1 Introduction

- Goal is to facilitate real-time herapeutic drug monitoring (TDM); translational research.
- PK predictions often not reported with uncertainty; cite literature.
- Need to quantify uncertianty in real time, but it's difficult.
- incorporate objective prior information
- We created a web app and suite of methods that accomplish this.

[need references] Adverse consequences can arise from both antibiotic underdosing and overdosing of hospitalized patients. In the former case, patients receive an insufficient amount of antibiotic to effectively combat the infection. In the latter, an overabundance of antibiotic often increases neurotoxicity risks and length of stay. In another study [cite], we showed roughly 20% of ICU patients being treated for sepsis were underdosed, while about 15% were overdosed. Variation in patient complications, such as multiple failing organs, can influence the dosing [and metabolism?] of an antibiotic. Simple dosing strategies (e.g. based on creatinine, alone) are unable to accurately capture the ideal amount of an antibiotic required for safe and effective treatment since individual pharmacokinetic heterogeneity varies greatly between patients.

The proposed statistical methodology and tools are designed to estimate how an individual is responding to treatment as measured by the concentration of antibiotic in the blood over time, predict how that individual will continue to respond, and determine how treatment can be adapted to reduce the risk of adverse outcomes. The goal is to design an effective procedure of therapeutic drug monitoring (TDM) to perform this assessment accurately and in real time. Dosing for a patient needs to be enough that the concentration in the blood remains above the necessary threshold for suppression of the offending microorganism (usually a multiple of the minimum inhibitory concentration), yet not so high as to increase the risk of adverse side effects.

Furthermore, predictions of pharmacokinetic performance often fail to provide estimates of uncertainty [cite literature]. We implement Bayesian methods using pharmacokinetic data from a prior study [cite] to provide posterior estimates of individual target attainment, measured as the fraction of the dosing period spent above the necessary blood concentration threshold. The fact that patients tend to vary widely in their response to treatment indicates that measures of uncertainty are critical to understanding response to treatment at both the population and patient-specific level. In order for the implementation of such procedures to be truly beneficial to individuals, it is necessary to have a method of assessing and predicting patient response in real time. This introduces the issue of selecting a statistical approximation technique that is both accurate in its quantification of uncertainty while also being computationally feasible to

Parameter	Units	Description
k_{10}	h^{-1}	Elimination rate from central compartment
k_{12}	h^{-1}	Distribution rate from central to peripheral compartment
k_{21}	h^{-1}	Distribution rate from peripheral to central compartment
k_R	$\mathrm{g{\cdot}h}^{-1}$	Infusion rate into central compartment
v_1	L	Volume of central compartment

Table 1: Two-compartment model parameter units (SI) and descriptions.

implement via a web application. Our statistical methodology is paired with a web application that implements these techniques and makes them accessible to, for example, a physician in real time. Assessments of the performance and computational efficiency are presented for various approximation methods.

2 Methods

2.1 Two-compartment model

The two-compartment pharmacokinetic model is expressed as a system of two ordinary differential equations as follows, where m_1 and m_2 are the masses of drug in the central and peripheral compartments, respectively.

$$\begin{aligned}\frac{dm_1}{dt} &= -k_{10}m_1 - k_{12}m_1 + k_{21}m_2 + k_R \\ \frac{dm_2}{dt} &= \qquad\qquad + k_{12}m_1 - k_{21}m_2\end{aligned}$$

The concentration of drug in the central compartment is given by $c_1 = m_1/v_1$, where v_1 is the volume of the central compartment. The parameters k_{10}, k_{12}, k_{21}, and k_R are described in Table 1

2.2 Bayes prediction model

Concentration measurements are modeled using a nonlinear regression method with either additive error as follows:

$$c_{ij} = \eta_i(t_{ij}, \theta_i) + \epsilon_{ij}$$

In this expression, c_{ij} is the measured concentration for subject $i = 1 \ldots n$ at time t_{ij} for $j = 1 \ldots m_i$, $\eta_i(t_{ij}, \theta_i)$ is the two-compartment model solution for subject i at time t_{ij} given parameters $\theta_i = [k_{10i}, k_{12i}, k_{21i}, v_{1i}]$, and ϵ_{ij} represents i.i.d. random error with mean zero. By specifying the random error density function, say $f(\epsilon_{ij}, \sigma)$, the subject-specific likelihood function is

$$L_i(\theta_i, \sigma) = \prod_{j=1}^{m_i} f(c_{ij} - \eta_i(t_{ij}, \theta_i), \sigma).$$

Thus, given a prior distribution $\pi_0(\theta_i)$, the subject-specific posterior is proportional to $\pi_i(\theta_i, \sigma) \propto \pi_0(\theta_i, \sigma) L_i(\theta_i, \sigma)$.

