

The optiDoseR package – facilitating application of OptiDose in R, NONMEM and Monolix

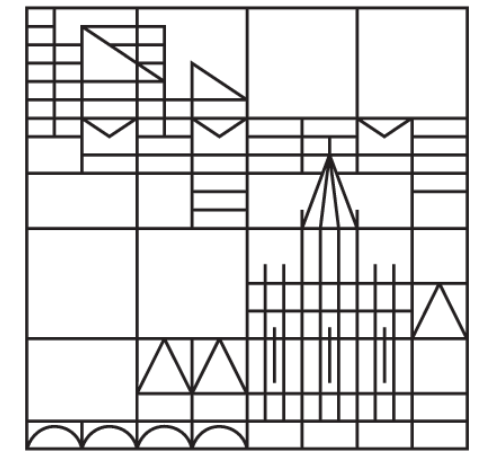
Freya Bachmann^{1,*}, Britta Steffens², Johannes Schropp¹
Marc Pfister², Gilbert Koch² and Dominic Bräm²

(1) Department of Mathematics and Statistics, University of Konstanz, Germany

(2) Pediatric Pharmacology and Pharmacometrics, University of Basel Children's Hospital (UKBB), Switzerland

(*) Corresponding author email: freya.bachmann@uni-konstanz.de

Universität
Konstanz



BRC
CH Basel Research
Centre for
Child Health

UKBB
kompetent & menschlich

Introduction

OptiDose [1, 2] enables users to solve their own optimal dosing tasks for their pharmacometric (PMX) models. Optimality is characterized by achieving a therapeutic goal specified by a reference function describing a desired outcome, or both efficacy and safety targets.

Defining an optimal dosing task requires

- a PMX model predicting the disease dynamics
- a dosing schedule with doses or dosing intervals to be optimized
- an objective function characterizing the therapeutic goal

While the first two are standard in PMX modeling, the third might be more challenging and eventually keep users from applying OptiDose.

Objectives:

- provide an R package “**optiDoseR**” [3] for automatic generation of code to solve an optimal dosing task in either NONMEM, Monolix or R via nlmixr2
- facilitating application of OptiDose to advance from modeling-and-simulation towards **modeling-and-optimization**

Methods

We illustrate the procedure alongside the warfarin PKPD example.

Step 1: install optiDoseR

Download from [3], or install via

```
devtools::install_github("braemd/optiDoseR")
```

Step 2: initialize new project

Load required libraries and define output language (Monolix, NONMEM or R) for new OptiProject

```
library(optiDoseR, [...])  
warf_proj <- newOptiProject("nlmixr2")
```

Step 3: define PMX model

Add previously performed nonlinear mixed effects run via

```
pk.turnover <- [...] # structural model  
fit <- nlmixr2(pk.turnover, warfarin, "saem")  
warf_proj <- addPmxRun(warf_proj, fit)
```

Otherwise, manually specify structural model, population and individual model parameters via addPmxModel, addPopParms, and addIndParms.

Step 4: define dosing schedule

Specify dosing schedule for doses to be optimized. For each dose indicate first dosing time point, number of repeated dose applications, and initial estimate of the dose, e.g.,

```
addDoseLevel1(time=0, n_dose=1, ini_est=100)
```

Similarly, dosing intervals for repeated application of a fixed dose can be optimized via addDosingInterval.

Step 5: define therapeutic goal

Therapeutic goals can be defined at a specific time point or for a time interval. Either a desired outcome is achieved as closely as possible, or both efficacy and safety targets are considered, such as minimizing or maximizing a certain state while this or another state is required to remain above or below a certain threshold.

Here, the therapeutic goal is to achieve, e.g., a warfarin exposure as high as possible without platula coagulation activity (PCA), the effect compartment, falling below the limit of 30. Hence, we have a constraint $PCA(t) \geq 30$ for all times $t \in [0, 200]$, and a secondary target of maximizing the area under the curve of the warfarin concentration C_p , i.e.,

$$\max \int_0^{200} C_p(t) dt \quad \text{subject to} \quad PCA(t) \geq 30, t \in [0, 200]$$

This is formulated in optiDoseR via

```
addConstraintAUC(state="effect", limit=30, pen_time  
=c(0,200), pen_values="below") %>%  
addSecondaryAUC(state="cp", pen_values="low", pen_  
time=c(0,200))
```

