Package 'plethem'

November 4, 2020

Title Population Life Course Exposure to Health Effects Modeling

Version 1.1.0

Description Functions, data and user interfaces for performing physiologically based pharmacokinetic ('PBPK') modeling, in-vitro-to-in-vivo Extrapolation ('IVIVE') and exposure estimation. Also contains user interfaces to run models from the 'httk' package. Taken together these provide an easy to use and powerful modeling tool that can be used for all steps along the source-to-outcome continuum.All the analysis tools in the package are run as interactive applications. Check package help for more information. Refer to the manuscript 'Population Life-course exposure to health effects model (PLETHEM): An R package for PBPK modeling' <doi: 10.1016/j.comtox.2019.100115> for more information on the models and algorithms used in the package. More information on PBPK modeling itself can be found in the book 'Physiologically Based Pharmacokinetic Modeling: Science and Applications' by Reddy et al <doi:10.1002/0471478768>.

Depends R (>= 3.6.0)

License MIT + file LICENSE

URL https://github.com/ScitoVation/plethem/

BugReports https://github.com/ScitoVation/plethem/issues/

Encoding UTF-8

LazyData true

Imports shiny, ggplot2, shinyBS, shinyWidgets, miniUI, shinydashboard, rstudioapi, shinythemes, rhandsontable, deSolve, truncdist, reshape2, dplyr, readxl, gdata, devtools, RSQLite, formatR, plotly, magrittr, sqldf, utils, stats, DT, shinyjs, V8, httk, NonCompart, stringr, doParallel, foreach, pracma, shinybusy, data.table, tidyr, shinyalert, shinyFiles, officer, DiagrammeR, DiagrammeRsvg, rsvg, devEMF

Suggests testthat, shinytest, rmarkdown, knitr

RoxygenNote 7.1.1

VignetteBuilder knitr

NeedsCompilation yes

Author Salil Pendse [aut],
Kevin Bronson [aut],
Jeremy Fitzpatrick [aut, cre],
Patrick McMullen [ctb],
ScitoVation, LLC [cph]

Maintainer Jeremy Fitzpatrick <jfitzpatrick@scitovation.com>

Repository CRAN

Date/Publication 2020-11-04 15:50:07 UTC

R topics documented:

`addChemsToHTTK` *Function that adds chemical to httk package chem list.*

Description

internal function that interfaces with httk package

Usage

```
addChemsToHTTK()
```

Examples

```
if(interactive()){
addChemsToHTTK()
}
```

`addDataSet` *Server side function for the UI used to add external observation datasets to projects*

Description

The sever function for the add dataset module used in PLETHEM. This function interacts with the server function of the PLETHEM model. It saves the dataset imported as a .RDS file in the project folder. It also adds the dataset reference to the project database for further use.

Usage

```
addDataSet(input, output, session, data_type)
```

Arguments

input	the input object from the add dataset module UI
output	the output object from the add dataset module UI
session	the shiny session information where the add dataset in called currently
data_type	the data_type returned by the module.

---

addDataSetUI *UI for adding external observation datasets to projects*

Description

The UI function for the add dataset module used in PLETHEM. This function creates the add dataset dialog box that is triggered by the modeling interface. This is a module function and the user will never need to call it

Usage

```
addDataSetUI(namespace, data_type)
```

Arguments

namespace	the namespace for this module
data_type	the type of data to be uploaded based on where the UI is called from

---

addParameters *Add user selected parameters to the PBPK reporting document. Should not be used by directly by the user*

Description

add user defined parameter tables

Usage

```
addParameters(HESI_doc, chemical, exposure, physiological)
```

Arguments

HESI_doc	This is the PBPK modeling document that the flowchart will be added too.
chemical	dataframe of chemical data added by the user
exposure	dataframe of exposure data added by the user
physiological	dataframe of physiological data added by the user

Value

HESI_doc, the same officer::read_docx object

`addPBPKequations` *Add PBPK equations to the PBPK reporting document. Should not be used by directly by the user*

Description

The function returns the PBPK reporting document with the PBPK modeling equations and starting variable values to it.

Usage

```
addPBPKequations(HESI_doc)
```

Arguments

`HESI_doc`	This is the PBPK modeling document that the flowchart will be added too.

Value

PBPK modeling document with equations and starting parameter values.

`biomDataExample` *NHANES Toluene Bio-monitoring Data Set For Testing The runReverseDosimetry Function*

Description

This dataset was taken from the Centers for Disease Control, National Health and Nutritional Examination Survey data. It has been significantly modified in its formatting and only contains data on toluene biomonitering for testing the runReverseDosimetry Function. The original data is availible through the NHANES reference. The modified data set is availible through the source reference.

Usage

```
data(biomDataExample)
```

Format

An object of class `data.frame` with 10000 rows and 1 columns.

