trialr: Bayesian Clinical Trial Designs in R and Stan

Kristian Brock

Cancer Research UK Clinical Trials Unit, University of Birmingham

Abstract

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Keywords: clinical trial, bayesian, dose finding, R, Stan.

1. Introduction

Clinical trials are sequential medical experiments in humans. Their collective goal is to identify experimental therapies that offer sufficient benefit to warrant use in standard clinical practice. Fundamental to this is the estimation of treatment efficacy and safety. Many experimental designs have been proposed to these ends.

Stan (Carpenter, Gelman, Hoffman, Lee, Goodrich, Betancourt, Brubaker, Li, and Riddell 2016) is a probabilistic programming language allowing full Bayesian inference via Markov chain Monte Carlo (MCMC) sampling. We present in this article the **trialr** package implementing several clinical trial designs in R and Stan.

Investigation of safety is the major preoccupation of early phase clinical trials. A typical phase I trial would conduct a dose-finding experiment to identify a dose of a novel therapy that is associated with an acceptable adverse event (AE) rate. The term *dose* here is used loosely to reflect the general intensity of treatment. In most cases, dose would have the intuitive property of reflecting the quantity of a molecule that is administered to the patient in a treatment window. It could, however, represent the frequency or the duration of administration, or a combination of the two. It could also contain information on those aspects of several experimental therapies that are given together.

In many instances, a dose-finding experiment would be designed to incorporate dose escalation, where successively higher doses are sequentially investigated in cohorts of patients. Such experimental designs typically assess the patient-level presence or absence of *dose-limiting toxicity* (DLT), the manifestation of at least one of potentially many prespecified adverse events that would be considered sufficiently serious to halt treatment administration. For illustration, treatment could commence at a low dose for which there is reasonable expectation of safety. If a sufficiently low number of these patients experience DLT, the dose may be escalated for the next group of patients. Dose-finding trials typically continue sequentially, with successive doses being chosen in response to the outcomes observed thus far. As such, dose-finding trials are one of the more simple types of *adaptive* trial, where some treatment parameter for patients (here, dose) is affected by the outcomes observed in earlier patients.

A key assumption of dose escalation and de-escalation solely with reference to DLT is that a higher dose is more likely to provide clinical benefit. A higher dose is accepted as being more attractive than a lower dose so long as both are associated with an acceptable rate of DLT, without any reference to the level of efficacy expected or observed. Logic, and the toxicologists' adage *sola dosis facit venenum* or *the dose makes the poison*, dictate that the probability of toxicity increases in dose. These so-called *monotonicity assumptions* are the cornerstones of traditional phase I dose-finding. The highest dose with predicted DLT probability less than or close to some target rate is referred to as the maximum tolerable dose (MTD).

Identification of the MTD is a typical objective of phase I trials. There are several experimental designs proposed to achieve this goal. The most common is 3+3 (Carter 1973), a simple algorithm that provides rules for changing dose based on the DLT outcomes of cohorts of 3 patients. It has been widely criticised for its relatively poor operating performance, its non-statistical nature, and its lack of flexibility (O'Quigley, Pepe, and Fisher 1990; O'Quigley and Zohar 2006; Iasonos, Wilton, Riedel, Seshan, and Spriggs 2008; Le Tourneau, Lee, and Siu 2009). Despite this, it stubbornly persists as the most frequently used dose-finding design (Rogatko, Schoeneck, Jonas, Tighiouart, Khuri, and Porter 2007; Chiuzan, Shtaynberger, Manji, Duong, Schwartz, Ivanova, and Lee 2017), due in no small part to its simplicity and familiarity.

There are also many statistical designs that seek the MTD (O'Quigley *et al.* 1990; Tighiouart and Rogatko 2010; Ji and Wang 2013, amongst others). The Continual Reassessment Method (CRM) by O'Quigley *et al.* (1990) is particularly noteworthy because of the great amount of innovation it has facilitated, including the time-to-event CRM (TITE-CRM) approach (Cheung and Chappell 2000) for studying late-onset toxic events.

