PATROLS
Advanced Tools for NanoSafety Testing

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 760813

PATROLS Standard Operating Procedures (SOP)

PBPK model for nanomaterials

This is a SOP used by members of PATROLS only

Adapted from the NanoImpactNet SOP, Clift *et al* (Deliverable 5.4 under the European Commission's 7th Framework Programme, Grant Agreement 218539).

Authored by:
J. Minnema (RIVM)
J.E. Delmaar (RIVM)

Reviewed by:
I. Gosens (RIVM)

Document History:

Version	Approval Date	Description of the change	Author(s) of change
1.0	14/04/2021	Initial Document	

Table of Contents

1 Introduction:

DOMAIN: Other

In vitro to in vivo extrapolation (IVIVE) of toxicological dose-response data obtained from engineered nanomaterial (ENM) would advance the 3Rs principles (Reduction, Refinement and Replacement). One of the requirements for IVIVE to succeed is that the delivered dose to target organs needs to be accurately estimated. One way to estimate this delivered dose, is by using physiologically-based (pharmaco)kinetic (PBPK) models.

PBPK models aim to describe the distribution of particles or substances by describing the body as a set of compartments, representing biological tissues and organs. The kinetic processes that lead to exchange of particles between compartments are captured by rate equations. Using a parametrized PBPK model, distribution and accumulation of particles in organs and tissues over time can be estimated.

Recently, Li et al. (2015) developed a PBPK model to simulate the kinetic distribution of cerium dioxide nanoparticles. The model is extrapolated tot other nanomaterials by re-parametrizing the model. This SOP describes how to parametrize and use the PBPK model developed by Li et al. to simulate the kinetic distribution of nanomaterials in rats after inhalation exposure.

1.1 Scope and limits of the protocol

This research protocol elaborates on mathematical construction of the PBPK model, and describes the methodology behind the parametrization of the model. This SOP can be used to implement and parametrize the PBPK model proposed by Li et al. The SOP is intended to be used in the domain of inhalation exposure to nanomaterials. The parametrization of the PBPK model is necessary to apply model to a specific nanomaterial. Therefore, the parametrization described in this SOP requires (high quality) empirical distribution data of the nanomaterial under study. Without the emperical data, this SOP can not be fully performed. Finally, it must be noted that a PBPK model parametrized to predict the kinetic distribution of a particular nanomaterial should solely be used predict the distribution of that nanomaterial, as it possibly does not generalize well to other nanomaterials, without re-parametrizing the model.

1.2 Validation state of protocol

Level of advancement towards standardization	Level reached (please mark only one with "X")
Stage 1: Internal laboratory method under development	X
Stage 2: Validated internal laboratory method	
Stage 3: Interlaboratory tested method	
Stage 4: Method validated by Round Robin testing	
Standardisation plans	
Is the method considered for standardisation (OECD SPSF or similar)?	N
Has the method been submitted for standardisation (to OECD, CEN, ISO,…) in its own right or as part of another standardisation project?	N
Is the method included in an existing standard (or ongoing standardisation work)	N

2 Terms and Definitions:

Nanoscale
Length range approximately from 1 nm to 100 nm

Note 1 to entry: Properties that are not extrapolations from larger sizes are predominantly exhibited in this length range.

[SOURCE : ISO/TS 80004-1: 2016, definition 2.1]

Nanomaterial
Material with any external dimension in the *nanoscale* or having internal structure or surface structure in the nanoscale.

Note 1 to entry: This generic term is inclusive of *nano-object* and *nanostructured material*.

[SOURCE: ISO/TS 80004-1: 2016, definition 2.4]

Engineered nanomaterial
Nanomaterial designed for specific purpose or function

[SOURCE: ISO/TS 80004-1: 2016, definition 2.8]

Particle
Minute piece of matter with defined physical boundaries.

Note 1 to entry: A physical boundary can also be described as an interface.

Note 2 to entry: A particle can move as a unit.

Note 3 to entry: This general particle definition applies to *nano-objects*.

[SOURCE: ISO 26824:2013, 1.1]

Substance
Single chemical element or compound, or a complex structure of compounds.

[SOURCE: ISO 10993-9:2009, definition 3.6]

3 Abbreviations:

CPU	central processing unit
ENM	engineered nanomaterial
GI	gastro-intestinal
IVIVE	*in vitro* to *in vivo* extrapolation
MCMC	markov chain monte carlo
MPPD	multiple-path particle dosimetry
ODE	ordinary differential equation
PBPK	physiologically-based (pharmaco)kinetic
PC	phagocytizing cells

4 Principle of the Method:

PBPK models are used to quantitatively simulate the accumulation and distribution of nanoparticles over the body. PBPK modelsdescribe the different tissues and organs and compartments and the kinetic transport between these compartments by rate equations,by means of coupled ordinary differential equations (ODEs). Systems of coupled ODEs are typically solved numerically using programming software.

In order to accurately describe the kinetics of nanomaterials, the rates with which nanomaterials are transferred from one compartment to another need to be known. Estimating values of the model parameters that describe this transfer is essential for the use of a PBPK model. In this SOP (section 5.3), the estimation of PBPK parameter values using a Bayesian approach in combination with Markov Chain Monte Carlo (MCMC) sampling is described.

5 Description of the Method:

5.1 Apparatus and equipment used:

The PBPK model simulations were run on a computer with 8 different CPU cores (each containing 8GB RAM). The PBPK model was implemented in the R modelling language, using a dedicated PBPK development package mrgSolve. The following software packages were used to run the simulations:

Name software: R (v. 3.6.0)

Manufacturer: The R Foundation for Statistical Computing

Place of manufacture: online

Year of manufacture: 2019

Description: A programming language for statistical computing

Name software: mrgSolve

Manufacturer: *Metrum Research Group*

Place of manufacture: online

Year of manufacture: 2020

Description: An R package that allows solving the ODEs required in the PBPK models

Name software: adaptMCMC (v.1.3)

Manufacturer: Andreas Scheidegger

Place of manufacture: online

Year of manufacture: 2020

Description: Simulates continuous distributions of random vectors using Markov chain Monte Carlo (MCMC)

Name software: foreach (v.1.5.1)

Manufacturer: Michelle Wallig

Place of manufacture: online

Year of manufacture: 2020

Description: Necessary for parallel computing

Name software: doParallel (v.1.0.16)

Manufacturer: Michelle Wallig

Place of manufacture: online

Year of manufacture: 2020

Description: Necessary for parallel computing

Name software: Multiple-path particle dosimetry model (MPPD) (v.3.0.4)

Manufacturer: Applied Research Associates, Inc

Place of manufacture: online

Year of manufacture: 2020

Description The MPPD model calculates the deposition and clearance of monodisperse and polydisperse aerosols in the respiratory tracts of laboratory animals and human adults and children (deposition only) for particles ranging in size from ultrafine (1 nm) to coarse (100 μm)

5.2 Quality control & acceptance criteria:

The quality of the PBPK model can be validated by predicting the kinetic distribution of the nanomaterial under study on an independent test dataset. Such a test set is a set of measured nanomaterial concentrations in animal models acquired after an inhalation exposure scenario that is independent from the inhalation exposure used to parametrize the PBPK model. The predicted nanomaterial kinetics were subsequently compared to emprically measured nanomaterial concentrations of the test set. If the predictions correspond well with the measured concentrations, then it can be concluded that the PBPK model is likely to generalize well to other inhalation exposure scenario's.

5.3 Procedure:

<u>General model:</u>

The applied PBPK model consisted of 12 different compartments: arterial blood, venous blood, upper airways, tracheobronchial region, pulmonary region, liver, spleen, kidney, heart, brain, gastro-intestinal (GI) tract, and the remaining organs (rest). All 12 compartments contained a subcompartment representing phagocytizing cells (PCs). Moreover, the organ compartments (i.e., all compartments except arterial blood and venous blood) contained additional subcompartments representing the capillary blood and the tissue. All compartments are interconnected through the systemic blood circulation. Finally, clearance of ENMs can occur through the lymph, urine and the feces.

The PBPK model was applied to simulate the distribution of ENMs after inhalation exposure in rats. After inhalation exposure, a fraction of the inhaled ENMs are

deposited in each part of the respiratory system (i.e., upper airways, tracheobronchial region and pulmonary region). The fractions for each respective region can be estimated in two ways: 1) deducing the fractions from the measured concentrations, or 2) estimating the fractions using the multiple-path particle dosimetry (MPPD) model.

Depending on the region where the ENMs were deposited, ENMs can migrate to different parts of the body. ENMs deposited in the upper aiways can either be transferred to the brain via the olfactory bulb, or to the GI tract by being swallowed. From the tracheobronchial region, ENMs can migrate to the pharynx in the upper airway and be subsequently swallowed to the GI-tract. ENMs that penetrate the interstitium of the lungs can enter the systemic blood circulation and be subsequently distributed over the different organ compartments, or they can migrate to the lymphs.

The PBPK model was implemented in the R programming language, using the software packages described in section 5.1. The PBPK source code is described in supplemental material 1.

Model equations

The PBPK model described in this SOP consists of a set of coupled ordinary differential equations (ODEs). These ODEs describe the movement of a particular substance between compartments over time. For each (sub)compartment of the model, a ODE was formulated. The ODEs used in the PBPK model are presented below.

Pulmonary region

Eq. 1)

$$\frac{dA_{pulm}}{dt} = -k_{lupi} \times A_{pulm} - k_{pulmPCAbs} \times A_{pulm} + k_{luip} \times A_{lungIS} + k_{de} \times A_{pulmPC}$$

A_{pulm} [µg] : Amount of nanoparticles in the pulmonary region

A_{lungIS} [µg] : Amount of nanoparticles in the lung interstitium

A_{pulmPC} [µg] : Amount of nanoparticles in the pulmonary phagocytizing cells (PCs)

k_{lupi} [h^{-1}] : Transfer rate of nanoparticles from pulmonary region to the interstitium of the lungs

$k_{pulmPCabs}$ [h^{-1}] : Uptake rate of nanoparticles by PCs in the pulmonary region

k_{luip} [h^{-1}] : Transfer rate of nanoparticles from lung interstitium to the pulmonary region

k_{de} [h^{-1}]: Rate of release of nanomaterial from PCs

Tracheobronchial region

Eq. 2)

$$\frac{dA_{tb}}{dt} = -k_{tbgi} \times A_{tb} + k_{pulmtb} \times A_{pulmPC}$$

A_{tb} [µg] : Amount of nanoparticles in the tracheobronchial region

k_{tbgi} [h^{-1}] : Transfer rate of nanoparticles from tracheobronchial region to the GI-tract

k_{pulmtb} [h^{-1}] : Transfer rate of nanoparticles from pulmonary region to the tracheobronchial region

Upper airway region

Eq. 3)

$$\frac{dA_{ua}}{dt} = -A_{ua} \times \left(k_{uagi} + k_{olfEp}\right)$$

A_{ua} [µg] : Amount of nanoparticles in the upper airway region

k_{uagi} [h^{-1}] : Transfer rate of nanoparticles from upper airway region to the GI-tract

k_{olfEp} [h^{-1}] : Transfer rate of nanoparticles from the upper aiway region to the olfactory epithelium

GI-tract

Eq. 4)

$$\frac{dA_{gilumen}}{dt} = A_{ua} \times k_{uagi} + A_{tb} \times k_{tbgi} + A_{liver} \times k_{lgi} - A_{gilumen} \times (k_{lumfeces} + k_{giab})$$

$A_{gilumen}$ [µg] : Amount of nanoparticles in the lumen of the GI-tract

A_{liver} [µg] : Amount of nanoparticles in the liver

k_{lgi} [h^{-1}] : Transfer rate of nanoparticles from the liver to the GI-tract

$k_{lumfeces}$ [h^{-1}] : Transfer (clearance) rate of nanoparticles from the GI-tract lumen to the feces

k_{giab} [h^{-1}] : Transfer rate of nanoparticles from the GI-tract lumen to the blood in the GI-tract

Organ tissues – spleen, GI-tract

Eq. 5)

$$\frac{dA_{tissue}}{dt} = Q_{tissue} \times \chi_{tissue} \times (C_{tissueBlood} - C_{tissue}/P) - (A_{tissue} \times k_{tissuePC} - A_{tissuePC} \times k_{de})$$

A_{tissue} [µg] : Amount of nanoparticles in the organ tissue of interest

$A_{tissuePC}$ [µg] : Amount of nanoparticles in the PC of the organ tissue of interest

Q_{tissue} [L/h] : Blood flow through the organ tissue of interest

χ_{tissue} [-] : permeability coefficient

$C_{tissueBlood}$ [µg/g] : Concentration of nanoparticles in the capillary blood of the organ tissue of interest

C_{tissue} [µg/g] : Concentration of nanoparticles in the organ tissue of interest

P [-] : Partition coefficient of nanoparticles between the organ tissue of interest and the blood

$k_{tissuePC}$ [h^{-1}] : Transfer rate of nanoparticles from the organ tissue of interest to PCs in that particular organ tissue. $k_{lumfeces}$ [h^{-1}] : Transfer (clearance) rate of nanoparticles from the GI-tract lumen to the feces

Organ tissues – heart, rest

Eq. 6)

$$\frac{dA_{tissue}}{dt} = Q_{tissue} \times \frac{\chi_{tissue}}{1+\chi_{tissue}} \times (C_{tissueBlood} - C_{tissue}/P) - (A_{tissue} \times k_{tissuePC} - A_{tissuePC} \times k_{de})$$

Organ tissues – kidney

Eq. 7)

$$\frac{dA_{kidney}}{dt} = Q_{tissue} \times \frac{\chi_{kidney}}{1+\chi_{tissue}} \times (C_{kidneyBlood} - C_{kidney}/P) - (A_{kidney} \times k_{kidneyPC} - A_{kidneyPC} \times k_{de}) - C_{art} \times k_{el} \times \frac{\chi_{kidney}}{1+\chi_{kidney}}$$

A_{kidney} [µg] : Amount of nanoparticles in the kidney tissue

$A_{kidneyPC}$ [µg] : Amount of nanoparticles in the PC of the kidney

Q_{kidney} [L/h] : Blood flow through the kidney tissue

χ_{kidney} [-] : permeability coefficient of the liver tissue

$C_{kidneyBlood}$ [µg/g] : Concentration of nanoparticles in the capillary blood of the kidney

