JSS: Abstract: list all packages + MCSIM (i.e., statistical software) used

# How to build a new pbtk model from .model files:

#First, create .model file and type in command prompt (with MCSIM/mod in path): mod -R <name>.model <name>.c

#This generates the <name>.c and <name>\_inits.R files.

#In order to avoid duplicate names, function names in these newly created files must be changed.

#The changes (adding \_<name>, an arbitrary choice) in the initialization file include: initparms\_<name>, getParms\_<name> (calling function from .c within initparms argument), Outputs\_<name>, and initState\_<name>.

#The changes in the .c file include: initmod\_<name>, initforc\_<name> (or delete function), getParms\_<name>, derivs\_<name>, jac\_<name>, event\_<name>, and root\_<name>.

#These function names then must also be changed in the corresponding solve\_pbtk function, specifically the state and parameter initilaizers as well as the function calls in the function ode.

#Parameter names passed to initparams must match those in .model file.

#what do we do with the inits .c file?

Need table of functions generated by MCsim – which must be renamed, which must be commented out

Explicitly state philosophy of Do things once

need big font table in paper explaining suffix c and f

include default PBTK multiplication

need clear explaination of rtosolvermap and get\_parms

explicitly state we do the bodyweight scaling in the model c code

need big font table in paper explaining suffix c and f

c means scaled to body weight, for a flow it is usually scaled to 3/4 BW ([Qcardiacc] = L/h/kg^(3/4)) for a volume it is usually scaled to BW ([Vliverc] = L/kg

f means fraction of a total

include default PBTK multiplication

need clear explaination of rtosolvermap and get\_parms

explicitly state we do the bodyweight scaling in the model c code

Developing Generic Toxicokinetic Models with R Package “httk” for Enhanced Reporting Accuracy and Statistical Evaluation

Target: Journal of Pharmacokinetics and Pharmacodynamics

Target Submission: 9/2021

John Wambaugh, Greg Honda, Mark Sfeir, Matthew Linakis, Michael Devito, Tom Purucker, Sarah Davidson, Caroline Ring

plus any interested contributor

# Abstract

To address greater numbers of chemicals *in vitro*, high throughput toxicokinetic (HTTK) data have been collected to characterize absorption, distribution, metabolism, and excretion (ADME). HTTK methods have been used by the pharmaceutical industry to determine range of efficacious doses and to prospectively evaluate success of planned clinical trials. For non-therapeutic compounds where clinical trials are unlikely, HTTK provides a predicted human dose context for bioactive *in vitro* concentrations from *in vitro* high throughput screening (that is*, in vitro-in vivo* extrapolation, or IVIVE). The R package “httk” provides open source data and models for evaluation and use by the broader scientific community. Here we discuss how new models may be created and interlaced with the “httk” software environment, so that previously evaluated tools and data may be incorporated alongside new approaches. Ultimately the trick is not building a new model, but rather evaluating that new model. New generic TK models can and should be statistically evaluated across the growing library of chemicals with HTTK data. Each new addition to “httk” can include an evaluation of model predictions relative to *in vivo* data where available. Decision makers can then consider whether the generic model predictions based on *in vitro* data can be extrapolated for a chemical without *in vivo* data

# Introduction

Schmolke et al. [1] wrote that “empirical approaches are often too limited to inform policy and decision making” and therefore “decision making requires models”. Decisions about the risk posed to the public health involves models for toxicity, exposure, and the dose-response relationship [2]. Toxicokinetics informs the dose-response relationship [3] by describing the absorption, distribution, metabolism, and excretion (ADME) of a chemical dose. Mathematical models for toxicokinetics allow prediction of the relationship between external doses to the body and internal tissue concentrations [4]. Within the pharmaceutical industry physiologically-based toxicokinetic (PBTK) models are widely used to better understand drug disposition and provide the backbone for insights into toxicodynamic effects such as drug-drug interactions[5,6]. To evaluate confidence in particular TK model one can compare the predictions to in vivo measured data[7]. The correspondence between predictions and observations allows bias and uncertainty to be estimated and decision makers may then consider using model to extrapolate to other situations (dose, route, physiology) where data may be unavailable [8]. However, despite the deterministic nature of scientific computing (even including pseudo-random number generation) it is often difficult to reproduce computational results [9]. In non-pharmaceutical regulatory settings, of data availability (that is, no clinical trials), model reproducibility, and statistical evaluation are all barriers to the incorporation of PBTK models[10].

There are tens of thousands of non-pharmaceutical chemicals in commerce and the environment for which it is desirable to assess risk posed to public health[11]. Since there are few reasons for obtaining data on non-therapeutical chemicals in humans, and limited resources for obtaining data in animals, there is in general a lack of the sort of in vivo data used to build and evaluate many PBTK models for pharmaceuticals[12,13]. In these cases, generic PBPK models offer an alternative path[14-19]. There are many TK processes that are important for only some classes of chemicals and so, we expect a generic model to have larger uncertainty than a "bespoke" chemical-specific model [20]. However, because the same model is used across chemicals, generic models offer greater confidence in model implementation and reproducibility. An advantage that is distinct to generic models is that we can evaluate the model predictions in the absence of *in vivo* data for a specific chemical. Model predictions may be evaluated using as much *in vivo* data as possible for other chemicals. This allows estimation of bias and uncertainty and subsequent correlation of residuals with chemical-specific properties. [21,22] Decision makers can then consider whether the generic model can be extrapolated for a chemical without *in vivo* data.

One approach to generic PBPK modeling is “high throughput toxicokinetics (HTTK)”, in which a generic toxicokinetic modeling is designed to be parameterized with chemical-specific in vitro toxicokinetic measurements. Over the past decade HTTK data have become publicly available for more than a thousand chemicals. Concurrently, public repositories of in vivo toxicokinetic (that is, chemical concentration in various tissues as a function of dose and time) observations have become available to allow statistical analysis of the performance of HTTK. That is, the comparison in HTTK predictions to *in vivo* observations allows for empirical estimates of HTTK uncertainty. Any approximations, omissions, or mistakes should work to increase the estimated uncertainty when evaluated systematically across chemicals.

