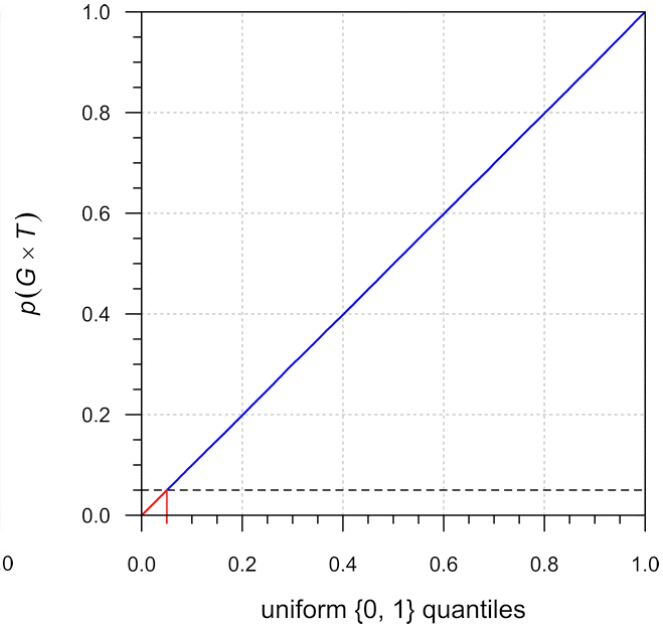

Fig. 4 Scenario 4: unequal group sizes (n_1=38, n_2=10), homoscedasticity (CV_w=33.5%), no true Group-by-Treatment interaction (GMR_1=GMR_2=1.0)
p(G×T)=0.05020 (≤0.0511), p(unif.)=0.5844
G×T correctly not detected in 95.0% of cases

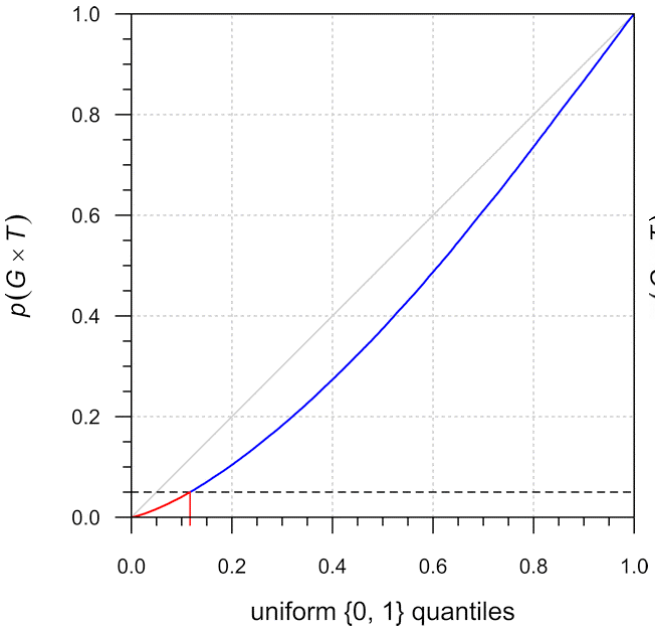

Fig. 5 Scenario 5: equal group sizes (n_1=n_2=24), homoscedasticity (CV_w=33.5%), true Group-by-Treatment interaction (GMR_1=0.95, GMR_2=1.056) p(G×T)=0.1166 (>0.0511), p(unif.)<$2.2 \cdot 10^{-16}$ G×T falsely not detected in 88.3% of cases