In the current context, f is taken as the normal density function with mean zero and standard deviation σ, and the prior distribution is generated to satisfy the following:

$$[\log \theta_i] \sim N_4(\mu_0, \Sigma_0) \tag{1}$$

$$[\log \sigma] \sim N_1(m_0, s_0) \tag{2}$$

where $N_4(\mu_0, \Sigma_0)$ represents the 4-variate normal distribution with mean μ_0 and covariance matrix Σ_0, and $N_1(m_0, s_0)$ represents the univariate normal distribution with mean m_0 and variance s_0. The prior distribution for the PK parameters represents our prior knowledge about the PK heterogeneity for a particular drug in a target population. Thus, the values of the prior hyperparameters should be carefully selected for the task at hand. In the current context, and by default, the hyperparameters are specified to correspond with estimates that arose from a study of piperacillin pharmacokinetics in a hospitalized, critically ill population. The particular values are listed in an appendix.

Due to the nonlinearity of the two-compartment model, the posterior distribution does not take a familiar form, and posterior summaries must be approximated. In particular, we sought to compute 95% credible bands for subject-specific concentration-time curves, and for a specific summary of the curve: the length of time in which the concentration exceeds a specified value. Monte Carlo techniques are often used to approximate these quantities. However, because these posterior summaries are presented in a web application in real time, we sought alternatives that were less computationally intensive and deterministic. We consider several methods as described in the following sections.

2.2.1 Laplace approximation

We first considered a method that makes use of two approximations. The first is a Laplace approximation to the subject-specific posterior density, and the second is a first order Taylor approximation of the target summary (i.e., the 'delta method'). The Laplace approximation is given as follows

$$\pi_i(\log \theta_i, \log \sigma) \approx N([\log \hat{\theta}_i, \log \hat{\sigma}], [-H]^{-1})$$

where $[\log \hat{\theta}_i, \log \hat{\sigma}]$ is the posterior mode and $H_{\hat{\theta}_i}$ is the posterior Hessian with respect to $[\log \theta_i, \log \sigma]$ evaluated at the posterior mode. The second approximation makes use of the delta method, such that a posterior functional $h(\log \theta_i)$ has an approximate normal distribution. In the present context, $h(\log \theta_i)$ represents the logit of the fraction of the dosing period in which the concentration of drug exceeds a spedified value. The first-order Taylor expansion of $h(\log \theta_i)$ is

$$h(\log \theta_i) \approx h(\log \hat{\theta}_i) + G^T(\log \theta_i - \log \hat{\theta}_i),$$

where G is the gradient of $h(\log \theta_i)$ with respect to $[\log \theta_i, \log \sigma]$ evaluated at the posterior mode. Thus, given that $[\log \theta_i, \log \sigma]$ has an approximate normal

distribution, the same is true for $h(\log \theta_i)$:

$$h(\log \theta_i) \sim N(h(\log \hat{\theta}_i), G^T[-H]^{-1}G).$$

An approximate $(1-\alpha)\cdot 100\%$ credible interval for $h(\log \theta_i)$ is thus given by the $\alpha/2$ and $(1-\alpha)/2$ quantiles of the approximate posterior distribution for $h(\log \theta)$. This method is computationally elegant, since the posterior Hessian and posterior mode can be computed simultaneously by most optimization software routines (e.g., the R function 'optim'). Indeed, this method usually has the smallest computational burden (i.e., number of likelihood evaluations) among the three methods considered here.

2.3 Evaluation of approximation method

The posterior coverage of the MIC statistic 95% confidence interval was estimated as follows:

50 samples were drawn from the prior distribution of PK parameters. For each set of prior parameters, 6 observations were simulated using the concentration predicted by the first 4 PK parameters and the standard error term corresponding to the 5th prior parameter. All observations were simulated during the final (5th) dosing period at 32, 32.5, 33, 34, 36, and 38 hours.

For each of the 50 sets of prior parameters and simulated data:

- A Laplace approximation to the posterior density was calculated and the delta method used to estimate the standard errors of the MIC statistic. A 95% asymptotic normal confidence interval was used with the delta method standard errors to estimate upper and lower bounds.
- 3,000 samples from the posterior distribution of PK parameters were taken using a Metropolis MCMC algorithm with 2,000 warm-up iterations. The proportion of the 3,000 posterior samples falling within the upper and lower bounds was used to estimate the proportion of the posterior distribution of the MIC statistic falling within the confidence interval.

