PharmaSUG 2025 - Paper ST-173

# Calculating exact posterior probabilities and credible intervals from Bayesian borrowing robust mixture priors for binary, count and continuous outcomes in R and SAS

Darren Scott & Armando Turchetta (AstraZeneca)

## ABSTRACT

Recently, Bayesian statistical methods have been developed that leverage historical clinical trial information to assist in the discovery, development, and delivery of medicines. “Dynamic” models preserve the primacy of the data in the target study by discounting the external information according to how closely it matches the target data. A popular approach for “dynamic borrowing” is a robust mixture prior (RMP), where an informed element(s) is combined with a vague element and a prior weight.

Typically, in the Bayesian paradigm inference is performed through various Monte Carlo (MC) sampling schemes. A prudent choice of RMP leads to a conjugate mixture posterior for the control and treatment parameter. In this article we provide insights into how simple mathematical concepts lead to the application of numerical integration in R and SAS to obtain fast an accurate inference from the posteriors after Bayesian borrowing. Exact probabilities also simplify and reduce timing of the validation process for clinical trial reporting. We use an example with binary outcome data to calculate the exact posterior probability of a treatment effect, the prior weight tipping point and posterior credible intervals on the control rate, treatment rate and treatment effect. Finally, we detail the approach for count and continuous data, with code which can implemented by statistical programmers.

## INTRODUCTION

Traditionally, randomised clinical trials are designed and analysed from a frequentist perspective using classical hypothesis testing. However, more recently, there has been a growing awareness of the benefits of Bayesian approaches which naturally allow for the explicit integration of previous knowledge with new empirical data. This is particularly attractive in clinical trials where multiple trials are often conducted on the same illness in the pursuit of an effective treatment, or an existing therapy has been established.

Bayesian models can be constructed to be “dynamic”, preserving the primacy of the data in the target study by discounting the external information according to how closely it matches the target data. Various models with informed priors have been proposed (Sharon, Brian, Alexander, & Joseph, 2021), (Scott & Lewin, 2024), where the structure of the model serves to discount the historical information in when the prior information and likelihood are in not agreement

A popular and simple approach is to use a robust mixture prior (RMP) where the prior is made up of an informed element and a vague component, characterised by the choice of hyperparameters with an associated weight

$$p(\theta) = wp(\theta|informed) + (1-w)p(\theta|vague). \tag{1}$$

Careful choice of the vague element in the mixture prior serves to fatten the tails of our prior from the external information, making the prior robust to the assumption that the historical and target trial data are exchangeable and prior-data conflict. If each element is conjugate to the likelihood, then the posterior will be available in closed form. Careful choice of the hyperparameters in the vague component and weight ensures that the prior is robust.

This article explains how to calculate the exact posterior quantities from the application of the RMP for Bayesian borrowing in R and SAS via Proc IML. A brief technical summary explains how the posterior can be calculated from our dynamic prior. We use an example from a RMP with respect to a binomial likelihood to describe a numerical approach for calculating the posterior probability of a treatment effect (primarily defined by the risk difference $\theta_t - \theta_c$) and credible intervals (CrI) for the parameters associated with the treatment and control groups. These numerical solutions are used to determine the minimum prior weight ($w$) which leads to a treatment effect being declared, or "tipping point" analysis. We address how these calculations can be performed for the RMP with count and continuous data in R and SAS. By using numerical integration, we avoid the need for any Monte Carlo simulation and guarantee a high degree of accuracy for a small amount of computing time.

## BAYESIAN BORROWING WITH MIXTURE PRIORS

### ROBUST MIXTURE PRIORS

The RMP is typically a linear combination of conjugate priors in the form of $(1)$, where an application of Bayes rule leads to a posterior of

$$p(\theta|y) = w^* p(\theta|informed, y) + (1 - w^*) p(\theta|vague, y), \tag{2}$$

where $y$ denotes the observed data from the trial.

Despite the mixture prior, each density in $(2)$ will remain in its conjugate form. There are plenty of resources which explain conjugacy in more detail (Gelman, et al., 2021). I leave this to the interested reader. The dynamic aspect of the Bayesian borrowing is from the posterior mixture weight $w^*$ associated with the informed prior,

$$w^* = w \frac{p(y|informed)}{p(y)}. \tag{3}$$

The prior weight $w$ is multiplied by the conditional likelihood with respect to the informed prior $p(y|informed)$ and normalised by the marginal likelihood $p(y)$. The more likely we are to observe the data with respect to the informed prior, the larger the fraction which scales the prior weight $w$. We are performing a comparison of how likely the data is under the two prior components, which explains why the choice of the vague density is so crucial to the dynamic aspect of the model.

Thus, for a general mixture prior of

$$p(\theta) = wp(\theta|a_1, b_1) + (1 - w)p(\theta|a_2, b_2),$$

the posterior weight is

$$w^* = \frac{wf_1(a_1, b_1)}{wf_1(a_1, b_1) + (1 - w)f_2(a_2, b_2)}, \tag{4}$$

where the combination of prior and likelihood determines the form of $f_j$ for $j = 1,2$.

To account for differences between the parameters in the historical study (Schmidli & Neuenschwander, 2014) proposed the robust meta-analytic predictive prior (RMAP). The vague and informed elements remain conjugate, but now approximate the posterior predictive distribution for $\theta$ extending $(1)$ to a general form

$$p(\theta) = \sum_{j=1}^{k} w_j p_j(\theta), \tag{5}$$

where $\sum_j w_j = 1$.

Although this prior changes the underlying assumption of exchangeability of the model, for our discussion these two priors are interchangeable. For simplicity, I will focus on a mixture prior of just two elements as the methods described are easily extended beyond this.

As the posterior is a mixture of densities, despite their conjugate form, inference is more difficult since we can no longer use the base functions to calculate probabilities and quantiles of interest. For binary and count data there is the added challenge that the difference between these two random variables have closed form solutions which have normalising constants containing an integral.

