# 第三章系统开发与功能实现

## 主界面

系统启动后，为确保系统的安全性和稳定性，提高使用体验为用户创建了登录界面，登录界面使用 textInput 函数创建用户名输入框，通过 passwordInput 函数和actionButton 函数分别创建了密码输入框和登陆按钮。进入系统后，系统的整体布局方式采用 shinydashboard 程序包的仪表盘风格，分为三部分：标题栏、侧边栏和主体部分。标题栏部分除了“Shiny-PopPK”标题外，还设置了“退出”按钮，通过调用Sys.info 函数获取本机名称，设置了“欢迎（本机名称）！”的字样。

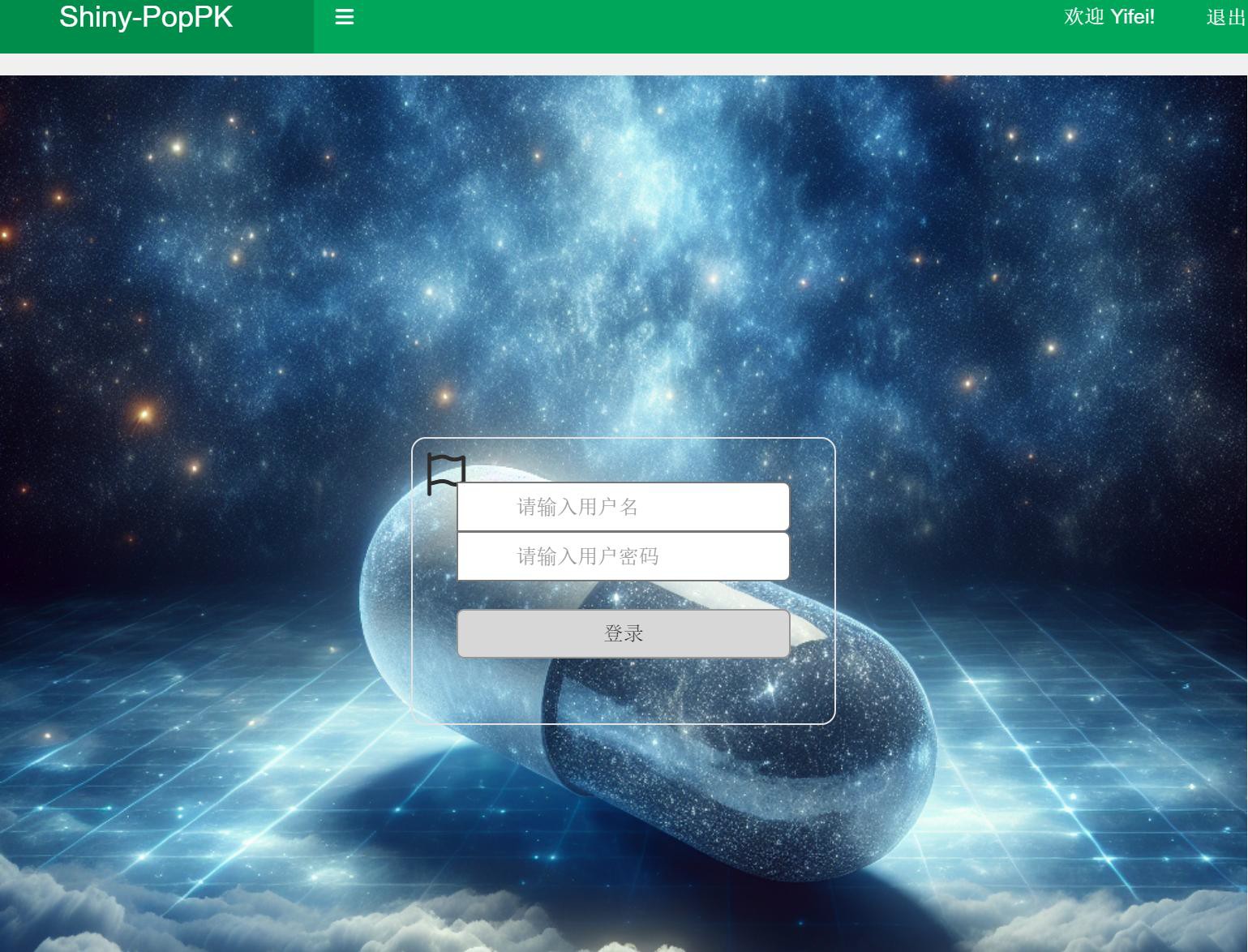
侧边栏通过 menuItem 函数设置了三种主要功能，分别为主界面、群体药动学建模和剂量推荐。又通过 menuSubItem 函数在主要功能下设置了子功能。主界面的主体部分通过 actionButton 函数设置了“Help”用户指南按钮，可方便非专业人士使用，调用 includeMarkdown 函数读取 www 文件夹中的 homepageAbstract.Rmd 文件来简要介绍该系统。又通过调用 includeCSS 函数加载 style.css 文件对界面及控件进行CSS 美化设计,使界面布局更合理。登录界面和主界面分别如图 3-1 和 3-2 所示。

图 3-1 登录界面示意图



## 群体药动学建模界面

图 3-2 主界面示意图

群体药动学建模界面包含的几大功能主要有数据读入、数据探索、基础模型选择、目标函数计算、协变量筛选、参数评估和模型诊断。

* + 1. 数据读入界面

数据读入界面主要通过 fileInput 函数设置上传文件的控件按钮，可以选择本地文件完成上传操作，然后用 selectizeInput 函数设置了四个选项框，目的是将数据中最重要的列名（ID、TIME、DV、AMT）进行更换。最后用 actionButton 函数设置了“确定上传”按钮。为避免用户对数据结构不转换导致建模失败，本系统通过 If 语句强制用户只有在完成数据列名转换的步骤后，才可激活“确定上传”按钮，所有上传的数据均保存在 PKdata 文件夹中。数据读入界面及具体流程如图 3-3，3-4 所示。主要的代码已放在附录 1 中。

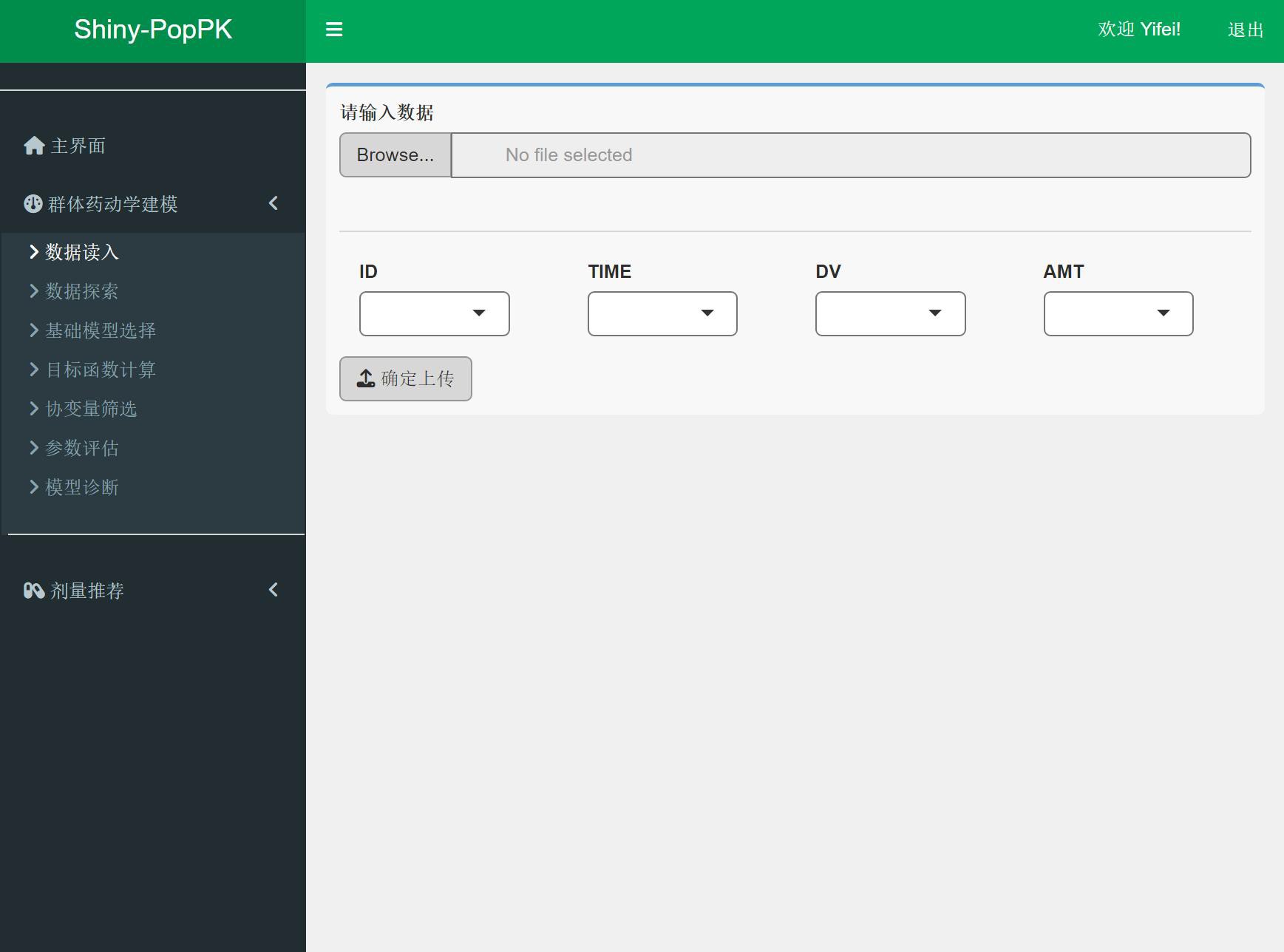


图 3-3 数据读入界面示意图

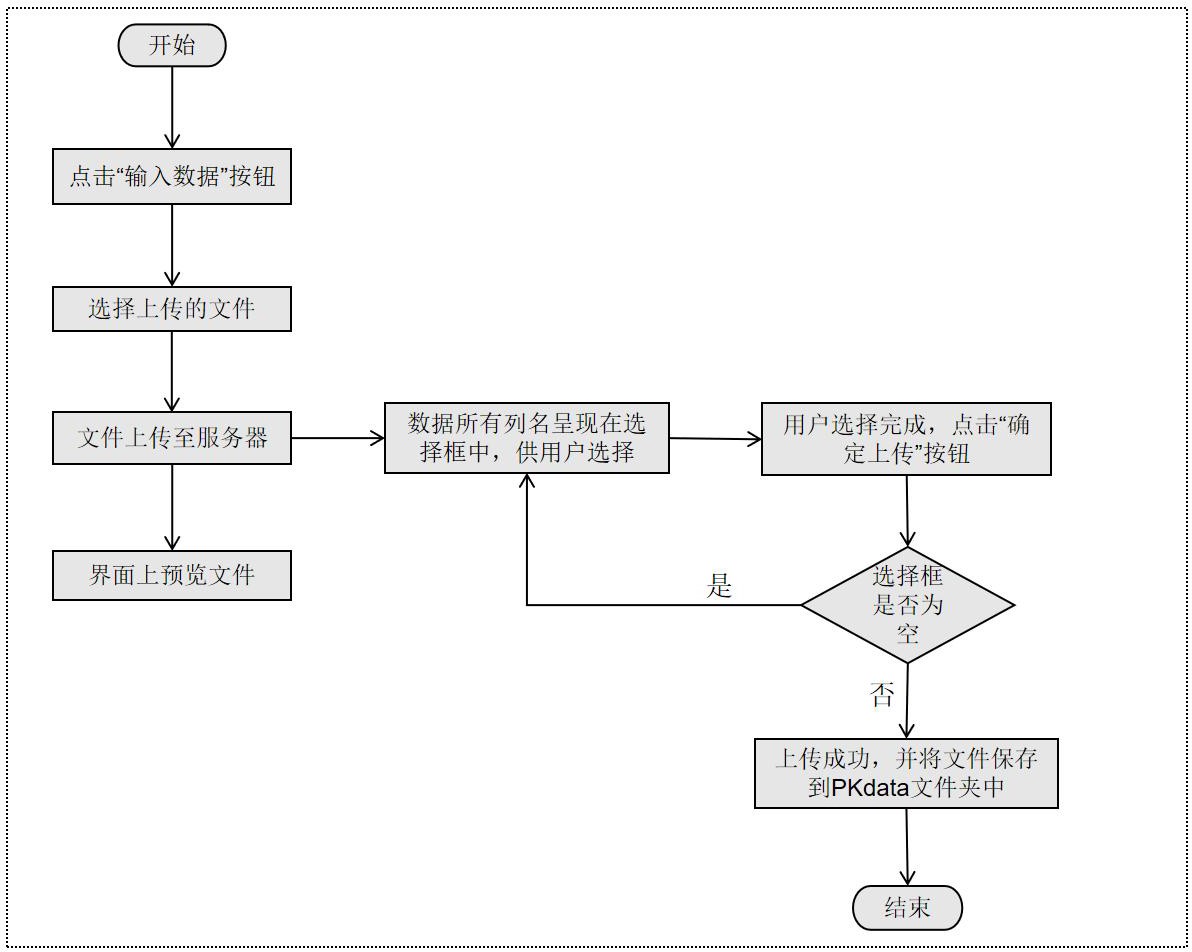


图 3-4 数据读入界面具体流程图

建模部分应用是 nlmixr2 包，该包对数据结构具有严格的要求，若建模数据存在不规范，系统就会报错，只有遵循数据结构规则才能完成建模过程。数据结构中常用变量以及含义如表 3-1 所示。

表 3-1 常用变量及含义

数据项 定义

ID 个体标识符

TIME 发生的时间

DATE 发生的日期

DV 观测值

MDV 逻辑变量；与 DV 连用

EVID 事件标识符（说明数据记录的类型）

AMT 给药剂量

RATE 药物的给药速率

ADDL 除首次给药外的额外给药次数

ll 给药间隔

CMT 指定给药或采样发生的房室

SS 过去和当前给药的状态

该包遵循的数据结构类型基本上采用的是 NONMEM 软件标准的数据集结构， 但是 EVID 和 AMT 这两项存在区别。

在 NONMEM 软件中，EVID 列的值可取 0、1、2、3 和 4，EVID=0 表示该记录为观测事件；EVID=1 表示该记录为给药事件；EVID=2 表示该记录为其他类型的事件；EVID=3 和 EVID=4 表示该记录为系统复位后的值，复位后模型各房室内的药物剂量或浓度会被置为 0，两者区别在于 EVID=4 是 EVID=3 和 EVID=1 的组合，表明在给药记录之前即发生了复位[17]。AMT 项（给药剂量）只有在给药事件发生时才能赋值，观测事件或其他类型的事件下 AMT 项必须缺失，即 AMT 和 DV 项不能同时有记录。在 nlmixr2 程序包中 EVID 项将剂量类型和隔室项目都包含在其中。当 DV 为观测值时，EVID = 0。否则， EVID 的公式为：

EVID = 10000 ×(1 If IV Infusion 0 If Bolus)+ 100 × (Compartment)+ 1. nlmixr2 程序包对 EVID 项和 AMT 项做的重新定义如表 3-2 所示

表 3-2 EVID 项和 AMT 项的重新定义示意表[61]

数据项 定义

AMT 对于口服给药：药物剂量对于静脉推注：药物剂量

对于静脉滴注：注射开始时记录 1 条，

显示物滴注速率;注射结束时记录 1 条，显示-

1 \*药物滴注速率

EVID 对于观测事件：定义为 0

对于指定的隔室：剂量定义为 100\*(隔室号)+ 1

对于静脉注射剂量：事件定义为 10000 +

100\*(隔室号)+ 1

对于静脉注射：需要两个给药事件—— 一次开始注射(A 列的注射速率)，一次停止注射(AMT 列中的-1\*(注射速率))

当使用口服给药或者静脉推注时，AMT 代表的是剂量；使用静脉滴注时，AMT 是滴注速率，要关闭滴注，需要在滴注停止时使用数据集中的附加行减去最初添加的完全相同的速率。系统要求的数据结构已放入用户指南中。

* + 1. 数据探索界面

数据探索界面通过 fluidRow 函数和 column 函数创建了两行两列的布局方式，又通过 box 函数设置了基本框，分别为绘图设置、绘图展示和图表展示。绘图设置基本框内设置了数据选择框、探索药时曲线和探索 NCA。数据选择框内的所有数据均来自 PKdata 文件夹，从中选择出所要探索的数据文件；两大探索功能都可以添加设置（利用 HTML 技术在该界面弹出添加设置的网页），添加完绘图或计算非房室的设置以后，点击“绘图模拟”按钮或者“NCA 计算”按钮就可以运行，其结果分别通过 renderPlot 函数和 renderDataTable 函数在绘图展示和图表展示中呈现。数据探索界面和具体流程如图 3-5，3-6 所示。

