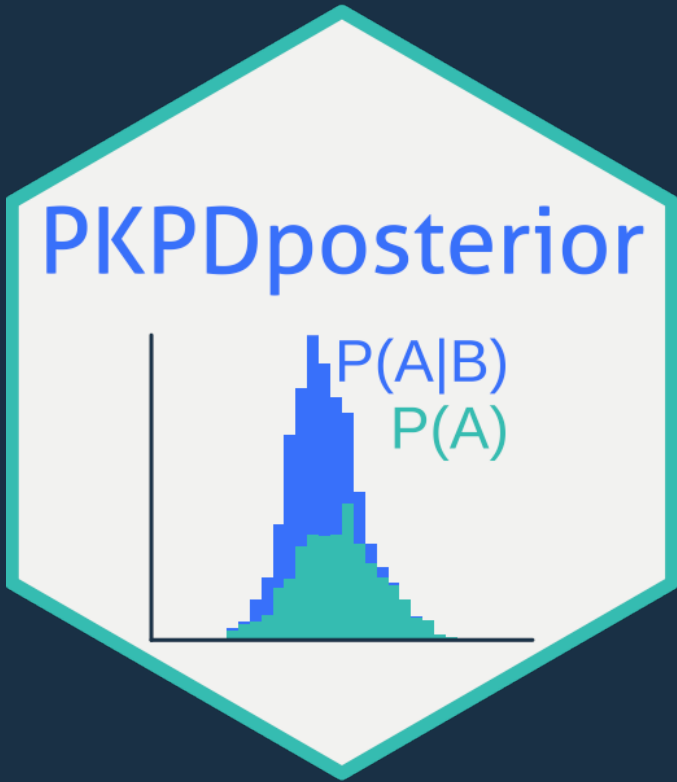




PKPDposterior: R package for structured full-Bayesian model-informed precision dosing



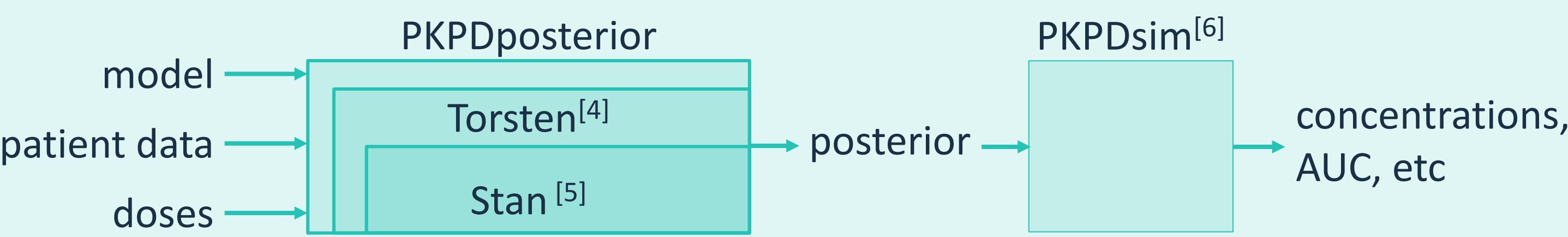
Jasmine H. Hughes^{1,2}, Kara H. Woo¹, Dominic M.H. Tong¹, Ron J. Keizer¹
1. InsightRX, San Francisco, CA
2. jasmine@insight-rx.com, @jas_hughes

Background

- MAP Bayesian methods for MIPD may not accurately estimate uncertainty [1-3]
- Existing full Bayesian approaches are geared towards population-level analysis, high flexibility comes with high complexity for incorporation in point-of-care software

