New open source R libraries for simulation & visualization:

PKPDsim and vpc

Ron Keizer1,2, Devin Pastoor4, Rada Savic1,2

1 University of California San Francisco 2 InsightRx 3 Pirana Software & Consulting BV 4 University of Maryland, Baltimore

PKPDsim

The PKPDsim R library offers functionality for model simulation and exploration, similar to Berkeley Madonna but within the R environment so that the user can take advantage of R’s powerful statistics and visualization tools. The library facilitates simulation of dosing regimens for PKPD mixed-effects models, leveraging the fast Boost C++ library (or optionally using R:deSolve) for numerical integration. The PKPDsim library generates ggplot-ready data and can facilitate simulation of dosing regimens for PKPD mixed-effects models, taking advantage of R’s powerful statistics and visualization tools. The library can be used from the R command line but can also dynamically generate Shiny frontends to allow interactive use for model exploration and teaching purposes.

Example 1. Simulate using PK model from builtin library:

```r
P <- list(CL = 1, V = 10, KA = 0.5)
pk1 <- new_ode_model("pk_lcmoral")
rt1 <- new_regimen(am=100, times=c(0, 12, 24, 36))
dat <- sim_ode (ode = "pk1", par = P, n_ind = 40, omega = c(0.1, 0.05, 0.1), regimen = rt1)
```

Example 2. Define custom model (PKPD):

```r
pkpd_model <- new_ode_model(
  code = "
  conc = A[1]/V
  ",
)
```

Example 3. Generate Shiny app ‘on-the-fly’

```r
sim_ode_shiny(ode = "pkpd", par = P, regimen = rt1, omega = omega)
```

vpc

Visual Predictive Checks are an essential part of pharmacometric workflows. Most modelers will be familiar with the functionality offered by e.g. PsN, Xpose or Monolix. These tools are somewhat inflexible, however, as they are limited to one specific simulation software and produce plots that are not easily tweakable and/or extensible. This new R library vpc includes:

- plots for continuous (vpc), categorical (vpc_cat), censored continuous (vpc_cens), and survival data (vpc_tte)
- prediction-correction (pred_corr=TRUE argument)
- Kaplan-Meier Mean Covariate plots (kmic argument to vpc_tte)
- npde plots with prediction intervals (npde=TRUE argument)

Example 1. Default vpc, using R data.frames or PsN folder as source data

```r
vpc(sim = sim, obs = obs)
vpc(psn_folder = ‘vpc-dir1’)```

Example 2. vpc with more extensive optional arguments

```r
vpc(sim = sim, obs = obs, obs_cols = list(dv = "dv", idv = "time"), sim_cols = list(dv = "sdv", idv = "time"), bins = "kmens", show = list(obs_dv = TRUE, pl_ci = TRUE), stratify = c("sex"), pi = c(0.05, 0.95), ct = c(0.05, 0.95), pred_corr = FALSE)
```

Example 3. vpc for survival-type data, with explicit bin method selection

```r
vpc_tte(sim = rtte_sim_nm, obs = rtte_obs_nm, stratify = c("sex","drug"), rtte = FALSE, bins = "kmens", n_bins = 20, smooth = TRUE)
```

Example 4. vpc for categorical data, with explicit column selection

```r
vpc_cat (sim = sim_cat, obs = obs_cat, obs_cols = list(dv = "SMXH"), sim_cols = list(dv = "SMXH"))```

Example 5. VPCs for continuous and survival data with default layout

![VPCs](image)

Installation: The PKPDsim and vpc libraries are currently only available from Github, but will be released to CRAN as soon as possible. For more information about their installation and usage please see http://ronkeizer.github.io/PKPDsim and http://ronkeizer.github.io/vpc.

References: