Simulating Adaptive Dosing Regimens from PK and PKPD Models Using mrgsolve

Kyle T. Baron
Metrum Research Group, Boston, MA, USA

Objective

Adaptive or dynamic dosing in pharmacokinetic (PK) or pharmacokinetic-pharmacodynamic (PKPD) models allows for changes in the dose regimen in response to changes in model state variables over the course of a simulation [1]. In contrast to fixed dosing with all doses specified in the data set prior to simulating, adaptive dosing requires special handling to update the regimen at various points after the simulation starts. A common way to handle this problem is to interrupt the simulation at milestone observation times to assess the current state and history, decide what changes to the regimen are needed, implement those changes, and restart the simulation. This approach can involve additional complexity for tracking the simulation history and can be computationally inefficient. The objective of this work was to address these limitations with an extension to the mrgsolve package for R allowing the dose regimen to be completely specified in the model code.

Methods

mrgsolve is a freely available open-source R package, available on GitHub, for simulating from compartmental PK and PKPD models commonly used in pharmacometrics [2]. The package was developed and is maintained using a transparent, open-source software development life cycle (SDLC) process, incorporating iterative development, validation, and release. Development of the adaptive dosing extension was based on experience implementing these simulations over the course of several different modeling and simulation projects. The initial release focused on a limited feature set to help identify a stable, extensible simulation API. Unit tests for each feature were developed and the code was reviewed according to the SDLC process. Features were documented in the mrgsolve user guide [3].

Results

The adaptive dosing extension was released in February 2024 (mrgsolve v1.4.1) as a plugin called evtools. When invoked, functions and objects to support adaptive dosing are made available under the evt namespace. Dosing information is specified by the user in the model code and sent back to mrgsolve package code through a C++ `struct` with type `ev`. The initial feature set included functionality to execute single doses (bolus or infusion) at arbitrary times during the simulation or as a regimen following a regular dosing interval. Single-dose objects are executable at the time of creation or retimed to happen later in the simulation. Repeated dosing was implemented through a regimen object with attributes for dose amount (`amt`), compartment (`cmt`), interval (`ii`), administration rate (`rate`), and termination time (`until`). A method (`flagnext`) was also included to ensure the simulation explicitly runs through each dose time in case the observation schedule is sparse.

But wait, there's more!

mrgsolve version 1.5.2 was recently released with additional functionality.

- Reset the ODE system
- Replace amount in a specific compartment
- Schedule additional doses
- Update compartment number, amount, infusion rate
- Give doses after advancing to steady state
- Example code via `modlib()` internal model library

Case study: platelet dynamics in NHL

A case study is presented using a previously-published PKPD model of platelet dynamics under once-daily dosing with valemetostat in patients with non-Hodgkin lymphoma (NHL) [4,5]. Both total and unbound valemetostat concentrations were modeled simultaneously with three-compartment disposition. An integrated model for simulation was created, joining the valemetostat population PK and a population PD model describing platelet dynamics, where unbound valemetostat was modeled to inhibit proliferation of platelet precursor cells (Figure 1). In the dynamic dosing simulation (Figure 3), all subjects started on 200 mg once-daily (QD). After the first Grade 4 thrombocytopenia (TCP) event (platelets < 25×10^9/L), the dose was held and restarted when platelets recovered to 50×10^9/L. The dose was reduced to 150 mg and 100 mg QD after recovery from the second and third Grade 4 TCP events, respectively. Valemetostat was discontinued after the fourth Grade 4 TCP event.

Conclusion

The evtools plugin to mrgsolve is a powerful tool for executing simulations involving adaptive or dynamic dosing. Because the dose regimen is managed from within the model itself, simulations remain computationally efficient.

