ICE IVIVE workflow

Description

The workflow allows the flexibility to select from five different rat and human PK models: a one-compartment model that incorporates Monte Carlo simulation to simulate the population variance (1C), and four models leveraging the EPA's httk package which include a 3 compartment model (solve_3comp), a multi-compartment PBPK model for modeling oral and i.v. exposure (solve_pbtk), a multi-compartment PBPK model for modeling inhalation (gas) route of exposure (solve_gas_pbtk), and a multi-compartment human PBPK model that includes both maternal and fetal compartments, and a placenta modeled as a joint organ shared by mother and fetus (solve_fetal_pbtk). The workflow is to predict the equivalent administered dose (EAD) that would lead to steady state plasma concentration or maximum plasma concentration (Cmax) equal to the bioactive concentration from in vitro assays. For inhalation exposures using the (solve_gas_pbtk) EAD is generated in uM or ppmv if using"concentration" exposure vs a dose.

Two example files are included: * ChemicalData_Rnotebook.txt * InvitroData_Rnotebookv.txt

Load libraries

```
#load libraries
library(tidyverse)   #needed for read-delim()
library(deSolve)
library(doParallel)
library(httk)        #this is needed for models: solve_3comp, solve_pbtk, solve_gas_pbtk, solve_fetal_p
                     #The code is compatible with httk_2.2.2
library(xlsx)        #for writing the excel file that is user output
date <- format(Sys.Date(), format="%y%m%d") #added to add information on date
```

Input data and output file path

There are several input variables needed to run the code. Some variables are model specific, as detailed. Adjust the file paths to point to the file in your directory.

chemFile: the file containing chemical data from ICE. this has the CASRN as the first field assayFile: the file with the in vitro bioactivity data. The first column is the CASRN and the subsequent columns are the bioactivity values output_file: this is the file for the outputs

```
chemFile <- "ChemicalData_Rnotebook.txt" #chemicals data from ICE, include CASRN field as identifier
assayFile <- "invitroData_Rnotebook.txt" #invitro data from ICE, includes CASRN field as identifier the

Userout <- "User_Results.xlsx"           #file name (and path) for the output file for the tissue; conc
EADplot_file <-"User_plot.pdf"           #the file for the plot of the EAD plot
invivo <- NULL                           #invivo data provided by user
```

```
#invivoFile <- "InvivoData_Rnotebook.txt"
#invivo <- as.data.frame(read_delim(invivoFile, delim = "\t"))
```

Model variables

Details about what model, route, and dose need to be specified. There are differences in what is needed if a 1 compartment model is used vs the PBPK models

What model?

Models type is limited to 5 different models. Currently, species in ICE is limited to human or rat. Using this notebook one can expand the species with minor editing of the code provided the parameters are available.

```
species <- "human"                          #human or rat
bWeight <- 70                               #the body weight of simulated subject
modelType <-  "solve_fetal_pbtk"            #"1C", solve_3comp", "solve_pbtk", "solve_gas_pbtk", "solve_fe
```

For the 1 compartment model, values are needed to parameterize the Monte Carlo simulation.

```
nsamples <- 300                             #user-provided value for the mc simulations, any number betwee
```

For the PBPK models some additional parameters can be modified. An inhalation exposure has additional parameters The route determines where the chemical will enter the system and route, interval, and days are needed for all PBPK models.

```
route <- "oral"                             #oral, iv, or inhalation needed for PBPK models; "solve_gas_pb
interv <- 24                                #dosing interval, hours
ndays <- 3                                  #number of days dosing is done
expDose <- 1                                #current calculations assume 1mg/kg/dose for "1C", "solve_3com
expConc <- 1                                #Dosing for "solve_gas_pbtk" model
ConcentrationUnit <- "uM"                   #Renamed as Output Conc. Units in PBPK tool since ICE4.0. Opti
                                            #the unit for in vitro activity concentration is consistent wi
gestationDays <- 91                         #gestational days when dose starts, range is 91 days (14 weeks
```

Inhalation specific parameters

For an inhalation route of exposure, there are 2 different ways that the exposure can be modeled. “expDose” models a set bolus dose like the exposure for an oral or IV exposure. The “expConc” is more typical of a gas exposure. This models an inhaled concentration over a duration of time, set by “expLength”. The “expDose” option is currently unavailable.

```
gasDosing <-"expConc"                       #two dosing option methods for solve_gas_pbtk model, "expConc"
                                            #but for ICE3.3 release, "expConc" is the only option due to a
                                            #This is only needed if gasDosing =="expConc" because otherwis
if(gasDosing =="expConc" & (!exists("expConc") || is.null(expConc))){
  expConc <- 1                              #current calculations assume 1uM of air concentration
}
gasInputUnit <- "ppmv"                      #Input concentration unit for gas model (httkv2.1.0 and later)
expLength <- .25                            #length of the gas exposure, hours
```

Gestation model specific parameters

```
gestationDays <- 91                #gestation model specific parameter;gestational days when dose start
```