Design Philosophy

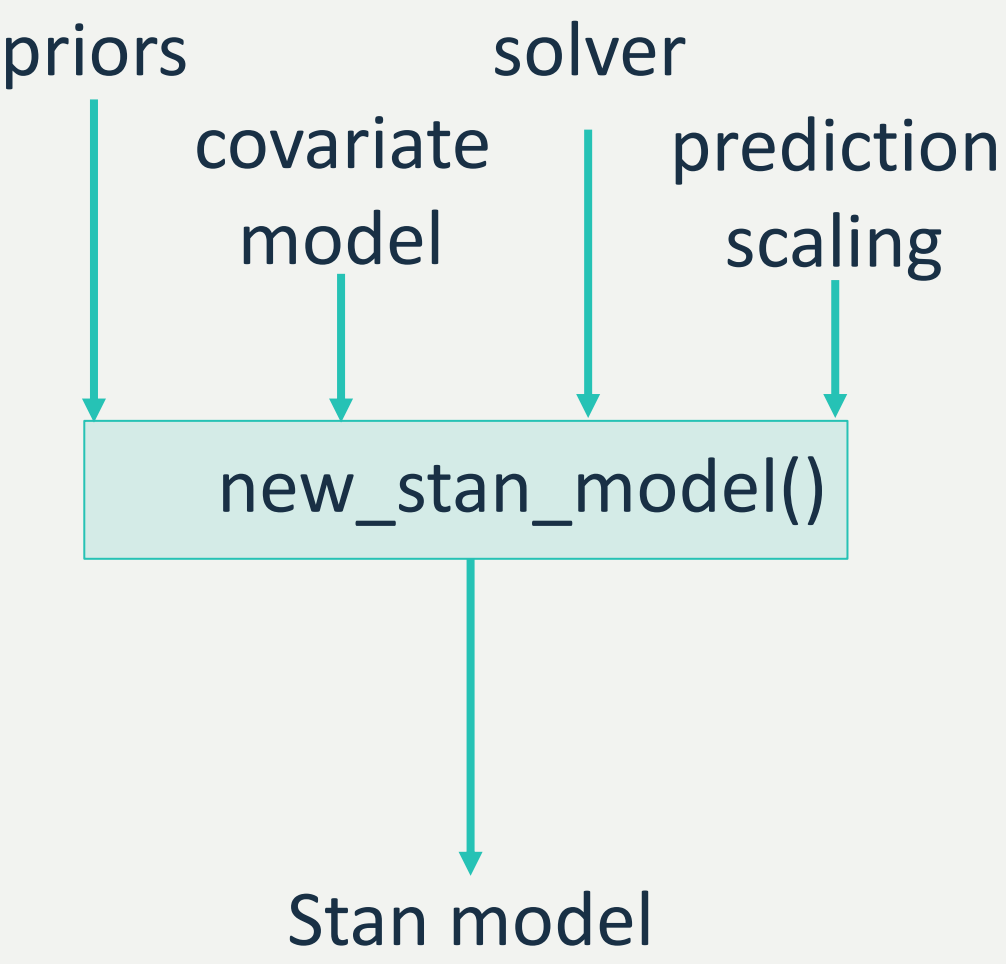
1. Point-of-care oriented APIs that work with existing R tools
2. Implement new models without expert knowledge of Stan
3. Easily compare models and posteriors
4. Rely only on open-source/free tools



Defining a new PK/PK model

1. Pass model structural information to R function as strings encourages standardized model inputs and outputs

```
model <- new_stan_model(
  parameters = list(
    CL = 2.99,
    Q = 2.28,
    V2 = 0.732,
    V1 = 0.675
  ),
  parameter_definitions = list(
    "CL" = "CL*(1+0.0154*(CRCL[j]-66))",
    "Q" = "Q",
    "V1" = "V1 * WT[j]",
    "V2" = "V2 * WT[j]",
    "KA" = "0"
  ),
  covariate_definitions = list(
    "CRCL" = "real",
    "WT" = "real"
  ),
  solver = "pmx_solve_twocpt",
  scale = "(V1 * mean(WT))"
)
```



2. export .stan model

```
modfile <- write_stan_model(
  model,
  "new_model.stan"
)
```

Generated Stan code:

```
transformed parameters{
  row_vector<lower = 0>[n_t] ipred_pk;
  vector<lower = 0>[n_obs_pk] ipred_obs_pk;
  matrix[n_cmt, n_t] A;
  array[n_t, n_theta] real theta;
  for(j in 1:n_t) {
    theta[j, 1] = CL * (1.0 + 0.0154 * ((CRCL[j] * 16.6667) - 66.0));
    theta[j, 2] = Q;
    theta[j, 3] = V1 * WT[j];
    theta[j, 4] = V2 * WT[j];
    theta[j, 5] = 0;
  }
  A = pmx_solve_twocpt(time, amt, rate, ii, evid, cmt, addl, ss, theta);
  ipred_pk = A[2, :] ./ (V1 * mean(WT));
  ipred_obs_pk = ipred_pk'[i_obs_pk];
}
```

References

[1] Tong et al. (2022) PAGE 2022
[2] Maier et al. (2020) CPT:PSP 10.1002/psp4.12492
[3] Wicha et al. (2015) PAGE abstract=3445
[4] Metrum Research Group. github.com/metrumresearchgroup/Torsten/
[5] Stan Development Team. mc-stan.org
[6] Keizer et al. CRAN.R-project.org/package=PKPDsim

