

Session 1:

Overview of Pharmacokinetic Models and Computational Toolkits

Nan-Hung Hsieh, PhD

Postdoc @ Texas A&M Superfund Decision Science Core

12/09/2019



About Me



MS in Safety, Health and Environmental Engineering @ National United University

PhD in Bioenvironmental Systems Engineering @ National Taiwan University

- Research Associate @ Institute of Labor, Occupational Safety And Health, Ministry of Labor
- Postdoctoral Research Associate @ Texas A&M University
- Associate Toxicologist @ California Environmental Protection Agency

My Research: Computational Toxicology & Risk Assessment

Interest: Software Development



Content



1 Basic Pharmacokinetics Concepts

2 Pharmacokinetic Models

3 Computational Toolkits

4 Hands-on Exercise

Basic Pharmacokinetics Concepts

Why We Need to Do **Computational Modeling** in Toxicology?

"Prediction"

Pharmacokinetics (PK) / Toxicokinetics (TK)



- The study of the movement of chemicals in and out of the body (“what the body does to the chemical”)
- **ADME** process

Asorption - How will it get in?

Distribution - Which tissue organ it will go?

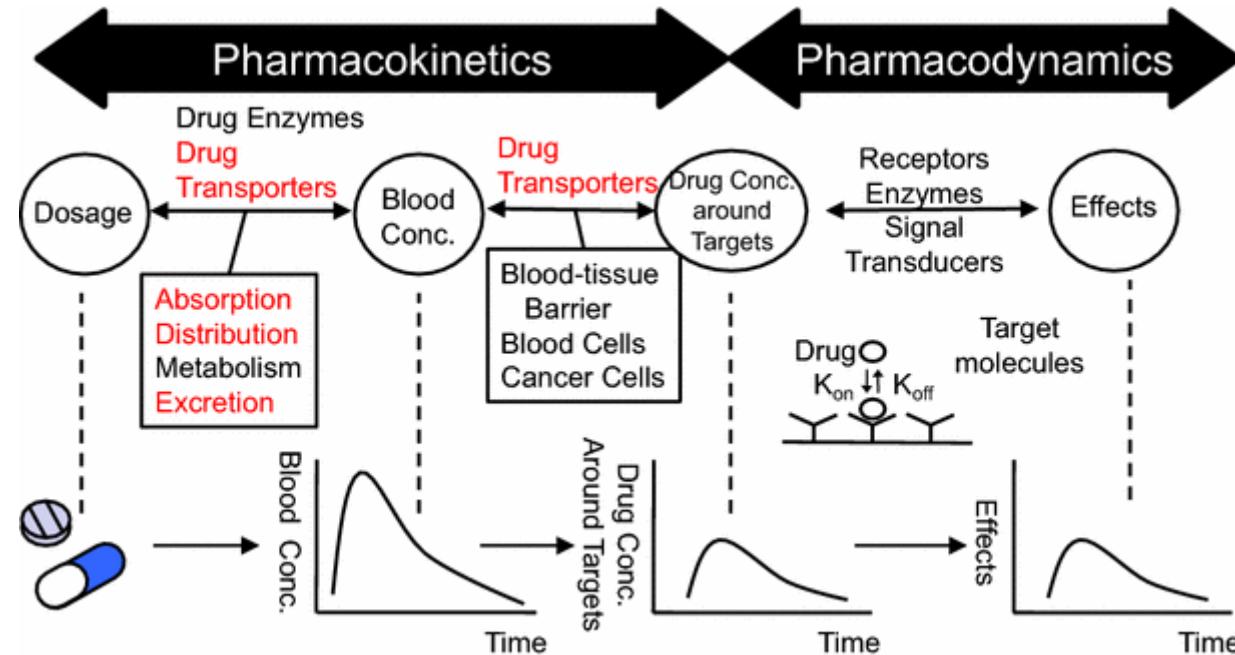
Metabolism - How is it broken down and transformation?

Elimination - How it leave the body?

- Kinetics: rates of change
- PK is focus on **TIME (t)** and **CONCENTRATION (C)**

Pharmacokinetics / Pharmacodynamics

Pharmacokinetics is the study of the fate of chemical in a living organism through **ADME** process (**absorption, distribution, metabolism, and elimination**).



"PK" focus on **DOSE** to **CONCENTRATION** / "PD" focus on **Concentration** to **Response**

Absorption

Exposure routes

- Inhalation – alveolar gas exchange
- Oral – gut absorption
- Dermal –permeation through skin

Movement of chemical into the body

- **Rate of absorption** (how quickly does it absorb?)
- **Fraction of absorption** (how percentage does it absorb?)

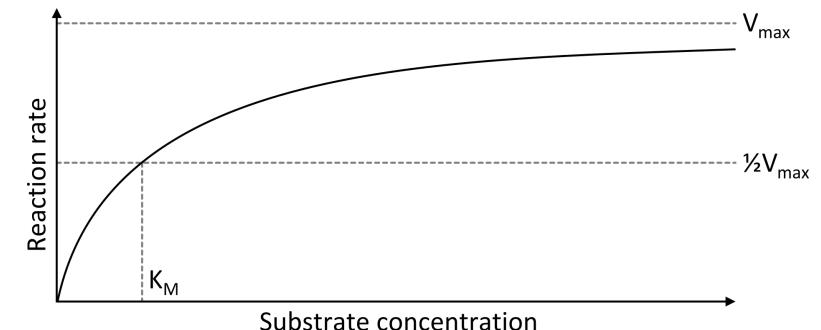
Bioavailability ia a key factor for substance becomes available to the target tissue after administration.

Distribution

- Transfer of the chemical between general circulation and tissues
- Important toxicologically because tissue concentration drives toxicity
- Factors affecting tissue distribution
 - **Blood flow** to tissues
- Tissue distribution (For chemicals that rapidly diffuse through membranes)
 - Assume rapid equilibrium between blood and tissue
 - Ratio is the tissue:blood **partition coefficient**
(Partition coefficient = Concentration in tissue / Concentration in blood)
 - Delivery to tissue limited by blood flow (“flow-limited”).

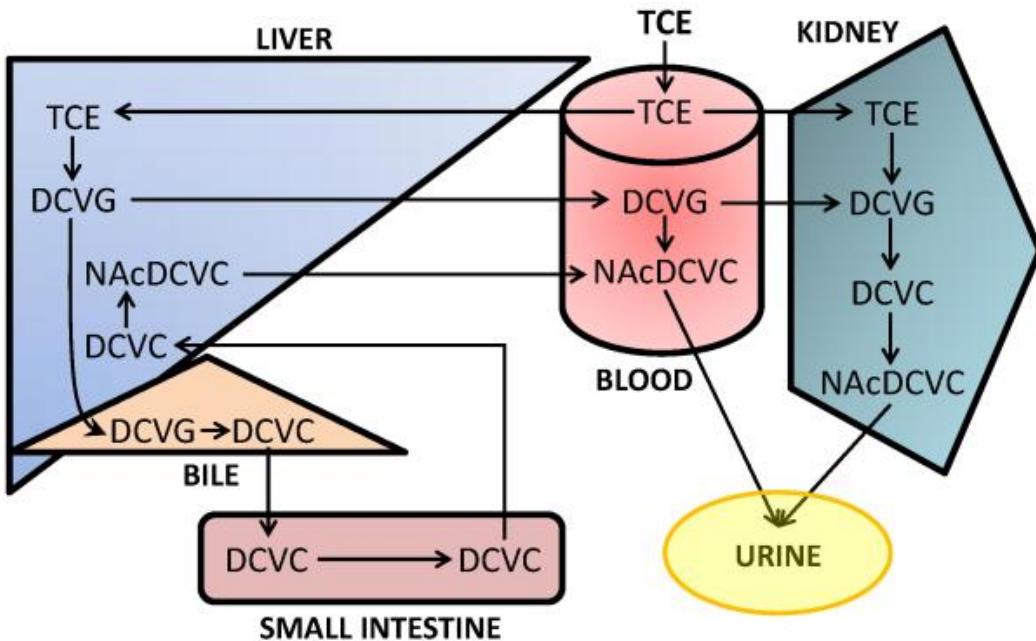
Metabolism/transformation

- Enzyme systems, etc. covered in separate class.
 - Phase I – conversion to more polar metabolite, e.g., oxidation by P450
 - Phase II – conjugation with an endogenous substrate to increase solubility, e.g., GSH conjugation, glucuronidation
- Key issues relevant to risk assessment
 - What are the metabolites?
 - In what tissues does metabolism occur?
 - What are the rates of metabolism?
 - What is known about interspecies and intraspecies differences?
- Metabolism process
 - First order kinetics (linear; constant)
 - Michaelis-Menten Kinetics (Non-linear; concentration dependent)

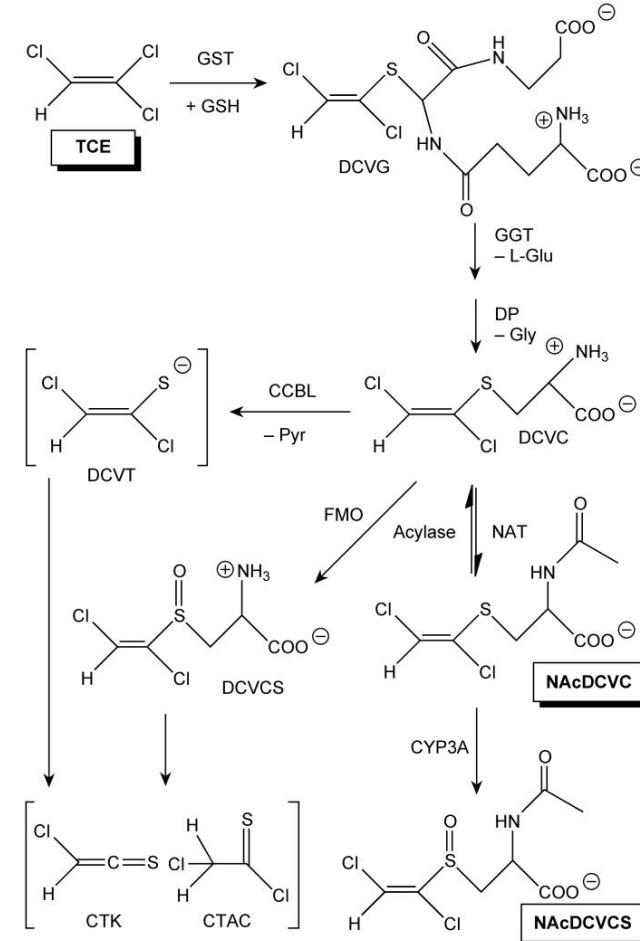


Trichloroethylene GSH conjugation pathway

- Initial conjugation with GSH
- Subsequent biotransformation occurs in multiple tissues via both Phase I and Phase II metabolism



Lash, L.H., et al. 2014. *Trichloroethylene biotransformation and its role in mutagenicity*.



Excretion

Excretion is the removal from the body

- Urine
- Exhalation
- Feces
- Minor pathways: sweat, saliva, milk

Excretion and elimination are often confused

- Excretion is reserved for exiting the body
- Elimination is the disappearance of a chemical, including both excretion and metabolism

Impact of TK on risk assessment



- Concentration prediction
- Identifying the toxicologically active agent(s)
- Providing a basis for extrapolating from experimental studies to humans
- Characterizing how humans may vary in their response to exposure due to differences in internal dose

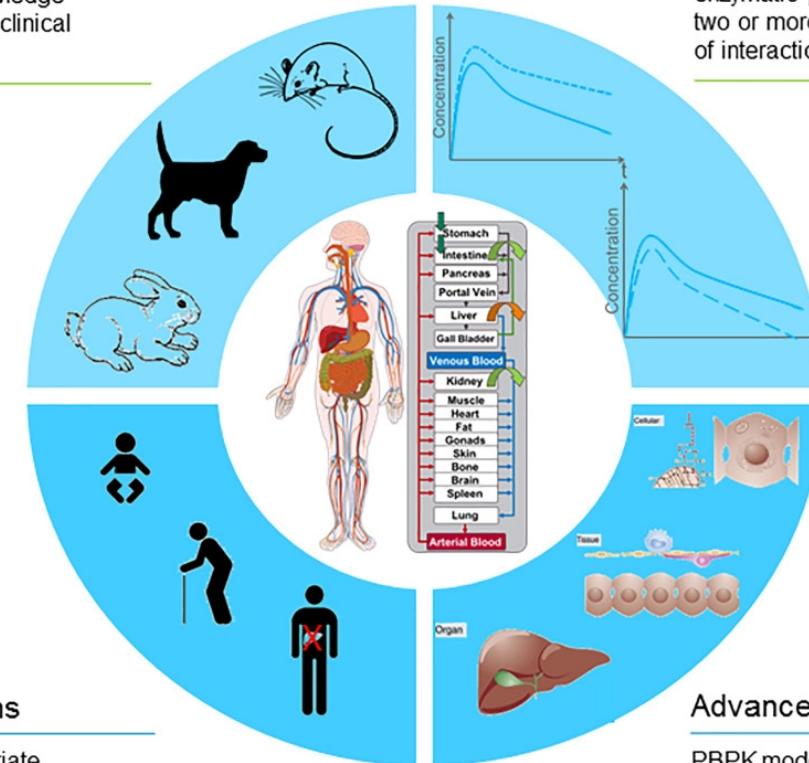
Pharmacokinetic Models

Why Pharmacokinetic Modeling

- Pharmaceutical research
- Drug development
- Health risk assessment

Cross-Species Extrapolation

PBPK models can be used to facilitate the extrapolation of knowledge generated in various preclinical species to humans