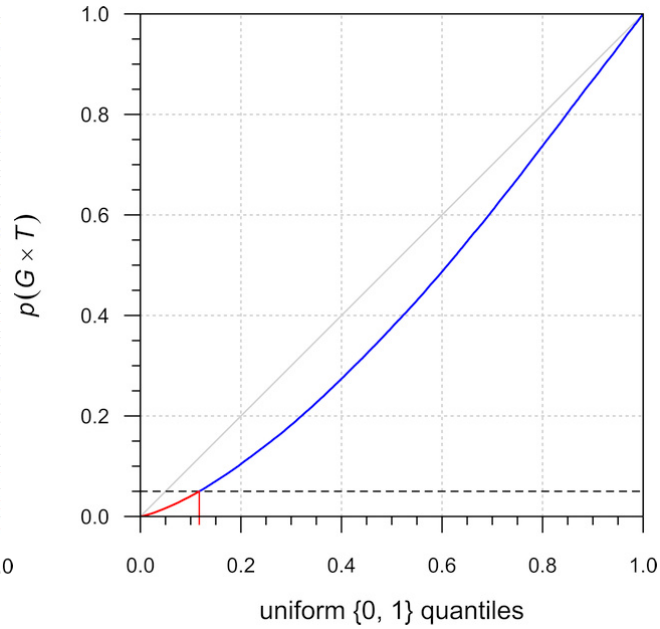

Fig. 6 Scenario 6: equal group sizes (n_1=n_2=24), heteroscedasticity (CV_w=29.8%, 36.9%), true Group-by-Treatment interaction (GMR_1=1.056, GMR_2=0.95) p(G×T)= 0.1170 (>0.0511), p(unif.)<$2.2 \cdot 10^{-16}$ G×T falsely not detected in 88.3% of cases

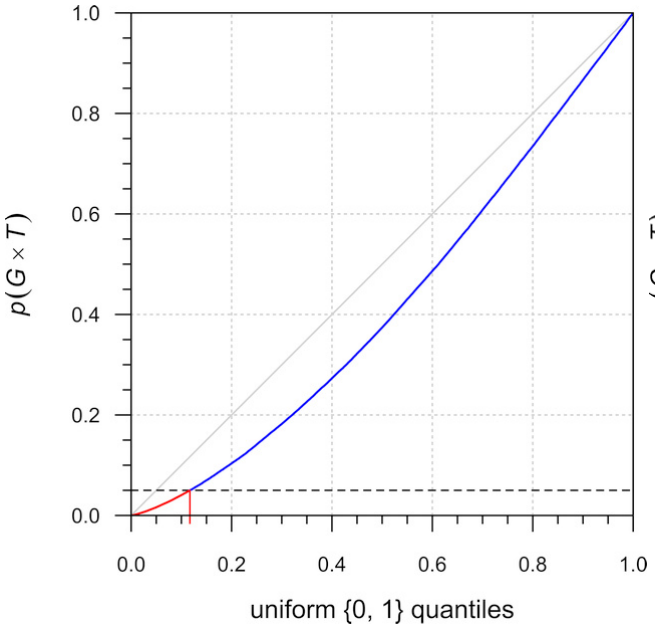

Fig. 7 Scenario 7: equal group sizes (n_1=n_2=24), heteroscedasticity (CV_w=36.9%, 29.8%), true Group-by-Treatment interaction (GMR_1=0.95, GMR_2=1.056) p(G×T)=0.1169 (>0.0511), p(unif.)<$2.2 \cdot 10^{-16}$ G×T falsely not detected in 88.3% of cases

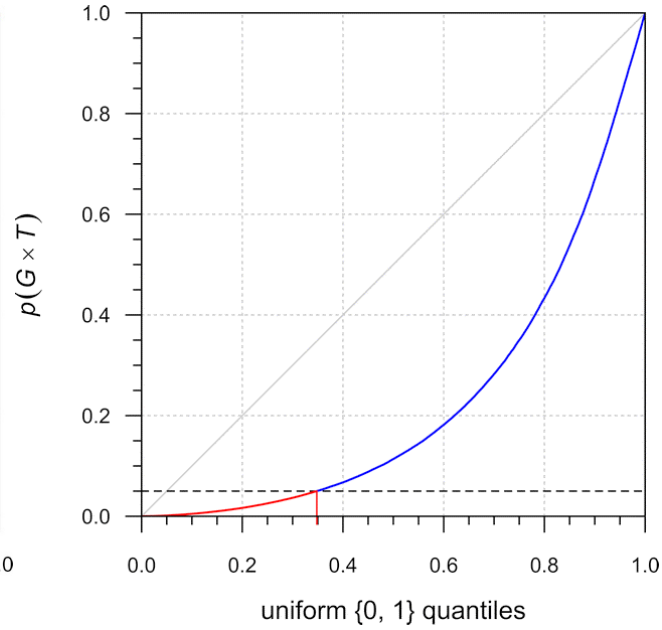

Fig. 8 Scenario 8: unequal group sizes (n_1=38, n_2=10), homoscedasticity (CV_w=33.5%), true Group-by-Treatment interaction (GMR_1=1.0605, GMR_2=0.80) p(G×T)=0.3480 (>0.0511), p(unif.)<$2.2 \cdot 10^{-16}$ G×T falsely not detected in 65.2% of cases