Models for decision making must be fully evaluated[1,23-25]. Clark et al. [10] identified six areas to be assessed for any PBTK model: 1) purpose, 2) structure and biology, 3) mathematical descriptions, 4) computer implementation, 5) parameter values and model fitness, and 6) any specialized applications. The use of generic models in a standardized environment helps address the first four points. Issues of parameter appropriateness and model fitness require further evaluation using tools such as statistical regression and machine learning. [26] The R statistical programming environment provides thousands of approaches for this evaluation, while databases like CvTdb [27] provide observations to which predictions may be compared. To support the development, evaluation, and use of HTTK in chemical risk decision making the U.S. Environmental Protection Agency Office of Research and Development has made the tool “httk” publicly available for the free, open-source statistical analysis platform R. “httk” includes generic TK models that can be parameterized from *in vitro* data and physico-chemical properties. The open-source nature of “httk” is intended to allow other interested parties to verify and evaluate the included modules, hopefully reducing the likelihood of programming errors. “httk” is constructed with a modular structure intended to allow others to build upon the data, models, and functions provided.

Pearce et al. [28] described the original release of “httk” in detail, while Ring et al. [29] focused on the “httk-pop” human variability simulator. Recently, Breen et al. (in preparation) [30] has reviewed how httk may be used to inform *in vitro-in vivo* extrapolation (IVIVE) for chemical risk assessment. “httk” uses the built-in R documentation functions, which currently provide hundreds of pages of details at the level of individual functions and data sets. Here we discuss the bigger picture of how new models may be created and interlaced with the HTTK environment, so that previously evaluated tools and data may be incorporated alongside new approaches. Ultimately the trick is not building a new model, but rather evaluating that new model. New generic TK models can and should be statistically evaluated across the growing library of chemicals with HTTK data. Each new addition to HTTK should include an evaluation of model predictions relative to *in vivo* data where available. Based on this evaluation, it may be possible to estimate of how generalizable the new model (or revision) is expected to be. "The first principle is that you must not fool yourself—and you are the easiest person to fool.” [31]

# Methods

Releases of “httk” are versioned with three numbers indicating, in order, “Major Releases”, “Minor Releases”, and “Patches”. The current version of is 2.0.3. There have been two major releases of “httk”, with version 2 including the provisions for adding models described here. Minor releases include new models or data sets. Both major and minor releases always accompany peer-reviewed publications. Patches include relatively minor alterations of the functions and data to address problems identified by the scientific community. Following standard R package conventions, changes should be described in the NEWS file. Version control software is used by EPA, and a public GitHub repository is maintained at:

<https://github.com/USEPA/CompTox-ExpoCast-httk>

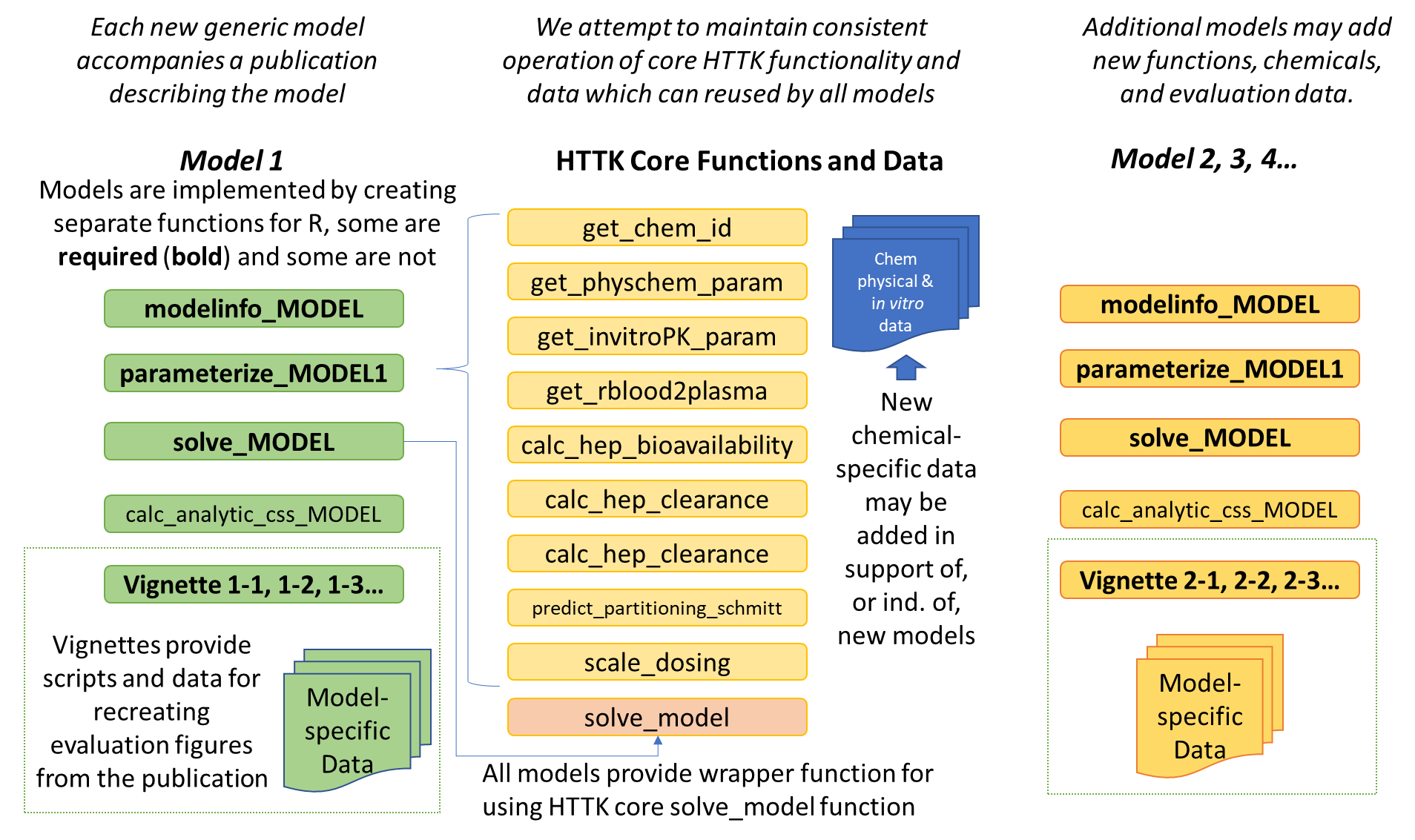
Unreleased versions of the package are indicated with a major version number of “99”. A version stamped, distributable tarball (with extension “.tar.gz”) can be created from the command shell using:

R CMD build [HTTKPATH]

where “HTTKPATH” is the location of the new version of HTTK. If you are testing/developing a new version locally and do not need to distribute, the following will directly install the new version:

R CMD INSTALL [HTTKPATH]

The capitalization difference between “build” and “INSTALL” must be observed for reasons presumedly known to the creators of R.

