Pharmacolibrary - Free Library to Model Pharmacology.

Tomáš Kulhánek[1,3,4] Filip Ježek[2,3] Jiří Kofránek[3,4] Marek Mateják[3,5] Stef Rommes[1]

[1]VITO, Mol, Belgium
[2]University of Michigan, Ann Arbor, USA
[3]First Faculty of Medicine, Charles University, Prague, Czechia
[4]Creative Connections, s.r.o., Prague, Czechia
[5]Institute for Clinical and Experimental Medicine, Prague, Czechia

Abstract

Mechanistic modeling of drug behavior and response is essential for rational drug development and personalized therapy, yet constructing, maintaining, reusing and customizing complex pharmacokinetic–pharmacodynamic and physiologically based pharmacokinetic models can be error-prone when implemented solely via equations or code. We introduce Pharmacolibrary, a free Modelica library offering standardized acausal components for pharmacokinetics, pharmacodynamics, toxicokinetics/toxicodynamics and pharmacogenomics from compartmental and physiologically based templates to effect models and genotype–phenotype records—to simplify model reuse, customization, and interoperability. Its utility is showcased with gentamicin, midazolam, and fentanyl case studies, including pharmacogenomics-driven clearance adjustments and pharmacodynamics simulations.
Keywords: Modelica, Pharmacology, Pharmacokinetics, Pharmacodynamics, Pharmacogenomics, Modeling

1 Introduction

Pharmacology investigates how drugs modulate biological functions—spanning their origin, mechanisms of action, therapeutic benefits, and adverse effects—while toxicology focuses on chemical-induced harm. Core subfields include pharmacokinetics (PK; absorption, distribution, metabolism and elimination processes of a drug in organism) and pharmacodynamics (PD; effect, concentration–response relationships), with analogous toxicokinetics/toxicodynamics (TK/TD) in the toxicity domain. Pharmacogenomics (PGx) quantifies how genetic variation shapes PK and PD. Physiologically based pharmacokinetics (PBPK) augments classical PK with anatomical and physiological detail to predict drug distribution(Rosenbaum 2016). Quantitative systems pharmacology (QSP) further integrates PK/PD with systems-biology to capture drug interactions within complex molecular networks(Azer et al. 2021).

A variety of commercial and academic tools support these methods: Certara's SimCyp(Jamei et al. 2009) and Bayer's PK-Sim/MoBi for PBPK and Open-system pharmacology community(Lippert et al. 2019); MediWare's MWPharm++ for pharmacogenetically informed therapeutic drug monitoring(Schön et al. 2022); MATLAB SimBiology® and its gPKPDSim extension for PK/PD workflows(Hosseini et al. 2018); and Modelica-based libraries (Biochem, Physiolibrary, Chemical) for pathway, physiological, and physicochemical modeling. However, these platforms either specialize in a single aspect or blend modeling with simulation in proprietary environments.

Some of the existing Modelica libraries can be currently used to mimic e.g pharmacokinetics like the Biochem library for biochemical pathways(Nilsson and Fritzson 2005), Physiolibrary for human physiology models(Mateják et al. 2014) or Chemical for using electrochemical potential and physical chemistry principles to model chemical reaction and behavior of various substances(Matejak et al. 2015). However, these libraries focus on different aspects of biology, physiology and physical chemistry.

Here, we introduce *Pharmacolibrary*, a reusable Modelica library unifying PK, PD, PBPK, TK/TD, and PGx constructs with standardized pharmacological terminology, units, and acausal connectors. By decoupling model definition from simulation, our library leverages Modelica's auto-generation of correct equations from diagrams, supports FMU export for integration into Python/Julia/-MATLAB/WebApp workflows, and interoperates with existing Modelica libraries to enable advanced QSP applications. In the following sections, we describe the library's components and demonstrate exemplar models that reproduce experimentally observed drug behaviors.

2 Pharmacolibrary

As a foundational concept for pharmacology modeling, we first define the following connectors, which hold and exchange drug mass and concentration via various types between components.

The ConcentrationPort connector (table 1) contains MassFlowRate as a flow variable and MassConcentration as a non-flow variable that is usually used to model chemical domain. The ConcentrationPort is intended to be used in compartment based pharmacokinetics model where the volumetric flow is not taken into account or is neglected and ideal mixing can be assumed among connected components.

The FlowPort connector (table 2) contains Volume-

	**ConcentrationPort** is acausal connector that contains mass flow rate quantity q_m and concentration c defined as follows:
	
	`flow MassFlowRate qm;`
	`MassConcentration c;`
	
	When a component is connected using such a connector, standard Modelica tool will ensure that flow variable q_m is not accumulated in connected connectors by ensuring that $\sum_{i=0}^{n} q_{m_i} = 0$ and also that all non-flow variables - in this case concentration c are equilibrated i.e. $c_1 = c_2 = ... = c_n$ (n is number of connected components via a connector).

Table 1. Graphical connectors for drug-mass and concentration exchange.

FlowRate as a flow variable and MassConcentration as a stream variable and Pressure as a non-flow variable. The combination of VolumeFlowRate and Pressure is usually used to model pressure-volume hydraulic domain, we additionally add a concentration. The FlowPort is intended to be used in physiology based pharmacokinetics model (PBPK) where the substance (e.g., drug) transport is resolved through branching and merging conduits, automatically enforcing mass-balanced mixing at junctions according to the instantaneous flow rates.

	**FlowPort** is acausal connector that contains flow quantity q_v, stream quantity c and non-flow quantity p defined as follows:
	
	`flow VolumeFlowRate qv;`
	`stream MassConcentration c;`
	`Pressure p;`
	
	Using such definition, when a component is connected using a connector, standard Modelica tool will ensure that flow variable q_v is not accumulated in connected connectors by ensuring that $\sum_{i=0}^{n} q_{v_i} = 0$ and additionally all variables in stream connector (in this case concentration c) are ensured to be correctly and physically consistently mixed based on the flow quantity. All non-flow (potential) quantities (in this case pressure p) are equalized among connected components.

Table 2. Graphical connectors for drug-mass and concentration exchange in case of volumetric flow and mixing is taken into account.

By employing these two connector types, we can distinguish between pure compartment based models (where active flow and compartmental distribution can be simplified to diffusion kinetic) and PBPK models (where drug particles are exchanged via active distribution, i.e. volume flow rate and compartment are matched with anatomical and/or physiological tissues or systems).

2.1 Pharmacokinetics - compartment models

Pharmacokinetics studies how drug is absorbed, distributed, metabolized, and excreted (the “ADME” processes). The compartmental pharmacokinetic (PK) model abstracts the organism as one, two, or several interconnected compartments through which the drug distributes, with exchange between compartments generally described by first-order kinetics.(Fleishaker and Smith 1987) A compartment need not map to a specific anatomical tissue or organ; its defining parameter—the apparent volume of distribution V_d—is usually obtained by numerically fitting the time-course of blood-concentration data after drug administration.

By measuring drug concentrations in blood, tissues, or other biological fluids, pharmacokinetics compartmental analysis seeks to characterize the parameters of the components involved in ADME processes. For absorption the known parameters are dose mass, dose duration, administration route and unknown is bioavailability F. For distribution the volume distribution V_d of compartments. For Metabolism and Excretion the kinetic rates and clearance rates.

For modeling mainly the ’Distribution’ part of ADME processes in pharmacokinetics, we define the reusable compartment components as in table 3. For modeling the rest of the ADME processes, we define additional components as in table 4.

	**NoPerfusedTissueCompartment** expresses basic theoretical volumetric distribution of a drug in a hypothetical compartment, that does not necessarily be equivalent to an anatomical tissue or organ. With the following equations: $$\frac{dM}{dt} = q_m$$ $$C = \frac{M}{V_d}$$ Where derivation of M is mass flow rate q_m and concentration C is mass of drug in compartment M divided by compartment’s volume of distribution V_d.
	**SystemicTissueCompartment** express volumetric distribution of a drug via arterial or venous bloodstream through 2 flowports and 1 concentration port using these main equations: $$\frac{dM}{dt} = \sum q_{vxi} \times \text{actualStream}(c_{xi}) + q_m$$ $$c_a = c_b = \frac{M}{V_d}$$ where M is drug total mass in compartment, parameter V_d is volume of distribution, $q_v a$ and $q_v b$ are distribution of volume flow rates in ports a and b and q_m is mass flow rate that may happen directly in the blood tissue. c_a and c_b are actual concentrations on connected ports a and b respectively.
	**TissueCompartment** expresses not only distribution of a drug through bloodstream, but additional a simple constant tissue-blood concentration ratio k_{TB} as $$c = \frac{c_t}{k_{TB}}$$

Table 3. Compartment components.