Modern cancer drugs like targeted therapies and immunotherapies do not necessarily support the monotonic efficacy assumption. For instance, two doses of the immunotherapy drug pembrolizumab that differed by a factor of five were shown in a large randomised trial to have very similar response rates (Herbst, Baas, Kim, Felip, Pérez-Gracia, Han, Molina, Kim, Arvis, Ahn, Majem, Fidler, De Castro, Garrido, Lubiniecki, Shentu, Im, Dolled-Filhart, and Garon 2016). When conducting dose-finding in therapies like these, there may be motivation to scrutinise doses by efficacy and toxicity outcomes. There are several designs that perform this task (Braun 2002; Thall and Cook 2004; Zhang, Sargent, and Mandrekar 2006; Wages and Tait 2015).

Traditional phase II trials take the dose recommended in phase I and investigate early signs of treatment activity or efficacy. The objective here is to screen out treatments that are

obviously ineffective. These trials can generally be single-arm or randomised. The onus is on conducting the trial and disseminating the results quickly so that onward clinical research can proceed promptly. As such, sample sizes tend to be small and outcomes that can be assessed over the short-term are generally preferred.

Finally, phase III trials seek to provide evidence good enough to affect medical practice. Most commonly, phase III trials test whether an experimental treatment is superior to the current standard of care. These trials commonly randomise patients to one of several treatment arms, use clinically-relevant long-term outcome measures, and have large sample sizes.

Bayesian methods in clinical trials are used predominantly, although not exclusively, in phases I and II. This is due in part to the limited amount of information available from the low sample sizes used. Using prior information to complement an analysis can be desirable, and the model-based methods can benefit computationally from the Bayesian approach.

In this article, we focus on three clinical trial models implemented in **trialr** under the Bayesian paradigm using R and Stan:

- the CRM and TITE-CRM models for traditional toxicity-oriented dose-finding;
- the EffTox model for efficacy and toxicity dose-finding;
- the Augmented Binary (AugBin) method (Wason and Seaman 2013) for analysing dichotomous response in phase II cancer trials whilst retaining the continuous information in measurements of tumour size.

There are several other R-packages that implement the Bayesian CRM in R, including **dfcrm** (Cheung 2013, 2011), **bcrm** (Sweeting, Mander, and Sabin 2013), and **crmPack** (Sabanes Bove, Yin Yeung, Palermo, and Jaki 2018). **dfcrm** uses a non-MCMC approach. **bcrm** and **crmPack** support MCMC via JAGS and WinBUGS. To the best of our knowledge, **trialr** is the only package on CRAN to implement the CRM and EffTox in Stan, and the only package on CRAN to implement the AugBin method.

To avoid large-scale repetition of material in previous publications on CRM software (Sweeting *et al.* 2013; Cheung 2011), we focus the dose-finding aspects of this article on functions related to dose-transtions pathways (Yap, Billingham, Cheung, Craddock, and O'Quigley 2017; Brock, Billingham, Copland, Siddique, Sirovica, and Yap 2017), a tool to aid trial planning. This approach is considered for both the CRM and EffTox designs. Then we introduce the AugBin method for single arm trials with two post-baseline analyses.

trialr was first added to CRAN in 2017 with Stan implementations of EffTox, a hierarchical model for analysing responses to a single drug in several related subtypes of a disease Thall, Wathen, Bekele, Champlin, Baker, and Benjamin (2003), and a joint-model for analysing co-primary efficacy and toxicity outcomes when baseline covariate information is available. Successive updates have seen the addition of the CRM and AugBin models. **trialr** relies heavily on **tidyverse** (Wickham 2017) packages to create a modern workflow in R for Bayesian approaches to early-phase clinical trials.

model	Link function	Parameters
`empiric`	$F(d, \beta) = d^{\exp \beta}$	normal prior on β
`logistic`	$F(d, \beta) = 1/(1 + \exp(-a_0 - \exp(\beta)d))$	a_0 fixed, normal prior on β
`logistic_gamma`	$F(d, \beta) = 1/(1 + \exp(-a_0 - \beta d))$	a_0 fixed, gamma prior on β
`logistic2`	$F(d, \alpha, \beta) = 1/(1 + \exp(-\alpha - \exp(\beta)d))$	normal priors on α and β

Table 1: CRM models implemented in **trialr**.