Typical Workflow

1. Compile a Stan model, or define a new one (see left)
2. Supply dosing history, TDMs, covariates

a) Using PKPDsim data

```
covs <- list(
  CRCL = new_covariate(value = c(4, 4.5), times = c(0, 30))
)
reg <- new_regimen(amt = 1000, n = 3, interval = 12)
tdms <- data.frame(y = c(30, 15), t = c(25, 47))

dat <- PKPDsim_to_stan_data(
  covs, reg, tdms, ...
)
```

b) From NONMEM-style data

```
dat <- nonmem_to_stan_data(read_csv("dat1.csv"), ...)
```

3. Sample from the posterior distribution

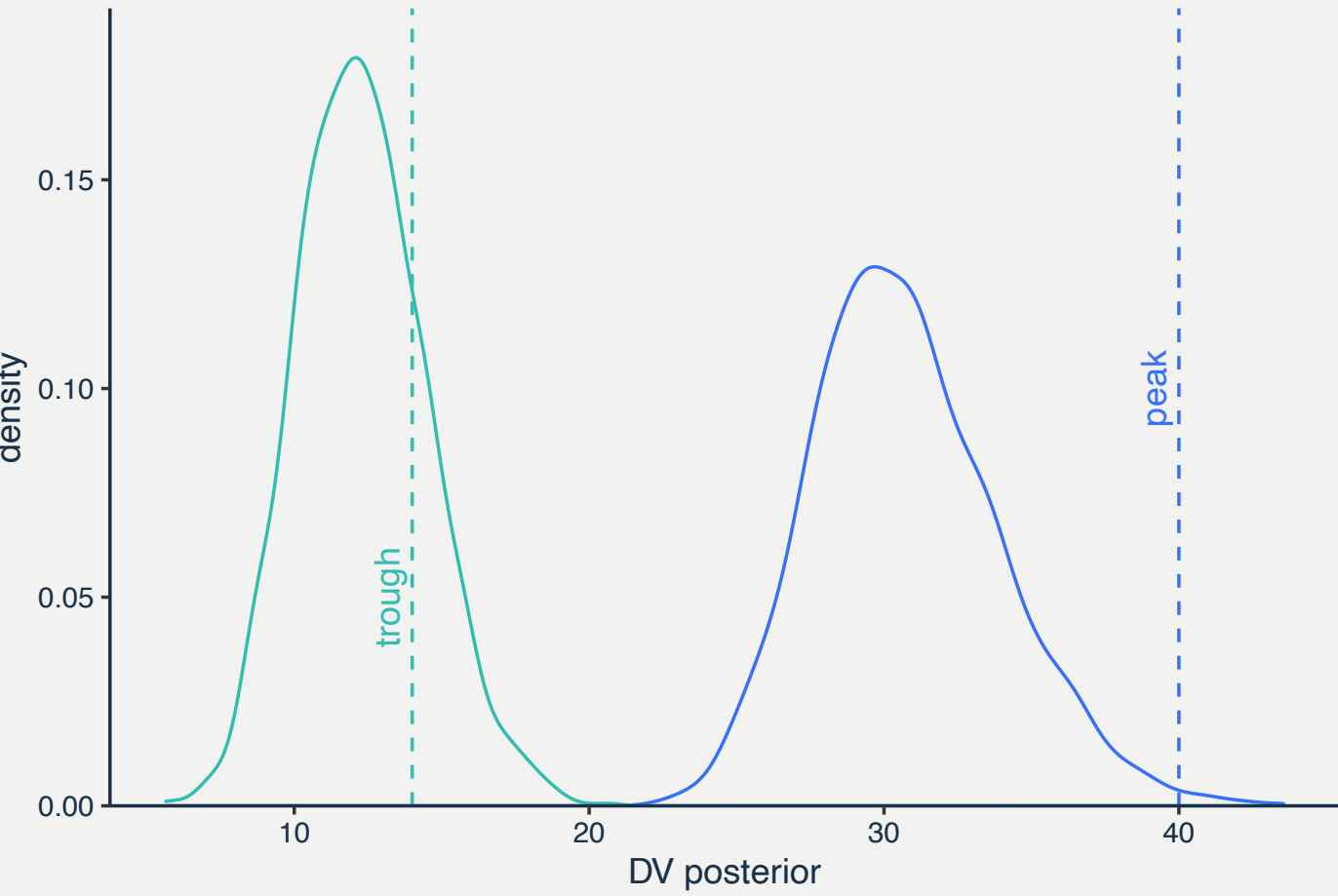
```
post <- get_mcmc_posterior(
  mod, dat, ...
)
```