Special Populations

By including the appropriate physiological information, PBPK models can be used to make predictions in special populations

Drug Drug Interactions (DDI)

Thanks of the explicit inclusion of enzymatic processes, the combination of two or more models allow the prediction of interaction between drugs

Advanced Applications

PBPK models can also be integrated in more complex models such as multiscale modelling or statistical modelling, using methods such as Bayesian approaches

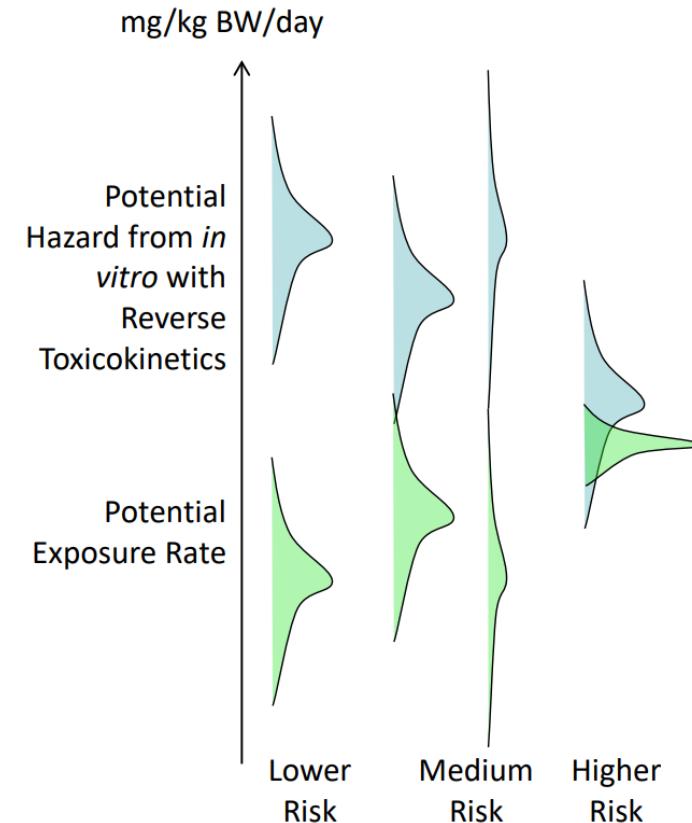
High Throughput Risk Prioritization



According to Toxic Substances Control Act (TSCA), to address thousands of chemicals, we need to use **new approach methodologies (NAMs)** to prioritize the existing and new chemicals for testing.

Main components

1. High throughput hazard characterization
2. High throughput exposure forecasts
3. High throughput **toxicokinetics**



Pharmacokinetic Model



What information we want to know through "**prediction**"?

Time

- When the chemical can reach the peak concentration in the body?
- How long it stay in the body?

Concentration

- What is the peak concentration?
- What is the average concentration?

Mathematical method



How to conduct simulation?

Numerical

- Use the differential equation containing one or more functions of one independent variable and the derivatives of those functions.

$$\frac{dA}{dt} = -k_e \cdot A$$

A : Amount in body (mass); D_0 initial amount (mass); t : time; k_e : Elimination rate constant (/time)

Analytical

- The simple equation with a independent variable of time

$$A(t) = A_0 \cdot e^{-k_e \cdot t}$$

Intravenous administration

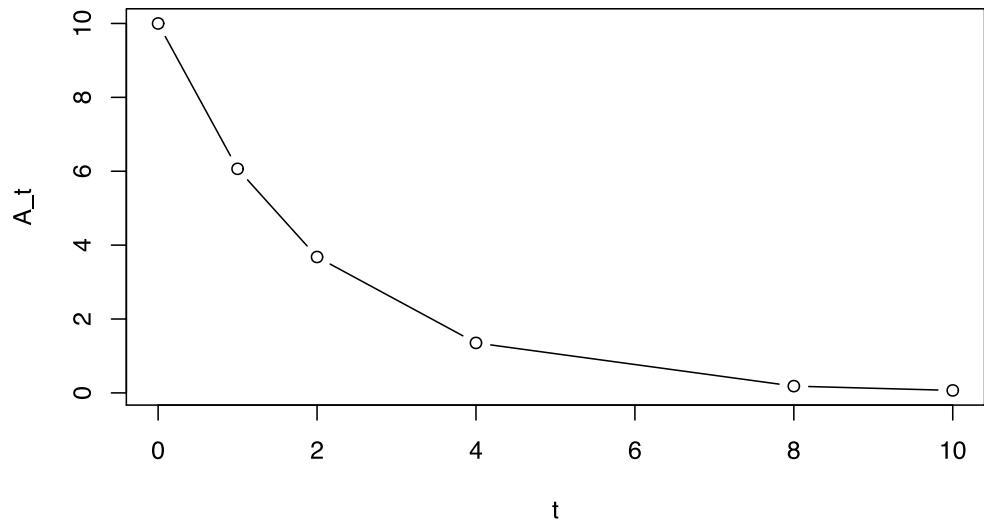
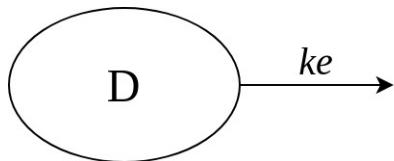
Numerical

$$\frac{dA}{dt} = -k_e \cdot A$$

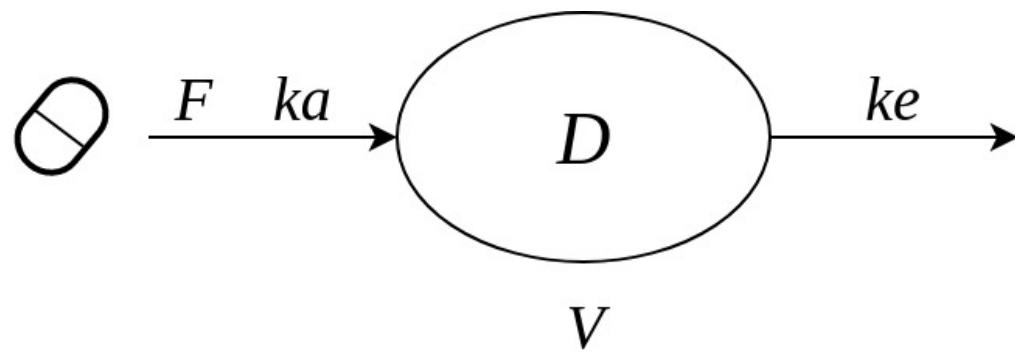
```
A_0 <- 10
t <- c(0, 1, 2, 4, 8, 10)
k_e <- 0.5
A_t <- A_0*exp(-k_e*t)
plot(t, A_t, type = "b")
```

Analytical

$$A_t = A_0 \cdot e^{-k_e \cdot t}$$



Oral administration



$$C(t) = \frac{F \cdot D \cdot k_a}{V(k_a - k_e)} \cdot (e^{-k_e t} - e^{-k_a t})$$

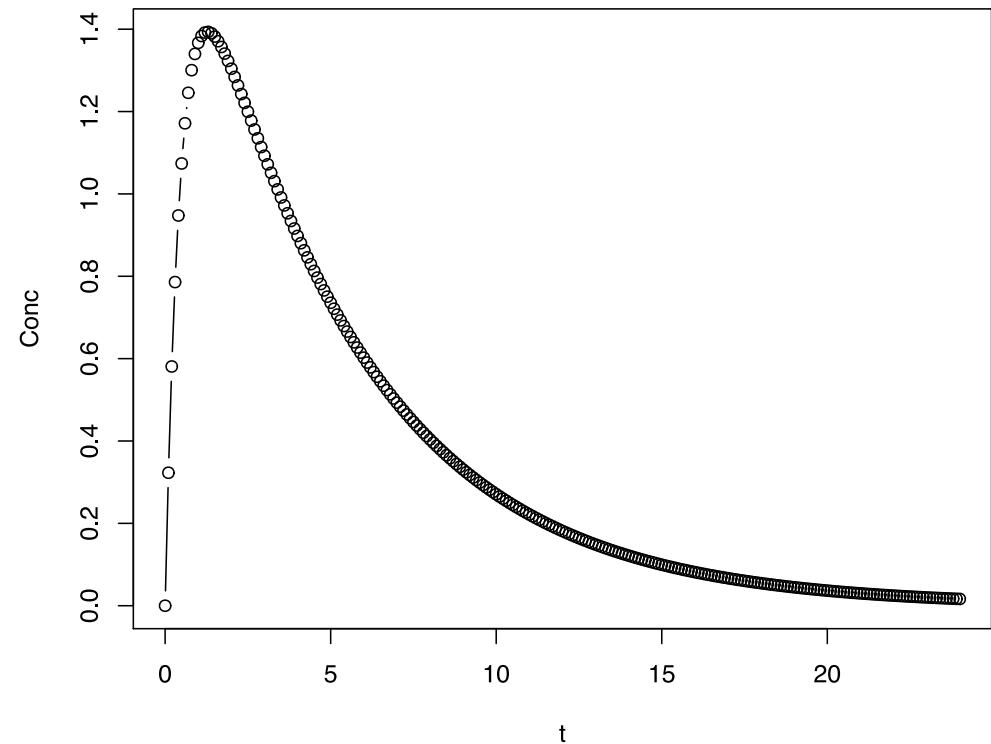
D: Dose (mass)

F: Bioavailability (-)

K_a Absorption rate constant (/time)

k_e Elimination rate constant (/time)

V: Distribution Volume (Vol)



Empirical compartmental models



Advantages

- Can continue adding compartments to fit more complex concentration-time profiles.
- Can provide useful descriptions of the data.

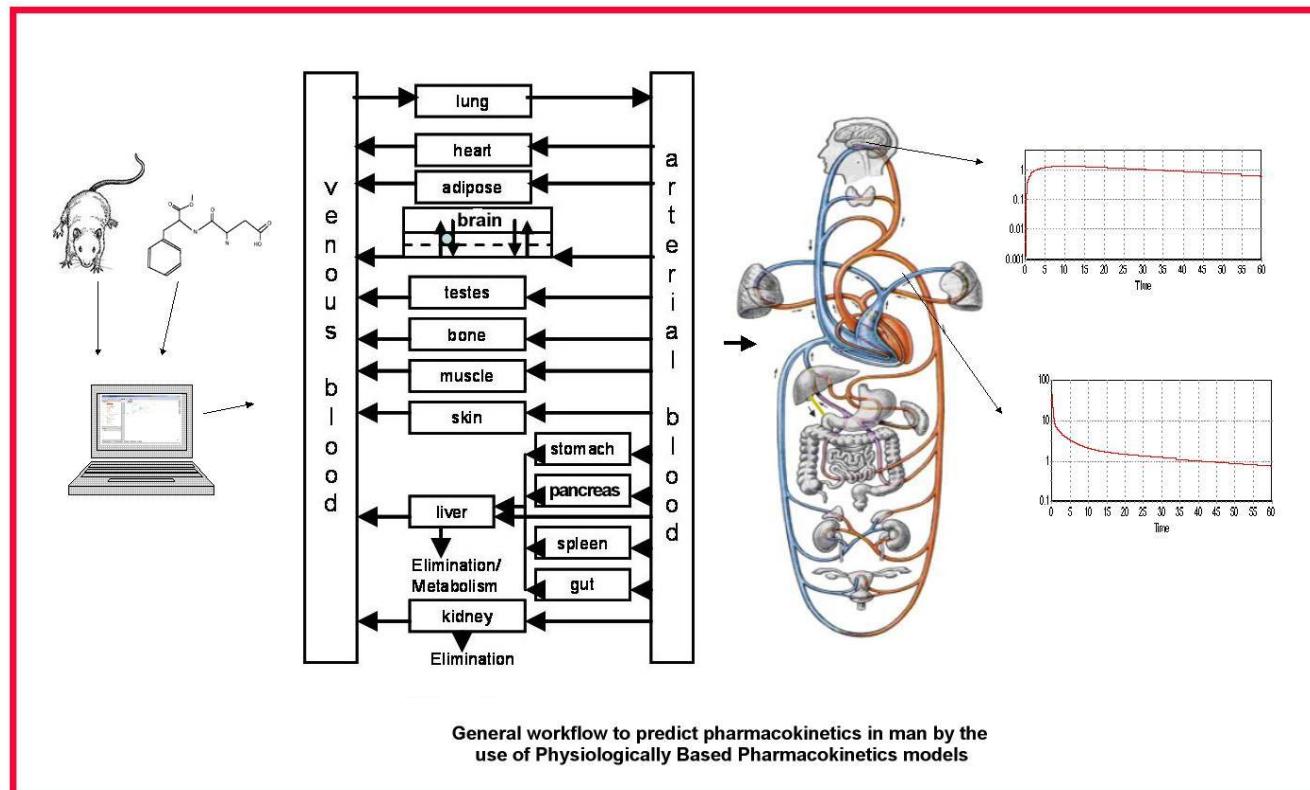
Disadvantages

- Cannot interpret compartments biologically.
- Many A,D,M, and E processes are interconnected.
- Cannot make predictions beyond the experiment being described.
 - Different tissues
 - Different exposure routes or conditions
 - Different species
 - Different individuals
- Cannot incorporate everything else we know about biology!