“

Figure

## Generic Functions for TK Models

Throughout this document we distinguish the general science of HTTK from the specific implementation of the R package “httk” using capitalization and quotation marks.

To ensure accurate description and evaluation of the models, the developers of “httk” have attempted to make use of good computer programming and modeling practices. The httk R package is designed as a “software ecosystem” [32] that allows reuse[33] of data and calculations (as functions) that have been documented and peer-reviewed in the scientific literature. Like most R packages[34], HTTK is constructed of modular functions [35], such that new models and analyses can be constructed by new developers from what has come before. Modularity also provides “scoping”, that is, helps to limit the ramifications of new changes[36]. It is hoped that the additions of new contributors do not break something that came before, although evaluation methods allow for such problems when they inevitably occur. Related to this, reuse of functions allows developers to “touch” the code in the fewest places possible – if the way something is done must be changed, hopefully it only needs to be changed in one place. Reynolds, Acock [37] wrote that, in the context of plant and ecosystem models, “modularity and genericness open models to contributions from many authors, facilitate the comparison of alternative hypotheses, and extend the life and utility of simulation models.” This is the same intent for which R package “httk” has been developed for high throughput toxicokinetics.

Each of these functions should work for any generic model added to the HTTK package. Since different models require differing parameters, the number of chemicals for which a model may be used varies. The get\_cheminfo() function serves to identify which chemicals work for which model. (the optional arguments for get\_cheminfo() also allow retrieval of data from the HTTK package).

Table **Reusable Functions for Interactive with Generic TK Models**

|  |  |
| --- | --- |
| get\_cheminfo | Interface for users to interact with internal HTTK data table of chemical-specific information. Response depends on model specified – only chemicals with data sufficient for running a given model are returned. |
| solve\_model | Generic numerical ODE solver function for using deSolve to simulate a concentration vs. time response. Often used by wrapper functions (for example, solve\_steadystate, solve\_pbtk). |
| calc\_mc\_tk | Function for running a Monte Carlo simulation with a TK model. Essentially runs solve\_model for a range of parameter sets. |
| calc\_css | This is a numerical approach to determine the steady-state concentration (Css) as a function of repeated dosing. Dosing can be changed to match different conditions (default is three times a day). Also returns the days needed to reach steady-state. |
| calc\_analytic\_css | This function calls the model-specific analytical solution when that is available. |
| calc\_mc\_css | Function for running a Monte Carlo calculation of Css – typically uses the analytic solution as that is much faster than the numerical approach. |
| calc\_mc\_oral\_equiv | This is the “reverse dosimetry” IVIVE function that uses the quantiles of the Css from calc\_mc\_css to estimate a daily oral dose rate that would cause a given plasma concentration steady-state. |
| calc\_mc\_gas\_equiv | This is a new “reverse dosimetry” IVIVE function from Breen et al. (in preparation) for daily inhalation rate that would cause a given plasma concentration at steady-state. |

## Parameter Tables

Most of the data that is loaded when you use “library(httk)” is provided via the following tables:

Table **Core HTTK Data Sets**

|  |  |
| --- | --- |
| chem.physical\_and\_invitro.data | Chemical-specific values for physico-chemical properties and in vitro measurements. |
| chem.invivo.PK.data | Chemical concentration vs. time data (CvTdb) for evaluating model predictions. [27] |
| physiology.data | Data describing species-specific physiology. |
| tissue.data | Tissue composition data for Schmitt’s method of partition coefficient prediction. |
| Tables.Rdata.stamp | Time stamp identifying when the other tables were created. |

These tables are stored in the Tables.RData file in the httk/data directory. They evolve over time and a specific to each version of HTTK. The string “Tables.Rdata.stamp” is included in Tables.RData and is the date on which those tables were created. A separate, currently not public Git repository “HTTKDataTables” maintains a script and source file for generating the Tables.RData file.

We aim to avoid hard coding parameter values into functions, so parameters should instead be stored in data tables. Data should be documented, including column descriptions and units, using roxygen in the file httk/R/data.r – this will cause user documentation to be generated for each data object.

## Checking Units

Rules of PBPK Modeling (Anderson and Clewell):

1. Check Your Units
2. Check Your Units
3. Check Mass Balance

In some cases, units are not a problem since the relevant parameters are fractions/ratios. However, we always need to be careful to translate:

* intrinsic hepatic clearance units should be µL/min/106 hepatocytes
* volume of distribution units should be L/kg body weight
* KM should probably be µM
* Vmax depends on the process

## Chemical-Specific Parameters

All chemical-specific information is stored in the table chem.physical\_and\_invitro.data. That table can be altered using the add\_chemtable() function. All httk functions retrieve data from chem.physical\_and\_invitro.data.

The functions get\_invitroPK\_param() and get\_physchem\_param() should be used to access chem.physical\_and\_invitro.data because they handle several contingencies (species problems, errors) in a standardized manner.

The function get\_cheminfo() by default returns a list of CASRN in chem.physical\_and\_invitro.data that have all the necessary parameters for a give model (defaults to “3compartmentss”) as defined by that models modelinfo.R file.

While most models should provide their own function for generating chemical-specific model parameters (parameterize\_MODELNAME.R) they may rely on the function parameterize\_pbtk which can be customized with respect to which tissues are lumped into various compartments. For example, both parameterize\_3comp makes use of parameterize\_pbtk.

Most of the in vitro measured data are for human tissues. To use these values with non-human physiologies we typically include an argument “default.to.human” that, if true, uses the human values.

There are several generic functions for calculating key aspects of ADME (absorption, distribution, metabolism, and excretion) available for use by any model.

Table  **Reusable Model Parametrization Functions**

|  |  |
| --- | --- |
| get\_invitroPK\_param | Retrieve a chemical- and species-specific in vitro-measured parameter from chem.physical\_and\_invitro.data table |
| get\_physchem\_param | Retrieve a chemical-specific physico-chemical property from chem.physical\_and\_invitro.data table |
| parameterize\_pbtk | Generate a chemical- and species-specific set of model parameters, including tissue:plasma partition coefficients and organ volumes and flows (from table physiology.data) for an arbitrary tissue lumping scheme (tissues must be described in table tissue.data) |
| calc\_hep\_clearance | Calculate the hepatic clearance based on the in vitro clearance for various models. |
| calc\_hep\_bioavailability | Estimate the first pass hepatic clearance for models without explicit blood flow from the gut to the liver. Requires both the intrinsic clearance and the blood:plasma ratio. |
| available\_rblood2plasma | Retrieve or estimated the chemical-specific blood:plasma ratio. |
| parameterize\_schmitt | Use the calibrated Schmitt’s method for predicting tissue:free fraction in plasma partition coefficients for all the tissues in the tissue.data table. |
| lumptissues | Lump the model-relevant tissues (as specified by the appropriate modelinfo.R file) into the appropriate tissue parameters (volumes, flows, scaled partition coefficients). |

### Hepatic Clearance

Hepatic clearance (Clint) is measured in vitro and then used to by the function calc\_hep\_clearance()calculate a whole-liver hepatic clearance based on various scaling models described by Ito, Houston [38], most often the well stirred model.

### Fraction Unbound in Plasma

Fraction unbound in plasma (fup) is an incredibly important parameter because it is used for many other predictors. Right now get\_invitroPK\_param() is the only major function involved in retrieving fup. In some instances fup is a triple value, with a median, lower 95th percentile, and upper 95th percentile value separated by commas. The Monte Carlo sampler will use this information to appropriately propagate uncertainty when available.

### Blood to Plasma Ratio

Because most of the PBPK models in HTTK use tissue:plasma partition coefficients, the blood to plasma ratio is used to scale the fraction of chemical available for partitioning and other processes (for example metabolism, glomerular filtration) throughput the body. The function available\_rblood2plasma() works through a series of options, trying to first obtain a species-specific measured value, then defaulting to human measured, then trying to use Schmitt’s method to predict the red blood cell to plasma partition coefficient (calc\_rblood2plasma()). When all else fails (because insufficient chemical-specific phys-chem info is available) the average measured value is used.

### Oral Absorption

Upcoming Honda et al. paper will describe in vitro measured Caco2 permeability data and a QSAR developed for predicting whether or not a chemical is well-absorbed based on Caco2, phys-chem, and a rough estimate of gut metabolism.

### Partition Coefficients

HTTK uses a customized variant of the Schmitt method [39] for making chemical specific partitioning into tissues (equilibrium tissue:free plasma concentration ratios). The tissues can be customized if certain descriptors of their make-up (for example, fraction lipid) are provided. For certain tissues empirical calibrations based upon literature data for partition coefficients are available. Schmitt’s method makes use of the fraction unbound of the chemical in plasma, as well as physico-chemical parameters including hydrophobicity (octanal:water partition coefficient or “log P”) and ionization equilibria (pKa). The method used in HTTK was described in Pearce et al. (2017) Pearce et al. [40].

Any model in HTTK may make use of the partition coefficient prediction code via predict\_partitioning\_schmitt(), which predicts a partition coefficient for each tissue in the tissue.data table. The function lumptissues() can then be used to lump the tissues from tissue.data that are relevant to the model (as specified by the .modelinfo file) into the appropriate lumped tissues for the model.

## Model Evaluation Data

We include with HTTK the CvTdb and parameters estimated from that database, to allow evaluation of the predictions of models. Need to describe all this briefly.

## Integrating New Models

We use the MCSim [41] "mod" function to convert MCSim models into C code that can be compiled. We then run these compiled models with R package deSolve [42].