Source

PLETHEM

References

NHANES

Examples

```
data(biomDataExample)
```

`calcMPCPPGL` *Calcuate MPPGL and CPPGL based on age*

Description

Calculates the Microsomal Protein per Gram liver (MPPGL) and Cytosolic Protein Per Gram Liver(CPPGL) for humans based on the age of the person in years. The equations for MPPGL and CPPGL were developed internally at Scitovation

Usage

```
calcMPCPPGL(age)
```

Arguments

`age`	age of the human in years

Value

list containing the "MPPGL" and "CPPGL" values for the

`calculatePartitionCoefficients`
Calculate Partition Coefficient

Description

This function calculates the partition coefficients based on the qsar model selected for the given tissues. Currently only one QSAR model is supported by PLETHEM

Usage

```
calculatePartitionCoefficients(
  selected_qsar = "one",
  chem_params = NULL,
  tissue_list,
  selected_org = "human"
)
```

Arguments

selected_qsar	QSAR model to use for estimating partition coefficient
chem_params	A named list of chemical params. The list should contain the minimal number of parameters needed to run the QSAR model selected
tissue_list	List of tissues for which the partition coefficients need to be calculated. See vignette on Qsar based parameter estimation of more details
selected_org	Organism. Either "human" or "rat"

---

calc_cyp2b6 *Calculate cyp2b6 fraction*

Description

get expression for cyp2b6 as a fraction of adult

Usage

```
calc_cyp2b6(age)
```

Arguments

age	age of the human in years

---

clearProjectDb *Clear Project Db*

Description

This function clears the project Db. It is called internally when a new project is created. It is also used by developers to make a clean project db

Usage

```
clearProjectDb()
```

createEmptyUserDb *Creates a new userDb based on the empty database in the package*

Description

The function allows the users to create a new empty user database file that is needed to run plethem.

Usage

```
createEmptyUserDb(path = NULL)
```

Arguments

path	path to where the user database needs to be stored. Make sure you have write permission to this folder. If no path is provided, it launches a folder select dialog.

Examples

```
if(interactive()){
createEmptyUserDb()
createEmptyUserDb(tempdir())
}
```

createHESIgraphs *Pipe to add concentration-timecourse data to the report Should not be used by directly by the user*

Description

add graphs to the PBPK reporting document

Usage

```
createHESIgraphs(report_doc, context, conc_units)
```

Arguments

report_doc	an officer::read_docx object
context	a data.frame of concentrationXtime data, as returned by concData()
conc_units	a string denoting the units of concentration

Value

report_doc, the same officer::read_docx object

`createPBPKflowchart` *Creates the PBPK model flowchart. Should not be used by directly by the user*

Description

The function returns the PBPK model flowchart

Usage

```
createPBPKflowchart(HESI_doc)
```

Arguments

`HESI_doc`	This is the PBPK modeling document that the flowchart will be added too.

Value

The PBPK modeling document with the PBPK flowchart added to it.

`externDbSelect` *Runs all select queries an arbitrary database*

Description

The function runs the select queries issued to the user db and returns the dataframe the path to user database is stored in main plethem database and is selected from there

Usage

```
externDbSelect(query, db_path)
```

Arguments

`query`	A valid SQL Query
`db_path`	A valid path

`getAllCypData` *get data for human cyp fraction by age*

Description

The function returns the expression of cyps at each age as a fraction of the adult (25 years). The number needs to be multiplied by the cyp abundance information in the database to get the nominal expression value for the cyp.

Usage

```
getAllCypData(age)
```

Arguments

`age`	the age of the person in years

Value

dataframe containing the fractional expression for all the cyps

---

`getAllParamValuesForModel` *Gets all the parameter values for the model. This function should not be used by the model*

Description

Get all the parameter values that are required for the model to run. The values are obtained from the Project database. Only those values that are used in the model as determined by the master database are returned by the function

Usage

```
getAllParamValuesForModel(simid, model)
```

Arguments

`simid`	Integer The id for simulation selected to run
`model`	Character The string identifying the model to be run

Value

list List that can be passed to the solver as model params

`getAllSetChoices` *get all set names for a given paramter set*

Description

This function returns all the sets of a given set type from the current project database This is used internally to update drop downs or to get simulation choices

Usage

```
getAllSetChoices(set_type = "physio")
```

Arguments

`set_type`	The type of set can be "physio", "chem","expo","metab" or "sim"

Value

named list of set names

`getAllVariabilityValuesForModel` *Gets all the variability values for the model. This data returned by the function is not meant to be understandable by the user*

Description

Get all the variability values required for creating paramter sets for montecarlo analysis. The values are obtained from the Project database.

Usage

```
getAllVariabilityValuesForModel(simid, params, mc_num)
```

Arguments

`simid`	Integer The id for simulation selected to run
`params`	list of model parameters
`mc_num`	number of montecarlo runs

Value

matrix of parameters that will be used for individual montecarlo runs

getDbConn *Gets the connection to the Db to run all the queries against*

Description

The function returns the connection object to the database passed in DbPath

Usage

```
getDbConn(db_path)
```

Arguments

db_path	The location of the project database. This function will not be called by the user directly

getFileFolderPath *Show dialogs to select files or folders*

Description

The function shows the dialog to select files or folders. The functions change depending on the OS in which RStudio is running. It is only called internally and should not be run by the user.

Usage

```
getFileFolderPath(type = "dir", caption = "", extension = "", new_flag = FALSE)
```

Arguments

type	Type of document to get a path for. dir for directory or file for file
caption	Caption to display for choose file/directory modal
extension	extensions to display for selecting file type
new_flag	Logical value for the "new" parameter in file.choose. Only used on MacOS

Value

path to the selected file or directory

getLifecourseBodyHeight

Get average body height using the life course equation

Description

The function is used to calculate the average body height for humans based on age and gender. The function uses life course equations developed by Scitovation.