C_{kidney} [µg/g] : Concentration of nanoparticles in the kidney tissue

C_{art} [µg/g] : Concentration of nanoparticles in the arterial blood

$k_{kidneyPC}$ [h^{-1}] : Transfer rate of nanoparticles from the kidney tissue to PCs in the kidney

k_{el} [h^{-1}] : Elimination rate from kidney to urine

Organ tissues – liver

Eq. 8)

$$\frac{dA_{liver}}{dt} = \left(Q_{liver} + Q_{spleen} + Q_{gitract}\right) \times \chi_{liver} \times (C_{liverBlood} - C_{liver}/P) - (A_{liver} \times k_{liverPC} - A_{liverPC} \times k_{de}) - A_{liver} \times k_{lgi}$$

A_{liver} [µg] : Amount of nanoparticles in the liver tissue

$A_{liverPC}$ [µg] : Amount of nanoparticles in the PC of the liver

Q_{liver} [L/h] : Blood flow through the liver tissue

Q_{spleen} [L/h] : Blood flow through the spleen tissue

$Q_{gitract}$ [L/h] : Blood flow through the GI-tract tissue

χ_{liver} [-] : permeability coefficient of the liver tissue

$C_{liverBlood}$ [µg/g] : Concentration of nanoparticles in the capillary blood of the liver

C_{liver} [µg/g] : Concentration of nanoparticles in the liver tissue

$k_{liverPC}$ [h^{-1}] : Transfer rate of nanoparticles from the liver tissue to the PCs in the liver

Organ tissues – brain

Eq. 9)

$$\frac{dA_{brain}}{dt} = A_{ua} \times k_{olfEp} + Q_{brain} \times \frac{\chi_{brain}}{1 + \chi_{brain}} \times (C_{brainBlood} - C_{brain}/P) - (A_{brain} \times k_{brainPC} - A_{brainPC} \times k_{de})$$

A_{brain} [µg] : Amount of nanoparticles in the brain tissue

$A_{brainPC}$ [µg] : Amount of nanoparticles in the PC of the brain

Q_{brain} [L/h] : Blood flow through the brain

χ_{brain} [-] : permeability coefficient of the brain

$C_{brainBlood}$ [µg/g] : Concentration of nanoparticles in the capillary blood of the brain

C_{brain} [µg/g] : Concentration of nanoparticles in the brain tissue

$k_{brainPC}$ [h^{-1}] : Transfer rate of nanoparticles from the brain tissue to the PCs in the brain

Capillary blood in organ tissues

Eq. 10)

$$\frac{dA_{tissueBlood}}{dt} = Q_{tissue} \times (C_{art} - C_{tissueBlood}) - Q_{tissue} \times \chi_{tissue} \times (C_{tissueBlood} - C_{tissue}/P)$$

$A_{tissueBlood}$ [µg] : Amount of nanoparticles in the capillary blood of the organ tissue of interest

Arterial and venous blood

Eq. 11)

$$\frac{dA_{art}}{dt} = C_{art} \times Q_{tot} + Q_{tot} \times \frac{(C_{ven} + \chi_{lungIS} \times C_{lungIS}/P)}{1 + \chi_{lungIS}} - (A_{art} \times k_{bloodab} - 0.2 \times A_{bloodPC} \times k_{de}$$

Eq. 12)

$$\frac{dA_{ven}}{dt} = -(Q_{liver} + Q_{spleen} + Q_{gitract}) \times C_{liverBlood} + Q_{rest} \times \frac{\left(C_{art} + \chi_{rest} \times \frac{C_{rest}}{P}\right)}{1+\chi_{rest}} + Q_{brain} \times \frac{\left(C_{art} + \chi_{brain} \times \frac{C_{brain}}{P}\right)}{1+\chi_{brain}} + Q_{heart} \times \frac{\left(C_{art} + \chi_{heart} \times \frac{C_{heart}}{P}\right)}{1+\chi_{heart}} + Q_{kidney} \times \frac{\left(C_{art} \times \left(\frac{1-k_{el}}{Q_{kidney}}\right) + \chi_{kidney} \times \frac{C_{rest}}{P}\right)}{1+\chi_{kidney}} - Q_{tot} \times C_{ven} - (k_{bloodab} \times A_{ven} - 0.8 \times A_{bloodPC} \times k_{de})$$

A_{art} [µg] : Amount of nanoparticles in the arterial blood

A_{ven} [µg] : Amount of nanoparticles in the venous blood

$A_{bloodPC}$ [µg]: Amount of nanoparticles in the PCs of the blood

Q_{tot} [L/h] : Total blood flow

Q_{rest} [L/h]: Blood flow through the rest tissue compartment

Q_{heart} [L/h]: Blood flow through the heart tissue

χ_{rest} [-]: Permeability coefficient of the rest tissue compartment

χ_{heart} [-]: Permeability coefficient of the heart tissue

$k_{bloodab}$ [h^{-1}]: Uptake rate of nanoparticles by PCs in the blood

Model parametrization

Parametrization of the PBPK model was performed using a Bayesian parameter estimation method that estimates parameter values from inhalation/distribution studies. Bayesian methods use prior information on parameter values and optimize those prior estimates using oberserved data. This essentially allows one to refine and optimize the parameters in a model when additional experimental datasets are acquired.