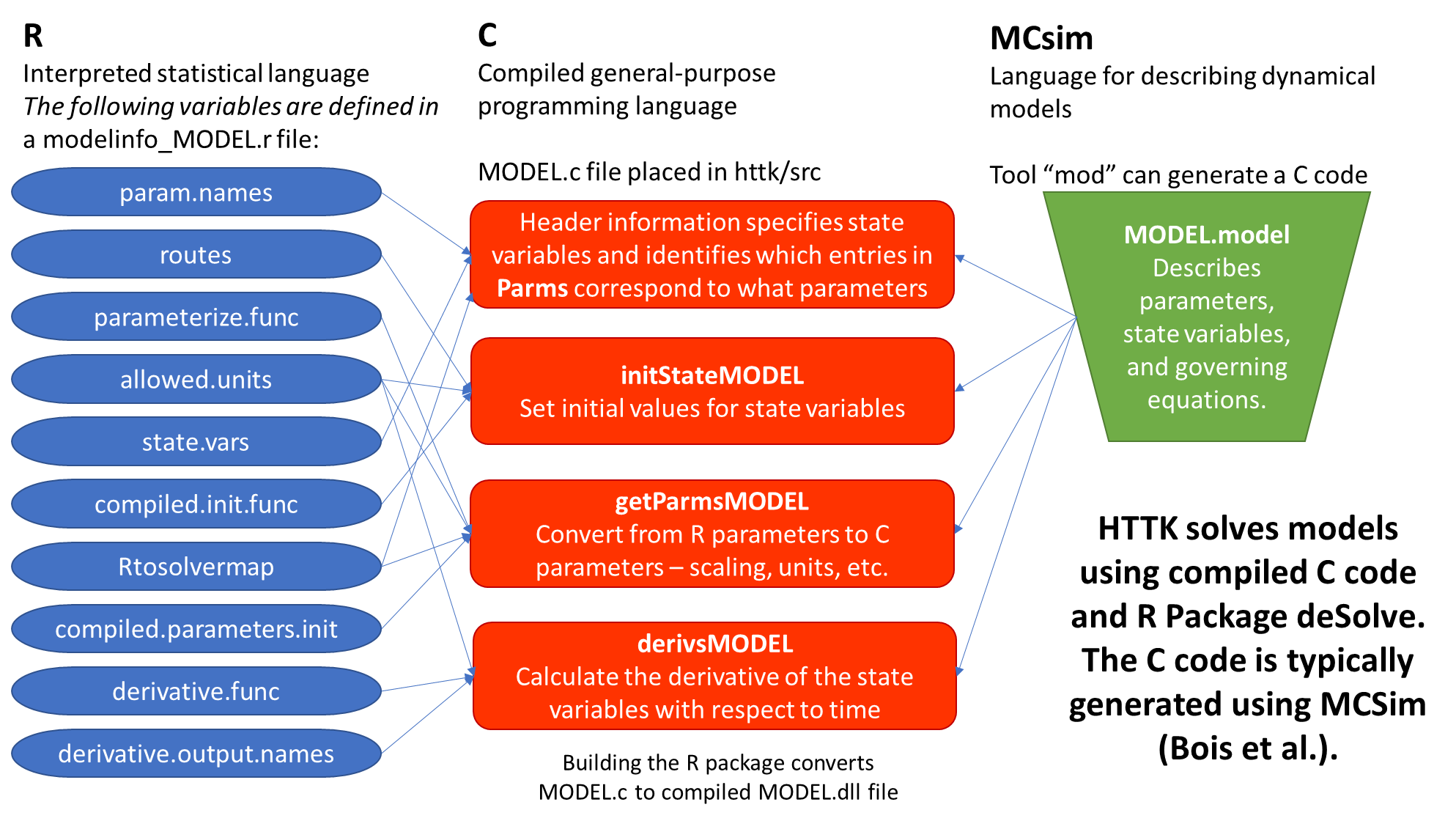


Figure HTTK solves models using compiled C code and R Package deSolve. The C code is typically generated using MCSim (Bois et al.).

### MCSim

MCSim is a model description language that was originally designed for applying Markov Chain Monte Carlo to dynamical models. We make use of the tools in MCSim that can translate a description of an ordinary differential equation (ODE) model into C code that evaluates the derivative of that model with respect to time.

### deSolve

deSolve is an R package for solving a system of differential equations. It runs much faster if a compilable language (like C) is used to provide the derivative of the system.

### How to Link MCSim and HTTK

How to build a new pbtk model from .model files:

First, create .model file and type in command prompt (with MCSIM/mod in path and in directory containing .model file): mod -R <name>.model <name>.c

This generates the <name>.c and <name>\_inits.R files.

To avoid duplicate names, function names in these newly created files must be changed.

The changes (adding \_<name>, an arbitrary choice) in the initialization file include: initparms\_<name>, getParms\_<name> (calling function from .c within initparms argument), Outputs\_<name>, and initState\_<name>.

The changes in the .c file include: initmod\_<name>, initforc\_<name> (or delete function), getParms\_<name>, derivs\_<name>, jac\_<name>, event\_<name>, and root\_<name>.

These function names then must also be changed in the corresponding solve\_pbtk function, specifically the state and parameter initializers as well as the function calls in the function ode.

Mod.exe creates other functions that we do not currently use and should be deleted from the .C file.

Parameter names passed to initparams must match those in .model file.

### Model Description

Starting with HTTK version 2.0.0 functions in HTTK are supposed to identify and interact with models based on how they are described in the file “modelinfo\_[MODELNAME].R”.

#### Model “aware” functions:

get\_cheminfo() – lists the chemicals that have data sufficient to run model X

calc\_analytic\_css() – if a analytic expression for Css is available for model X, this will calculate steady-state plasma concentration

solve\_model() Makes call to deSolve using C code for the derivative.

convert\_httkpop() Function for using httk-pop predicted variables to parameterize your model for Monte Carlo simulation [NOT YET COMPLETE]

#### What You Need:

Modelinfo\_[MODEL].R – This code defines a number of global variables used by functions(see table)

parameterize\_[MODEL].R – This code uses in vitro and phys-chem info to parameterize your model. In some cases (for example, parameterize\_schmitt) this is the only functionality available for

Table  **modelinfo\_[MODEL].R file Information**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Model Information** | **Description** | **Css** | **Monte Carlo** | **Httk-pop MC** | **Dynamic** | **List Chemicals** |
| analytic.css.func | Analytic expression for steady-state plasma concentration. | X | X | X |  |  |
| parameterize.func | The is the R function for generating model parameter | X | X | X | X |  |
| param.names | These are all the parameters returned by the R model parameterization function. Some of these parameters are not directly used to solve the model, but describe how other parameters were calculated |  |  |  |  |  |
| Rtosolvermap | This subset of R parameters are needed to initially parametrize the compiled code for the solver: (must match ORDER under "parameters" in C code) |  |  |  | X |  |
| compiled.parameters.init | This function translates the R model parameters into the compiled model parameters |  |  |  | X |  |
| compiled.param.names | This is the ORDERED full list of parameters used by the compiled code to calculate the derivative of the system of equations describing the model |  |  |  | X |  |
| compiled.init.func | This function initializes the state vector for the compiled model |  |  |  | X |  |
| derivative.func | This is the function that calculates the derviative of the model as a function  # of time, state, and parameters |  |  |  | X |  |
| derivative.output.names | This is the ORDERED list of variables returned by the derivative function |  |  |  | X |  |
| conc.units | Allowable units (and whether they are for amounts or concentration) |  |  |  | X |  |
| amount.units | Allowable units (and whether they are for amounts or concentration) |  |  |  | X |  |
| dosing.params | These parameters specific the exposure scenario simulated by the mode |  |  |  | X |  |
| routes | Vector of character names for each exposure route |  |  |  | X |  |
| dose.variable | Named list of which compartment gets the dose for each route. |  |  |  | X |  |
| dose.type | Can take the values "add" to add dose C1 <- C1 + dose,"replace" to change the value C1 <- dose or "multiply" to change the value to C1 <- C1\*dose (deSolve eventdata) |  |  |  | X |  |
| amount.compartments | Tissue (or other) compartments that are always modeled and tracked as amounts |  |  |  | X |  |
| Other.compartments | Tissue (or other) compartments that are modeled as amounts but may be reported as amounts or concentrations |  |  |  |  |  |
| required.params | Parameters needed to make a prediction |  |  |  |  | X |
| exclude.fup.zero | Do we ignore the Fups where the value was below the limit of detection? |  |  |  |  | X |

### How to Link to HTTK-Pop

HTTK-Pop is the built-in human variability Monte Carlo Sampler [29]. To make efficient use of it you need to have a function to calculate the analytical steady-state solution to your model.

The function httkpop\_generate creates a table with one row per individual and various columns corresponding to biometric. The biometrics are consistent with the NHANES cohort and any arguments given to httkpop\_generate (for example, specifying a sex or age). These biometrics are converted to typical PBTK parameters by the function httkpop\_biotophys\_default. They can also be used to create model-specific parameters if the function convert\_httkpop\_MODEL exists for the given model.