Usage

```
getLifecourseBodyHeight(age, gender)
```

Arguments

age	age in years
gender	Either "M" for male or "F" for female

Value

Body Height

getLifecourseBodyHeightMale

Calculate body height of average human male using the lifecourse equation

Description

This function is not called directly by the user.

Usage

```
getLifecourseBodyHeightMale(age)
```

Arguments

age	age in years

Value

Body height

getLifecourseBodyWeight

Get average body weight using the life course equation

Description

The function is used to calculate the average body weight in kgs for humans based on age and gender. The function uses life course equations developed by Scitovation.

Usage

```
getLifecourseBodyWeight(age, gender)
```

Arguments

age	age in years
gender	Either "M" for male or "F" for female

Value

Body Weight in kgs

getLifecourseBodyWeightFemale

Calculate body weight in kg of average human female using the life-course equation

Description

This function is not called directly by the user. See `getLifeCourseBodyWeight`

Usage

```
getLifecourseBodyWeightFemale(age)
```

Arguments

age	age in years

Value

Body weight in kg

getLifecourseBodyWeightMale

Calculate body weight in kg of average human male using the life-course equation

Description

This function is not called directly by the user. See `getLifeCourseBodyWeight`

Usage

```
getLifecourseBodyWeightMale(age)
```

Arguments

age	age in years

Value

Body weight in kg

getLifecourseCardiacOutput

Get the cardiac output based on the life course equations

Description

Get the cardiac output for a specific age and gender. Uses either Bosgra or ScitoVation equations

Usage

```
getLifecourseCardiacOutput(age, gender, source = "sciv", qc_var = 0)
```

Arguments

age	Age in years
gender	Gender can be either "M" for male or "F" for female
source	Source of the equations. Either from Bosgra et al. "bosgra" or from ScitoVation publications as "sciv"
qc_var	variability for cardiac output. Used to account for exercise level. Can only be used when source = "sciv"

`getLifecourseGlomerularFiltrationRate` *Get the Glomerular Filteration Rate for a given age and gender*

Description

Uses life course equation to calculate a GFR for a specific age and gender

Usage

```
getLifecourseGlomerularFiltrationRate(age, gender)
```

Arguments

`age`	Age in years
`gender`	Either "M" for Male for "F" for Female

`getLifecourseLungDeadSpace` *Get the Lung Dead Space for a given age and gender*

Description

Uses life course equation to calculate the lung dead space for a specific age and gender

Usage

```
getLifecourseLungDeadSpace(age, gender)
```

Arguments

`age`	Age in years
`gender`	Either "M" for Male for "F" for Female

getLifecourseTidalVolume

Get the tidal volume for a given age and gender

Description

Uses life course equation to calculate the tidal volume for a specific age and gender

Usage

```
getLifecourseTidalVolume(age, gender, activity = "rest")
```

Arguments

age	Age in years
gender	Either "M" for Male for "F" for Female
activity	Activity level. Can be "rest" or "light activity"

getLifecourseTissuePerfusion

Get perfusion for tissues provided on the basis of age and gender

Description

Get perfusion for tissues provided on the basis of age and gender

Usage

```
getLifecourseTissuePerfusion(
  age = 25,
  gender = "M",
  tissues = list(),
  source = "sciv"
)
```

Arguments

age	Age of the organism in years
gender	The gender of the organism "M" for Male or "F" for Female
tissues	List of tissues for which blood perfusion is needed. The tissues can be any of the following liver, fat, bone, brain, gonad, heart, intestine, kidney, lung, pancreas, skin, spleen, stomach, thymus, muscle, remaining.
source	source of the equation. defaults to "sciv" for scitovation

Value

list containing blood perfusion of tissues for the given age and gender.

`getLifecourseTissueVolumes` *Get volumes for tissues provided on the basis of age and gender*

Description

Get volumes for tissues provided on the basis of age and gender

Usage

```
getLifecourseTissueVolumes(
  age = 25,
  gender = "M",
  perf_frct = 0.85,
  tissues = list()
)
```

Arguments

`age`	Age of the organism in years
`gender`	The gender of the organism "M" for Male or "F" for Female
`perf_frct`	fraction of perfused tissue. default to 0.85
`tissues`	List of tissues for which the volumes are needed. The tissues can be one of the following liver, blood, fat, bone, brain, gonad, heart, intestine, kidney, lung, pancreas, skin, spleen, stomach, thymus, remaining, muscle

Value

list containing volumes for age, gender and tissues.

`getLifecourseUrineProductionRate` *Get the urine production rate for a given age and gender*

Description

Uses life course equation to calculate a urine production rate for a specific age and gender

Usage

```
getLifecourseUrineProductionRate(age, gender)
```

Arguments

age	Age in years
gender	Either "M" for Male for "F" for Female

---

getLifecourseVentilationRate *Get the ventilation rate for a given age and gender*

Description

Uses life course equation to calculate a ventilation rate for a specific age and gender

Usage

```
getLifecourseVentilationRate(age, gender, activity = "rest", source = "sciv")
```

Arguments

age	Age in years
gender	Either "M" for Male for "F" for Female
activity	Activity level. Can be "rest" or "light exercise"
source	source of equations either "sciv" or "bosgra"

---

getMetabData *Gets the metabolism data. Should not be used by directly by the user*

Description

The function returns the relavent metabolism data if the simulation contains data from the metabolism set

Usage

```
getMetabData(admeid, model = "rapidPBPK")
```

Arguments

admeid	The id for ADME set. The admeid is used to obtain information about the other sets.
model	Model name

Value

List containing the metabolism values needed to run PBPK model or display simulation information

`getNextID` *Get the next Id for the table entry*

Description

Gets the next valid id for a given table. This function is used internally to decide what id number should be used to save a parameter set

Usage

```
getNextID(tble_name, db_path = "")
```

Arguments

`tble_name`	Name of the table for which the ID is required
`db_path`	The location of the project database. This defaults to database/project.sqlite and is not expected to change

Value

int Integer ID of the next row in the table.

`getObservationSetChoices` *Get all observation sets*

Description

Get all the sets associated with observation in a given projects. Observations need to be handled differently from the other set types since they can themselves be of multiple types

Usage

```
getObservationSetChoices(obs_type)
```

Arguments

`obs_type`	type of observation to return, can be "cl" or "conc" for clearance and concentration data

Value

named list of all sets of the obs_type

getParameterSet *Get the values for parameters in a given set*

Description

Get all the parameter values for a given dataset and id

Usage

```
getParameterSet(set_type = "physio", id = 1)
```

Arguments

set_type	Either "physio","chem"or "expo"
id	integer id for the required set

getProjectChemicalList *get the list of chemicals currently in the project database*

Description

Gets the chemicals that are currently a part of the project. This list comes from the chemical table in the project database. This function returns the chemical properties needed by all the models within PLETHEM. Chemical information specific to the model currently used is requested through a different function. This function can be used directly by the user to get a list of chemicals in the current project.

Usage

```
getProjectChemicalList()
```

Value

Returns a chemical list containing the names(chem_name), CAS numbers(CAS), Molecular Weight(MW),KM and Fraction unbound Unbound in Plasma (FuPls) for all the chemicals in the project

getVariabilitySetChoices *Get all variability sets*

Description

Get all the variability datasets in a given projects. Variabilities need to be handled differently from the other set types since they can themselves be of multiple types

Usage

```
getVariabilitySetChoices(var_type = "physio")
```

Arguments

var_type	type of Variability set to return, can be "physio","chem" or "expo" or "conc"

Value

named list of all sets of the var_type

httkCalcOralEqDose *Function that runs the httk oral equivalent dose gadget.*

Description

Launches the HTTK oral equivalent dose gadget. It servers as a UI wrapper around HTTK's functions to calculate oral equivalent dose.

Usage

```
httkCalcOralEqDose()
```

Examples

```
if(interactive()){
httkCalcOralEqDose()
}
```

httkParameterPBTK *Call httk function for parameterizing PBPK models*

Description

Function that calls parameterize_pbpk function within the httk package

Usage

```
httkParameterPBTK(chem_name, species = "Human")
```

Arguments

chem_name	name of the chemical for which to parameterize the model. Has to be in the httk database.
species	species for which to parameteize the model for. Defaults to human

Value

list containing parameters for the PBPK model

HT_IVIVE *server function of high throughput dosimetry*

Description

This function is needed internally by the package to handle the server functions related to adding compounds in the HT-IVIVE UI. It is never intended to be called by the user.

Usage

```
HT_IVIVE(
  input,
  output,
  session,
  vals = "",
  type = "",
  chem_list = list(),
  idx = 0,
  row_selected = 0
)
```

Arguments

input	input object from the data input UI
output	output object from the data input UI
session	session in which this module is called
vals	values for clearance
type	IVIVE type
chem_list	List of imported chemicals in the project
idx	index of the row
row_selected	row selected for editing

HT_IVIVEUI *Module for editing high throughput reverse dosimetry functions*

Description

The UI for defining HT-IVIVE parameters in the HT-IVIVE project. It is called by the HT-IVIVE server script when a new row is added or existing row is edited. It is never called directly by the user.

Usage

```
HT_IVIVEUI(namespace = "", set_list = NULL)
```

Arguments

namespace	namespace for the module. This is unique and decided by the project server function
set_list	A list of inputs for the dropdown menus.

---

importAllExposureData *Server function for all (Batch Exposure, TRA, SEEM, and SHEDS) data module*

Description

Server function for import all (Batch Exposure, TRA, SEEM, and SHEDS) data module. This function should not be called by the user

Usage

```
importAllExposureData(input, output, session, expo_name_df)
```

Arguments

input	input object for the UI
output	input object to the UI
session	session object for the module
expo_name_df	dataframe containing variable names for exposure values

---

`importAllExposureDataUI` *UI for importing all (Batch Exposure, TRA, SEEM, and SHEDS) data module.*

Description

This function is called by the pbpk model to import all (Batch Exposure, TRA, SEEM, and SHEDS) data module estimates. Never called by the user

Usage

```
importAllExposureDataUI(namespace)
```

Arguments

namespace	namespace for the module UI.

`importBatchExposure` *Server function for batch data module*

Description

Server function for import batch data module. This function should not be called by the user

Usage

```
importBatchExposure(input, output, session, expo_name_df)
```

Arguments

input	input object for the UI
output	input object to the UI
session	session object for the module
expo_name_df	dataframe containing variable names for exposure values

importBatchExposureUI *UI function for importing generic exposure data*

Description

UI function for importing generic exposure data into PLETHEM. The function should not be called by the user

Usage

```
importBatchExposureUI(namespace)
```

Arguments

namespace	namespace for the module

importHTTKData *Server function for importing HTTK data*

Description

This is the server function for the importHTTKData module. This is not to be called by the user.

