Building model-informed precision dosing software using R
blueprint for a state-of-the-art development process

Ruben Faelens(1), Nicolas Luyckx(2,4current), Quentin Leirens(2,5current), Dirk Kuypers(3), Pieter Annaert(1)

(1) *Drug Delivery and Disposition, Department of Pharmaceutical and Pharmacological Sciences, KU Leuven – University of Leuven, Leuven, Belgium (2) SGS Exprimo (3) Department of Nephrology, University Hospital Leuven, Belgium (4) Calvagone SARL (5) Pharmetheus AB*

THE CHALLENGE

Empirical Bayesian Estimation is easy

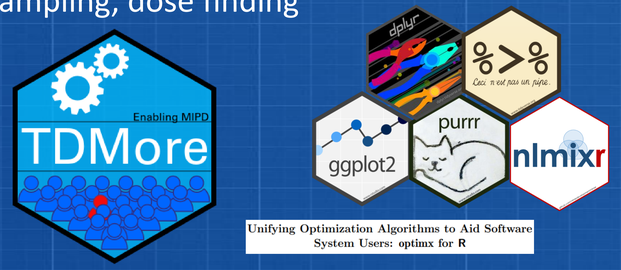
$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

$$P(\eta|Y) = \prod \Phi(\vec{\eta}, \Theta, \Omega) \times \prod \Phi(f(\vec{\eta}, t_i), Y_i, \Sigma)$$

But how do you build a stable **software?**

IMPLEMENT MATHS: TDMORE

Implement ODE solving, define model, calculate loglikelihood, find most likely η, MCMC sampling, dose finding



Unifying Optimization Algorithms to Aid Software System Users: optimx for R

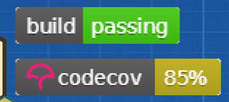
Tdmore offers all tools to implement precision dosing. It also allows investigating MIPD *in silico* to ***predict*** a potential benefit in dosing accuracy.

github.com/tdmore-dev/tdmore

tdmore-dev.github.io/tdmore/dev

```
devtools::install_github("tdmore-dev/tdmore")
```

...AS A VALIDATED PACKAGE



Tests and documentation are crucial

INTERFACE COMPONENTS

Ready-to-use building blocks for precision dosing software:

`dosingTable, covariatesTable, parameterPlot, recommendationPlot`

with automated testing

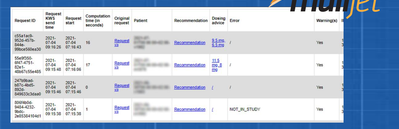
BUILD INTEGRATION WITH EHR SYSTEM

Data transfer through REST API

DEPLOYMENT

Build a docker image passing tests and deploy automatically on Google Cloud

FOLLOW-UP



Get overview of all recent dosing advice, with interface to *dig deeper*.

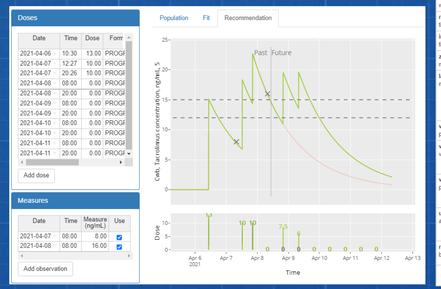
GET CE APPROVAL (?)

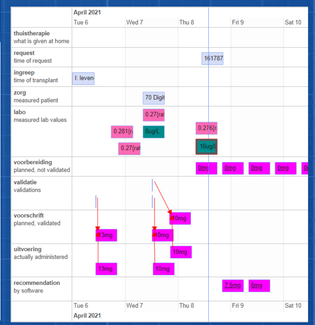
RESULTS

tdmore – core component, simulate MIPD performance a priori, quantify benefit, design trial

shinytdmore – building blocks for MIPD user interface

tacrolimuskws – tacrolimus PK model, shiny app, database, REST API, clinical business rules





Key learnings

- Math component needs to be **trustworthy**. Test own code *and* dependent packages. Automated tests will catch bugs from upstream as well.
- R is not designed for these types of applications. Expect a bumpy ride.
- CE approval assumes a commercial manufacturer.

Correspondence: ruben.faelens@kuleuven.be