

图 3-8 NCA 分析功能的设置示意图

### 3.2.3 基础模型选择界面

基础模型选择界面利用 `box` 函数设置了一个基本框，内部包含多个选择项和按钮，根据对数据的理解来选择模型，选择完之后点击由 `actionButton` 函数设置的“加载模型”按钮，系统自动会将用户的选择生成选项清单（某个具体模型的名称），直接从模型库中加载出来，点击“参数添加”按钮可以弹出添加初始参数的弹窗，添加完模型初始值之后，点击“保存”按钮，触发由 `showModel` 函数所生成的模态框将出现加载成功或失败的提示，保存成功的模型均放在 `PKBaseModelFolder` 文件夹中。

由于给药途径的不同，吸收方式和时间延迟也会不同（注：口服给药存在一级吸收或零级吸收的情况，静脉滴注和静脉推注只有零级吸收），通过 `conditionalPanel` 函数对吸收方式和时间延迟两个选择框设置了显示或隐藏功能，当给药途径选择口服给药时才会显示，否则选择隐藏这两个功能。另外对于 `BSV`，本文研究并测试了大量模型，发现指数型在建模过程中成功的概率最大且最为稳定，其它模型对于 `nlmixr2` 包来说不稳定，因此本文暂时只研究指数型，基础模型选择界面和具体流程如图 3-9，3-10 所示。

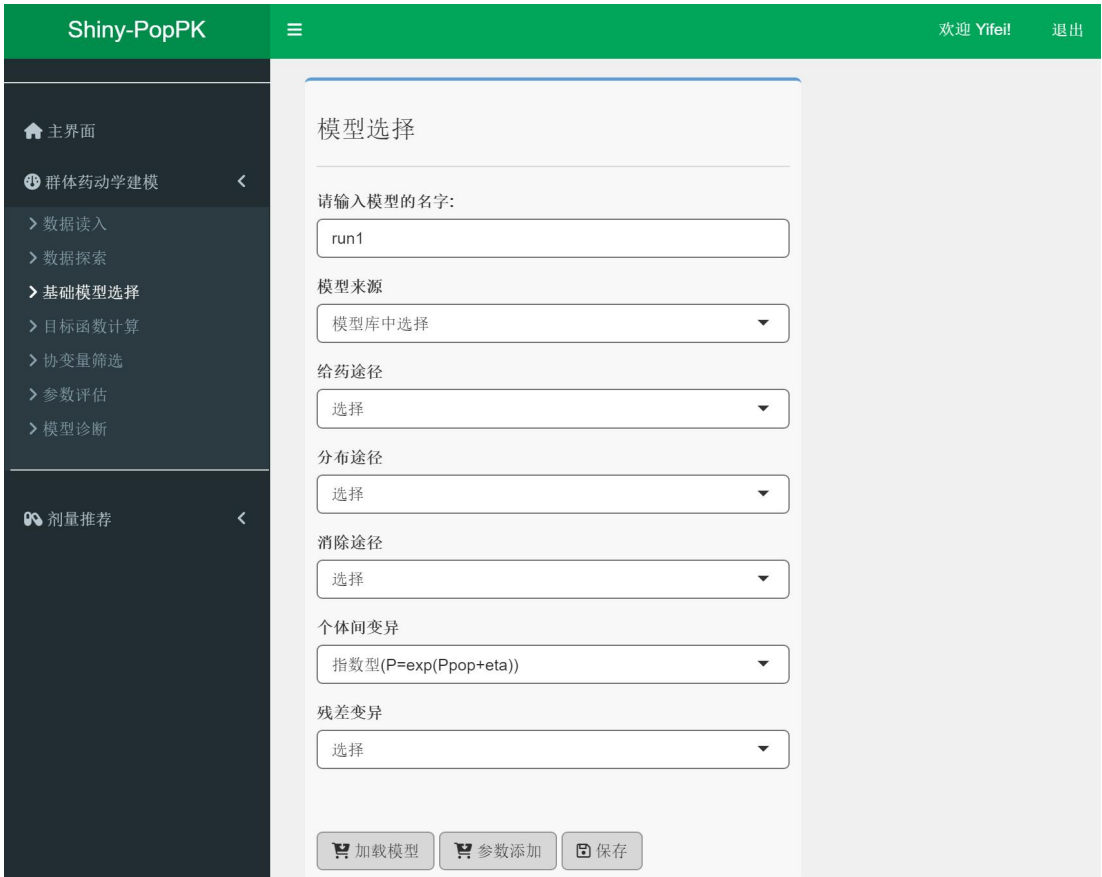


图 3-9 基础模型选择界面示意图

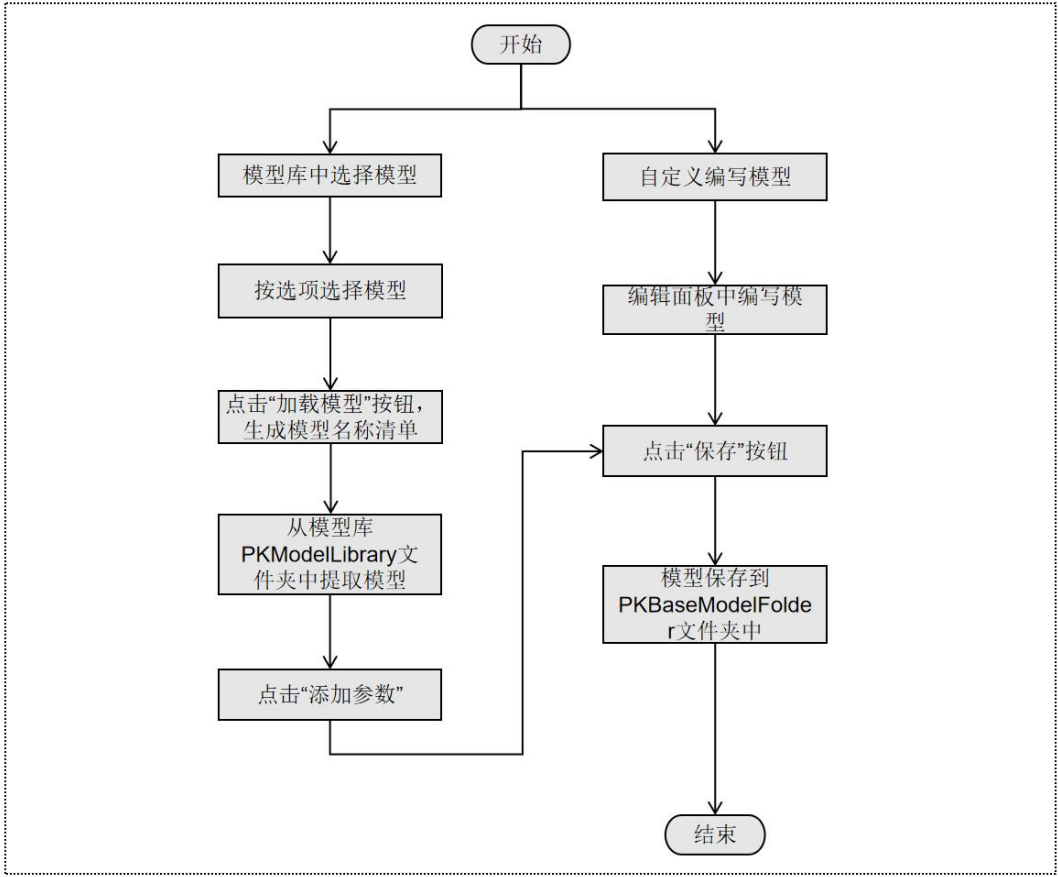


图 3-10 基础模型选择界面具体流程图

为了满足用户可以自定义编写模型的需求，可以从“模型来源”选择框中选择“自定义编写”选项，则能弹出代码编写面板，其所用的技术还是调用 conditionalPanel 函数对特定条件显示或隐藏，自定义编写模型如图 3-11 所示。