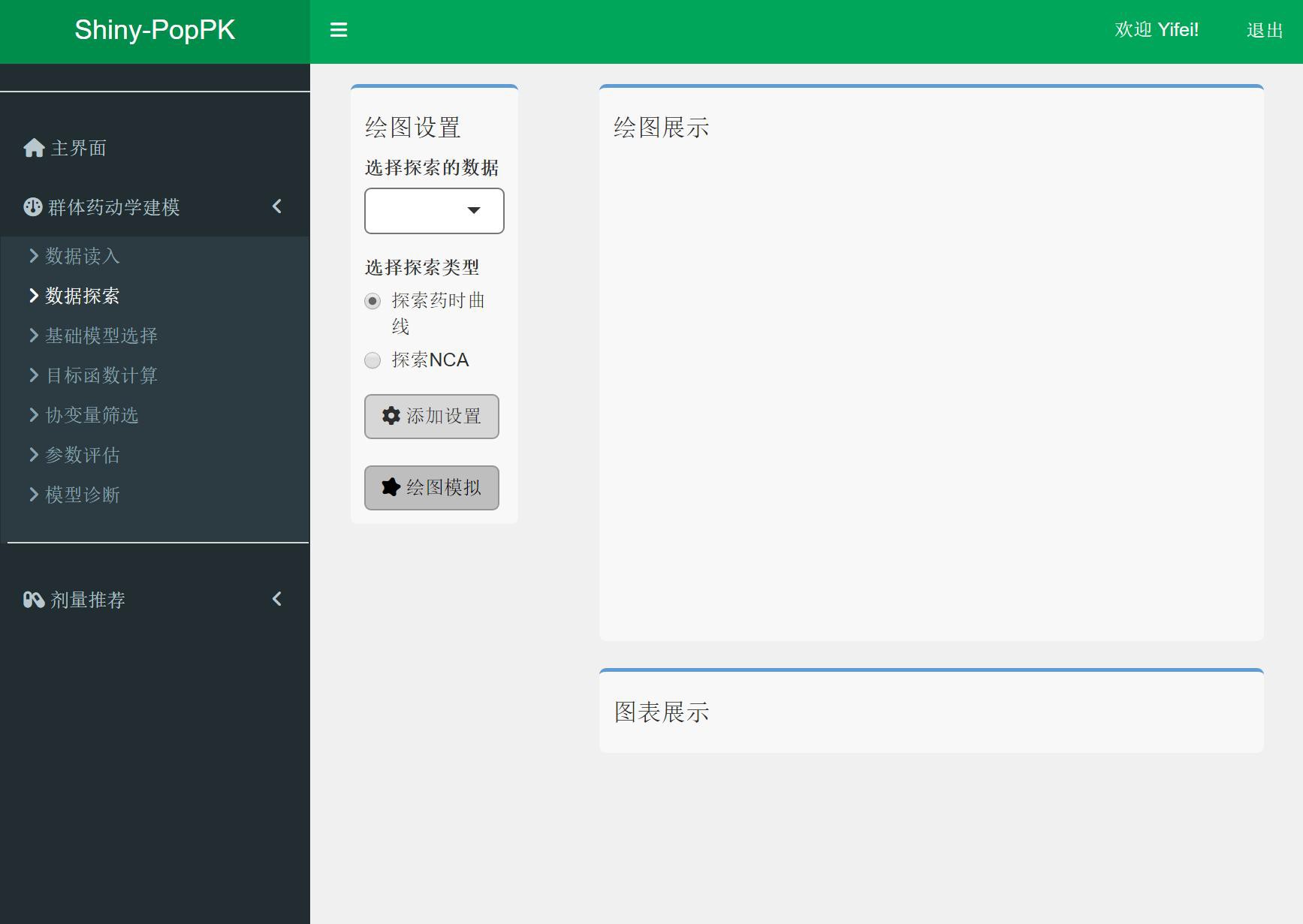


图 3-5 数据探索界面示意图

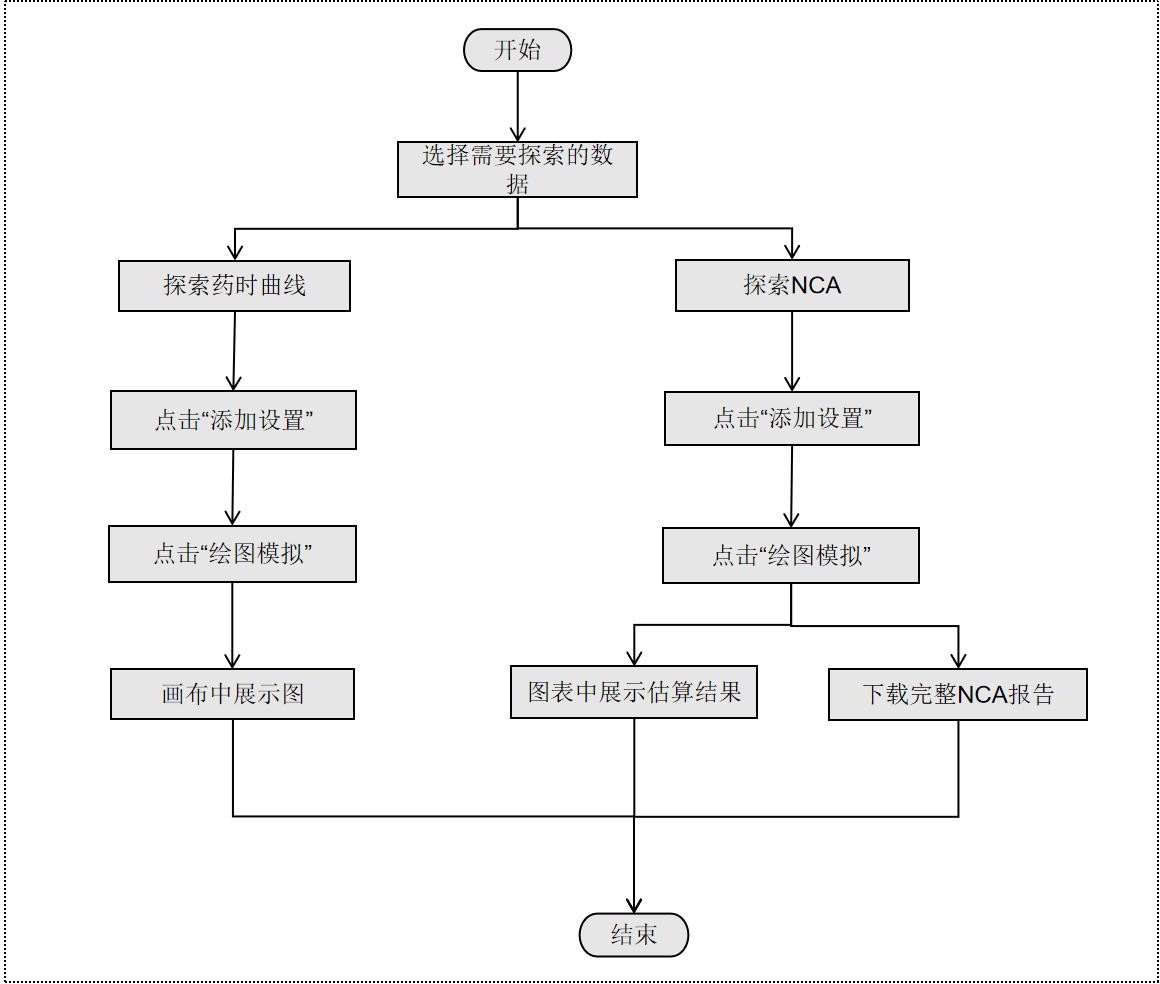


图 3-6 数据探索界面具体流程图

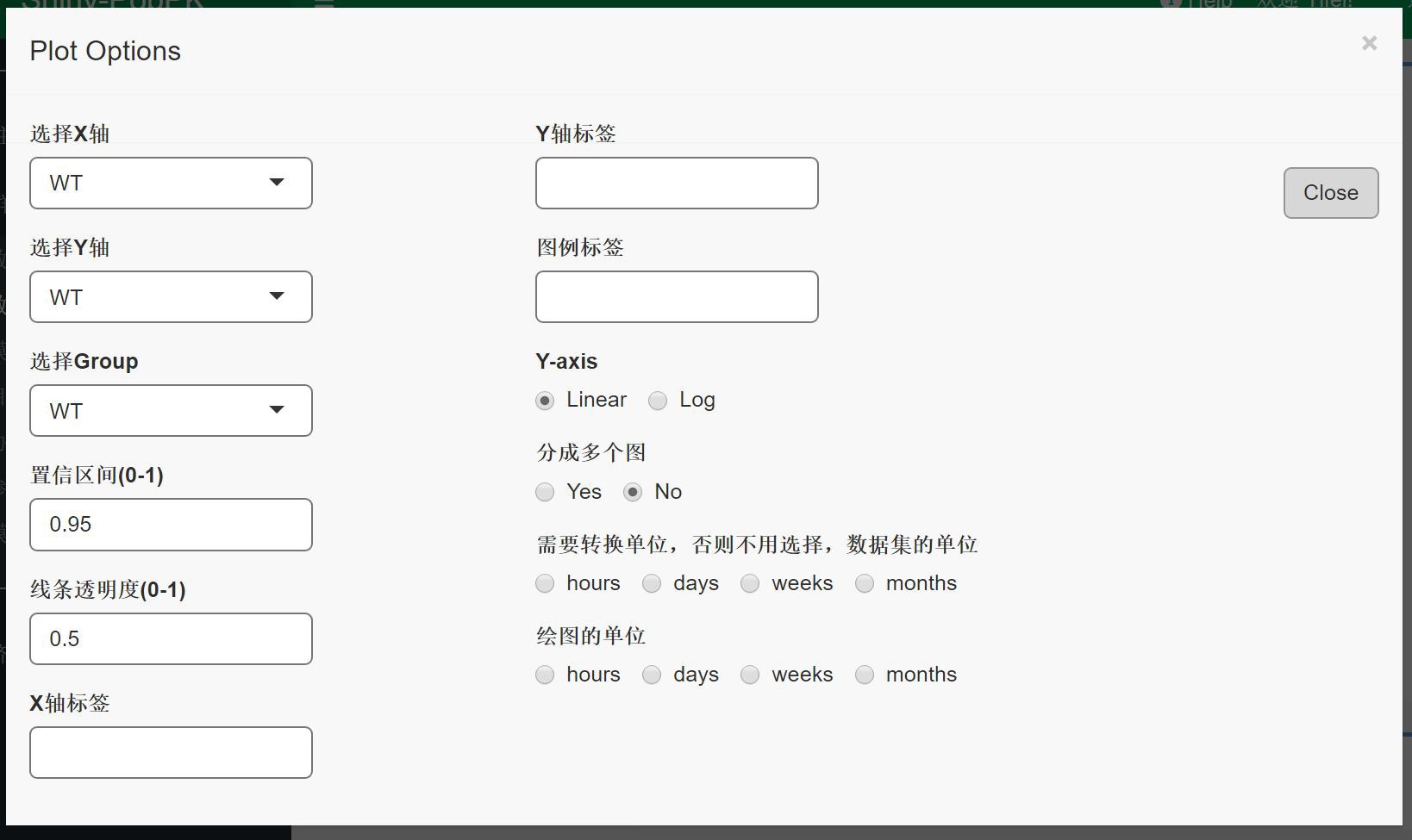
药时曲线绘图的设置内容比较丰富，用户需要分别为 X 轴、Y 轴和 Group 分组指定数据中某一列，选择框内的选项是通过 names(read.csv(p))函数获取的，p 是指选择需要绘图的数据的路径；在设置中用户可添加置信区间和线条透明度；对于 X 轴、Y 轴和图例标签，用户可以自行添加，若不添加，系统将默认采用选择的数据列名来进行标注；Y-axis 支持两种选择可以将 Y 轴的线性和对数形式进行转换；X 轴刻度上的时间单位也可以转换，支持的转换单位包括小时、天、周、月和年；也可将数据中的所有个体生成单独折线图，药时曲线绘图的设置如图 3-7 所示。

图 3-7 药时曲线绘图的设置示意图

NCA 分析功能的设置主要包括三个单位设置分别是浓度、时间和剂量，除此之外，还包括给药途径和输注时间。若为非静脉滴注，直接添加‘0’即可。NCA 分析功能的设置如图 3-8 所示。