## BINARY OUTCOME

In our worked example, a mixed beta prior is posited for a binomial likelihood. A previous study for the same target population, but for a different region, is used to inform both the control and treatment groups for the upcoming (target) trial. In the historical trial the control arm had 110 responses and 250 non-responses, whilst the treatment group had 175 response and 190 non-responses. Assigning a 30% weight to the information in both the control and treatment arm, our prior for the control is

$$p(\theta_c) = \ 0.3 \times beta(\theta_c|110, 250) + 0.7 \times beta(\theta_c|1, 1),$$

where the choice of 1 for the hyperparameters in vague element is equivalent to a uniform prior. The prior for the treatment group is

$$p(\theta_t) = \ 0.3 \times beta(\theta_t|175, 190) + 0.7 \times beta(\theta_c|1, 1).$$

In a trial of 30 patients in each arm, we observe 10 responses in the control arm and 15 in the treatment arm. The posterior distributions take the same form as the prior. The posterior for the control parameter is

$$p(\theta_c) = \ w^* \times beta(\theta_c|110 + 10, 250 + 30 - 10) + (1 - w^*) \times beta(\theta_c|1 + 10, 1 + 30 - 10), \qquad (6)$$

with the posterior weight of

$$w_c^* = \frac{0.3 f_{IC}}{0.3 f_{IC} + 0.7 f_{VC}},$$

and elements

$$f_{IC} = \frac{B(120, 270)}{B(110,250)} \quad f_{VC} = \frac{B(11, 21)}{B(1,\ 1)},$$

where $B(\cdot,\cdot)$ is the Beta function, $f_{IC}$ is the marginal likelihood with respect to the informed prior component and $f_{IC}$ is the marginal likelihood with respect to the vague prior component.

The posterior distribution of the treatment parameter is updated in the same fashion.

## COUNT OUTCOME

The RMP for a count outcome $y_{ik}$ assumed to follow a Poisson distribution where $i$ is the individual and $k$ is the arm of the trial is

$$p(\theta_k) = w gamma(\theta_k|a_1, b_1) + (1 - w) gamma(\theta_k|a_2, b_2).$$

The hyperparameters $a_1$and $a_2$ are the shape parameters and $b_1$and $b_2$ are the scale parameters of the gamma distribution. The posterior mixture weight (4) is

$$f_{jk}(a_j, b_j) = \frac{b_j^{a_j} \Gamma\left(\sum_i y_{ik} + a_j\right)}{\left(n_k + b_j\right)^{\sum_i y_{ik} + a_j} \Gamma\left(a_j\right)} \qquad (7)$$

for $j = 1,2$.

## CONTINUOUS OUTCOME

With the sample mean outcome $\bar{y}_k$ with known standard error $\sigma$ from a Gaussian likelihood, the RMP prior is

$$p(\theta_k) = wN(\theta_k|a_1, b_1) + (1 - w)N(\theta_k|a_2, b_2)$$

where $a_1$ and $a_2$ are the mean parameters and $b_1$ and $b_2$ are the variance parameters of the normal distribution. The posterior mixture weight in $(4)$ is

$$f_{jk}(a_j, b_j) = \frac{\exp\left(-\frac{(\bar{y} - a_j)^2}{2(\sigma^2 + b_j)}\right)}{\sqrt{b_j + \sigma^2}}, \qquad (8)$$

and the component posterior update has variance and mean

$$b_j^* = \left(\frac{\bar{y}_k}{\sigma^2} + \frac{a_j}{b_j}\right)^{-1} \qquad a_j^* = b_j^*\left(\frac{1}{\sigma^2} + \frac{1}{b_j}\right) \qquad (9)$$

for $j = 1,2$.

# POSTERIOR PROBABILITY OF AN EFFICACIOUS TREATMENT EFFECT AND TIPPING POINT

The posterior probability of an efficacious treatment effect can be expressed as an integral by using the law of total probability and conditioning on one of the random variables ($\theta_t$ or $\theta_c$). If we wish to obtain the posterior probability of the risk difference, where a positive value from $\theta_t - \theta_c$ represents an improvement, then

$$\begin{aligned} p(\delta \ = \theta_t - \theta_c > 0|y) &= \int_{\Theta_t} p\,(\theta_t^* - \theta_c > 0|y, \theta_t = \theta_t^*)p(\theta_t = \theta_t^*|y)d\theta_t, \\ &= \int_{\Theta_t} p\,(\theta_c < \theta_t^*|y, \theta_t = \theta_t^*)p(\theta_t = \theta_t^*|y)d\theta_t. \qquad (10) \end{aligned}$$

The integrand in expression $(10)$ is a product of the posterior cumulative probability that the control parameter is less than the treatment and the posterior probability density of the treatment. Although $(10)$ is a double integral

$$p(\delta \ > 0|y) = \int_{\Theta_t} F_{\theta c}\,(\theta_t|y)p_{\theta_t}(\theta_t|y)d\theta_t,$$

where $F_{\theta c}(\cdot)$ is the control cumulative probability function (CDF) and $p_{\theta_c}$ is the control probability mass function (PMF), we can exploit the base functions in SAS and R. When the outcome changes, the support of the prior and thus the limits of the integral $(10)$ change.

In accordance with the definition of “Bayesian significance” given by (Spiegelhalter, Abrams, & P, 2003), a result is considered significant if the posterior probability that the parameter of interest $\delta$ belongs to the alternative hypothesis space $\Delta_1$ is not less than a specified threshold $1 - \alpha$. As the borrowing is dependent on the weight we give to the informed element, we may wish to determine the minimum prior weight required to declare a significant treatment effect or “tipping point”

$$\min_{w} p(\delta = \theta_t - \theta_c > 0|y, w) > 1 - \alpha. \tag{11}$$

In our binary example our threshold is set to $0.975$. The treatment effect random variable is denoted by $\Delta$ and a realised value is $\delta$.

## POSTERIOR PROBABILITY DENSITY OF THE RISK DIFFERENCE

The distribution for the posterior treatment effect from the RMP will not be available in R or SAS. Rather than deriving a closed form expression, we use a change of variable up to the order of integration and apply numerical integration.

We let $\delta = \theta_t - \theta_c$, $\theta_c = \theta_c$ so that the probability density function (PDF) for the posterior treatment effect $p(\delta|y)$ up to an order of integration is

$$p(\delta|y) = \int_{\theta_c} p_{\theta t}(\delta + \theta_c|y) p_{\theta c}(\theta_c|y)|J|d\theta_c\,, \tag{12}$$

where $J$ is the Jacobian which equals 1. By using a change of variable, we have a convolution of two distributions.

To obtain the PDF we need to define the inequalities for $\delta$ and $\theta_c$, which will depend on the prior and likelihood combination. In the next section we define these for each response as we use $(12)$ to calculate the CDF and CrI.