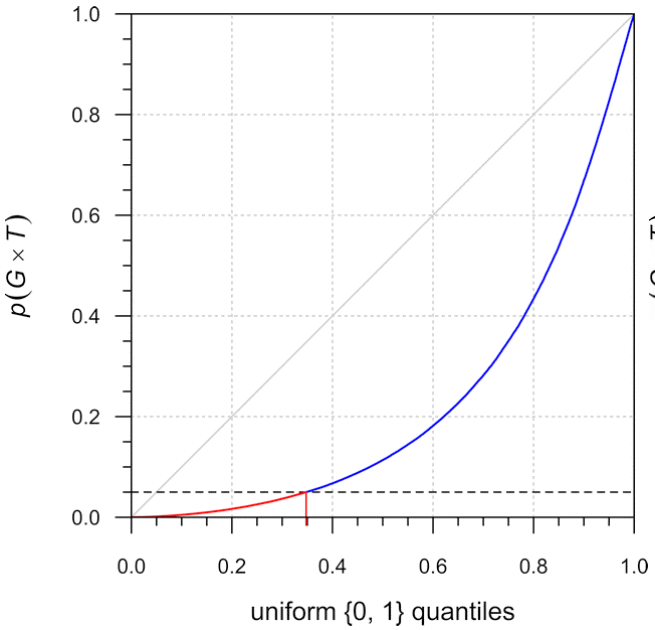

Fig. 9 Scenario 9: unequal group sizes (n_1=38, n_2=10), heteroscedasticity (CV_w=29.8%, 36.9%), true Group-by-Treatment interaction (GMR_1=1.0605, GMR_2=0.80) p(G×T)= 0.4020 (>0.0511), p(unif.)<$2.2 \cdot 10^{-16}$ G×T falsely not detected in 59.8% of cases

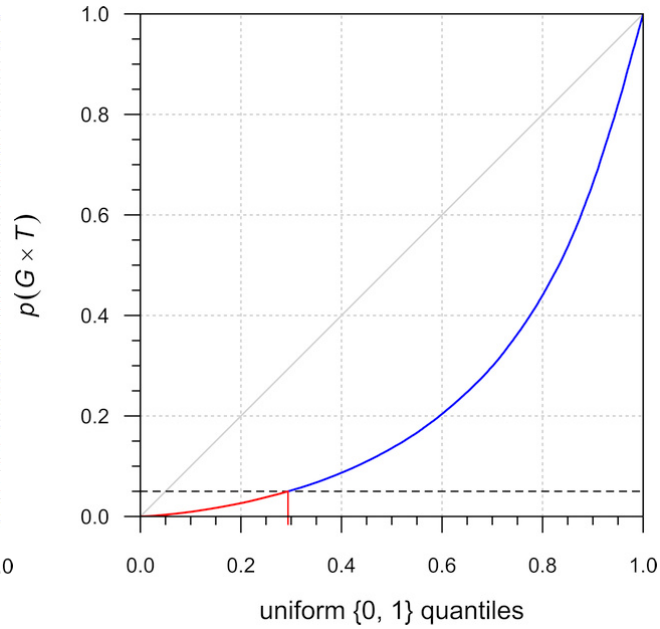

Fig. 10 Scenario 10: unequal group sizes (n_1=38, n_2=10), heteroscedasticity (CV_w=36.9%, 29.8%), true Group-by-Treatment interaction (GMR_1=1.0605, GMR_2=0.80) p(G×T)= 0.2938 (>0.0511), p(unif.)<$2.2 \cdot 10^{-16}$ G×T falsely not detected in 70.6% of cases