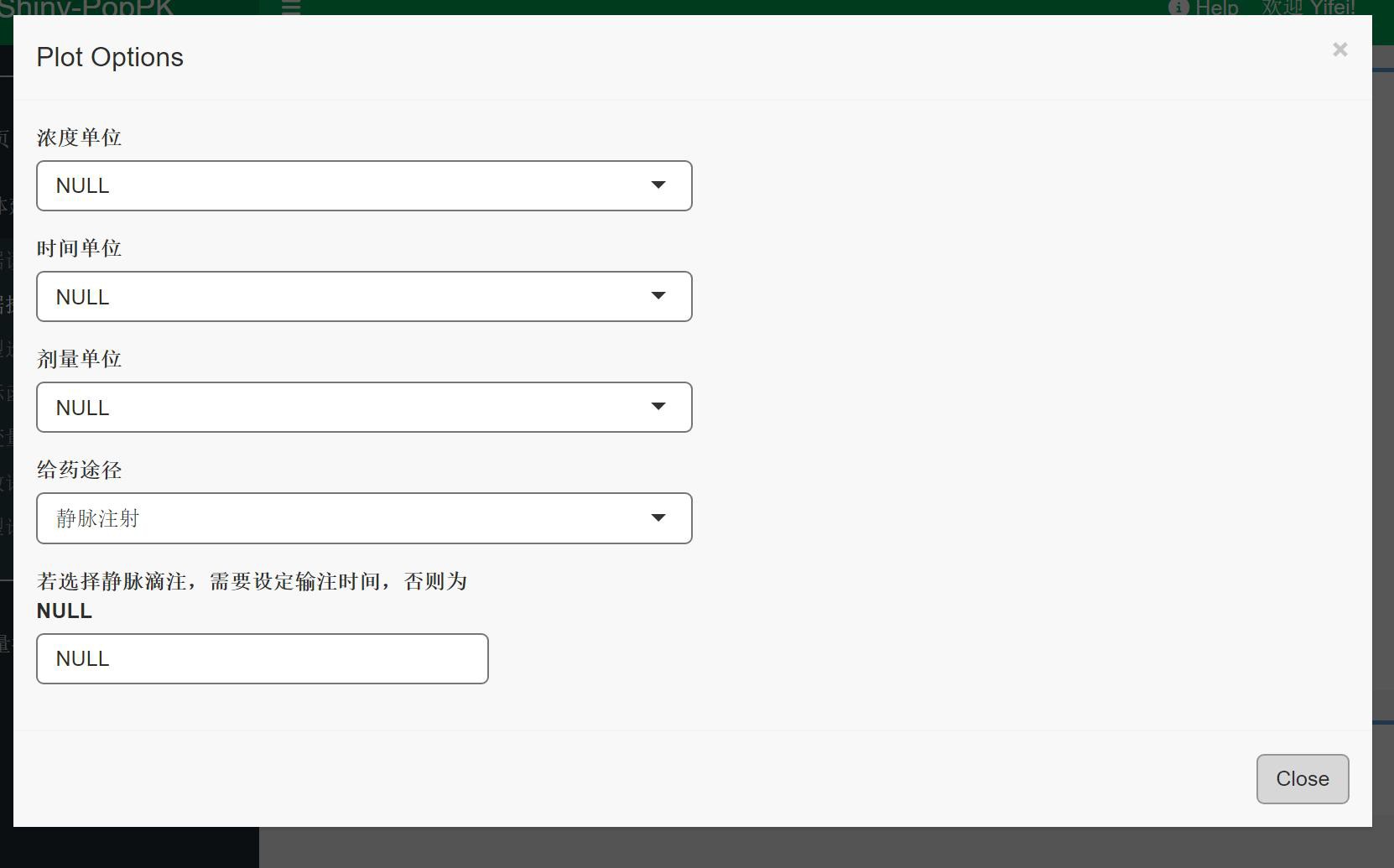


图 3-8 NCA 分析功能的设置示意图

* + 1. 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。

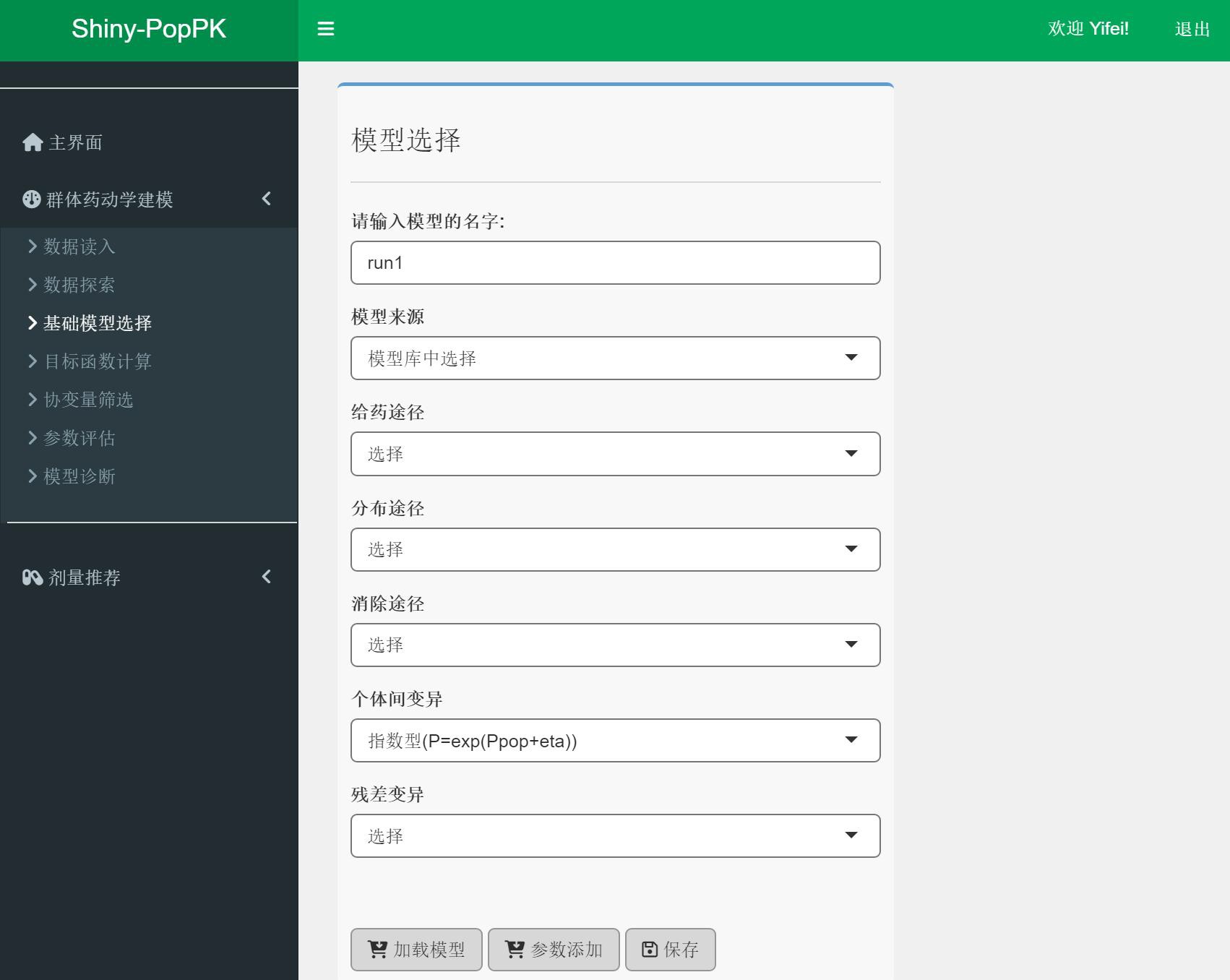


图 3-9 基础模型选择界面示意图

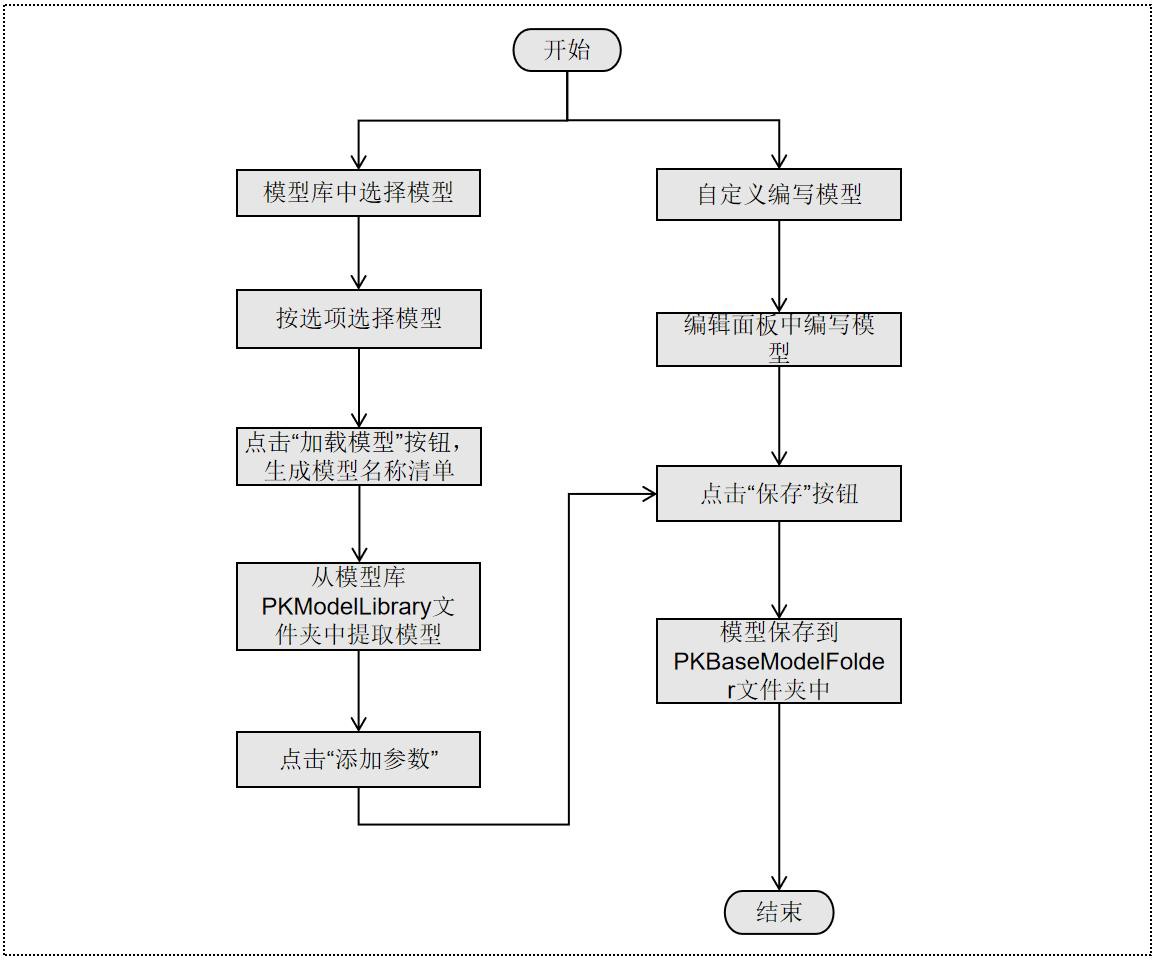


图 3-10 基础模型选择界面具体流程图

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

* + 1. 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。

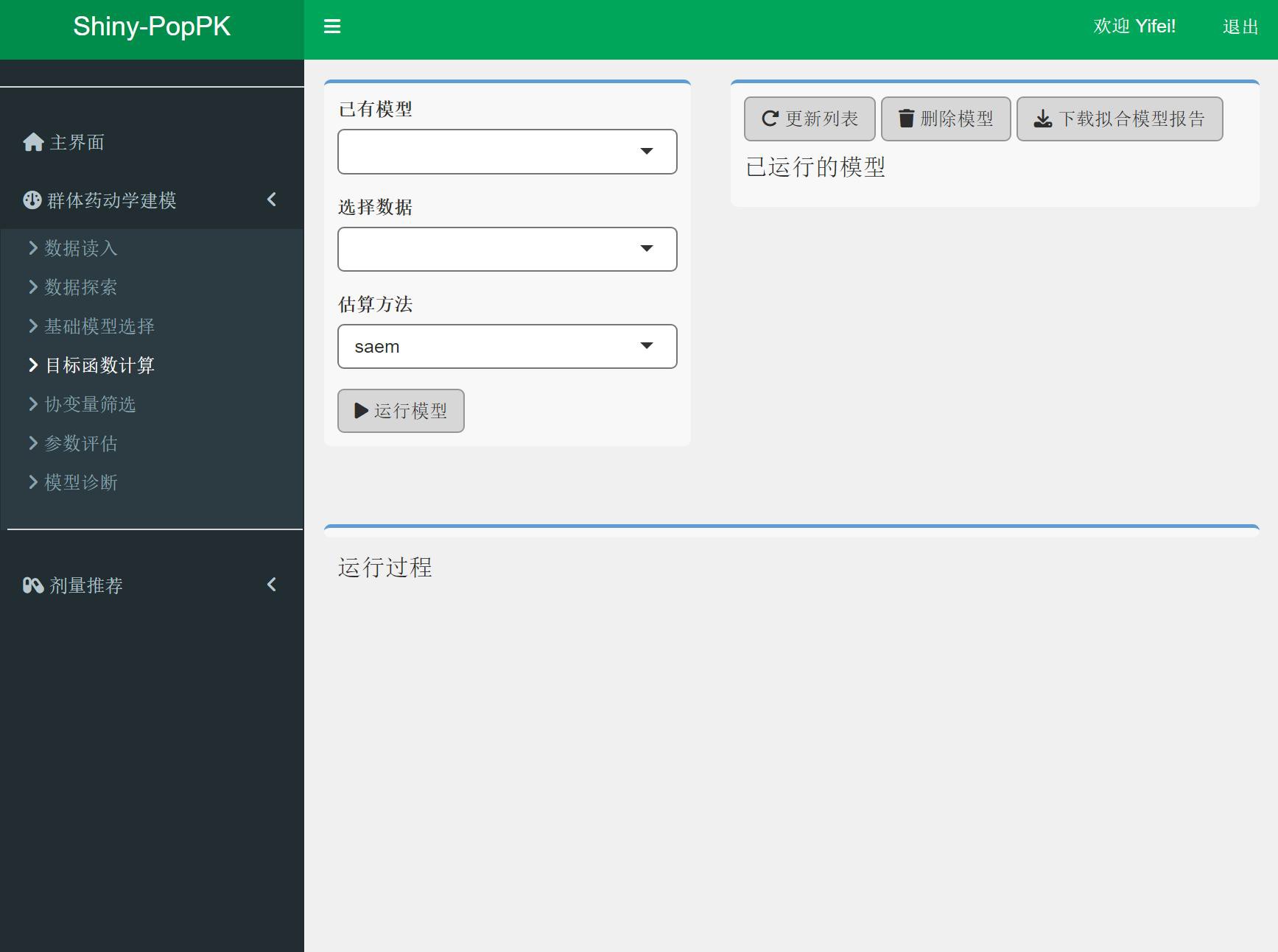


图 3-12 目标函数计算界面示意图

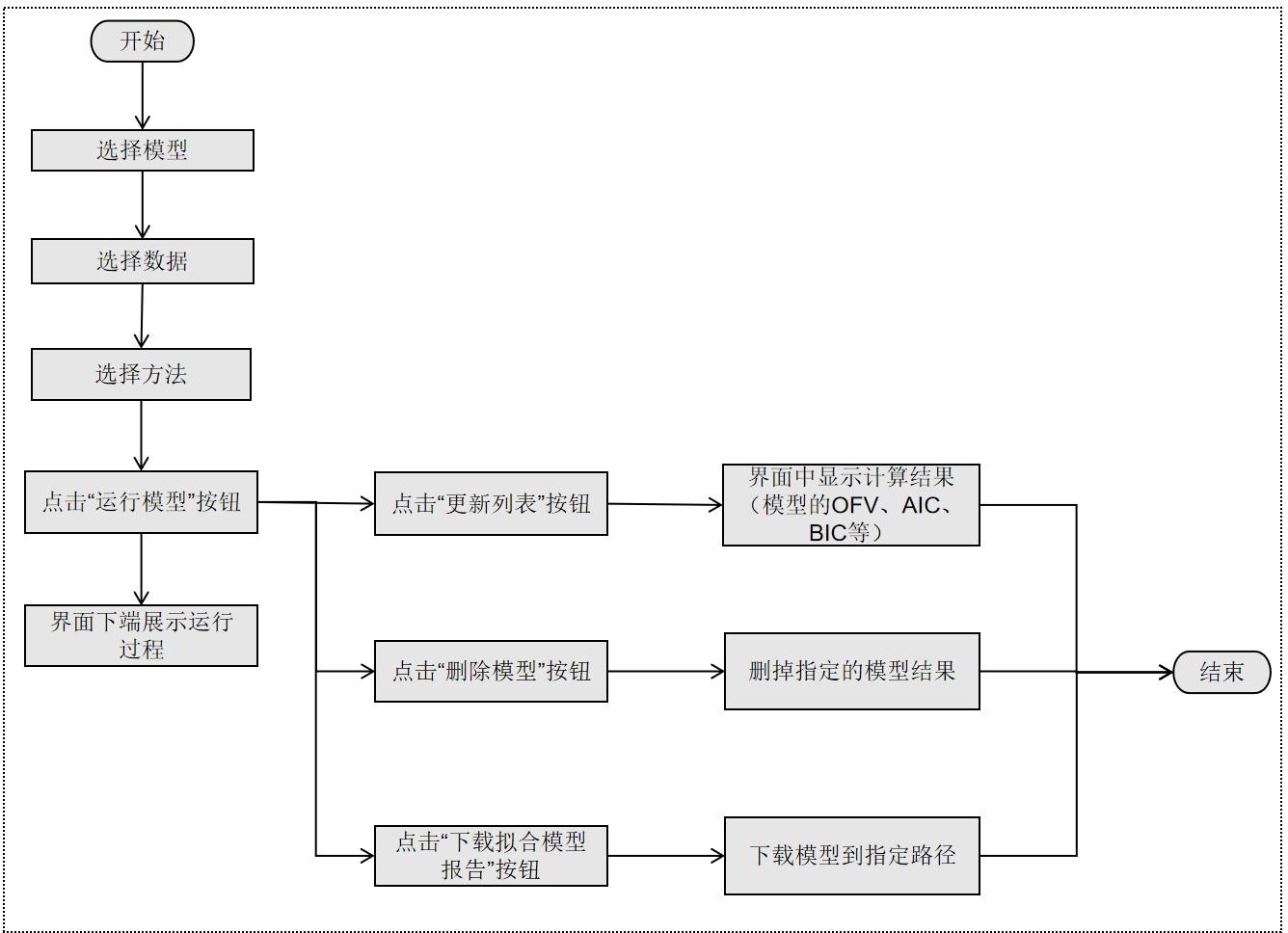


图 3-13 目标函数计算界面具体流程图

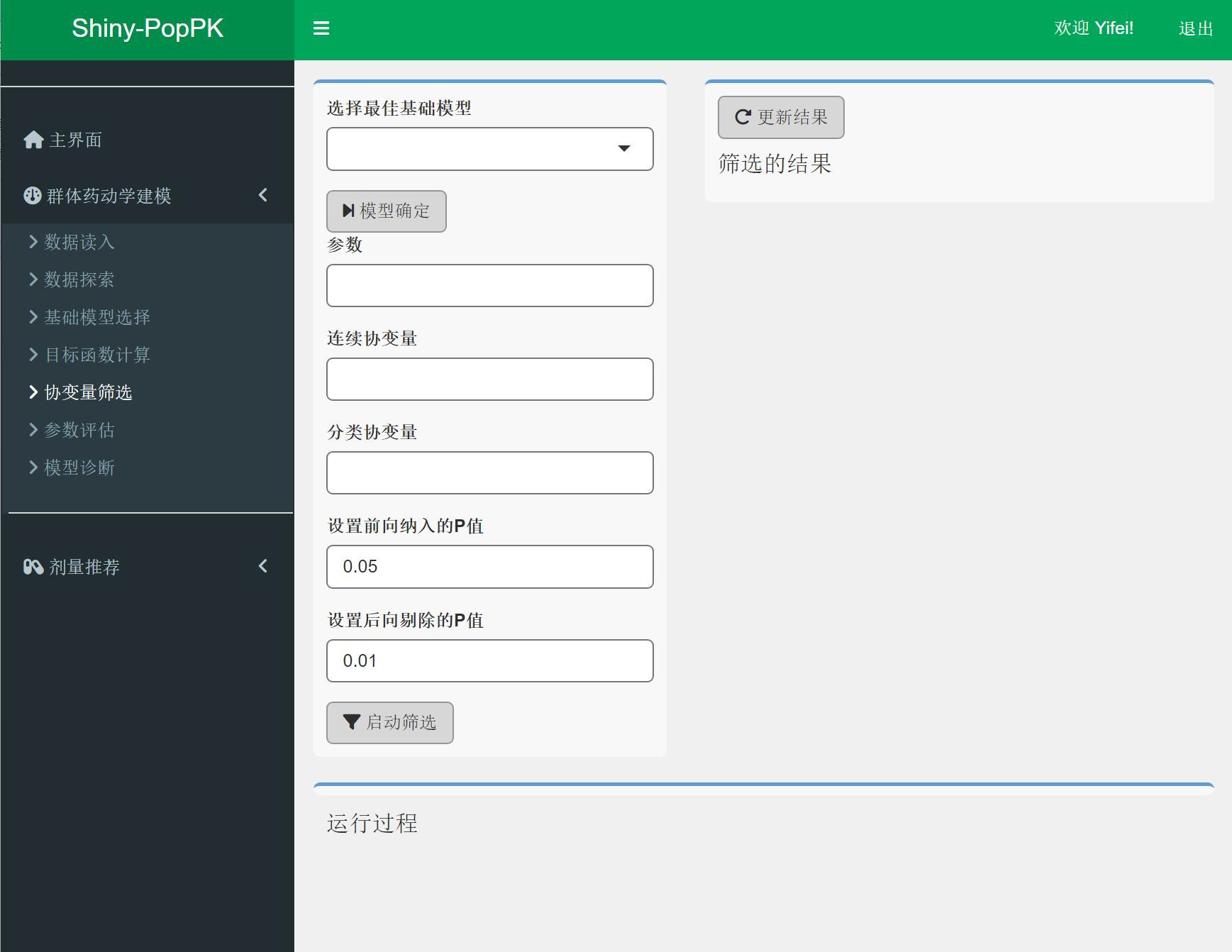
建模运行时，在 Rstudio 的 Console 控制台中呈现整个过程没有任何问题，但通

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。

* + 1. 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。



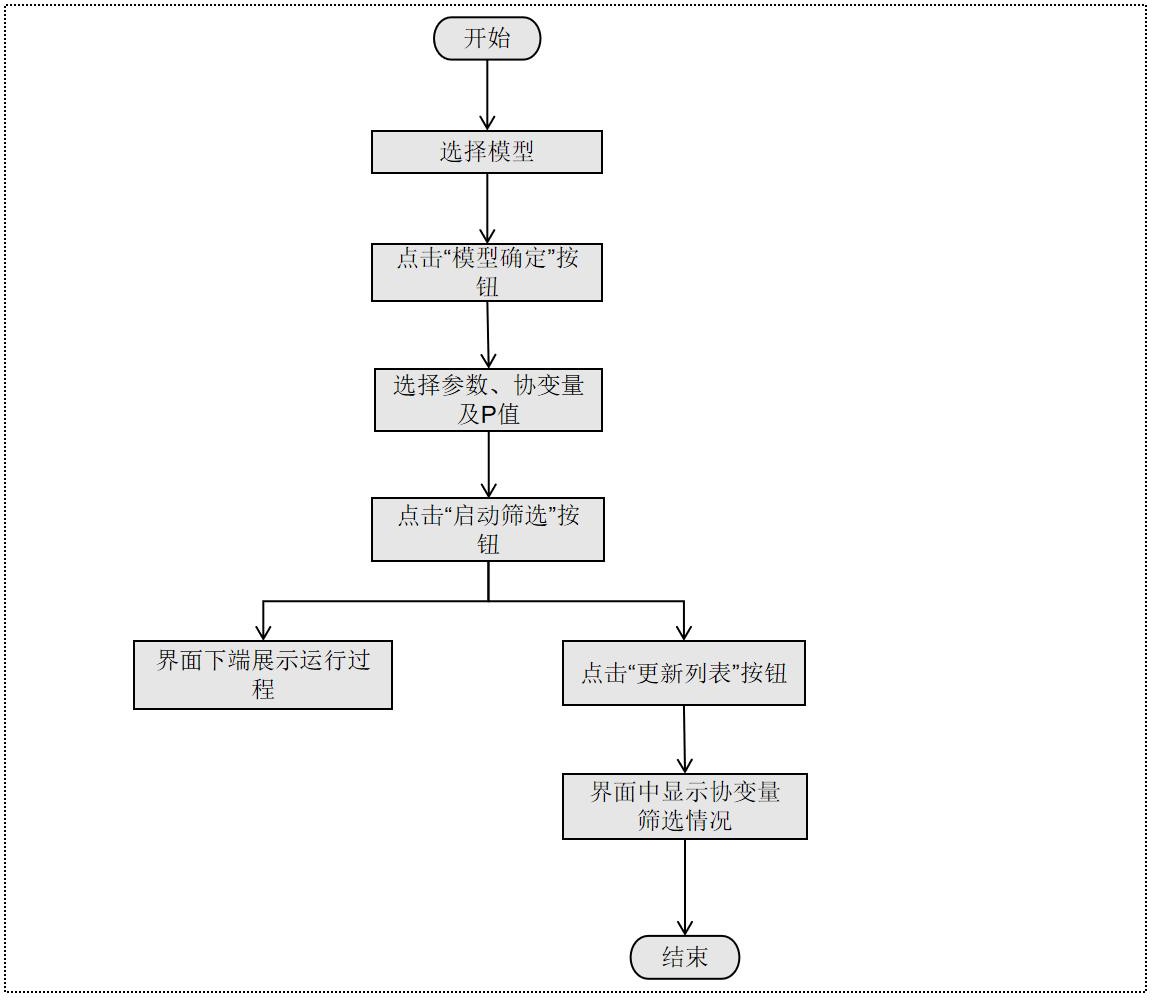
图 3-14 协变量筛选界面示意图

图 3-15 协变量筛选界面具体流程图

* + 1. 参数评估界面

参数评估界面整体布局依然采用上下界面两个部分，上部分界面包含协变量添加方式、选择需要评估的模型、模型确定等。下部分界面是运行过程的展示，参数评估界面和具体流程如图 3-16，3-17 所示。

图 3-16 参数评估界面示意图

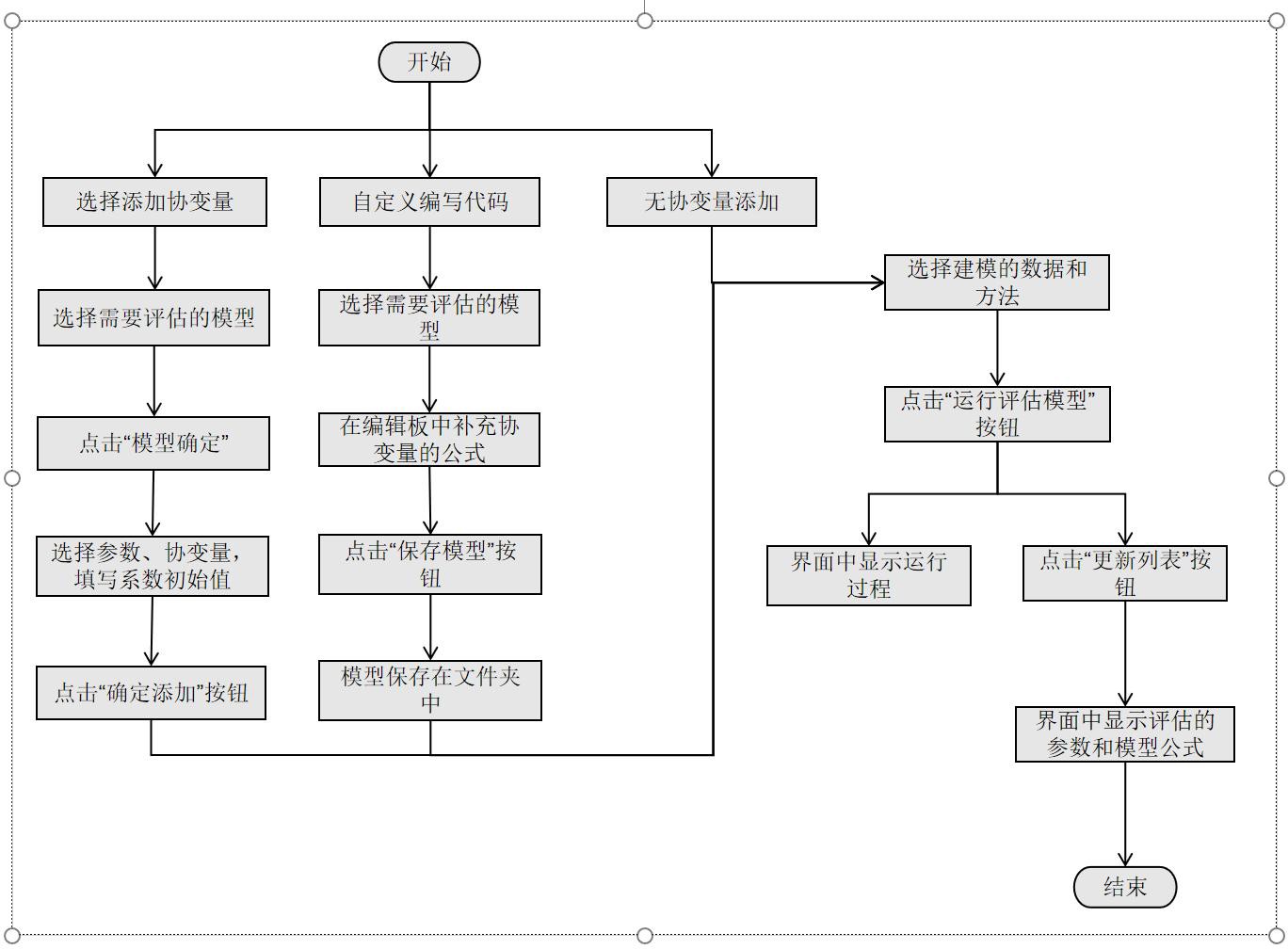


图 3-17 参数评估界面具体流程图

协变量添加有三种方式：

* + - 1. 用户按照系统内部设定的格式自动选择式添加；
      2. 用户在基础结构模型之上自行编辑添加协变量的代码；
      3. 用户没有筛选出协变量，就不用添加，直接进行参数评估。

选择第一种添加方式后， 首先选择需要评估的模型， 该模型均来自于PKObjResultsFolder 文件夹中，然后点击“模型确定”按钮，激活 ParameterOneTime 函数和 Extractparm 函数从 PKObjResultsFolder 文件夹的.ressum.rds 文件中获取参数和协变量，输入系数初始值（协变量加入参数公式时的权重系数），再点击确定添加按钮后，系统按照内部设定的格式写入模型公式中，内部设定的格式在本小节最后会详细介绍。

之后选择数据和估算方法，然后点击运行，这个运行过程会在下部分界面中展示出来，其结果将以.res.rds 格式和.ressum.rds 格式保存在 PKPEResultsFolder 文件夹中，.res.rds 格式的文件保存的依然是拟合后的 fit 对象，将用来进行模型诊断， 而.ressum.rds 格式的文件保存的是参数评估结果和模型的公式等信息。点击“更新列表”可激活“查看模型估计的参数”选择框，呈现 PKPEResultsFolder 文件夹中所有

的参数结果名称，选择其中一个，下面的展示框中将呈现参数评估结果以及模型公式，目的是让用户了解该模型的具体公式，以便得到完整的最终模型。

选择第二种添加方式之后，首先选择对哪个基础模型进行添加协变量，在“选择需要评估的模型”选择框内选中模型后，下方的编辑面板就会显出该基础模型的代码，用户在这个基础代码之上编写添加协变量的公式，然后保存模型，最后选择数据和方法，点击运行即可，自定义编写代码部分如图 3-18 所示。

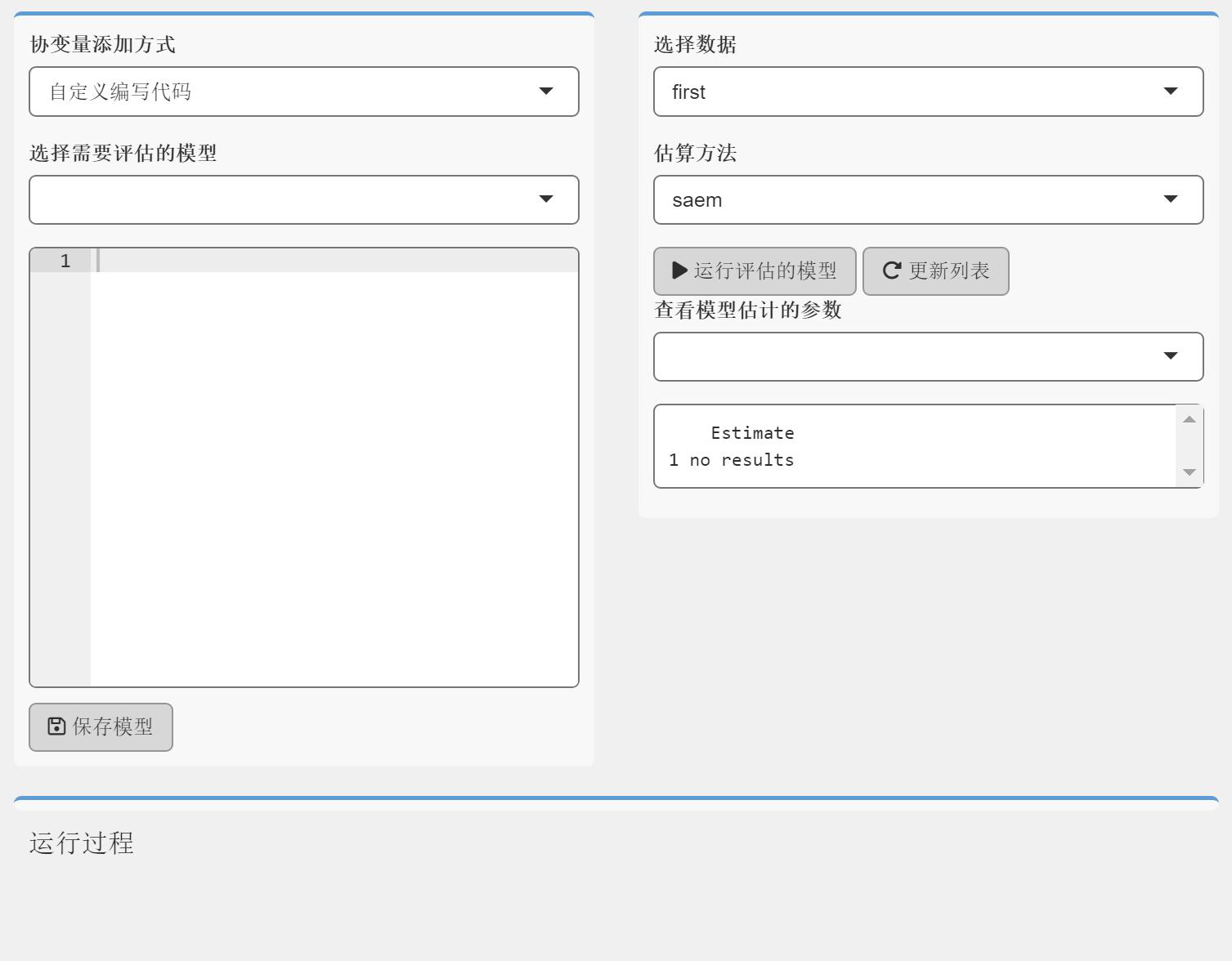
若选择第三种，没有筛选出协变量，则无需添加，直接选择数据和算法，运行模型即可。

图 3-18 自定义编写代码部分示意图

系统内部设定的协变量添加格式，需要先定参数，然后在定一个或多个协变量以及权重，比如体重 WT 与清除率 CL 有关系，WTeff 为它们之间的权重,则具体的公式为 CL = exp(tcl + WTeff\*(log(WT/70)+ eta.cl),若性别 SEX 也与清除率有关，SEXeff 为权重，则具体的公式为 CL = exp(tcl + WTeff\*(log(WT/70)+SEXeff \*SEX + eta.cl)。因协变量比较多，目前文献中没有过多涉及，只有常见的体重和性别较为统一，如体重一般常用 WTeff\*(log(WT/70)格式，性别常用 SEXeff \*SEX 格式，为了实现协变量添加功能的开发，本文将其它协变量添加格式设定为权重\*协变量。

* + 1. 模型诊断界面

模型诊断界面分上下界面两个部分，上部分界面用于选择绘图的条件，下部分界面展示图片。上部分界面包含选择需要绘图的模型、绘图类型选择、选择绘制的类别（内部有四十种绘图方法）、启动绘图和保存绘图。

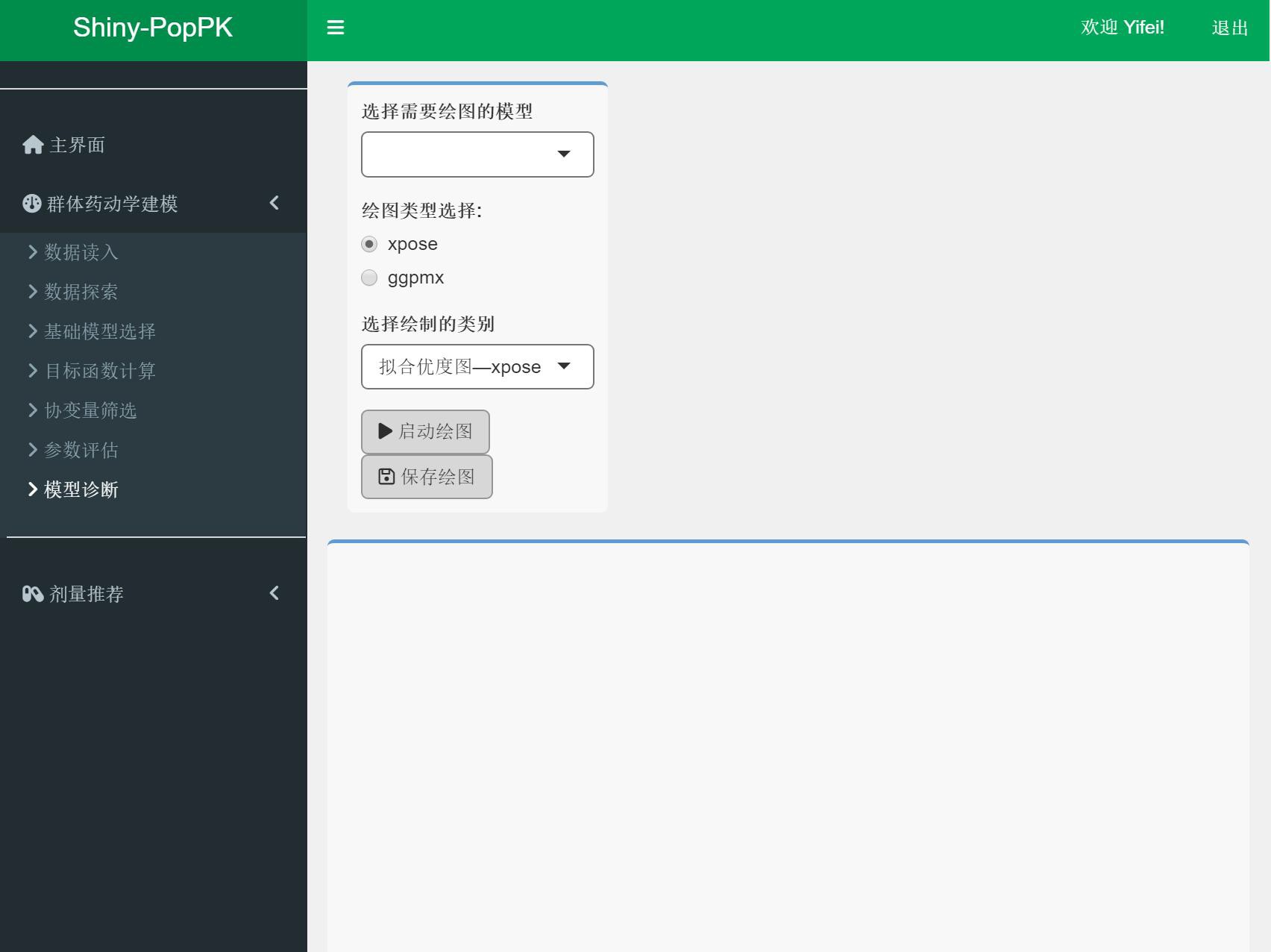
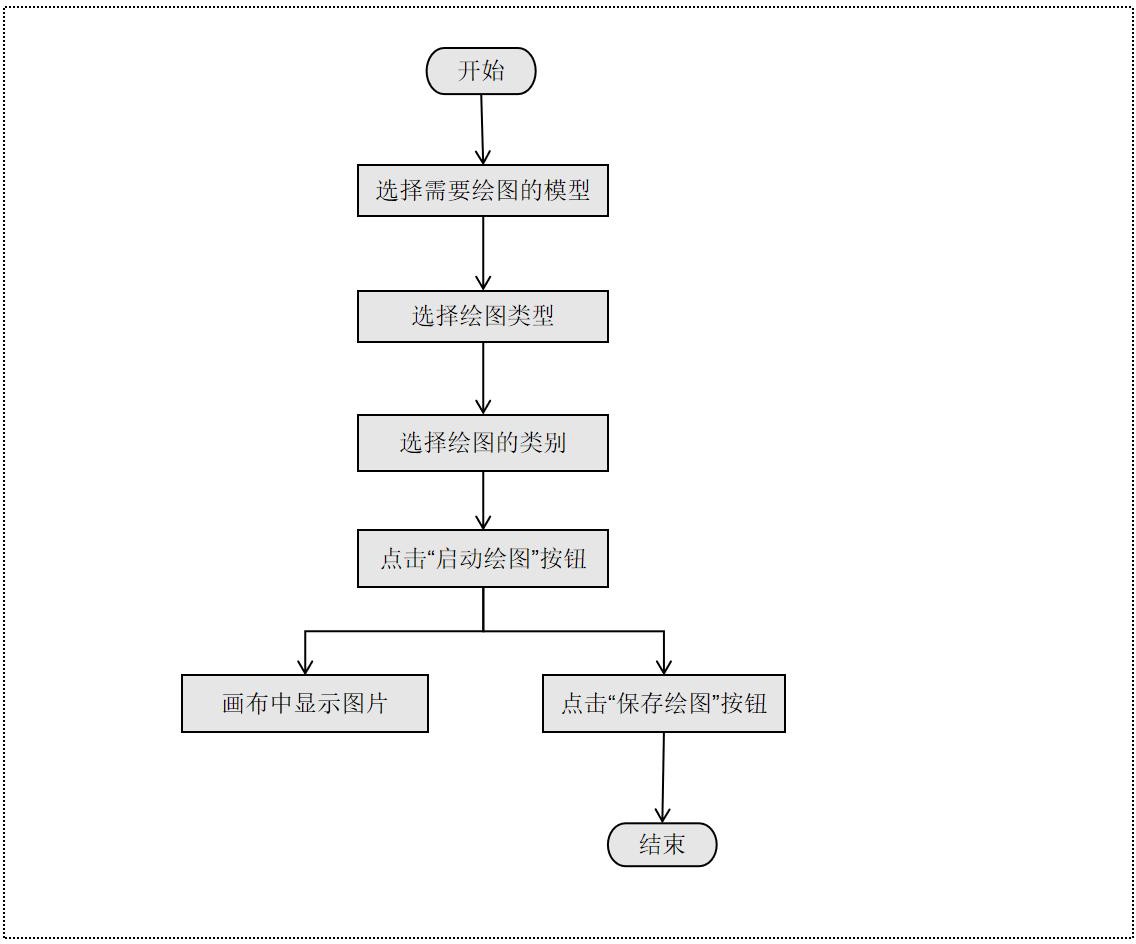
在“ 选择需要绘图的模型” 选择框中选择模型， 所有模型均来自于PKPEResultsFolder 文件夹中的.res.rds 文件；用 radioButtons 函数设置了两种绘图类别的单选按钮，其代表的是 xpose 和 ggpmx 两种绘图风格不同的程序包，ggpmx 程序包绘制的图片种类比较多，xpose 程序包绘制的图片较为简洁优雅；点击“启动绘图”按钮即可在下部分界面的面板中显示其结果；添加了 withLoader 函数可以在绘图过程中显示加载动画，避免用户认为系统未响应；点击“保存绘图”按钮后，首先系统会在 PKDrawingFolder 文件夹中生成一个以该模型名称命名的空文件夹，然后将绘制的图片以 PDF 格式保存至该文件夹中，避免保存图片过多而产生混乱的现象。模型诊断界面和具体流程如图 3-19，3-20 所示。

图 3-19 模型诊断界面示意图



## 剂量推荐界面

图 3-20 模型诊断界面具体流程图

剂量推荐界面包括模型输入和剂量优化两大功能。

* + 1. 模型输入界面

模型输入界面分左右两个界面，左部分界面用来输入最终模型，该模型将保存在 PLModelFolder 文件夹中。右部分界面用来输入患者数据信息，其输入方式有两种：（1）在界面上点击添加设置，在弹出的数据框内填写数据即可，此种方法仅限无协变量的情况；（2）上传数据，此种方法有无协变量都可以。第一种填写数据的方式还细分了填写类型：静脉滴注和口服给药或静脉推注两种情况，因为静脉滴注还需要填写滴注时间。点击“添加设置”后激活 renderUI 函数可滑出弹窗，弹窗内需要填写的信息包括数据名称、ID、观测时间、血药浓度、给药剂量、给药事件。若填写类型为静脉滴注，还需填写滴注时间。数据填写完成后，点击“确定数据” 按钮，系统将以 csv 格式保存到 PLData 文件夹中，并显示保存成功的提示；若选项中存在漏填将提示保存失败。模型输入界面、具体流程和数据填写设置如图 3-21， 3-22，3-23 所示。

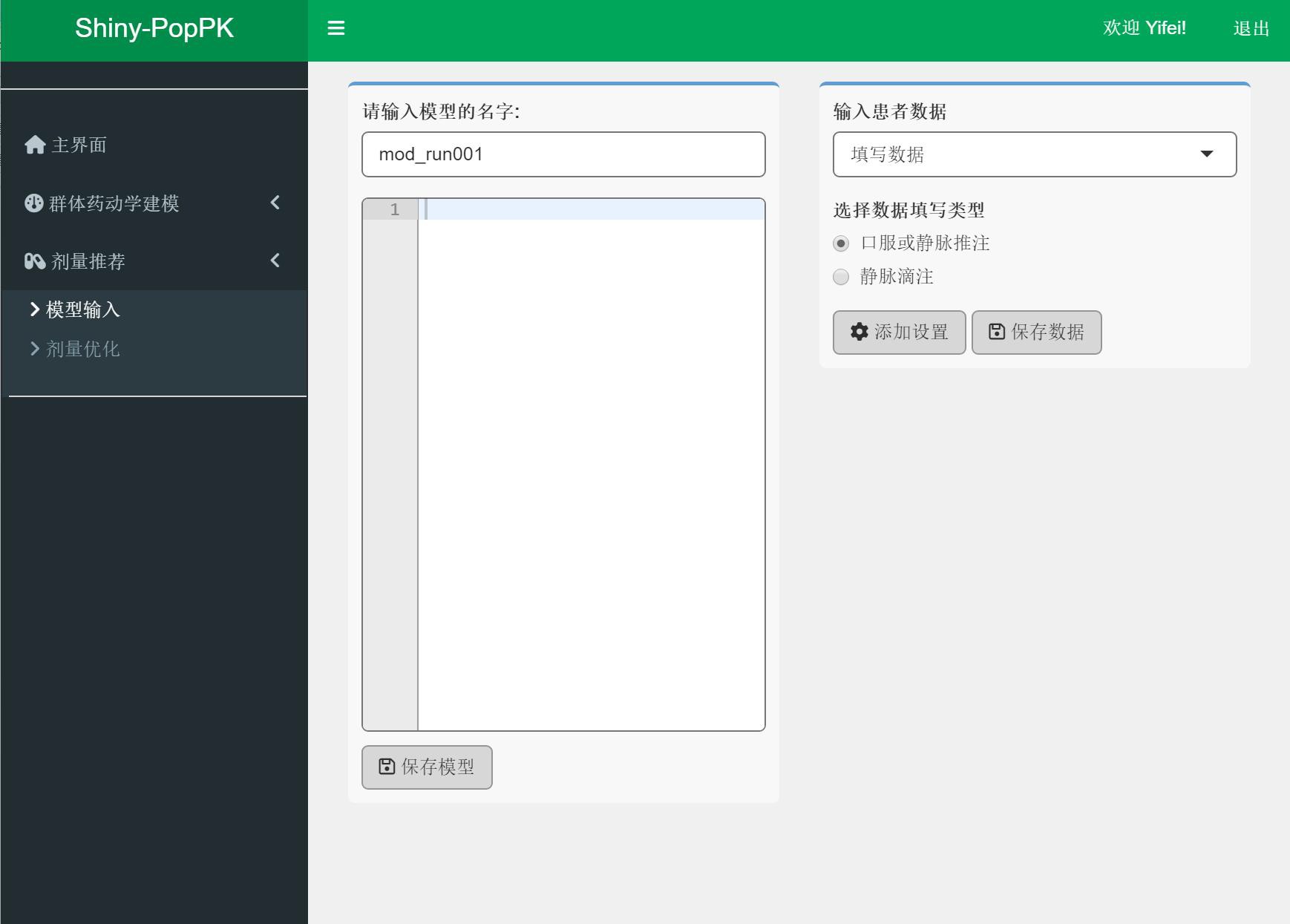


图 3-21 模型输入界面示意图

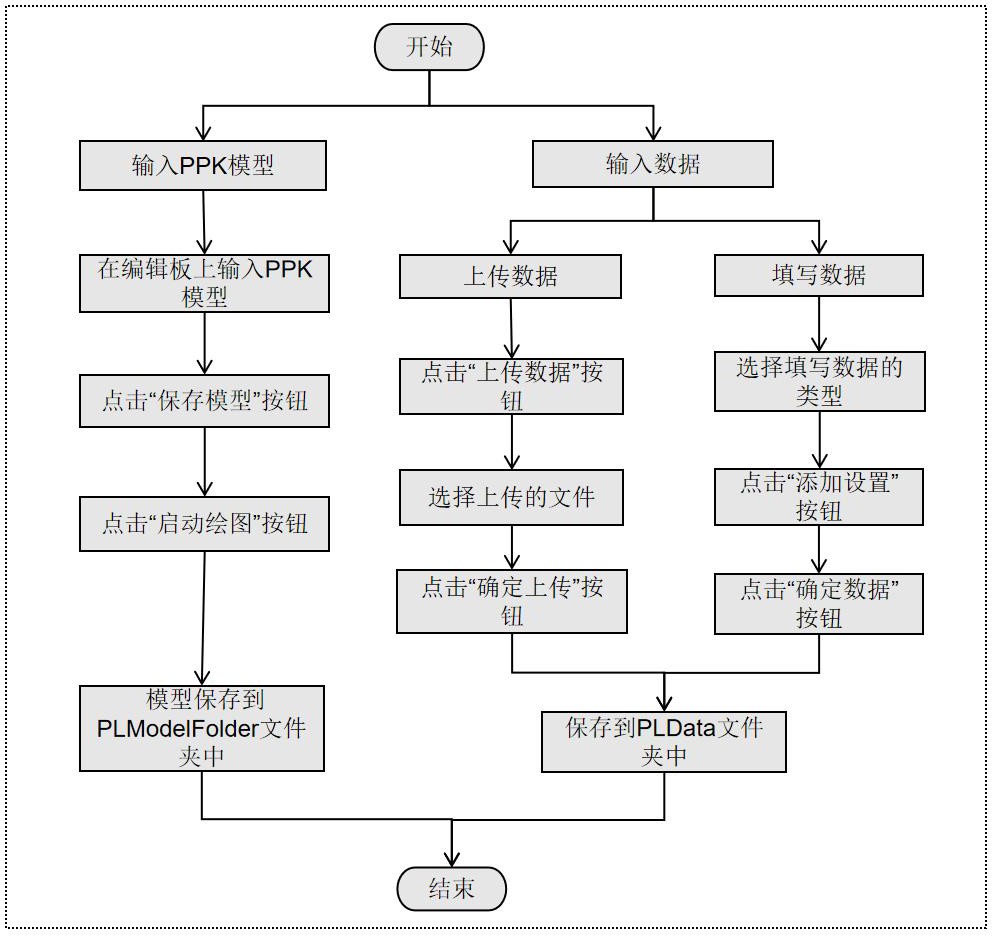


图 3-22 模型输入界面具体流程图

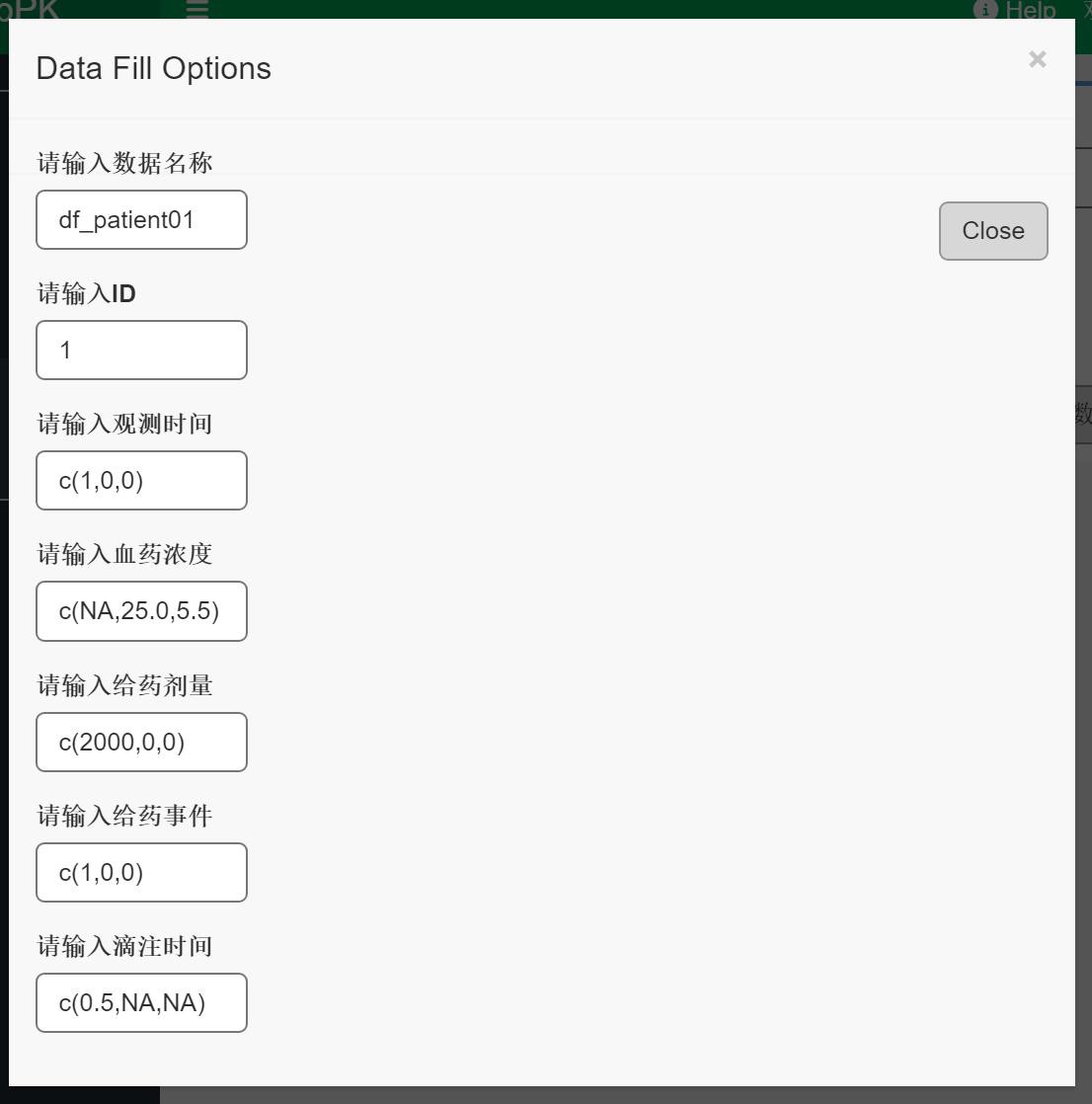
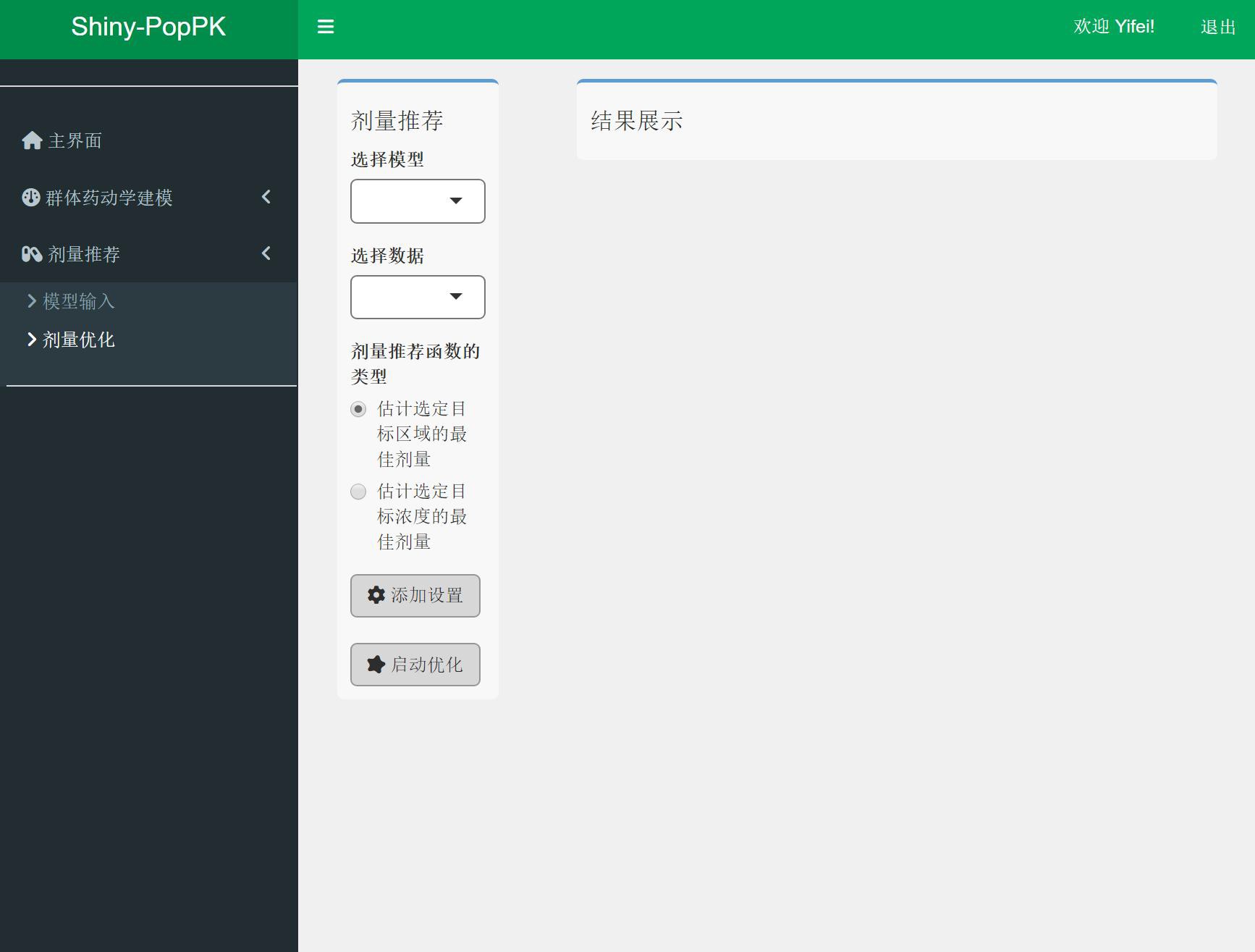


图 3-23 数据填写示意图

* + 1. 剂量优化界面

剂量优化界面分为左右两个界面，左部分界面包含选择数据和模型以及方法， 右边界面通过调用 renderText 函数显示结果。“选择模型”选择框内的模型均来自PLModelFolder 文件夹，“选择数据”选择框内的数据均来源于 PLData 文件夹；然后选择剂量优化的方法，剂量优化的方法共有两种，分别是估计选定目标区域的最佳剂量(目标 AUC 法)和估计选定目标浓度的最佳剂量（目标浓度法）。剂量优化界面和具体流程如图 3-24，3-25 所示。



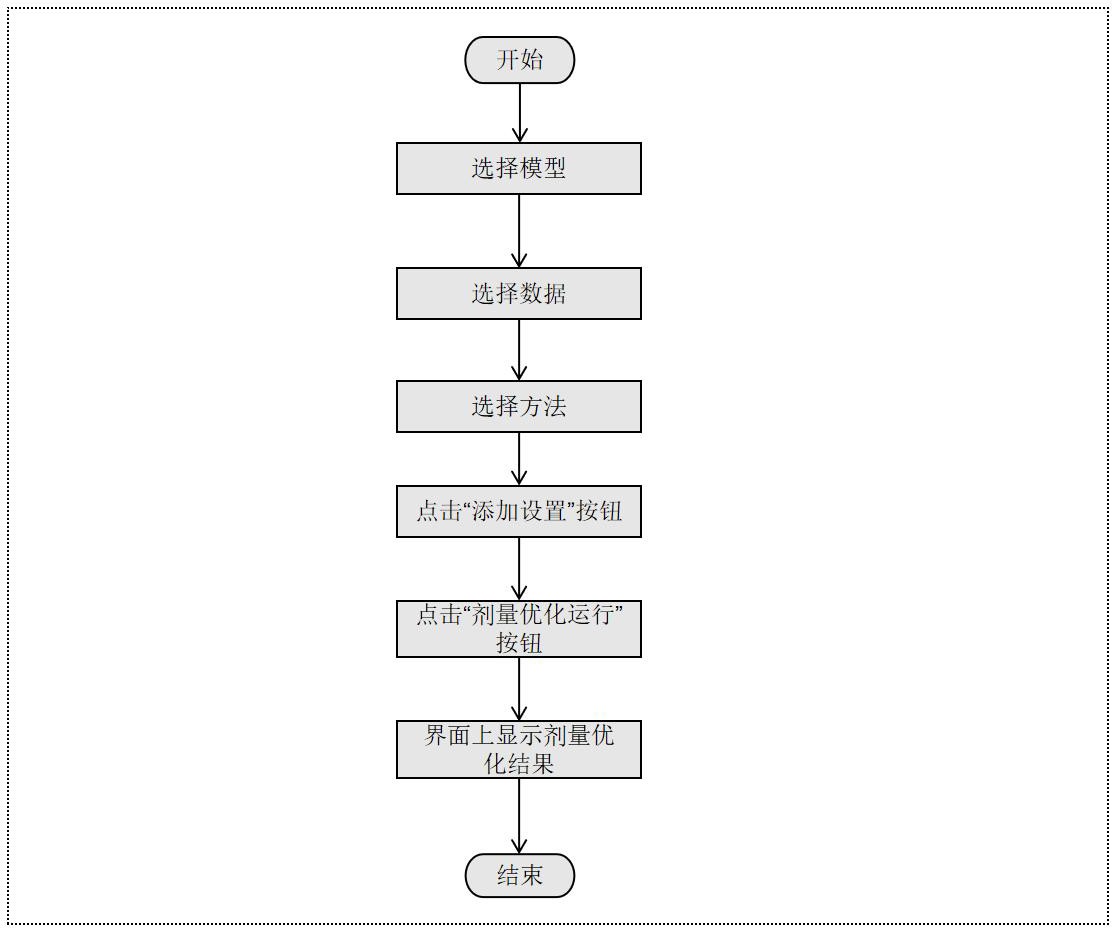
图 3-24 剂量优化界面示意图

图 3-25 剂量优化界面具体流程图

选择“估计选定目标区域的最佳剂量”单选按钮，然后点击添加设置，激活renderUI 函数可生成弹窗，根据弹窗内包含需要填写的设置内容。设置内容包含计算AUC 开始的时间、终止的时间、给药时间、目标 AUC 的值、给药时间间隔、给药次数、输注时间和起始剂量。给药次数若为 1 次，即代表是单剂量给药，大于 1 次则为多剂量给药，这里剂量优化是对起始剂量进行的优化， 估计选定目标区域的最佳剂量设置如图 3-26 所示。

图 3-26 估计选定目标区域的最佳剂量设置示意图

选择“估计选定目标浓度的最佳剂量”单选按钮，然后点击添加设置，激活renderUI 函数可生成弹窗，填写的信息基本和第一种方法相同，不同的是该方法主要是对达到目标 AUC 的时间点进行优化，比如设置的目标 AUC 为 80，达到目标 AUC 的时间点为 12，则表示在 12 这个时间点达到 AUC 为 80 所需要的剂量，估计所选目标浓度的最佳剂量设置如图 3-27 所示。

图 3-27 估计所选目标浓度的最佳剂量示意图

# 第四章系统测试与应用

## 系统测试

系统开发完成后，需要进行系统测试，系统测试的目的是用来检验系统在运行过程中的完整性和安全性，保证系统能在安全可控的条件下完成测试目标。

* + 1. 测试数据

本系统采用多种不同的数据对软件进行测试，测试数据的类型包括模拟数据集、多剂量苯巴比妥数据集和单剂量茶碱数据集。

模拟数据集分为口服给药、静脉滴注和静脉推注三种类型，均为多剂量给药， 每个类型含有 120 例病人，共 7921 条数据，除了 ID、TIME、DV 和 AMT 基本信息外，还含有 WT 和 AGE 信息。

多剂量苯巴比妥数据集为口服给药，含有 59 例病人，共 745 条数据，除四项基本信息外，还有 WT、EVID 和 MDV 信息。

单剂量茶碱数据集为口服给药，含有 12 例病人，共 145 条数据，除四项基本信息外，还有 WT、EVID 和 CMT 信息。

* + 1. 测试内容

测试的内容包括系统功能测试和系统兼容性测试。常用的测试方法是单元测试和集成测试。单元测试的内容是对系统中每个单独的模块进行测试，确保每个单元的功能符合设计规范，减少代码中的错误。集成测试的内容是对系统进行整体测试，确保模块的集成是无缝的。本系统划分成多个功能模块，在设计时采用高内聚、低耦合的原则，所以测试重点放在单元测试。

* + - 1. 系统功能测试

Shiny-PopPK 系统的主要功能可划分为两大类：PPK 建模和剂量推荐。本文采用

* + 1. 节中的测试数据对系统两大功能进行测试，为保证测试有效性，本测试执行的次数为三次以上,采用 shinytest2 包和 testthat 包进行功能测试，其记录每个模块运行的时间,除此之外还利用 profvis 包测试了性能，其功能测试和性能测试运行过程见附录 3。
       1. 群体药动学建模模块测试的功能包含数据上传、列名转换、药时曲线绘制、

NCA、基础模型选择、自定义模型编写、目标函数计算、拟合报告下载等，对这些功

能采用多种数据多次执行的方式进行测试，确保用户都能准确无误地完成整个 PPK 建模过程。PPK 建模模块具体的功能测试如表 4-1 所示。

表 4-1 群体药动学建模模块功能测试表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 编号 | 测试功能 | 具体操作 | 运行时间/ms | 预期结果 | 实际结果 |
| 1 | 数据上传 | 系统正常运 | 12 | 系统成功将数 | 符合预期 |
|  |  | 行，点击“输 |  | 据上传到服务 |  |
|  |  | 入数据”按 |  | 器，界面中可 |  |
|  |  | 钮，选择本地 |  | 预览上传的数 |  |
| 2 | 列名转换 | 文件  系统正常运 | 14 | 据  系统成功完成 | 符合预期 |
|  |  | 行，从四种选 |  | 列名转换，并 |  |
|  |  | 择框中选择对 |  | 将数据保存到 |  |
|  |  | 应的数据列 |  | 文件夹中 |  |
| 3 | 药时曲线绘制 | 名，点击按钮  系统正常运 | 33 | 系统成功完成 | 符合预期 |
|  |  | 行，选择数 |  | 曲线绘制，并 |  |
|  |  | 据，添加设 |  | 显示在界面中 |  |
|  |  | 置，点击“绘 |  |  |  |
| 4 | NCA | 制模拟”按钮系统正常运 | 45 | 系统成功完成 | 符合预期 |
|  |  | 行，选择数 |  | NCA 计算，并 |  |
|  |  | 据，添加设 |  | 显示在界面中 |  |
|  |  | 置，点击 |  |  |  |
|  |  | “NCA 计算” |  |  |  |
| 5 | 基础模型选择 | 按钮  系统正常运 | 9 | 系统成功完成 | 符合预期 |
|  |  | 行，选择模 |  | 模型加载过 |  |
|  |  | 型，点击按 |  | 程，并保存在 |  |
|  |  | 钮，加载模型 |  | 文件夹中 |  |
|  |  | 并添加初始参 |  |  |  |
| 6 | 自定义编写模 | 数  系统正常运 | 8 | 系统成功将模 | 符合预期 |
|  | 型 | 行，在编辑板 |  | 型保持在文件 |  |
|  |  | 中编写模型代 |  | 夹中 |  |
|  |  | 码，并点击 |  |  |  |
| 7 | 目标函数计算 | “保存”按钮  系统正常运 | 8166 | 系统成功实现 | 符合预期 |
|  |  | 行，选择数 |  | OFV 计算，并 |  |
|  |  | 据、模型和方 |  | 显示运行过程 |  |
|  |  | 法，点击“运 |  | 和结果 |  |
|  |  | 行模型”按钮 |  |  |  |
|  |  | 和“更新列 |  |  |  |
| 8 | 拟合报告下载 | 表”按钮  系统正常运 | 11 | 系统成功将报 | 符合预期 |
|  |  | 行，点击“下 |  | 告下载到指定 |  |
|  |  | 载拟合报告” |  | 文件夹中 |  |
| 9 | 协变量筛选 | 按钮  系统正常运 | 32800 | 系统成功实现 | 符合预期 |
|  |  | 行，选择模 |  | 协变量的筛选 |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | 型、参数、协 |  | 并显示运行过 |  |
|  |  | 变量和设置 P |  | 程和结果 |  |
|  |  | 值，点击“启 |  |  |  |
|  |  | 动筛选”按钮 |  |  |  |
|  |  | 和“更新结 |  |  |  |
| 10 | 参数评估 | 果”按钮  系统正常运 | 9873 | 系统成功实现 | 符合预期 |
|  |  | 行，分别使用 |  | 协变量的添加 |  |
|  |  | 两种协变量添 |  | 并显示估算过 |  |
|  |  | 加的方式和无 |  | 程和结果 |  |
|  |  | 协变量添加的 |  |  |  |
|  |  | 方式，选择数 |  |  |  |
|  |  | 据和模型，点 |  |  |  |
| 11 | 模型诊断 | 击按钮  系统正常运 | 2240 | 系统成功实现 | 符合预期 |
|  |  | 行，选择模型 |  | 四十种诊断图 |  |
|  |  | 和类别，点击 |  | 的绘制，并显 |  |
|  |  | “启动绘图” |  | 示在界面中 |  |
| 12 | 诊断图保存 | 按钮  系统正常运 | 10 | 系统成功实现 | 符合预期 |
|  |  | 行，点击“保 |  | 图片保存功 |  |
|  |  | 存绘图”按钮 |  | 能，并保存到 |  |
|  |  |  |  | 文件夹中 |  |

* + - 1. 剂量推荐模块测试的功能包含模型输入、数据输入、数据填写和剂量推荐功能，对这些功能依然采用多种数据多次执行的方式进行测试，以确保用户可以成功完成整个剂量推荐的过程。剂量推荐模块具体的功能测试如表 4-2 所示。

