PBPK in PuMaS, A Model for ACAT

Vaibhav Dixit and Chris Rackauckas

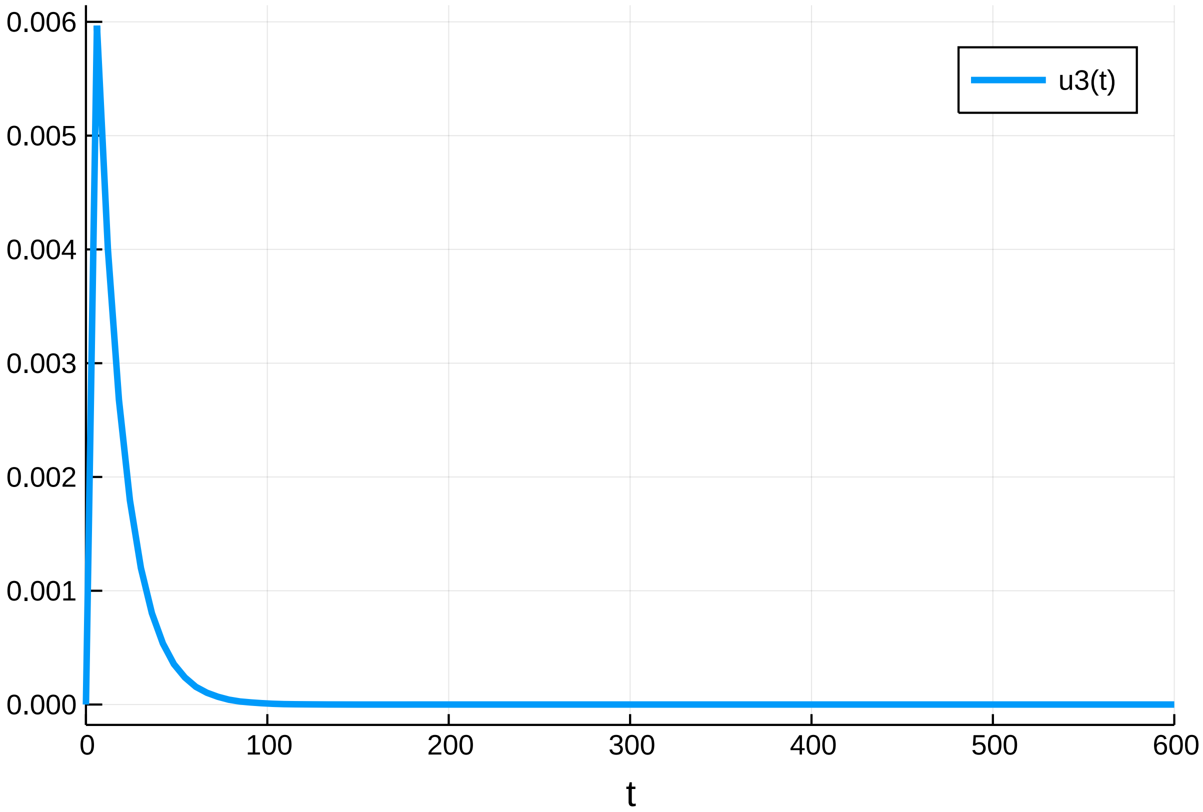
6th November 2018

1 Introduction

2 Code

```
using PuMaS, LinearAlgebra, DiffEqSensitivity, Distributions, Optim, QuadGK

pbpkmodel = @model begin
    @param begin
        GER ∈ ConstDomain(0.066)
        ρ ∈ ConstDomain(5e-6)
        r ∈ ConstDomain(1)
        T ∈ ConstDomain(3e-5)
        d ∈ ConstDomain(1e-4)
        SST ∈ ConstDomain(5.5)
        kilST ∈ ConstDomain(0.5)
        kaST ∈ ConstDomain(14040.00076)
        kaGU ∈ ConstDomain(14040.000063)
        kt ∈ ConstDomain(0.035)
        SGU1 ∈ ConstDomain(5.5)
        SGU2 ∈ ConstDomain(5.5)
        SGU3 ∈ ConstDomain(5.5)
        SGU4 ∈ ConstDomain(5.5)
        SGU5 ∈ ConstDomain(5.5)
        SGU6 ∈ ConstDomain(5.5)
        SGU7 ∈ ConstDomain(5.5)
        kilGU1 ∈ ConstDomain(0.5 )
        kilGU2 ∈ ConstDomain(0.5)
        kilGU3 ∈ ConstDomain(0.5)
        kilGU4 ∈ ConstDomain(0.5)
        kilGU5 ∈ ConstDomain(0.5)
        kilGU6 ∈ ConstDomain(0.5)
        kilGU7 ∈ ConstDomain(0.5)
        EHR ∈ ConstDomain(0 )
        kbil ∈ ConstDomain(0.0)
        VLI ∈ ConstDomain(1690)
        Kp ∈ ConstDomain(1.3)
        ktCO ∈ ConstDomain(0.0007)
        SCO ∈ ConstDomain(5.5)
        VCO ∈ ConstDomain(700)
        kilCO ∈ ConstDomain(0.0007)
        kaCO ∈ ConstDomain(14040.0000542)
        CP ∈ ConstDomain(0)
        QLU ∈ ConstDomain(5233)
    VLU ∈ ConstDomain(1172)
    VST1 ∈ ConstDomain(50)
    VST2 ∈ ConstDomain(154)
    VGU ∈ ConstDomain(1650)
    VAR ∈ ConstDomain(1698)
    AIR ∈ ConstDomain(0.0)
    VVE ∈ ConstDomain(3396)
    VIR ∈ ConstDomain(0.0)
    QBR ∈ ConstDomain(700)
    VBR ∈ ConstDomain(1450)
    QLI ∈ ConstDomain(1650)
    QKI ∈ ConstDomain(1100)