Bayesian parameter estimation methods are based on Bayes' theorem, which can be expressed as

Eq. 13)

$$P(\vartheta|D) = \frac{P(\vartheta) \cdot P(D|\vartheta)}{\int P(\vartheta') \cdot P(D|\vartheta')\, d\vartheta'}$$

, where ϑ denotes a set of parameters (e.g., those in a PBPK model) and D denotes a dataset (e.g., measured CeO_2 distribution). $P(\vartheta|D)$ is the posterior likelihood, $P(\vartheta)$ is the prior distribution and $P(D|\vartheta)$ is the data likelihood. $\int P(\vartheta')P(D|\vartheta')\, d\vartheta'$ is a normalization term. This integral required for the normalization is usually very complicated to solve for models with multiple model parameters. However, methods exist that allow one to sample from the resulting posterior parameter distribution without explicitly solving Eq.13. Basically, these methods sample from the un-normalized distribution (eq 14). The resulting sample follows the same distribution as eq. 13.

Eq. 14)

$$P(\vartheta|D) \propto P(\vartheta) \cdot P(D|\vartheta)$$

Eq 14. shows that the posterior likelihood distribution can be found using prior distribution of the parameter values and the data likelihood. The prior distribution expresses knowledge on the parameter value before evidence (e.g., the observed ENM concentrations). In case little prior information is available, the use of so-called uninformed priors is adviced. A particular uninformed prior, 'Jeffreys prior distribution' was used in this SOP:

Eq. 15)

$$P(\vartheta) \propto \frac{1}{\vartheta}$$

Besides the prior distribution, also the data likelihood needs to be calculated to compute the the posterior distribution ($P(\vartheta|D)$. The data likelihood is a measure for the difference between the output predicted by a model parametrized with the set of parameters ϑ, and the actual observed values. The data likelihood was calculated as follows:

Eq. 16)

$$P(D|\vartheta) = e^{-\frac{1}{N}\sum_{i=1}^{N}(\log(predicted_i)-\log(observed_i))^2}$$

where *i* indicates a single measurement of an amount or concentration in a compartment of the PBPK-model, and *N* is the total number of measurements available.

Markov Chain Monte Carlo

The posterior parameter distributions were sampled using the Markov Chain Monte Carlo (MCMC) method. MCMC provides a way to efficiently sample from high-dimensional probability distributions. Basically, a MCMC method estimates a data likelihood distribution $P(D|\vartheta)$ by sequentially calculating data likelihoods based on randomly sampled (i.e., Monte Carlo) sets of parameters. If a calculated data likelihood exceeds the previously calculated likelihood, then the set of parameters is added to the chain. If the data likelihood does not exceed the previously calculated likelihood, then the parameter sample is rejected with a probability that increases as the data likelihood decreases. Thus, the idea behind this methodology is that the Markov Chain will settle on the highest possible data likelihood. Additional information on the MCMC method can be found in Ravenzwaaij (2016).

Choice of parameters to estimate

It is computationally infeasible to find reliable data likelihood distributions of all parameters of the PBPK model. Therefore, it is recommended to select a subset of parameters (5-10 different parameters) to reduce the computational cost. In this context, parameter selection is based on a formal sensitivity analysis, indicating the parameters that most strongly influenced the model predictions.

The values of the remaining parameters in the model were directly adopted from the parametrized model of Li et al (2015).

6 Data Analysis and Reporting of Data:

After estimating the optimal values of the parameters (i.e., parametrizing) in the PBPK model, the quality of the model was validated on an independent test set. In this context, the PBPK model is used to predicted nanomaterial distributions in an inhalation scenario that is different from that used to parameterize the model. The model predictions are subsequently compared to the empiric measurements of the nanomaterial concentrations. To allow for qualitative visual comparison, the predicted nanomaterial concentrations were plotted over time and compared to nanomaterial concentrations empirically measured in inhalation studies.

Concerning the parametrization of the model, credible regions were reported for the seven parameters that were optimized. Besides the parameter value with highest posterior likelihood, these credible regions also represent the likely range of possible parameter values, and are thus a indication for the uncertainty in the parameter estimates.

7 References

Li D, et al. 2015 In vivo biodistribution and physiologically based pharmacokinetic modeling of inhaled fresh and aged cerium oxide nanoparticles in rats. Particle and Fibre Toxicology 13(45)

Ravenzwaaij D, et al. (2018) A simple introduction to Markov Chain Monte–Carlo sampling. Psychonomic Bulletin & Review 25.p 143-154