表 4-2 剂量推荐模块功能测试表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 编号 | 测试功能 | 具体操作 | 运行时间 | 预期结果 | 实际结果 |
| 1 | 模型输入 | 系统正常运 | 12 | 系统成功实现 | 符合预期 |
|  |  | 行，在编辑板 |  | 模型编写并保 |  |
|  |  | 中编写最终模 |  | 存到文件夹中 |  |
|  |  | 型代码，点击 |  |  |  |
|  |  | “保存模型” |  |  |  |
| 2 | 数据输入 | 按钮  系统正常运 | 14 | 系统成功将数 | 符合预期 |
|  |  | 行，点击上传 |  | 据上传到服务 |  |
|  |  | 按钮，选择数 |  | 器，界面中可 |  |
|  |  | 据 |  | 预览上传的数 |  |
| 3 | 数据填写 | 系统正常运 | 13 | 据  系统成功将数 | 符合预期 |
|  |  | 行，根据提示 |  | 据保存到文件 |  |
|  |  | 填写数据，并 |  | 夹中 |  |
|  |  | 点击“保存数 |  |  |  |
| 4 | 剂量推荐 | 据”按钮  系统正常运 | 9724 | 系统成功实现 | 符合预期 |
|  |  | 行，选择模 |  | 两种优化方 |  |
|  |  | 型、数据和两 |  | 法，并在界面 |  |

种剂量推荐方法，添加设 置，点击“启动优化”按钮

中显示剂量优化结果

* + - 1. 系统兼容性测试

系统兼容性测试主要指测试系统在不同操作系统平台上和不同浏览器中的能否正常的运行，防止出现网页界面未发生紊乱、错位等现象，以确保系统的页面布局具备稳定性。采用 RSelenium 包对其进行兼容性测试，主要查看在不同操作系统和浏览器中能否使用正常的功能。

测试系统能否在主流 Windows 和 Linux 操作系统环境下正常运行，具体的测试内容如表 4-3 所示。

表 4-3 操作系统测试表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 操作系统 | 具体操作 | 预期结果 | 实际结果 |
| 1 | Windows | 安装 Windows 11 操作系统 | Shiny-PopPK 系统在此系统中正常  运行 | 符合预期 |

2 Linux 安装 Ubuntu

22.04 LTS 操作系统

Shiny-PopPK 系统在此系统中正常运行

符合预期

测试系统能否在主流浏览器环境下正常运行，具体的测试内容如表 4-4 所示。

表 4-4 浏览器测试表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 浏览器 | 具体操作 | 预期结果 | 实际结果 |
| 1 | 谷歌浏览器 | 安装谷歌浏览器 | Shiny-PopPK 系统 | 符合预期 |
|  |  | （版本 | 在该浏览器上可 |  |
| 2 | Microsoft Edge 浏 | 83.0.4103.106）  安装 Microsoft | 正常运行  Shiny-PopPK 系统 | 符合预期 |
|  | 览器 | Edge 浏览器（版 | 在该浏览器上可 |  |
|  |  | 本 | 正常运行 |  |
| 3 | 火狐浏览器 | v119.0.2151.58）  安装火狐浏览器 | Shiny-PopPK 系统 | 符合预期 |
|  |  | （版本 | 在该浏览器上可 |  |
| 4 | 猎豹浏览器 | v115.0.1.8587）  安装猎豹浏览器 | 正常运行  Shiny-PopPK 系统 | 符合预期 |
|  |  | （版本 | 在该浏览器上可 |  |
|  |  | 8.0.0.21681） | 正常运行 |  |
| 4.2 系统应用 |  |  |  |  |
| 4.2.1 实验数据 |  |  |  |  |

来自于 nlmixr2 包中用于测试性能的数据（examplomycin.csv），该数据集共有

40 个患者信息，共 680 条数据，其中男性和女性均为 20 名，平均 WT 为 65.6±

14.3kg（38~98kg）。给药方式为口服给药，且为多剂量给药，数据列名包含 ID、TIME、DV、WT、SEX、AMT、EVID，文件格式为 CSV。examplomycin.csv 数据集如表 4-5 所示。

表 4-5 examplomycin.csv 部分数据集

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ID | TIME | DV | WT | SEX | AMT | EVID |
| 1 | 0 | 0 | 52 | 0 | 1200 | 101 |
| 1 | 0.302 | 174.8 | 52 | 0 | 0 | 0 |
| 1 | 2.908 | 319.6 | 52 | 0 | 0 | 0 |
| 1 | 3.144 | 336.2 | 52 | 0 | 0 | 0 |
| 1 | 9.943 | 137.7 | 52 | 0 | 0 | 0 |
| 1 | 18.021 | 99 | 52 | 0 | 0 | 0 |
| 1 | 24 | 0 | 52 | 0 | 1200 | 101 |
| 1 | 48 | 0 | 52 | 0 | 1200 | 101 |
| 1 | 72 | 0 | 52 | 0 | 1200 | 101 |
| 1 | 96 | 0 | 52 | 0 | 1200 | 101 |
| 1 | 120 | 0 | 52 | 0 | 1200 | 101 |
| 1 | 144 | 0 | 52 | 0 | 1200 | 101 |
| 1 | 168 | 0 | 52 | 0 | 1200 | 101 |
| 1 | 168.009 | 342.6 | 52 | 0 | 0 | 0 |
| 1 | 169.948 | 721.4 | 52 | 0 | 0 | 0 |
| 1 | 177.614 | 477.2 | 52 | 0 | 0 | 0 |
| 1 | 188.3 | 417.5 | 52 | 0 | 0 | 0 |

* + 1. 实例应用
       1. 对于新手用户和非专业人士刚使用本系统时，为了能够快速掌握使用方法和技巧，用户可以点击“Help”按钮，点击后，会弹出帮助窗口，在帮助窗口中包含了本系统可实现的所有功能及特点、快速入门的整体使用流程，每一功能模块的使用方法、理论知识及常见结果的分析；另外，还设置常见问题的解答,帮助窗口如图 4-1 所示。



图 4-1 帮助窗口示意图

* + - 1. 点击“请输入数据”按钮，在本地选择上传文件的数据，在界面中可预览上传的数据，然后进行数据映射，将列名转换为本系统可以正确识别的数据，目的是为了后续的建模做准备，最后点击“确定上传”按钮，数据将自动上传到PKdata 文件夹中。数据上传如图 4-2 所示。

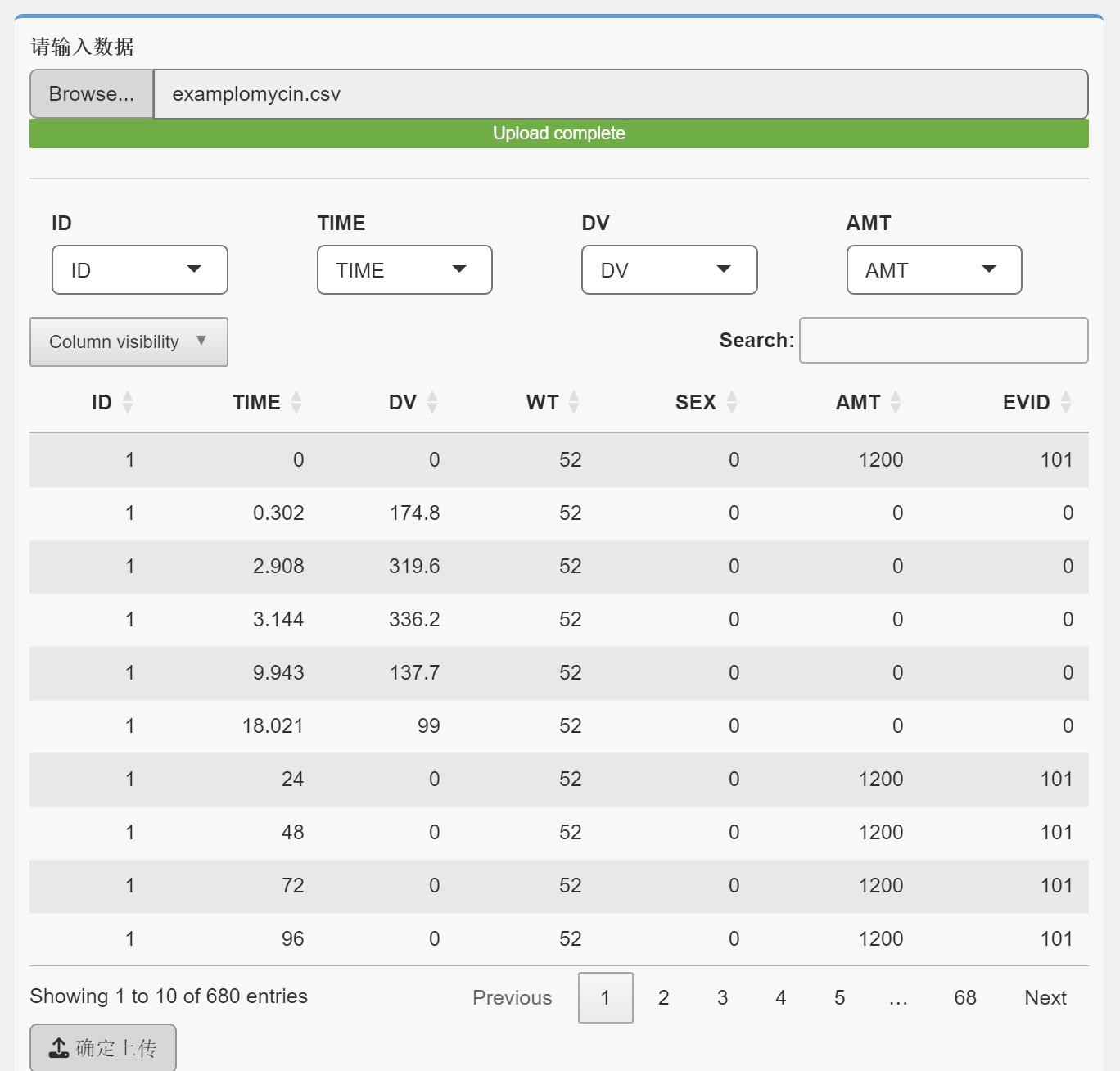
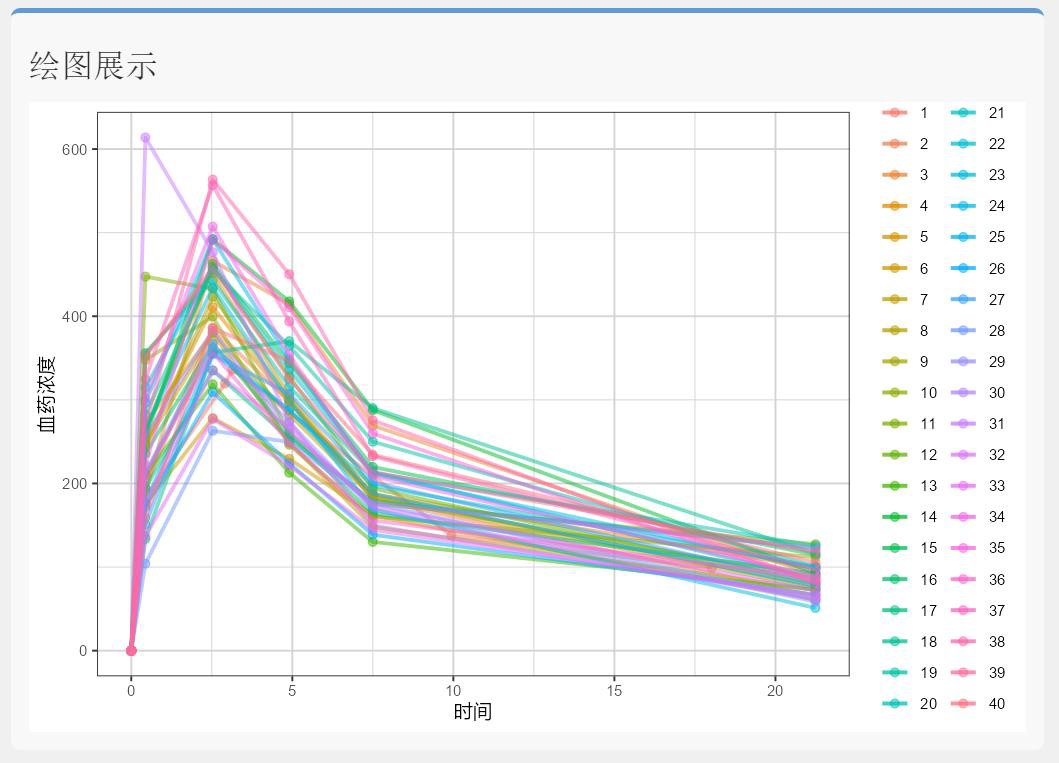


图 4-2 数据上传示意图

* + - 1. 接下来对数据进行探索，首先是对数据绘制药时曲线图，本数据集为多剂量给药，绘制出图会有多个峰，为了更好研究其曲线，提前利用 Microsoft Excel 软件将单剂量给药的数据提取出来可转换为单剂量，X 轴选择 TIME，Y 轴选择 DV，选择 ID 为组，此处还可以选择其它选项；对于置信区间和线条透明度以及各轴的标签如若要求不高，均可采用默认值；Y 轴数值采用线性形式，也可采用对数形式则可比较容易区分高低浓度点，识别异常值，也更方便观察药动学特征；本系统还支持单位转换功能，还可将每个个体的药时曲线单独绘制，添加好设置后，点击“绘图模拟”，总体药时曲线图和各个体药时曲线图如图 4-3，4-4 所示。



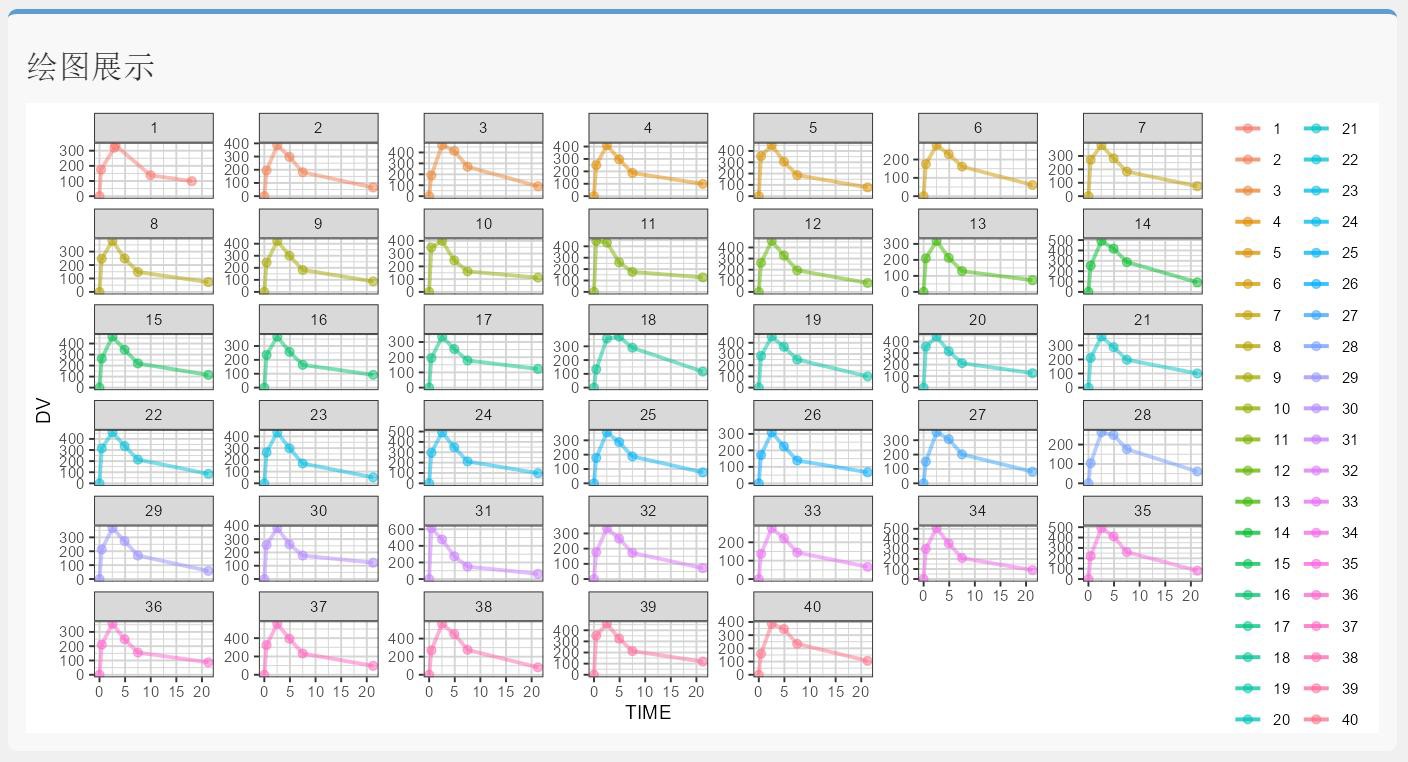
图 4-3 总体药时曲线图

图 4-4 各个体药时曲线图

观察图 4-3 观察吸收相曲线（峰浓度值以前的血药浓度点）为一条直线；再观察消除相（峰浓度值以后的血药浓度点）曲线不是一条直线，而是下降速率发生了变化，中间有明显的一个拐点。所以可以初步得到：该数据符合二室模型的概率较大。在选择结构模型时可以尝试一室和二室模型进行建模。

* + - 1. 接着对数据进行 NCA 计算，NCA 计算可用于估算药物的药代动力学参数，也可用于早期药物开发和生物等效研究。因该软件的 NCA 功能仅支持单剂量给药，所以

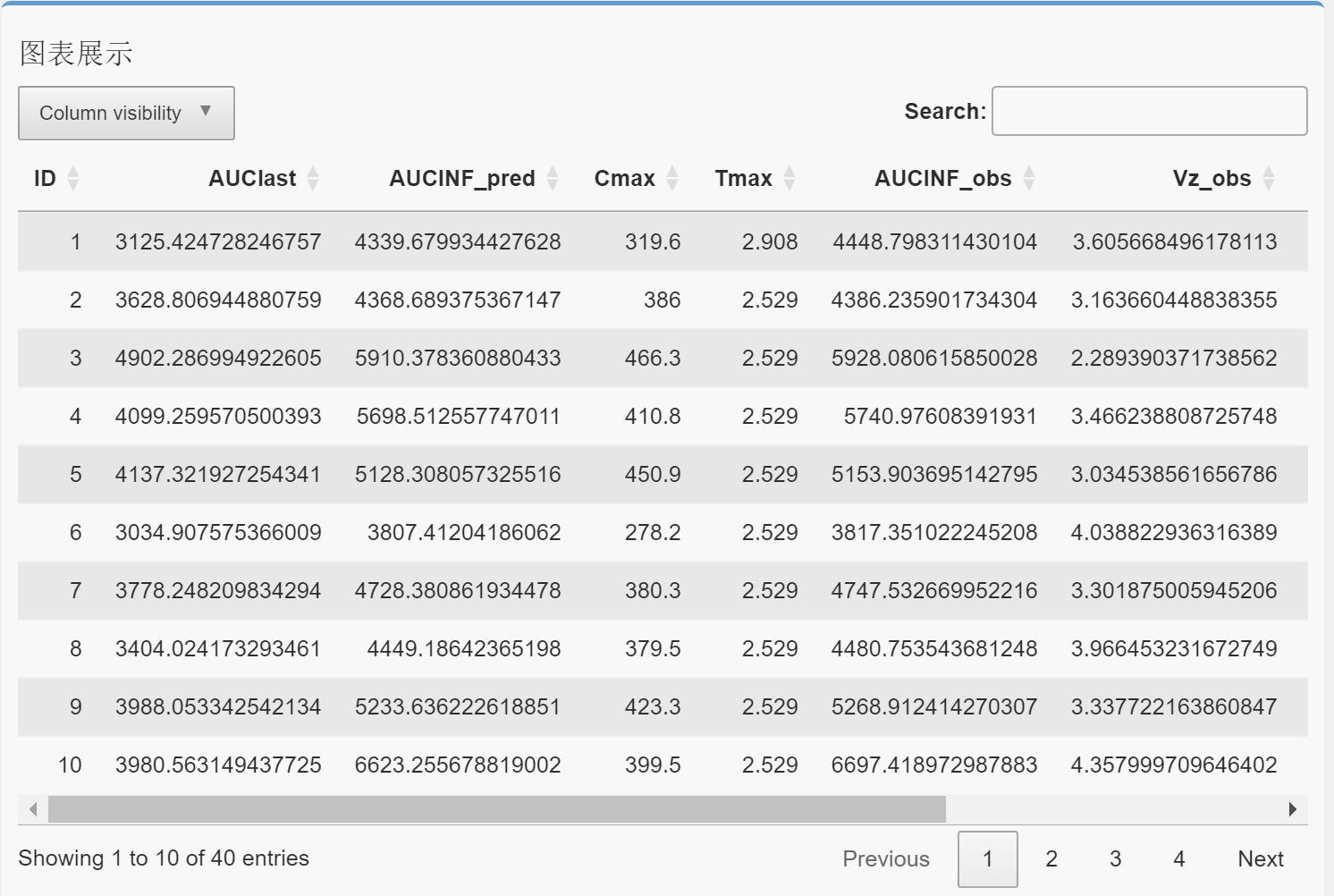
也需要提前利用 Microsoft Excel 软件将单剂量给药的数据提取出来，因浓度、时间和剂量单位在该数据中并未标注，设置条件可以忽略，只需要设置好口服给药的方式即可， 然后点击“NCA 计算”按钮，部分重要的参数就在界面中显示出来了，还可以通过点击“下载完整 NCA 报告”按钮获得完整的参数报告，报告会自动以“PopPK-NCAReport- 年-月-日 时：分：秒”的名称保存。NCA 计算结果如图 4-5 所示。