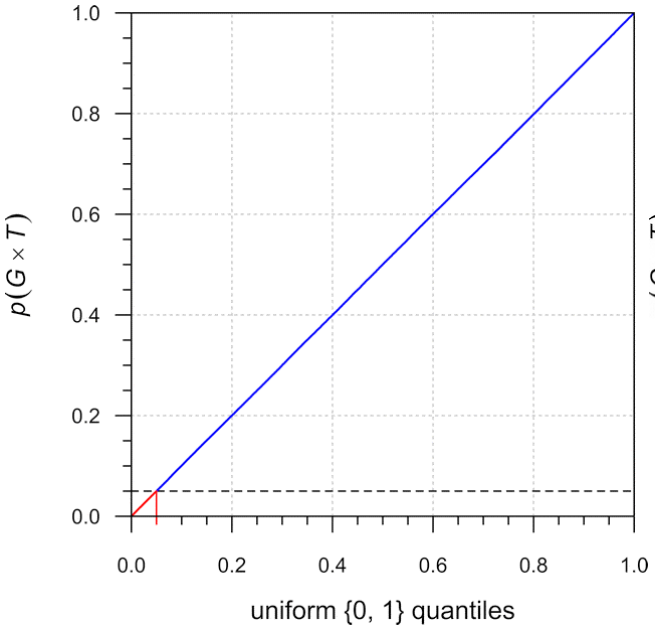

Fig. 11 Scenario 11: equal group sizes (n_1=n_2=40), homoscedasticity (CV_w=30.0%), no true Group-by-Treatment interaction (GMR_1=GMR_2=0.90) p(G×T)=0.04991 (≤0.0511), p(unif.)=0.7330 G×T correctly not detected in 95.0% of cases

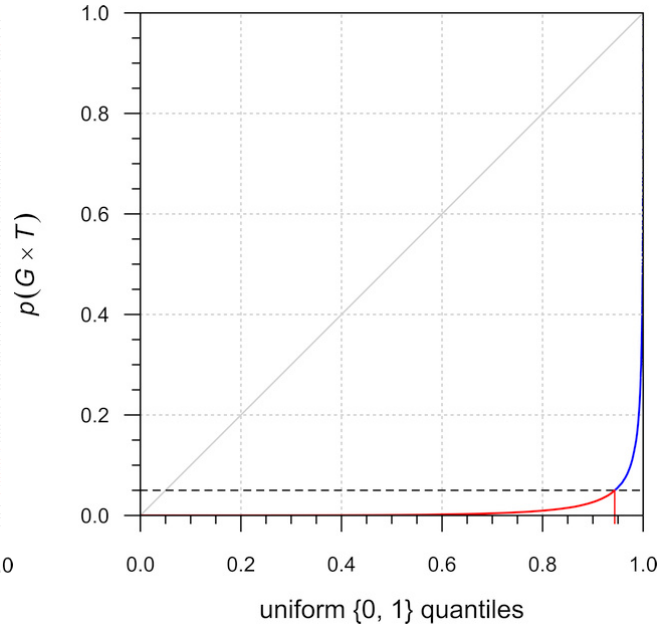

Fig. 12 Scenario 12: unequal group sizes (n_1=64, n_2=16), heteroscedasticity (CV_w=33.0%, 26.7%), true Group-by-Treatment interaction (GMR_1=0.8290, GMR_2=1.25) p(G×T)= 0.9435 (>0.0511), p(unif.)<$2.2 \cdot 10^{-16}$ G×T falsely not detected in 5.65% of cases