## CREDIBLE INTERVALS

To calculate the $k$th CrI from our posterior distributions for the treatment parameter $\theta_t$, control parameter $\theta_c$ and treatment effect $\delta = \theta_t - \theta_c$, we find $q$ which satisfies

$$F(q|y) - \frac{k}{100} = 0, \tag{13}$$

where $F(\cdot)$ is the corresponding posterior CDF for our random variable.

The quantiles can also be used to calculate an equal-tailed two-sided CrI, however these may not have the shortest width if the posterior is asymmetric. As we are primarily motivated with using the quantile to declare a significant treatment effect this is not a concern. However, for the interested reader, if the distribution is unimodal the shortest width interval $(L, U)$ can be obtained using the Nelder-Mead algorithm (Nelder & Mead, 1965) to minimize the function

$$G(L, U) = |F_\delta(U) - F_\delta(L) - (1 - \alpha)| + |p_\delta(U) - p_\delta(L)|.$$

To calculate the CDF for the risk difference we can integrate over the PDF in $(12)$

$$p(\Delta < \delta|y) = \int_{l}^{\delta} \int_{\Theta_C} p_{\theta_t}(u\ + \theta_c|y) p_{\theta_c}(\theta_c|y) d\theta_c\, du. \tag{14}$$

where $\delta$ and $l$ denotes the upper and lower limit for treatment effect integral. This is then used in $(13)$ to calculate $q$th interval to check for Bayesian significance of the treatment effect.

Alternatively, if we do not require the PDF, we can avoid the Jacobian calculation by using the law of total probability for the CDF of the risk difference

$$P(\Delta < \delta|y) = \int_{\theta_c} F_{\theta_t}(\delta + \theta_c|y, \theta_c = \theta_c^*)p(\theta_c = \theta_c^*|y)d\theta_c. \tag{15}$$

## BINARY OUTCOME

We proceed to obtain the CDF for the risk difference for a binary outcome. The lower limit $l$ of $(14)$ depends on the support of the treatment effect. In the code below we use $(14)$ to calculate the CrI for our binary data, to highlight the limitations of SAS.

The non-zero region of the integrand for our beta prior distribution is identified from the inequalities

$$-1 \leq \delta \leq 1, \quad 0 \leq \theta_c \leq 1, \quad -\theta_c \leq \delta \leq 1 - \theta_c,$$

and displayed in Figure 1;

- $-1 \leq \delta < 0$, $-\delta < \theta_c \leq 1$, yellow region,
- $0 \leq \delta \leq 1$, $0 \leq \theta_c \leq 1 - \delta$, blue region.

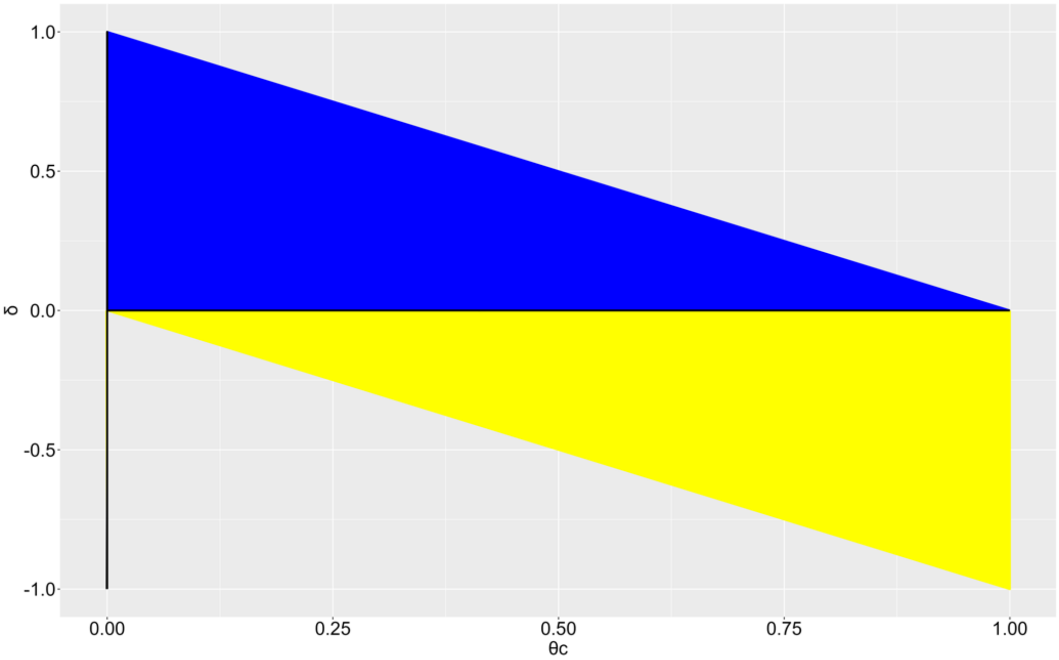

**Figure 1. Region of non-zero density for the treatment effect from the binary outcome. The yellow area is for $\delta < 0$ and the blue is for $\delta > 0$.**

If we wish to calculate $p(\Delta < \delta|y)$ with $(14)$ then we use the following integrals, with limits from Figure 1.

If $\delta < 0$ then

$$p(\Delta < \delta|y) = \int_{-1}^{\delta}\int_{-\delta}^{1} p_{\theta_t}(u + \theta_c|y)p_{\theta_c}(\theta_c|y)d\theta_c\, du,$$

else if $\delta > 0$ then

$$p(\Delta < \delta|y) = \int_{-1}^{0}\int_{0}^{1} p_{\theta_t}(u + \theta_c|y)p_{\theta_c}(\theta_c|y)d\theta_c\, du + \int_{0}^{\delta}\int_{0}^{1-\delta} p_{\theta_t}(u + \theta_c|y)p_{\theta_c}(\theta_c|y)d\theta_c\, du. \tag{16}$$

Alternatively, we can use $(15)$ taking care to define the CDF of $F_{\theta_t}(\cdot)$ correctly for $\delta + \theta_c$ and using Figure 1 to help choose the integration limits. If $\delta < 0$ (yellow region) then $F_{\theta_t}(\delta + \theta_c | y, \theta_c = \theta_c^*) = 0$ when $\delta + \theta_c < 0$

$$p(\Delta < \delta | y) = \int_{-\delta}^{1} F_{\theta_t}(\Delta^* + \theta_c | y, \theta_c = \theta_c^*)\, p(\theta_c = \theta_c^* | y) d\theta_c,$$