8 Supplemental material 1: Model source code

```
codeGeneralModel<- '

$PARAM
includeLiver  = 1,
includeSpleen = 1,
includeGITract= 1,
includeBrain  = 1,
includeHeart  = 1,
includeLung   = 1,
includeKidney = 1,
frCapBlBrain  = 0.371,
frCapBlOther  = 0.144,
BW  = 216.3,
Rho = 1.0,
Xrich     = 0.776,
Xrest     = 0.0171,
Xbr       = 6.75e-7,
P         = 0.209,
fQl      = 0.0456,
fQgi     = 0.213,
fQs      = 0.0146,
fQlung   = 0.0456,
fQbrain  = 0.02,
fQheart  = 0.051,
fQkidney = 0.141,
kab0        = 1.45,
ksab0       = 0.518,
kde         = 5.30e-19,
PCCap       = 5.52e-7,
NBloodCap   = 0.185e4,
NLcap       = 2.72e7,
NSpleenCap  = 2.08e8,
NPulmCap    = 3.90e6,
NLungISCap  = 2.69e6,
NGICap      = 0.506e6,
NBrainCap   = 0.306e6,
NHeartCap   = 0.076e6,
NKidneyCap  = 0.99e5,
NRestCap    = 8.11e6,
kuabr     = 8.36e-5,
kuagi     = 0.335,
kpulmtb   = 8.65e-4,
ktbgi     = 5.52e-3,
kluip     = 1.12e-6,
klupi     = 0.126,
klgi      = 1.52e-3,
klumfeces = 0.141,
kkidneyEl = 3.26,
kgiab     = 5.41e-3,
klisly    = 0.000175
```

```
$GLOBAL
#define cVBlood (aVBlood/VVBlood)
#define cABlood (aABlood/VABlood)
#define cLiver (aLiver/VLiver)
#define cSpleen (aSpleen/VSpleen)
#define cSBlood (aSBlood/VSBlood)
#define cGITissue (aGITissue/VGITract)
#define cGIBlood (aGIBlood/VGIBlood)
#define cBrain (aBrain/VBrain)
#define cHeart (aHeart/VHeart)
#define cKidney (aKidney/VKidney)

#define cLungIS (aLungIS/VLung)
#define cPulmPCs (aPulmPCs/VLung)
#define cLBlood (aLBlood/VLBlood)
#define cRest (aRest/VRest)

#define cBloodTotal (aBloodTotal/VBloodTot)
#define cLiverTotal (aLiverTotal/VLiver)
#define cSpleenTotal (aSpleenTotal/VSpleen)
#define cHeartTotal (aHeartTotal/VHeart)
#define cKidneyTotal (aKidneyTotal/VKidney)
#define cBrainTotal (aBrainTotal/VBrain)
#define cPulmTotal (aPulmTotal/VLung)
#define cLungISTotal (aLungISTotal/VLung)
#define cLungTotal (aLungTotal/VLung)

#define cLiverTissueCumulative ((aLiverTissueCumulative)/VLiver)
#define cPulmTotalCumulative (aPulmTotalCumulative/VLung)
#define cLungISTotalCumulative (aLungISTotalCumulative/VLung)

#define totalMass
(aABlood+aBloodPCs+aLBlood+aLiver+aLiverPCs+aVBlood+aSpleen+aSBlood+aSpleen
PCs+aGITissue+aGIBlood+aGIPCs+aGILumen+aUA+aTB+aPulmPCs+aPulm+aLungIS+aLung
ISPCs+aBrain+aBrainPCs+aHeart+aHeartPCs+aKidney+aKidneyPCs+
aRest+aRestPCs+aFeces+aUrine)

#define aLiverTotal  (aLiver + aLiverPCs + frCapBlOther * cBloodTotal *
VLiver)

#define aLungISTotal (aLungIS + aLungISPCs)
#define aPulmTotal   (aPulm + aPulmPCs)
#define aLungTotal   (aTB + aLungISTotal + aPulmTotal)

#define aBloodTotal  (aVBlood + aABlood + aBloodPCs  + aLBlood + aSBlood +
aGIBlood)
#define aSpleenTotal (aSpleen + aSpleenPCs + frCapBlOther * cBloodTotal *
VSpleen)
#define aHeartTotal  (aHeart + aHeartPCs + frCapBlOther * cBloodTotal *
VHeart)
#define aKidneyTotal (aKidney + aKidneyPCs + frCapBlOther * cBloodTotal *
VKidney)
#define aBrainTotal  (aBrain + aBrainPCs + frCapBlOther * cBloodTotal *
VBrain)
#define aRestTotal   (aRest + aRestPCs + frCapBlOther * cBloodTotal *
VRest)
#define aLiverTissueCumulative (aLiverCumulative + aLiverPCsCumulative)
#define aPulmTotalCumulative   (aPulmCumulative + aPulmPCsCumulative)
#define aLungISTotalCumulative (aLungISCumulative + aLungISPCsCumulative)
```

```
#define Vtot       (BW/Rho)
#define VLiver     (0.034 * Vtot)
#define VSpleen    (0.002 * Vtot)
#define VSBlood    (0.22 * VSpleen)
#define VGITract   (0.050 * Vtot)