> post

Parameters:							
	posterior_mode	mean	median	sd	q5	q95	rhat
CL	2.569	2.585	2.562	0.532	1.775	3.525	1.000
Q	1.688	2.045	1.833	1.003	0.871	3.914	1.006
V2	0.272	1.103	0.458	2.642	0.061	3.767	1.000
V1	0.597	0.587	0.587	0.069	0.475	0.699	1.001

Observed data, posterior:

	type	time	dv	mean	loc	pct	pct5	pct95
1	pk	2.5	40	30.74615	----- -----o	0.9940000	26.051325	36.36597
2	pk	11.5	14	12.29354	----- -----o	0.7833333	8.876648	15.91948

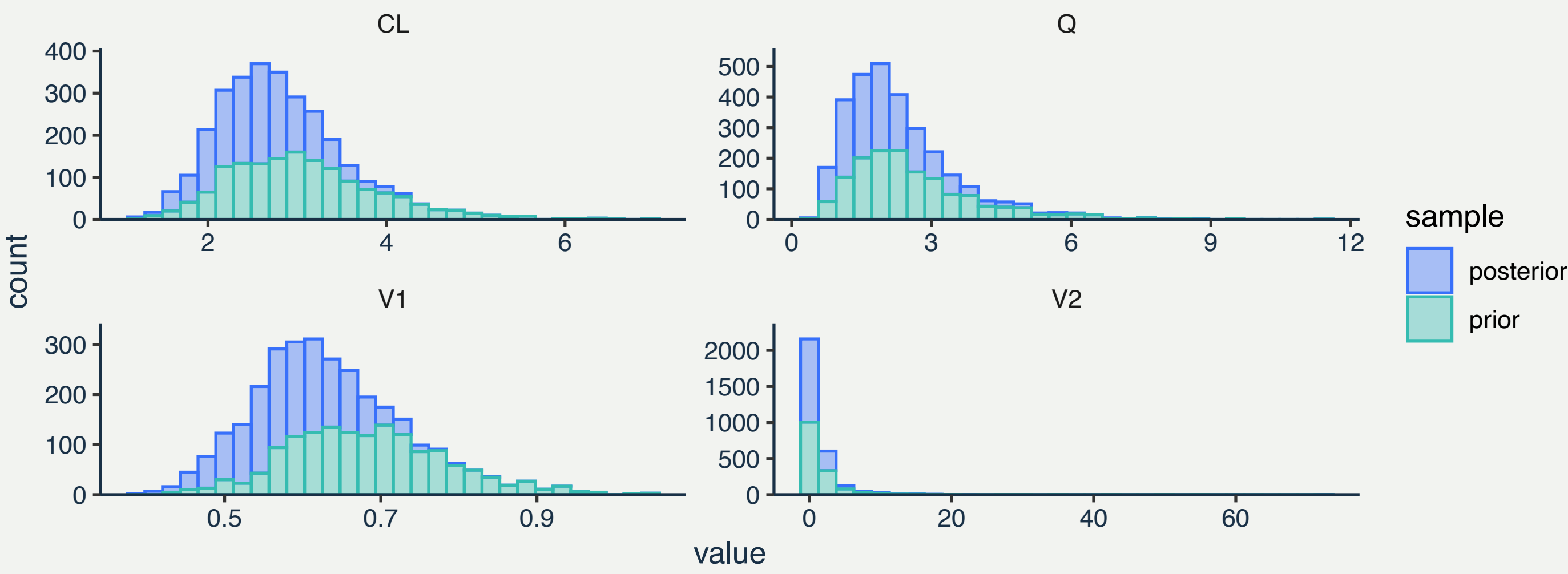


4. Simulate posterior concentration-time curves with uncertainty using

```
PKPDsim::sim(...)
```

5. Optionally, plot posterior parameter distributions:

```
plot_params(post)
```



Conclusion

PKPDposterior is an open-source Full Bayesian software for individualizing patient PK/PD parameter estimates implemented in R

- Human-readable estimation summaries
- User-friendly posterior/diagnostic visualization
- Well-tested code: 85% test coverage
- Acceptable speed for linear PK models (<1 s)

Get started:

```
devtools::install_github("InsightRX/PKPDposterior")
```

More information: insightrx.github.io/PKPDposterior/