图 3-11 自定义编写模型示意图

本系统在开发过程中，参考 Monlix 软件内部搭建了模型库，其模型库中含有多多种模型，可供用户选择使用，本文在群体药动学理论基础之上严格按照 nlmixr2 程序包的基础模型的编写原则搭建了模型库。下面主要介绍模型库的搭建过程和模型的命名规则。

模型库是由基础模型所组成，基础模型的种类主要从房室模型、给药方式、吸收方式、消除方式、有无时间延迟、个体间变异和残差变异这几个方面考虑。

房室模型分为一房室、二房室、三房室及多房室模型，本文只研究一、二和三

房室；根据给药方式可分为口服给药，静脉滴注和静脉推注三种方式，此三种方式结合房室模型，其区别在于口服给药的药物是逐渐被人体吸收进入血液，这个吸收过程存在吸收速率  $ka$ ，而静脉滴注和推注则无吸收过程，它们是直接进入血液。吸收方式又可分为一级吸收和零级吸收，区别在于一级吸收速率与该部位的药量或者浓度的一次方成正比，而零级吸收速率是恒定的，与药物量或药物浓度无关。口服给药存在一级吸收或零级吸收的情况，静脉滴注和推注只有零级吸收。

对于消除方式分为线性消除和非线性消除，非线性消除会存在一个米氏常数；有无时间延迟的情况常出现在一些缓释药物当中，该药物在体内的 ADME 过程会出现延迟现象；个体间变异和残差变异在 1.1.1 小节已阐述，此处不在重复解释。考虑到对于 BSV，本文暂时只研究指数型，将这些条件组合起来将有 108 种模型，因模型种类较多，这里基于 `nlmixr2` 程序包以口服给药的一级吸收、线性消除、无时间延迟的一室模型为例编写模型，其代码见附录 1。

若存在延迟现象则在 `ini` 模块添加 `talag <- log(1.1)`，在 `model` 模块添加 `lagD <- exp(talag)`；若为非线性消除，需要将 `cl/v*cent` 改为 `(vm*center)/(km + center)`，`ini` 模块的参数也需要改变；残差变异若为加和型，需要将 `cp ~ prop(prop.err)` 改为 `cp ~ add(add.err)`；若是结合型，则改为 `cp ~ add(add.err) + prop(prop.err)`。需要说明的是 `tka`、`tcl` 和 `tv` 这三者开头带有 `t`，已经对原参数进行了取对数操作，若要得到原参数需要取指数操作。因篇幅有限，静脉滴注和推注在这里就不在阐述，部分模型已放在附录 2 中。将这 108 种模型放入到 `PKModelLibrary` 文件夹中，模型库就搭建完成。

本文将模型库搭建完成后，为了在系统中准确地加载某种模型，需要给每一种模型按照一定原则进行命名，给模型的命名格式为给药方式-房室-吸收方式-消除方式-有无时间延迟-个体间变异类型-残差变异类型，如口服给药-1 室-零级吸收-线性消除-无时间延迟-个体间变异指数型-残差变异加和型可以表示为 OAB-1CMT-ZOD-LIN-NTD-IEXP-OADD，每种模型格式的缩写如表 3-3 所示。

表 3-3 每种模型格式的缩写示意图

模型结构名称	模型结构命名
静脉注射	IVB
静脉滴注	IVI
口服给药	OAB

一室	1CMT
二室	2CMT
三室	3CMT
零级吸收	ZOD
一级吸收	FOD
线性消除	LIN
非线性消除	MIM
无时间延迟	NTD
有时间延迟	WTD
个体间变异指数型	IEXP
个体间变异加和型	IADD
个体间变异比例型	IPRO
残差变异加和型	OADD
残差变异比例型	OPRO
残差变异结合型	OCOM

### 3.2.4 目标函数计算界面

目标函数计算界面首先用 `fluidRow` 函数将整个界面分为上下界面两个部分，上部分界面包含选择建模的模型、数据和方法及结果，下部分界面主要显示整个运行模拟过程。在“已有模型”选择框内的模型均是通过调用 `get_model_list` 函数获取的 `PKBaseModelFolder` 文件夹中用户已加载的模型。“选择数据”选择框内的数据均是通过调用 `get_data_list` 函数获取的 `PKdata` 文件夹中用户已上传的数据，“估算方法”选择框内的算法包括 `saem`、`nlme`、`focei`、`foce`、`foi` 和 `fo`。

选择完上述选项后，点击“运行模型”按钮，系统的整个运行模拟过程将在下部分界面“运行过程”框中显示出来。运行结束后，系统将生成 `.res.rds` 文件和 `.ressum.rds` 文件保存在 `PKObjResultsFolder` 文件夹中。`.res.rds` 文件保存的是拟合后的 `fit` 对象，该对象需要用于协变量筛选，`.ressum.rds` 文件保存的是结算结果、模型名称、训练模型的数据名称及列名等一些信息。可通过点击“更新列表”按钮，激活 `overall_ofv` 函数读取 `PKObjResultsFolder` 文件夹中的 `.ressum.rds` 文件，获取 `ofv` 计算结果，系统还提供删除模型和下载拟合模型报告的功能，用户首先选中列表中的某个模型，才可以使用这两个功能。目标函数计算界面和具体流程如图 3-12，3-13

所示。

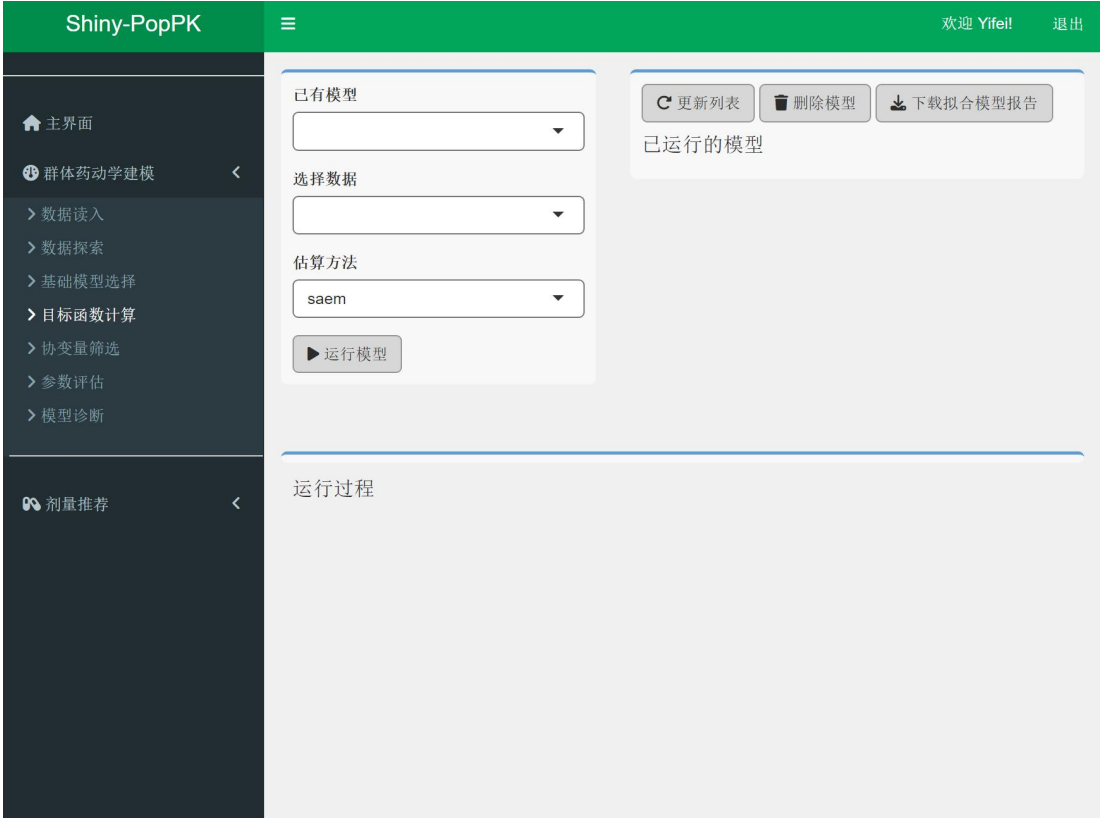


图 3-12 目标函数计算界面示意图

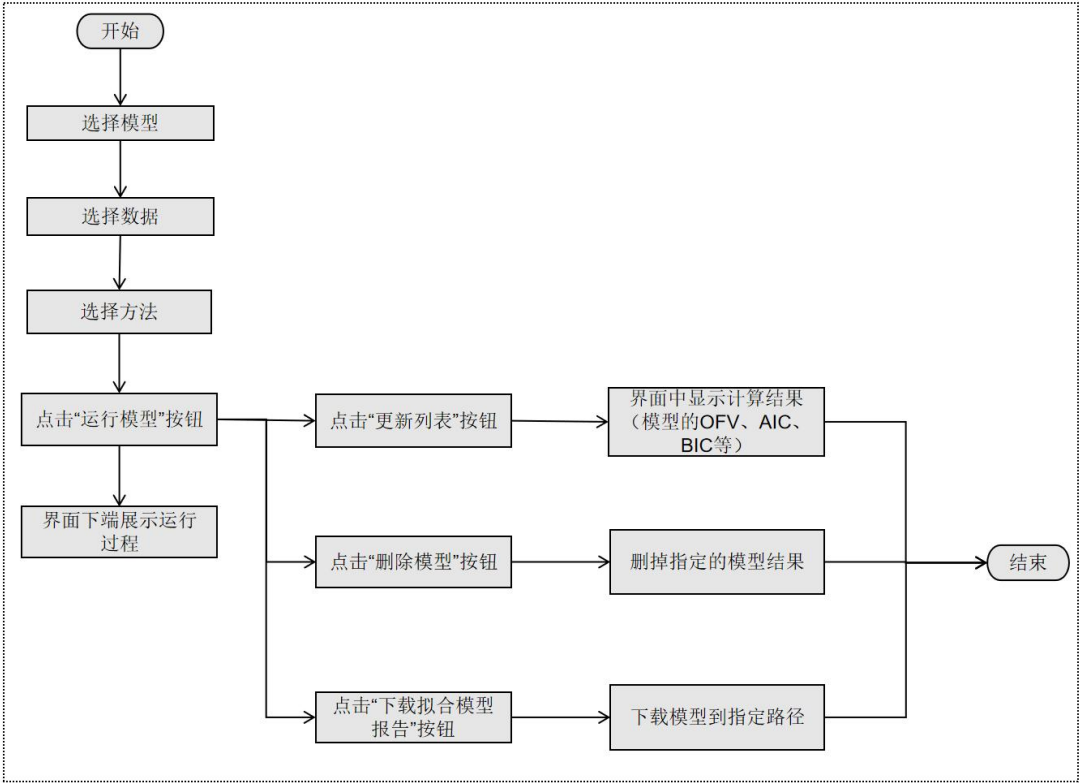


图 3-13 目标函数计算界面具体流程图

建模运行时，在 Rstudio 的 Console 控制台中呈现整个过程没有任何问题，但通

过 `render` 函数直接输入到 Ui 界面时，只会输出部分模拟数据。具体解决办法：将运行所需要的所有信息（加载的 R 包，建模的数据，调用的函数等）全部重新写入到一个脚本中，然后通过调用 `shell` 函数运行该脚本，就可通过 `txt` 格式的文本捕获到脚本运行过程所产生的数据信息，最后利用 `readLines` 函数读取该文本，再通过 `runpar_progrTxt` 函数将整个过程输入 Ui 界面中。

### 3.2.5 协变量筛选界面

协变量筛选界面的整体布局和目标函数计算界面的基本相似，都采用了 `fluidRow` 函数和 `column` 函数进行布局，分上下界面两个部分，上部分界面包含选择模型、确认模型、选择需要进行筛选的协变量和对应的参数、设置前向纳入和后向剔除的 P 值、启动筛选以及展示更新结果。下部分界面主要展示整个筛选过程，其运用的方法和目标函数计算界面使用的方法一样。

在“选择最佳残差模型”选择框内的模型通过调用 `get_model_fit_list` 函数读取 `PKObjResultsFolder` 文件夹中的 `.res.rds` 文件，该文件内部含有 `fit` 对象，可继续进行协变量的筛选过程。点击“确认模型”按钮后，激活 `ExtractCode` 函数和 `Extractparm` 函数分别从 `PKObjResultsFolder` 文件夹中的 `.ressum.rds` 文件中获取所选模型的所有参数和训练数据中的列名，获取的参数直接显示在 Ui 界面，而获取的数据列名还需要经过筛选删除掉非协变量的列名才可呈现在 Ui 界面中，筛选出的结果将在 `PKCovariatesFolder` 文件夹中以 `.cov.rds` 格式进行保存。点击“更新结果”按钮后，激活 `all_covresult` 函数从 `PKCovariatesFolder` 文件夹中读取 `.cov.rds` 文件获取结果，协变量筛选界面和具体流程如图 3-14，3-15 所示。

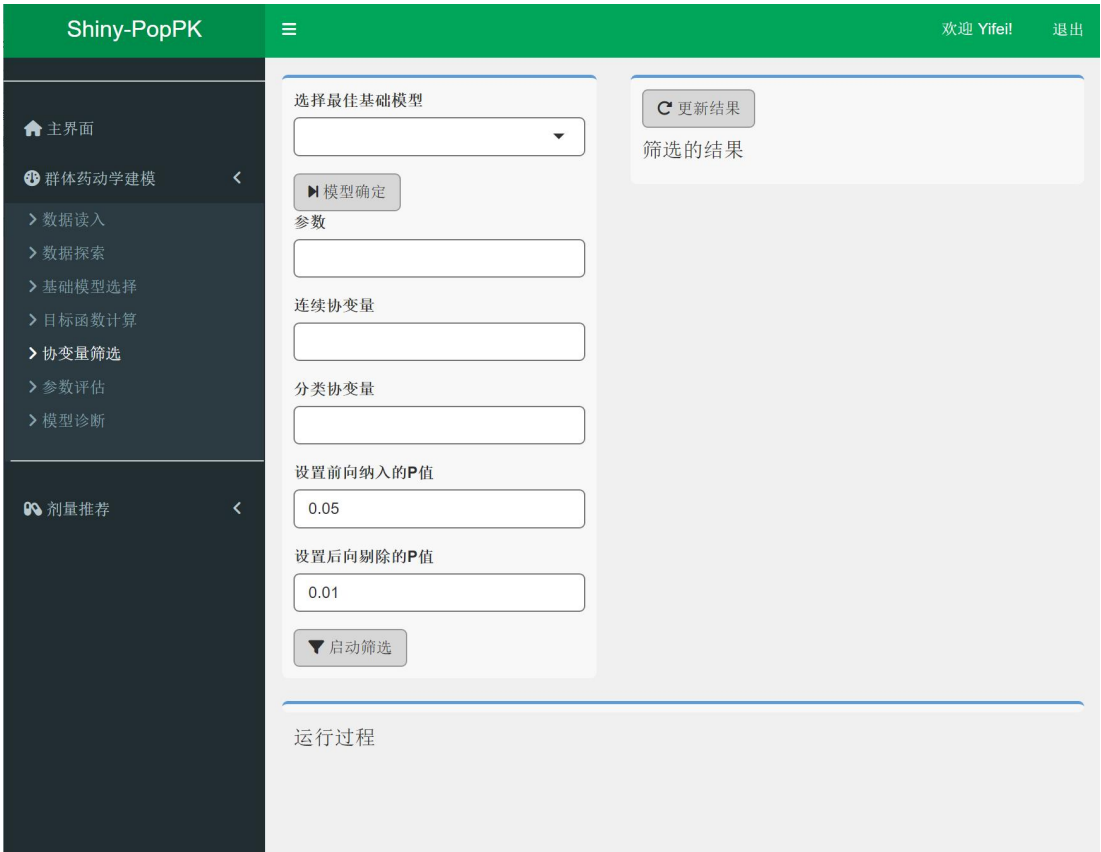
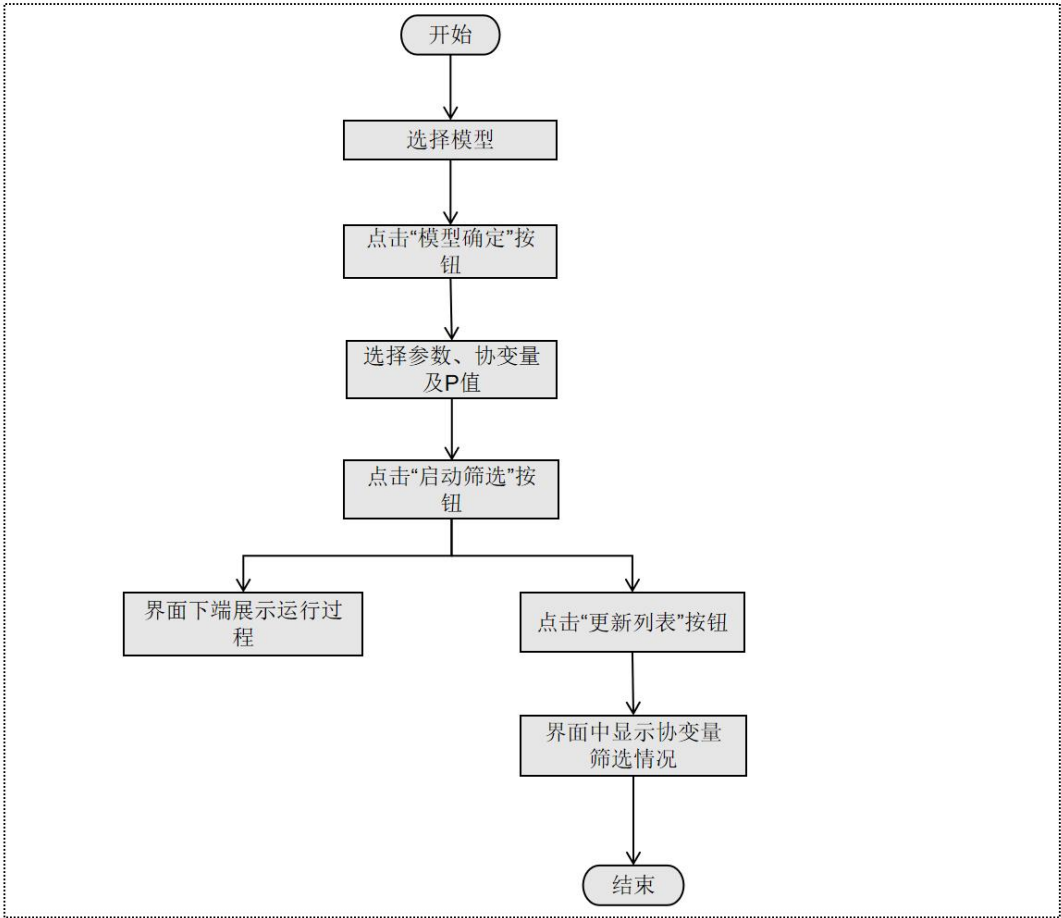


图 3-14 协变量筛选界面示意图





## 附录

### 附录 1 主要代码

(1) 主界面该界面的主要代码:

```

dashboardBody(shinyDashboardThemes(
  theme="grey_light" ),
  tabItems(
    tabItem(tabName = "homePage",
      br(),br(),
      h1("Shiny-PopPK      软      件      简      介      "),
      img(src="homepagepicture.jpg",.noWS="after-begin",height=200),withMathJax(),
      includeMarkdown("./www/homepageAbstract.Rmd"))
    userName <- Sys.info()[7]
    output$loggedUser <- renderText({
      return(paste0("欢迎 ", userName, "!"))})
    observeEvent(input$login, { stopApp()})