Step 6: save OptiProject & run

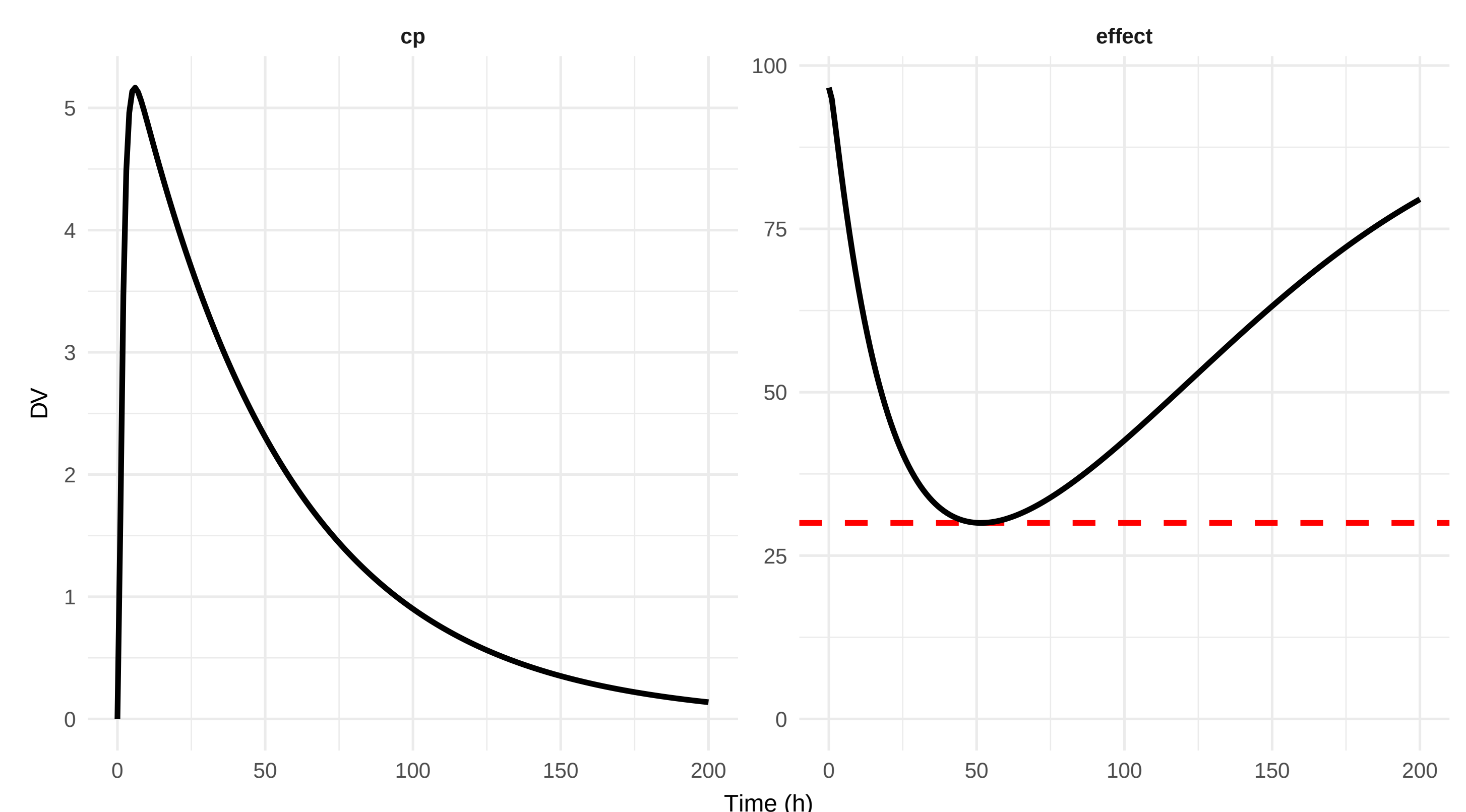
Automatically generate necessary code to solve the optimal dosing task in the desired software

```
saveOptiProject(warf_proj, [...])
```

Option pop=FALSE considers the typical representative of the population only, whereas pop=TRUE optimizes doses for all individuals in the population. For an nlmixr2-OptiProject, option run_nlmixr2=TRUE directly runs the code in R.

Results

For the typical individual, an optimal dose of 42.3 mg is required to achieve a maximal warfarin exposure while keeping PCA above 30 at all times.



Conclusion

Presented R package contains all functions required to generate code for solving optimal dosing tasks in NONMEM, Monolix or R. Optimal dosing can be computed for a particular individual or all individuals in a population. optiDoseR facilitates application of OptiDose allowing to advance towards modeling-and-optimization.

References:

- [1] Bachmann F, Koch G, Bauer RJ, Steffens B, Szinnai G, Pfister M, Schropp J (2023) Computing optimal drug dosing with OptiDose. Implementation in NONMEM. J Pharmacokinet Pharmacodyn 50(3):173-188
- [2] Bachmann F, Koch G, Bauer RJ, Steffens B, Szinnai G, Pfister M, Schropp J (2024) Computing optimal drug dosing regarding efficacy and safety: the enhanced OptiDose method in NONMEM. J Pharmacokinet Pharmacodyn 51(6):919-934
- [3] Bräm D (2025) R package optiDoseR. <https://github.com/braemd/optiDoseR>