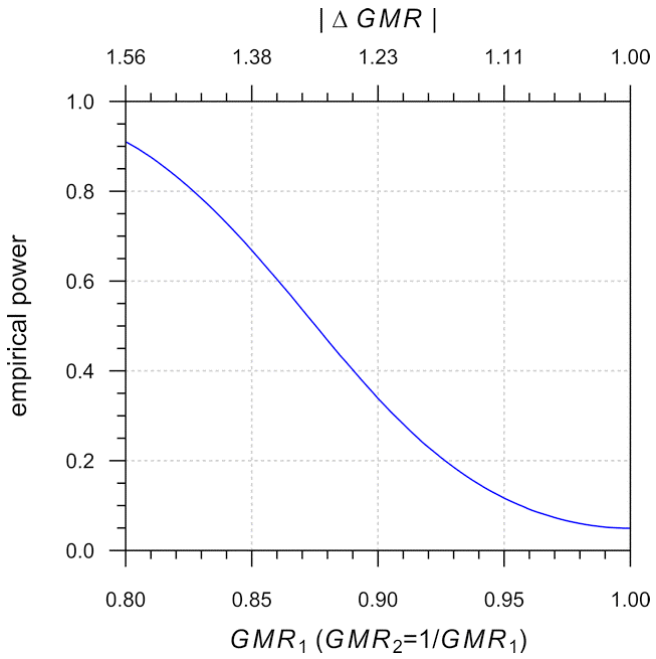

Fig. 13 Equal group sizes (n_1=n_2=24), homoscedasticity (CV_w=33.5%)

Meta-analysis

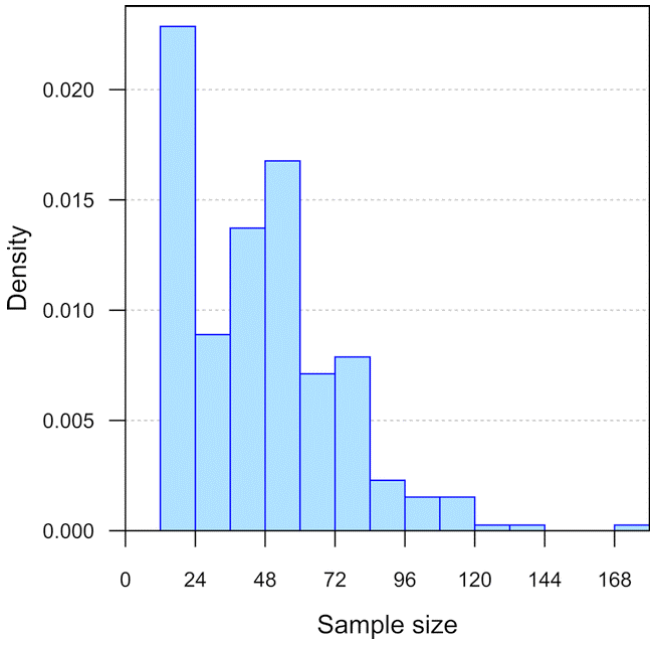

Fig. 14 *AUC* (*n*=328)

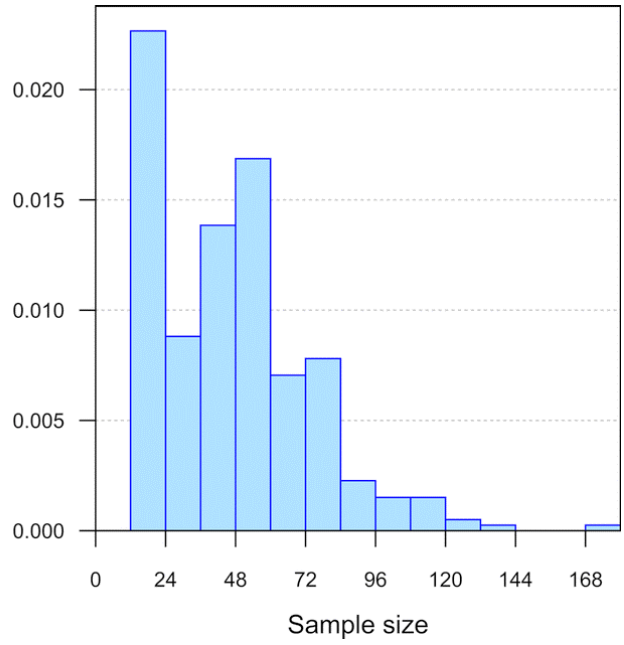

Fig. 15 C_{max} (*n*=331)