Argument for EADboxplot.R

```
chemDisplay <- "ChemicalName"              #choice is "CASRN" or "ChemicalName"
if(modelType != "solve_gas_pbtk"){         #"mg/kg/dose" for "1C", "solve_3comp", "solve_pbtk"; uM or ppm
  EADUnit <- "mg/kg/dose"
}else{EADUnit <- gasInputUnit}
```

Load functions

```
                                           #All required R scripts and input files should be in the worki
source("steadyState.R")                    #required for "1C" model
source("CalcEAD.R")                        #required for "1C", "solve_3comp", "solve_pbtk", "solve_gas_pb
source("EADboxplot.R")
```

Load data

Notice that the chem input has the source of the FU and Clint data. ICE now has 2 different sources of these values so the source is important for tracking this information.

```
#there are single quotes in chemical names that need to be addressed, the tidyR handles well
chemical <- as.data.frame(read_delim(chemFile, delim = "\t"))
chemical[1:2,]
```

```
##   Substance Name      CASRN        DTXSID Chemical Parameters fu fu Source
## 1 Beclomethasone  4419-39-0 DTXSID5040750                  0.130     OPERA
## 2     Ketanserin 74050-98-9 DTXSID3023188                  0.052     OPERA
##   Chemical Parameters Clint ul/min/10^6 cells Clint Source
## 1                                   34.43499        OPERA
## 2                                   46.88134        OPERA
##   Chemical Parameters pKa, Acidic   Chemical Parameters LogP log10
## 1                           8.689                           1.980
## 2                           8.597                           3.282
##   Chemical Parameters MW g/mol  Chemical Parameters HL log10, atm-m3/mole
## 1                      408.916                                    -10.130
## 2                      395.427                                     -7.616
##   Chemical Parameters pKa, Basic
## 1                          6.546
## 2                          5.826
```

```
invitro <- as.data.frame(read_delim(assayFile, delim = "\t"))
invitro[1:2,]
```

```
##        CASRN ACEA_ER_80hr    ATG_Ahr_CIS_up    ATG_GRE_CIS_up
## 1  4419-39-0             NA                NA                NA
## 2 74050-98-9             NA                NA                NA
##   OT_ER_ERaERb_0480    OT_ER_ERaERb_1440    OT_ER_ERbERb_0480
## 1                NA                   NA                   NA
## 2                NA                   NA                   NA
##   OT_ER_ERbERb_1440    TOX21_AhR_LUC_Agonist    TOX21_CAR_Agonist
## 1                NA                       NA                   NA
## 2                NA                       NA                   NA
##   TOX21_ERb_BLA_Antagonist_ratio    TOX21_GR_BLA_Agonist_ratio
## 1                             NA                         0.033
## 2                             NA                            NA
##   TOX21_GR_BLA_Antagonist_ratio    TOX21_TR_LUC_GH3_Agonist
## 1                            NA                          NA
## 2                            NA                          NA
##   TOX21_TR_LUC_GH3_Antagonist    TOX21_TSHR_Agonist_ratio
## 1                          NA                          NA
## 2                          NA                          NA
##   TOX21_TSHR_Antagonist_ratio
## 1                          NA
## 2                          NA
```

Preparing data

Minor prep work is done on the data that comes from ICE Partitioning coefficients are obtained by internal function from the httk package using the provided phys chem parameters. Compare with the example input file for naming Note that a few checks are done. These will generate flags that can be used to follow up as needed.

```
##column names should be labeled correctly. RegEx have been used to help, see the example file
##cleaning the names for taking from an ICE Query-stricter matching is used to handel including the ADM
colnames(chemical)<-gsub(".*Name.*", "ChemicalName", colnames(chemical))
colnames(chemical)<-gsub(".*Parameters fu.*", "fu", colnames(chemical), ignore.case =TRUE)
colnames(chemical)<-gsub(".*funbound.*", "fu", colnames(chemical), ignore.case =TRUE)
colnames(chemical)<-gsub(".*fu Source.*", "fu Source", colnames(chemical), ignore.case =TRUE)
colnames(chemical)<-gsub(".*MW.*", "MW", colnames(chemical))
colnames(chemical)<-gsub(".*HL.*", "HL", colnames(chemical))
colnames(chemical)<-gsub(".*Parameters Clint.*", "Clint", colnames(chemical), ignore.case = TRUE)
colnames(chemical)<-gsub(".*Clint Source.*", "Clint Source", colnames(chemical), ignore.case = TRUE)
colnames(chemical)<-gsub(".*LogP.*", "LogP", colnames(chemical))
colnames(chemical)<-gsub(".*pKa, Basic.*", "pka_Accept", colnames(chemical))
colnames(chemical)<-gsub(".*pKa, Acidic.*", "pka_Donor", colnames(chemical))

#Converting units for internsic clearance values
#this moves from log10 ul/ml/10^6 cells to just ul/ml/10^6 cells
#chemical$Clint<-10^chemical$Clint
chemical$logPwa<--1*chemical$HL #this gets the water octanal coeff
#Add the flag here as a warning if model not appropriate or other issues related to generating predicti
```

```
chemical$Flag <-""
#catch for low Fu
chemical$Flag[chemical$fu<=1e-10]<-"fu is zero,likely due to rounding. Setting to 1e-5"
chemical$fu[chemical$fu<=1e-10]<-1e-5
```

1C modeling

If modelType given by the user is "1C" This will run a simple, 1 compartment model. In this model the C steady state is determined vs the C max from the other models.

