Proença et al.:

Insights into *In Vitro* Biokinetics Using Virtual Cell Based Assay Simulations

Supplementary Data

VCBA code in R language

```
#######################DESCRIPTION OF xdot function=DIFFERENTIAL EQUATIONS ########################
xdot<-function(t,state,parameters) {
        with(as.list(c(state,parameters)), {

                #x1 total concentration in the medium
                #x2 concentration in the headspace
                #x3 concentration inside the cells

                rmax<-0
                Ksat<-0
                Cdcomp<-xx1/(1+(Ks*St)+(Kl*Lt)+(Kp*SP/V))          #Concentration of dissolved compound in mol.l-1. Eq. 4.

                #CONDITION 1) there are via cells;
                # 1) i) there is no compound entering then:  xdot3<- -kmet*xx3-weight_change*xx3
                # 1) ii) there is compound entering and so xdot3 is as represented in equation
                if(ncells>1) {
                        W<-mcells/ncells        #Wet Weight (g)
                        Vcell<-Vcells*1E6/ncells  #cm3
                        DeltaC<-Cdcomp-(xx3/(MWcomp*BCF ))

                        if(DeltaC==0.0) {
                                rexchange<-0.0
                        }else {
                                rexchange<-MWcomp*((Vcell ^(2/3))*rda*DeltaC+rmax*DeltaC/(DeltaC+Ksat))/W
                        }
                        xdot3<-rexchange-kmet*xx3-weight_change*xx3 #Function for concentration inside the cell (g.gww-1) ###
                        #compound uptake or sendit back to the medium by cells in mol.l-1.s-1
                        cells_up<--chemdead+(rexchange-kmet*xx3)*mcells/MWcomp/((1e3)*V)
                } else {
                #CONDITION 2) cells are all dead
                        xdot3<-0
                        cells_up<--chemdead
                }

                # kgcomp mass transfer coefficient on the air (m.s-1)
                # klcomp mass transfer coefficient on the water film (m.s-1)
                # KGLcomp dimensionless gas-liquid distribution coefficient.
                Fawcomp<-kaw*(-Cdcomp+xx2/KGLcomp)      # diffusive air-water exchange (mol.m-2.s-1)
                Fdecomp<-kdeccomp*Cdcomp #medium decomposition in mol.l-1.s-1
                Fdecompa<-kdecacomp*xx2 #headspace decomposition in mol.l-1.s-1
                Flosses<-Fexch*xx2 #headspace losses. For Fexch is set to 0, this is 0, but can be used when modeling cross contamination

                                                                        ########## Equation for total concentration the medium (mol.l-1.s-
1) #########
                xdot1<-(P*Fawcomp-Fdecomp-cells_up)
                                                                        ########### Equation for concentration in the headspace (mol.l-
1.s-1) #########
                xdot2<-(-Ph*Fawcomp-Fdecompa-Flosses)
                list(c(xdot1,xdot2,xdot3))
        })
}
#############################################################################
```

doi:10.14573/altex.1812101s1

```
END OF DIFFERENTIAL EQUATIONS
##############################################################################################

library(deSolve) #Program to solve differential equations

##################################### DESCRIPTION OF CONSTANTS ##########################

######## WELL GEOMETRY ############
# To return AssayVol, P, Ph and SP

PI<-pi
#Well Description in meters
diaTop<-0.00685                                                    #Diameter of the top of the well
diaBot<-0.00635                                                    #Dimater of the bottom of the well
Depth<-0.01076                                                     #Depth of the well
V<-1E-7                                                            #Volume of medium in the well
##Equations
H<-diaTop*Depth/(diaTop-diaBot)
h<-H-Depth
#Total Volume well:
vVessel<-(1/3)*PI*H*((diaTop*0.5)^2)-(1/3)*PI*h*((diaBot*0.5))^2


vt<-PI*h*((0.5*diaBot)^2)/3
hh<-((V+vt)*12*h*h/(PI*diaBot*diaBot))^(1/3)
xx<-diaBot*hh*0.5/h
llt<-(hh^2+xx^2)^(1/2)
lli<-(h^2+(diaBot*0.5)^2)^(1/2)
SPt<-PI*xx*llt
SPi<-PI*diaBot*0.5*lli
#surface of the plastic in contact with the medium
SP<-SPt-SPi+PI*((diaBot*0.5)^2)

#cell assay surface:
AS<-PI*(xx^2)
P<-AS/V #element to simplify differential equation xdot1
#head space volume (m3)
Vh=vVessel-V
Ph<-AS/Vh #element to simplify differential equation xdot2

######## AIR-WATER EXCHANGE #########
# To return kdeccomp, kdecacomp, kgcomp,KGLcomp, Klcomp and kaw
# Constant values introduced knime
kdeccomp<-2.14E-07                                                 #Compound's Rate of Degradation in water in s-1
kdecacomp<-9.21E-07                                                #Compound's Rate of Degradation in water in s-1
Te<- 37+273.15                                  #Air temperature from Celsius to K. Experiment temperature constant at 37 C.