On 50 samples we calculated a mean posterior coverage of 89.87% with an IQR 88.10% to 92.39%. We calculate a Monte-Carlo standard error of approximately $\sqrt{(0.9)(0.1)/3000} = 0.005$. Confidence intervals for the MIC statstic were also considered on the logit and probit scales in order to restrict the bounds to the interval [0,1]; however, as these transformations resulted in slightly lower and more variable posterior coverage, the original scale was used.

- used MCMC as gold-standard
- quantified posterior probability contained within approximate intervals for $logit^{-1}(h(log\theta_i))$
- quantified number of posterior evaluations
- quantified computing time
- note: make sure to uncomment line below and set up the bibtex doc

Parameter	Value
μ_0	
Σ_0	
α_0	
β_0	

Table 2: Prior hyperparameters for piperacilling in a population of hospitalized, critically ill patients.

3 Appendix: Prior hyperparameters for piperacillin among critically ill

4 Example

```
#Other packages for this demo
library(xtable)
```

- Generate ivt data frame
- obtain raw mic estimate
- fit pkm model
- plot estimates
- update the model and re-plot estimates
- shiny app demo

```
# devtools::install_github("hlweeks/pkpredict")
library(pkpredict)

## Error in library(pkpredict):  there is no package called 'pkpredict'
```

4.1 Sample infusion schedule and concentation data

package functions: `ivt_toList`

```r
# Time in hours since first infusion
start_times <- c(0, 8, 16, 24, 32)
# Duration of each infusion in hours
duration <- 0.5
# Rate in g/h
rate_of_infusion <- rep(6, 5)

ivt_d <- ivt_toList(begin = start_times, dur = duration, rate = rate_of_infusion)
```

```
## Error in eval(expr, envir, enclos):  could not find function "ivt_-
toList"
```

Table 3: Sample Infusion Schedule

Begin (h)	End (h)	Infusion Rate (g/h)
0.0	0.5	6.0
8.0	8.5	6.0
16.0	16.5	6.0
24.0	24.5	6.0
32.0	32.5	6.0

```r
# Time is in hours since first infusion
# Concentration is in mcg/ml (or equivalently, mg/L)
dat_d <- data.frame("time" = c(1, 4, 40),
                    "con" = c(82.7, 80.4, 60),
                    check.names = FALSE)
```

Table 4: Sample Data (observed from blood draws)

Time (h)	Concentration (mcg/mL)
1.0	82.7
4.0	80.4
40.0	60.0

4.2 MIC estimates

package functions: `mic_stat`

```r
# Threshold is in mcg/mL (same scale as observed data)
# By default, considering time through 12 hours after end of the final dose

# Prior estimate
prior_ft.mic <- mic_stat(ivt = ivt_d, th = 64)
```

```
## Error in eval(expr, envir, enclos):  could not find function "mic_-
stat"

prior_ft.mic$ftmic

## Error in eval(expr, envir, enclos):  object 'prior_ft.mic' not found

# Default parameters used
(pars <- prior_ft.mic$pars)

## Error in eval(expr, envir, enclos):  object 'prior_ft.mic' not found

# Posterior estimate after observed measurements
est <- optim(pars, log_posterior, ivt = ivt_d, dat = dat_d,
             control = list(fnscale=-1), hessian=TRUE)

## Error in optim(pars, log_posterior, ivt = ivt_d, dat = dat_d, control
= list(fnscale = -1), :  object 'pars' not found

post_ft.mic <- mic_stat(ivt = ivt_d, th = 64, pars = est$par)

## Error in eval(expr, envir, enclos):  could not find function "mic_-
stat"

post_ft.mic$ftmic

## Error in eval(expr, envir, enclos):  object 'post_ft.mic' not found
```

In order to get confidence intervals for these estimates, it is necessary to fit the PK model.

4.3 PK Model

package functions: `pkm`

```
pk.fit <- pkm(formula = con ~ time, data = dat_d,
              ivt = ivt_d, thres = 64)

## Error in eval(expr, envir, enclos):  could not find function "pkm"

pk.fit$ftmic

## Error in eval(expr, envir, enclos):  object 'pk.fit' not found
```

```
## Error in eval(expr, envir, enclos):  object 'pk.fit' not found
## Error in names(pred.df) <- c("Time Since First Infusion (h)", "Concentration
(mcg/mL)", :  object 'pred.df' not found
## Error in xtable(pred.df, digits = 2, caption = "Predicted Values"):
object 'pred.df' not found
```

```
## Error in align(xdf) <- rep("c", 4):  object 'xdf' not found
## Error in print(xdf, caption.placement = "top", include.rownames
= FALSE): object 'xdf' not found
```

4.4 Model Methods

package functions: `plot.pkm`, `predict.pkm`, `update.pkm`, `confint.pkm`

```
plot(pk.fit)

## Error in plot(pk.fit):  object 'pk.fit' not found
```