DOI
10.3384/ecp21861

	**TransferFirstOrderNonSym** express basic first order non-symetric transfer between compartments defined by intercompartmental clearance parameters Cl_a and Cl_b.
	**SingleDose** expresses a source that contributes to concentration change by administering a certain amount of drug during administering duration into the 1 concentration port. $$q_m = \frac{\text{adminMass}}{\text{duration}}$$
	**FlowGround** The component FlowGround express the ground of concentration within the stream from the flowport connector. Note, that this component does not affect the model behavior, as no volume normally flows in or out, yet is required for mathematical completeness. An error is raised when any flow leaks or is added inadvertently.
	**ClearanceDrivenElimination** expresses simplified elimination that is driven by the clearance parameter by the following equation: $$q_m = CL \times c$$ $$\frac{dM_{exc}}{dt} = q_m$$ Where M_{exc} is excreted drug mass, CL is clearance parameter usually in liters per minute, c is drug concentration and q_m is massflowrate of the drug that is eliminated.
	**FixedFlow** express enumerated flow that is forced to be through this component. It can be used e.g. to mimic cardiac output or partial flow within a specific organ or tissue. It's parameter is prescribed flow Q which is then given to the connected ports a and b as: $$q_{va} = -q_{vb} = Q$$

Table 4. Additional components

2.2 1-compartment models

The single-compartment pharmacokinetic (PK) model is a theoretical model where whole organism is represented as a single, well-mixed compartment. In this model, the drug is absorbed according to its bioavailability, distributed uniformly throughout the apparent volume of distribution, and eliminated at a constant rate. Such a PK model can be described by the following integral equation of concentration c over time t where initial concentration c_0 and clearance rate of elimination Cl:

$$c = c_0 \times e^{-Cl \cdot t} \tag{1}$$

or in derivative form

$$\frac{dc}{dt} = -Cl \times c \tag{2}$$

When a mass of a drug is taken into account M, a theoretical parameter "volume distribution" V_d is in equation:

$$c = \frac{M}{Vd} \tag{3}$$

So the PK model can be described as

$$\frac{dM}{dt} = -Cl \times c \tag{4}$$

In Modelica language, the equations (3) and (4) can be implemented directly as seen in model implementation fig.1 (A). However the component NonPerfusedTissueComponent (implementing equation (3)) is connected with ClearanceDriveElimination which implements and correctly connect equation (4) so the diagram mode Fig.1 (B) is equivalent to the text mode of implementation.

(A)

```
within Pharmacolibrary.Examples;

model PK_1C_GentamicinEquations
  extends Modelica.Icons.Example;
  Pharmacolibrary.Types.Mass M(start=0.000400) "Mass of
drug [kg]";
  Pharmacolibrary.Types.MassConcentration C
(displayUnit="mg/l") "Concentration of drug";
  parameter Pharmacolibrary.Types.Volume Vd = 0.0175
"volume of distribution [m3]";
  parameter Pharmacolibrary.Types.VolumeFlowRate Cl = 0.1/
(1000*60) "elimination clearance [m3/s]";
equation
  C = M / Vd; //(12)
  der(M) = - Cl * C; //(13)
  annotation( ...);
end PK_1C_GentamicinEquations;
```

(B)

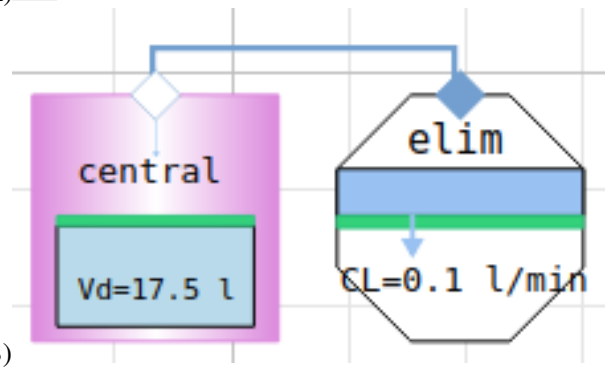


Figure 1. Equivalent representations of the single-compartment model. (A) Text mode, in which the governing equations (3) and (4)—are written out explicitly. Text mode is ideal for simple models or for defining individual components. (B) Diagram mode, which links predefined component icons into a acausal component diagram. Diagram mode is preferred when domain-specific components are already well defined and can be reused and composed in different settigns. Example parameters of PK of gentamicin with an initial dose of 400 mg, a clearance of 0.1 L/min, and a volume of distribution of 17.5 L. In text mode, the parameter values are declared in basic SI units. In diagram mode, tools such as OpenModelica(Fritzson et al. 2020) allow user to chose the prefered unit and automatically recalculate and display all units in SI within the component icons, dialogs and plots.

This model requires setting the initial drug mass at time zero. To incorporate single-dose absorption, a SingleDose component can be connected as shown in the modified diagram in Fig 2.

Single-compartment models describe the PK of a drug whose distribution via the cardiovascular arterial tree system can be neglected—typically because they are absorbed and distributed rapidly and do not significantly bind to tissues.

For example, gentamicin is an antibiotic administered intravenously, usually once daily, to treat serious bacterial infections. Sample parameter values of gentamicine's PK model are given in Figure 1 and its caption. A primary concern in drug administration is efficacy; for gentamicin, the minimum inhibitory concentration (MIC) represents the lowest concentration that prevents bacterial growth. Another concern is toxicity: gentamicin can cause nephrotoxicity (kidney toxicity) and ototoxicity (ear toxicity). The toxicity for gentamicine should not reach a $C_{tox,peak}$ value for short term toxic effect and minimal concentration before second dose should be bellow $C_{tox,valley}$ or sometimes dedicated as $C_{tox,trough}$. The dosing goal for clinical pharmacologist can be described as simplified rule to maintain drug concentrations above the MIC for a sufficient portion of the dosing interval while keeping peak level and trough level below toxic thresholds. For gentamicin the MIC and toxicity thresholds are in Fig.2(C).

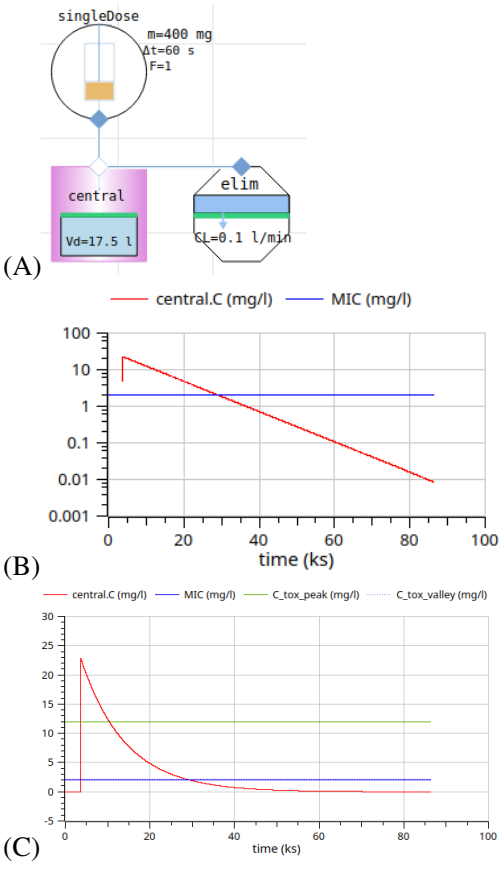


Figure 2. (A) Single-compartment model with a SingleDose component, a central compartment, and an elimination component. (B) Simulated concentration plotted on a logarithmic scale, illustrating the model's constant log-linear decline. (C) Example simulation configured for gentamicin pharmacokinetics—initial dose 400 mg, volume of distribution 17.5 L, clearance 0.1 L/min—showing the central compartment concentration (red), the minimum inhibitory concentration (MIC)(Hodiamont et al. 2022), and observed peak and valley (trough) toxicity levels (green) and (dotted blue).

2.3 2-compartment models

If the experimental concentration–time curve is biphasic, a two-compartment model usually fits better: the central compartment represents blood and well-perfused tissues, while the peripheral compartment represents tissues that transiently store the drug. Figure 3(A) shows this arrangement—administration and elimination occur from the central compartment, and drug moves between compartments by first-order transfer.

Midazolam—a fast-acting benzodiazepine used for sedation, anxiolysis, and anesthesia induction—exemplifies a drug whose intravenous pharmacokinetics are well described by a two-compartment model. Sample PK parameters are in Figure 3 caption. The plasma concentration required for anesthesia induction must remain above a defined minimal effective threshold C_{min}.