```

(2) 数据列名更换主要的代码:

```

reNaMe <- function(inpo){
  data1 <- read.csv(inpo$eventData$datapath)
  aid <- data1[inpo$selectID]
  atime <- data1[inpo$selectTIME]
  adv <- data1[inpo$selectDV]
  aamt <- data1[inpo$selectAMT]
  data2<- data1[,!(names(data1)%in%c(inpo$selectID,inpo$selectTIME,
    inpo$selectDV,inpo$selectAMT))]
  data2["ID"] <- aid
  data2["TIME"] <- atime
  data2["DV"] <- adv
  data2["AMT"] <- aamt
  file_path <-paste0("./PKdata/",inpo$eventData$name) # 指定文件路径
  write.csv(data2, file = file_path, row.names = FALSE) # 保存数据到 CSV 文件
}

```

(3) 药时曲线绘图和 NCA 计算的主要代码:

```

drugtime_plotOption <- function(inpt){
  xVar <- cstr(inpt$selectXAxis)
  yVar <- cstr(inpt$selectYAxis)
  groupBy <- cstr(inpt$selectGroup)
  xlabI <- ifelse(cstr(inpt$XlabI)=="",xVar, cstr(inpt$XlabI))
  ylabI <- ifelse(cstr(inpt$YlabI)=="",yVar, cstr(inpt$YlabI))
  clabI <- ifelse(cstr(inpt$ClabI)=="",groupBy, cstr(inpt$ClabI))
  conf<-ifelse(cnum(inpt$confidenceInterval)==0.95,0.95,
              cnum(inpt$confidenceInterval))
  alp <- ifelse(cnum(inpt$lineTransparency)==0.3,0.3,
              cnum(inpt$lineTransparency))
pkdata<-read.csv(normalizePath(get_data_list()[[inpt$selectDataGnPlot]]
                          $infoma,winslash = "/",mustWork = FALSE))
gg <- ggplot(data = pkdata, aes(x = get(xVar), y = get(yVar),
  group= factor(get(groupBy)),color = factor(get(groupBy))))
  gg <- gg + xgx_theme()
  gg <- gg + xgx_geom_ci(conf_level = conf, alpha = alp)
  gg <- gg + labs(x=xlabI ,y=ylabI,color=clabI)
  if(inpt$LogY=="Log"){
    gg <- gg + xgx_scale_y_log10()
  }
  if(!is.null(inpt$convertUnits) & !is.null(inpt$presentUnits)){
    gg <- gg + xgx_scale_x_time_units(units_dataset = inpt$convertUnits,
                                      units_plot = inpt$presentUnits)}
  if(inpt$singleGraph == "Yes"){
    if(inpt$ScalesO != "free_y"){
      gg <- gg + facet_wrap(~factor(get(groupBy)), scales = inpt$ScalesO)
    }else{
      gg <- gg + facet_wrap(~factor(get(groupBy)), scales = "free_y")}}
  print(gg)}

NCA_Calculate <- function(inpt){
  tdr <- read.csv(normalizePath(get_data_list()[[inpt$selectDataGnPlot]]

```

```

    $infoma,winslash = "/",mustWork = FALSE))
out_1 <- ncappc(obsFile=tdr,
  onlyNCA = TRUE,
  extrapolate = TRUE,
  concUnit = inpt$concUnit,
  timeUnit = inpt$timeUnit,
  doseUnit = inpt$doseUnit,
  adminType = inpt$adminType,
  printOut = FALSE,
  evid = FALSE,
  TI = inpt$IFDTIME,
  timing=TRUE)
return(out_1)}

```

(4) 参数添加时的主要代码:

```

combcode <- function(modelPath,value){
  mcode <- readLines(modelPath)
  ftmcode <- mcode[2:(which(mcode=="model({")-2)]
  bdmcode <- mcode[which(mcode=="model({"):length(mcode)]
  stmcode <- lapply(trimws(ftmcode),function(x){strsplit(x,"<|~")})
  namelt=list()
  for(i in 1:length(stmcode)){
    namelt[i] <- trimws(stmcode[[i]][[1]][1])
    Eqsym<-lapply(ftmcode,function(x){if(grepl("<",x)) return(which(ftmcode==x))})
    eqsym[sapply(eqsym, is.null)] <- NULL
    neqsym<-lapply(ftmcode,function(x){if(grepl("~",x)) return(which(ftmcode==x))})
    neqsym[sapply(neqsym, is.null)] <- NULL
    fcode <-lapply(eqsym,function(i){
      paste0(" ",(namelt)[i]," <- ",value[i])})
    rcode <-lapply(neqsym,function(i){
      paste0(" ",(namelt)[i]," ~ ",value[i])})
    acodelt <- c("ini({",fcode,rcode,"})",bdmcode)
    codele <- unlist(acodelt)
  }
}

```

```
return(codele)}
```

(5) 基于 `nlmixr2` 程序包以口服给药的一级吸收、线性消除、无时间延迟的一室模型为例编写模型，其主要代码：

```
model.1cpt.ode <- function(){#定义模型名称
  ini({                                #ini 模块
    tka <- 2 # log(ka)
    tcl <- 3 # log(CL)
    tv <- 5# log(V)
    eta.ka ~ 1 # IIV on ka
    eta.cl ~ 1 # IIV on CL
    eta.v ~ 1 # IIV on V
    prop.err <- 0.1 # Prop. RE})
  model({
    # 定义个体间变异为指数型
    ka <- exp(tka + eta.ka)
    cl <- exp(tcl + eta.cl)
    v <- exp(tv + eta.v)
    #定义口服给药一级吸收线性消除一室微分方程
    d/dt(depot) = -ka * depot
    d/dt(cent) = ka*depot - cl/v*cent
    cp = cent/v
    #定义残差变异为比例型
    cp ~ prop(prop.err)})}
```

(6) 目标函数计算界面的主要代码：

```
run_model<- function(mod,select_data,estimationMethod,get_data_list,
  saverds=TRUE,projloc=".",addcwres=TRUE,addnpde=TRUE){
  dir.create("./PKObjResultsFolder/temp",showWarnings = FALSE,recursive = TRUE)
  cntrl1 <- trimws(readLines(get_model_list()[[mod]]$model))
  tmp1 <- readLines("./run_nmx1.tmp")
  rlst <- list(modelloc=normalizePath(get_model_list()[[mod]]$model,
    winslash = "/",mustWork = FALSE),
```

```

datalog=normalizePath(get_data_list()[[select_data]]$infoma,
                      winslash = "/",mustWork = FALSE),est=estimationMethod,
                      setlj="./PKObjResultsFolder/temp",saveres=saverds, modelname=mod,
                      dataname=select_data,locproj=projloc,addcwres=addcwres,
                      addnpde=addnpde)

tscr<- paste0(projloc,"/PKObjResultsFolder/temp/script.",
              stringi::stri_rand_strings(1,6),".r")

writeLines(whisker::whisker.render(tmpl,rlst),tscr)
if(Sys.info()['sysname']=="Windows"){
  shell(paste0("Rscript \"", tscr, "\" > \"",projloc,"/PKObjResultsFolder/temp/",
              mod,".prog.txt\" 2>&1"),wait=FALSE)
} else {
  system(paste0("Rscript \"", tscr, "\" > \"",projloc,"/PKObjResultsFolder/temp/",
              mod,".prog.txt\" 2>&1"),wait=FALSE)}