2. Methodology

2.1. CRM

The Continual Reassessment Method (CRM) was originally published by O'Quigley *et al.* (1990) to conduct dose-finding trials seeking a maximum tolerable dose (MTD). It is a truly seminal design, with many variants appearing over the years to handle different clinical scenarios, such as related groups (O'Quigley and Paoletti 2003), late-onset toxicity (Cheung and Chappell 2000), efficacy and toxicity outcomes (Braun 2002; Zhang *et al.* 2006), and more. In this section, we focus on the original Bayesian design that seeks the MTD in an homogeneous patient group.

In a trial of doses $d_1, ..., d_K$, the investigators provide their best guess of the probability of DLT at each dose to create a toxicity *skeleton* $(p_1, ..., p_K)$. CRM assumes monotonically increasing toxicity risk so we stipulate that $p_1 < ... < p_K$.

The probability of DLT at dose x is modelled to be $F(x, \theta)$, where F is a smooth mathematical function, and θ is a general vector of parameters. Different variants of this model use different forms for F, and different prior distributions on θ. The CRM variants currently implemented in `trialr` are listed in Table 1.

A core assumption of the CRM and dose-escalation methods in general, is that the probability of toxicity increases with dose. The slope parameter β is exponentiated in some of the models in Table 1 to ensure a positive value. In the model with gamma prior, this step is not necessary because the prior constrains the posterior to take only positive values.

For the purposes of statistical modelling, the explanatory dose variables d_k are codified such that

$$p_k = F(d_k, \hat{\theta}),$$

where $\hat{\theta}$ is the vector of prior expected values of the elements of θ. That is, the d_k are derived so that, using the prior expected value of each model parameter, the prior expected probability of DLT at d_k is p_k. Cheung (2011) calls the d_k *dose-labels* to distinguish them from the actual dose units like "10 mg" that a clinician might actually administer to a patient.

For patient i, let Y_i be a random variable taking values $\{0, 1\}$ reflecting the absence and presence of DLT, and $x_i \in \{d_1, ..., d_K\}$ be the dose administered. After the evaluation of I patients, the aggregate likelihood is:

$$L_I(\theta) = \prod_{i=1}^{I} \{F(x_i, \theta)\}^{Y_i} \{1 - F(x_i, \theta)\}^{1-Y_i}$$

This quantity can either be maximised with respect to θ to obtain the maximum likelihood estimate, or augmented with priors and analysed in a Bayesian way. CRM was originally published as a Bayesian design and that is the approach we take in `trialr` using Stan.

Cheung and Chappell (2000) introduced the notion of weighting the Y_i to reflect the extend to which they have been evaluated. For instance, if a patient is 50% of the way through the evaluation period without having yet experienced toxicity, we might interpret them as providing an observation $Y_i = 0$ with weight 0.5. This leads to the weighted likelihood statement

$$L_I(\theta) = \prod_{i=1}^{I} \{w_i F(x_i, \theta)\}^{Y_i} \{1 - w_i F(x_i, \theta)\}^{1-Y_i},$$

as used in the TITE-CRM design. This insight facilitated the study of long-term toxicity events, without the requirement that each patient complete the evaluation period before they contribute information to the dose selection analysis.

CRM models are accessed in **trialr** primarily through the `stan_crm` function, as described below.

2.2. EffTox

The EffTox design (Thall and Cook 2004; Thall, Cook, and Estey 2006; Thall, Herrick, Nguyen, Venier, and Norris 2014) conducts dose-finding clinical trials where both efficacy and toxicity events guide dose selection. This contrasts methods like 3+3 and CRM where dose selection is determined by toxicity events only. It is no longer assumed that higher doses are implicitly more likely to be beneficial to patients. Instead, EffTox assigns doses attractiveness or *utility* scores to doses that trade-off the estimated probabilities of efficacy and toxicity. The design iteratively seeks the dose with the highest utility score. We will call this the *optimum biological dose* (OBD) to distinguish it from the MTD and the pratice of escalating under assumed montonically increasing efficacy. When recommending the next dose, EffTox does not blindly recommend the prevailing OBD. Instead, it uses rules that forbid the skipping of untested doses. Full details are given in the authors' publications but we provide a brief recap of the statistical methodology here.