Calculating the EADs

The 1C model has a monte carlo simulation that calculates the Css at the 50th and the 95th percentile The first step is generating the CSS object with the call to "steadState".

```
### model "1C"
if(modelType == "1C"){

  chemInput <- chemical[,c("CASRN", "ChemicalName", "Clint", "fu", "MW", "Flag")]
  CSS<- steadyState(inputData = chemInput, nsamples = nsamples, species = species, ConcentrationUnit = C
  head(CSS)

  #There are 2 different steady state predictions, 50 and 95 percentile.
  #The EAD is calculated for each using the "CalcEAD" function. The data is then formated into a single

  EAD.out50 <- CalcEAD(Css = CSS[,c("CASRN", "50%", "fu")], inVitro = invitro, adj.fu = "fu")
  colnames(EAD.out50)<-gsub("EAD","EAD.50", colnames(EAD.out50))
  #add in the flag column
  EAD.out50<-left_join(chemInput[,c("CASRN","Flag")],EAD.out50)
  EAD.out95 <- CalcEAD(Css = CSS[,c("CASRN","95%", "fu")], inVitro = invitro, adj.fu = "fu")
  colnames(EAD.out95)<-gsub("EAD","EAD.95", colnames(EAD.out95))
  EAD.out<-left_join(EAD.out50,EAD.out95)
  EAD.out<-EAD.out[,setdiff(colnames(EAD.out), c("adj.fu", "fu", "adj.arm","arm"))] # remove the columns
  #creating output file for user, should have EAD values and the parameters for calculating the circula
  CSS2<-as.data.frame(CSS, stringsAsFactors=FALSE); colnames(CSS2)<-gsub("50%", "Css, 50%ile", colnames
  CSS2$Species<-species; CSS2$Model<-modelType; CSS2$"Dose, mg/kg"<-expDose;CSS2$Route<-route
  CSS2$nSimulations <- nsamples
  colnames(CSS2)<-gsub("Css_Unit","Css, Units", colnames(CSS2))

  CSS2<-left_join(CSS2, chemical %>% select(any_of( c("CASRN","Clint Source", "fu Source"))))

  #reformatting columns to match the other models:
  CSS2<-CSS2 %>% select(any_of(c("CASRN", "ChemicalName", "Css, 50%ile", "Css, 95%ile", "Css, Units", "
                               "Species", "Model", "Route", "Dose, mg/kg", "Clint", "Clint Source", "fu
  ssEAD.out<-EAD.out[,c("CASRN", setdiff(colnames(EAD.out), c(colnames(invitro), "50%","95%")))]
  outputData<-full_join(CSS2,ssEAD.out)
  colnames(outputData)<-gsub("Clint$", "Clint, ul/ml/10^6 cells", colnames(outputData))#add units

  #The output file is an excel workbook. This has 2 tabs, one for the EAD results and the other for the
  xlsx::write.xlsx(file=Userout,x=outputData, sheetName ="EADResults", append=FALSE, row.names = FALSE,
```

```
  xlsx::write.xlsx(file=Userout,x=invitro, sheetName ="inVitroData", append=TRUE, row.names = FALSE, sh

  #To view the results, the "EADboxplot" function plots the values.
  EADplot <- EADboxplot(EAD.out = outputData, invivo = invivo, label="EAD", EADUnit = EADUnit, species =
  #Arguments for EADplot functions include "EAD.out, invivo, label, chemDisplay, EADUnit, modelType, sp
  EADplot
  }
```

script to run pbpk models from httk package

This code is used if modelType given by the user is a PBPK model: "solve_3comp", "solve_pbtk", "solve_gas_pbpk", or "solve_fetal_pbtk. This is wrapper code to format the inputs and parameters needed to run the models specified by the httk package. In addition, the output is formatted so that it matches the 1C model. This allows easy comparison between the different files for subsequent processing.

Parameter processing for the pbpk models

The httk package needs some additional parameters including the correct capitalization of the species and ensuring that the adding of the user-specified data to the chem table is properly integrated.

```
# preprocessing variables:
if (tolower(species) == "rat") {
  species_1 <- "Rat"
}
if (tolower(species) == "human") {
  species_1 <- "Human"
}
options(stringsAsFactors = FALSE)
# add chemical info to the table. Using variable coming from ICE
# to deal with mapping issues
chemical$DTXSID2<-paste0(chemical$DTXSID, "_n")
if(species_1=="Human"){
  chem.physical_and_invitro.data <- add_chemtable(chemical, current.table = chem.physical_and_invitro.da
}else{
  chem.physical_and_invitro.data <- add_chemtable(chemical, current.table = chem.physical_and_invitro.da
  chem.physical_and_invitro.data <- add_chemtable(chemical, current.table = chem.physical_and_invitro.da
}

if(route != "iv"){
  iv.dose = FALSE
}else{iv.dose =TRUE}

dpd <- 24/interv #calculating the doses per day
```