Usage

```
importHTTKData(input, output, session)
```

Arguments

input	input object from the UI
output	output object with results
session	session from which this module is called

importHTTKDataUI *UI for importing data from HTTK*

Description

UI function for the shiny module called by the rapidPBPK Exposure UI. This will never be called directly by the user.

Usage

```
importHTTKDataUI(namespace)
```

Arguments

namespace	namespace for the module

importParameterSet *Server for import parameter module*

Description

server function for importing parameter sets from user databases. This function should never be called by the user

Usage

```
importParameterSet(input, output, session, set_type, module_source = "PBPK")
```

Arguments

input	input object from the UI
output	output object for the UI
session	session object for the server
set_type	type of data to be imported
module_source	workflow from which the module is called

importParameterSetUI *UI for importing parameter sets*

Description

UI for importing parameters from user or main databases

Usage

```
importParameterSetUI(namespace, set_type)
```

Arguments

namespace	namespace for this module
set_type	type of data to be imported; physiological, chemical or exposure

importSEEMData *Server function for seem data module*

Description

Server function for import seem data module. This function should not be called by the user

Usage

```
importSEEMData(input, output, session, fpath, expo_name_df)
```

Arguments

input	input object for the UI
output	input object to the UI
session	session object for the module
fpath	path to the SEEM database
expo_name_df	dataframe containing variable names for exposure values

importSEEMDataUI *UI for importing SEEM data.*

Description

This function is called by the pbpk model to import SEEM exposure estimates. Never called by the user

Usage

```
importSEEMDataUI(namespace)
```

Arguments

namespace	namespace for the module UI.

importShedsData *Server function for importing SHEDS data*

Description

Server function for importing SHEDS exposures estimates into PLETHEM. The function should not be called by the user

Usage

```
importShedsData(input, output, session, path, expo_name_df)
```

Arguments

input	input object for the UI
output	input object to the UI
session	session object for the module
path	path where SHEDS results are stored
expo_name_df	dataframe containing variable names for exposure values

importShedsDataUI *UI function for importing SHEDS data*

Description

UI function for importing SHEDS exposures estimates into PLETHEM. The function should not be called by the user

Usage

```
importShedsDataUI(namespace)
```

Arguments

namespace	namespace for the module

interactiveHT *Launch HT-IVIVE interface*

Description

Used internally to launch the HT-IVIVE UI. HT-IVIVE does not use the project management system that PBPK models uses.

Usage

```
interactiveHT(name = "HT-IVIVE")
```

Arguments

name	name of the model. Has to be "HT-IVIVE"

Examples

```
if(interactive()){
interactiveHT("HT-IVIVE")
}
```

`interactivePBPK` *Launch the interactive PBPK workflow for the given model*

Description

Used to launch the PBPK workflow for the given model. This interface can be used to launch either the rapidPBPK model, The HTTK model or the fishPBPK model.

Usage

```
interactivePBPK(name = "rapidPBPK")
```

Arguments

`name`	Name of the model. "rapidPBPK" or "fishPBPK". Defaults to rapidPBPK.

Examples

```
if(interactive()){
interactivePBPK("rapidPBPK")
}
```

`interactiveReverseDosimetry`
Launch Reverse Dosimetry Interface

Description

Used to launch the reverse dosimetry UI. This UI allows the user to perform reverse dosimetry if they have already run Monte Carlo Anlaysis outside of PLETHEM.

Usage

```
interactiveReverseDosimetry()
```

See Also

`interactivePBPK` for running reverse dosimetry using the rapidPBPK model in PLETHEM.

Examples

```
if(interactive()){
interactiveReverseDosimetry()
}
```

loadProject *Load the project from the project file located at the given path*

Description

Loads the project data from the project file and then launches the shiny UI that corresponds to the analysis type that the project belongs to.

Usage

```
loadProject(file_path = "", runUI = TRUE)
```

Arguments

file_path	path to the project file. If no path is provided, launches a select file dialog box for the user to select the path
runUI	trigger the appropriate interface after data is loaded into the database

Examples

```
if(interactive()){
loadProject(file_path = "TestPBPK.Rdata")
loadProject()
}
```

loadTRA *run TRA gadget and save files as needed.*

Description

Run TRA gadget and save CSV files needed to export TRA exposures for the PBPK model

Usage

```
loadTRA()
```

Examples

```
if(interactive()){
loadTRA()
}
```

`mainDbSelect` *Runs all select queries to the main database.*

Description

The function runs the select queries issued to the main db and returns the dataframe

Usage

```
mainDbSelect(query, db_path = "")
```

Arguments

`query`	A valid SQL Query
`db_path`	The location of the project database. This defaults to database/plethemdb.sqlite and is not expected to change This function will not be called by the user directly

`mainDbUpdate` *Runs all update queries to the main database.*

Description

The function runs the update queries issued to the main db

Usage

```
mainDbUpdate(query, db_path = "")
```

Arguments

`query`	A valid SQL Query
`db_path`	The location of the main database. This defaults to database/plethemdb.sqlite and is not expected to change This function will not be called by the user directly

mcDataExample — *Monte Carlo Example Data Set For Testing The runReverseDosimetry Function*

Description

This dataset was produced from the rapidPBPK PLETHEM module and is intended to be used solely for the automated testing of the runReverseDosimetry function.