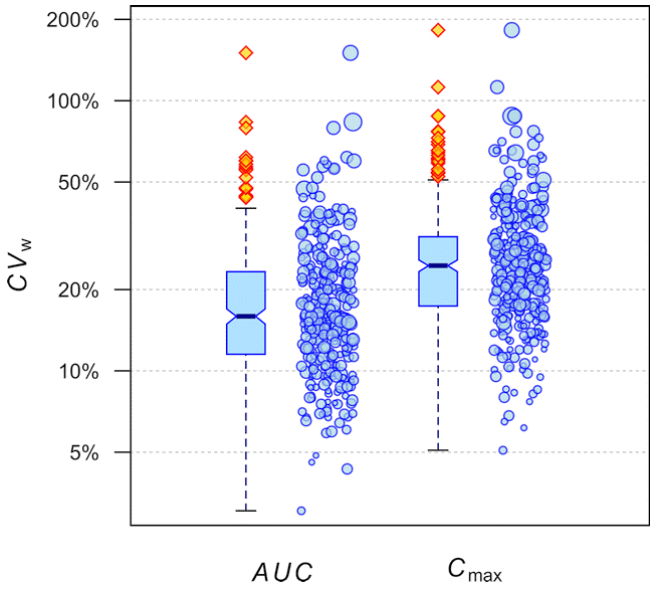

Fig. 16 Within-subject Coefficient of Variation.
Box plots: The lower edge of the box represents the 1st quartile, its the upper edge the 3rd quartile) and the line within the lower and the upper edges indicate the median. The distance from the lower edge to the upper edge of the box represents the interquartile range (IQR). Whiskers are drawn above the 1st quartile to the largest data value that is ≤ to the value that is 1.5 × IQR above the 3rd quartile and below the 1st quartile to the smallest data value that is ≤ to the value that is 1.5 × IQR below the 3rd quartile percentile. Any data value larger than that is marked as a moderate outlier (yellow diamonds).
Jitter plots: Study values are plotted as filled circles next to the boxes and shifted randomly preventing overlap; the size of circles indicate the sample size.
The median *CV* of *AUC* was 15.9% (3.03 – 150%) and the median *CV* of C_{max} was 24.5% (5.08 – 183%). As usual, variabilities of the one-point pharmacokinetic metric C_{max} were larger than the ones of the integrated PK metric *AUC*.

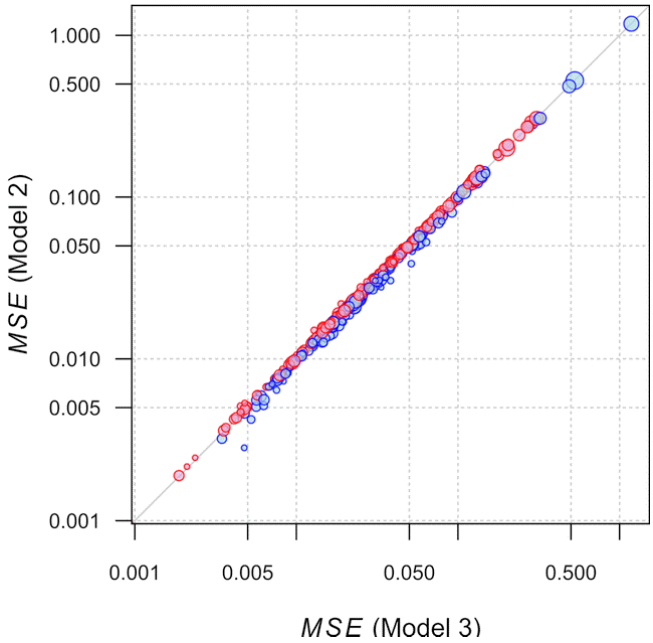

Fig. 17 Residual Mean Square Error by Models 3 (conventional) and 2 (with group terms); *AUC*.
blue circles if *MSE* (Model 2) ≤ *MSE* (Model 3)
red circles if *MSE* (Model 2) > *MSE* (Model 3)

