How to Make a Salad? Rethinking Pharmacometric/QSP Model Composition Using Open-Source Julia Tools

Ahmed Elmokadem, Ph.D, Ellen Swanson, Ph.D, Jimena Davis, Ph.D, Timothy Knab, Ph.D, and Daniel Kirouac, Ph.D
Metrum Research Group, Boston, MA, USA

Abstract

Objectives: Pharmacometric and systems pharmacology models are often modular as different, independent components can be joined together to form a more complex model. The process of combining and reusing model components can be challenging with no clear framework and, as such, investigators often resort to rewriting models from scratch rather than reusing the individual components. Additionally, model components could be written in different notations such as ordinary differential equations (ODEs) or reactions, depending on the most convenient way to represent a system. This adds an additional complexity to the model composition process. A model salad framework is presented that allows an investigator to seamlessly combine different model components represented in their respective notations and reuse these independent components to create multiple combinations of integrated models, just like mixing the components of a salad.
Methods: Julia [1] open-source tools, namely ModelingToolkit.jl [2] and Catalyst.jl [3], were used to present a convenient framework for pharmacometric model composition. The symbolic-numeric model representation of ModelingToolkit.jl and the reaction notation provided by Catalyst.jl allowed for seamless composition of independent model components presented as ODEs or reactions.
Results: The framework was demonstrated by composing different model components (e.g., pharmacokinetic (PK), pharmacodynamic (PD), physiological organs) to build larger models (e.g., PKPD, physiologically-based PK (PBPK), quantitative systems pharmacology (QSP)). Both ODEs and reaction notations were combined into integrated PKPD and QSP models with examples drawn from bispecific T-cell engagers, viral dynamics, and drug-drug interactions (DDI). The framework enabled seamless transitions from *in vitro* to *in vivo* murine to clinical settings for a bispecific T cell engager application [4].
Conclusions: A framework based on Julia open-source tools was proposed in this work to allow for seamless pharmacometric and QSP model composition. This framework enables model reusability and translation using convenient and flexible model notation.

Methods

The model salad framework demonstrated in Figure 1 composes different independent model components to create more complex models in a seamless is based on Julia tools withing the SciML ecosystem:

- **ModelingToolkit.jl**. This package allows for the symbolic-numeric model representation.
- **Catalyst.jl**. This package allows for representing models using reaction notation.

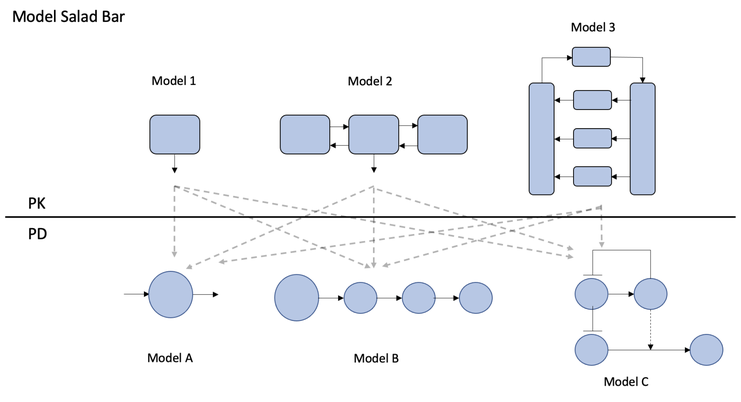


Figure 1. Model Salad Bar. Different models can be composed together to create more complex models using Julia tools. PK represents pharmacokinetic models and PD represents pharmacodynamic models.

A simple PKPD example

A simple PKPD was used to demonstrate the model salad framework. The PK model was a one-compartment model with an IV bolus dose that characterized the change in drug concentration C and the PD model was an indirect response model for a response R whose induction gets inhibited by the drug (Figure 2). The equations to describe the model were:

$$\frac{dC}{dt} = -(\frac{CL}{V}) * C$$
$$\frac{dR}{dt} = kin * I(C) - kout * R$$
$$I(C) = 1 - \frac{C}{EC50 + C}$$

where CL = drug clearance, V = drug volume of distribution, kin = response synthesis rate, $kout$ = response degradation rate, and $EC50$ is the drug concentration required to reach half-maximal response inhibition.