```

(7) 协变量筛选界面的主要代码:

```

all_covresult <- function(){
  sumcovls<- normalizePath(list.files("./PKCovariatesFolder",pattern="*.cov\\.rds",
                                     full.names = TRUE))

  if(length(sumcovls)!=0){
    dafr <- data.frame()
    for (i in 1:(length(sumcovls))){
      sinresu <- readRDS(sumcovls[i])$summaryTable
      dafr <- rbind(dafr,sinresu)}
    return(dafr)}
  else{cat("警告:协变量文件夹为空，请先筛选协变量!")}}

```

(8) 协变量筛选界面的主要代码:

```

ParForCom <- function(mod,selectPar,covconame,Weightnum){
  fit <- readRDS(get_model_fit_list()[[mod]]$model)
  modelcode <- as.character(fit$funPrint[3])
  modelcodeSplit <- strsplit(modelcode,"\n")
  formula <- ""
  for (i in modelcodeSplit[[1]]){

```

```

if(grepl(paste0(selectPar," <-"),i)){
  formula <- i}}
rtc <- strsplit(trimws(formula),split = " ")
if(covconame == "WT"){
  covWeight <- paste0(covconame,'eff*(log(',covconame,'/70)) + ')}
else{
  covWeight <- paste0(covconame,'eff*(',covconame,') + ') }
stri_sub(rtc[[1]][length(rtc[[1]])],1,0) <- covWeight
FinalFormula <- ""
for (i in rtc[[1]]){
  FinalFormula <- paste0(FinalFormula,' ',i)}
FinalFormulaTw <- trimws(FinalFormula)
covWeightname <- paste0(covconame,'eff=',as.numeric(Weightnum))
capture.output(FinalFormulaTw, file = "/POPCacheFolder/formula.txt",append = TRUE)
capture.output(covWeightname,file="/POPCacheFolder/weightInitial.txt",append =TRUE)}

```

(9) 剂量优化界面的主要代码:

```

run_poso_dose_auc<-function(data,mod,StartingTime,timeAuc0,timeDose0,
  targetAuc0,doseInterval0,addDose0,duration0,Starting_dose0){
  data1 <- read.csv(get_data_list(FALSE)[[data]]$infoma)
  prior_model <- eval(parse(text=readLines(get_model_list(3)[[mod]]$model)))
  timeAuc <- ifelse(timeAuc0 == "NULL",NULL, cnum(timeAuc0))
  timeDose <- ifelse(timeDose0 == "NULL",NULL, cnum(timeDose0))
  targetAuc <- ifelse(targetAuc0 == "NULL",NULL, cnum(targetAuc0))
  doseInterval <- ifelse(doseInterval0 == "NULL",NULL, cnum(doseInterval0))
  addDose <- ifelse(addDose0 == "NULL",NULL, cnum(addDose0))
  duration <- ifelse(duration0 == "NULL",NULL, cnum(duration0))
  Starting_dose <- ifelse(Starting_dose0 == "NULL",NULL, cnum(Starting_dose0))
  zx <- poso_dose_auc(dat=data1,prior_model=prior_model,
    time_auc=timeAuc,
    time_dose = timeDose,
    target_auc=targetAuc,
    interdose_interval=doseInterval,

```

---

```

        add_dose=addDose,
        duration=duration,          #infused over 2 h
        starting_dose=Starting_dose)
    return(zx)}

run_poso_dose_conc <- function(data,mod,timeC0,timeDose0,targetConc0,
    doseInterval0,addDose0,duration0,Starting_dose0){
  data1 <- read.csv(get_data_list(FALSE)[[data]]$infoma)
  prior_model <- eval(parse(text=readLines(get_model_list(3)[[mod]]$model)))
  timeC <- ifelse(timeC0=="NULL",NULL, cnum(timeC0))
  timeDose <- ifelse(timeDose0=="NULL",NULL, cnum(timeDose0))
  targetConc <- ifelse(targetConc0=="NULL",NULL, cnum(targetConc0))
  doseInterval <- ifelse(doseInterval0=="NULL",NULL, cnum(doseInterval0))
  addDose <- ifelse(addDose0=="NULL",NULL, cnum(addDose0))
  duration <- ifelse(duration0=="NULL",NULL, cnum(duration0))
  Starting_dose <- ifelse(Starting_dose0=="NULL",NULL, cnum(Starting_dose0))
  zx <- poso_dose_conc(dat=data1,prior_model=prior_model,
    time_c=timeC,
    time_dose =timeDose,
    target_conc=targetConc,
    interdose_interval=doseInterval,
    add_dose=addDose,
    duration=duration,
    starting_dose=Starting_dose)
  return(zx)}

```

## 附录 2 部分常见的模型

(1) 口服给药、一室模型、一级吸收、线性消除、有时间延迟，个体间变异为指数型，残差变异为加和型 (OAB-1CMT-FOD-LIN-WTD-IEXP-OADD)

```
ini({
  talag <- log(1.1)
  tka <- log(1.5)
  tcl <- log(1.5)
  tv <- log(3)
  eta.ka ~ 1
  eta.cl ~ 1
  eta.v ~ 1
  add.err <- 0.1
})
model({
  lagD <- exp(talag)
  ka <- exp(tka + eta.ka)
  cl <- exp(tcl + eta.cl)
  v <- exp(tv + eta.v)
  d/dt(depot) = -ka * depot
  alag(depot) <- lagD
  d/dt(center) = ka * depot - cl / v * center
  cp = center / v
  cp ~ add(add.err)
})
```

(2) 口服给药、一室模型、一级吸收、非线性消除、有时间延迟，个体间变异为指数型，残差变异为加和型 (OAB-1CMT-FOD-MIM-WTD-IEXP-OADD)

```
ini({
  talag <- log(1.1)
  tka <- log(1.5)
  tvn <- log(0.121)
  tkm <- log(0.23)
  tv <- log(3)
  eta.ka ~ 1
  eta.vn ~ 1
  eta.km ~ 1
  eta.v ~ 1
  add.err <- 0.1
})
model({
  lagD <- exp(talag)
  ka <- exp(tka + eta.ka)
  vn <- exp(vn + eta.vn)
  km <- exp(tkm + eta.km)
  v <- exp(tv + eta.v)
  d/dt(depot) = -ka * depot
  alag(depot) <- lagD
  d/dt(center) = ka * depot - (vn*center)/(km + center)
  cp = center / v
  cp ~ add(add.err)
})
```



```
})
```