图 4-5 NCA 计算结果示意图

观察图 4-5NCA 计算出来的参数结果，可以得知表观分布容积 V 的范围在 2-5 之间，清除率 Cl 的范围在 0.15-0.35 之间，这对于后续建模输入初始值具有参考作用。

* + - 1. 选择基础模型进行建模，需要根据数据的特征来选择合适的模型，根据药时曲线可知，该数据可能符合一室模型或者二室模型，因此本文可采用的结构模型：口服给药、线性消除、一级吸收的一室模型和口服给药、线性消除、一级吸收的二室模型； 统计学模型：个体间变异为指数型，残差变异为加和型和比例型，共有四种模型，模型名称分别为“run1add”、“run1prop”、“run2add”、“run2prop” ，然后点击“加载模型”和“参数添加”按钮，可根据 NCA 分析结果添加初步的参数值，最后再进行保存。将这些模型进行比较，从其选择最佳模型，模型“run2add”参数如图 4-5 所示。

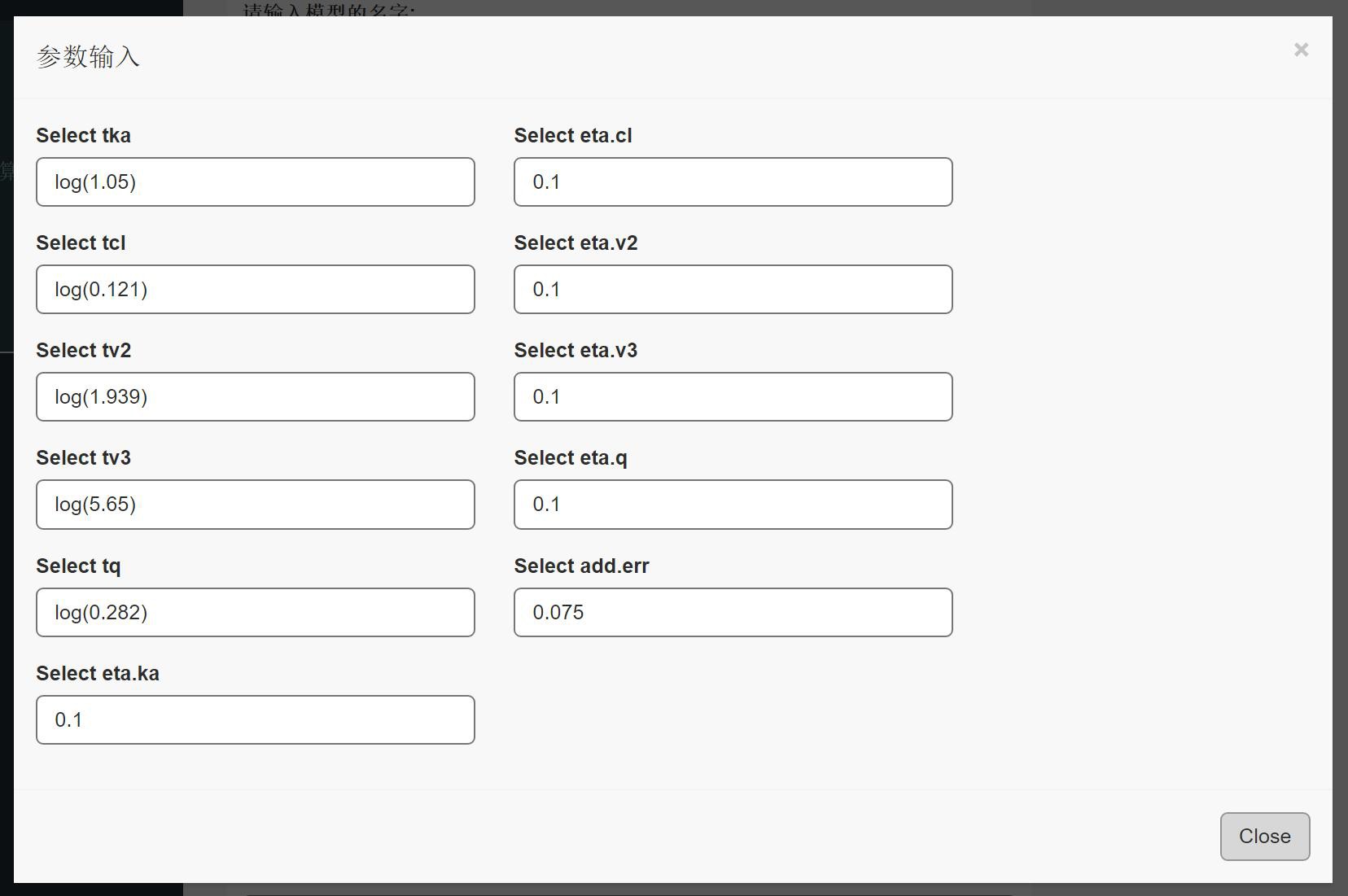


图 4-5 参数示意图

* + - 1. 模型加载完成后，然后计算目标函数值，这四种模型都采用先进的“saem” 算法，特别适合大数据集，也为了看一下其它算法的准确率，本文选择“run1add”和“run2add”两种模型使用“focei”算法。其运行过程和结果如图 4-6,4-7 所示。

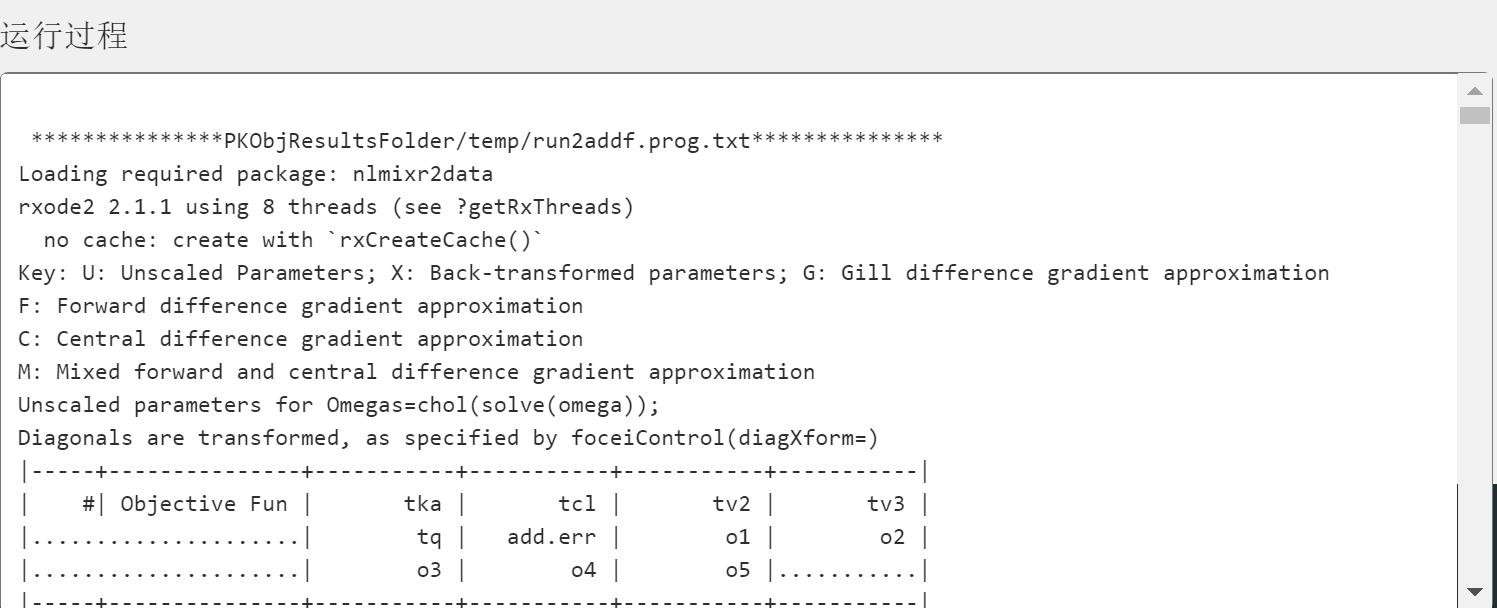


图 4-6 运行过程示意图

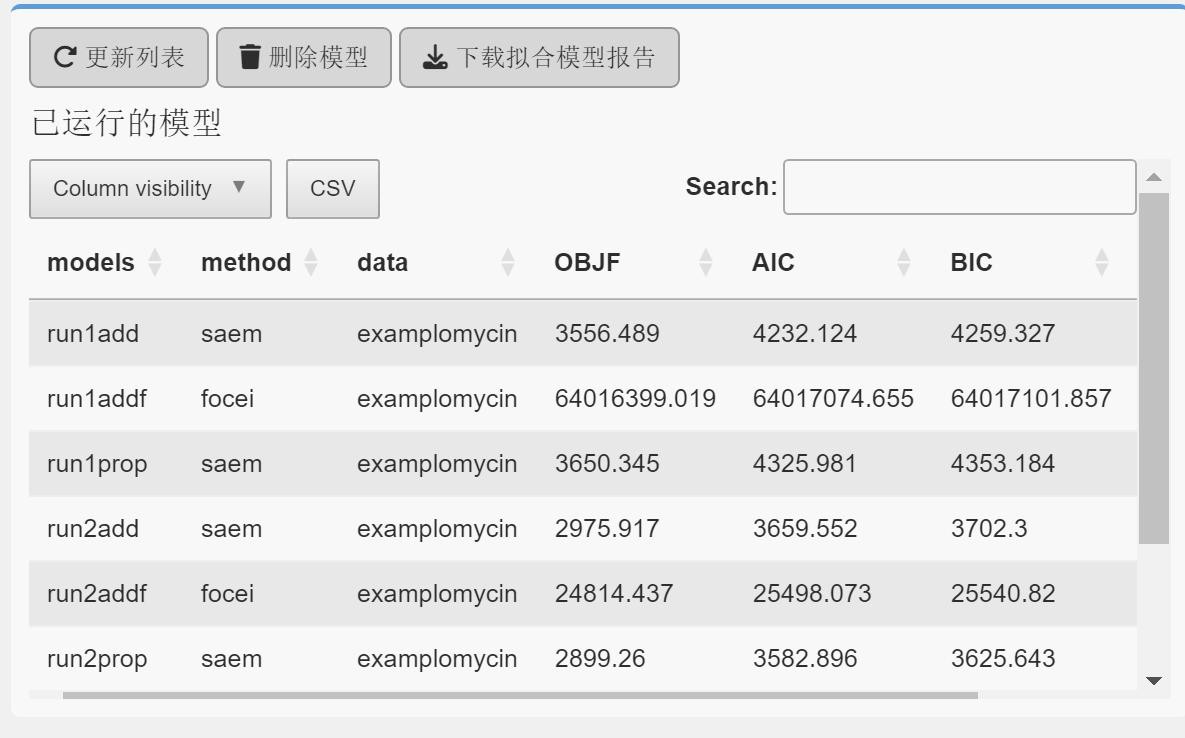


图 4-7 结果示意图

观察图 4-7 可以得知采用“saem”算法的四种模型和采用“focei”算法的两种模型，其 OFV 值、AIC 值和 BIC 值三者最小的是模型“run2prop”，因此，最终选择“run2prop”将其作为 PPK 基础模型。

* + - 1. 将模型“run2prop”作为基础模型后，然后再进行协变量筛选，该数据中的协变量主要是 WT 和 SEX，其中 WT 属于连续协变量，SEX 属于分类协变量，模型的参数有 ka、cl、v2、v3 和 q，因涉及的参数过多，本文只采用 ka，cl，v2 作为演示，将前向纳入和后向剔除的 P 值分别设置为 0.05 和 0.01，点击“启动筛选”按钮和“更新结果” 按钮，筛选结果就会在界面中显示出来，其筛选选项和筛选结果如图 4-8 和 4-9 所示。



图 4-8 筛选选项示意图

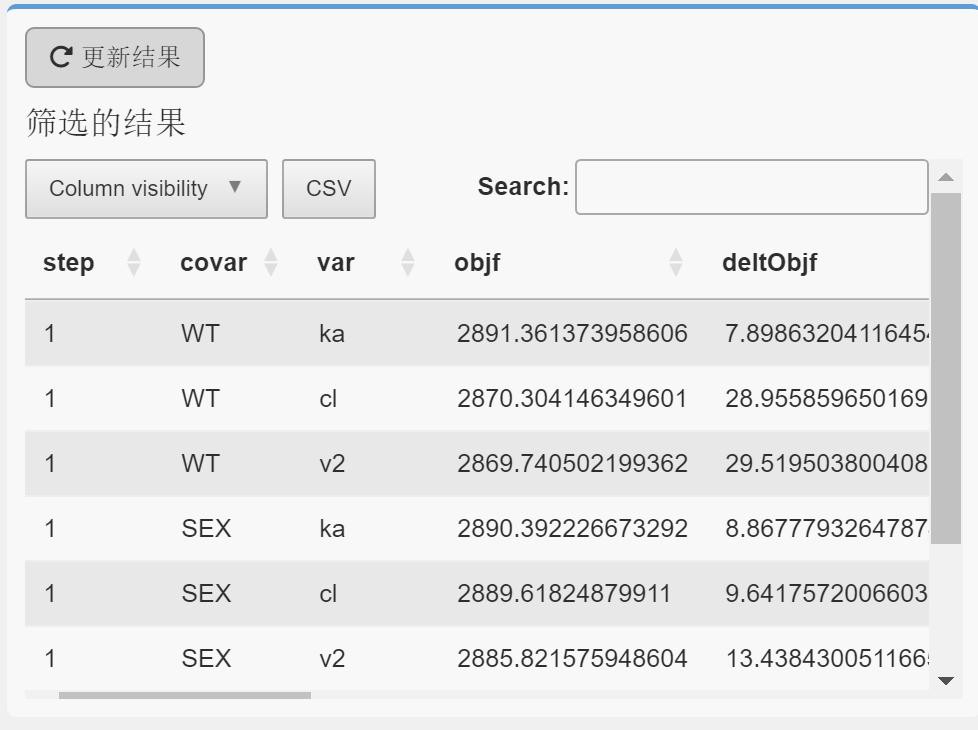


图 4-9 筛选结果示意图

由图 4-9 可知，因图片大小有限，最终协变量 SEX 对参数 v2 存在影响，协变量WT 对参数 cl 存在影响，将协变量 SEX 和 WT 分别纳入到参数 ka 和 cl 的模型公式中。

* + - 1. 筛选出协变量后，将协变量纳入到模型中，然后进行参数评估，本文采用第一种协变量添加方式---“选择添加协变量”（按内部固定公式法进行添加），将 SEX 纳入到参数 v2 模型公式中 ，WT 纳入到参数 cl 模型公式中，然后运行模型，其协变量添加过程和评估结果如图 4-10、4-11 所示。



图 4-10 协变量添加过程示意图

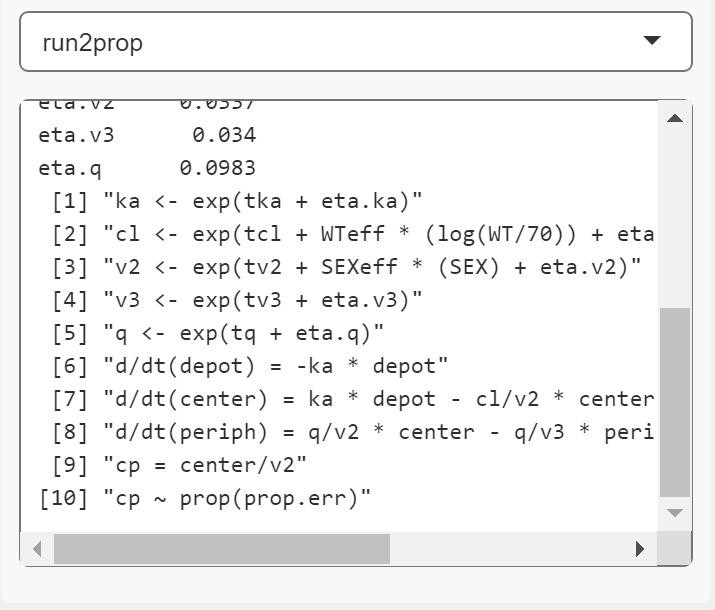


图 4-11 评估结果示意图

由图 4-11 可知模型的各个参数的评估值，下方的模型公式因页面有限未显示出来。将估算出的参数带入模型公式之中，便能得到最终模型。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。最终模型如表 4-6 所示。

表 4-6 最终模型公式

**参数 模型公式**

ka ka  exp(tka  eta.ka)  1.25

cl  exp(tcl  WTeff \* (log(WT/70)) + eta.cl)

cl  0.138 \*（ WT） 70

v2  exp(tv2  SEXeff \* (SEX) + eta.v2)

v2  2.16 \* e-0.2\*(SEX)(if SEX1)