#define VGIBlood   (0.1 * VGITract)
#define VPulm      (0.005 * Vtot)
#define VBrain     (0.006 * Vtot)
#define VHeart     (0.003 * Vtot)
#define VKidney    (0.007 * Vtot)

#define VLBlood    (0.21 * VLiver)
#define VLung      (0.005 * Vtot)
#define VLungBlood (0.36 * VLung)
#define VBloodTot  (0.074 * Vtot)
#define VBlood     (VBloodTot - VLBlood - VSBlood - VGIBlood)
#define VVBlood    (0.8 * VBlood)
#define VABlood    (0.2 * VBlood)
#define VBBlood    (0.03 * VBrain)
#define VHBlood    (0.26 * VHeart)
#define VKBlood    (0.16 * VKidney)

#define VRest (Vtot - VLiver*includeLiver - VLung*includeLung -
VSpleen*includeSpleen - VGITract*includeGITract-VBrain * includeBrain -
VHeart * includeHeart- VKidney * includeKidney- VBloodTot)
#define VRestBlood (0.017 * VRest)
#define fQrest (1 - fQl*includeLiver- fQgi*includeGITract-
fQs*includeSpleen- fQbrain*includeBrain - fQheart*includeHeart-
fQkidney*includeKidney)


#define Qtot     ((1000 * 60) * 0.235 * pow(BW/1000, 0.75))
#define QLiver   (fQl * Qtot * includeLiver)
#define QSpleen  (fQs * Qtot * includeSpleen)
#define QGIT     (fQgi * Qtot * includeGITract)
#define QBrain   (fQbrain * Qtot * includeBrain)
#define QLung    (fQlung * Qtot)
#define QHeart   (fQheart * Qtot * includeHeart)
#define QKidney  (fQkidney * Qtot * includeKidney)
#define QRest    (fQrest * Qtot)

#define BloodPCCap (PCCap * NBloodCap)
#define kbloodab (std::max(0.0, kab0 * (1-aBloodPCs/(BloodPCCap * VBlood *
Rho))))

#define LiverPCCap (PCCap * NLcap)
#define kLiverPCAbs (std::max(0.0, kab0 *(1-aLiverPCs/(LiverPCCap * VLiver
* Rho))))

#define SpleenPCCap (PCCap * NSpleenCap)
#define kSpleenPCAbs (std::max(0.0, ksab0 *(1-aSpleenPCs/(SpleenPCCap *
VSpleen * Rho))))

#define PulmPCCap (PCCap * NPulmCap)
#define kPulmPCAbs (std::max(0.0, kab0 *(1-aPulmPCs/(PulmPCCap * VLung *
Rho))))

#define LungISPCCap  (PCCap * NLungISCap)
```

```
#define kLungISPCAbs (std::max(0.0, kab0 *(1-aLungISPCs/(LungISPCCap *
VLung * Rho))))

#define GIPCCap (PCCap * NGICap)
#define kGIPCAbs (std::max(0.0, kab0 * (1-aGIPCs/(GIPCCap * VGITract *
Rho))))

#define BrainPCCap (PCCap * NBrainCap)
#define kbrab (std::max(0.0, kab0 * (1-aBrainPCs/(BrainPCCap * VBrain *
Rho))))

#define HeartPCCap (PCCap * NHeartCap)
#define khab (std::max(0.0, kab0 * (1-aHeartPCs/(HeartPCCap * VHeart *
Rho))))

#define KidneyPCCap (PCCap * NKidneyCap)
#define kkab (std::max(0.0, kab0 * (1-aKidneyPCs/(KidneyPCCap * VKidney *
Rho))))

#define RestPCCap (PCCap * NRestCap)
#define kRestPCAbs (std::max(0.0, kab0 * (1-aRestPCs/(RestPCCap * VRest *
Rho))))

#define kAbsOlfactoryEpithelium (kuabr * includeBrain)


$CMT
aABlood aBloodPCs aLBlood aLiver aLiverPCs aVBlood aSpleen aSBlood
aSpleenPCs aGITissue
aGIBlood aGIPCs aGILumen aUA aTB aPulmPCs aPulm aLungIS aLungISPCs
aBrain aBrainPCs aHeart aHeartPCs aKidney aKidneyPCs aRest aRestPCs aFeces
aUrine aLymph
aLiverCumulative aLiverPCsCumulative aPulmCumulative aPulmPCsCumulative
aLungISCumulative aLungISPCsCumulative

$ODE
dxdt_aABlood = -cABlood * Qtot + Qtot * (cVBlood + Xrest *
cLungIS/P)/(1+Xrest)
               -(kbloodab * aABlood  - 0.2 * aBloodPCs * kde);
dxdt_aBloodPCs = (0.2 * aABlood + 0.8 * aVBlood) * kbloodab - aBloodPCs *
kde;

dxdt_aLBlood    =  -(QLiver + QSpleen + QGIT) * cLBlood
                    -(QLiver + QSpleen + QGIT) * Xrich * (cLBlood -
cLiver/P)
                    + QLiver * cABlood + QSpleen * cSBlood + QGIT *