    QHR ∈ ConstDomain(150)
    VHR ∈ ConstDomain(310)
    QMU ∈ ConstDomain(750)
    VMU ∈ ConstDomain(35000)
    QAD ∈ ConstDomain(260)
    VAD ∈ ConstDomain(10000)
    QSK ∈ ConstDomain(300)
    VSK ∈ ConstDomain(7800)
    QBO ∈ ConstDomain(250)
    VBO ∈ ConstDomain(4579)
    QTH ∈ ConstDomain(80)
    VTH ∈ ConstDomain(29)
    QST ∈ ConstDomain(38)
    QGU ∈ ConstDomain(1100)
    Ker ∈ ConstDomain(0.0)
    QPA ∈ ConstDomain(133)
    VPA ∈ ConstDomain(77)
    QSP ∈ ConstDomain(77)
    VSP ∈ ConstDomain(192)
    CLint ∈ ConstDomain(0)
    QHA ∈ ConstDomain(302)
    VKI ∈ ConstDomain(280)
    R ∈ ConstDomain(1)
end

@random begin
    η ~ MvNormal(Matrix(1.0I,2,2))
end

@dynamics begin
    #Absorption compartments
    #Stomach compartment
    AUNDST' = -GER * AUNDST - (((3*d)/(ρ*r*T)) * AUNDST* (SST - (ADIST/VST1)))
    ADIST' = -GER * ADIST + (((3*d)/(ρ*r*T)) *AUNDST*(SST - (ADIST/VST1))) - kilST*
ADIST -kaST *ADIST
    ADEGST' = -GER * ADEGST + kilST * ADIST
    AABSST' = kaST * ADIST

    #GU1 small intestinal compartment
    AUNDGU1' = GER * AUNDST - kt * AUNDGU1 - ((3*d)/(ρ*r*T)) * AUNDGU1 *(SGU1 -
(ADISGU1/VGU))
    ADISGU1' = GER * ADIST - kt * ADISGU1 + ((3*d)/(ρ*r*T)) * AUNDGU1 *(SGU1 -
(ADISGU1/VGU)) - kilGU1*ADISGU1 - kaGU*ADISGU1 + (EHR*kbil*CLI *VLI)/( Kp)
    ADEGGU1' = GER * ADEGST - kt * ADEGGU1 + kilGU1 * ADISGU1
    AABSGU1' = kaGU * ADISGU1

    # Other small intestinal compartments (GU2-GU7)
    AUNDGU2' = kt * AUNDGU1 - kt * AUNDGU2 -((3*d)/(ρ*r*T)) * AUNDGU2 *(SGU2 -
(ADISGU2/VGU))
    ADISGU2' = kt * ADISGU1 - kt * ADISGU2 +((3*d)/(ρ*r*T)) * AUNDGU2 *(SGU2 -
(ADISGU2/VGU)) - kilGU2*ADISGU2 - kaGU*ADISGU2
    ADEGGU2' = kt*ADEGGU1 - kt*ADEGGU2 + kilGU2 * ADISGU2
    AABSGU2' = kaGU * ADISGU2

    AUNDGU3' = kt * AUNDGU2 - kt * AUNDGU3 -((3*d)/(ρ*r*T)) * AUNDGU3 *(SGU3 -
(ADISGU3/VGU))
    ADISGU3' = kt * ADISGU2 - kt * ADISGU3 +((3*d)/(ρ*r*T)) * AUNDGU3 *(SGU3 -
(ADISGU3/VGU)) - kilGU3*ADISGU3 - kaGU*ADISGU3
    ADEGGU3' = kt*ADEGGU2 - kt*ADEGGU3 + kilGU3 * ADISGU3
    AABSGU3' = kaGU * ADISGU3

    AUNDGU4' = kt * AUNDGU3 - kt * AUNDGU4 -((3*d)/(ρ*r*T)) * AUNDGU4 *(SGU4 -
(ADISGU4/VGU))
    ADISGU4' = kt * ADISGU3 - kt * ADISGU4 +((3*d)/(ρ*r*T)) * AUNDGU4 *(SGU4 -
(ADISGU4/VGU)) - kilGU4*ADISGU4 - kaGU*ADISGU4
    ADEGGU4' = kt*ADEGGU3 - kt*ADEGGU4 + kilGU4 * ADISGU4
    AABSGU4' = kaGU * ADISGU4

    AUNDGU5' = kt * AUNDGU4 - kt * AUNDGU5 -((3*d)/(ρ*r*T)) * AUNDGU5 *(SGU5 -
(ADISGU5/VGU))
    ADISGU5' = kt * ADISGU4 - kt * ADISGU5 +((3*d)/(ρ*r*T)) * AUNDGU5 *(SGU5 -
(ADISGU5/VGU)) - kilGU5*ADISGU5 - kaGU*ADISGU5
    ADEGGU5' = kt*ADEGGU4 - kt*ADEGGU5 + kilGU5 * ADISGU5
    AABSGU5' = kaGU * ADISGU5

    AUNDGU6' = kt * AUNDGU5 - kt * AUNDGU6 -((3*d)/(ρ*r*T)) * AUNDGU6 *(SGU6 -
(ADISGU6/VGU))
    ADISGU6' = kt * ADISGU5 - kt * ADISGU6 +((3*d)/(ρ*r*T)) * AUNDGU6 *(SGU6 -
(ADISGU6/VGU)) - kilGU6*ADISGU6 - kaGU*ADISGU6
    ADEGGU6' = kt*ADEGGU5 - kt*ADEGGU6 + kilGU6 * ADISGU6
    AABSGU6' = kaGU * ADISGU6

    AUNDGU7' = kt * AUNDGU6 - kt * AUNDGU7 -((3*d)/(ρ*r*T)) * AUNDGU7 *(SGU7 -
(ADISGU7/VGU))
    ADISGU7' = kt * ADISGU6 - kt * ADISGU7 +((3*d)/(ρ*r*T)) * AUNDGU7 *(SGU7 -
(ADISGU7/VGU)) - kilGU7*ADISGU7 - kaGU*ADISGU7
    ADEGGU7' = kt*ADEGGU6 - kt*ADEGGU7 + kilGU7 * ADISGU7
    AABSGU7' = kaGU * ADISGU7

    # Colon compartment
    AUNDCO' = kt * AUNDGU7 - ktCO * AUNDCO - ((3*d)/(ρ*r*T)) * AUNDCO * (SCO -
(ADISCO/VCO))
    ADISCO' = kt * ADISGU7 - kt * ADISCO + ((3*d)/(ρ*r*T)) * AUNDCO * (SCO -