MWair<-28.8                                                                  #Air molecular weight
MWcomp<-78.1                                                                 #Compound Molecular Weight
Press<-1                                                           #Air Pressure in atm.
Svair<-20.1                                                                  #atomic diffusion air
Svcomp<-90.96                                                      #atomic diffusion
RR<-8.3144                                                         #Universal gas constant kJ(mol.K)-1
H37<-5.62E+02                                                  #Henry´s constant at 37 C
vb<-89.40                                                      #molar volume at its normal boiling point (cm^3/g mol)
kgH2O<-3.0e-3                                                              #mass transfer coefficient for water m*s-1
kLCO2<-4.1e-2                                                  #mass transfer coefficient of CO2 in the Water side
fi<-2.6                                                        #association factor or organic solutes diffusing into water

#Parameter for the exchange of compound between wells. Null for now.
Fexch<-0

##Evaporation and Degradation equations
#Organic compound gas phase diffusion coefficient (m2/s). From Fuller 1966
valgas<-1E-7*((MWair+MWcomp)/(MWair*MWcomp))^0.5
valgas<-valgas/(Press*((Svair^0.33)+(Svcomp^0.33))^2)
DGcomp<-valgas*Te^(1.75)                                       #Diffusion coefficient in the air.
DGw<-1.23655e-9*Te^(1.75)                                      #Diffusion coefficients for water in air
kgcomp<-kgH2O*(DGcomp/DGw)^0.67                                #Gas phase mass transfer coefficient.

KGLcomp<-H37/(RR*Te)                                           #Gas-Liquid partition Coefficient (dimensionless)
```

```
muw<-0.6913                                                    #simplified water viscosity in cP at 37C.
denw<-0.933                                                    #simplified water density in gr/cm3 at 37C.
Fi<-7.4E-12*(fi*MWcomp)^0.5/(vb)^0.6
DLcomp<-Fi*Te/muw                                              #Liquid phase diffusion coefficients (m2/s).
SCAcomp<-muw*1e-3/(1e3*denw*DLcomp)                                                    # Schimdt number.
klcomp<-kLCO2*(SCAcomp/600)^(-0.5)                             #mass transfer coefficient on the water film .

kaw<-(kgcomp*KGLcomp*klcomp)/(klcomp+kgcomp*KGLcomp)           #mass transfer coefficient.