Figure 2. Simple PKPD Model Structure. Different models can be composed together to create more complex models using Julia tools. PK represents pharmacokinetic models and PD represents pharmacodynamic models.

Model building

The PK and PD models were built using ModelingToolkit.jl where a function was created to include the model components defined as blocks (parameters, variables, and equations) and returned an ODE system. The model objects were created by calling the function. Finally, the PK and PD models were combined using the convenient function "extend". The algebraic model structure was simplified using the function "structural_simplify". The following code demonstrates the steps to build the PK model and how the PK and PD models were combined together.

```
function PK(; name)
    pars = @parameters begin
        CL=0.3
        V=1.0
    end

    @independent_variables t
    D = Differential(t)

    vars = @variables begin
        C(t)=10.0
    end

    eqs = [D(C) ~ -(CL/V) * C]

    ODESystem(eqs, t, vars, pars; name=name)
end

@mtkbuild pk = PK()
```

```
function PD(; name)
    pars = @parameters begin
        R0=5.0
        kout=1.0
        IC50=5.0
    end

    @independent_variables t
    D = Differential(t)

    vars = @variables begin
        C(t)=0.0
        R(t)=R0/kout
    end

    kin = R0 / kout
    I = 1 - (C / (C + IC50))

    eqs = [D(R) ~ kin*I - kout*R]

    ODESystem(eqs, t, vars, pars; name=name)
end
```

```
pkpd = structural_simplify(extend(pk,pd))
```

Results

The model salad framework was demonstrated through a number of applications. A generic integrated model including viral kinetics, drug PK, and immune response components was created (Figure 3), and the model predictions showed the decrease in viral load as a response to the drug and immune response effects (Figure 4). The framework's flexibility allows for the creation of other models by exchanging the different model components with alternative viral kinetics models, different PK models for different drugs, and different models for the immune response. A generic PBPK DDI model was created by combining the PBPK models of the victim and perpetrator drugs (Figure 5). The victim drug exposure was simulated showing an increased exposure when co-administered with the perpetrator that inhibits the victim drug metabolism (Figure 6). The framework allows for exchanging different victim and different perpetrator drug PBPK models. The proposed framework flexibility also allows for combining ODEs and reactions, which was demonstrated using a generic PKPD model of a monoclonal antibody (mAb) binding to a soluble receptor where the mAb PK was represented as ODEs while the PD binding component was represented as reactions (Figure 7). mAb and receptor conentration-time profiles demonstrated the expected behavior (Figure 8). A published model of a bispecific mAb that targets CD3 receptors on T-cells and P-cadherin on tumor cells was used to demonstrate the model salad framework [4]. The independent model components were a tumor growth model, a T-cell dynamics model, a tumor microenvironment (TME) binding model, and a drug PK model. The different components were combined to create different models to accommodate for the different settings. As such, the tumor growth and TME models were combined to create the *in vitro* model, the tumor growth, TME, T-cell dynamics, and the drug PK models were combined to create the *in vivo* mouse model, and the tumor growth, TME, and drug PK models were combined to create the human model (Figure 9). Drug PK prediction at different doses (0.5 and 0.05 mg/kg) and validation against observed data from the *in vivo* model and tumor trimer concentration from the human model at different doses (0.01, 0.1, and 1 ug/kg) were shown in Figure 10.

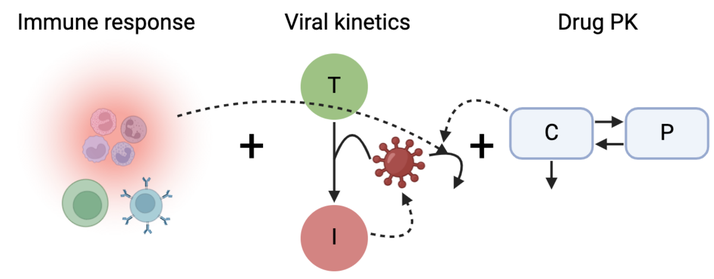


Figure 3. Viral Dynamics Model Salad. A viral kinetics, a drug PK, and an immune response models were combined to create an integrated model with all components. T and I represent the target and infected cell populations, respectively. C and P represent the drug concentrations in the central and peripheral compartments, respectively.

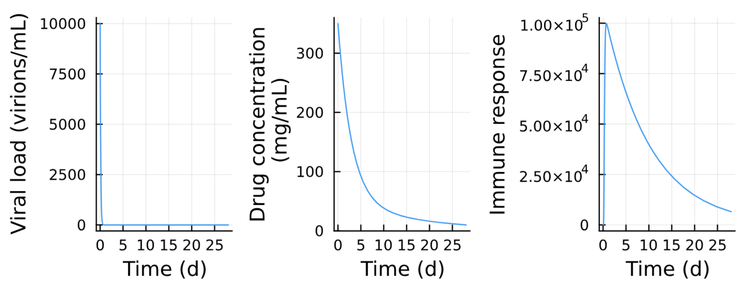


Figure 4. Integrated Viral Dynamics Model Simulation Results. The integrated model simulation results showing the dynamics of the viral load (left), the drug concentration in the central compartment (middle), and the immune response (right).

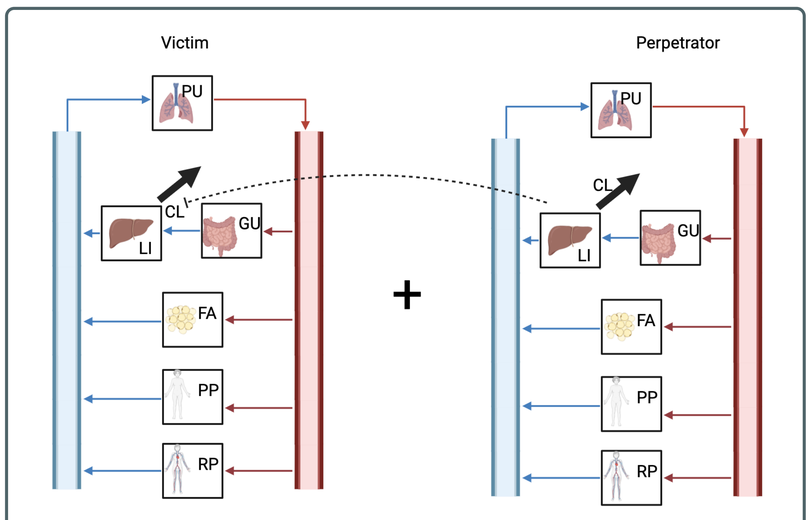


Figure 5. PBPK DDI Model Salad. Independent PBPK models for the victim and perpetrator drugs were combined to create a DDI model where the perpetrator drug has an inhibitory effect on the victim drug metabolism. PU, LI, GU, FA, PP, and RP represent the lungs, liver, gut, fat, poorly perfused, and richly perfused tissues, respectively. CL represents clearance.

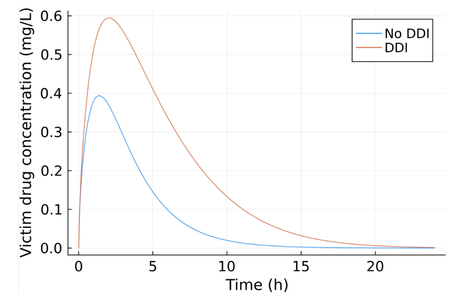
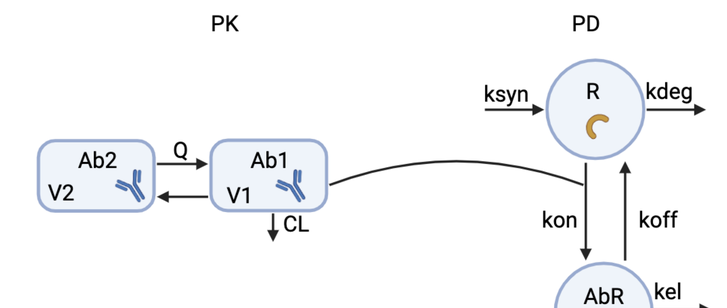