Physiological-Based Pharmacokinetic (PBPK)



Mathematically transcribing **anatomical, physiological, physical, and chemical** descriptions of the phenomena involved in the complex **ADME** processes.



Why PBPK models?



- There are some processes that empirical models have difficulty simulating – such as inhalation exposures.
- We know a lot about the physiology/anatomy of the organisms we study and want to protect.
 - Organ/tissue sizes
 - Connections between organs/tissues
 - Blood flow rates
 - Ventilation rates
 - Metabolizing enzymes
 - Etc...

By developing a model, we can incorporate that information to help us make inferences beyond the experimental data at hand.

Building a PBPK model

- Decide on what compartments are needed
 - Target vs. non-target tissues
 - Tissues with in vivo vs. in vitro vs. no data
 - Exposure routes
- Specify the (interrelated) equations for each compartment
- Specify parameter values
 - Physiological/anatomical parameters
 - Chemical-specific parameters
- Set up the inputs (doses, exposure concentrations, etc.) and outputs (blood, tissue, exhaled breath concentrations).
- Solve using numerical differential equation solver and compare with data

Iteration may be needed for refining model and/or conducting additional experiments

PBPK Workflow

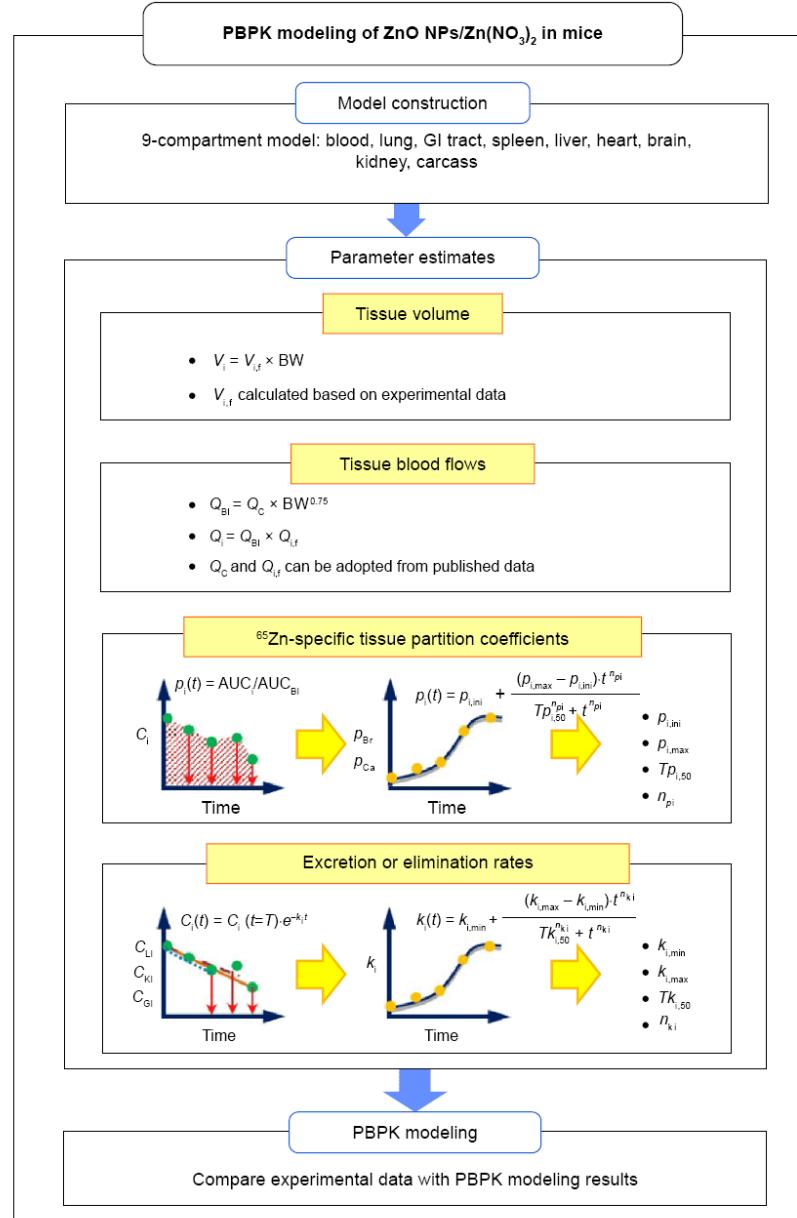
Model Construction

- What type of data we have?
 - A: Input routes
 - D: Blood and other tissues
 - M: Metabolism
 - E: Urine or feces

Parameter Estimates

- Do we have any prior information?
 - Physiological parameter
 - Chemical-specific parameter

PBPK modeling



Compartmental equations



- Flow in and out of “storage” compartments involve only blood flow
- Key parameters/variables

Q_i = tissue blood flow

P_i = tissue-blood partition coefficient

V_i = volume of tissue

C_a = arterial blood concentration

CV_i = venous blood concentration leaving tissue

A_i = amount in tissue

C_i = concentration in tissue

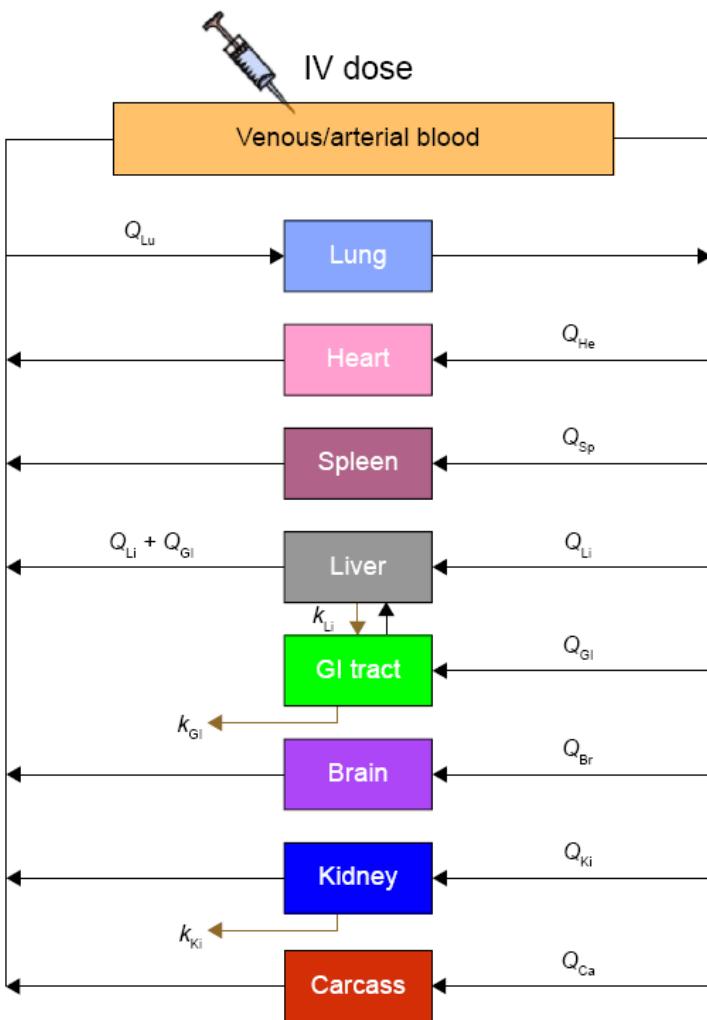


Figure 2 Constructed PBPK model describing the transportation of 10 nm and 71 nm ^{65}ZnO NPs and $^{65}\text{Zn}^{2+}$ and interactions between blood and tissues or organs.

Abbreviations: PBPK, physiologically based pharmacokinetic; NP, nanoparticle; IV, intravenous; GI, gastrointestinal; Q , blood flow to organ; k , excretion or elimination rate; Li, liver; Ki, kidney; Sp, spleen; Lu, lung; Br, brain; He, heart; Ca, carcass.

Table 2 Physiological parameters used in PBPK model for ^{65}ZnO NPs and $^{65}\text{Zn}(\text{NO}_3)_2$ in mice

Symbol	Value	Description (unit)	Source
BW	0.032	Body weight (kg)	Estimated
Q_c	9.025	Cardiac output constant ($\text{L} \cdot \text{h}^{-1}$)	Davies and Morris ¹³
Organ volume (as percentage of BW)			
V_{Bl}	0.060	Blood volume	Estimated
V_{Lu}	0.007	Lung volume	Estimated
V_{Gl}	0.127	GI tract volume	Estimated
V_{Sp}	0.004	Spleen volume	Estimated
V_{Li}	0.059	Liver volume	Estimated
V_{He}	0.007	Heart volume	Estimated
V_{Br}	0.014	Brain volume	Estimated
V_{Ki}	0.016	Kidney volume	Estimated
V_{Ca}	0.706	Carcass volume	Estimated
Blood flow to organ (as percentage of cardiac output)			
Q_{Lu}	1	Lung blood flow	Brown et al ¹⁸
Q_{Gl}	0.188	GI tract blood flow	Davies and Morris ¹³
Q_{Sp}	0.011	Spleen blood flow	Davies and Morris ¹³
Q_{Li}	0.161	Liver blood flow	Davies and Morris ¹³
Q_{He}	0.060	Heart blood flow	Brown et al ¹⁸
Q_{Br}	0.030	Brain blood flow	Brown et al ¹⁸
Q_{Ki}	0.091	Kidney blood flow	Brown et al ¹⁸
Q_{Ca}	0.454	Carcass blood flow	Estimated

Abbreviations: PBPK, physiologically based pharmacokinetic; NP, nanoparticle; BW, body weight; GI, gastrointestinal; Q_c , cardiac output constant; V , tissue volume; Q , Blood flow to organ; Bl, blood; Li, liver; Ki, kidney; Sp, spleen; Lu, lung; Br, brain; He, heart; Ca, carcass.

Basic transport equations



The model equations follow the principles of mass transport, fluid dynamics, and biochemistry in order to simulate the fate of a substance in the body.

$$\text{rate in} = Q_i C_{art} \quad \& \quad \text{rate out} = Q_i C_{VT} = Q_i \cdot C_i / P_i$$

Change in amount = rate in – rate out

$$\frac{dA_i}{dt} = Q_i \left(C_{art} - \frac{A_i}{P_i V_i} \right)$$

or

$$\frac{dC_i}{dt} = \frac{Q_i}{V_i} \left(C_{art} - \frac{C_i}{P_i V_i} \right)$$

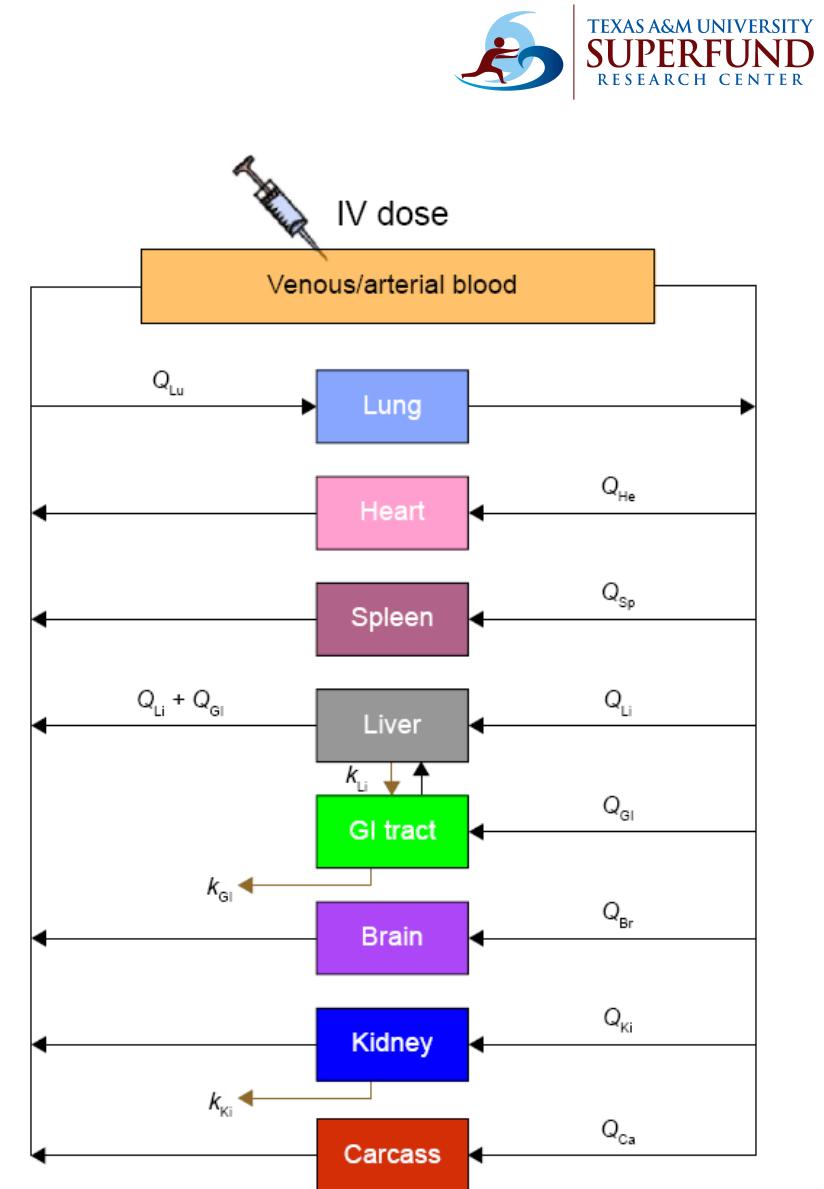
A_i is amount of chemical, Q_i is blood flow, C_{art} incoming arterial blood concentration, P_i the tissue over blood partition coefficient and V_i the volume of compartment i .