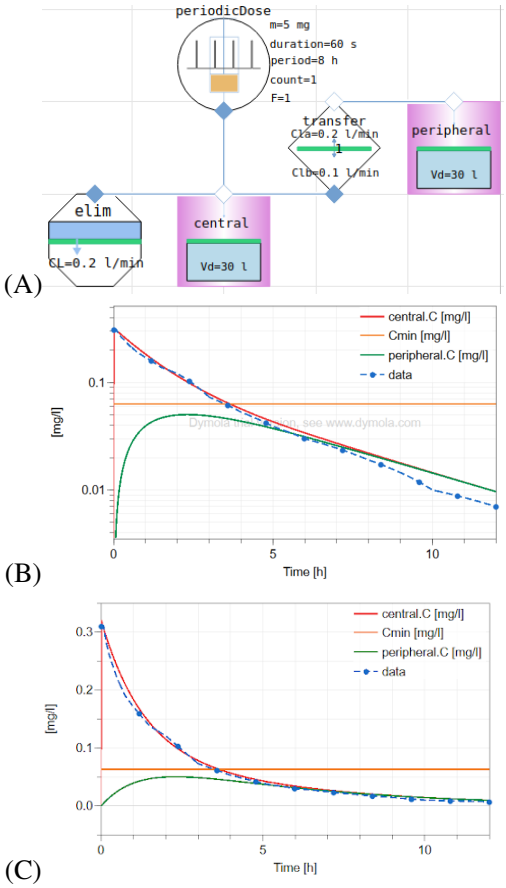


Figure 3. (A) Diagram view of a two-compartment pharmacokinetic model, showing drug administration and elimination in the central compartment with bi-directional transfer to the peripheral compartment. (B) Simulation on a logarithmic scale, illustrating the non-constant log-linear decline of concentrations in both compartments. validated against data (blue dots) from Heizmann et al. (1983)(Heizmann, Eckert, and Ziegler 1983). (C) Simulation of midazolam concentration in the central (red) and peripheral (green) compartments, alongside the minimal effective concentration (blue). A 9.6 mg IV dose is administered; both compartments have an apparent volume of distribution $V_d = 30l$, inter-compartmental transfer rate constants $k_{12} = 0.2h^{-1}$ and $k_{21} = 0.1h^{-1}$ and a systemic clearance $Cl = 0.2l.min^{-1}$

DOI
10.3384/ecp21861

2.4 Multicompartmental models

When a drug distributes into multiple tissue types, its post-administration concentration decline exhibits more complex behavior, making multi-compartment models more appropriate than single- or two-compartment models.

Fentanyl is a potent synthetic opioid used to manage severe pain and as an anesthetic adjunct during surgery. After intravenous administration, it rapidly distributes into highly perfused organs—such as the brain, heart, lungs, and kidneys—while also partitioning more slowly into poorly perfused tissues like fat and muscle. Over time, both tissue groups release fentanyl back into the plasma. The pharmacokinetics of fentanyl can be captured by the three-compartment model shown in Figure 4, in which the "peripheral" and "peripheral2" compartments represent highly and poorly perfused tissues, respectively.

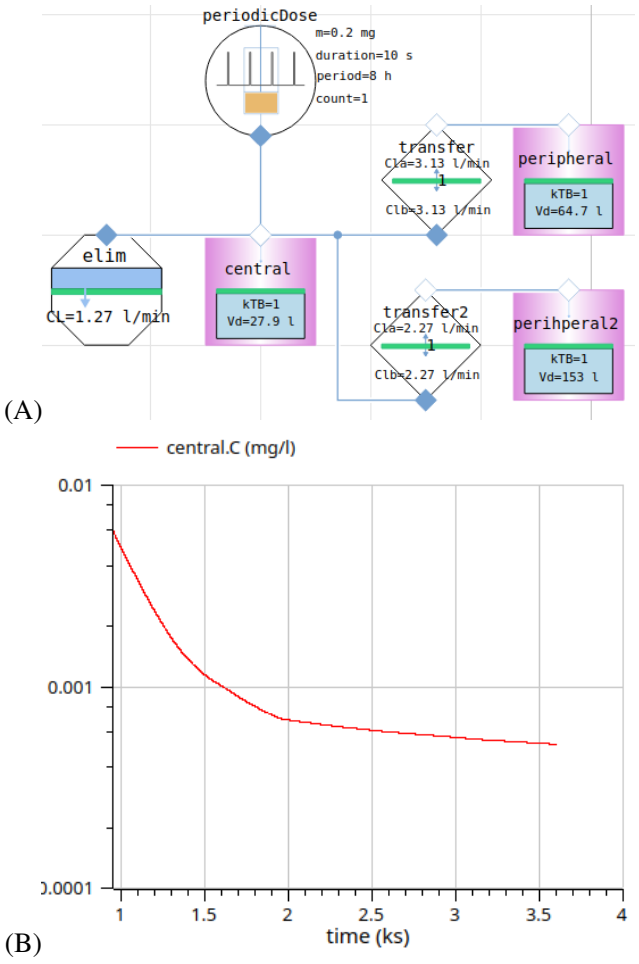


Figure 4. (A) Three-compartment PK model diagram. (B) Log-scale simulation of fentanyl plasma concentration 10–60 min after a 0.2 mg bolus. Parameters: V_d=27.9L (central), 64.7L (peripheral$_1$), 153L (peripheral$_2$); Cl=1.27, 3.13, 2.27L min^{-1}, respectively(Kaneda and Han 2009). The curve inflects as both peripheral pools return drug to the central compartment.

2.5 Physiology based pharmacokinetics models - component models

Compartmental PK models fit empirical compartments to concentration–time data but lack anatomical meaning. Physiologically based PK (PBPK) models instead assign compartments to real organs, enabling organ-level and inter-individual simulations when parameters are available.

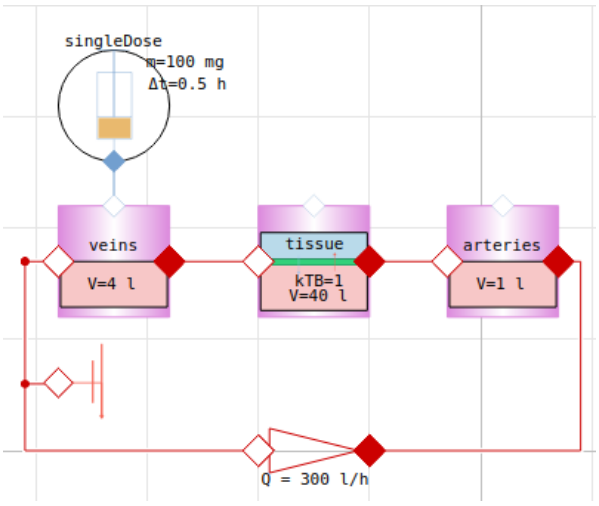


Figure 5. Model diagram of three-component PBPK model. Linked red FlowPorts enforce mass flow rate and resolve stream variables by flow direction. ConcentrationPorts join SingleDose→Veins (IV 100 mg in 0.5 h). A dedicated block sets unidirectional flow at a cardiac output of 5 L min^{-1} (300 L h^{-1})