(ADISCO/VCO)) - kilCO*ADISCO -  kaCO*ADISCO + (CP*CLI*VLI*kbil)/( Kp)
    ADEGCO' = kt * ADEGGU7 - ktCO * ADEGGU7 + kilCO * ADISCO
    AABSCO' = kaCO * ADISGU7

    #Total intestinal absorption (IA)
    AIA' = kaGU*ADISGU1 + kaGU*ADISGU2 + kaGU*ADISGU3 + kaGU*ADISGU4 + kaGU*ADISGU5
+ kaGU*ADISGU6 + kaGU*ADISGU7

    #Somatic Compartments
    # Lungs
    CLU' = (QLU/VLU) *(CVE - (CLU*R)/(Kp))

    #Arterial blood (AR)
```

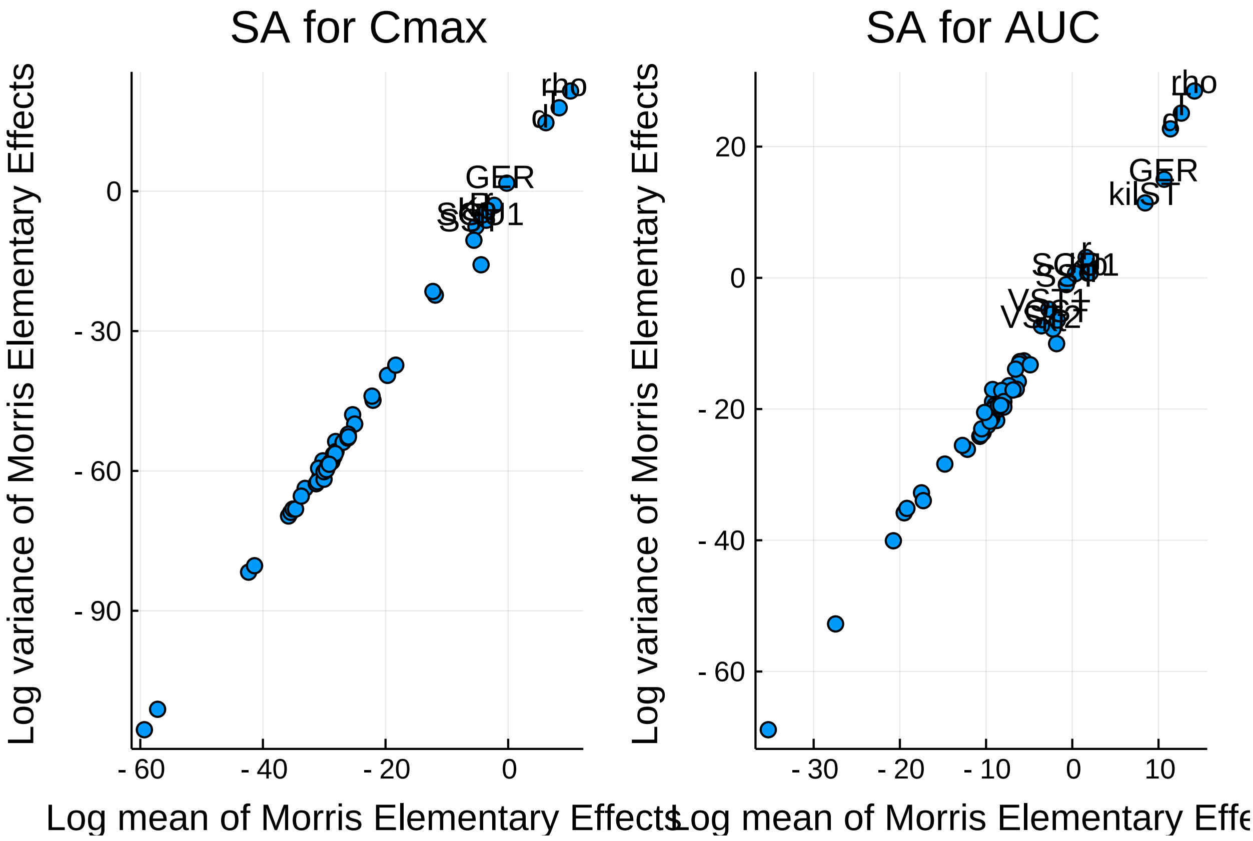
```
        CAR' = (1/VAR) * (QLU*(((CLU*R)/(Kp)) -CAR) + AIR)

        # Venous blood (VE)
        CVE' = (1/VVE) * (((QBR *CBR*R)/( Kp))  + ((QLI *CLI*R)/( Kp)) + ((QKI *CKI*R)/(
    Kp)) + ((QHR *CHR*R)/( Kp)) + ((QMU *CMU*R)/( Kp)) + ((QAD *CAD*R)/( Kp)) + ((QSK
    *CSK*R)/( Kp)) + ((QBO *CBO*R)/( Kp)) +((QTH *CTH*R)/( Kp)) - QLU * CVE + VIR)

        #Brain
        CBR' = (QBR/VBR)* (CAR - (CBR*R)/( Kp))
        #Heart
        CHR' = (QHR/VHR)* (CAR - (CHR*R)/( Kp))
        #Muscle
        CMU' = (QMU/VMU)* (CAR - (CMU*R)/( Kp))
        #Adipose
        CAD' = (QAD/VAD)* (CAR - (CAD*R)/( Kp))
        #Skin
        CSK' = (QSK/VSK)* (CAR - (CSK*R)/( Kp))
        #Bone
        CBO' = (QBO/VBO)* (CAR - (CBO*R)/( Kp))
        #Thymus
        CTH' = (QTH/VTH)* (CAR - (CTH*R)/( Kp))
        #Pancreas
        CPA' = (QPA/VPA)* (CAR - (CPA*R)/( Kp))
        #Spleen
        CSP' = (QSP/VSP)* (CAR - (CSP*R)/( Kp))

        #Stomach
        CST' = (1/VST2) * (QST*(CAR - ((CST*R)/(Kp))) + AABSST)
        #Gut
        CGU' = (1/VGU) * (QGU*(CAR - ((CGU*R)/(Kp))) + AIA)
        #Kidney
        CKI' = ((1/VKI) * QKI*(CAR - ((CKI*R)/(Kp)))) - ((CKI*Ker)/Kp)
        #Liver
        CLI' = (1/VLI)*(QHA*CAR + ((QGU *CGU*R)/( Kp)) + ((QPA *CPA*R)/( Kp)) + ((QSP