Table I Physiologically based pharmacokinetic model equations in mice

Compartments	Equations	Equation ID
Blood (C_{Bi} , $\mu\text{g}\cdot\text{mL}^{-1}$)	$\frac{dC_{Bi}}{dt} = \frac{I}{V_{Bi}} \left(Q_{Lu} \cdot \frac{C_{Lu}}{p_{Lu}} + Q_{Sp} \cdot \frac{C_{Sp}}{p_{Sp}} + (Q_{Li} + Q_{Gi}) \cdot \frac{C_{Li}}{p_{Li}} + Q_{He} \cdot \frac{C_{He}}{p_{He}} + Q_{Br} \cdot \frac{C_{Br}}{p_{Br}} + Q_{Ki} \cdot \frac{C_{Ki}}{p_{Ki}} + Q_{Ca} \cdot \frac{C_{Ca}}{p_{Ca}} - \frac{C_{Bi}}{V_{Bi}} (Q_{Lu} + Q_{Sp} + Q_{Li} + Q_{He} + Q_{Br} + Q_{Ki} + Q_{Ca} + Q_{Gi}) \right)$	(A)
Lung (C_{Lu} , $\mu\text{g}\cdot\text{g}^{-1}$)	$\frac{dC_{Lu}}{dt} = \frac{Q_{Lu}}{V_{Lu}} \left(C_{Bi} - \frac{C_{Lu}}{p_{Lu}} \right)$	(B)
GI tract (C_{Gi} , $\mu\text{g}\cdot\text{g}^{-1}$)	$\frac{dC_{Gi}}{dt} = \frac{Q_{Gi}}{V_{Gi}} \left(C_{Bi} - \frac{C_{Gi}}{p_{Gi}} \right) + C_{Li} \cdot k_{Li}/p_{Li} - C_{Gi} \cdot k_{Gi}/p_{Gi}$	(C)
Spleen (C_{Sp} , $\mu\text{g}\cdot\text{g}^{-1}$)	$\frac{dC_{Sp}}{dt} = \frac{Q_{Sp}}{V_{Sp}} \left(C_{Bi} - \frac{C_{Sp}}{p_{Sp}} \right)$	(D)
Liver (C_{Li} , $\mu\text{g}\cdot\text{g}^{-1}$)	$\frac{dC_{Li}}{dt} = \frac{I}{V_{Li}} \left(Q_{Lu} \cdot C_{Lu} + Q_{Gi} \cdot \frac{C_{Gi}}{p_{Gi}} - (Q_{Li} + Q_{Gi}) \cdot \frac{C_{Li}}{p_{Li}} \right) - C_{Li} \cdot k_{Li}/p_{Li}$	(E)
Heart (C , $\mu\text{g}\cdot\text{g}^{-1}$)	$\frac{dC_{He}}{dt} = \frac{Q_{He}}{V_{He}} \left(C_{Bi} - \frac{C_{He}}{p_{He}} \right)$	(F)
Brain (C_{Br} , $\mu\text{g}\cdot\text{g}^{-1}$)	$\frac{dC_{Br}}{dt} = \frac{Q_{Br}}{V_{Br}} \left(C_{Bi} - \frac{C_{Br}}{p_{Br}} \right)$	(G)
Kidney (C_{Ki} , $\mu\text{g}\cdot\text{g}^{-1}$)	$\frac{dC_{Ki}}{dt} = \frac{Q_{Ki}}{V_{Ki}} \left(C_{Bi} - \frac{C_{Ki}}{p_{Ki}} \right) - C_{Ki} \cdot k_{Ki}/p_{Ki}$	(H)
Carcass (C_{Ca} , $\mu\text{g}\cdot\text{g}^{-1}$)	$\frac{dC_{Ca}}{dt} = \frac{Q_{Ca}}{V_{Ca}} \left(C_{Bi} - \frac{C_{Ca}}{p_{Ca}} \right)$	(I)

Abbreviations: C, concentration; V, tissue volume; Q, Blood flow to organ; p, partition coefficient; k, excretion or elimination rate; Bi, blood; Li, liver; Ki, kidney; Sp, spleen; Lu, lung; Br, brain; He, heart; Gi, gastrointestinal; Ca, carcass.

Chen et al. (2015) **Physiologically based pharmacokinetic modeling of zinc oxide nanoparticles and zinc nitrate in mice.**



Different exposure routes

Inhalation

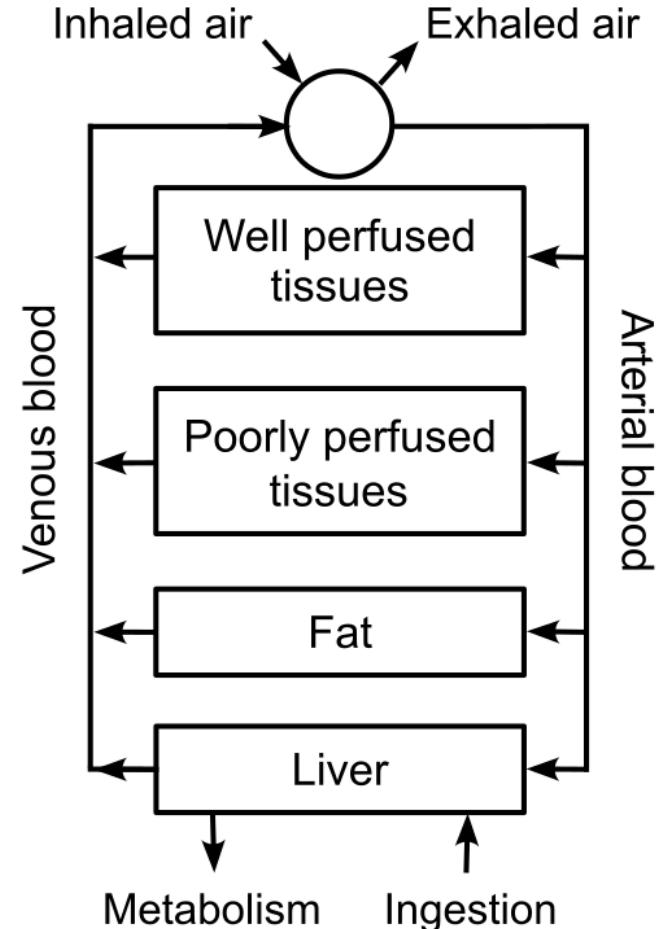
$$C_{\text{art}} = \frac{Q_{\text{pul}} (1 - r_{\text{ds}}) C_{\text{inh}} + Q_{\text{tot}} C_{\text{ven}}}{Q_{\text{pul}} (1 - r_{\text{ds}}) / P_{\text{air}} + Q_{\text{tot}}}$$

Ingestion

$$\frac{\partial A_{\text{iv}}}{\partial t} = Q_{\text{liv}} \left(C_{\text{art}} - \frac{A_{\text{liv}}}{P_{\text{liv}} V_{\text{liv}}} \right) - k_{\text{met}} A_{\text{liv}} + R_{\text{ing}}$$

A: quantity; *Q*: flow rate; *r_{ds}*: deadspace ratio; *C*: concentration; *R_{ing}*: administration rate; *P* partition coefficient; *k* rate constant

Bois F.Y., Brochot C. (2016) **Modeling Pharmacokinetics**. In: Benfenati E. (eds) In Silico Methods for Predicting Drug Toxicity. Methods in Molecular Biology



Computational toolkits

Computational toolkits



Multiple software packages available to solve differential equations

- ASCL – “traditionally” used by many established PBPK modelers (but now discontinued!)
- Berkeley Madonna – used in some PBPK modeling courses
- MatLab – flexible interactive interface (more expensive, but more powerful)
- GNU MCSim – specialized software for Bayesian uncertainty and population variability analyses

All of them incorporate methods for “optimizing” the model fit by adjusting parameter values.

Computational toolkits



GNU MCSim

Simulation package, which allows you to:

- design and run simulation models (using algebraic or differential equations)
- perform Monte Carlo stochastic simulations
- do Bayesian inference through Markov Chain Monte Carlo simulations
- has faster computing speed than other simulation software/packages (e.g., Asclx, Berkeley Madonna, RStan)

R

Programming language that allows you to:

- conduct statistical analysis (summarization, estimation)
- visualize simulation results
- use various packages to analyze results (e.g., CODA, BOA, rstan)
- perform sensitivity analysis (e.g., sensitivity, pksensi)
- access community support (e.g., Stack Overflow, R User groups)

Advantages of MCSim and R

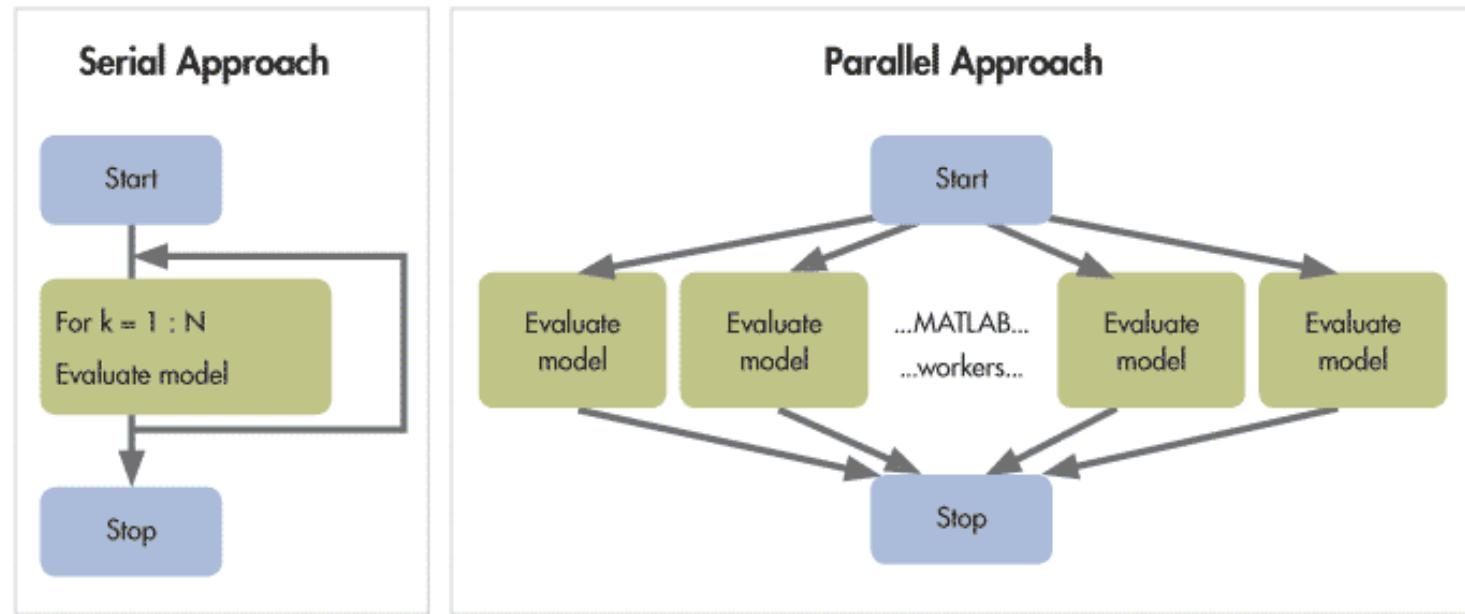
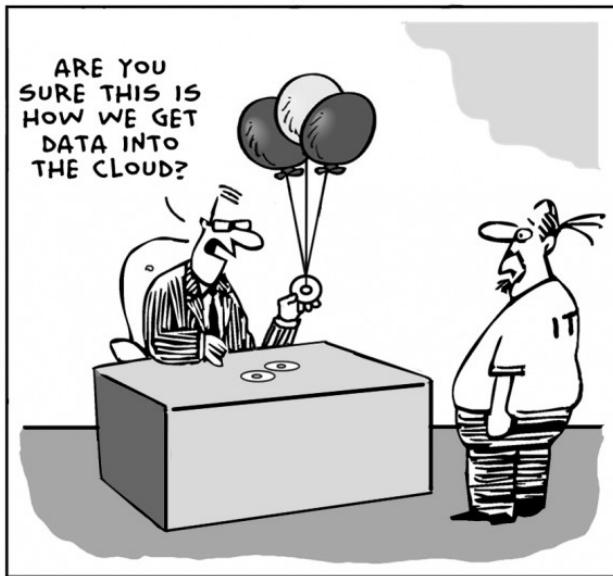
"Free and Open Source Software" under GNU General Public License

- The freedom to run the program as you wish, for any purpose (freedom 0).
- The freedom to study how the program works, and change it so it does your computing as you wish (freedom 1). Access to the source code is a precondition for this.
- The freedom to redistribute copies so you can help others (freedom 2).
- The freedom to distribute copies of your modified versions to others (freedom 3). By doing this you can give the whole community a chance to benefit from your changes. Access to the source code is a precondition for this.