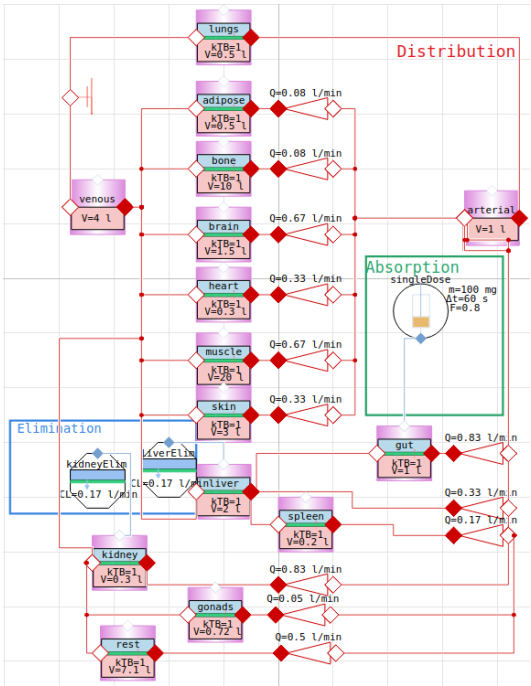


Figure 6. Whole-body PBPK model, utilizing main tissues with normal blood flow distribution. This is suitable for extension by a particular drug.

```
322  equation
323    kidney.cport.c = kidneyElim.cport.c;
324    liver.cport.c = liverElim.cport.c;
325    singleDose.cport.c = gut.cport.c;
326    gut.cport.qm + singleDose.cport.qm = 0.0;
327    venous.cport.qm = 0.0;
328    lungs.port_a.qv + flowGround.port_a.qv + venous.port_a.qv = 0.0;
329    lungs.cport.qm = 0.0;
330    arterial.port_b.qv + lungs.port_b.qv = 0.0;
331    arterial.cport.qm = 0.0;
332    restFlow.port_a.qv + testesFlow.port_a.qv + fixedFlow9.port_a.qv + fixedFlow8.port_a.qv + fixedFlow7.port_a.qv + fixedFlow6.port_a.qv +
   fixedFlow5.port_a.qv + fixedFlow4.port_a.qv + fixedFlow3.port_a.qv + fixedFlow2.port_a.qv + fixedFlow1.port_a.qv + fixedFlow.port_a.qv +
   arterial.port_a.qv = 0.0;
333    adipose.port_b.qv + fixedFlow.port_b.qv = 0.0;
334    adipose.cport.qm = 0.0;
335    rest.port_a.qv + gonads.port_a.qv + kidney.port_a.qv + liver.port_a.qv + skin.port_a.qv + muscle.port_a.qv + heart.port_a.qv + brain.port_a.qv +
   bone.port_a.qv + adipose.port_a.qv + venous.port_b.qv = 0.0;
336    bone.port_b.qv + fixedFlow1.port_b.qv = 0.0;
337    bone.cport.qm = 0.0;
338    brain.port_b.qv + fixedFlow2.port_b.qv = 0.0;
339    brain.cport.qm = 0.0;
340    heart.port_b.qv + fixedFlow3.port_b.qv = 0.0;
341    heart.cport.qm = 0.0;
342    muscle.cport.qm = 0.0;
343    fixedFlow4.port_b.qv + muscle.port_b.qv = 0.0;
344    skin.cport.qm = 0.0;
345    fixedFlow5.port_b.qv + skin.port_b.qv = 0.0;
346    fixedFlow6.port_b.qv + gut.port_b.qv = 0.0;
347    spleen.cport.qm = 0.0;
348    liverElim.cport.qm + liver.cport.qm = 0.0;
349    fixedFlow7.port_b.qv + liver.port_b.qv + spleen.port_a.qv + gut.port_a.qv = 0.0;
350    fixedFlow8.port_b.qv + spleen.port_b.qv = 0.0;
351    kidney.port_b.qv + fixedFlow9.port_b.qv = 0.0;
352    kidneyElim.cport.qm + kidney.cport.qm = 0.0;
353    gonads.cport.qm = 0.0;
354    testesFlow.port_b.qv + gonads.port_b.qv = 0.0;
355    rest.cport.qm = 0.0;
356    restFlow.port_b.qv + rest.port_b.qv = 0.0;
357    singleDose.cport.qm = if singleDose.adminTime <= time and time < singleDose.adminTime + singleDose.duration then -singleDose.F * singleDose.adminMass /
   singleDose.duration else 0.0;
358    der(singleDose.TotalCumulativeMass) = -singleDose.cport.qm;
359    der(venous.M) = smooth(0, venous.port_a.qv * (if venous.port_a.qv > 0.0 then $OMC$inStreamDiv(($OMC$PositiveMax(-flowGround.port_a.qv,
   1.6666666666666666e-15) * flowGround.port_a.c + $OMC$PositiveMax(-lungs.port_a.qv, 1.6666666666666666e-15) * lungs.port_a.c) / ($OMC$PositiveMax(-
   flowGround.port_a.qv, 1.6666666666666666e-15) + $OMC$PositiveMax(-lungs.port_a.qv, 1.6666666666666666e-15)), venous.port_a.c) else venous.port_a.c)) +
   smooth(0, venous.port_b.qv * (if venous.port_b.qv > 0.0 then $OMC$inStreamDiv(($OMC$PositiveMax(-adipose.port_a.qv, 1.6666666666666666e-15) *
   adipose.port_a.c + $OMC$PositiveMax(-bone.port_a.qv, 1.6666666666666666e-15) * bone.port_a.c + $OMC$PositiveMax(-brain.port_a.qv, 1.6666666666666666e-15)
   * brain.port_a.c + $OMC$PositiveMax(-heart.port_a.qv, 1.6666666666666666e-15) * heart.port_a.c + $OMC$PositiveMax(-muscle.port_a.qv,
   1.6666666666666666e-15) * muscle.port_a.c + $OMC$PositiveMax(-skin.port_a.qv, 1.6666666666666666e-15) * skin.port_a.c + $OMC$PositiveMax(-liver.port_a.qv,
   1.6666666666666666e-15) * liver.port_a.c + $OMC$PositiveMax(-gonads.port_a.qv, 1.6666666666666666e-15) * gonads.port_a.c + $OMC$PositiveMax(-
   kidney.port_a.qv, 1.6666666666666666e-15) * kidney.port_a.c + $OMC$PositiveMax(-rest.port_a.qv, 1.6666666666666666e-15) * rest.port_a.c) /
   ($OMC$PositiveMax(-adipose.port_a.qv, 1.6666666666666666e-15) + $OMC$PositiveMax(-bone.port_a.qv, 1.6666666666666666e-15) + $OMC$PositiveMax(-
   brain.port_a.qv, 1.6666666666666666e-15) + $OMC$PositiveMax(-heart.port_a.qv, 1.6666666666666666e-15) + $OMC$PositiveMax(-muscle.port_a.qv,
   1.6666666666666666e-15) + $OMC$PositiveMax(-skin.port_a.qv, 1.6666666666666666e-15) + $OMC$PositiveMax(-liver.port_a.qv, 1.6666666666666666e-15) +
   $OMC$PositiveMax(-gonads.port_a.qv, 1.6666666666666666e-15) + $OMC$PositiveMax(-kidney.port_a.qv, 1.6666666666666666e-15) + $OMC$PositiveMax(-
   rest.port_a.qv, 1.6666666666666666e-15)), venous.port_b.c) else venous.port_b.c)) + venous.cport.qm;
360    venous.port_a.c = venous.CB;
361    venous.port_b.c = venous.CB;
362    venous.port_a.qv + venous.port_b.qv = 0.0;
363    venous.C = venous.M / venous.VNonZero;
364    venous.CB = venous.C / venous.kTB;
365    venous.freeTissueConc = venous.fu * venous.C;
366    venous.freeBloodConc = venous.fu * venous.C / venous.kTB;
367    venous.cport.c = venous.freeTissueConc;
368    flowGround.port_a.c = 0.0;
369    assert(abs(flowGround.port_a.qv) < 1e-15, "some flow is lost");
370    der(lungs.M) = smooth(0, lungs.port_a.qv * (if lungs.port_a.qv > 0.0 then $OMC$inStreamDiv(($OMC$PositiveMax(-flowGround.port_a.qv,
   1.6666666666666666e-15) * flowGround.port_a.c + $OMC$PositiveMax(-venous.port_a.qv, 1.6666666666666666e-15) * venous.port_a.c) / ($OMC$PositiveMax(-
   flowGround.port_a.qv, 1.6666666666666666e-15) + $OMC$PositiveMax(-venous.port_a.qv, 1.6666666666666666e-15)), lungs.port_a.c) else lungs.port_a.c)) +
   smooth(0, lungs.port_b.qv * (if lungs.port_b.qv > 0.0 then arterial.port_b.c else lungs.port_b.c)) + lungs.cport.qm;
371    lungs.port_a.c = lungs.CB;
372    lungs.port_b.c = lungs.CB;
```

Figure 7. Excerpt from the flattened model—the text form automatically generated from the PBPK diagram by a standard Modelica tool. At line 332 the flows from all connected tissues sum to zero, and at line 359 the derivative of venous drug mass equals the flow-weighted mix of incoming concentrations, as dictated by the stream-connector logic. In a purely equation-based implementation these relations would have to be coded manually.

The basic three-component PBPK model, which accounts for cardiac output and drug distribution through the arterial and venous systems to target tissues, is illustrated in Figure 5.

Complex PBPK model follows cardiovascular distribution across multiple tissues (Figure 6). Linked red FlowPorts connect venous and arterial pools to fourteen organ/tissue compartments, automatically enforcing mass balance and setting stream variables by flow direction. A SingleDose block drives gut absorption (oral 100 mg, F = 0.8), while hepatic and renal clearance components (Cl=0.17 L min^{-1} each) handle elimination. A dedicated component sets cardiac output at 5 L min^{-1} (300 L h^{-1}). Figure 8 shows simulated tissue concentrations. An equivalent text model must state every equation and link, as illustrated by the flattened equation-based model generated by the Modelica tool in Figure 7.