to allow the user to change the body weight for rat and human

```
  if(species_1=="Human"){
    physiology.data[4,6] <- bWeight
```

```
  }else{
    physiology.data[4,4] <- bWeight #the rat body weight
  }
```

solve_3comp model

The solve_3comp model is a 3 compartment model that has the gut,gut lumen, liver, and rest of the body compartments with the plasma equivalent to the liver plasma concentrations.

```
if(modelType == "solve_3comp"){
  cmaxall <- NULL
  # note that dose=0 stops the initial dosing of the compartments and is needed to accurately model the
  for(this.cas in chemical[,"CASRN"]) {
    concMax <- max(solve_3comp(chem.cas = this.cas, parameters = NULL, doses.per.day = dpd, days = ndays
    cmax_temp <- as.data.frame(cbind(this.cas, concMax, ConcentrationUnit))
    cmaxall <- rbind(cmaxall, cmax_temp)
  }
}
```

solve_pbtk model

the solve_pbtk model is a PBPK model that has gutlumen, gut, liver, kidneys, veins, arteries, lungs, and the rest of the body compartments.

```
if(modelType == "solve_pbtk"){
  cmaxall <- NULL
  # note that dose=0 stops the initial dosing of the compartments and is needed to accurately model the
  for(this.cas in chemical[,"CASRN"]) {
      #p<-parameterize_pbtk(chem.cas = this.cas, species = species_1, default.to.human = TRUE) # to con
      #print(p$BW)
      if(route == "oral"){
      concMax <- max(solve_pbtk(chem.cas = this.cas, parameters = NULL, doses.per.day = dpd, days = nda
    }else if(route =="iv"){
      output0 <- solve_pbtk(chem.cas = this.cas, parameters = NULL, doses.per.day = dpd, days = ndays,
      #store output
      trim_output <- as.data.frame(output0)
      #ignore possible computational errors
      to_ignore <- 2 #<- time point 2 will always be anomalous
      n_to_ignore <- max(1,floor((24*ndays)/interv))
      if((24/interv) == 1){
        steps_between_doses <- 97
      }else if(24/interv > 1){
        steps_between_doses <- floor(97 / (24/interv))+1
      }else if(24/interv < 1){
        steps_between_doses <- floor(97 / (24/interv))
      }
      for(i in 1:n_to_ignore){
        to_ignore[i+1] = to_ignore[i] + steps_between_doses
      }
      trim_output <- trim_output[-to_ignore,]
```

```
      concMax <-max(trim_output[,'Cplasma'])
    }
    cmax_temp <- as.data.frame(cbind(this.cas, concMax, ConcentrationUnit))
    cmaxall <- rbind(cmaxall, cmax_temp)
  }
  }
```

solve_gas_pbtk model

The solve_gas_pbtk is a PBPK model similar to the "solve_pbtk" model but is uses an inhalation route of exposure assuming the chemical is volatile (gas). As a result, it has some addtional checks to see if the assumption of gas exposure is reasonable but will still proceed with a flag warning. It also includes 2 different dosing approaches, a concentration over time (expConc) or a single dose (expDose)

```
if(modelType == "solve_gas_pbtk"){
  #check the assumption of volatility
  chemical$Flag[chemical$HL <= -7.80388 & chemical$Flag!=""] <-"fu is zero,likely due to rounding. Setti
  chemical$Flag[chemical$HL <= -7.80388 & chemical$Flag==""] <-"Chemical likely nonvolatile, consider ap

  cmaxall <- NULL
  #note that dose=0 stops the initial dosing of the compartments and is needed to accurately model the
  for(this.cas in chemical[,"CASRN"]) {

    # --- expDose option has currently been disabled due to a bug in the httk package ---
    # if(modelType=="solve_gas_pbtk" & gasDosing == "expDose"){
    #   ConcentrationUnit <- "uM"  # output unit only has one option of "uM" for solve_gas_pbtk model
    #   concMax <- max(solve_gas_pbtk(chem.cas = this.cas, parameters = NULL, doses.per.day = dpd, days
    # }

    if(modelType == "solve_gas_pbtk"& gasDosing == "expConc"){
      concMax <- max(solve_gas_pbtk(chem.cas = this.cas, parameters = NULL, doses.per.day = NULL, days =
    } #note for v2.2.2, must specify total "daily.dose" when "doses.per.day" is not set to NULL.
    cmax_temp <- as.data.frame(cbind(this.cas, concMax, ConcentrationUnit))
    cmaxall <- rbind(cmaxall,cmax_temp)
  }
}
```