(3) 口服给药、二室模型、一级吸收、非线性消除、无时间延迟, 个体间变异为指数型, 残差变异为加和型 (OAB-2CMT-FOD-MIM-NTD-IEXP-OADD)

```
ini({
  tka <- log(1.05)
  tvn <- log(0.121)
  tkm <- log(0.23)
  tv2 <- log(1.939)
  tv3 <- log(5.65)
  tq  <- log(0.282)
  eta.ka ~ 0.1
  eta.vn ~ 0.1
  eta.km ~ 0.1
  eta.v2 ~ 0.1
  eta.v3 ~ 0.1
  eta.q ~ 0.1
  add.err <- 0.075
})
model({
  ka <- exp(tka + eta.ka)
  vn <- exp(tvn + eta.vn)
  km <- exp(tkm + eta.km)
  v2 <- exp(tv2 + eta.v2)
  v3 <- exp(tv3 + eta.v3)
  q  <- exp(tq + eta.q)
  d/dt(depot) = -ka * depot
  d/dt(center) = ka * depot - (vn*center)/(km + center) + q/v3 * periph - q/v2 * center
  d/dt(periph) = q/v2 * center - q/v3 * periph
  cp = center / v2
  cp ~ add(add.err)
})
```

(4) 口服给药、二室模型、零级吸收、非线性消除、无时间延迟, 个体间变异为指数型, 残差变异为加和型 (OAB-2CMT-ZOD-MIM-NTD-IEXP-OADD)

```
ini({
  tka <- log(1.05)
  tvn <- log(0.121)
  tkm <- log(0.23)
  tv2 <- log(1.939)
  tv3 <- log(5.65)
  tq  <- log(0.282)
  eta.ka ~ 0.1
  eta.vn ~ 0.1
  eta.km ~ 0.1
  eta.v2 ~ 0.1
  eta.v3 ~ 0.1
  eta.q ~ 0.1
  add.err <- 0.075
})
model({
  ka <- exp(tka + eta.ka)
```

```

vm <- exp(tvm + eta.vm)
km <- exp(tkm + eta.km)
v2 <- exp(tv2 + eta.v2)
v3 <- exp(tv3 + eta.v3)
q <- exp(tq + eta.q)
d/dt(depot) = -ka
d/dt(center) = ka - (vm*center)/(km + center) + q/v3 * periph - q/v2 * center
d/dt(periph) = q/v2 * center - q/v3 * periph
cp = center / v2
cp ~ add(add.err)
})

```

(5) 口服给药、三室模型、零级吸收、线性消除、无时间延迟，个体间变异为指数型，残差变异为比例型（OAB-3CMT-ZOD-LIN-NTD-IEXP-OPRO）

```

ini({
  tka <- log(1.42)
  tcl <- log(0.044)
  tv2 <- log(2)
  tv3 <- log(10)
  tv4 <- log(10)
  tq2 <- log(0.5)
  tq3 <- log(0.5)
  eta.ka ~ 0.1
  eta.cl ~ 0.1
  eta.v2 ~ 0.1
  eta.v3 ~ 0.1
  eta.v4 ~ 0.1
  eta.q2 ~ 0.1
  eta.q3 ~ 0.1
  prop.err <- 0.075
})
model({
  ka <- exp(tka + eta.ka)
  cl <- exp(tcl + eta.cl)
  v2 <- exp(tv2 + eta.v2)
  v3 <- exp(tv3 + eta.v3)
  v4 <- exp(tv4 + eta.v4)
  q2 <- exp(tq2 + eta.q2)
  q3 <- exp(tq3 + eta.q3)
  d/dt(depot) = -ka
  d/dt(center) = ka - cl / v2 * center + q2/v3 * periph1 - q2/v2 * center + q3/v4 * periph2 -
  q3/v2 * center
  d/dt(periph1) = q2/v2 * center - q2/v3 * periph1
  d/dt(periph2) = q3/v2 * center - q3/v4 * periph2
  cp = center / v2
  cp ~ prop(prop.err)
})

```

(6) 静脉注射、一室模型、零级吸收、线性消除、无时间延迟，个体间变异为指数型，残差变异为加和型（IVB-1CMT-ZOD-LIN-NTD-IEXP-OADD）

```

ini({
  tcl <- log(1.5)

```

```

tv <- log(3)
eta.cl ~ 1
eta.v ~ 1
add.err <- 0.1
})
model({
  cl <- exp(tcl + eta.cl)
  v <- exp(tv + eta.v)
  d/dt(center) = - cl / v * center
  cp = center / v
  cp ~ add(add.err)
})

```

(7) 静脉注射、一室模型、零级吸收、非线性消除、无时间延迟，个体间变异为指数型，残差变异为加和型 (IVB-1CMT-ZOD-MIM-NTD-IEXP-OADD)

```

ini({
  tvn <- log(0.121)
  tkn <- log(0.23)
  tv <- log(3)
  eta.vn ~ 0.1
  eta.kn ~ 0.1
  eta.v ~ 1
  add.err <- 0.1
})
model({
  vn <- exp(tvn + eta.vn)
  kn <- exp(tkn + eta.kn)
  v <- exp(tv + eta.v)
  d/dt(center) = - (vn*center)/(kn + center)
  cp = center / v
  cp ~ add(add.err)
})

```

(8) 静脉注射、二室模型、零级吸收、非线性消除、无时间延迟，个体间变异为指数型，残差变异为比例型 (IVB-2CMT-ZOD-MIM-NTD-IEXP-OPRO)

```

ini({
  tvn <- log(0.121)
  tkn <- log(0.23)
  tv2 <- log(1.939)
  tv3 <- log(5.65)
  tq <- log(0.282)
  eta.vn ~ 0.1
  eta.kn ~ 0.1
  eta.v2 ~ 0.1
  eta.v3 ~ 0.1
  eta.q ~ 0.1
  prop.err <- 0.075
})
model({
  vn <- exp(tvn + eta.vn)
  kn <- exp(tkn + eta.kn)
  v2 <- exp(tv2 + eta.v2)

```

```

v3 <- exp(tv3 + eta.v3)
q <- exp(tq + eta.q)
d/dt(center) = - (vm*center)/(km + center) + q/v3 * periph - q/v2 * center
d/dt(periph) = q/v2 * center - q/v3 * periph
cp = center / v2
cp ~ prop(prop.err)
})