cGIBlood;
dxdt_aLiver     = (QLiver + QSpleen + QGIT) * Xrich * (cLBlood - cLiver/P)
                  -(aLiver * kLiverPCAbs - aLiverPCs * kde) - klgi *
aLiver;
dxdt_aLiverPCs  = aLiver * kLiverPCAbs - aLiverPCs * kde;


dxdt_aVBlood =  (QLiver + QSpleen + QGIT) * cLBlood
              + QRest *(cABlood + Xrest*cRest/P)/(1+Xrest)
              + QBrain/(1 + Xbr) * (cABlood + Xbr * cBrain/P)
              + QHeart/(1+Xrest) * (cABlood + Xrest * cHeart/P)
              + QKidney/(1+Xrest) * (cABlood * (1-kkidneyEl/QKidney) +
Xrest * cKidney/P)
```

```
                - Qtot * cVBlood
                -(kbloodab * aVBlood - 0.8 * aBloodPCs * kde);

dxdt_aSpleen    = QSpleen * Xrich * (cSBlood - cSpleen/P) - (aSpleen *
kSpleenPCAbs - aSpleenPCs * kde);
dxdt_aSBlood    = QSpleen * (cABlood - cSBlood) - QSpleen * Xrich *
(cSBlood - cSpleen/P);
dxdt_aSpleenPCs = aSpleen * kSpleenPCAbs - aSpleenPCs * kde;

dxdt_aGITissue = QGIT * Xrest * (cGIBlood - cGITissue/P) - (aGITissue *
kGIPCAbs - aGIPCs * kde);
dxdt_aGIBlood  = QGIT * (cABlood - cGIBlood) - QGIT * Xrest * (cGIBlood -
cGITissue/P) + kgiab * aGILumen;
dxdt_aGIPCs    = aGITissue * kGIPCAbs - aGIPCs * kde;
dxdt_aGILumen  = kuagi * aUA + ktbgi * aTB - klumfeces * aGILumen - kgiab *
aGILumen + klgi * aLiver;

dxdt_aUA      = - kuagi * aUA - aUA * kAbsOlfactoryEpithelium;
dxdt_aTB      = - ktbgi * aTB  + kpulmtb * aPulmPCs;
dxdt_aPulm    = - klupi * aPulm + kluip * aLungIS - (aPulm * kPulmPCAbs -
aPulmPCs * kde);
dxdt_aPulmPCs = (aPulm * kPulmPCAbs - aPulmPCs * kde)  - kpulmtb *
aPulmPCs;

dxdt_aLungIS  = Xrest/(1+Xrest) * Qtot * (cVBlood - cLungIS/P)
                -(aLungIS * kLungISPCAbs - aLungISPCs * kde)
                + klupi * aPulm - kluip * aLungIS;

dxdt_aLungISPCs = aLungIS * kLungISPCAbs - aLungISPCs * kde - klisly *
aLungISPCs;


dxdt_aBrain     = aUA * kAbsOlfactoryEpithelium + Xbr/(1 + Xbr) * QBrain *
(cABlood - cBrain/P) - (aBrain * kbrab - aBrainPCs * kde);
dxdt_aBrainPCs  = (aBrain * kbrab - aBrainPCs * kde);

dxdt_aHeart     = Xrest/(1+Xrest) * QHeart * (cABlood - cHeart/P) - (aHeart
* khab - aHeartPCs * kde);
dxdt_aHeartPCs  = (aHeart * khab - aHeartPCs * kde);

dxdt_aKidney    = Xrest/(1+Xrest) * QKidney * (cABlood - cKidney/P) -
(aKidney * kkab - aKidneyPCs * kde)
                  - cABlood * kkidneyEl * Xrest/(1+Xrest);
dxdt_aKidneyPCs = (aKidney * kkab - aKidneyPCs * kde);

dxdt_aRest   = Xrest/(1+Xrest) * QRest * (cABlood - cRest/P)
              -(aRest * kRestPCAbs - aRestPCs * kde);
dxdt_aRestPCs= aRest * kRestPCAbs - aRestPCs * kde;

dxdt_aFeces = klumfeces * aGILumen;
dxdt_aUrine = cABlood * kkidneyEl * Xrest/(1+Xrest);

dxdt_aLymph = klisly * aLungISPCs;

dxdt_aLiverCumulative     = aLiver;
dxdt_aLiverPCsCumulative  = aLiverPCs;
dxdt_aPulmCumulative      = aPulm;
dxdt_aPulmPCsCumulative   = aPulmPCs;
dxdt_aLungISCumulative    = aLungIS;
```

```
dxdt_aLungISPCsCumulative = aLungISPCs;


$CAPTURE
aLungTotal aBloodTotal aLiverTotal aSpleenTotal aBrainTotal aKidneyTotal
aHeartTotal aLungISTotal
aBloodTotal aRestTotal aPulmTotal totalMass
cABlood cVBlood cSpleen cLiver cGITissue cBrain cHeart cKidney cLungISTotal
cPulmTotal cRest
cSpleenTotal cLiverTotal cBloodTotal cHeartTotal cKidneyTotal cBrainTotal
cLungTotal
cPulmTotalCumulative cLiverTissueCumulative cLungISTotalCumulative
aPulmTotalCumulative
aLiverTissueCumulative aLungISTotalCumulative aLungISTotal
'
```