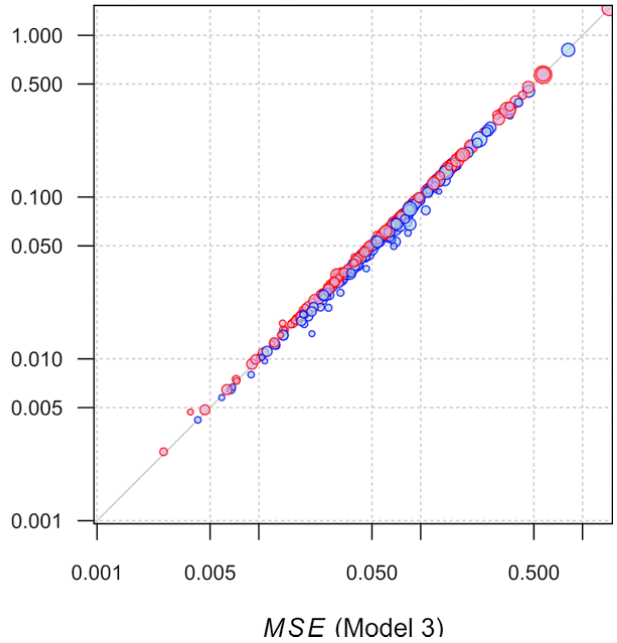

Fig. 18 Residual Mean Square Error by Models 3 (conventional) and 2 (with group terms); C_{max}.
blue circles if *MSE* (Model 2) ≤ *MSE* (Model 3)
red circles if *MSE* (Model 2) > *MSE* (Model 3)

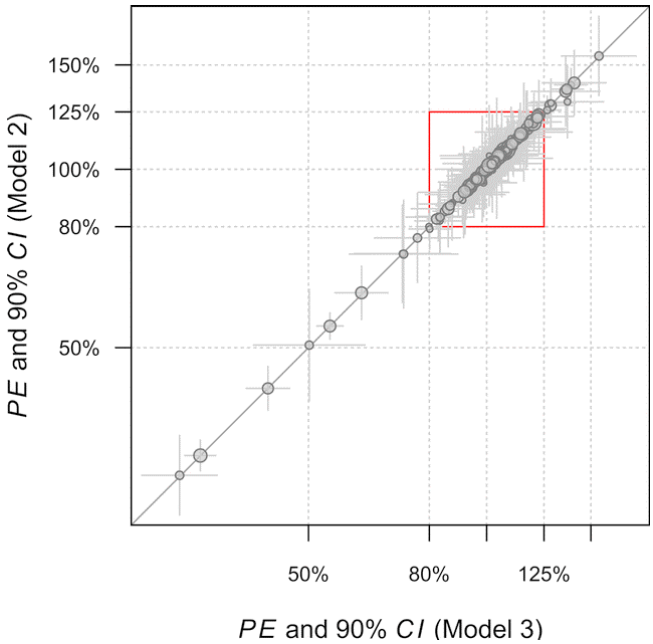

Fig. 19 Point Estimate and 90% Confidence Interval by Models 3 (conventional) and 2 (with group terms); *AUC*.
Red square conventional BE limits (80–125%).
Some studies food effect, drug-drug interaction, and dose proportionality. Hence, the extreme values.

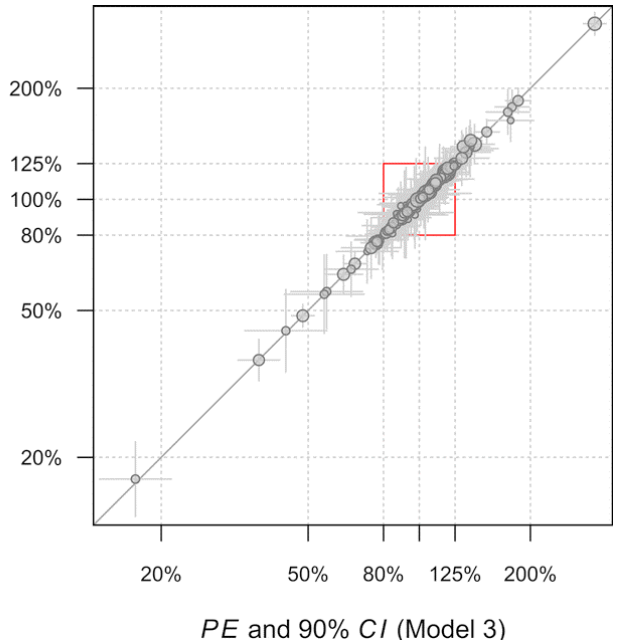

Fig. 20 Point Estimate and 90% Confidence Interval by Models 3 (conventional) and 2 (with group terms); C_{max}.
Red square conventional BE limits (80–125%).
Some studies food effect, drug-drug interaction, and dose proportionality. Hence, the extreme values.