solve_fetal_pbtk model

The solve_fetal_pbtk is a human PBPK model that includes both maternal and fetal compartments and a placenta modeled as a joint organ shared by mother and fetus. It simulated gestation day of 91-280 days. A flag warning will show if the gestation day is out of this range. The exposure routes for this model are oral and IV.

```
if(modelType == "solve_fetal_pbtk"){

  if(species_1 == "Rat"){
      stop("Gestation model is for human only")
  }
  if(gestationDays < 91 || gestationDays > 280) {
```

```
    print("Warning: the gestation day is the day when dosing starts, which shall be within the range of
  }

  cmaxall <- NULL
  for(this.cas in chemical[,"CASRN"]) {
     if(species_1 == "Human" && route == "oral"){
      concMax <- max(solve_fetal_pbtk(chem.cas = this.cas, times=seq(gestationDays, gestationDays+ndays
      #note - httk v2.1.0 and later, species = "human" is the default
      fconcMax <- max(solve_fetal_pbtk(chem.cas = this.cas, times=seq(gestationDays, gestationDays+nday
    } #initial.values = NULL; #times: time sequence in days. Default is from 13th week of pregnancy to
     else if(species_1 == "Human" && route =="iv"){
      output0 <- solve_fetal_pbtk(chem.cas = this.cas, times=seq(gestationDays, gestationDays+ndays, 0.
      #store output
      trim_output <- as.data.frame(output0)
      #ignore computational errors
      to_ignore <- 2 #<- time point 2 will always be anomalous
      n_to_ignore <- max(1,floor((24*ndays)/interv))
      if((24/interv) == 1){
        steps_between_doses <- 41
      }else if(24/interv > 1){
        steps_between_doses <- floor(41 / (24/interv))+1
      }else if(24/interv < 1){
        steps_between_doses <- floor(41 / (24/interv))
      }
      for(i in 1:n_to_ignore){
        to_ignore[i+1] = to_ignore[i] + steps_between_doses
      }
      trim_output <- trim_output[-to_ignore,]
      concMax <-max(trim_output[,'Cplasma'])
      fconcMax <- max(trim_output[,'Cfplasma'])
    }  #initial.values = NULL; #times: time sequence in days. Default is from 13th week of pregnancy to
    cmax_temp <- as.data.frame(cbind(this.cas, concMax, fconcMax, ConcentrationUnit))
    cmaxall <- rbind(cmaxall,cmax_temp)
  }
}
```

Calc EAD from PBPk models except for "solve_fetal_pbtk"

The PBPK models currently give estimates based on the median (50%ile) of the population. As such there is minor formatting that needs to be done to generate the EAD predictions and make the output file

```
if(modelType == "solve_3comp" | modelType == "solve_pbtk" | modelType == "solve_gas_pbtk"){
  cmaxall$concMax <- as.numeric(cmaxall$concMax)
  #they are bound as characters so converting to numeric

  Cmax <- merge(chemical, cmaxall, by.x="CASRN", by.y="this.cas")
  names(Cmax) <- gsub("concMax", "Cmax", names(Cmax) )
  #Calculating the EADs
  EAD.out_max <- CalcEAD(Css = Cmax[,c("CASRN", "Cmax")], inVitro = invitro)
  #creating output file for user, should have EAD values and the parameters for calculating the circula
  Cmax2<-as.data.frame(Cmax, stringsAsFactors=FALSE)
  #adding in the source of the parameters
```