For doses $(y_1, ..., y_n)$, the authors define standardised doses $(x_1, ..., x_n)$ using the transform

$$x_j = \log y_j - \sum_{k=1}^{n} \frac{\log y_k}{n}$$

The model uses six parameters, $\theta = (\alpha, \beta, \gamma, \zeta, \eta, \epsilon)$. At dose x, the marginal probabilities of toxicity and efficacy are estimated by the logit models:

$$\text{logit } \pi_T(x, \theta) = \alpha + \beta x$$

and

$$\text{logit } \pi_E(x, \theta) = \gamma + \zeta x + \eta x^2$$

Let $\boldsymbol{Y} = (Y_E, Y_T)$ be indicators of binary efficacy and toxicity events and let $\pi_{a,b}(x, \theta) = \Pr(Y_E = a, Y_T = b | x, \theta)$ for $a, b \in \{0, 1\}$. The efficacy and toxicity events are associated by the joint probability function

$$\pi_{a,b}(x, \theta) = (\pi_E)^a (1 - \pi_E)^{1-a} (\pi_T)^b (1 - \pi_T)^{1-b} + (-1)^{a+b} (\pi_E)(1 - \pi_E)(\pi_T)(1 - \pi_T) \frac{e^{\psi} - 1}{e^{\psi} + 1}$$

,

where ψ is a parameter that measures the strength of association between the co-primary outcomes, and (x, θ)-notation has been suppressed on the right-hand side for brevity. Normal priors are specified for the elements of θ.

At each dose update decision, the dose x is acceptable if

$$\Pr \{ \pi_T(x, \boldsymbol{\theta}) < \overline{\pi}_T | \mathcal{D} \} > p_T,$$

$$\Pr \{ \pi_E(x, \boldsymbol{\theta}) > \underline{\pi}_E | \mathcal{D} \} > p_E$$

and is no more than than one position below the lowest dose-level given and no more than one position above the highest dose-level given. The effect of these last two criteria is that untried doses may not be skipped in escalation or de-escalation. The values $\underline{\pi}_E$ and $\overline{\pi}_T$ are provided by the investigators with reference to what is acceptable in the clinical scenario, given the set of alternative treatments. Thall and Cook (2004) selected $p_E = p_T = 0.1$ so that doses were excluded from the acceptable set only if we are very sure that they are either inactive or excessively toxic.

The utility of dose x, with efficacy $\pi_E(x, \theta)$ and toxicity $\pi_T(x, \theta)$ is

$$u(\pi_E, \pi_T) = 1 - \left(\left(\frac{1 - \pi_E}{1 - \pi_{1,E}^*} \right)^p + \left(\frac{\pi_T}{\pi_{2,T}^*} \right)^p \right)^{\frac{1}{p}}$$

where p is calculated to intersect the points $(\pi_{1,E}^*, 0)$, $(1, \pi_{2,T}^*)$ and $(\pi_{3,E}^*, \pi_{3,T}^*)$ in the efficacy-toxicity plane. We will refer to these as *hinge points* but that is not nomenclature proposed by the authors.

At the dose selection decision, the dose-level from the subset of doses that are acceptable with maximal utility is selected to be given to the next patient or cohort. If there are no acceptable doses, the trial stops and no dose is recommended.

There are several published EffTox examples, including explanations and tips on parameter choices (Thall and Cook 2004; Thall *et al.* 2006, 2014; Brock *et al.* 2017).

The MD Anderson Cancer Center publishes software[1] to perform calculations and simulations for EffTox trials. The software is available for Windows in compiled-form. Thus, trialists cannot run the software on Mac or Linux unless via a virtual machine or emulator. Furthermore,

---

[1]see https://biostatistics.mdanderson.org/SoftwareDownload/SingleSoftware/Index/2

they may not alter the behaviour of the model. (Brock *et al.* 2017) describe a clinical trial scenario where some alteration to the default model behaviour would have been preferable. It was this that prompted the author to write the open-source implementation provided in `trialr`.

2.3. Augmented Binary method

In oncology clinical trials, tumours are often measured at baseline and repeatedly post-baseline to gauge the response to therapy. The Response Evaluation Criteria in Solid Tumors (RE-CIST)(Eisenhauer, Therasse, Bogaerts, Schwartz, Sargent, Ford, Dancey, Arbuck, Gwyther, Mooney, Rubinstein, Shankar, Dodd, Kaplan, Lacombe, and Verweij 2009) define an algorithm for classifying into categories the tumour measurements compared to baseline. Up to five so-called *target lesions* are identified at baseline and the sum of their largest diameters is recorded. A *complete response* is said to have occurred post-baseline when all tumour lesions disappear. A *partial response* occurs when the aggregate diameter of the target lesions shrinks by at least 30% and no new lesions are detected. Commonly, the complete and partial response events are combined to form an *objective response* category, and this is used to calculate the binary response rate in the trial.