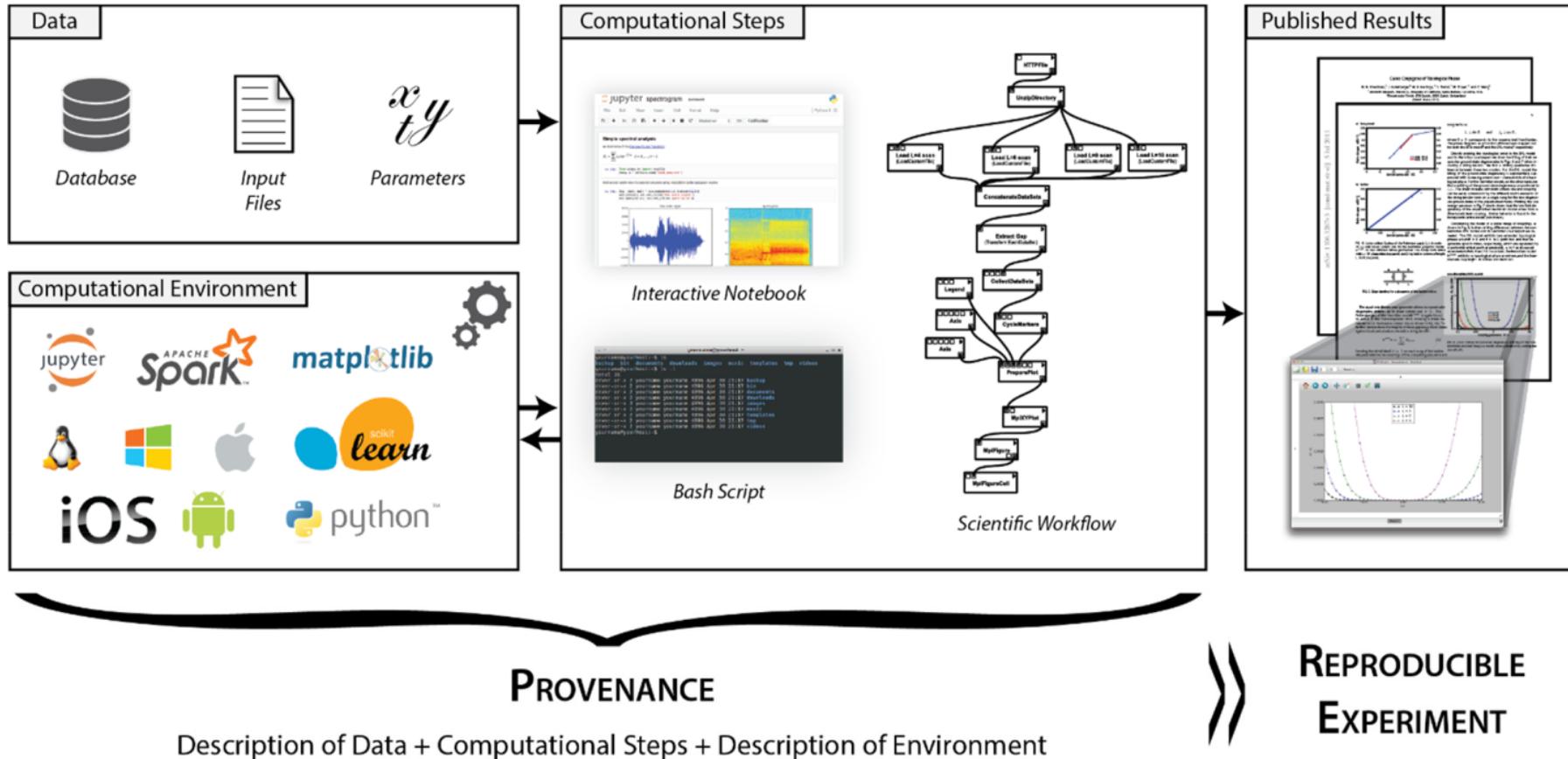
Advantages

"Parallel computing"



- Run multiple MCMC chains with multiple CPUs
- High performance (cloud) computing

Advantages - Reproducible research



Conducting data analysis through R

- Implement a wide variety of statistical and graphical techniques
- Comprehensive research workflow and toolkits (e.g., RMarkdown)
- Integration with low-level language (e.g., C, C++, Fortran)
- Highly extensible through the use of user-submitted packages
- Webapp development (e.g., <http://webpopix.org/shiny/ShinyExamples.html>)

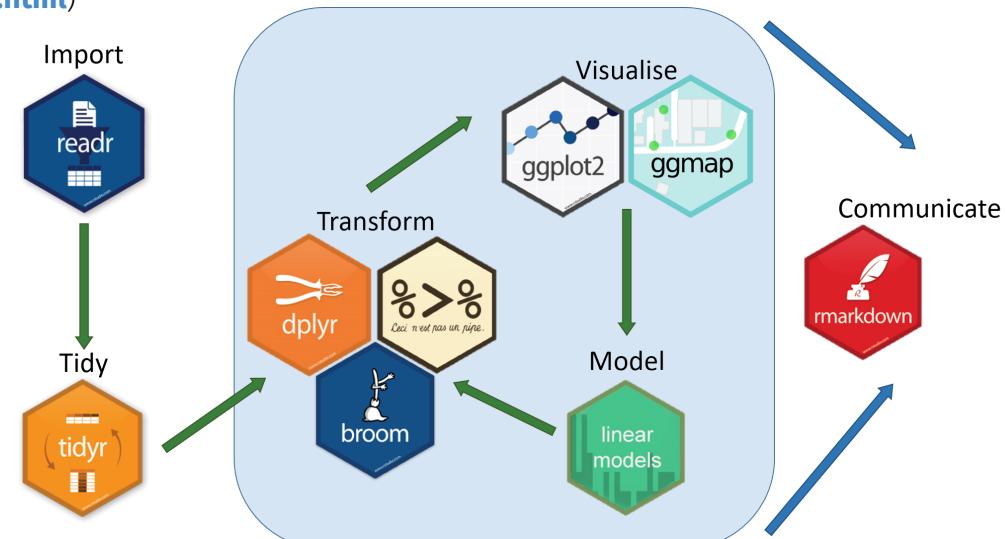


Image source

RStudio



Free and open-source integrated development environment

Powerful and user friendly programming interface

Designed to make it easy to write scripts

Easy to view and interact with the objects

R project with version control (e.g., git)

Support cloud computing <https://rstudio.cloud/>



Stack Overflow Trends

See how technologies have trended over time based on use of their tags since 2008, when Stack Overflow was founded. Enter up to 15 tags to compare growth and decline.

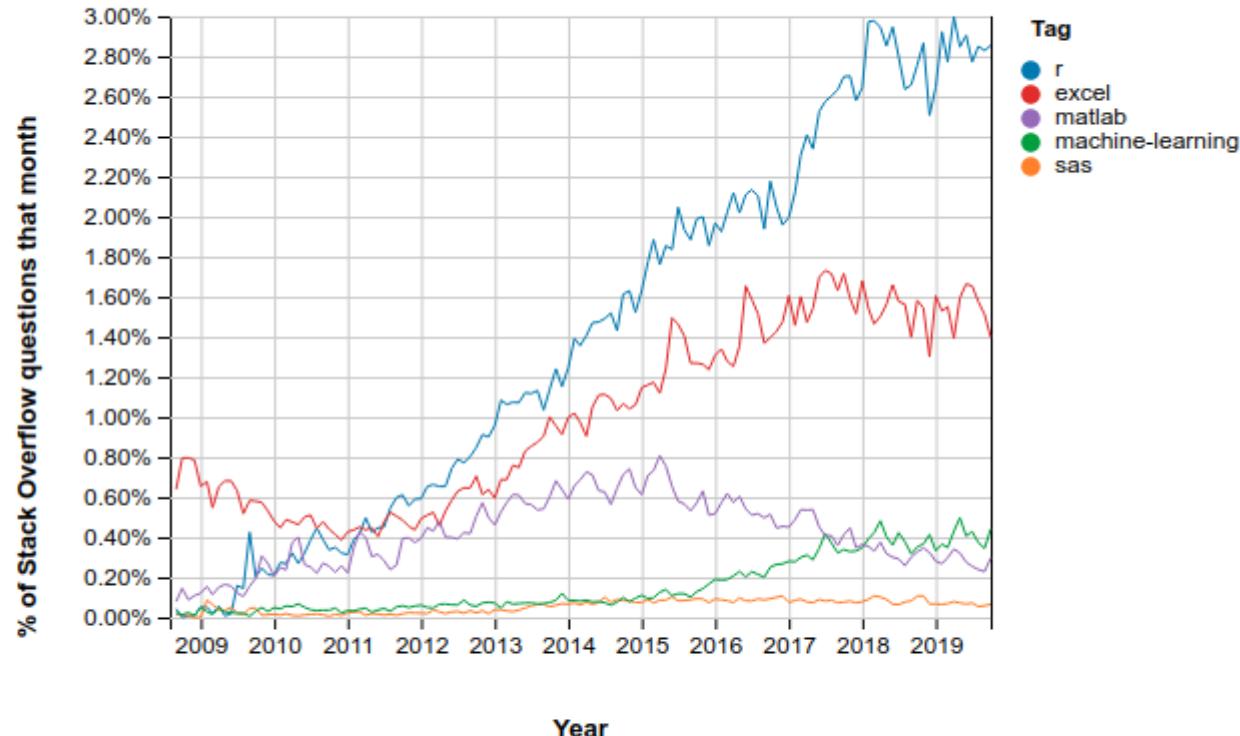
Tags:

r x sas x machine-learning x excel x

matlab x

.htaccess
.net
.net-2.0
.net-3.5
2d
32bit-64bit
3d
64-bit
abap
abstract-class
accessibility
acl

For more on this tool and what you can learn from it, see our [blog post](#).



GNU MCSim



- The project started by Don Maszle and **Frederic Y. Bois** in UC Berkeley, 1991. It is written in standard C language as a model preprocessor.
- First public release in 1993 (straight simulations with Monte Carlo modeling).

GNU MCSim is a general purpose modeling and simulation program which can performs "standard" or "Markov chain" Monte Carlo simulations. It allows you to specify a set of **linear or nonlinear algebraic equations** or **ordinary differential equations**. They are solved numerically using parameter values you choose or parameter values sampled from statistical distributions. Simulation outputs can be compared to experimental data for Bayesian parameter estimation (model calibration).