v3 v3  exp(tv3 + eta.v3)  5.16

q q  exp(tq + eta.q)  0.32

* + - 1. 模型建好后需要采用模型诊断图来评估模型好坏，GOFs 和个体拟合图如图 4-

12 和 4-13 所示。

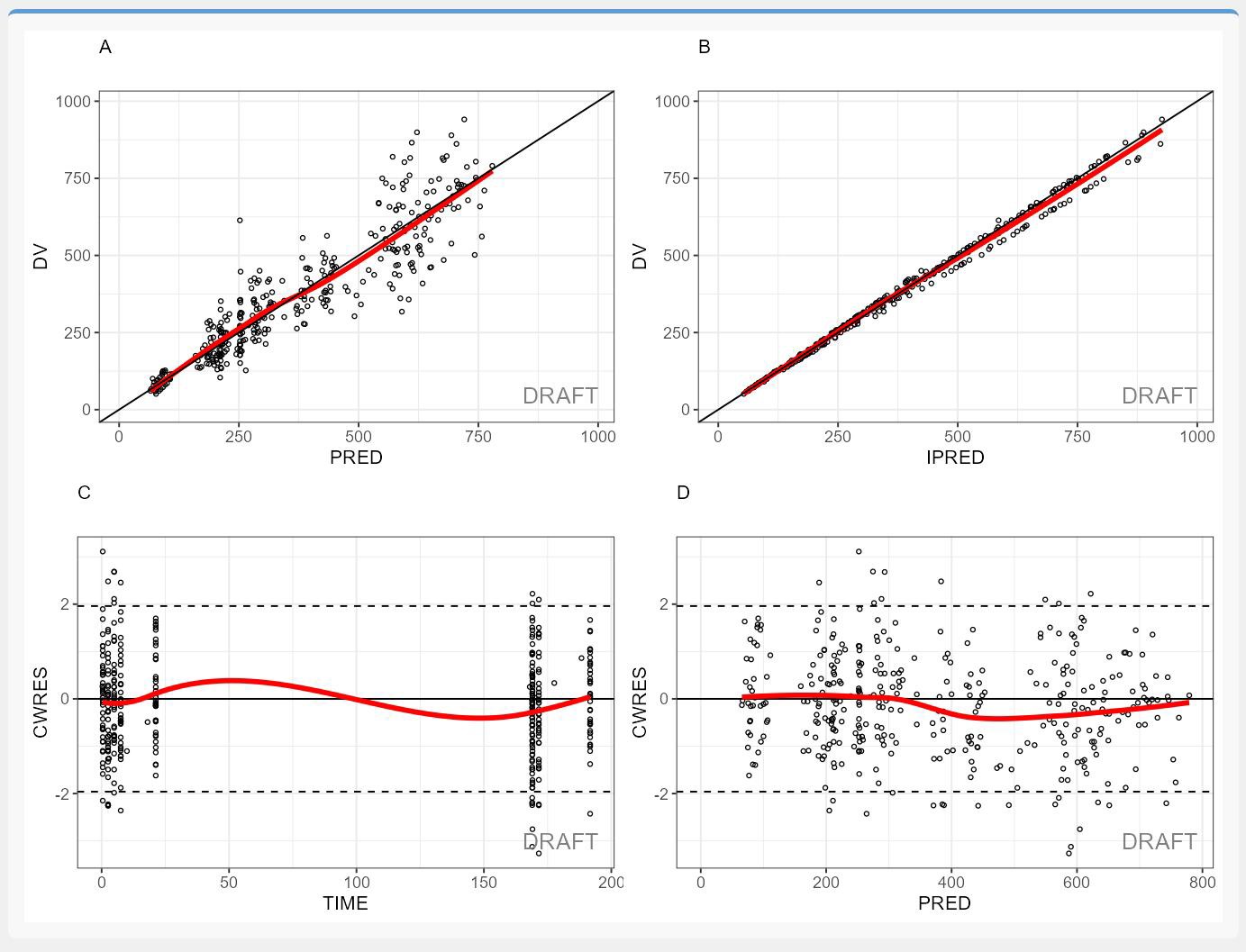


图 4-12 拟合优度图示意图

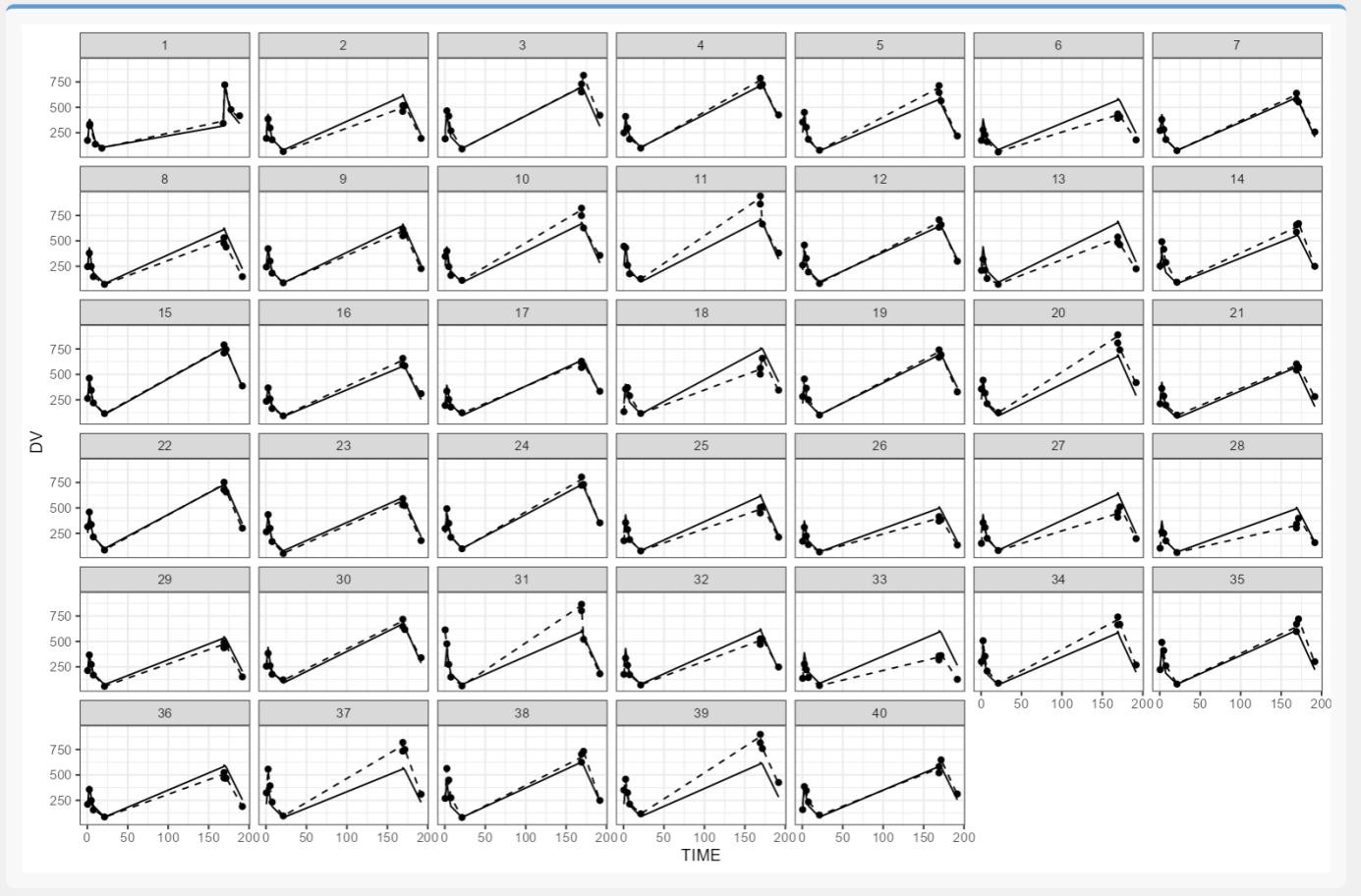


图 4-13 个体拟合图示意图

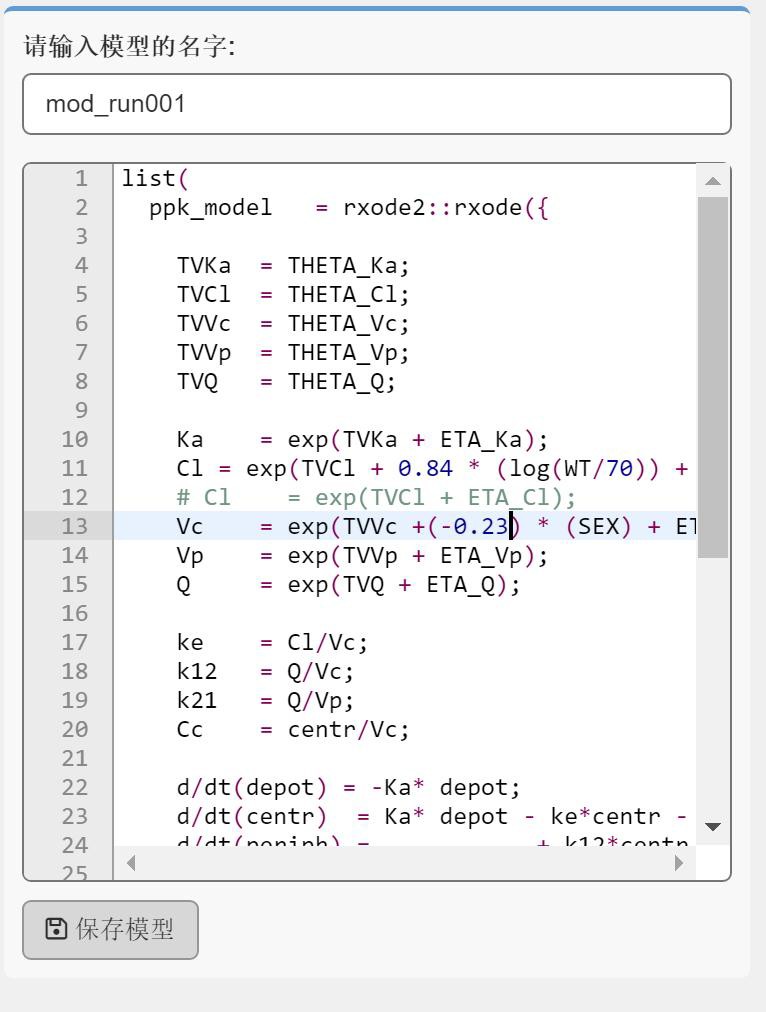
观察图 4-12 中的 A 和 B 可知， 因变量- 条件群体预测值（Dependent variable versus conditional population predictions, DV-CPRED）和因变量-个体预测值都均匀分布在 y=x 参考线两侧，这表明模型有良好的拟合性。再观察图 4-13 个体拟合图，图中的虚线代表群体预测值，实线代表个体预测值，由图可知大多数的个体拟合也较好，存在极少数个体拟合略差的情况，这可能与选取的初始值有关，因本文所使用的是模拟数据，所以无法通过查阅大量文献得到更为准确的初始值。

由 C 和 D 可知，加权残差-条件群体预测值（Conditional weighted residual versus conditional population prediction, CWRES-CPRED ） 散点图和条件加权残差- 时间

（CWRES-TIME）散点图均能分布在 y=±2 以内,这也表明模型拟合状况良好。

* + - 1. 群体模型建好之后，就可以进行剂量推荐，首先输入 PPK 模型，此时的 PPK 模型来源于上面所建立的最终 PPK 模型，将模型公式和评估的参数按照 posologyr 程序包的模型规范来写，具体的实际操作部分已放在 Help 指南中。然后输入患者数据，因为涉及到协变量，所以患者数据只能通过上传文件的方式输入。模型的输入如图 4-14 所

示，患者数据如表 4-7 所示。



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | 图 4-14 输入模型示意图  表 4-7 患者数据 |  | | |
| ID | TIME | DV | AMT | EVID | WT | SEX |
| 1 | 0 | NA | 2000 | 1 | 70 | 1 |
| 1 | 1 | 25 | 0 | 0 | 70 | 1 |
| 1 | 14 | 5.5 | 0 | 0 | 70 | 1 |

* + - 1. 最后采用剂量推荐的两种方法进行剂量优化，对于第一种方法估计选定目标区域的最佳剂量（ 目标 AUC 法）， 目的是调整剂量， 以使个体患者的药物暴露量

（AUC，曲线下面积）达到预定的目标值。将计算 AUC 开始的时间设置为 0h，终止时间设置为 12h，给药时间点在 12h,给药间隔设置为 12h,给药次数设置为 2 次（0h 和 12h 分别给药一次），输注时间设置为 0（即非静脉滴注方式），起始剂量为 100mg，目标AUC 为 400，点击“剂量优化运行”按钮，结果得到的优化剂量为 826.3722mg，结果如图 4-15 所示。



图 4-15 估计选定目标区域的最佳剂量计算结果示意图

采用第二种剂量优化方法估计选定目标浓度的最佳剂量（目标浓度法），目的是调整剂量，以使个体患者的稳态药物浓度达到预定的目标值。将达到目标 AUC 的时间点设置为 12h，给药时间点为 12h，给药间隔为 12h，给药次数为 2，输注持续时间为 0，目标 AUC 为 80，起始剂量为 100mg。点击“剂量优化运行”按钮，最终得到的结果为 26085.77mg，结果如图 4-16 所示。



图 4-16 估计所选目标浓度的最佳剂量计算结果示意图

综上所述，通过对 examplomycin.csv 的数据集进行全过程的实例应用，分别完成了 PPK 模型的建立和模型的应用（剂量推荐），表明了 Shiny-PopPK 交互式系统在数据探索、非房室分析、PPK 建模以及剂量推荐等功能方面能够正常运行。

* + - 1. 有多篇文章表明 nlmixr2 包建模的准确率可以和 Monolix 软件等一些商业软件差不多，有些地方还可能超过这些商业软件[63,85]。但为了进一步验证 Shiny-PopPK 交互式系统建立的 PPK 模型的可靠性，遵循严谨的态度，选择使用六种不同的数据集采用相同算法与模型进行建模，对本系统与 Monolix 软件（2021R1 版）得到的计算结果进行对比。主要对比了估算的 PPK 参数和个体 PK 参数。对于估算的 PPK 参数直接进行了比较，并对其结果进行 t 检验；对于估算的个体 PK 参数通过线性回归的方法进行了比较，其结果如表 4-8、4-9 和如图 4-17(因论文篇幅有限，此处仅展示数据集 1 的个体 PK 参数估计值的回归图，其它数据集见附录 4)所示。

表 4-8 Shiny-PopPK 与 Monolix 的 PPK 参数估计值对比

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 参数 | 1S | 1M | 2S | 2M | 3S | 3M | 4S | 4M | 5S | 5M | 6S | 6M |
| ka | 1.12 | 1.15 | 0.09 | 0.09 | 0.30 | 0.32 | 0.20 | 0.20 | 0.52 | 0.51 | 0.28 | 0.29 |
| cl | 0.13 | 0.13 | 2.64 | 2.67 | 0.42 | 0.42 | 0.24 | 0.23 | 0.29 | 0.30 | 0.13 | 0.13 |
| v1 | 2.09 | 2.13 | 1.84 | 1.88 | 2.32 | 2.54 | 0.66 | 0.68 | 1.70 | 1.68 | 4.39 | 4.42 |
| v2 | 5.00 | 4.99 | 7.77 | 7.28 | 5.42 | 5.24 | 27.11 | 28.78 | 6.20 | 6.30 | 3.32 | 3.33 |

q 0.29 0.28 0.27 0.25 0.82 0.84 0.39 0.39 1.22 1.26 0.39 0.38

注释：1S:Shiny-PopPK 使用数据集 1 得到的参数估计值；2M：Monolix 使用数据集 2 得到的参数估计值,依次类推。

表 4-9 Shiny-PopPK 与 Monolix 的 PPK 参数估计值的 t 检验

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 参数 | 自由度 | t 值 | p 值 | 置信区间  (下限) | 置信区间  (上限) |
| ka | 5 | -1.46 | 0.20 | -0.02 | 0.01 |
| cl | 5 | -0.91 | 0.40 | -0.02 | 0.01 |
| v1 | 5 | -1.61 | 0.17 | -0.14 | 0.03 |
| v2 | 5 | -0.41 | 0.70 | -0.94 | 0.69 |
| q | 5 | -0.38 | 0.72 | -0.03 | 0.02 |

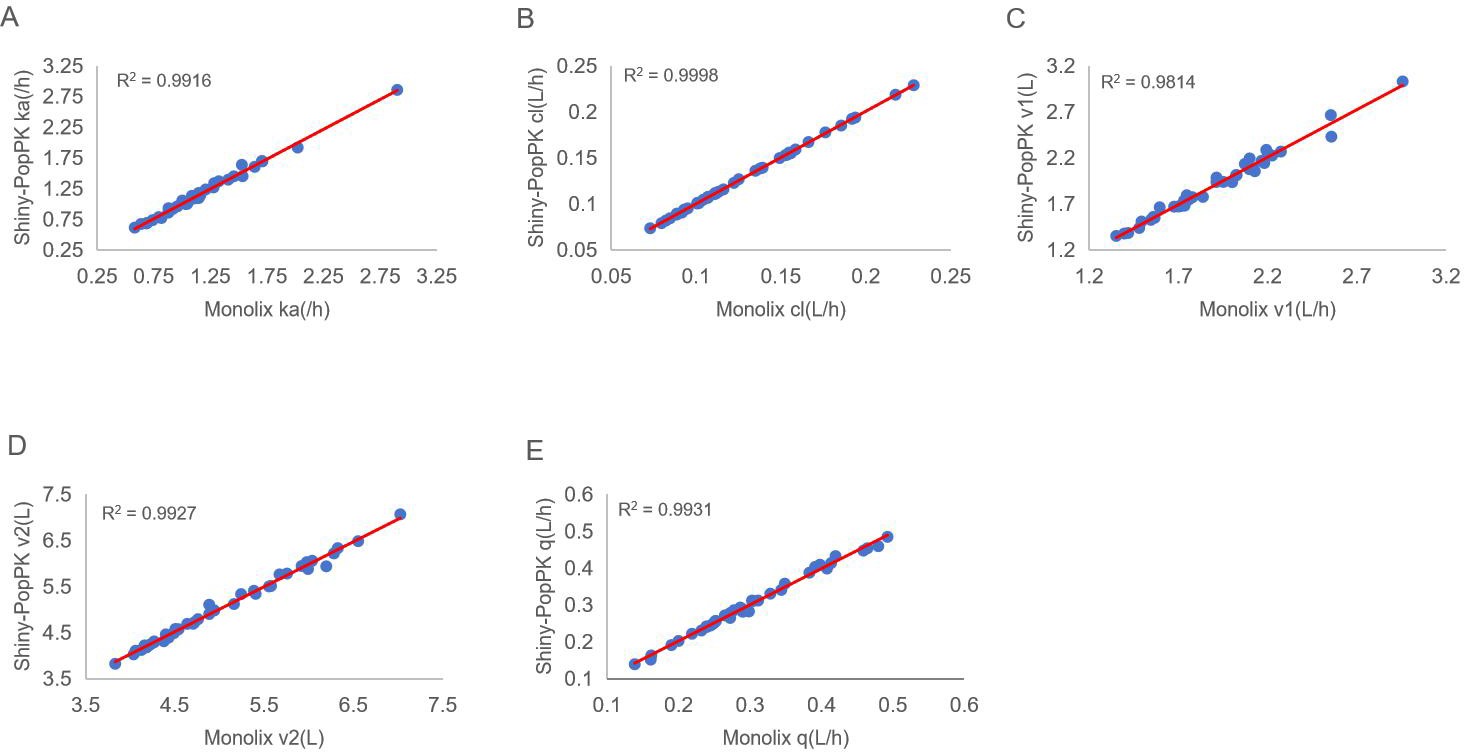


图 4-17 个体 PK 参数估计值的回归图

由表 4-8 和 4-9 可知，Shiny-PopPK 系统与 Monolix 软件在估算 PPK 参数时，计算得到的群体典型值都非常相近，说明 Shiny-PopPK 系统的 PPK 参数估算性能与Monolix 软件相差不大。由图 4-17 可知，A、B、C、D 和 E 图这四者的R 2 均>0.9， 说明两者的结果高度相关。总体上评估 Shiny-PopPK 系统的估算性能和 Monolix 软件比较接近。由以上结果进一步验证了 Shiny-PopPK 系统平台的可靠性。

# 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()})

1. 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

}

1. 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)}

1. 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)}

1. 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)})}

1. 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) cntrll <- trimws(readLines(get\_model\_list()[[mod]]$model))

tmpl <- readLines("./run\_nmx1.tmp")

rlst <- list(modelloc=normalizePath(get\_model\_list()[[mod]]$model,

winslash = "/",mustWork = FALSE),

dataloc=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)}

1. 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!")}}

1. 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)}

1. 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)

})

1. 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) tvm <- log(0.121) tkm <- log(0.23) tv <- log(3) eta.ka ~ 1

eta.vm ~ 1

eta.km ~ 1

eta.v ~ 1

add.err <- 0.1

})

model({

lagD <- exp(talag)

ka <- exp(tka + eta.ka) vm <- exp(tvm + eta.vm) km <- exp(tkm + eta.km) v <- exp(tv + eta.v) d/dt(depot) = -ka \* depot alag(depot) <- lagD

d/dt(center) = ka \* depot - (vm\*center)/(km + center) cp = center / v

cp ~ add(add.err)

})

1. 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) tvm <- log(0.121) tkm <- log(0.23) tv2 <- log(1.939) tv3 <- log(5.65) tq <- log(0.282) eta.ka ~ 0.1

eta.vm ~ 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 \* depot

d/dt(center) = ka \* depot - (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)

})

1. 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) tvm <- log(0.121) tkm <- log(0.23) tv2 <- log(1.939) tv3 <- log(5.65) tq <- log(0.282) eta.ka ~ 0.1

eta.vm ~ 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)

})

1. 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)

})

1. 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)

})

1. 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({

tvm <- log(0.121) tkm <- log(0.23) tv <- log(3) eta.vm ~ 0.1

eta.km ~ 0.1

eta.v ~ 1

add.err <- 0.1

})

model({

vm <- exp(tvm + eta.vm) km <- exp(tkm + eta.km) v <- exp(tv + eta.v)

d/dt(center) = - (vm\*center)/(km + center) cp = center / v

cp ~ add(add.err)

})

1. 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({

tvm <- log(0.121) tkm <- log(0.23) tv2 <- log(1.939) tv3 <- log(5.65) tq <- log(0.282) eta.vm ~ 0.1

eta.km ~ 0.1

eta.v2 ~ 0.1

eta.v3 ~ 0.1

eta.q ~ 0.1

prop.err <- 0.075

})

model({

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(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)

})

1. 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)

})

1. 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)

})

1. 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)

})

1. 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)

})

1. 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)

})

1. 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({

tvm <- log(0.121) tkm <- log(0.23) tv2 <- log(1.939) tv3 <- log(5.65) tq <- log(0.282) eta.vm ~ 0.1

eta.km ~ 0.1

eta.v2 ~ 0.1

eta.v3 ~ 0.1

eta.q ~ 0.1

add.err <- 0.1

prop.err <- 0.075

})

model({

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) k0 <- rate

d/dt(center) = k0 - (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) + prop(prop.err)

})

1. 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)

})