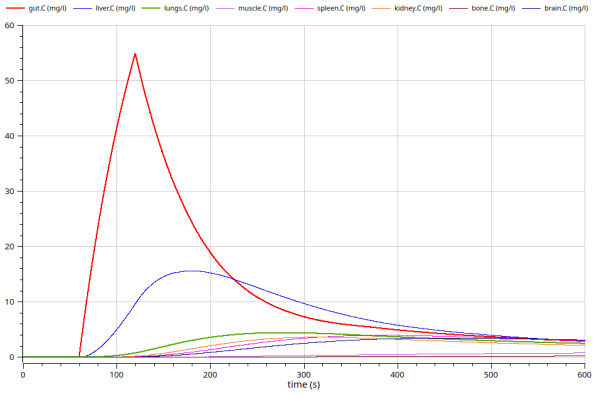


Figure 8. Simulation of the first 10 minutes of a generic drug using physiology-based pharmacokinetic model from Fig.6. Simulation reveals that majority of the drug administered orally goes first through liver and then it is subsequently distributed to other tissues.

DOI
10.3384/ecp21861

2.6 Pharmacodynamic models

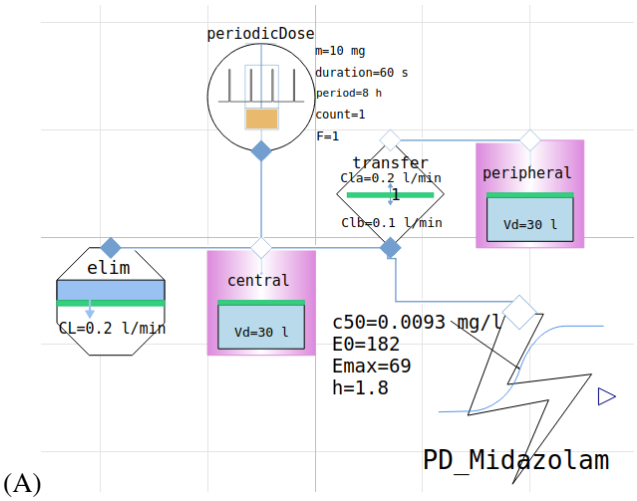
Pharmacodynamics (PD) is the bridge that converts "how much drug is present" into "how much effect is produced." In its direct or static form, the instantaneous drug concentration at the site of action is passed through an algebraic dose-response relationship. We present several such direct-effect components (Table 5): a linear effect component for low-dose proportionality, a hyperbolic E_{max} effect component for simple saturation, and a sigmoid-E_{max} effect component whose Hill exponent tunes the steepness of the transition. These components assume the effect is reversible: when the concentration falls, the response follows it back to baseline with no memory.

	**LinearEffect** depicted as equation $$E = E_0 + S \times C$$ Where E is effect, E_0 is baseline effect without drug influence, S is slope (effect per concentration), C is drug concentration.
	**EmaxEffect** Depicted by the equation $$E = E_0 + \frac{E_{max} \times C}{C_{50} + C}$$ where E is the final effect, E_0 is the baseline effect in the absence of drug, C is the drug concentration, C_{50} is the concentration producing 50% of the maximal effect E_{max}, and E_{max} is the maximal effect.
	**SigmoidEmaxEffect** Depicted by the equation $$E = E_0 + \frac{E_{max} \times C^h}{C_{50} + C^h}$$ where E is the final effect, E_0 is the baseline effect in the absence of drug, C is the drug concentration, C_{50} is the concentration producing 50% of the maximal effect E_{max}, E_{max} is the maximal effect, and h is the Hill exponent.

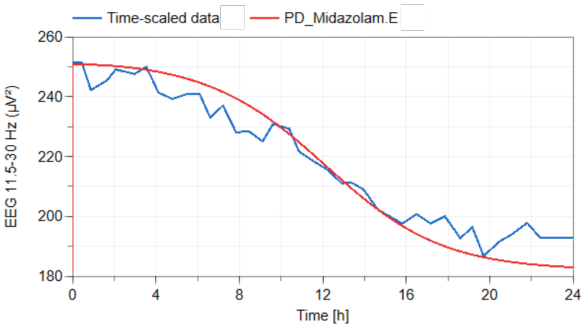
Table 5. Graphical connectors for drug-mass and concentration exchange.

Some drugs act irreversibly—e.g., covalent enzyme inhibition, receptor down-regulation, or DNA alkylation—so their effect depends on cumulative target inactivation rather than current concentration; these indirect, irreversible mechanisms are not yet covered in the current library.

Midazolam—whose intravenous disposition is captured by the two-compartment PK model in Figure 3—rapidly partitions into the highly perfused central nervous system, where it depresses electroencephalographic (EEG) activity, producing sedation, anxiolysis, and anaesthetic hypnosis. Cleton et al.(Cleton, Voskuyl, and Danhof 1998) showed that the concentration–effect relationship for the EEG β-ratio follows a sigmoid E_{max} curve. In the library, the midazolam PK–PD model couples the two-compartment PK model to a sigmoid-E_{max} PD component (Figure 9).



(A)



(B)

Figure 9. (A) PKPD midazolam model extension from Figure 3. (B) Comparison of simulation of the midazolam effect represented as EEG activity to the data reported in (Cleton, Voskuyl, and Danhof 1998).

2.7 Pharmacogenomic models

Pharmacogenomic (PGx) modeling seeks to quantify how inherited genetic variation modulates both drug pharmacokinetics and pharmacodynamics. We propose the following components to integrate genotype–phenotype–specific parameters—such as transporter activity, metabolic capacity, and receptor sensitivity—directly into established PK/PD frameworks. We define six commonly used metabolizer status classes:

```
type MetabolizerStatus =
    enumeration(
      Unknown      "No mapping implemented / uncertain",
      Poor         "Little or no functional activity",
      Intermediate "Reduced activity",
      Normal       "Reference activity",
      Rapid        "Increased activity",
      Ultra        "Greatly increased activity");
```

Next, we define two generic record types Genotype and Phenotype to encapsulate an individual's genotype (allelic composition), corresponding phenotype, and simplified functional modifications of PK/PD effects:

```
partial record Genotype
  parameter String allele[2] = {"*1","*1"};
  replaceable parameter PGx.Phenotype ph
      constrainedby PGx.Phenotype
      annotation( ...);
end Genotype;
```

```
record Phenotype
  parameter String[:] poor = fill("", 0);
  parameter String[:] intermediate = fill("", 0);
  parameter String[:] rapid = fill("", 0);
  parameter String[:] ultra = fill("", 0);
  parameter Real[6] CLscale = {1.0,0.7,0.85,1.0,1.2,1.3};
  parameter Real[6] Fscale  = {1,1,1,1,1,1};
end Phenotype;
```

Additional helper functions are defined as follows:

`phStatus(ph, allele1, allele2)` Maps a pair of star-alleles (`allele1`, `allele2`) and a `PhenotypeTemplate` instance (`ph`) to a `MetabolizerStatus`, using the allele lists specified in `ph`.

`phScaleCL(ph, allele1, allele2)` Returns the clearance-scaling factor corresponding to the combined genotype (`allele1`/`allele2`) and phenotype instance `ph`.

Based on these record templates, a modeler can define sample genotypes and phenotypes that may affect, for example, the renal clearance of a drug. For instance, the SLC22A2 gene encodes organic cation transporter 2 (OCT2), a key renal transporter responsible for the uptake and clearance of many endogenous substances and drugs from the blood into renal tubular cells, thereby influencing their urinary excretion (Koepsell 2013).

Sample implementation of SLC22A2 genotype holds default normal alleles denoted as $*1*1$ which can be changed later in runtime as a common parameter in a model:

```
record SLC22A2Genotype
  extends PGx.Genotype(
   allele = {"*1","*1"},
   redeclare parameter Phenotypes.OCT2_Phenotype ph );
end SLC22A2Genotype;
```

Phenotype OCT2 defines an intermediate metabolizer with at least one allele denoted as 808T and according altered clearance by factor 0.85:

```
record OCT2_Phenotype
  extends PGx.Phenotype(
    intermediate = {"808T"},
    CLscale = {1.0, 0.7, 0.85, 1.0, 1.2, 1.3},
    Fscale = {1, 1, 1, 1, 1, 1}
  );
end OCT2_Phenotype;
```

In order to integrate genotype–phenotype influences into the gentamicin PK model, we must adjust the clearance parameter—either by directly invoking the function phScaleCL() or by delegating the adjustment to a specialized ClearanceModifier model. An example of the gentamicin PK model, implemented in equations and augmented with PGx influence via ClearanceModifier, is shown in the following listing:

```
model PKPG_1C_GentamicineEquation
extends Modelica.Icons.Example;
  Pharmacolibrary.Types.Mass M(start = 0.000400);
  Pharmacolibrary.Types.MassConcentration C(displayUnit = "mg/l");
  parameter Pharmacolibrary.Types.Volume Vd = 0.0175;
  parameter Pharmacolibrary.Types.VolumeFlowRate Cl_nom = 0.1/(1000*60);
  parameter Boolean usePGx = true "true = apply genotype scaling";
  Pharmacolibrary.Pharmacogenomics.Modifiers.ClearanceModifier clMod(
    base = Cl_nom,
    redeclare type Gtype = Pharmacolibrary.Pharmacogenomics.Genotypes.SLC22A2Genotype,
    redeclare parameter Pharmacolibrary.Pharmacogenomics.Genotypes.SLC22A2Genotype g(
      allele = {"*1", "*1"}/* <- change one entry to "808T" for Intermediate */)
    );
  Types.VolumeFlowRate Cl_eff "effective clearance";
equation
  Cl_eff = if usePGx then clMod.modified else Cl_nom;
  C = M/Vd;
  der(M) = -Cl_eff*C;
  annotation( ...);
end PKPG_1C_GentamicineEquation;
```