Usage

```
data(mcDataExample)
```

Format

An object of class `data.frame` with 1000 rows and 25 columns.

Source

PLETHEM

References

PLETHEM

Examples

```
data(mcDataExample)
```

newEditVariability — *Server function for defining variability and uncertainty datasets*

Description

Server function for defining variability and uncertainty datasets in the rapidPBPK model. This should not be called by the user

Usage

```
newEditVariability(
  input,
  output,
  session,
  set_type,
  ops_type,
  var_params_list,
  set_id = 0
)
```

Arguments

input	input object from UI
output	output object to the UI
session	session object for this module
set_type	type of set for which variability is defined
ops_type	Operation requested. new variability or edit existing
var_params_list	List of parameters for variability
set_id	id for the variability set in the database

---

newEditVariabilityUI *UI function for defining variability and uncertainty datasets*

Description

UI function for defining variability and uncertainty datasets in the rapidPBPK model. This should not be called by the user

Usage

```
newEditVariabilityUI(namespace)
```

Arguments

namespace	namespace for the module when it is called form the PBPK UI

---

newProject *Start a new PLETHEM project.*

Description

A project consists of chemicals, organisms and datasets. The function asks the user for a location to save the project files on exit. It then launches the shiny user interface used to parameterize and run the model.

Usage

```
newProject(
  name = "new_project",
  save_path = "",
  type = "PBPK",
  model = "rapidPBPK",
  mode = "MC",
  runUI = FALSE
)
```

Arguments

name	The name for the project
save_path	The path at which the new project will be saved
type	The type of the model that the project is tied to
model	The model to be used for the project
mode	Either Forward Dosimetry(FD) or Monte Carlo(MC) mode. Only valid for PBPK type models
runUI	trigger the appropriate interface after data is loaded into the database

Examples

```
if(interactive()){
newProject(name = "TestPBPK",type = "PBPK",model = "rapidPBPK",mode = "MC")
newProject(name = "TestPBPK",type = "PBPK",mode = "httk_pbpk",mode = "MC")
}
```

---

newProjectGadget *Command line function launching the new project gadget*

Description

This launches the new project gadget for launching a new project in PLETHEM. With RStudio, this function is available as an addin

Usage

```
newProjectGadget()
```

Examples

```
if(interactive()){
newProjectGadget()
}
```

parseConsExpoFile *Parse uploaded file for consexpo*

Description

Parses the uploaded ConsExpo exposure SpreadSheet to extract all the exposure names and values This function will not be called directly by the user

Usage

```
parseConsExpoFile(path)
```

Arguments

path	Path to Excel File

parseTRAFile *Parse uploaded file for Consumer TRA*

Description

Parses the uploaded Consumer TRA exposure SpreadSheet to extract all the exposure names and values This function will not be called directly by the user

Usage

```
parseTRAFile(path)
```

Arguments

path	Path to Excel File

`performIVIVE` *server side function for performing IVIVE for a chemical*

Description

The server side function for running the IVIVE module from within the PBPK UI

Usage

```
performIVIVE(input, output, session, km)
```

Arguments

`input`	input object for the UI
`output`	input object to the UI
`session`	session object for the module
`km`	km for the chemical

`performIVIVEUI` *Shiny module that is called when perform IVIVE button is clicked on the chemical tab of a PBPK model*

Description

UI for performing IVIVE from within the PBPK UI. This function should not be called by the user

Usage

```
performIVIVEUI(namespace)
```

Arguments

`namespace`	namespace for the UI

performPlethemNCA *Functions for performing NCA and returning the data*

Description

Calculates the The user will never need to call this function

Usage

```
performPlethemNCA(result, var_names, mode = "FD")
```

Arguments

result	Result from the forward dosimetry or individual montecarlo run
var_names	Variable names from the model for which NCA should be performed
mode	Forward Dosimetry or Montecarlo

Details

Calculate AUC, Cmax and Time to Cmax from the results of PBPK model run

preprocessUIData *preprocess value list from UI*

Description

The function converts the data from the UI to standard units of liters/h for clearances and mg/L for concentrations. This function is not available to the end user as it relied on UI names

Usage

```
preprocessUIData(val)
```

Arguments

val	list containing data for each row from the UI

Value

list of clearance values for point of departure values and type of reverse dosimetry

projectDbSelect *Runs all select queries to the project database.*

Description

The function runs the select queries issued to the project db and returns the dataframe

Usage

```
projectDbSelect(query, db_path = "")
```

Arguments

query	A valid SQL Query
db_path	The location of the project database. This defaults to database/project.sqlite and is not expected to change This function will not be called by the user directly

projectDbUpdate *Runs all update queries to the project database.*

Description

The function runs the update queries issued to the project db

Usage

```
projectDbUpdate(query, db_path = "")
```

Arguments

query	A valid SQL Query
db_path	The location of the project database. This defaults to database/project.sqlite and is not expected to change This function will not be called by the user directly

projectReadTable *Read all the contents of a table*

Description

The functions reads all the contents of the given table. It is used internally to save project data