HTTK-pop does not have to be used (httkpop=FALSE) and should not be used for non-human populations. It is possible to use the default Monte Carlo distributions to provide alternative Monte Carlo simulation.

### What you’ll likely want for additional functionality:

MCSim code describing the model -> C code generated from MCSim

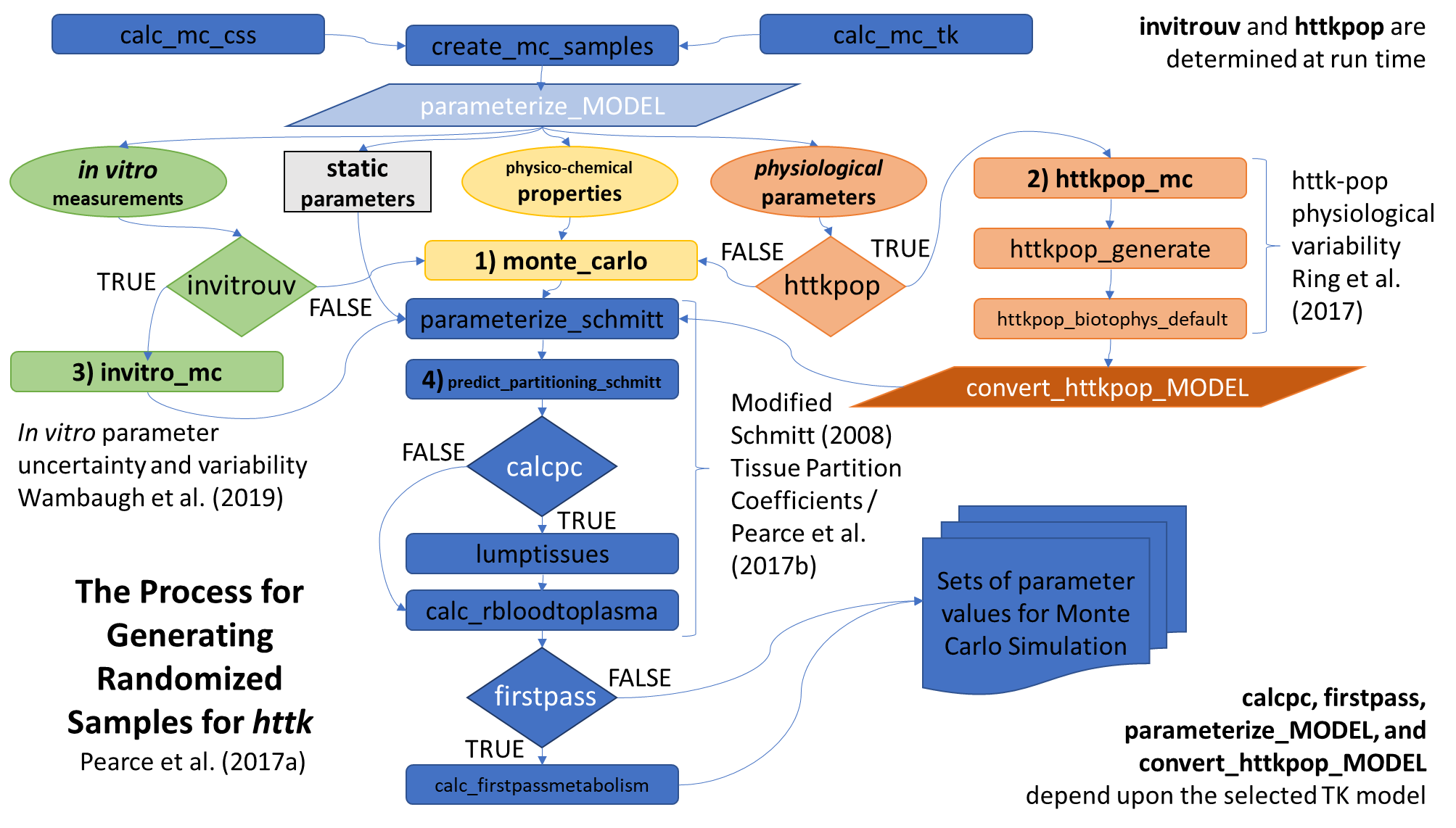
Analytic function for the steady-state concentration (needed for effective Monte Carlo)

Wrapper function for calling solve\_model (for example, solve\_pbpk)

## Monte Carlo

Within httk we propagate both uncertainty and variability [8] with Monte Carlo (MC) simulation. monte\_carlo() is the primary function, and creates a table of samples with a column for each parameter and a row for each “draw” of the MC simulation. Two distributions can be used with monte\_carlo(): a normal distribution characterized by a mean and coefficient of variation (truncated at zero so as to always be positive) and a censored normal distribution, characterized by a mean, coefficient of variation, and a censoring limit (such as a limit of detection) below which all values are equally likely.

Certain parameters may be varied by other distributions if they are human physiological parameters (varied by httkpop\_mc()[29]) or in vitro measurements (invitro\_mc()[43]).



Figure

# Results

To date we have taken the approach of adding new models to the package rather than creating a single, super model. Each of these models should have been evaluated against measured data. We should maintain the ability to calculate a root mean squared error (RMSE) for each model, and then recalculate it every time we are done revising the package to make sure that a change made for one model affects the other models in the intended way (often, not at all).

A goal of the “httk” project is to add new TK models exploring, for example, different exposure scenarios, varying physiological processes, and non-terrestrial species. We hope to maximize the reuse of in vitro measurements of TK determinants, physico-chemical property predictions, and functions making calculations based upon those values. By conserving data and functions across models we increase the verification of the data and functions and conversely enhance the transparency and reproducibility of new models.

To be included in the R package a new model must be peer-reviewed (for example, published in an appropriate scientific journal) and evaluated. The evaluation is often the most difficult part, as it requires compilation of data permitting evaluation, for example in vivo observations relating to the scenario covered by the model for some subset of chemicals for which the model may be used. Appropriate statistics should be presented in the peer review, such as RMSE and R2. There is no prescribed threshold of performance for inclusion, though a case must be made that predictions made by the model do, in some situations, provide value. Metrics such as RMSE and R2 also serve as benchmarks – as the shared calculations and functions of the package are revised and new data are obtained it is hoped that a general trend toward more accurate predictions might be observed.