PGx simulation of such PK model with normal allele and with variant 808T with altered clearance is seen in the following Figure 10:

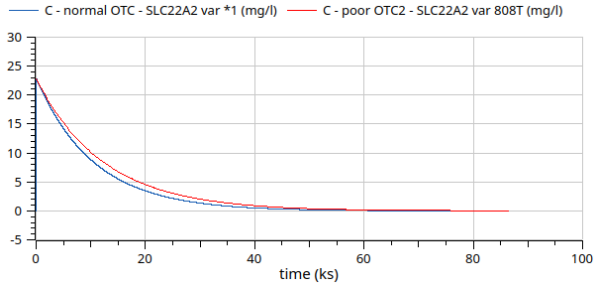


Figure 10. comparison of PGx simulation of PK model of gentamicine with normal SLC22A2 alleles compared to allele '808T' with altered clearance.

To support PGx modeling using diagram view, we define the components in Table 6.

With these components, we can extend PK models with additional PGx influence by applying clearance modifications appropriately and cascading the influence of multiple genes—e.g., CYP2C19 and SLC22A2—as illustrated in the model diagram in Figure 11.

	**ClearanceModifier** q_{vin} is baseline volume flowrate value, u is modification factor from an outside component, e.g. from genotype-phenotype component, q_{vout} is modified output volume flowrate value by equation: $$q_{vout} = u \times q_{vin}$$
	PhenotypeGenotype component joins genotype with it's phenotype and outputs the scale factor that can be used to modify other quantities $$\text{scale} = \text{phScaleCL}(p_h, a_1, a_2)$$ Where p_h is phenotype, a_1 is first allele and a_2 second allele.

Table 6. Components for diagram view to support pharmacogenomic modeling.

DOI 10.3384/ecp21861

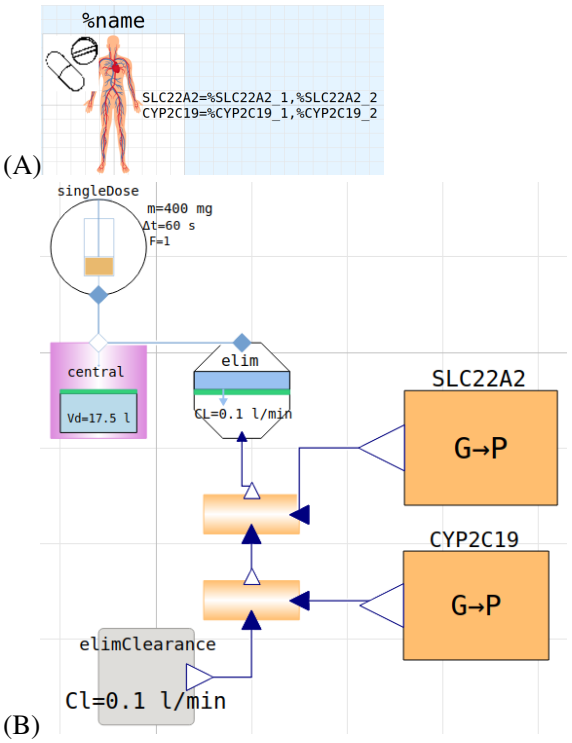


Figure 11. PKPGx model icon (A) and diagram (B) of 1-compartment PK of gentamicine where elimination clearance rate is influenced by PGx of CYP2C19 genotype and SLC22A2 genotype.

Theoretical influence of all combination of the variants of gene mutation of SLC22A2 and CYP2C19 on periodical administration of gentamicine is visible in model and simulation of 4 variants in Figure 12.

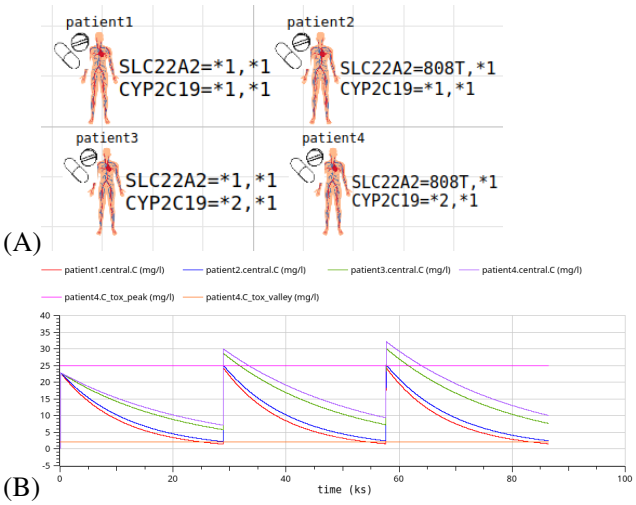


Figure 12. PGx of 4 variants of patients with various gene mutation of SLC22A2 and CYP2C19. Model diagram (A) showing the instance of PKPGx model with specific SLC22A2 alleles and CYP2C19 alleles (patient1 has normal alleles, patient2 with mutation 808T, patient3 with mutation *2 and patient4 with both mutations 8080T and *2). Simulation (B) of PK of gentamicine on all these patients during 3 days, gentamicin is administered every day. Toxicity peak and trough (valey) levels are shown too.

Note that while standard dosing in a normal patient keeps drug concentrations below toxic peak and trough thresholds, rare variants in the cited genes slow elimination; in patient3 and patient4 this raises levels above the toxicity limit, markedly increasing the risk of side effects.

The models in Figures 11 and 12 are illustrative only; the real impact of pharmacogene mutations on drug pharmacokinetics and pharmacodynamics is likely more complex and lies beyond the scope of this paper.

2.8 Integration to simulation workflows

Models exported as FMUs can be instantiated and customized via runtime parameters without requiring recompilation. The following figure illustrates a demo workflow that integrates FMU export into a Python environment using the FMPy library, retrieves experimental data from an external PK database, overlays the data on the simulation chart, and supports interactive parameter updates.

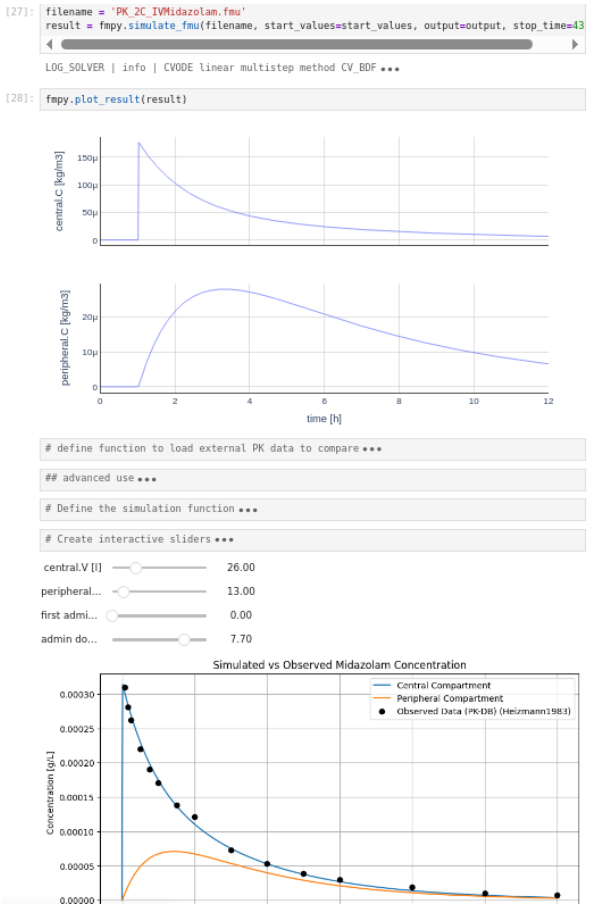


Figure 13. 2-compartment PK model of Midazolam exported to FMU and instantiated in Python using fmpy library. Jupyter notebook is used to visualize interactively model simulation and data of specific drug concentration time series retrieved from PK-DB (Grzegorzewski et al. 2020).