If $\delta > 0$, $P_{\theta_t}(\delta + \theta_c | y, \theta_c = \theta_c^*) = 1$ when $\delta + \theta_c > 1$

$$p(\Delta < \delta | y) = \int_{0}^{1-\delta} F_{\theta_t}(\delta + \theta_c | y, \theta_c = \theta_c^*)\, p(\theta_c = \theta_c^* | y) d\theta_c + \int_{1-\delta}^{1} p(\theta_c = \theta_c^* | y) d\theta_c.$$

In the case of a binary outcome, we may also be interested in defining the treatment effect as a risk ratio $M$. We avoid the need to calculate the Jacobian by using

$$p(M < \mu | y) = \int_{\theta_c} F_{\theta_t}(\mu\theta_c | y, \theta_c = \theta_c^*) p_{\theta_c}(\theta_c = \theta_c^* | y) d\theta_c. \qquad (17)$$

Defining $0 < M < \infty$, if $\mu < 1$

$$p(M < \mu | y) = \int_{0}^{1} F_{\theta_t}(\mu\theta_c | y, \theta_c = \theta_c^*) p_{\theta_c}(\theta_c = \theta_c^* | y) d\theta_c\,. \qquad (18)$$

And if $\mu > 1$, $F_{\theta_t}(\mu\theta_c | y, \theta_c = \theta_c^*) = 1$ for $\theta_c > 1/\mu$

$$p(M < \mu | y) = \int_{0}^{1/\mu} F_{\theta_t}(\mu\theta_c | y, \theta_c = \theta_c^*) p_{\theta_c}(\theta_c = \theta_c^* | y) d\theta_c + \int_{1/\mu}^{1} p_{\theta_c}(\theta_c = \theta_c^* | y) d\theta_c.$$

### COUNT DATA REGION

To obtain the CDF for the risk difference for count data, the region of the integrand in $(14)$ is

- $\delta > 0 \quad 0 \leq \theta_c$, the integral over $\theta_c$ is unchanged,
- $\delta < 0 \quad \theta_c \geq -\delta$.

We avoid making both the integral limits dependent on $t$ for the CDF, speeding up the calculation, by noting that if we condition on $\theta_t$ the region is

- $\delta > 0 \quad \delta \leq \theta_t$, the integral over $\theta_t$ is unchanged,
- $\delta < 0 \quad \theta_t \geq 0$.

Thus, the integration can be defined by the following.

- If $\delta > 0$ integrate $\theta_c$ over $\int_0^\infty p_{\theta_t}(\delta + \theta_c | y) p_{\theta_c}(\theta_c | y) d\theta_c$,
- If $\delta < 0$ integrate $\theta_t$ over $\int_0^\infty p_{\theta_t}(\theta_t | y) p_{\theta_c}(\theta_t - \delta | y)\, d\theta_t$.

The CDF for a risk difference of $\delta = 0.5$ is thus evaluate

$$F_\delta(0.5) = \int_{-\infty}^{0} \int_{0}^{\infty} p_{\theta_t}(\theta_t | y) p_{\theta_c}(\theta_c - u | y) d\theta_c du + \int_{0}^{0.5} \int_{0}^{\infty} p_{\theta_t}(u + \theta_c | y) p_{\theta_c}(\theta_c | y) d\theta_c du.$$

## CODE AND WORKED EXAMPLE

In this section we detail the code in R and SAS for a binary outcome and a risk difference treatment effect which calculates

- the mixture posterior distributions,
- posterior probability of our treatment effect,
- minimum prior weight $w$ for a significant treatment effect conditional on the observed data or “tipping point”,
- credible intervals for our treatment effect.

We also include R code for a count outcome, to illustrate how easy it is to change the code provided for the binary outcome, when the response changes. The continuous outcome in either SAS or R naturally follows (so the code is omitted). Each step in the code is outlined. The binary outcome is first, where the code is also applied to our dataset.

## R ANALYSIS - BINARY DATA

In the worked example for the binary outcome, we posit the following robust mixture prior for the control and treatment group

- $p(\theta_c) = 0.3 \times beta(\theta_c|110, 250) + 0.7 \times beta(\theta_c|1, 1)$,
- $p(\theta_t) = 0.3 \times beta(\theta_t|175, 190) + 0.7 \times beta(\theta_c|1, 1)$.

In R, the probability of a treatment effect for our binary outcome is calculated by

- Define a function (*post_bw*) for calculating the posterior weight $(4)$ from the prior weight $(w)$, prior hyperparameters $(a, b)$ and the observed data (number of patients $n$ and events $x$). The vague hyperparameters are set to $1$.
- Define the integrand $(10)$ for the binomial likelihood as a function of the treatment parameter (*thetat*), prior weight $(w)$, prior hyperparameters for the treatment $(a_1, b_1)$ and control $(a_2, b_2)$, and data for the treatment $(n_1, x_1)$ and control $(n_2, x_2)$.
- Use the R package cubature to perform numerical integration using the *int_btrt_effect* function. The limits of the integration are defined by the upperLimit and lowerLimit.

```
post_bw <- function(w, x, n, a, b){

   wup1 = (w * beta(x + a, n - x + b)) / beta(a, b)
   wup2 = (1 - w) * beta(x + 1, n - x + 1)

   dw = wup1 / (wup1 + wup2)

   return(dw)
}
```

**Program 1. R: Posterior weight of the RMP for binary data.**

```
int_btrt_effect <- function(thetat, w,
                            a1, b1, x1, n1,
                            a2, b2, x2, n2){

  (post_bw(w, x1, n1, a1, b1)  * pbeta(thetat, a1 + x1, b1 + n1 - x1 ) +
     (1 - post_bw(w, x1, n1, a1, b1)) *  pbeta(thetat, 1 + x1, 1 + n1 -x1)) *
  (post_bw(w, x2, n2, a2, b2) * dbeta(thetat, a2 + x2, b2 + n2 - x2) +
     (1 - post_bw(w, x2, n2, a2, b2)) *  dbeta(thetat, 1 + x2, 1 + n2 - x2))
}
```

**Program 2. R: Integrand for posterior probability of an efficacious treatment effect (binary).**

```
library(cubature)

# Hyperparameters
w <- 0.3
a1 <- 110; b1 <- 250
a2 <- 175; b2 <- 190
# Control data
x1 <- 10; n1 <- 30
# Treatment data
x2 <- 15; n2 <- 30

out <- cubature::hcubature(f = int_btrt_effect,
                           lowerLimit = 0,
                           upperLimit = 1,
                           tol = 1e-8,
                           w = w,
                           a1 = a1, b1 = b1, x1 = x1, n1 = n1,
                           a2 = a2, b2 = b2, x2 = x2, n2 = n2)
```