######### BCF AND PARTITION #########
# To return mass fractions and densities fo each component Kp, Ks, Kl, KL, KP, BCF and rad

logkow<-2.31 #logarithm of partition octanol-water

#mass fractions in % weight
faq<-0.614              #3T3 cell line  aqueous fraction
fL<-0.142  #3T3 cell line  lipidic fraction
fP<-0.244  #3T3 cell line  proteic fraction

#densities in g/L
rhoaq<-1000             #aqueous phase density
rhoL<-900               #lipid phase density
rhoP<-1350              #proteic phase density

#Lipid and protein in medium serum. Mind the % of serum supplementing the medium
S0<-0.0234 #protein content for 5 % serum
L0<-0.08 #lipid content for 5% serum

##Equations for partition

#plastic well partition
Kp<-10^(0.97*logkow-6.94)

#serum well partition
if (logkow <1.09) {
                        vals<-1.31
            } else if (logkow>=1.09&&logkow<=4.6) {
            vals<-0.57*logkow+0.69
            } else if (logkow>4.6) {
                        vals<-logkow-1.3
            }
Ks<-10^(vals-1.178)

#lipid partition
Kl<-10^(1.25*logkow-3.70)

##cell partition##
LC<-170.4
SC<-4.4
KL<-LC*Kl
KP<-SC*Ks

#Bioconcentration factor
#Does not contain Lipid and protein concentration as these parameters are already contained in KL and KP
BCF<-(faq/rhoaq)+(fL*KL/rhoL)+(fP*KP/rhoP)
alldiss<-function(Lt=Lt,St=St) {
            disss<-(1+(Ks*St)+(Kl*Lt)+(Kp*SP/V))
            disscs<-(1+(Ks*St)+(Kl*Lt)+(Kp*SP/V))/(Ks*St)
            disscl<-(1+(Ks*St)+(Kl*Lt)+(Kp*SP/V))/(Kl*Lt)
            disscp<-(1+(Ks*St)+(Kl*Lt)+(Kp*SP/V))/(Kp*SP/V)
            allinone<-c(disss,disscs,disscl,disscp)
return(allinone)
}
###Cell Permeability Equations###
logp<--1.1711+0.98*logkow-0.0011*MWcomp # QSAR for log cell permeability(cm*h-1)
Per<-10^logp # Cell permeability still in cm*h-1
rda<-Per/36000 # Rate of uptake=Cell permeability from cm*h-1 to L*cm-2*s-1
#Herein is considered rate of uptake is the same of elimination. If this assumption change parameters have to be revised
```

```
####Metabolism####
kmet<-0 #3T3 are considered metabolic imcompetent cells

################################## END CONSTANTS DESCRIPTION##############################


coreModel<-function(ci,cell_i,numCells,nec,kt) {

#Parameters for leslie matrix. Already structured in arrays of 1 row and 4 columns
di<-c(9.63,3.65,3.45,2.26) # duration at each stage in min
zi<-c(0.005,0.005,0.04,0.04) #mortality at each stage
di<-array(di,dim=c(1,4))
zi<-array(zi,dim=c(1,4))

##Initial values for the time loop. Once the program runs once the loop, these values stop being used and actualised for the time run(kl)
numCells<-1680 ##Initial cell number
ncells<-numCells
N<-c(50.7,19.2,18.1798,11.9202)*ncells/100 #fraction of cells in each cell cycle phase
N<-array(N,dim=c(4))
mcells_old<-2.08E-9*N[1]+2.4E-9*N[2]+2.4E-9*N[3]+2.4E-9*N[4] #Cell Mass per cell cycle phase In g
mcells<-mcells_old
Vcells<-1.73E-15*N[1]+2.4E-15*N[2]+2.4E-15*N[3]+2.4E-15*N[4] #Cell volume per cell cycle phase in m^3
Lt<-L0
St<-S0
weight_change<-0
chemdead<-0
x1<-ci
x2<-0
x3<-cell_i

#Definition of the Duration
totalTime<-48 #Time in h
kl_fin<-totalTime #in h
val<-array(kl_fin) #array for cell growth
dead_cells<-array(kl_fin) #array for cell death