The appropriate files must be added to a new version of the R package, including the model code (either in C or in a MCsim .model file) and any new or modified functions or data (including evaluation data). We strongly prefer that a vignette be added to the package illustrating use of the new model to calculate results and generating figures from the peer-reviewed publication. The ability to recreate exactly the figures from the peer review using a vignette provides an important measure of quality control. Finally, the “domain of applicability” must be noted – for what type of chemicals (for example, non-volatile or non-PFAS) may the model be used with confidence? A new version of the R package (a minor release) can be included as supplemental information if allowed by a journal.

## Case Study Example Model

Possibly the dermal model?

delete functions: lagvalue, calcdelay

1. compile .model file to .c file
2. updates in the .c file prior
3. compile .c file to .o file by re-building the R package

function names need to be changed manually

number of parameters

## Documenting Results with Vignettes

“httk” makes use of the “vignette” functionality in R to provide working code examples that can generate figures and perform other analyses using the models. Vignettes can be thought of as “long-form documentation” or extended examples potentially combining many functions from the package. We have aimed to provide vignettes for creating most major figures for each HTTK-related paper. Within a vignette it is especially important to include model assessment figures that calculate things like RMSE because we want to see how that evolves as we (hopefully) improve the model. If a new addition to the package causes problems elsewhere, the vignettes provide a mechanism for identifying the existence of the problem, similar to the scripts in the “/tests” directory.

When a vignette is added to the package it is necessary to distribute all the necessary data with the package. Save all the relevant tables to an .RData file and place that file in the httk httk/data directory. As with any other data, the new tables and other objects should be documented using roxygen in httk/R/data.r

The default figure size for vignettes is 3" by 3". This is quite small, but is necessary to keep the package from getting too large (CRAN is pretty strict about the total archive being under 5mb). If you are using ggplot2 to make your figures, then in order to make your figures better (not best, but better) at that small size, just comment out all text size instructions. Ggplot2 does a decent job of figuring out the rest.

### What Vignettes Are Available?

You can display all the available vignettes using:

vignette(package="httk")

You can display a specific vignette by using its name:

vignette("Frank2018")

**Note, the names of the vignettes that the user has to type for the vignette command are the same as the names of the files in the httk/vignettes directory. Don't make them too long**.

## Model Evaluation

I am strongly tempted to add a function that will automatically calculate quantities like RMSE using the CvT database, with provisions for the user to subset the chemicals involved. We could also add provisions for alternative evaluation data. Then we can ask people to use that function in their analysis.

Need a paragraph or two of my usual stump speech covering:

* The point of generic models
* Review Cvt database.
* Prediction vs. observed plots.
* RMSE, RPE over R2
* We need to make sure we don't break anything.

To evaluate a chemical-specific TK model for “chemical x” you can compare the predictions to in vivo measured data. However, we do not typically have TK data. We can parameterize a generic TK model, and evaluate that model for as many chemicals as we do have data. We do expect larger uncertainty, but also greater confidence in model implementation. Estimate bias and uncertainty, and try to correlate with chemical-specific properties Can consider using model to extrapolate to other situations (chemicals without in vivo data) [22]

# Discussion

# References

References

1. Schmolke A, Thorbek P, DeAngelis DL, Grimm V (2010) Ecological models supporting environmental decision making: a strategy for the future. Trends in ecology & evolution 25 (8):479-486

2. National Research Council (1983) Risk Assessment in the Federal Government: Managing the Process. In. National Academies Press, Washington (DC). doi:10.17226/317

3. Coecke S, Pelkonen O, Leite SB, Bernauer U, Bessems JG, Bois FY, Gundert-Remy U, Loizou G, Testai E, Zaldívar J-M (2013) Toxicokinetics as a key to the integrated toxicity risk assessment based primarily on non-animal approaches. Toxicology in Vitro 27 (5):1570-1577

4. O'Flaherty EJ (1981) Toxicants and drugs: kinetics and dynamics. John Wiley & Sons,

5. Sager JE, Yu J, Ragueneau-Majlessi I, Isoherranen N (2015) Physiologically based pharmacokinetic (PBPK) modeling and simulation approaches: a systematic review of published models, applications, and model verification. Drug Metabolism and Disposition 43 (11):1823-1837

6. Zhao P, Zhang L, Grillo J, Liu Q, Bullock J, Moon Y, Song P, Brar S, Madabushi R, Wu T (2011) Applications of physiologically based pharmacokinetic (PBPK) modeling and simulation during regulatory review. Clinical Pharmacology & Therapeutics 89 (2):259-267

7. World Health Organization (2010) Characterization and application of physiologically based pharmacokinetic models in risk assessment. World Health Organization, International Programme on Chemical Safety, Geneva, Switzerland

8. Barton HA, Chiu WA, Setzer RW, Andersen ME, Bailer AJ, Bois FY, DeWoskin RS, Hays S, Johanson G, Jones N (2007) Characterizing uncertainty and variability in physiologically based pharmacokinetic models: state of the science and needs for research and implementation. Toxicological Sciences 99 (2):395-402

9. Piccolo SR, Frampton MB (2016) Tools and techniques for computational reproducibility. GigaScience 5 (1):s13742-13016-10135-13744

10. Clark LH, Setzer RW, Barton HA (2004) Framework for evaluation of physiologically‐based pharmacokinetic models for use in safety or risk assessment. Risk Analysis 24 (6):1697-1717

11. Judson R, Richard A, Dix DJ, Houck K, Martin M, Kavlock R, Dellarco V, Henry T, Holderman T, Sayre P (2009) The toxicity data landscape for environmental chemicals. Environmental Health Perspectives 117 (5):685

12. Wetmore BA, Wambaugh JF, Ferguson SS, Sochaski MA, Rotroff DM, Freeman K, Clewell HJ, 3rd, Dix DJ, Andersen ME, Houck KA, Allen B, Judson RS, Singh R, Kavlock RJ, Richard AM, Thomas RS (2012) Integration of dosimetry, exposure, and high-throughput screening data in chemical toxicity assessment. Toxicological Sciences 125 (1):157-174. doi:10.1093/toxsci/kfr254