**Program 3. R: Numerical integration to calculate the posterior probability of a significant treatment effect (risk difference).**

The minimum weight $w$ for the control and the treatment prior, which achieves a significant treatment effect, can be calculated from the root of

$$p(\theta_t - \theta_c > 0|y, w) - (1 - \alpha) = 0. \qquad (12)$$

Using our defined functions, we add;

- A function (*tip_point*) which subtracts $1 - \alpha$, $(\alpha = 0.025)$ from our numerical integration calculation.
- Apply the R *uniroot* function on *tip_point* with limits 0 and 1.

```
tip_point <- function(w, a1, b1, x1, n1,
                      a2, b2, x2, n2,
                      tpv # threshold for significant effect
                      ){

  cubature::hcubature(f = int_btrt_effect,
                      lowerLimit = 0,
                      upperLimit = 1,
                      tol = 1e-8,
                      w = w,
                      a1 = a1, b1 = b1, x1 = x1, n1 = n1,
                      a2 = a2, b2 = b2, x2 = x2, n2 = n2)$integral - tpv
}

w_tp <- uniroot(tip_point, lower = 0, upper = 1,
                a1 = a1, b1 = b1, a2 = a2, b2 = b2,
                n1 = n1, x1 = x1, n2 = n2, x2 = x2,
                tpv = 0.975)

w_tp$root
0.3542543
```

**Program 4. R: Minimum prior weight w for a significant treatment effect.**

To calculate the $q$th CrI for the treatment parameter $\theta_t$ and the control parameter $\theta_c$ we define the following functions;

- *ci_integrand_beta*, the integrand for $F(\theta|y)$.
- *qauntile_fun*, to perform the integration.
- *quantile_mix*, calculate the $q$th quantile by applying the uniroot function.

```
ci_integrand_beta <- function(theta, w, a, b, x, n){

   (post_bw(w, x, n, a, b) * dbeta(theta, a + x, b + n - x ) +
    (1 - post_bw(w, x, n, a, b)) * dbeta(theta, 1 + x, 1 + n - x))

}

qauntile_bfun <- function(ul, w, a, b, x, n, prob){

  cubature::hcubature(f = ci_integrand_beta,
                      lowerLimit = 0,
                      upperLimit = ul,
                      tol = 1e-9,
                      w = w,
                      a = a, b = b, x = x, n = n)$integral - prob
}
```

**Program 5. R: Integration for the CrI for the posterior for control $(\boldsymbol{\theta_c})$ and treatment $(\boldsymbol{\theta_t})$ parameters.**

```
quantile_bmix <- function(w, a, b, x, n, prob){

  qle <- uniroot(qauntile_bfun, lower = 0, upper = 1,
                 w = w, a = a, b = b, x = x, n=n, prob = prob)$root

  return(qle)
}
```

**Program 6.R: Calculate $\boldsymbol{q}$th equal tailed posterior CrI for control $(\boldsymbol{\theta_c})$ and treatment $(\boldsymbol{\theta_t})$ parameters.**

```
quantile_bmix(w = w, a = a1, b = b1, x = x1, n = n1, prob = 99/100)
0.5085713

quantile_bmix(w = w, a = a2, b = b2, x = x2, n = n2, prob = 99/100)
0.6653837
```

**Program 7.R: Posterior 99th CrI for control and treatment parameters.**

For the treatment effect, we define the integrand $(14)$ with respect to the posterior distributions of $\theta_t$ and $\theta_c$, where the arguments are hyperparameters updated with the data, which is clear from Program 8.

To calculate the $q$th CI for the treatment effect $\delta = \theta_t - \theta_c$ we define the following functions;

- *fun_bmix_int*, the integrand to calculate $F(\delta|y)$, where the arguments are hyperparameters updated with the data.
- *delta_b_cdf*, to perform numerical integration to obtain the CDF for $\delta$,
- *quantile_bdiff*, calculate the $q$th quantile by applying uniroot function.

```
fun_bmix_int <- function(params,
                         pw1, pa1, pb1, py1, pz1,
                         pw2, pa2, pb2, py2, pz2) {

  (pw1 * pw2) * dbeta(params[1], pa1, pb1) * dbeta(params[1] + params[2], pa2, pb2) +
  (pw1 * (1 - pw2)) * dbeta(params[1], pa1, pb1) * dbeta(params[1]+params[2], py2, pz2) +
  ((1 - pw1) * pw2) * dbeta(params[1], py1, pz1) * dbeta(params[1]+params[2], pa2, pb2) +
  ((1-pw1) * (1-pw2)) * dbeta(params[1], py1, pz1) * dbeta(params[1]+params[2], py2, pz2)
}
```

**Program 8. R: Vectorised integrand for the posterior treatment effect $(\delta)$ CrI.**

```
delta_b_cdf <- function(diff,
                        pw1, pa1, pb1, py1, pz1,
                        pw2, pa2, pb2,  py2, pz2){

  if(diff <= 0){

    prob <- cubature::hcubature(f = fun_bmix_int,
                                lowerLimit = c(-diff, -1),
                                upperLimit = c(1, diff),
                                tol = 1e-5,
                                pa1 = pa1, pb1 = pb1, py1 = py1, pz1 = pz1,
                                pa2 = pa2, pb2 = pb2, py2 = py2, pz2 = pz2,
                                pw1 = pw1, pw2 = pw2)$integral

  }else{

    prblz <- cubature::hcubature(f = fun_bmix_int,
                                 lowerLimit = c(0, -1),
                                 upperLimit = c(1, 0),
                                 tol = 1e-5,
                                 pa1 = pa1, pb1 = pb1, py1 = py1, pz1 = pz1,
                                 pa2 = pa2, pb2 = pb2, py2 = py2, pz2 = pz2,
                                 pw1 = pw1, pw2 = pw2)$integral

    prbgzero <- cubature::hcubature(f = fun_bmix_int,
                                    lowerLimit = c(0, 0),
                                    upperLimit = c(1, diff),
                                    tol = 1e-5,
                                    pa1 = pa1, pb1 = pb1, py1 = py1, pz1 = pz1,
                                    pa2 = pa2, pb2 = pb2, py2 = py2, pz2 = pz2,
                                    pw1 = pw1, pw2 = pw2)$integral

    prob <- prblz + prbgzero
  }

  return(prob)

}
```