2.9 Integration to other Modelica libraries

Pharmacolibrary can be integrated with other Modelica libraries. The acausal connectors are unique for Phar-

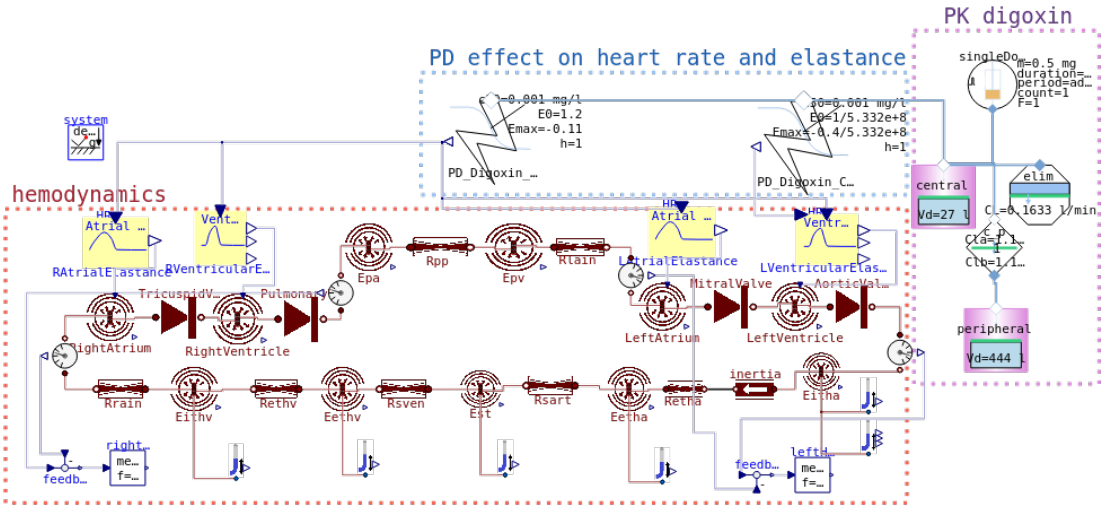


Figure 14. Example integration of PKPD model with Physiolibrary 3.0 (Mateják 2023) 1. hemodynamics of cardiocascular system as published by (Van Meurs 2011), 2. example of 2-compartment PK model of digoxin and PD model of influence of variable elastance during cardiac cycle as published by (Hornestam et al. 2003). PD effect on elastance/compliance just illustrative.

macolibrary, however causal connectors that holds only one variable can be directly connected to any appropriate component. Following the common convention in Modelica libraries, connector nominal values use SI units: compliance is expressed in $m^3\,Pa^{-1}$ (converted from $mL\,mmHg^{-1}$), and heart rate in s^{-1} (Hz) rather than beats min^{-1}. Digoxin is a drug used to treat e.g. acute atrial fibrilation as it has combined effect on inotropy (ventricular elastance) and heart rate. Example of PK PD integration of digoxin to model of hemodynamics is visible in Fig.14 with PD effect parameters in basic SI units and illustrative simulation is at Fig. 15.

3 Discussion

Implementing multicompartmental pharmacokinetic/pharmacodynamic models can be cumbersome and error-prone when using generic programming language, or modeling language and methods using equation-based or block-diagram approaches. While simple one- and two-compartment models (Fig. 1 A, B) can be implemented either in diagram mode or in equivalent equation text mode, more complex architectures quickly become unwieldy in text mode (e.g., Fig. 6, and 7). In contrast, Modelica tools automatically generate consistent textual representations from acausal diagram views—translating connected variables and component definitions into correct equations. Combination of acausal and stream connectors (connectors exchanging flow, non-flow and stream variables) ensure that non-flow (equality) and flow variables equilibrate appropriately, while stream connectors enforce mass balance by mixing based on flow direction. This component-oriented approach preserves the precision of underlying equations while retaining an intuitive, schema-like structure plausible for non-programmers.

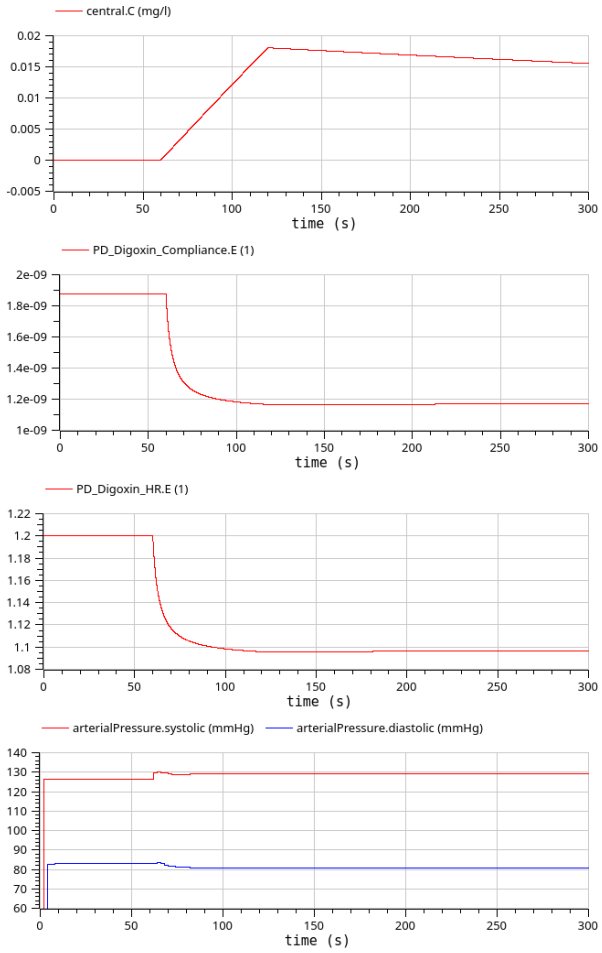


Figure 15. Simulation of PK, PD effect on compliance and heart rate and systolic/diastolic pressure in 5 minutes before, during and after administration of intravenously applied digoxin. Digoxin is administered after 60 seconds. Note that while systolic pressure slightly increases the diastolic pressure slightly decreases when the drug is administered.

DOI 10.3384/ecp21861

Pharmacology modeling must also accommodate extensive genetic and physiological variability. Embedding all genotype- and phenotype-specific variants directly in static diagrams can become impractical as well as to model PK of all drugs using drug-specific models. To address this, our library's core components are designed for post-compilation parameterization: users can adjust pharmacokinetic, pharmacodynamic and genotype-phenotype influences (e.g., via ClearanceModifier or phScaleCL()) at runtime without recompiling it.

Pharmacogenomics is rapidly maturing: e.g. the Dutch Pharmacogenetics Working Group and the Clinical Pharmacogenetics Implementation Consortium now recommend genotype-based dose adjustments for many drugs. Embedding these recommendations into physiologically based models enables truly personalized simulations that account for both genetic polymorphisms and physiological differences. Current Pharmacolibrary allows only basic components for PGx modeling which may be a foundation for systematic approach in this context interconecting phenotypes and it's effect on different tissues during drug lifecycle in organism.

Pharmacolibrary contains also basic PK models of all approved drugs that was generated by a workflow extracting knowledge from the literature—or infers it from structurally similar compounds—using recent advances in AI and large language models (LLMs). These PK models are stored in the "Drugs" package and categorized by Anatomical Therapeutic Chemical (ATC) codes[1]. Each model contains essential drug information and references generic one-, two-, or three-compartment PK model with specific instance parameter. Such generated models needs manual verification and refinement, however, may be useful as starting point for drug modeling. If a human modeler later derives additional parameters or adds another route, that supplementary model is saved alongside the generated one.

Drug effects often exhibit time lags or hysteresis between tissue concentrations and observable responses. Standard Modelica libraries support modeling such delays (e.g., via transport or delay blocks). Additionally, pharmacodynamic outcomes can be modulated by the body's regulatory feedback. By connecting PKPD models to broader physiological frameworks (for example, Modelica's Physiolibrary or Chemicals library), one can simulate homeostatic mechanisms and regulatory dynamics alongside drug action as it is illustrated in one example in Fig.14 and Fig.15.

As models can be exported to FMUs following the FMI standard and integrated into diverse simulation workflows (Python, Julia, MATLAB, or bespoke environments) the interoperability can be made with various other systems biology libraries, frameworks and workflows. One such example is illustrated integrating model in Jupyter notebook with Python code and data from PK-DB in Fig.13.

Because Pharmacolibrary is still in its infancy, a thorough comparison with commercial or open-source alternatives would be premature. Even so, within the broader Modelica ecosystem, the library already fills a crucial niche for modeling biological systems with an emphasis on chemical compounds and drugs.

3.1 Library limitations

To keep models transparent and user-friendly, we adopted pharmacology-centric units and terminology—such as volume flow rate (SI base unit m^3/s, displayed as l/min) via FlowPort connectors, even though under variant temperatures or pressures (e.g., during gas exchange or hypothermia) mass flow rate (kg/s) would be more accurate. A similar point applies to the use of concentration as a quantity: at membranes, it is actually the chemical potentials—not the concentrations—that equilibrate. However, modeling this explicitly would introduce unnecessary complexity, whereas our goal is to keep the model intuitive, and user-friendly.