Statisticians are aware that this dichotomisation of a continuous underlying measure leads to information loss and less-efficient analysis. However, responder analysis is endemic in cancer trials. In response, Wason and Seaman (2013) introduced the Augmented Binary model. It analyses the continuous log-tumour size ratio, whilst building in mechanisms for so-called *non-shrinkage failures*, like the manifestation of new lesions, that preclude success, even if requisite tumour shrinkage is observed. Wason & Seaman introduced versions of their model for single-arm and randomised groups trials. We describe here the single-arm variant.

Let z_{0i} be the tumour size at baseline for patient i, z_{1i} the size at interim, and z_{2i} the size at the end of the trial. Let $(y_{1i}, y_{2i}) = (\log(z_{1i}/z_{0i}), \log(z_{2i}/z_{0i}))$ be the log tumour size ratios with respect to baseline. Furthermore, the authors define variables D_{1i} and D_{2i} to represent the observation of non-shrinkage failure at interim and final assessments. Finally, they define S_i to be a composite success indicator. Using 30% shrinkage as the threshold for success, $S_i = 1$ if $D_{1i} = D_{2i} = 0$ and $y_{2i} < \log(0.7)$.

The authors specify the following statistical models. The tumour shrinkage variables are modelled using a bivariate normal distribution:

$$(y_{1i}, y_{2i})^T \sim N\left((\mu_{1i}, \mu_{2i})^T, \Sigma\right),$$

where $\mu_{1i} = \alpha + \gamma z_{0i}$ and $\mu_{2i} = \beta + \gamma z_{0i}$, and Σ is assumed to be unstructured.

The non-shrinkage failures are estimated using logit models:

$$\text{logit}\,\Pr(D_{1i} = 1 | Z_{0i}) = \alpha_{D1} + \gamma_{D1} z_{0i}$$

and

$$\text{logit}\,\Pr(D_{2i} = 1 | D_{1i} = 0, Z_{0i}, Z_{1i}) = \alpha_{D2} + \gamma_{D2} z_{1i}.$$

Finally,

$$\Pr(S_i = 1 | z_{0i}, \theta) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \Pr(S_i = 1 | z_{0i}, y_{1i}, y_{2i}, \theta) f_{Y_1, Y_2}(y_{1i}, y_{2i}) dy_{1i} dy_{2i}.$$

In their paper, Wason & Seaman fit the model using the R functions `gls` and `glm`, and used cubature to resolve the double integral above. In **trialr**, we provide priors on the elements of θ and sample from the posterior using Stan.

3. Using trialr

3.1. Describing outcomes in dose-finding trials

In the following sections, we will make frequent use of the following scheme for describing outcomes in dose-finding trials. In situations where toxicity only is being assessed, as with CRM, we will use the characters `T` and `N` to represent the outcomes toxicity and no-toxicity. We will string these characters behind integer dose-levels to represent the outcomes of patients in cohorts treated at a dose. For instance, the string `1NNN 2TNT` represents the outcomes of six patients, treated in two cohorts of three. The first cohort was treated at dose-level 1 and none of the patients had toxicity. The second cohort received dose-level 2 and two of the three experienced toxicity.

In the setting where efficacy and toxicity outcomes are valuated, as with EffTox, we use the characters `E`, `T`, `B`, and `N` to represent the outcomes efficacy only, toxicity only, both and neither. Again, we string these characters behind integer dose-levels to represent cohorts. The string `1NN 2EB` represents the outcomes of two cohorts of two patients. The second cohort was treated at dose-level 2 and both experienced the efficacy event, but one of the two also experienced toxicity.

This syntax was originally described in the context of an EffTox trial (Brock *et al.* 2017). It is used in **trialr** to describe outcomes in dose-finding trials.

3.2. CRM

The core function for fitting a CRM model in **trialr** is `stan_crm`. To demonstrate usage, we will fit a model to the scenario described by Cheung (2011, p. 21).