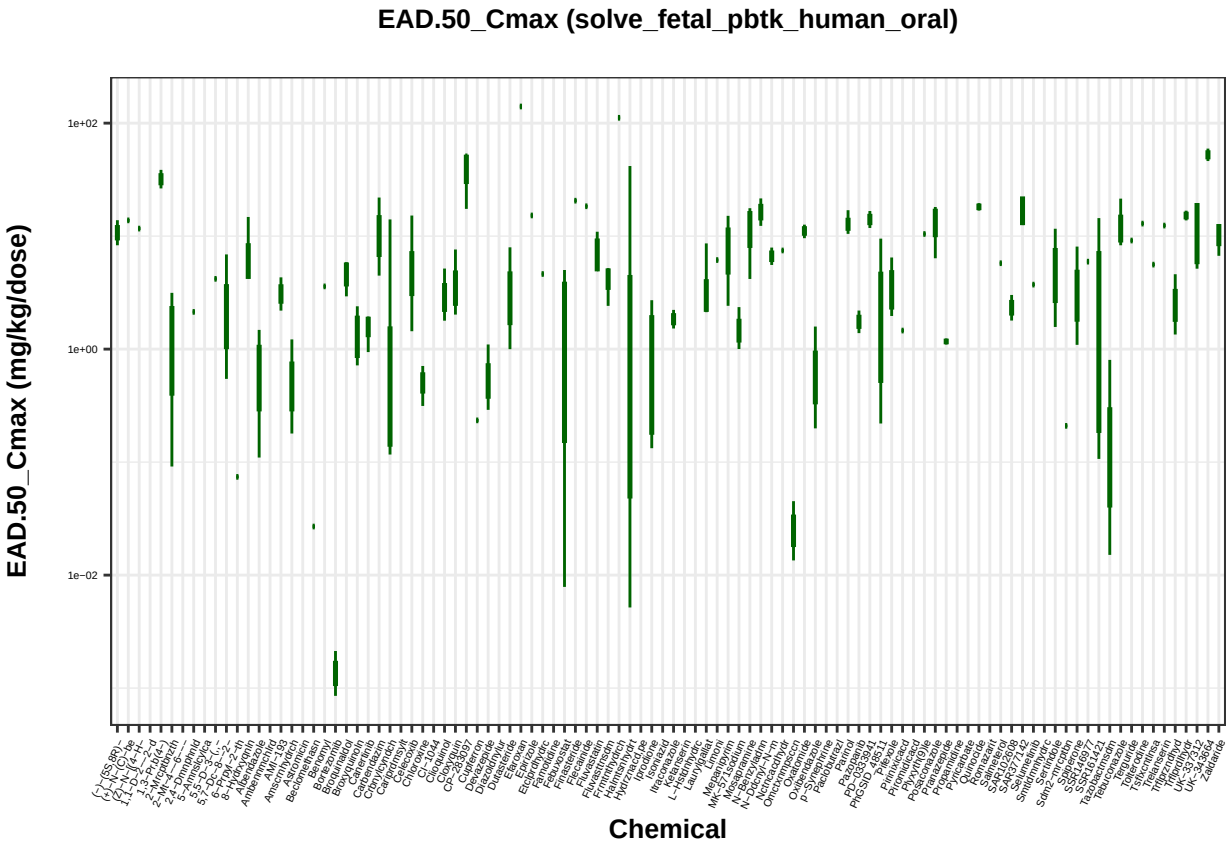
```
  Cmax2<-left_join(Cmax2, chemical %>% select(any_of(c("CASRN","Clint Source", "fu Source"))))
  Cmax2$Species<-species; Cmax2$Model<-modelType; Cmax2$"Dose, mg/kg"<-expDose;
  ssEAD.out<-EAD.out_max[,c("CASRN", setdiff(colnames(EAD.out_max), c(colnames(invitro), "Cmax")))]
  colnames(Cmax2) <- gsub("ConcentrationUnit", "Cmax, Units", names(Cmax2) )
  Cmax2$Route <- route
  Cmax2$"Dose Interval, hrs" <- interv
  Cmax2$"Length of Dosing, Days" <- ndays
  if(modelType =="solve_gas_pbtk" && gasDosing =="expConc"){        #for if inhalation/gas
    Cmax2$"Exposure Concentration"<-expConc
    Cmax2$"Exposure Concentration Unit"<-gasInputUnit
    Cmax2$"Exposure Length, hr"<-expLength
    Cmax2$"Exposure Period, hr"<-interv
    Cmax2<- Cmax2[, setdiff(colnames(Cmax2), c("Dose, mg/kg"))]
   }

  #reformatting columns to match the other models:
  Cmax2<-Cmax2 %>% select(any_of(c("CASRN", "ChemicalName", "Cmax", "Cmax, Units", "Flag", "Species","M
                                   "Route", "Dose, mg/kg","Exposure Concentration","Exposure Concentrati
                                   "Dose Interval, hrs", "Length of Dosing, Days", "Clint", "Clint Sour

  outputData<-full_join(Cmax2,ssEAD.out)
  colnames(outputData)<-gsub("Clint$", "Clint, ul/ml/10^6 cells", colnames(outputData)) #add units

  #The output file is an excel workbook. This has 2 tabs, one for the EAD results and the other for the
  xlsx::write.xlsx(file=Userout,x=outputData, sheetName ="EADResults", append=FALSE, row.names = FALSE,
  xlsx::write.xlsx(file=Userout,x=invitro, sheetName ="inVitroData", append=TRUE, row.names = FALSE, sh

  #to view the results, the "EADboxplot" function plots the values.
  #Note that if the gas model is used EADUnit needs to be changed to "uM"

  #plotting
  EADplot <- EADboxplot(EAD.out = outputData, invivo = invivo, label="EAD", EADUnit = EADUnit, species =
  #Arguments for EADplot functions include "EAD.out, invivo, label, chemDisplay, EADUnit, modelType, sp
  EADplot
}
```