Usage

```
projectReadTable(tble_name, db_path = "")
```

Arguments

tble_name	Name of the table to save. NOTE SQLITE table names are not case sensitive
db_path	The location of the project database. This defaults to database/project.sqlite and is not expected to change

Value

table as a dataframe

projectWriteTable *Write the dataframe to the table*

Description

The functions writes the datafarme to the sqlite table. It will overwrite any data It is used internally to load project data

Usage

```
projectWriteTable(data, tble_name, db_path = "")
```

Arguments

data	Data frame containing the data to save to the table. This will overwrite any existing data
tble_name	Name of the table to write the data to. NOTE SQLITE table names are not case sensitive
db_path	The location of the project database. This defaults to database/project.sqlite and is not expected to change

Value

None

qsarModelone *Calculate Partition Coefficient using the default QSAR model*

Description

Calculates the partition coefficient using the default QSAR model. This QSAR model is based on the one described by Jongneelan et al as a part of the IndusChemFate model

Usage

```
qsarModelone(chem_params, selected_org, tissue_list)
```

Arguments

chem_params	A named list of parameters needed to run the model
selected_org	Either "human" or "rat"
tissue_list	List of tissues for which partition coefficients have to be calculated

Value

Named list of partition coefficients, one for each tissue in the tissue list

qsarModeltwo *Calculate Partition Coefficient using the default QSAR model*

Description

Calculates the partition coefficient using the default QSAR model. This QSAR model is based on the one described by Jongneelan et al as a part of the IndusChemFate model

Usage

```
qsarModeltwo(chem_params, selected_org, tissue_list)
```

Arguments

chem_params	A named list of parameters needed to run the model
selected_org	Either "human" or "rat"
tissue_list	List of tissues for which partition coefficients have to be calculated

Value

Named list of partition coefficients, one for each tissue in the tissue list

readBatchChemicalFile *Read and parse the chemical batch file*

Description

Function to read the chemical batch file. The batch file description can be found in the user guide.

Usage

```
readBatchChemicalFile(file_path)
```

Arguments

file_path	path to the batch chemical csv file

readOperaPredictions *Read and parse the predictions from OPERA*

Description

Function to read the OPERA predictions. The function removes predictions made by OPERA that are not used by PLETHEM.

Usage

```
readOperaPredictions(file_path)
```

Arguments

file_path	path to OPERA predictions

reshapePlotData *reshape plotted data to create wide form*

Description

Reshapes plot data in long form to wide form. The plot data has time as the id

Usage

```
reshapePlotData(plotData, type = "fd")
```

Arguments

plotData	Plot Data in long form
type	Workflow type - either fd (Forward Dosimetry) or mc (Monte Carlo Analysis)

runFDPBPK *Run the PBPK models in forward dosimetry mode*

Description

Run the pbpk model in forward dosimetry mode. This function is common across all PBPK models. This function can be used from the console if all the inputs are provided

Usage

```
runFDPBPK(initial_values, model = "rapidPBPK")
```

Arguments

initial_values	A list containing initial values needed to run the model
model	The name of the PBPK model to simulate

runHTIVIVE *Run HT-IVIVE*

Description

Launches the HT-IVIVE UI.

Usage

```
runHTIVIVE()
```

Examples

```
if(interactive()){
runHTIVIVE()
}
```

runPlthemHTIVIVE *High Throughput reverse dosimetry calculation using UI*

Description

Main function called from PLETHEM UI to run HT- Reverse Dosimetry. This should not be called by the user

Usage

```
runPlthemHTIVIVE(vals)
```

Arguments

vals	values passed from the PLETHEM UI

Value

List of oral equivalent dose, steady state plasma concentration and steady state for each name in vals

runReverseDosimetry *Estimate exposure from montecarlo results and biomonitoring data*

Description

The function estimates exposures for the observed biomonitoring data using montecarlo simulation results over a large range of exposures. The montecarlo results are obtained from a PBPK model. The biomonitoring results are obtained from a population level study. The montecarlo results and biomonitoring data should have the same units and should be for the same physiological data source (eg: metabolite concentration in the urine).

Usage

```
runReverseDosimetry(
  mcData,
  biomData,
  percentiles = c(5, 10, 25, 50, 75, 95, 99, 100),
  dose_list = NULL
)
```

Arguments

mcData	M by N data frame where M is the the individual exposures at which the PBPK model is run and N is the number of monte carlo runs at each exposure. It is recommended that M is between 25 and 40 and N is greater than 1000.
biomData	List consisting of biomonitoring data. It is recommended that atleast 1000 biomonitering values be provided to ensure accuracy for results.
percentiles	Vector of percentiles for which exposure needs to be estimated. By default returns the 5th, 50th, 95th and 99th exposure estimate.
dose_list	A list of M elements that contain exposures at which monte carlo simulations were run. If no list is provided, the first column names of the mcData input are assumed to contain exposure values.