######################################## TIME CYCLE ##############################

for (kl in 1:kl_fin) {

t<-seq((kl-1)*3600,kl*3600,by=1) #time in seconds but making steps of 60 s

##Differential equations solving
parameters<-
c(Ks=Ks,St=St,Kl=Kl,Kp=Kp,Lt=Lt,SP=SP,V=V,kgcomp=kgcomp,KGLcomp=KGLcomp,klcomp=klcomp,ncells=ncells,mcells=mcells,Vcells=Vcells,faq=faq,rhoaq=rhoaq,fL=fL,K
L=KL,rhoL=rhoL,fP=fP,KP=KP,rhoP=rhoP,MWcomp=MWcomp,rda=rda,kmet=kmet,weight_change=weight_change,chemdead=chemdead,kdeccomp=kdeccomp,P=P,kdecaco
mp=kdecacomp,Fexch=Fexch,Ph=Ph)
state<-c(xx1=x1,xx2=x2,xx3=x3)
out<-ode(y=state,times=t,func=xdot,parms=parameters,method="radau",atol=1e-4,rtol=1e-4,hmax=1)
out_row<-length(out[,1])
out_col<-length(out[1,])
out_<-matrix(data=out,nrow=out_row,ncol=out_col)


##Mortality NEC and Kt
cq<-mean(out_[,4])
val2 <- cq - nec
val1<-max(0,val2)
za<-zi+kt*val1
pii<-exp(-za)
gamma<-(1-pii)*(pii^(di-1))/(1-pii^di)
PS<-pii*(1-gamma)
GS<-pii*gamma

##density dependence HepG2. Needs to be revised for other cell lines may also have density dependence. To run take the commen-out
#if(cellType =="HepG2") {
#Fii<-3.8062
#dvol<-35.0676e-5
```

```
#F<-Fii*exp(-ncells/dvol)
#} else {
F<-1.026 #optimised fecundity
#}

##Leslie Matrix##

numSteps<-4 #how many cell phases the cell line has; Ex:3T3 has 4 and HepaRG 1.

if(numSteps==1) {
dead_cells[kl]<-N[1]*(1-(PS[1]+GS[1]))
LE<-c(PS[1])
LE<-array(LE,dim=c(1,1))
N[1]=LE*N[1]
#The PS and GS for other cell phases are set to 0 so in valdead it will not count with these phases
PS[2:4]<-0
GS[1:4]<-0
}
#Because the cell we are using here is 3T3 and it has 4 phases, then it will run this option
if(numSteps==4) {
dead_cells[kl]<-N[1]*(1-(PS[1]+GS[1]))+N[2]*(1-(PS[2]+GS[2]))+N[3]*(1-(PS[3]+GS[3]))+N[4]*(1-(PS[4]+GS[4]))
LE<-c(PS[1],GS[1],0,0,0,PS[2],GS[2],0,0,0,PS[3],GS[3],F,0,0,PS[4])
LE<-array(LE,dim=c(4,4))
N<-LE%*%N
}

##Number of cells at each moment##
val[kl]<-sum(N)
ncells<-val[kl]

##When all cells die
if(ncells<1) {
ncells<-0
out_[,4]<-0
Wcells<-0
mcells<-0
Vcells<-0

} else {

mcells<-2.08E-9*N[1]+2.4E-9*N[2]+2.4E-9*N[3]+2.4E-9*N[4] # recalculate the total mass of cells
Vcells<-1.73E-15*N[1]+2.4E-15*N[2]+2.4E-15*N[3]+2.4E-15*N[4] # recalculate the total volume of cells
Wcell<-mcells/ncells
weight_change<-(mcells-mcells_old)/mcells/ncells/3600
mcells_old<-mcells    #So for the new cycle the herein calculated mcells will be the mcells(kl-1)=mcells_old

##Return of chemical, lipid and protein to medium from cells death##
chemdead<-dead_cells[kl]*out_[out_row,4]*Wcell*1E-3/MWcomp/V/3600

}
valdead<-N[1]*(1-(PS[1]+GS[1]))*2.08E-9+N[2]*(1-(PS[2]+GS[2]))*2.40E-9+N[3]*(1-(PS[3]+GS[3]))*2.4E-9+N[4]*(1-
(PS[4]+GS[4]))*2.4E-9
Lt<-Lt+(fL*valdead*1E-3)/V
#albumin MW= 66400 g/mol
St<-St+(fP*valdead)/66400/V
#output of the differential equations and for cycle (concentration in air, medium and intracellularly)
x0<-out_[out_row,1]
x1<-out_[out_row,2]
x2<-out_[out_row,3]
x3<-out_[out_row,4]
            }
# Start new cycle

return(c(x1,x2,x3,ncells,Lt,St))
}

################################## END LOOP ##################################
###Description of the procedures for different Modes, can only run one mode at a time.