- 6.1.0 (19 February 2019)
- 6.0.1 (05 May 2018)
- 6.0.0 (24 February 2018)
- 5.6.6 (21 January 2017)
- 5.6.5 (27 February 2016)
- 5.6.4 (30 January 2016)
- 5.6.3 (1 January 2016)
- 5.6.2 (24 December 2015)
- 5.6.1 (21 December 2015)
- 5.6.0 (16 December 2015)
- 5.5.0 (17 March 2013)
- 5.4.0 (18 January 2011)
- 5.3.1 (3 March 2009)
- 5.3.0 (12 January 2009)
- 5.2 beta (29 January 2008)
- 5.1 beta (18 September 2006)
- 5.0.0 (4 January 2005)
- 4.2.0 (15 October 2001)
- 4.1.0 (1 August 1997)
- 4.0.0 (24 March 1997)

Founder: *Frédéric Y. Bois*

Staff Toxicologist (Specialist),
Reproductive and Cancer Hazard Assessment Section,
CalEPA, Berkeley, USA, 1991-96



GNU MCSim Overview



The *GNU MCSim* consists in two pieces, a **model generator** and a **simulation engine**:

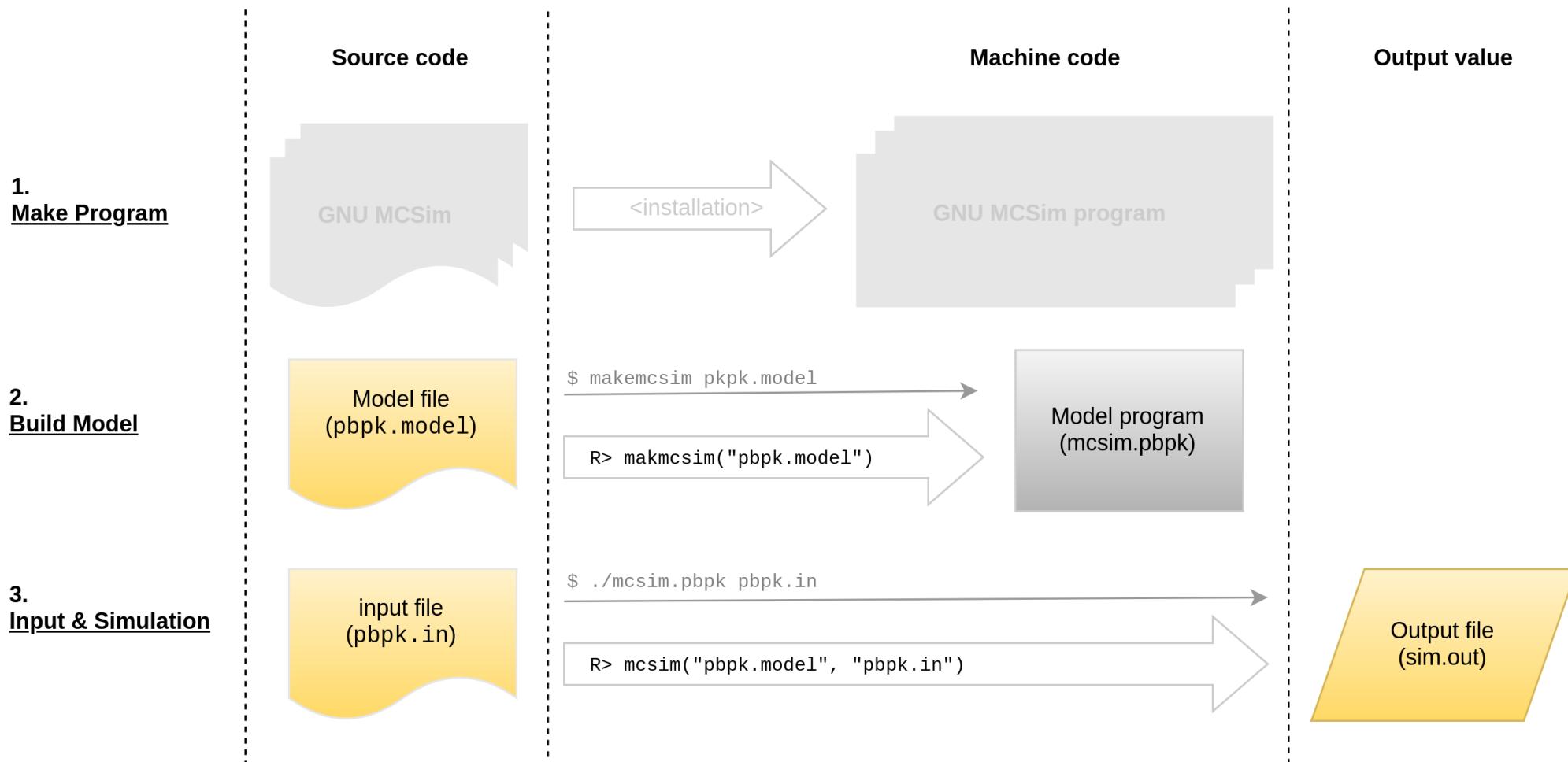
The model generator, "**mod**"

- Created to facilitate structural model definition and maintenance, while keeping execution time short. You can code your model using a simplified syntax and use mod to translate it to c (`model.c`).

The simulation engine, "**sim**"

- A set of routines which are linked to your model during compilation to produce executable program (`mcsim.model`). After that, you can run simulations of your model under a variety of conditions, specify an associated statistical model, and perform simulations.

GNU MCSim Workflow



Types of Simulation



Simple simulation

- Straight simulations (set parameter values and initial conditions).

Used to: *Model testing when building the model (e.g., mass balance)*

Monte Carlo simulations

- Perform repeated (stochastic) simulations across a randomly sampled region of the model parameter space.

Used to: *Check possible simulation (under given parameter distributions) results before model calibration*

SetPoints simulation

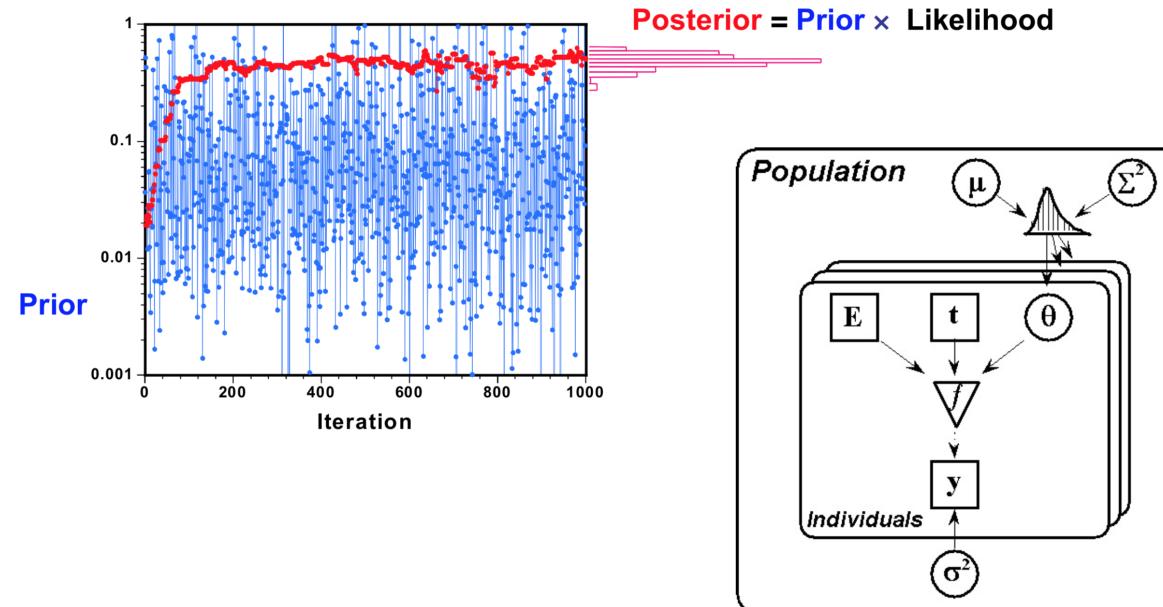
- Solves the model for a series of specified parameter sets. You can create these parameter sets yourself or use the output of a previous Monte Carlo or MCMC simulation.

Used to: *Posterior analysis, Local/global sensitivity analysis*

Types of Simulation

Markov-chain Monte Carlo (MCMC) simulation

- Performs a series of simulations along a Markov chain in the model parameter space.
- They can be used to obtain the Bayesian **posterior** distribution of the model parameters, given a statistical model, **prior** parameter distributions and data for which a **likelihood function** can be computed.
- GNU MCSim can handle hierarchical statistical models as well.



Source

MCSim-related R packages

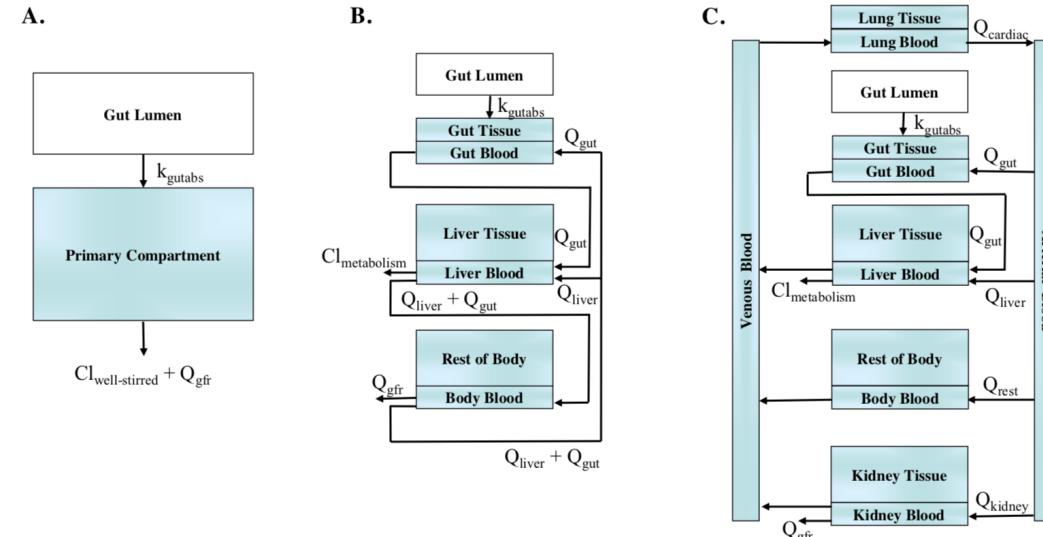
httR: R Package for High-Throughput Toxicokinetics

Robert G. Pearce, R. Woodrow Setzer, Cory L. Strope, Nisha S. Sipes, John F. Wambaugh

MCSim (Bois and Maszle 1997) was used for converting the model equations into C code, which is used with **deSolve** (Soetaert et al. 2016) in solving each system of equations.

Journal of Statistical Software; <http://dx.doi.org/10.18637/jss.v079.i04>

GNU MCSim model code → C code → **deSolve** package → Prediction



MCSim-related R packages



pksensi: R Package for Global Sensitivity Analysis in Pharmacokinetic Modeling

Nan-Hung Hsieh, Brad Reisfeld, Weihsueh A. Chiu

pksensi implements the global sensitivity analysis workflow to investigate the parameter uncertainty and sensitivity in pharmacokinetic (PK) models, especially the physiologically based pharmacokinetic (PBPK) model with multivariate outputs.

CRAN 1.1.4 – 2019-09-19

Two types of model solver

`solve_fun()`

GNU MCSim model code → C code → **deSolve** package → Prediction

`solve_mcsim()`

GNU MCSim model code → Prediction

Note: `solve_mcsim()` is faster than `solve_fun()`

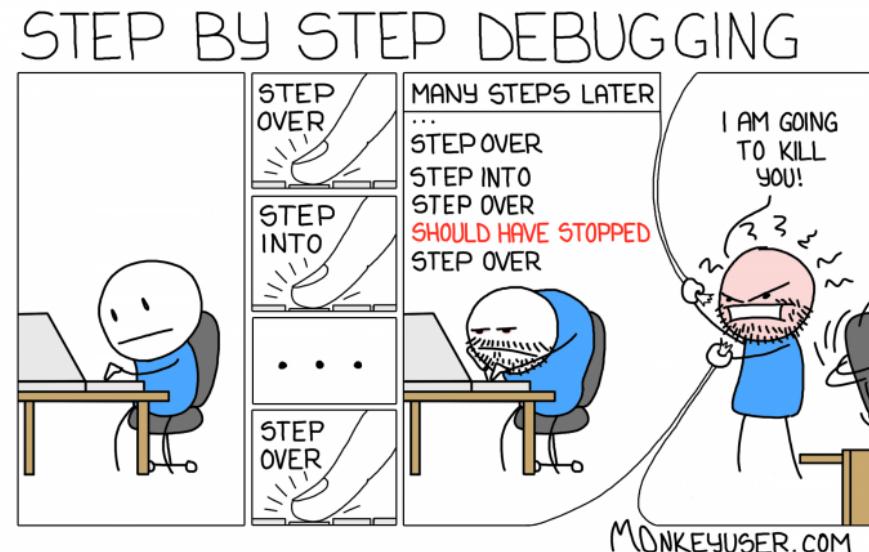
Disadvantages in MCSim and R

Difficult learning curve (command line interface-based)

Requires coding/programming skill

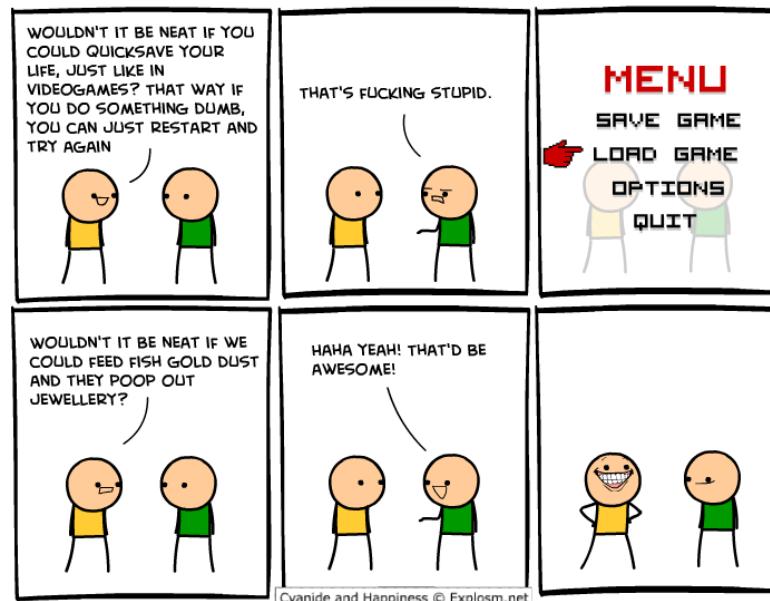
Requires installation of extra program or package