13. Bell SM, Chang X, Wambaugh JF, Allen DG, Bartels M, Brouwer KLR, Casey WM, Choksi N, Ferguson SS, Fraczkiewicz G, Jarabek AM, Ke A, Lumen A, Lynn SG, Paini A, Price PS, Ring C, Simon TW, Sipes NS, Sprankle CS, Strickland J, Troutman J, Wetmore BA, Kleinstreuer NC (2018) In vitro to in vivo extrapolation for high throughput prioritization and decision making. Toxicology in Vitro 47:213-227. doi:<https://doi.org/10.1016/j.tiv.2017.11.016>

14. Jamei M, Marciniak S, Feng K, Barnett A, Tucker G, Rostami-Hodjegan A (2009) The Simcyp® population-based ADME simulator. Expert Opinion on Drug Metabolism & Toxicology 5 (2):211-223

15. Bois FY, Jamei M, Clewell HJ (2010) PBPK modelling of inter-individual variability in the pharmacokinetics of environmental chemicals. Toxicology 278 (3):256-267

16. Jamei M (2016) Recent advances in development and application of physiologically-based pharmacokinetic (PBPK) models: a transition from academic curiosity to regulatory acceptance. Current pharmacology reports 2 (3):161-169

17. Poulin P, Theil F-P (2002) Prediction of pharmacokinetics prior to in vivo studies. II. Generic physiologically based pharmacokinetic models of drug disposition. Journal of pharmaceutical sciences 91 (5):1358-1370

18. Jones HM, Gardner IB, Watson KJ (2009) Modelling and PBPK simulation in drug discovery. The AAPS journal 11 (1):155-166

19. Jongeneelen FJ, Berge WFT (2011) A generic, cross-chemical predictive PBTK model with multiple entry routes running as application in MS Excel; design of the model and comparison of predictions with experimental results. Annals of Occupational hygiene 55 (8):841-864

20. Ellison CA, Blackburn KL, Carmichael PL, Clewell III HJ, Cronin MT, Desprez B, Escher SE, Ferguson SS, Grégoire S, Hewitt NJ (2019) Challenges in working towards an internal threshold of toxicological concern (iTTC) for use in the safety assessment of cosmetics: Discussions from the Cosmetics Europe iTTC Working Group workshop. Regulatory Toxicology and Pharmacology 103:63-72

21. Wambaugh JF, Wetmore BA, Pearce R, Strope C, Goldsmith R, Sluka JP, Sedykh A, Tropsha A, Bosgra S, Shah I (2015) Toxicokinetic triage for environmental chemicals. Toxicological Sciences 147 (1):55-67

22. Cohen Hubal EA, Wetmore B, Wambaugh J, El-Masri H, Sobus J, Bahadori T (2018) Advancing internal exposure and physiologically-based toxicokinetic modeling for 21st-century risk assessments. Journal of Exposure Science and Environmental Epidemiology

23. Oreskes N (1998) Evaluation (not validation) of quantitative models. Environmental Health Perspectives 106 (Suppl 6):1453

24. Caldwell JC, Evans MV, Krishnan K (2012) Cutting edge PBPK models and analyses: providing the basis for future modeling efforts and bridges to emerging toxicology paradigms. Journal of toxicology 2012

25. Chiu WA, Barton HA, DeWoskin RS, Schlosser P, Thompson CM, Sonawane B, Lipscomb JC, Krishnan K (2007) Evaluation of physiologically based pharmacokinetic models for use in risk assessment. Journal of Applied Toxicology 27 (3):218-237

26. Pilkey OH, Pilkey-Jarvis L (2007) Useless arithmetic: why environmental scientists can't predict the future. Columbia University Press,

27. Sayre RR, Wambaugh JF, Grulke CM (2020) Database of pharmacokinetic time-series data and parameters for 144 environmental chemicals. Scientific Data

28. Pearce RG, Setzer RW, Strope CL, Sipes NS, Wambaugh JF (2017) Httk: R package for high-throughput toxicokinetics. Journal of Statistical Software 79 (1):1-26

29. Ring CL, Pearce RG, Setzer RW, Wetmore BA, Wambaugh JF (2017) Identifying populations sensitive to environmental chemicals by simulating toxicokinetic variability. Environment International 106:105-118. doi:<https://doi.org/10.1016/j.envint.2017.06.004>

30. Breen M, Ring CL, Kreutz A, Goldsmith M-R, Wambaugh JF (2021) High-Throughput PBTK models for In Vitro to In Vivo Extrapolation.

31. Feynman RP (1974) Cargo cult science. Engineering and Science 37 (7):10-13

32. Manikas K (2016) Revisiting software ecosystems research: A longitudinal literature study. Journal of Systems and Software 117:84-103

33. Frakes WB, Kang K (2005) Software reuse research: Status and future. IEEE transactions on Software Engineering 31 (7):529-536

34. Wickham H (2015) R packages: organize, test, document, and share your code. " O'Reilly Media, Inc.",

35. Dennis JB (1975) Modularity. In: Software Engineering. Springer, pp 128-182

36. Scott ML (2000) Programming language pragmatics. Morgan Kaufmann,

37. Reynolds JF, Acock B (1997) Modularity and genericness in plant and ecosystem models. Ecological modelling 94 (1):7-16

38. Ito K, Houston JB (2004) Comparison of the use of liver models for predicting drug clearance using in vitro kinetic data from hepatic microsomes and isolated hepatocytes. Pharmaceutical research 21 (5):785-792

39. Schmitt W (2008) General approach for the calculation of tissue to plasma partition coefficients. Toxicology in Vitro 22 (2):457-467

40. Pearce RG, Setzer RW, Davis JL, Wambaugh JF (2017) Evaluation and calibration of high-throughput predictions of chemical distribution to tissues. Journal of Pharmacokinetics and Pharmacodynamics 44 (6):549-565. doi:10.1007/s10928-017-9548-7

41. Bois FY (2009) GNU MCSim: Bayesian statistical inference for SBML-coded systems biology models. Bioinformatics 25 (11):1453-1454

42. Soetaert KE, Petzoldt T, Setzer RW (2010) Solving differential equations in R: package deSolve. Journal of statistical software 33

43. Wambaugh JF, Wetmore BA, Ring CL, Nicolas CI, Pearce R, Honda G, Dinallo R, Angus D, Gilbert J, Sierra T, Badrinarayanan A, Snodgrass B, Brockman A, Strock C, Setzer W, Thomas RS (2019) Assessing Toxicokinetic Uncertainty and Variability in Risk Prioritization.