**Program 9. R: Integrating to obtain the CDF of the posterior treatment effect.**

```
quantile_bdiff_kernal <- function(diff,
                                  pw1, pa1, pb1, py1, pz1,
                                  pw2, pa2, pb2, py2, pz2,
                                  prob){

     delta_b_cdf(diff,
                 pw1, pa1, pb1, py1, pz1,
                 pw2, pa2, pb2, py2, pz2) - prob
}
quantile_bdiff <- function(diff,
                           pw1, pa1, pb1, py1, pz1,
                           pw2, pa2, pb2, py2, pz2,
                           prob){

  qle <- uniroot(quantile_bdiff_kernal,
                 lower = -1,
                 upper = 1,
                 pw1 = pw1, pa1 = pa1, pb1 = pb1, py1 = py1, pz1 = pz1,
                 pw2 = pw2, pa2 = pa2, pb2 = pb2, py2 = py2, pz2 = pz2,
                 prob = prob)$root

  return(qle)
}
```

**Program 10. R: Calculate the quantile for the posterior treatment effect**

```
## Vague hyperparameters
y1 <- 1; z1 <- 1
y2 <- 1; z2 <- 1
w1 <- 0.3; w2 <- 0.3

## Posterior updates
pw1 <- post_bw(w1, x1, n1, a1, b1)
pa1 <- a1 + x1; pb1 <- b1 + n1 - x1
py1 <- y1 + x1; pz1 <- z1 + n1 - x1

pw2 <- post_bw(w2, x2, n2, a2, b2)
pa2 <- a2 + x2; pb2 <- b2 + n2 - x2
py2 <- y2 + x2; pz2 <- z2 + n2 - x2

quantile_bdiff(diff,
              pw1, pa1, pb1, py1, pz1,
              pw2, pa2, pb2, py2, pz2,
              prob = 0.95)
0.2961478
```

**Program 21. R: 95th quantile of the posterior treatment effect.**

## SAS ANALYSIS - BINARY DATA

The calculation of the posterior probability of a treatment effect performed in IML within SAS is very similar to R. We start the IML SAS environment with “*proc iml;”* and close with “*finish;*”. The hyperparameters and data are defined as global parameters, so that we can identify the parameter we are integrating over in the SAS numerical integration routine Quad.

In Program 14, after calling the Quad routine, the result is passed into a data frame called margpost.

```
proc iml;

start dynw1(w) global(x1, n1, a1, b1);

      /* Dynamic weight one */

      wup1 = (w * beta(x1 + a1, n1 - x1 + b1)) / beta(a1, b1);
      wup2 = (1- w) * beta(x1 + 1, n1 - x1 + 1);

      w1 = wup1 / (wup1 + wup2);

      return(w1);
```

```
finish;


start dynw2(w) global(x2, n2, a2, b2);

      /* Dynamic weight two */

      wup1 = (w * beta(x2 + a2, n2 - x2 + b2)) / beta(a2, b2);
      wup2 = (1- w) * beta(x2 + 1, n2 - x2 + 1);

      w2 = wup1 / (wup1 + wup2);

      return(w2);

finish;
```

**Program 12. SAS: Posterior weights for control and treatment of the RMP for binary data.**

```
start fun(thetat) global(pw1, pw2,
                         x1, n1, a1, b1,
                         x2, n2, a2, b2);

   int = (pw1 * cdf('BETA', thetat, x1 + a1, n1 - x1 + b1) + (1 - pw1) *
          cdf('BETA', thetat, x1 + 1, n1 - x1 + 1)) *
         (pw2 * pdf('BETA', thetat, x2 + a2, n2 - x2 + b2) + (1 - pw2) *
          pdf('BETA', thetat, x2 + 1, n2 - x2 + 1));

   return(int);

finish;
```

**Program 13. SAS: Integrand for posterior probability of an efficacious treatment effect.**

```
/* Hyperparameters for prior */
a1 = 110; b1 = 250;
a2 = 175; b2 = 190;

/* Data */
x1 = 10; n1 = 30;
x2 = 15; n2 = 30;

pw1 = dynw(0.3);
pw2 = dynw(0.3);

interval = {0 1};
call quad(postp, "fun", interval) eps=1E-8;

create margpost from postp[colname="Posterior_prob"];
      append from postp;
close margpost;

quit;
```

**Program 14. SAS: Numerical integration to calculate the posterior probability of a significant treatment effect (risk difference).**

The tipping point calculation is not as straight forward in SAS, because there is no in-built root finder routine. Instead, a function which performs the binary search method is created called *rfinder*.

```
start tippingfun(wloc) global(pw1, pw2, a1, b1, x1, n1, a2, b2, x2, n);

      pw1 = dynw1(wloc);
      pw2 = dynw2(wloc);

      interval = {0  1};
      call quad(z, "fun_int", interval) eps=1E-10 peak=0.5;
      return(z);

finish;

start rfinder(a, b, tpv); /* tpv is the threshold for a significant trt effect */

      dx = 1e-6; dy = 1e-6;
      do i = 1 to 500;
         c = (a + b) / 2;
         if abs(tippingfun(c) - tpv) < dy | (b - a)/2 < dx then
            return(c);
         if (tippingfun(a) - tpv) * (tippingfun(c)-tpv) > 0 then
            a = c;
         else b = c;
      end;

      return (.);

finish;

tpp = rfinder (0, /* lower limit */
               1, /* upper limit */
               0.975);

quit;
```

**Program 15. SAS: Minimum prior weight w for a significant treatment effect.**

Below is the SAS code for calculating a $q$th CI from the posterior distribution of the treatment and control parameters. We define the posterior weight *postw* as a global parameter alongside the hyperparameters and the data, before we call *rfinder_ci* to calculate the integral.

```
start ci_integrand(theta) global(pw1, x1, n1, a1, b1);

    int = pw1 * pdf('BETA', theta, x1 + a1, n1 − x1 + b1) +
          (1 − pw1) * pdf('BETA', theta, x1 + 1, n1 − x1 + 1);

   return(int);

finish;

start qauntile_fun(ul) global(pw1, a1, b1, x1, n1);

    interval = 0 || ul;
    call quad(z, "ci_integrand", interval) eps=1E-10 peak=0.5;

    return(z);

finish;
```