4 Conclusion

In this work, we have introduced *Pharmacolibrary*, a flexible Modelica-based library for mechanistic pharmacological modeling that spans pharmacokinetics, pharmacodynamics, toxicokinetics/toxicodynamics, physiology based PK/PD and pharmacogenomics. By providing standardized acausal connectors, reusable compartment and effect components, and genotype–phenotype records, our library simplifies model construction, ensures mass and concentration balance. The models are Exportable as FMUs in accordance with the FMI standard, Pharmacolibrary models can be embedded into diverse computational workflows (e.g., Python, Julia, R,...), enabling interactive parameterization, data-driven validation, and high-throughput simulations.

The library is open-source maintained at GITHUB [2] and listed in ZENODO (Kulhánek and Ježek 2025)

Implementations of gentamicin, midazolam, fentanyl, and digoxin—including genotype/phenotype-adjusted clearance—demonstrate that Pharmacolibrary can replicate experimental data and underpin personalised dosing. Moreover, the library contains simplified generated PK models of all drugs classified by anatomical therapeutical chemical (ATC) codes. As of 06/2025 there were 5496 unique records at ATC level 5, corresponding to 4363 distinct chemical substances.

By separating model definition from the solver and building on the Modelica ecosystem (Physiolibrary, Chemicals, Biochem), it promotes reproducibility, interoperability, and extensibility. Ongoing work will add transporter and enzyme-kinetic blocks, compile and improve a catalogue of simplified drug models, incorporate

[1] https://www.who.int/tools/atc-ddd-toolkit/atc-classification

[2] https://github.com/creative-connections/Pharmacolibrary

physiological feedback, and validate clinical case studies to further advance quantitative systems pharmacology.

Conflict of Interest

No conflicts are to be reported.

Acknowledgements

This library was inspired by the PharmaKin Modelica library (MediWare a.s., CZ, co-authored by Jan Šilar) as a part of a joint research grant on computational modeling for optimized pharmacotherapy (Potůček et al. 2016). F.J. has been partially supported by NIH; subaward to R01 HL173346. T.K. was supported by the ONCOSCREEN project, funded by the European Union's Horizon Europe research and innovation programme under grant agreement No. 101097036.

References

Azer, Karim et al. (2021). "History and future perspectives on the discipline of quantitative systems pharmacology modeling and its applications". In: *Frontiers in physiology* 12, p. 637999.

Cleton, A, R A Voskuyl, and M Danhof (1998-10). "Adaptive changes in the pharmacodynamics of midazolam in different experimental models of epilepsy: kindling, cortical stimulation and genetic absence epilepsy: Epilepsy induced reduction in midazolam efficacy". en. In: *Br. J. Pharmacol.* 125.4, pp. 615–620. ISSN: 0007-1188,1476-5381. DOI: 10.1038/sj.bjp.0702088. URL: http://dx.doi.org/10.1038/sj.bjp.0702088.

Fleishaker, J C and R B Smith (1987-12). "Compartmental model analysis in pharmacokinetics". en. In: *J. Clin. Pharmacol.* 27.12, pp. 922–926. ISSN: 0091-2700,1552-4604. DOI: 10.1002/j.1552-4604.1987.tb05591.x. URL: http://dx.doi.org/10.1002/j.1552-4604.1987.tb05591.x.

Fritzson, Peter et al. (2020). "The OpenModelica Integrated Environment for Modeling, Simulation, and Model-Based Development". In: *Modeling, Identification and Control* 41.4, pp. 241–295. DOI: 10.4173/mic.2020.4.1.

Grzegorzewski, Jan et al. (2020-11). "PK-DB: pharmacokinetics database for individualized and stratified computational modeling". In: *Nucleic Acids Research* 49.D1, pp. D1358–D1364. ISSN: 0305-1048. DOI: 10.1093/nar/gkaa990. eprint: https://academic.oup.com/nar/article-pdf/49/D1/D1358/35364573/gkaa990.pdf. URL: https://doi.org/10.1093/nar/gkaa990.

Heizmann, P, M Eckert, and W H Ziegler (1983). "Pharmacokinetics and bioavailability of midazolam in man". en. In: *Br. J. Clin. Pharmacol.* 16 Suppl 1, 43S–49S. ISSN: 0306-5251,1365-2125. DOI: 10.1111/j.1365-2125.1983.tb02270.x. URL: http://dx.doi.org/10.1111/j.1365-2125.1983.tb02270.x.

Hodiamont, Caspar J et al. (2022-08). "Clinical pharmacokinetics of gentamicin in various patient populations and consequences for optimal dosing for Gram-negative infections: An updated review". en. In: *Clin. Pharmacokinet.* 61.8, pp. 1075–1094. ISSN: 0312-5963,1179-1926. DOI: 10.1007/s40262-022-01143-0. URL: http://dx.doi.org/10.1007/s40262-022-01143-0.

Hornestam, Björn et al. (2003). "Intravenously administered digoxin in patients with acute atrial fibrillation: a population pharmacokinetic/pharmacodynamic analysis based on the Digitalis in Acute Atrial Fibrillation trial". In: *European journal of clinical pharmacology* 58.11, pp. 747–755.

Hosseini, Iraj et al. (2018). "gPKPDSim: a SimBiology®-based GUI application for PKPD modeling in drug development". In: *Journal of pharmacokinetics and pharmacodynamics* 45, pp. 259–275.

Jamei, Masoud et al. (2009). "The Simcyp® population-based ADME simulator". In: *Expert opinion on drug metabolism & toxicology* 5.2, pp. 211–223.

Kaneda, Kotaro and Tae-Hyung Han (2009-09). "Comparative population pharmacokinetics of fentanyl using non-linear mixed effect modeling: burns vs. non-burns". en. In: *Burns* 35.6, pp. 790–797. ISSN: 0305-4179,1879-1409. DOI: 10.1016/j.burns.2008.12.006. URL: http://dx.doi.org/10.1016/j.burns.2008.12.006.

Koepsell, Hermann (2013). "The SLC22 family with transporters of organic cations, anions and zwitterions". In: *Molecular Aspects of Medicine* 34.2. The ABCs of membrane transporters in health and disease (SLC series), pp. 413–435. ISSN: 0098-2997. DOI: https://doi.org/10.1016/j.mam.2012.10.010. URL: https://www.sciencedirect.com/science/article/pii/S0098299712001240.

Kulhánek, Tomáš and Filip Ježek (2025-07). *Pharmacolibrary: Modelica library for pharmacology*. Version v25.08-beta2. DOI: 10.5281/zenodo.15322052. URL: https://doi.org/10.5281/zenodo.15322052.

Lippert, Jörg et al. (2019). "Open systems pharmacology community—an open access, open source, open science approach to modeling and simulation in pharmaceutical sciences". In: *CPT: pharmacometrics & systems pharmacology* 8.12, p. 878.

Matejak, Marek et al. (2015). "Free modelica library for chemical and electrochemical processes". In: *11th International Modelica Conference, Versailles, France*. Vol. 118, pp. 359–66.

Mateják, Marek (2023). "Creating cardiovascular and respiratory models using Physiolibrary 3.0". In: *Modelica Conferences*, pp. 463–468.

Mateják, Marek et al. (2014). "Physiolibrary - Modelica Library for Physiology". In: *Proceedings of the 10th International Modelica Conference, March 10-12, 2014, Lund, Sweden*. Linköping University Electronic Press, pp. 499–505. DOI: 10.3384/ecp14096499.

Nilsson, Emma Larsdotter and Peter Fritzson (2005). "A metabolic specialization of a general purpose modelica library for biological and biochemical systems". In: *4th International Modelica Conference*. Citeseer.

Potůček, J et al. (2016). "Use of computer models for optimized pharmacotherapy (in Czech)". In: *MEDSOFT* 28, pp. 120–129. URL: https://www.medsoft.website/sbornik/2016/Medsoft_2016_Potucek.pdf.

Rosenbaum, Sara E (2016). *Basic pharmacokinetics and pharmacodynamics: An integrated textbook and computer simulations*. John Wiley & Sons.

Schön, Kristyna et al. (2022). "Comparison of Mw\Pharm 3.30 and Mw\Pharm++, a Windows version of pharmacokinetic software for PK/PD monitoring of vancomycin. Part 1: A-posteriori modelling". In: *Computer Methods and Programs in Biomedicine* 214, p. 106552.

Van Meurs, Willem (2011). *Modeling and simulation in biomedical engineering: applications in cardiorespiratory physiology*. McGraw-Hill Education.