Figure 6. PBPK DDI Model Simulaton Result. Victim drug concentration-time profile when administered alone (blue - no DDI) or with the perpetrator drug (red - DDI). The figure shows the DDI effect that results in a higher victim drug exposure.



$$\frac{\mathrm{dAb1}(t)}{dt} = \frac{-Q\mathrm{Ab1}(t)}{V1} + \frac{Q\mathrm{Ab2}(t)}{V2} - \mathrm{Ab1}(t)\frac{CL}{V1}$$
$$\frac{\mathrm{dAb2}(t)}{dt} = \frac{Q\mathrm{Ab1}(t)}{V1} + \frac{-Q\mathrm{Ab2}(t)}{V2}$$

$$\varnothing \underset{kdeg}{\overset{ksyn}{\rightleftharpoons}} \mathrm{R}$$
$$\mathrm{Ab1} + \mathrm{R} \underset{koff}{\overset{kon}{\rightleftharpoons}} \mathrm{AbR}$$
$$\mathrm{AbR} \xrightarrow{kel} \varnothing$$

Figure 7. Monoclonal Antibody PKPD Model Salad. Generic monoclonal antibody (mAb) PK and PD model components were combined to create an integrated PKPD model. The PK model is a two-compartment mAb model that was represented as ODEs and the PD model is mass action binding kinetics of the mAb to a soluble receptor represented as reactions.

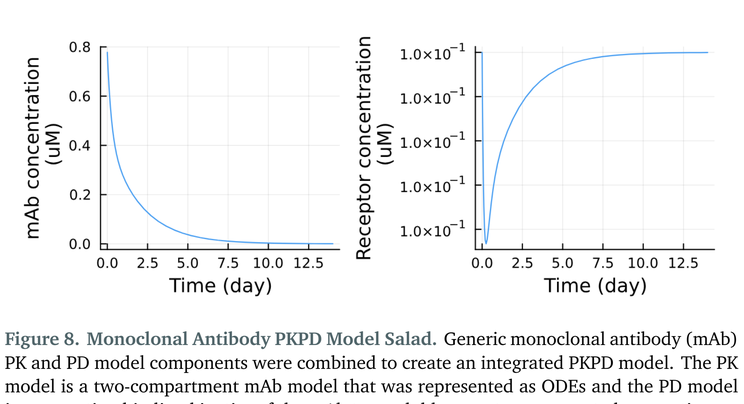


Figure 8. Monoclonal Antibody PKPD Model Salad. Generic monoclonal antibody (mAb) PK and PD model components were combined to create an integrated PKPD model. The PK model is a two-compartment mAb model that was represented as ODEs and the PD model is mass action binding kinetics of the mAb to a soluble receptor represented as reactions.