**Program 16. SAS: Integration for the CrI for the posterior for control $(\boldsymbol{\theta_c})$ and treatment $(\boldsymbol{\theta_t})$ parameters.**

```
start rfinder_ci(lower, upper, tpv);

    dx = 1e-5; dy = 1e-5;
    do i = 1 to 500;
        c = (lower + upper)/2;
        if abs(qauntile_fun(c) - tpv) < dy | (upper - lower)/2 < dx then
            return(c);
        if (qauntile_fun(lower) - tpv) * (qauntile_fun(c)-tpv) > 0 then lower = c;
            else upper = c;
    end;

    return (.);

finish;
```

**Program 17. SAS: Adapt the root finder for the quantile function.**

```
a1 = 110; b1 = 250;
x1 = 10; n1 = 30;
postw = dynw1(0.3);

quantile_half_p1  = rfinder_ci(0, /* Lower limit */
                               1, /* Upper limit */
                               0.99); /* Probability for quantile */
```

**Program 18. SAS: 99th posterior CrI for control parameter $(\theta_c)$. The same approach can be used to obtain the CrI for the treatment parameter $(\theta_t)$**

Tor calculate the $q$th CrI a double integration is required but there is no in-built routine which supports this. One option is to use the Quad function within a nested call, however the routine will return an error if the integrand appears to be identical to 0, which will happen in our example due to underflow. Rather than manually adjust the region of integration and risk underestimating the probability, a simple numerical integration function is written.

```
start fun_mix_inner(t, delta) global(pw1, pw2, x1, n1, a1, b1, x2, n2, a2, b2);

  int = (pw1 * pw2) * pdf('BETA', delta, x1 + a1, n1 - x1 + b1) *
            pdf('BETA', t + delta, x2 + a2, n2 - x2 + b2)+
        (pw1 * (1 - pw2)) * pdf('BETA', delta, x1 + a1, n1 - x1 + b1) *
            pdf('BETA', t + delta, x2 + 1, n2 - x2 + 1)+
        ((1 - pw1) * pw2) * pdf('BETA', delta, x1 + 1,n1 - x1 + 1) *
            pdf('BETA', t + delta, x2 + a2, n2 - x2 + b2)+
        ((1 - pw1) * (1 - pw2)) * pdf('BETA', delta, x1 + 1, n1 -  x1 + 1) *
            pdf('BETA', t + delta, x2 + 1,n2 - x2 + 1);

  return(int);

finish;
```

**Program 19. SAS: Integrand for the treatment effect CI.**

```
start integration_man(diff) global(pw1, pw2, x1, n1, a1, b1, x2, n2, a2, b2);

      /* Define integration parameters per region*/
      if diff < 0 then do;
          a = -diff; b = 1; /* limits delta */
          c = -1; d = diff; /* limits t */
      end;
      else do;
          a = 0; b = 1;
          c = -1; d = diff;
      end;

      /* Set resolution for numerical integration */
      nx = 100; ny = 100;
      dx = (b - a) / nx;
      dy = (d - c) / ny;

      /* Double integration using nested loops */
      integral = 0;

      do i = 1 to nx;
          theta = a + (i - 0.5)* dx; /* Midpoint rule for x */
          do j = 1 to ny;
              t = c + (j - 0.5)* dy; /* Midpoint rule for y */
               integral = integral + fun_mix_inner(t, theta)* dx * dy;
            end;
         end;

         return(integral);

      finish;
```

**Program 20. SAS: Numerical integration of double integral to obtain the CDF for the posterior treatment effect.**

```
start rfinder_delta(lower, upper, tpv);

    dx = 1e-6; dy = 1e-4;
    do i = 1 to 500; /** max iterations **/
        c = (lower + upper)/2;
        if abs(integration_man(c) - tpv) < dy | (upper - lower)/2 < dx then
        return(c);
        if (integration_man(lower) - tpv)# (integration_man(c)-tpv) > 0 then lower = c;
        else upper = c;
    end;

    return (.); /** no convergence **/

finish;
```

**Program 21. SAS: Quantile from the posterior treatment effect**

```
quant_delta = rfinder_delta(-1, /* Lower limit */
                            1,  /* Upper limit */
                            0.99); /* Probability for quantile*/
```

**Program 22. SAS: After defining the data and the posterior weights as global parameters, as in Program 14, the 95th quantile of the posterior treatment effect is calculated by calling the *rfinder_delta* function.**

## R ANALYSIS - COUNT OUTCOME

For count data, the posterior weight and integrand need to be adjusted for the steps outlined above.

The hyperparameters $y$ and $z$ define the vague component of the mixture prior, $n$ is the number of patients with the arm and $sx$ is the sum of the individual count data of these individuals.

```
post_cw <- function(w, sx, n, a, b, y, z){
  wup1 = w * b**a * gamma(sx + a) / ((n + b)**(sx + a) * gamma(a))
  wup2 = (1 - w) * z**y * gamma(sx + y) / ((n + z)**(sx + y) * gamma(z))

  dw = wup1 / (wup1 + wup2)

  return(dw)
}
```

**Program 23. R: Posterior weight of the RMP for count data.**

```
int_ctrt_effect <- function(thetat, w,
                            a1, b1, y1, z1, sx1, n1,
                            a2, b2, y2, z2, sx2, n2) {

  (post_cw(w,sx1,n1,a1,b1,y1,z1) * pgamma(thetat, shape=a1+sx1,rate=b1+n1) +
     (1 - post_cw(w,sx1,n1,a1,b1,y1,z1)) * pgamma(thetat,shape=y1+sx1,rate=z1+n1)) *
  (post_cw(w,sx2,n2,a2,b2,y2,z2) * dgamma(thetat,shape=a2 + sx2,rate=b2+ n2) +
     (1 - post_cw(w, sx2,n2,a2,b2,y2,z2)) * dgamma(thetat,shape=y2+sx2,rate=z2+n2))
}
```

**Program 24. R: Integrand for posterior probability of an efficacious treatment effect (count).**

Below is the additional code for calculating the $q$th credible interval for the control and treatment parameters.

```
ci_integrand_gamma <- function(theta, w, a, b, y, z, sx, n) {

  (post_cw(w, sx, n, a, b, y, z) * dgamma(theta, a + sx, b + n) +
   (1 - post_cw(w, sx, n, a, b, y, z)) *  dgamma(theta, y + sx, z + n))

}

qauntile_cfun <- function(ul, w, a, b, y, z, sx, n, prob){

  cubature::hcubature(f = ci_integrand_gamma,
                      lowerLimit = 0,
                      upperLimit = ul,
                      tol = 1e-9,
                      w = w,
                      a = a, b = b, y = y, z = z,
                      sx = sx, n = n)$integral - prob
}
```