    *CSP*R)/( Kp)) + ((QST *CST*R)/( Kp)) - ((QLI *CLI*R)/( Kp)) - (CLI*CLint)/Kp )
    end
end

subject = Subject(evs = DosageRegimen(250,cmt=[1],time=[0.0]))
param = (GER = 0.066,ρ = 5e-6,r = 1,T = 3e-5,d = 1e-4,SST = 5.5,kilST = 0.5,kaST =
    14040.00076,kaGU = 14040.000063,kt = 0.035,SGU1 = 5.5,SGU2 = 5.5,SGU3 = 5.5,SGU4 =
    5.5,SGU5 = 5.5,SGU6 = 5.5,SGU7 = 5.5,kilGU1 = 0.0 ,kilGU2 = 0.0,kilGU3 = 0.0,kilGU4
    = 0.0,kilGU5 = 0.0,kilGU6 = 0.0,kilGU7 = 0.0,EHR = 0 ,
        kbil = 0.0,VLI = 1690,Kp = 1.3,ktCO = 0.0007,SCO = 5.5,VCO = 700,kilCO =
    0.0007,kaCO = 14040.0000542,CP = 0,QLU = 5233,VLU = 1172,VST1 = 50,VST2 = 154,VGU =
    1650,VAR = 1698,AIR = 0.0,VVE = 3396,VIR = 0.0,QBR = 700,VBR = 1450,QLI = 1650,QKI =
    1100,QHR = 150,VHR = 310,QMU = 750,VMU = 35000,QAD = 260,
        VAD = 10000,QSK = 300,VSK = 7800,QBO = 250,VBO = 4579,QTH = 80,VTH = 29,QST =
    38,QGU = 1100,Ker = 10.0,QPA = 133,VPA = 77,QSP = 77,VSP = 192,CLint = 0.315,QHA =
    302,VKI = 280,R = 1)

y0 = (η = [0.0,0.0])

t = collect(range(0.0,stop=600.0,length=100))
sol_diffeq = solve(pbpkmodel,subject,param,y0,tspan=(0.0,600.0),saveat=t,progress=true)

using Plots
plot(sol_diffeq,vars=3)
```



3 Global Sensitivity Analysis

```
function sensivity_func(pars)
    y0 = (η = [0.0,0.0])
    sim = solve(pbpkmodel,subject,pars,y0,tspan=(0.0,600.0),saveat=t)
    f = t -> -sim(t;idxs=3)
    res = optimize(f,0.0,600.0,Brent())
    i,e = quadgk(f,0.0,600.0)
    [-Optim.minimum(res),-250/i]
end

a = []
for i in param
    if i != 0
        push!(a,[i-0.05*i,i+ 0.05*i])
    else
        push!(a,[0.0,1e-4])
    end
end

using Random
Random.seed!(5)
m = DiffEqSensitivity.morris_sensitivity(
                      sensivity_func,a,[10 for i in 1:70];