Study with the largest interval of 62 days separating two groups (n_1=41, n_2=39)

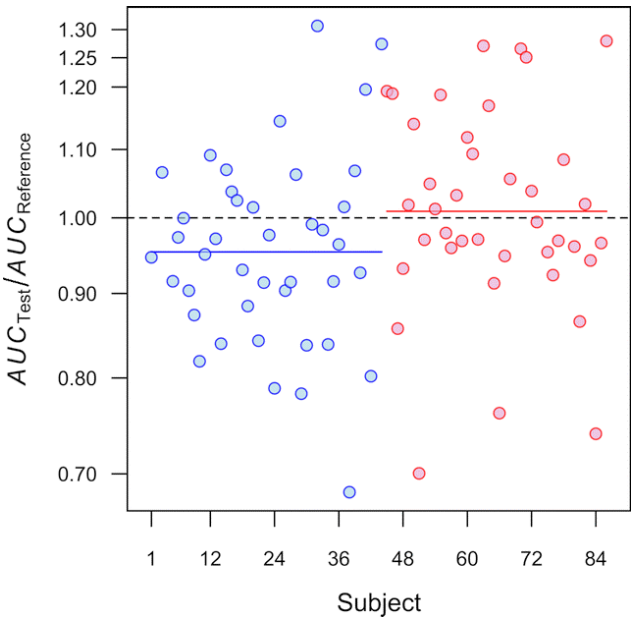

Fig. 21 Blue circles and horizontal lines group 1 (mean$_{geom}$=0.9537),
red circles and horizontal lines group 2 (mean$_{geom}$=1.0092).
Group 1: *PE*= 95.31% (90% CI 92.02– 98.72%)
Group 2: *PE*=101.13% (90% CI 97.35–105.06%)
Model 1: *p*(G×T)=0.05719, *p*(unif.)=0.1144
Model 2: *PE*= 98.10% (90% CI 95.59–100.69%)
Model 3: *PE*= 98.10% (90% CI 95.58–100.69%)

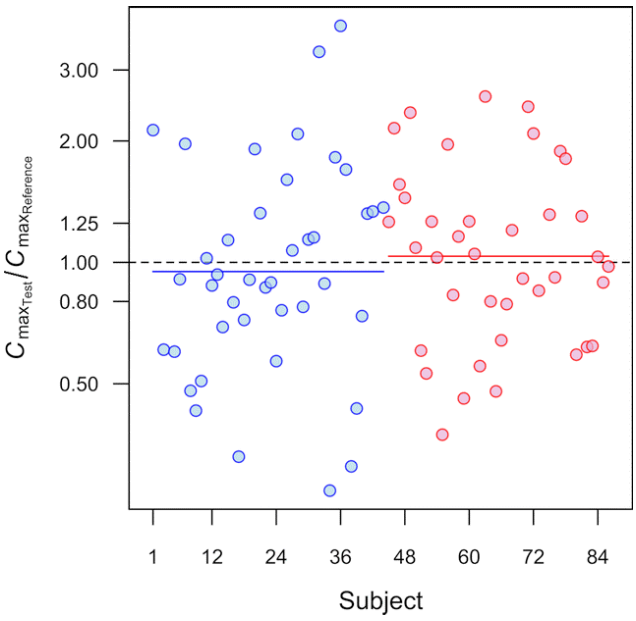

Fig. 22 Blue circles and horizontal lines group 1 (mean$_{geom}$=0.9486),
red circles and horizontal lines group 2 (mean$_{geom}$=1.0358).
Group 1: *PE*= 94.57% (90% CI 80.53–111.05%)
Group 2: *PE*=103.01% (90% CI 89.82–118.14%)
Model 1: *p*(G×T)=0.4990, *p*(unif.)=0.9979
Model 2: *PE*= 98.59% (90% CI 88.82–109.44%)
Model 3: *PE*= 98.59% (90% CI 88.88–109.37%)