Value

List of values related to reverse dosimetry

cdf Cumulative Distribution function of the exposure estimate

pdf Probability distribution function of the exposure estimate

percentiles Data frame of percentiles and exposure estimates for the percentile

Examples

```
pD <- c(seq(.05,.90,.05),.94,.95,.96,.97,.98,.99,1.00)
dosing <- c(0.010, 0.011, 0.012, 0.014, 0.016, 0.018, 0.021, 0.023, 0.027, 0.030, 0.034, 0.039)
dosing <- c(dosing, 0.044, 0.050, 0.057, 0.065, 0.073, 0.083, 0.094, 0.107, 0.121, 0.137)
dosing <- c(dosing, 0.155, 0.176, 0.200)
runReverseDosimetry(mcDataExample, biomDataExample, percentiles = pD, dose_list = dosing)
```

---

saveAsParameterSet *server side function for saving a new physiological, chemical or exposure set to the project database*

Description

Server side function for running the save parameter module. This function should not be called by the user

Usage

```
saveAsParameterSet(
  input,
  output,
  session,
  set_type,
  main_input,
  name_df,
  other = NULL
)
```

Arguments

input	input object for the UI
output	input object to the UI
session	session object for the module
set_type	type of parameter set to save
main_input	input from the pbpk UI
name_df	variable names for parameters
other	placeholder paramter for data needed for certain sets

---

saveAsParameterSetUI *UI for saving a new physiological, chemical or exposure set to the project database*

Description

UI for saving parameter sets. This function should not be called by the user

Usage

```
saveAsParameterSetUI(namespace, set_type)
```

Arguments

namespace	namespace for the UI
set_type	type of parameter set to save

---

saveProject *Save the current project to a location*

Description

Save the current PBPK or HTIVIVE project the user is working on. This cannot be used to save exposure or IVIVE gadget data. This function should not be called directly from the console. It will be called by the app on exit

Usage

```
saveProject()
```

saveRestoreParameterSet *Server for the restore/save dialog*

Description

Server for the save restore dialog box. This function should not be called by the user.

Usage

```
saveRestoreParameterSet(
  input,
  output,
  session,
  UI_values,
  set_values,
  param_names,
  type
)
```

Arguments

input	input object for the UI
output	input object to the UI
session	session object for the module
UI_values	values for the parameters in the UI
set_values	values for the parameters in the database
param_names	names of parameters to save or restore
type	type of parameter set to save

saveRestoreParameterSetUI *UI for the restore/save dialog*

Description

UI for the save restore dialog box. This function should not be called by the user.

Usage

```
saveRestoreParameterSetUI(namespace)
```

Arguments

namespace	namespace for the UI

scale_cellular_enzymatic *Scale clearance by age when both cellular and enzymatic clearance are known at reference age.*

Description

This function is used internally to calculate age specific metabolism using the IVIVE gadget. It needs both cellular and enzymatic clearance at atleast one age, the reference age, to extrapolate to values at other ages.

Usage

```
scale_cellular_enzymatic(
  out_ages,
  tot_scaled_hepcl,
  tot_scaled_recomcl,
  cypDb,
  cypCl,
  gender
)
```

Arguments

out_ages	Ages for which the clearance needs to be calculated including reference age
tot_scaled_hepcl	Total cellular clearance at reference age in L/h/kg Liver
tot_scaled_recomcl	Total Recombinant enzyme clearance at reference age in L/h/kg Liver
cypDb	Dataframe containing cyp datasets to scale measured clearance values from in-vitro to in-vivo
cypCl	Datafarame containing measured invitro enzymatic clearance.
gender	Gender either "M" for male or "F" for female

Value

List with individual enzyme and total clearance at all ages.

`scale_enzymatic` *Scale clearance by age for when enzymatic clearance is known at reference age.*

Description

Scale clearance by age for when enzymatic clearance is known at reference age.

Usage

```
scale_enzymatic(out_ages, tot_scaled_recomcl, cypDb, cypCl, gender)
```

Arguments

`out_ages`	out_ages Ages for which the clearance needs to be calculated including reference age
`tot_scaled_recomcl`	Total Recombinant enzyme clearance at reference age in L/h/kg Liver
`cypDb`	Dataframe containing cyp datasets to scale measured clearance values from in-vitro to in-vivo
`cypCl`	Datafarame containing measured invitro enzymatic clearance.
`gender`	Gender, either "M" for male or "F" for female. Needed to get Liver weight

`setUserDb` *Sets the path to the existing Db*

Description

The function allows the users to set an existing database as the user database to use for plethem.

Usage

```
setUserDb(path = NULL)
```

Arguments

`path`	path to the user database file. If no path is provided, it launches a file select dialog.

Examples

```
## Not run:
setUserDb(),
setUserDb(tempdir())

## End(Not run)
```

updateUIInputs *Update Inputs for PLETHEM UI*

Description

This a common function used to update the inputs for any PLETHEM User Interface. It is used by the apps to provide a common pathways for all UI updates to happen

Usage

```
updateUIInputs(session, param_df)
```

Arguments

session	The Shiny session in which all the inputs have to be updated
param_df	The parameter dataframe. The parameter dataframe has the following columns Name - The name of the parameter Var - The variable in the UI representing this parameter ParamType - The type of input, either Numeric, Radio, Checkbox, Select, Tabset. Used to identify which update function to call Val - The value to be update with. If the type is numeric, the value is coerced to be a number.

userDbSelect *Runs all select queries to the user database.*

Description

The function runs the select queries issued to the user db and returns the dataframe the path to user database is stored in main plethem database and is selected from there

Usage

```
userDbSelect(query)
```

Arguments

query	A valid SQL Query

`userDbUpdate` *Runs all update queries to the user database.*

Description

The function runs the update queries issued to the user db

Usage

```
userDbUpdate(query)
```

Arguments

`query`	A valid SQL Query

Index