Open access

- mrgsolve is on GitHub at
  - http://github.com/metrumresearchgroup/mrgsolve
- All code for the case study and an `evtools` vignette is available at
  - http://github.com/mrgsolve/dynamic-dosing

References

1. Hooijmaijers, R. et al. Building an adaptive dose simulation framework to aid dose and schedule selection. CPT Pharmacometrics Syst Pharmacol (2023).
2. https://github.com/metrumresearchgroup/mrgsolve
3. https://mrgsolve.org/user-guide/
4. Fukae, M., Baron, K., Tachibana, M., Mondick, J. & Shimizu, T. Population pharmacokinetics of total and unbound valemetostat and platelet dynamics in healthy volunteers and patients with non-Hodgkin lymphoma. CPT Pharmacometrics Syst. Pharmacol. (2024).
5. Fukae, M., et al. Landmark and longitudinal exposure-response analyses for multiple efficacy and safety endpoints to justify the clinical dose of valemetostat for adult T-cell leukemia/lymphoma. ACoP13 (2022) PMX-493 https://www.go-acop.org/default.asp?abstract=493

Metrum Research Group Publications and Posters

Case study results: platelet dynamics in NHL

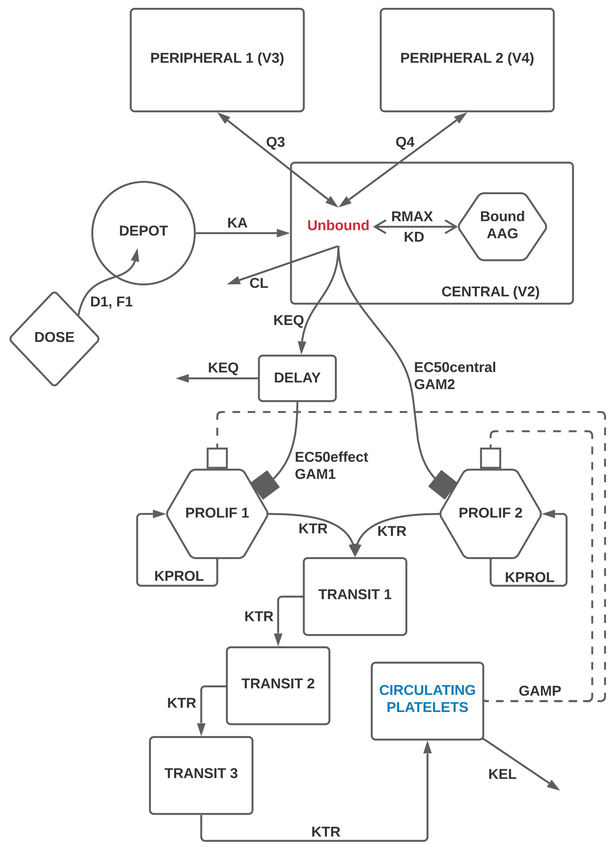

Figure 1. PKPD model schematic. Valemetostat unbound concentration inhibits proliferation of platelet precursors [4].

```
mod <- mread("platelet.mod", end = 1680, delta = 24)

idata <- expand.idata(ID = 1:30, ADJUST = 0)

out <- mrgsim_i(mod, idata, tscale = 1/168)
```

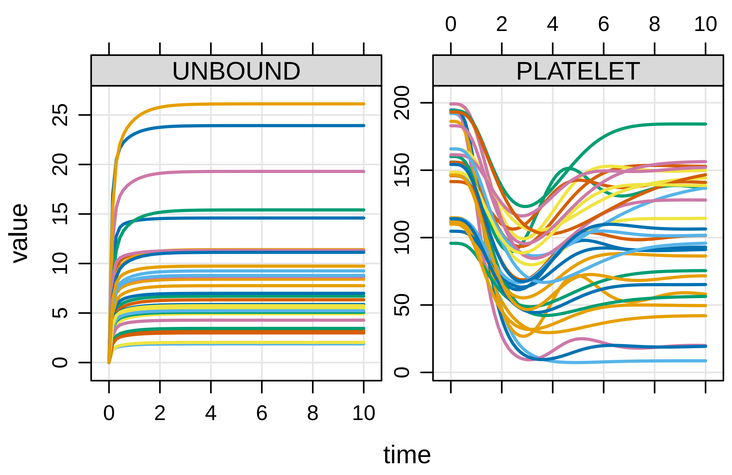

Figure 2. Example simulation. Valemetostat unbound concentration (left) and platelets (right) simulated over 10 weeks with no dose adjustments. Only the trough concentration is shown here for clarity.