                      relative_scale= false,len_trajectory=75,
                      total_num_trajectory=50,num_trajectory=20)

q = keys(param)
sensitivities = NamedTuple()
for i in 1:length(m.means)
    global sensitivities = merge(sensitivities,[q[i] => m.means[i]])
end
```

```
sensitivity_var = NamedTuple()
for i in 1:length(m.means)
    global sensitivity_var = merge(sensitivity_var,[q[i] => m.variances[i]])
end

cvemax_sens = [[],[]]
cvemax_sens[1] = [log(i[1]) for i in m.means]
cvemax_sens[2] = [log(i[1]) for i in m.variances]
cl_sens = [[],[]]
cl_sens[1] = [log(i[2]) for i in m.means]
cl_sens[2] = [log(i[2]) for i in m.variances]

ann1 = []
for i in 1:length(cvemax_sens[2])
    if cvemax_sens[2][i] > -10
        push!(ann1,[cvemax_sens[1][i]-1.0,cvemax_sens[2][i]+1.5,string(q[i] == :ρ ?
    "rho" : q[i]),10])

    end
end
plot1 = scatter(cvemax_sens[1],cvemax_sens[2], annotations=ann1,legend=false,xlabel="Log
    mean of Morris Elementary Effects",ylabel="Log variance of Morris Elementary
    Effects",title="SA for Cmax")

ann2 = []
for i in 1:length(cl_sens[1])
    if cl_sens[2][i] > -10
        push!(ann2,[cl_sens[1][i],cl_sens[2][i]+1.5,string(q[i] == :ρ ? "rho" :
    q[i]),10])
    end
end
plot2 = scatter(cl_sens[1],cl_sens[2],annotations=ann2,legend=false,xlabel="Log mean of
    Morris Elementary Effects",ylabel="Log variance of Morris Elementary
    Effects",title="SA for AUC")

plot(plot1,plot2,figsize=(20,55))
```


SA for Cmax
SA for AUC
Log variance of Morris Elementary Effects
Log mean of Morris Elementary Effects
Log variance of Morris Elementary Effects
Log mean of Morris Elementary Effe
0
- 30
- 60
- 90
- 60
- 40
- 20
0
20
0
- 20
- 40
- 60
- 30
- 20
- 10
0
10
rho
L
d
GER
rho
L
d
GER
kilST
r
VST1
VSt2