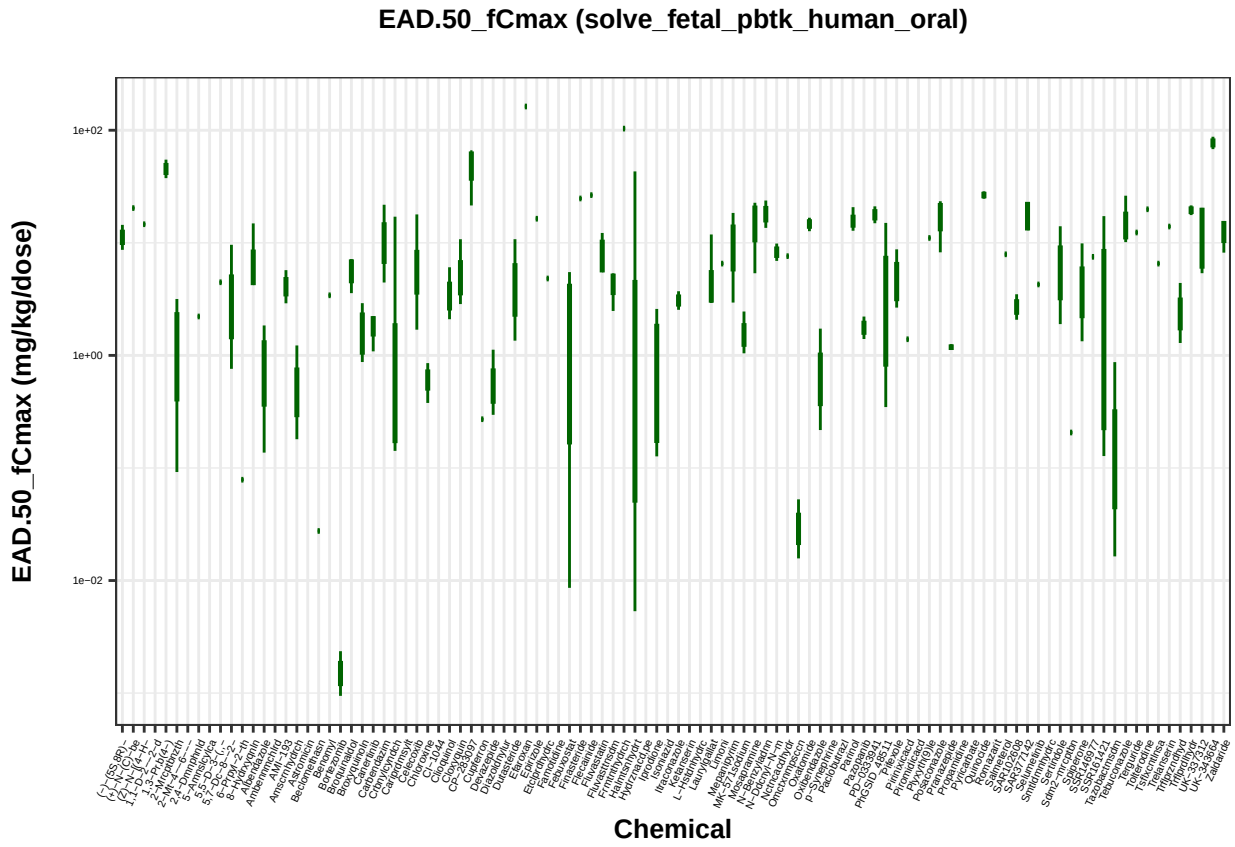
Calc EAD from solve_fetal_pbtk

The PBPK models currently give estimates based on the median (50%ile) of the population. As such there is minor formatting that needs to be done to generate the EAD predictions and make the output file

```
if(modelType == "solve_fetal_pbtk"){
  cmaxall$concMax <- as.numeric(cmaxall$concMax)
  #they are bound as characters so converting to numeric

  Cmax <- merge(chemical, cmaxall, by.x="CASRN", by.y="this.cas")
  names(Cmax) <- gsub("concMax", "Cmax", names(Cmax) )
  #Calculating the EADs corresponding to maternal and fetal plasma Cmax
  EAD.out_max <- CalcEAD(Css = Cmax[,c("CASRN", "Cmax")], inVitro = invitro)
  EAD.out50<-EAD.out_max
  colnames(EAD.out50)<-gsub("EAD","EAD.50_Cmax", colnames(EAD.out50))
  EAD.out95<-cbind(EAD.out50[1], matrix(NA, nrow=nrow(EAD.out50), ncol = ncol(EAD.out50)-1))
```

```
  colnames(EAD.out95)<-gsub("EAD.50","EAD.95_Cmax", colnames(EAD.out50)) #creating a blank data object
  #add in the flag after creating the "95" object
  EAD.out50<-left_join(chemical[,c("CASRN","Flag")],EAD.out50)
  EAD.out<-left_join(EAD.out50,EAD.out95)
  EAD.out<-EAD.out[,setdiff(colnames(EAD.out), c("adj.fu",'fu', "arm", "adj.arm"))]

  EAD.out_fmax <- CalcEAD(Css = Cmax[,c("CASRN", "fCmax")], inVitro = invitro)
  EAD.out50_fmax<-EAD.out_fmax
  colnames(EAD.out50_fmax)<-gsub("EAD","EAD.50_fCmax", colnames(EAD.out50_fmax)) #rename EAD to fetal E
  EAD.out95_fmax<-cbind(EAD.out50_fmax[1], matrix(NA, nrow=nrow(EAD.out50_fmax), ncol = ncol(EAD.out50_
  colnames(EAD.out95_fmax)<-gsub("EAD.50","EAD.95", colnames(EAD.out50_fmax)) #creating a blank data ob
  #add in the flag after creating the "95" object
  EAD.out50_fmax<-left_join(chemical[,c("CASRN","Flag")],EAD.out50_fmax)
  EAD.out_fmax<-left_join(EAD.out50_fmax,EAD.out95_fmax)
  EAD.out_fmax<-EAD.out_fmax[,setdiff(colnames(EAD.out_fmax), c("adj.fu",'fu', "arm", "adj.arm"))]

  EAD.out_both<-left_join(EAD.out, EAD.out_fmax)

  #creating output file for user, should have EAD values and the parameters for calculating the circula
  Cmax2<-as.data.frame(Cmax, stringsAsFactors=FALSE);
  #adding in the source of the parameters
  Cmax2<-left_join(Cmax2, chemical %>% select(any_of( c("CASRN","Clint Source", "fu Source"))))
  Cmax2$Species<-species; Cmax2$Model<-modelType; Cmax2$"Dose, mg/kg"<-expDose;
  ssEAD.out<-EAD.out_both[,c("CASRN", setdiff(colnames(EAD.out_both), c(colnames(invitro), "Cmax")))]
  colnames(Cmax2) <- gsub("ConcentrationUnit", "Cmax, Units", names(Cmax2) )
  Cmax2$Route <- route
  Cmax2$"Dose Interval, hrs" <- interv
  Cmax2$"Length of Dosing, Days" <- ndays
  #reformatting columns to match the other models:
  Cmax2<-Cmax2 %>% select(any_of(c("CASRN", "ChemicalName", "Cmax", "fCmax", "Cmax, Units","Flag", "Spe

  outputData<-full_join(Cmax2,ssEAD.out)
  colnames(outputData)<-gsub("Clint$", "Clint, ul/ml/10^6 cells", colnames(outputData))#add units

  #The output file is an excel workbook. This has 2 tabs, one for the EAD results and the other for the

  xlsx::write.xlsx(file=Userout,x=outputData, sheetName ="EADResults", append=FALSE, row.names = FALSE,
  xlsx::write.xlsx(file=Userout,x=invitro, sheetName ="inVitroData", append=TRUE, row.names = FALSE, sh

  #to view the results, the "EADboxplot" function plots the values.
  #Note that if the gas model is used EADUnit needs to be changed to "uM" or "ppmv"

  #plotting
  EADplot <- EADboxplot(EAD.out = outputData, invivo = invivo, label="EAD.50_Cmax", EADUnit = EADUnit, s
  fEADplot <- EADboxplot(EAD.out = outputData, invivo = invivo, label="EAD.50_fCmax", EADUnit = EADUnit
  #Arguments for EADplot functions include "EAD.out, invivo, label, chemDisplay, EADUnit, modelType, sp
  print(EADplot)
  print(fEADplot)
}
```