```

(9) 静脉注射、二室模型、零级吸收、线性消除、无时间延迟，个体间变异为指数型，残差变异为比例型 (IVB-2CMT-ZOD-LIN-NTD-IEXP-OPRO)

```

ini({
  tcl <- log(0.121)
  tv2 <- log(1.939)
  tv3 <- log(5.65)
  tq <- log(0.282)
  eta.cl ~ 0.1
  eta.v2 ~ 0.1
  eta.v3 ~ 0.1
  eta.q ~ 0.1
  prop.err <- 0.075
})
model({
  cl <- exp(tcl + eta.cl)
  v2 <- exp(tv2 + eta.v2)
  v3 <- exp(tv3 + eta.v3)
  q <- exp(tq + eta.q)
  d/dt(center) = - cl / v2 * center + q/v3 * periph - q/v2 * center
  d/dt(periph) = q/v2 * center - q/v3 * periph
  cp = center / v2
  cp ~ prop(prop.err)
})

```

(10) 静脉注射、三室模型、零级吸收、线性消除、无时间延迟，个体间变异为指数型，残差变异为结合型 (IVB-3CMT-ZOD-LIN-NTD-IEXP-OCOM)

```

ini({
  tcl <- log(0.044)
  tv2 <- log(2)
  tv3 <- log(10)
  tv4 <- log(10)
  tq2 <- log(0.5)
  tq3 <- log(0.5)
  eta.cl ~ 0.1
  eta.v2 ~ 0.1
  eta.v3 ~ 0.1
  eta.v4 ~ 0.1
  eta.q2 ~ 0.1
  eta.q3 ~ 0.1
  add.err <- 0.075
  prop.err <- 0.075
})
model({
  cl <- exp(tcl + eta.cl)

```

```

v2 <- exp(tv2 + eta.v2)
v3 <- exp(tv3 + eta.v3)
v4 <- exp(tv4 + eta.v4)
q2 <- exp(tq2 + eta.q2)
q3 <- exp(tq3 + eta.q3)
d/dt(center) = - cl / v2 * center + q2/v3 * periph1 - q2/v2 * center + q3/v4 * periph2 - q3/v2
* center
d/dt(periph1) = q2/v2 * center - q2/v3 * periph1
d/dt(periph2) = q3/v2 * center - q3/v4 * periph2
cp = center / v2
cp ~ add(add.err) + prop(prop.err)
})

```

(11) 静脉滴注、一室模型、零级吸收、线性消除、无时间延迟，个体间变异为指数型，残差变异为加和型 (IVI-1CMT-ZOD-LIN-NTD-IEXP-OADD)

```

ini({
  tcl <- log(1.5)
  tv <- log(3)
  eta.cl ~ 1
  eta.v ~ 1
  add.err <- 0.1
})
model({
  cl <- exp(tcl + eta.cl)
  v <- exp(tv + eta.v)
  k0 <- rate
  d/dt(center) = k0 - cl / v * center
  cp = center / v
  cp ~ add(add.err)
})

```

(12) 静脉滴注、一室模型、零级吸收、线性消除、无时间延迟，个体间变异为指数型，残差变异为比例型 (IVI-1CMT-ZOD-LIN-NTD-IEXP-OPRO)

```

ini({
  tcl <- log(1.5)
  tv <- log(3)
  eta.cl ~ 1
  eta.v ~ 1
  prop.err <- 0.1
})
model({
  cl <- exp(tcl + eta.cl)
  v <- exp(tv + eta.v)
  k0 <- rate
  d/dt(center) = k0 - cl / v * center
  cp = center / v
  cp ~ prop(prop.err)
})

```

(13) 静脉滴注、二室模型、零级吸收、线性消除、无时间延迟，个体间变异为指数型，残差变异为比例型 (IVI-2CMT-ZOD-LIN-NTD-IEXP-OPRO)

```

ini({

```

```

tcl <- log(0.121)
tv2 <- log(1.939)
tv3 <- log(5.65)
tq <- log(0.282)
eta.cl ~ 0.1
eta.v2 ~ 0.1
eta.v3 ~ 0.1
eta.q ~ 0.1
prop.err <- 0.075
})
model({
  cl <- exp(tcl + eta.cl)
  v2 <- exp(tv2 + eta.v2)
  v3 <- exp(tv3 + eta.v3)
  q <- exp(tq + eta.q)
  k0 <- rate
  d/dt(center) = k0 - cl / v2 * center + q/v3 * periph - q/v2 * center
  d/dt(periph) = q/v2 * center - q/v3 * periph
  cp = center / v2
  cp ~ prop(prop.err)
})

```

(14) 静脉滴注、二室模型、零级吸收、非线性消除、无时间延迟，个体间变异为指数型，残差变异为结合型 (IVI-2CMT-ZOD-MIM-NTD-IEXP-OCOM)

```

ini({
  tvn <- log(0.121)
  tkn <- log(0.23)
  tv2 <- log(1.939)
  tv3 <- log(5.65)
  tq <- log(0.282)
  eta.vn ~ 0.1
  eta.kn ~ 0.1
  eta.v2 ~ 0.1
  eta.v3 ~ 0.1
  eta.q ~ 0.1
  add.err <- 0.1
  prop.err <- 0.075
})
model({
  vn <- exp(tvn + eta.vn)
  kn <- exp(tkn + eta.kn)
  v2 <- exp(tv2 + eta.v2)
  v3 <- exp(tv3 + eta.v3)
  q <- exp(tq + eta.q)
  k0 <- rate
  d/dt(center) = k0 - (vn*center)/(kn + center) + q/v3 * periph - q/v2 * center
  d/dt(periph) = q/v2 * center - q/v3 * periph
  cp = center / v2
  cp ~ add(add.err) + prop(prop.err)
})

```

(15) 静脉滴注、三室模型、零级吸收、线性消除、无时间延迟，个体间变异为指数

型，残差变异为结合型（IVI-3CMT-ZOD-LIN-NTD-IEXP-OCOM）

```

ini({
  tcl <- log(0.044)
  tv2 <- log(2)
  tv3 <- log(10)
  tv4 <- log(10)
  tq2 <- log(0.5)
  tq3 <- log(0.5)
  eta.cl ~ 0.1
  eta.v2 ~ 0.1
  eta.v3 ~ 0.1
  eta.v4 ~ 0.1
  eta.q2 ~ 0.1
  eta.q3 ~ 0.1
  add.err <- 0.075
  prop.err <- 0.075
})
model({
  cl <- exp(tcl + eta.cl)
  v2 <- exp(tv2 + eta.v2)
  v3 <- exp(tv3 + eta.v3)
  v4 <- exp(tv4 + eta.v4)
  q2 <- exp(tq2 + eta.q2)
  q3 <- exp(tq3 + eta.q3)
  k0 <- rate
  d/dt(center) = k0 - cl / v2 * center + q2/v3 * periph1 - q2/v2 * center + q3/v4 * periph2 -
q3/v2 * center
  d/dt(periph1) = q2/v2 * center - q2/v3 * periph1
  d/dt(periph2) = q3/v2 * center - q3/v4 * periph2
  cp = center / v2
  cp ~ add(add.err) + prop(prop.err)
})

```