Requires "**debugging**"



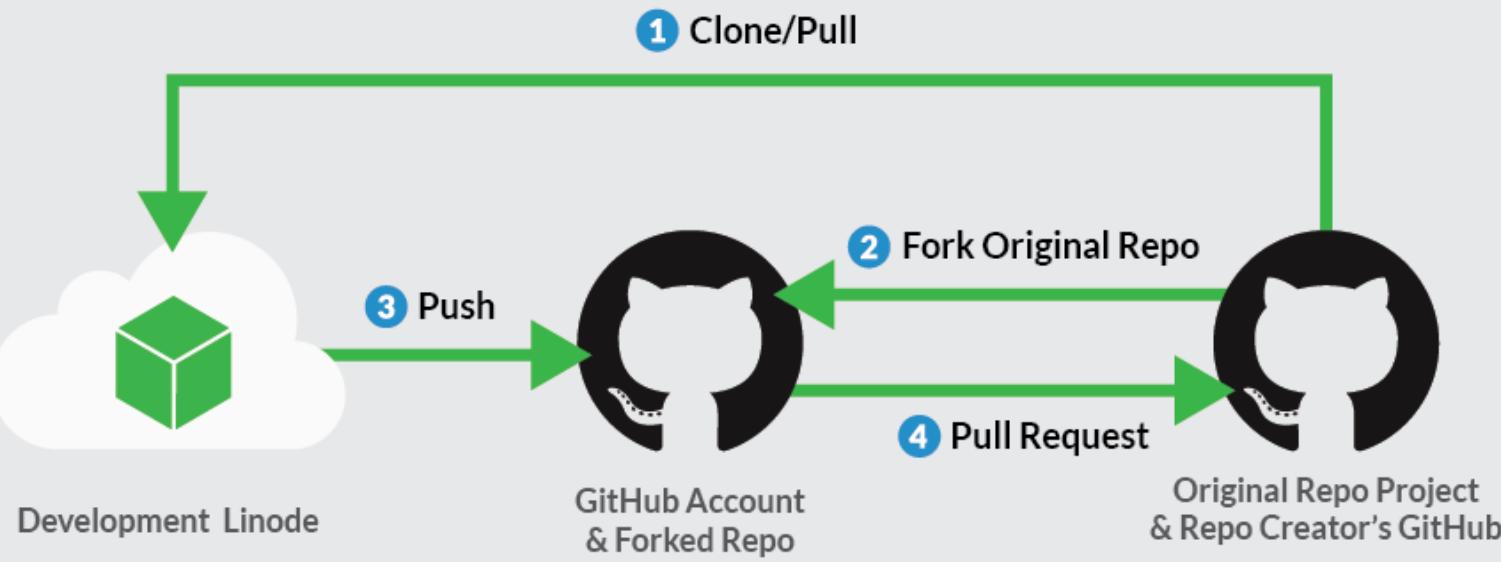
Git

- Manage your code (models, inputs, R script)
- Transfer your file (through GitHub or GitLab)
- Collaboration work



GitHub

Git and GitHub



U.S. Environmental Protection Agency

github.com/USEPA

Search or jump to... Pull requests Issues Marketplace Explore

U.S. Environmental Protection Agency <http://www.epa.gov>

Repositories 186 Packages People 21 Projects

Pinned repositories

- OneEPA-Standalone-App-Template**
Standalone application template for non-www EPA.gov applications
PLSQL ★ 12 8
- WNTR**
An EPANET compatible python package to simulate and analyze water distribution networks under disaster scenarios.
Python ★ 100 70
- CMAQ**
Code base for the U.S. EPA's Community Multiscale Air Quality Model (CMAQ). For additional background on CMAQ please visit: www.epa.gov/CMAQ
Fortran ★ 88 78

open-source-projects
This repository contains information on how to work with EPA open source projects.
5

Find a repository... Type: All Language: All

useeior
R CC0-1.0 2 2 2 10 Updated 4 hours ago

epanet-solver
ORD Water Distribution Network Model
C MIT 26 51 2 1 Updated yesterday

kdhe_region7_r
TeX CC0-1.0 0 0 0 0 Updated yesterday

Top languages

- R Python JavaScript HTML Jupyter Notebook

People 21 >

The screenshot shows the GitHub organization page for the U.S. Environmental Protection Agency (USEPA). The page features a dark header with the agency logo and a search bar. Below the header, there are tabs for Repositories (186), Packages, People (21), and Projects. A section titled "Pinned repositories" displays four repositories: "OneEPA-Standalone-App-Template", "WNTR", "CMAQ", and "open-source-projects". Each pinned repository card includes a thumbnail, the repository name, a brief description, and statistics (language, stars, forks, issues, pull requests). Below the pinned repositories, there is a search bar and dropdown menus for "Type: All" and "Language: All". Further down, three more repositories are listed: "useeior", "epanet-solver", and "kdhe_region7_r", each with its own card showing language, license, and update status. To the right of the repositories, there is a "Top languages" chart and a "People" section showing a grid of user profiles. The overall layout is clean and organized, typical of a GitHub organization page.

High Performance Research Computing



The efficient way to conduct parallel computing

A screenshot of a web browser displaying the Texas A&M High Performance Research Computing (HPRC) website. The URL in the address bar is hprc.tamu.edu/resources/. The page has a dark header with the Texas A&M logo and the text "TEXAS A&M HIGH PERFORMANCE RESEARCH COMPUTING". Below the header is a navigation menu with links for Home, User Services, Resources, Research, Policies, Events, and About. A search icon is also present. The main content area features a large image of server racks in a data center. On the left, there is a sidebar with sections for "Resources" (listing HPC Systems, Terra, Ada, Curie, Lonestar, Workstations, Software, Documentation) and "Quick Links" (listing New User Information, Accounts, Apply for Accounts, Manage Accounts, User Consulting, Training, Documentation, Software, FAQ). The main content area has a section titled "RESOURCES" with text about the HPRC group's three clusters. Below this is a section titled "Terra" with a photograph of the Terra HPC cluster and a detailed description of its hardware. At the bottom of the page are three buttons: "System Information", "Quick Start Guide", and "User Guide".

TEXAS A&M HIGH PERFORMANCE RESEARCH COMPUTING

Home User Services Resources Research Policies Events About

RESOURCES

The HPRC group currently administers three HPC clusters totaling 920 TF in peak performance with 11 PB of high-performance storage. Consult the [resource comparison page](#) for the hardware differences among these HPC clusters.

Terra

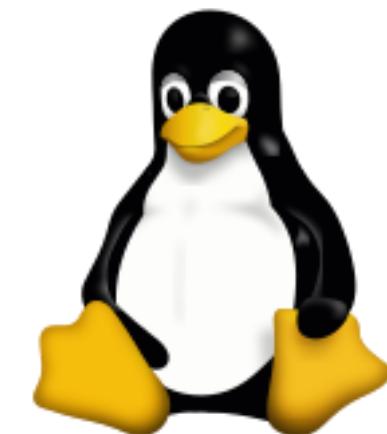
Terra is a 320-node heterogeneous Intel cluster from Lenovo with an Omni-Path Architecture (OPA) interconnect and 48 NVIDIA K80 dual-GPU accelerators. There are 304 nodes based on the Intel Broadwell processor and 16 nodes based on the Intel Knights Landing processor.

System Information Quick Start Guide User Guide

The open source Unix-like operating systems for high performance research computing

```
nhsieh@login-0502:~
```

```
*****  
*      == IMPORTANT POLICY INFORMATION ==  
*  
* -Unauthorized use of HPRC resources is prohibited and subject to  
*   criminal prosecution.  
* -Use of HPRC resources in violation of United States export control laws  
*   and regulations is prohibited. Current HPRC staff members are  
*   US citizens and legal residents.  
* -Sharing HPRC account and password information is in violation of State  
*   Law. Any shared accounts will be DISABLED.  
* -Authorized users must also adhere to ALL policies at:  
*   https://hprc.tamu.edu/policies  
*****  
  
!! WARNING: There are NO active backups of user data. !!  
  
Please restrict usage to 8 CORES across ALL Terra login nodes.  
Users found in violation of this policy will be SUSPENDED.  
  
**** Terra Cluster Maintenance, December 17 ****  
  
The Terra cluster will be unavailable from 9am to 6pm on Tuesday,  
December 17th. Software and hardware maintenance will be performed  
during this downtime. Jobs will not be scheduled if they will overlap  
with this maintenance window.  
  
**** Two-Factor Authentication for HPRC Clusters, November 4th ****  
  
Starting on November 4, 2019, Duo two-factor authentication will be  
required for all SSH logins to the HPRC clusters. The test system,  
duotest.hprc.tamu.edu, is available for HPRC users to try Duo  
two-factor authentication with their existing SSH/SCP/SFTP programs.  
  
For more information, visit the  
https://hprc.tamu.edu/wiki/Two\_Factor wiki page.  
  
To see these messages again, run the motd command.  
  
Your current disk quotas are:  
Disk      Disk Usage      Limit      File Usage      Limit  
/home      1.047G       10G        8621     10000  
/scratch    0           1T          1        50000  
Type 'showquota' to view these quotas again.  
[nhsieh@terra2 ~]$
```



Summary

- Role of pharmacokinetics in risk assessment
 - toxicity is driven by concentration at the **target tissue site**.
- Basic pharmacokinetics
 - **A,D,M,E** each may affect internal dose, and differences between species or among individuals.
- Pharmacokinetic modelling
 - Information from **physiology** and **chemistry** are common ways to characterize the concentration-time relationship.
- Computational toolkits
 - The **open source** computational tools (e.g., R & GNU MCSim) are powerful and can help us conduct the model simulation and data analysis

Hands on Exercise

Hands on Exercise



Task 1. **Exploratory analysis of PK data** (code: https://rpubs.com/Nanhung/SPR19_1)

- Learn how to use R to conduct basic analysis for PK data

Task 2. **PK model development** (code: https://rpubs.com/Nanhung/SPR19_2)

- Learn how to use R and MCSim to build a model

Task 3. **Parameter setting and model simulation** (code https://rpubs.com/Nanhung/SPR19_3)

- Understand the parameter setting in PK model and conduct the simulation

Task 4. **PBPK model development** (code https://rpubs.com/Nanhung/SPR19_4)

- Instead of PK model, we need to know how to build a PBPK model

Task 5. **Application of PBPK model** (code https://rpubs.com/Nanhung/SPR19_5)

- Here, we have a well-built PBPK model and its parameters, let's apply the model in the exposure assessment

Hands on Exercise

Task 1: Exploratory analysis of PK data

- Now, we have a theophylline PK dataset. The purpose of this exercise is to develop the simple PK model and use it to describe the PK of theophylline. First, look into the theophylline dataset. The `Theoph` data frame has 132 rows and 5 columns of data from an experiment on the pharmacokinetics of theophylline. Then, find the Cmax and Tmax for each individual.

```
head(Theoph)
```

```
##   Subject    Wt Dose Time  conc
## 1      1 79.6 4.02 0.00  0.74
## 2      1 79.6 4.02 0.25  2.84
## 3      1 79.6 4.02 0.57  6.57
## 4      1 79.6 4.02 1.12 10.50
## 5      1 79.6 4.02 2.02  9.66
## 6      1 79.6 4.02 3.82  8.58
```

- Plot the pharmacokinetic diagram for each individual.