In a five-dose scenario, they seek the dose associated with the target toxicity level 25% using the initial toxicity skeleton (5%, 12%, 25%, 40%, 55%). They use a one-parameter logistic model with intercept fixed at 3 and a $N(0, 1.34)$ prior distribution on β. In their scenario, they have already evaluated five patients at dose-levels 3, 4 and 5 with outcomes `3N 5N 5T 3N 4N` and they want to know the dose recommended by the model for the sixth patient.

```
R> library(trialr)
R>
R> skeleton <- c(0.05, 0.12, 0.25, 0.40, 0.55)
R> target <- 0.25
R> fit <- stan_crm(outcome_str = '3N 5N 5T 3N 4N',
R+                 skeleton = skeleton, target = target,
R+                 model = 'logistic', a0 = 3,
```

```
R+                     beta_mean = 0, beta_sd = sqrt(1.34),
R+                     seed = 123, refresh = 0)
R>
R> fit
```

```
  Patient Dose Toxicity Weight
1       1    3        0      1
2       2    5        0      1
3       3    5        1      1
4       4    3        0      1
5       5    4        0      1
```

```
  Dose Skeleton N Tox ProbTox MedianProbTox ProbMTD
1    1     0.05 0   0  0.0343       0.00833   0.043
2    2     0.12 0   0  0.0697       0.02838   0.074
3    3     0.25 2   0  0.1371       0.08599   0.161
4    4     0.40 1   0  0.2295       0.18900   0.246
5    5     0.55 2   1  0.3507       0.34000   0.476
```

```
The model targets a toxicity level of 0.25.
The dose with estimated toxicity probability closest to target is 4.
The dose most likely to be the MTD is 5.
Model entropy: 1.32
```

trialr currently supports models listed in Table 1. The choice of `model` determines what model parameters are required. We see above that the `logistic` model requires the parameter `a0` for the fixed intercept, and `beta_mean` and `beta_sd` parameters to describe the prior on β. In contrast, the `logistic2` parameterisation requires prior hyperparameters on α and β. See `? stan_crm` for full details.

Extra parameters in calls to `stan_crm` are passed onwards via the `...` operator to the function in **rstan** that performs the MCMC sampling, `rstan::sampling`. The parameter choice `seed = 123` allows reproducible sampling. The effect of `refresh = 0` is to suppress the log messages created by Stan.

The object returned by `stan_crm` has type `crm_fit`, which subclasses the type `dose_finding_fit`. Its default `print` method displays a table of patient-level information, followed by a table of dose-level information, followed by some summary information. `ProbTox` is the posterior mean probability of toxicity at each dose calculated from the full posterior distributions.

trialr works with **tidybayes** (Kay 2019) so that posterior samples are simple to access and flexible to work with.

```
R> library(dplyr)
R> library(tidybayes)
R> fit %>%
R+   gather_draws(prob_tox[dose]) %>%
R+   head
```

```
# A tibble: 6 x 6
# Groups:   dose, .variable [5]
  .chain .iteration .draw  dose .variable    .value
   <int>      <int> <int> <int> <chr>         <dbl>
1      1          1     1     1 prob_tox   0.0103
2      1          1     1     2 prob_tox   0.0339
3      1          1     1     3 prob_tox   0.0986
4      1          1     1     4 prob_tox   0.209
5      1          1     1     5 prob_tox   0.363
6      1          2     2     1 prob_tox   0.000737
```

This is particularly valuable because it facilitates inference and visualisation. For instance, each posterior sample for β generates a dose-vs-Prob(DLT) curve, and each of these nominates a candidate for the MTD, being the dose with estimated Prob(DLT) closest to the toxicity target. We can plot some of these generated curves, colouring each by the MTD that they nominate, to visualise the level of uncertainty still inherent in the situation at this early stage.

```
R> fit %>%
R+   gather_draws(prob_tox[dose]) %>%
R+   group_by(.draw) %>%
R+   summarise(mtd = dose[which.min(abs(.value - target))]) -> mtd_candidates
R>
R> library(ggplot2)
R> fit %>%
R+   gather_draws(prob_tox[dose]) %>%
R+   left_join(mtd_candidates, by = '.draw') %>%
R+   filter(.draw <= 200) %>%
R+   ggplot(aes(x = dose, y = .value, group = .draw)) +
R+   geom_line(aes(col = as.factor(mtd)), alpha = 0.5) +
R+   geom_hline(yintercept = target, col = 'red', linetype = 'dashed') +
R+   labs(title = 'The identify of the MTD is still shrouded in mystery',
R+        y = 'Prob(DLT)', col = 'MTD') +
R+   theme(legend.position = 'bottom')
```