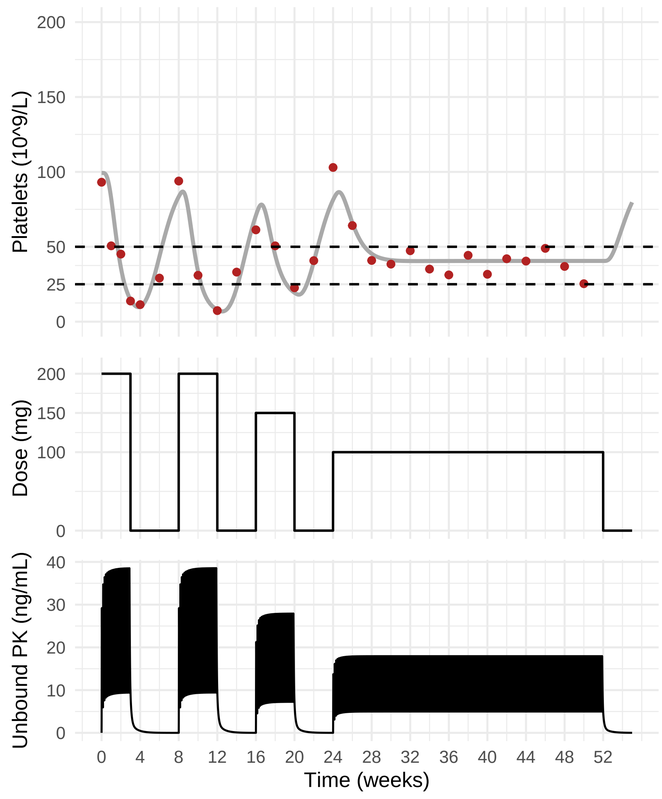

Figure 3. 52-week simulation with adaptive dosing. A single subject is shown. Top: platelets; red points indicate observations at clinic visits; solid gray line is the individual predicted value. Middle: valemetostat dose. Bottom: valemetostat unbound concentration.

Dynamic dosing syntax

Bolus and infusion doses

```
$PLUGIN evtools

$EVENT
evt::bolus(self, 100, 2);
evt::ev dose = bolus(100, 2);

evt::infuse(self, 100, 2, 50);
evt::ev dose = infuse(100, 2, 50);
```

Remember to use the plugin in your code. We also show use of the `$EVENT` block for writing code to execute modeled dosing or other events. Code in `$EVENT` happens right before `$ERROR` is called.

Work with event object

Once you've created an object (above), you can customize it with these functions. Note: there is no return value.

```
evt::amt(dose, 200);
evt::rate(dose, 100);
evt::cmt(dose, 2);
evt::ii(dose, 24);
evt::addl(dose, 9);
evt::ss(dose, 1);
evt::retime(dose, 72);
evt::now(dose);
```

Send event back to mrgsolve; these are equivalent.

```
evt::push(self, dose);
self.push(dose);
```

Reset and replace

Reset all compartments to modeled initial conditions.

```
evt::reset(self);
```

Reset and (bolus) dose. You can also reset and start an infusion.

```
evt::reset(self, 100, 1);
```

Reset compartment 1 with 0 and compartment 2 with 1.

```
evt::replace(self, 1, 0);
evt::replace(self, 2, 1);
```

In contrast to `evt::reset()`, `evt::replace()` can be any value at any time.

Dose regimen

The dose, rate, and interval in a regular dosing regimen can all be set in your model code and updated as the simulation progresses. Values were hard-coded in this example, but will likely be set through parameters or variables in production code.

```
$GLOBAL
evt::regimen reg;

$PK
if(NEWIND <= 1) reg.init(self);

$EVENT
reg.amt(100);
reg.cmt(2);
reg.ii(24);
reg.until(168);

reg.execute();
```

Choose a time grid that ensures the simulation passes through each dose time.