EAD.50_Cmax (solve_fetal_pbtk_human_oral)
EAD.50_Cmax (mg/kg/dose)
1e+02
1e+00
1e−02
Chemical



Information on the session

```
sessionInfo()
```

```
## R version 4.3.2 (2023-10-31 ucrt)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 19045)
##
## Matrix products: default
##
##
## locale:
## [1] LC_COLLATE=English_United States.utf8
## [2] LC_CTYPE=English_United States.utf8
## [3] LC_MONETARY=English_United States.utf8
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.utf8
##
## time zone: America/New_York
## tzcode source: internal
##
## attached base packages:
## [1] parallel  stats     graphics  grDevices utils     datasets  methods
```

```
## [8] base
##
## other attached packages:
##  [1] scales_1.3.0      xlsx_0.6.5        httk_2.2.2        doParallel_1.0.17
##  [5] iterators_1.0.14  foreach_1.5.2     deSolve_1.40      lubridate_1.9.3
##  [9] forcats_1.0.0     stringr_1.5.1     dplyr_1.1.4       purrr_1.0.2
## [13] readr_2.1.5       tidyr_1.3.1       tibble_3.2.1      ggplot2_3.5.1
## [17] tidyverse_2.0.0   knitr_1.48
##
## loaded via a namespace (and not attached):
##  [1] gtable_0.3.5      xfun_0.47         rJava_1.0-11      lattice_0.21-9
##  [5] tzdb_0.4.0        vctrs_0.6.5       tools_4.3.2       Rdpack_2.6.1
##  [9] generics_0.1.3    fansi_1.0.6       highr_0.11        pkgconfig_2.0.3
## [13] Matrix_1.6-1.1    data.table_1.16.0 lifecycle_1.0.4   farver_2.1.2
## [17] compiler_4.3.2    textshaping_0.4.0 munsell_0.5.1     mitools_2.4
## [21] codetools_0.2-19  survey_4.4-2      htmltools_0.5.8.1 yaml_2.3.10
## [25] pillar_1.9.0      crayon_1.5.3      tidyselect_1.2.1  digest_0.6.37
## [29] mvtnorm_1.3-1     stringi_1.8.3     splines_4.3.2     fastmap_1.2.0
## [33] grid_4.3.2        colorspace_2.1-1  expm_1.0-0        cli_3.6.2
## [37] magrittr_2.0.3    xlsxjars_0.6.1    survival_3.5-7    utf8_1.2.4
## [41] withr_3.0.1       bit64_4.5.2       timechange_0.3.0  rmarkdown_2.28
## [45] bit_4.5.0         ragg_1.3.3        hms_1.1.3         msm_1.8
## [49] evaluate_1.0.0    rbibutils_2.2.16  rlang_1.1.3       Rcpp_1.0.12
## [53] glue_1.6.2        DBI_1.2.3         rstudioapi_0.16.0 vroom_1.6.5
## [57] R6_2.5.1          systemfonts_1.1.0
```