#Description experimental viability curve
```

```
ExpCon<-c(0,0.023,0.040,0.072,0.128,0.227,0.404,0.719,1.280)                    #Concentrations in M .
ExpVia<-c(100,100.92,96.30,89.44,78.45,60.56,39.78,26.86,16.15)                 #Viability in %
StDExp<-c(0,11.31,6.11,23.43,33.83,47.05,47.51,25.92,14.24)                     #Standard deviation of each point (just for plotting)


##Mode: OPTIMIZATION

##
vallerror<-function(initialValues) {
sol<-array(0,dim=c(NumberRow,6))
viaT<-array(0,dim=c(NumberRow,1))
NEC<-initialValues[1]
kt<-initialValues[2]
valerror<-0.0 #Initial value of error value

#Loop for each experimental concentration-viability comparison with predicted
for(i in 1:NumberRow) {
sol[i,1:6]=coreModel(ci=ExpCon[i],cell_i=0,numCells=numCells,nec=NEC,kt=kt)
viaT[i]<-sol[i,4]/sol[1,4]*100 #Final viability in %
valerror<-valerror+(viaT[i]-ExpVia[i])^2
}
print(c(error=valerror,nec=NEC,kt=kt))
return(valerror)
}
#optim is a defined R function for optimization.
#First input is the initial values to be optimised.
#Second is the function to be minimised, function that has descriminated the values to be optimised.Then the method of optimization and the lower values
possible.
#This optimization can take days . an initial testing of approximated values can be made mannually and ranges adjusted so the optimization does not drift to
non suitable values
output<-optim(c(0.03,0.135),vallerror,method='L-BFGS-B',lower=c(0.007,0.07),upper=c(0.09,0.25))


## MODE: Run a single concentration with previously optimised nec and kt to obtain the final concentrations in the several compartments
res<-array(0,dim=c(1,6))
##
singleExposure<-function() {
conc<-0.231276405                                #Concentration in M . Can be the IC50
res[1,1:6]<-coreModel(ci=conc,cell_i=0,numCells=1680,nec=0.0070000 ,kt=0.1005614)
return(res)
}
### SELECTION OF DIFFERENT MODES ###
modeSel<-"Single Exposure"

if(modeSel=="Single Exposure") {
#if(knime.flow.in[["modeSel"]]=="Single Exposure") {
        final<-singleExposure()
all_diss<-alldiss(Lt=final[,5],St=final[,6])
cdiss<-as.double(all_diss[1])
disscs<-as.double(all_diss[2])
disscl<-as.double(all_diss[3])
disscp<-as.double(all_diss[4])
caq<-final[,3]/MWcomp/BCF
final<-
data.frame(name="Benzene",Dissolved=(final[1,1]/cdiss),protein=(final[1,1]/disscs),lipid=(final[1,1]/disscl),plastic=(final[1,1]/disscp),BCF=BCF,ct=final[1,1],head
space=final[1,2],conInside=final[1,3])
}

#####MODE:Description of plots#####



##Chart viability vs concentration (experimental & predicted)##
conc<-seq(0,1.3,by=0.3E)#concentrations to me simulated

#Description of function for plots#
Graphs<-function (ci=ci){
        FData<-array(0,dim=c(48,6))
        FData[,]<-coreModel(ci=ci,cell_i=0,numCells=1680,nec=0.00063,kt=35.001) # adjust to the optimised nec and kt