Hands on Exercise

Task 2: PK model development

- Develop the non compartment model and compartment model in R and MCSim
- Non compartment model

$$C(t) = \frac{F \cdot D \cdot k_a}{V(k_a - k_e)} \cdot (e^{-k_e t} - e^{-k_a t})$$

- Compartment model

$$\frac{dA_{gut}}{dt} = -k_a A_{gut}$$

$$\frac{dA}{dt} = k_a A_{gut} - k_e A_e$$

$$C = A/V$$

Hands on Exercise

Task 3: Parameter setting and simulation

- Extract the chemical information from httk package

```
library(httk)
parms <- httk::parameterize_1comp(chem.name = "theophylline")
```

- Use the parameters in the developed model and conduct the simulation
- Compare the difference between data and the model simulation result (Cmax, Tmax)

Hands on Exercise



Task 4: PBPK model development

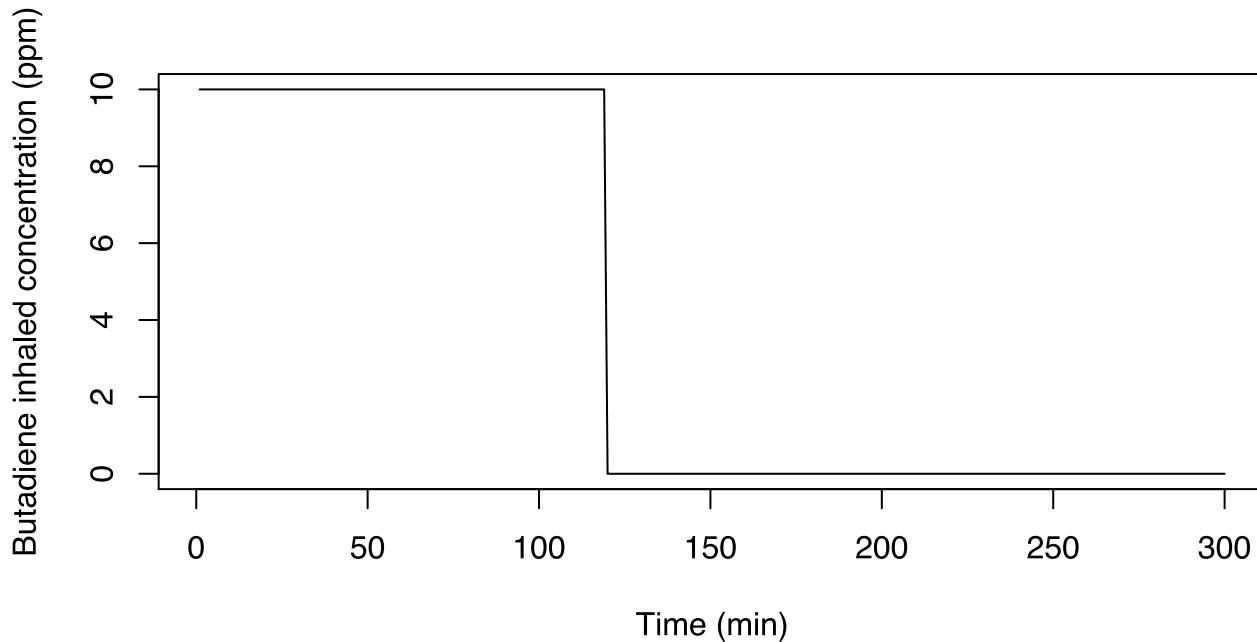
- Instead of simple PK model, in this exercise we want to apply the well developed PBPK model for 1,3 butadiene.
- First, reproduce the simulation result from the published paper*.

[*] Bois F.Y., Brochot C. (2016) **Modeling Pharmacokinetics**. In: Benfenati E. (eds) In Silico Methods for Predicting Drug Toxicity. Methods in Molecular Biology, vol 1425. Humana Press, New York, NY

Inputs

```
C_inh ← approxfun(x = c(0, 120), y=c(10,0), method = "constant", f = 0, rule = 2)
```

```
plot(C_inh(1:300), type="l", xlab = "Time (min)", ylab = "Butadiene inhaled concentration (ppm)")
```



Parameters and outputs

```
parameters <- c("BDM" = 73,                      # Body mass (kg)
               "Height" = 1.6,                    # Body height (m)
               "Age" = 40,                      # in years
               "Sex" = 1,                       # code 1 is male, 2 is female
               "Flow_pul" = 5,                   # Pulmonary ventilation rate (L/min)
               "Pct_Deadspace" = 0.7,          # Fraction of pulmonary deadspace
               "Vent_Perf" = 1.14,              # Ventilation over perfusion ratio
               "Pct_LBDM_wp" = 0.2,            # wp tissue as fraction of lean mass
               "Pct_Flow_fat" = 0.1,            # Fraction of cardiac output to fat
               "Pct_Flow_pp" = 0.35,             # ~ to pp
               "PC_art" = 2,                   # Blood/air partition coefficient
               "PC_fat" = 22,                  # Fat/blood ~
               "PC_wp" = 0.8,                  # wp/blood ~
               "PC_pp" = 0.8,                  # pp/blood ~
               "Kmetwp" = 0.25)                # Rate constant for metabolism
```

```
y <- c("Q_fat" = 0, # Quantity of butadiene in fat (mg)
      "Q_wp" = 0,   # ~ in well-perfused (mg)
      "Q_pp" = 0,   # ~ in poorly-perfused (mg)
      "Q_met" = 0)  # ~ metabolized (mg)
```

Model structure

```
# Define the model equations
bd.model <- function(t, y, parameters) {
  with (as.list(y), {
    with (as.list(parameters), {
      # Define constants

      # Calculate flow and volumes

      # Calculate the tissue, blood, and air

      # Differentials for quantities

      # The function bd.model must return at least the derivatives
      list(c(dQ_fat, dQ_wp, dQ_pp, dQ_met), # derivatives
           c("C_ven" = C_ven, "C_art" = C_art)) # extra outputs

    }) # end with parameters
  }) # end with y
} # end bd.model
```

Constants



Known constants

```
# Known constants
Height = 1.6                      # use to calculate fraction of body fat
Age = 40                            # use to calculate fraction of body fat
Sex = 1                             # use to calculate fraction of body fat
MW_bu = 54.0914                     # butadiene molecular weight (in grams)
```

Conversions from/to ppm

```
ppm_per_mM = 24450                  # ppm to mM under normal conditions
ppm_per_mg_per_l = ppm_per_mM / MW_bu
mg_per_l_per_ppm = 1 / ppm_per_mg_per_l
```

Flows and volumes



Air and blood flow

```
# Calculate Flow_alv from total pulmonary flow
Flow_alv = Flow_pul * (1 - Pct_Deadspace)

# Calculate total blood flow from Flow_alv and the V/P ratio
Flow_tot = Flow_alv / Vent_Perf

# Calculate fraction of body fat
Pct_BDM_fat = (1.2 * BDM / (Height * Height) - 10.8 * (2 - Sex) + 0.23 * Age - 5.4) * 0.01

# Calculate actual blood flows from total flow and percent flows
Flow_fat = Pct_Flow_fat * Flow_tot
Flow_pp = Pct_Flow_pp * Flow_tot
Flow_wp = Flow_tot * (1 - Pct_Flow_pp - Pct_Flow_fat)
```

Volumes

```
# Actual volumes, 10% of body mass (bones...) get no butadiene
Eff_V_fat = Pct_BDM_fat * BDM
Eff_V_wp = Pct_LBDM_wp * BDM * (1 - Pct_BDM_fat)
Eff_V_pp = 0.9 * BDM - Eff_V_fat - Eff_V_wp
```

Concentrations - tissues & blood



```
# Calculate the concentrations
C_fat = Q_fat / Eff_V_fat
C_wp = Q_wp / Eff_V_wp
C_pp = Q_pp / Eff_V_pp

# Venous blood concentrations at the organ exit
Cout_fat = C_fat / PC_fat
Cout_wp = C_wp / PC_wp
Cout_pp = C_pp / PC_pp

# Sum of Flow * Concentration for all compartments
dQ_ven = Flow_fat * Cout_fat + Flow_wp * Cout_wp + Flow_pp * Cout_pp
C_inh.current = C_inh(t)      # to avoid calling C_inh() twice

# Arterial blood concentration
# Convert input given in ppm to mg/l to match other units
C_art = (Flow_alv * C_inh.current * mg_per_l_per_ppm + dQ_ven) / (Flow_tot + Flow_alv / PC_art)

# Venous blood concentration (mg/L)
C_ven = dQ_ven / Flow_tot
```

Concentrations - air



```
# Alveolar air concentration (mg/L)
C_alv = C_art / PC_art

# Exhaled air concentration (ppm)
if (C_alv <= 0) {
  C_exh = 10E-30    # avoid round off errors
} else {
  C_exh = (1 - Pct_Deadspace) * C_alv * ppm_per_mg_per_l + Pct_Deadspace * C_inh.current
}
```

Differentials for quantities



```
# Quantity metabolized in liver (included in well-perfused)
dQmet_wp = Kmetwp * Q_wp

# Differentials for quantities
dQ_fat = Flow_fat * (C_art - Cout_fat)
dQ_wp = Flow_wp * (C_art - Cout_wp) - dQmet_wp
dQ_pp = Flow_pp * (C_art - Cout_pp)
dQ_met = dQmet_wp
```

Outputs

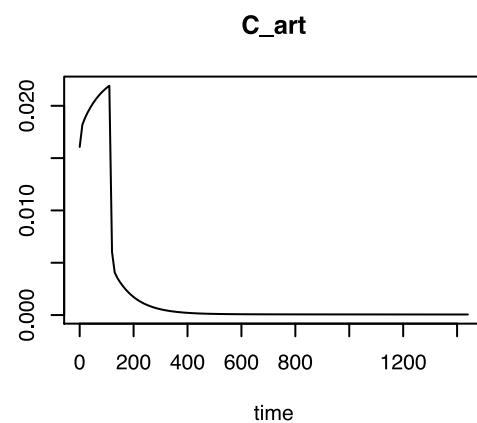
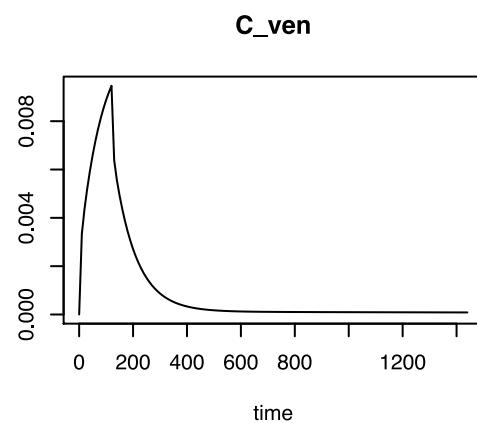
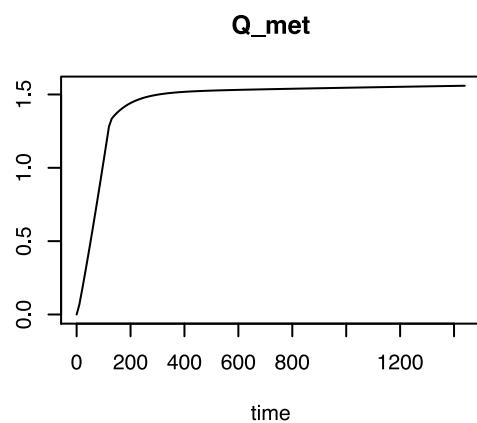
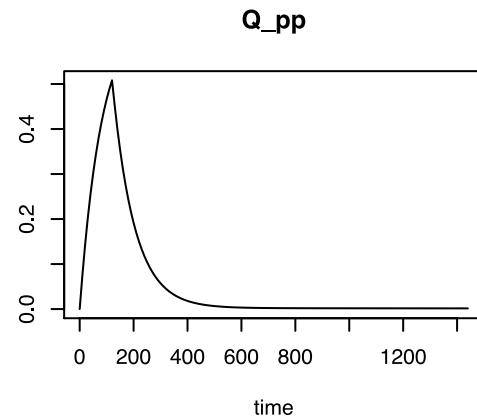
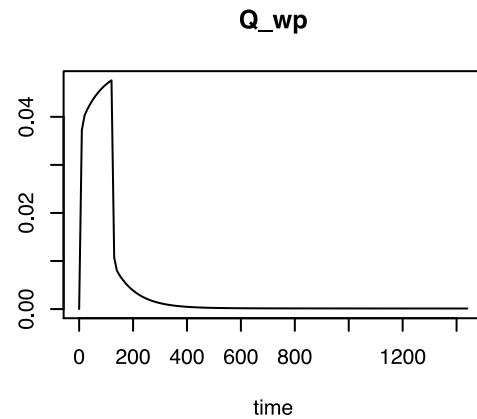
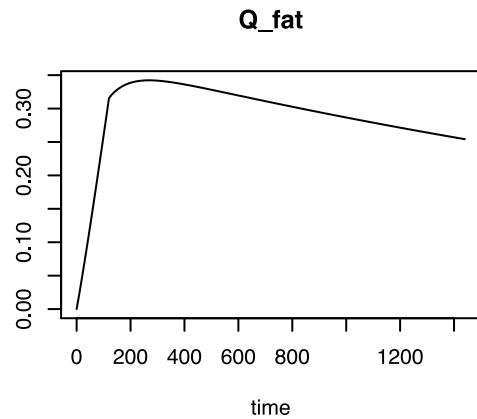


```
# Define the computation output times
t ← seq(from=0, to=1440, by=10)

# Solve ODE
library(deSolve)
out ← ode(times=t, func=bd.model,
          y=y, parms=parameters)
head(out)

##      time      Q_fat      Q_wp      Q_pp      Q_met      C_ven
## [1,]    0 0.00000000 0.00000000 0.00000000 0.00000000 0.000000000
## [2,]   10 0.02293618 0.03724892 0.07427798 0.06645654 0.003338318
## [3,]   20 0.04722954 0.04026245 0.14189019 0.16431358 0.004379098
## [4,]   30 0.07221315 0.04152080 0.20176415 0.26661643 0.005210310
## [5,]   40 0.09777386 0.04256838 0.25471138 0.37175647 0.005941936
## [6,]   50 0.12383371 0.04349410 0.30153555 0.47935418 0.006589697
##      C_art
## [1,] 0.01606403
## [2,] 0.01819035
## [3,] 0.01885327
## [4,] 0.01938270
## [5,] 0.01984870
```

```
plot(out)
```



Hands on Exercise



Task 5: Apply the PBPK model

- Based on the OSHA permissible exposure limit (PEL), the time-weighted average is 1 ppm. If a worker is under the exposure for a long time, what is the estimated blood concentration.