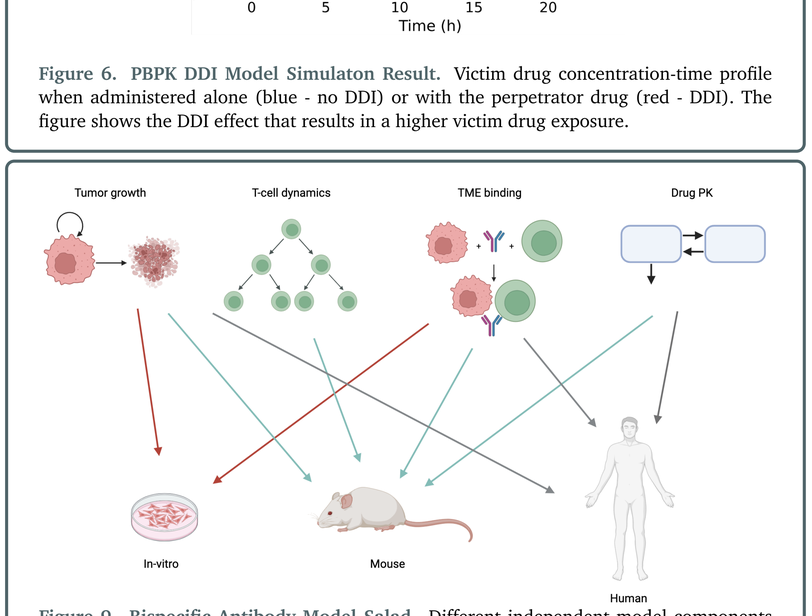


Figure 9. Bispecific Antibody Model Salad. Different independent model components were composed to build the *in vitro* model (Tumor growth + Tumor microenvironment (TME) binding reactions), *in vivo* mouse model (Tumor growth + T-cell dynamics + TME binding + Drug PK), and human model (Tumor growth + TME binding + Drug PK).

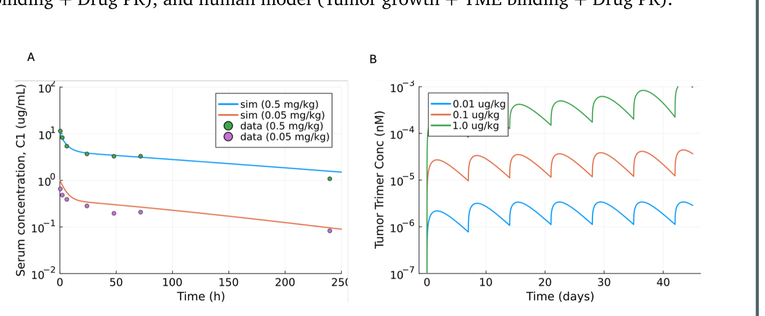


Figure 10. Bispecific Antibody Model Simulation Results. The simulation results from the different created models show the *in vivo* mouse model predicted drug concentration-time profiles for 0.5 and 0.05 mg/kg doses (lines) and validated against observed data (points) (A) and tumor trimer predicted concentrations in humans for 0.01, 0.1, and 1 ug/kg doses (B).

Conclusion

This work demonstrated a framework that utilized open-source Julia tools (ModelingToolkit.jl and Catalyst.jl) to integrate independent models into more complex models in a seamless way and to combine models described as differential equations and reactions. The convenience and flexibility of the proposed framework allows investigators to build complex pharmacometric and QSP models from simple components, reduces the errors that may result from copying code, and minstreams the quality control process.

References

[1] Bezanson, J., Edelman, A., Karpinski, S. and Shah, V.B. Julia: A Fresh Approach to Numerical Computing. *SIAM Rev.* **59** (2017):65–98.

[2] Ma, Y., Gowda, S., Anantharaman, R., Laughman, C., Shah, V. and Rackauckas, C. ModelingToolkit: A Composable Graph Transformation System For Equation-Based Modeling (2021).

[3] Loman, T.E., Ma, Y., Ilin, V., Gowda, S., Korsbo, N., Yewale, N., Rackauckas, C. and Isaacson, S.A. Catalyst: Fast and flexible modeling of reaction networks. *PLoS Comput. Biol.* **19** (2023):e1011530.

[4] Betts, A., Haddish-Berhane, N., Shah, D.K., van der Graaf, P.H., Barletta, F., King, L., Clark, T., Kamperschroer, C., Root, A., Hooper, A. and Chen, X. A translational quantitative systems pharmacology model for CD3 bispecific molecules: Application to quantify T cell-mediated tumor cell killing by P-cadherin LP DART®. *AAPS J.* **21** (2019):66.

Metrum Research Group Publications and Posters