**Program 25. R: Integrand for control and treatment parameters posterior CrI.**

```
quantile_cmix <- function(w, a, b, y, z, sx, n, upper, prob){

  qle <- uniroot(qauntile_cfun, lower = 0, upper = upper,
                 w = w, a = a, b = b, y = y, z = z, sx = sx, n = n, prob = prob)$root

  return(qle)
}
```

**Program 26. R: Calculate $q$th equal tailed posterior CrI.**

The argument upper determines the upper limit of the interval to be searched for the root. As this needs to be real number, this can be set to a large number relative to $sx/n$.

```
fun_cmix_integrand <- function(params,
                               pw1, pa1, pb1, py1, pz1,
                               pw2, pa2,  pb2,  py2, pz2) {

  (pw1 * pw2) * dgamma(params[1],pa1,pb1) * dgamma(params[1]+params[2],pa2,pb2) +
  (pw1 * (1-pw2)) * dgamma(params[1],pa1,pb1) * dgamma(params[1]+params[2],py2,pz2) +
  ((1-pw1) * pw2) * dgamma(params[1],py1,pz1) * dgamma(params[1]+params[2],pa2,pb2) +
  ((1-pw1) * (1-pw2)) * dgamma(params[1],py1,pz1) * dgamma(params[1]+params[2],py2,pz2)

}
```

**Program 27.R: Vectorised integrand for posterior treatment effect CrI.**

```
delta_c_cdf <- function(diff,
                        pw1, pa1, pb1, py1, pz1,
                        pw2, pa2, pb2, py2, pz2){

  if(diff <= 0){

    prob <- cubature::hcubature(f = fun_cmix_integrand,
                                lowerLimit = c(-diff, -Inf),
                                upperLimit = c(Inf, diff),
                                tol = 1e-5,
                                pa1 = a1, pb1 = b1, py1 = py1, pz1 = pz1,
                                pa2 = pa2, pb2 = pb2, py2 = py2, pz2 = pz2,
                                pw1 = pw1, pw2 = pw2)$integral

  }else{

    prblz <- cubature::hcubature(f = fun_cmix_integrand,
                                 lowerLimit = c(-diff, -Inf),
                                 upperLimit = c(Inf, 0),
                                 tol = 1e-5,
                                 pa1 = pa1, pb1 = pb1, py1 = py1, pz1 = pz1,
                                 pa2 = pa2, pb2 = pb2, py2 = py2, pz2 = pz2,
                                 pw1 = pw1, pw2 = pw2)$integral

    prbgzero <- cubature::hcubature(f = fun_cmix_integrand,
                                    lowerLimit = c(0, 0),
                                    upperLimit = c(Inf, diff),
                                    tol = 1e-5,
                                    pa1 = pa1, pb1 = pb1, py1 = py1, pz1 = pz1,
                                    pa2 = pa2, pb2 = pb2, py2 = py2, pz2 = pz2,
                                    pw1 = pw1, pw2 = pw2)$integral

    prob <- prblz + prbgzero
  }

  return(prob)

}
```

**Program 28. R: Integrating to obtain the CDF of the posterior treatment effect.**

```
qauntile_cdiff_kernal <- function(diff,
                                  pw1, pa1, pb1, py1, pz1,
                                  pw2, pa2, pb2, py2, pz2,
                                  prob){

  delta_c_cdf(diff,
              pw1, pa1, pb1, py1, pz1,
              pw2, pa2, pb2, py2, pz2) - prob
}

quantile_cdiff <- function(diff,
                           pw1, pa1, pb1, py1, pz1,
                           pw2, pa2, pb2, py2, pz2,
                           lower, upper,
                           prob){

  qle <- uniroot(quantile_cdiff_kernal,
                 lower = lower,
                 upper = upper,
                 pw1 = pw1, pa1 = pa1, pb1 = pb1, py1 = py1, pz1 = pz1,
                 pw2 = pw2, pa2 = pa2, pb2 = pb2, py2 = py2, pz2 = pz2,
                 prob = prob)$root

  return(qle)
}
```

**Program 29. R: Calculate the quantile for the posterior treatment effect**

### R ANALYSIS - CONTINUOUS OUTCOME

The calculations for the continuous data are easily performed by using the same steps described for the count data above, with the adjusted posteriors. Given the support for the treatment and control parameters is over the real line, there is no need to adjust the region in the numerical integration for the treatment effect.

## CONCLUSION

This article describes how to calculate the exact probabilities for a treatment effect in R and SAS from Bayesian posteriors, where RMPs have been used to dynamically borrow information and minimise prior-data conflict. To help with inference, a prior weight tipping point and posterior credible intervals are also obtained using exact probabilities. It is clear from the code blocks that there is more support in R than SAS when calculations become slightly more involved. A root finder within the IML environment is absent, and it is often difficult to extend the single numerical integration routine Quad beyond a single integral. Hopefully, this article helps both to understand and implement fast and accurate inference using posteriors from Bayesian borrowing.

## REFERENCES

Gelman, A., Carlin, J., Stern, H., Dunson, D., Aki, V., & Rubin, D. (2021). *Bayesian Data Analysis.* Electronic Version.

Nelder, J. A., & Mead, R. (1965). A simplex method for function minimization. *The Computer Journal*, 27.

Schmidli, H., & Neuenschwander, B. (2014). Robust meta-analytic-predictive priors in clinica ltrials with historical control information. *Biometrics*, 1023-1032.

Scott, D., & Lewin, A. (2024). Borrowing from historical control data in a Bayesian time-to-event model with flexible baseline hazard function. *arXiv*, 1-20.

Sharon, L., Brian, H., Alexander, K., & Joseph, K. (2021). Calibrated Dynamic Borrowing Using Capping Priors. *Journal of Biopharmaceutical Statistics*, 1-22.

Spiegelhalter, D. J., Abrams, K. R., & P, M. J. (2003). *Bayesian approaches to clinical trials and health-care evaluation.* Cambridge: John Wiley & Sons.

## ACKNOWLEDGMENTS

I wish to thank Enti Spata for discussions on this topic and support with the abstract.

## CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Darren Scott
AstraZeneca, The Discovery Centre, Trumpington, Cambridge, CB2 0AA
Darren.scott@astrazeneca.com