        ViaCells<-FData[,4]                                                     #% Viability along C´
```

```
        RelNCells<-FData[,4]/1680                                   #Relative Nr cells
        all_diss<-alldiss(Lt=FData[,5],St=FData[,6])
        DissC<-(FData[,1]/all_diss[1])                              #Dissolved concentration
        Intra<-FData[,3]
        LipC<-(FData[,1]/all_diss[3])
        ProteC<-(FData[,1]/all_diss[2])                             #Intracellular Concentration
        Fin<-cbind(ViaCells,RelNCells,DissC,Intra,LipC,ProteC)
        return(Fin)
}

#Defining cell viability in %#
PredV<-function(CI=CI){
        Pred<-array(0,dim=c(1,length(conc)))
        Pred<-Graphs(ci=CI)[48,1]/Graphs(ci=conc[1])[48,1]*100
        return(Pred)
        }

##Charts frame##
par(mfrow=c(3,2))

##plot cell viability  vs concentrations##

plot(ExpCon,ExpVia,ylim=c(0,100),type="p",pch=19,col=2,cex=1.3,xlab="Concentration(M)",ylab="% Viability",cex.lab=1.25)
arrows(ExpCon,ExpVia-StDExp,ExpCon,ExpVia+StDExp,col=2,length=0.05,angle=90,code=3,cex=1)#Error bars
PredVia<-array(0,dim=c(1,length(conc)))
for (con in 1:length(conc)){
        PredVia[1,con]<-PredV(CI=conc[con])
}
lines(conc,PredVia,lwd=2)

                #Definition of Color Gradient##
colvector<-colorRampPalette(c("lightskyblue","dodgerblue4"))
Colvector<-array(colvector(length(conc)),dim=c(1,length(conc)))

                ##plot N cells vs Time##
plot(1:48,Graphs(ci=conc[1])[,2],type="l",ylim=c(0,6),xlab="time (s)",ylab="Relative n°Cells ",cex.lab=1.25)
for(con in 1:length(conc)) {
        lines(1:48,Graphs(ci=conc[con])[,2],col=Colvector[con],lwd=1.5)
}
                ##plot Dissolved concentration vs time##
plot(1:48,Graphs(ci=conc[1])[,3],type="l",xlab="time (s)",ylab="Dissolved Concentration (M) ",cex.lab=1.25)
for(con in 1:length(conc)) {
        lines(1:48,Graphs(ci=conc[con])[,3],col=Colvector[con],lwd=1.5)
}
                ##plot Intracellular concentration vs time#
plot(1:48,Graphs(ci=conc[1])[,4],type="l",xlab="time (s)",ylab="Intracellular Concentration (g/g ww) ",cex.lab=1.25)
for(con in 1:length(conc)) {
        lines(1:48,Graphs(ci=conc[con])[,4],col=Colvector[con],lwd=1.5)
}
                ##plot Lipid concentration vs time#
plot(1:48,Graphs(ci=conc[1])[,5],type="l",xlab="time (s)",ylab=" [Chemical] bound to Lipid (M) ",cex.lab=1.25)
for(con in 1:length(conc)) {
        lines(1:48,Graphs(ci=conc[con])[,5],col=Colvector[con],lwd=1.5)
}
                ##plot Protein concentration vs time#
plot(1:48,Graphs(ci=conc[1])[,6],type="l",xlab="time (s)",ylab=" [Chemical] bound to Protein (M) ",cex.lab=1.25)
for(con in 1:length(conc)) {
        lines(1:48,Graphs(ci=conc[con])[,6],col=Colvector[con],lwd=1.5)
}
```