We plot just 200 curves to avoid saturating the plot. We could also calculate the posterior probability that each dose is the true MTD according to this model.

```
R> mtd_candidates %>%
R+   count(mtd) %>%
R+   mutate(prob_mtd = n / sum(n))
```

```
# A tibble: 5 x 3
    mtd     n prob_mtd
  <int> <int>    <dbl>
1     1   172    0.043
2     2   296    0.074
```
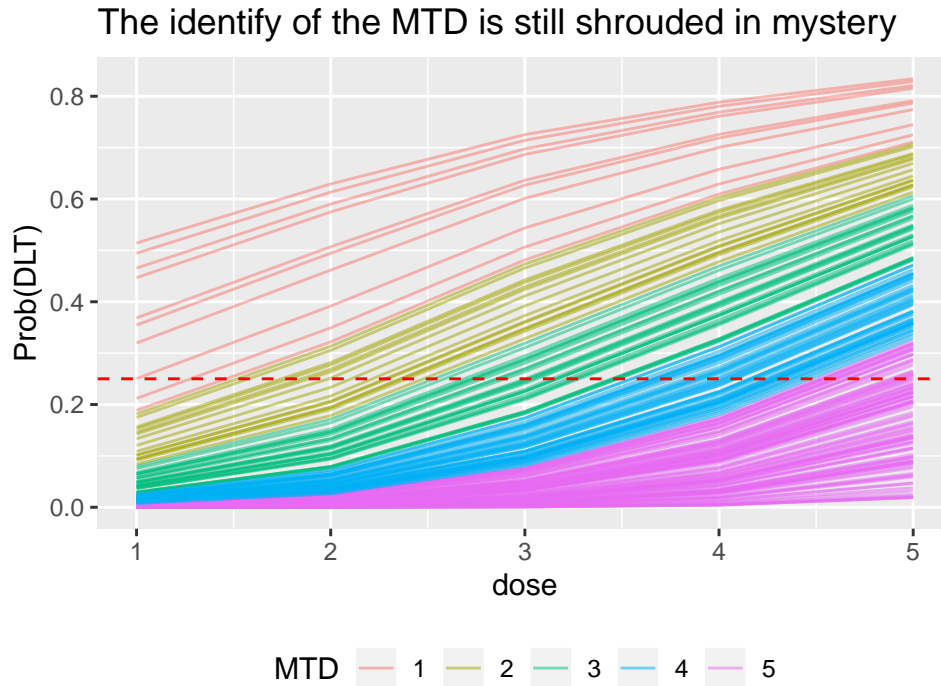
Figure 1: Posterior dose-toxicity curves generated by a CRM model fit to five patients' data.

```
3      3    643     0.161
4      4    984     0.246
5      5   1905     0.476
```

This matches the `ProbMTD` statistics in the model output above.

Yap *et al.* (2017) introduced *dose transition pathways* (DTPs) as the link between model-based statistical dose-finging algorithms and simple decision making. A pathway in a dose-finding trial is simply a sequence of doses that were delivered to patients and the outcomes that the patients experienced.

Consider a trial that has already treated four patients in two cohorts:

```
R> outcomes <- '2NN 3TN'
```

Beyond questioning what dose will be advised for the third cohort, we might wonder about all the doses could possibly be advised for the fourth and fifth cohorts. This might have material consequences for the amount of drug that is to be manufactured and shipped to research pharmacies. **trialr** provides flexible methods to automate such analyses.

For the purposes of this example, let us assume we are using the `empiric` model form and a unit normal prior on β, and the following parameters:

```
R> skeleton <- c(0.05, 0.15, 0.25, 0.4, 0.6)
R> target <- 0.25
```

We calculate DTPs using the `crm_dtps` method:

```
R> paths1 <- crm_dtps(skeleton = skeleton, target = target, model = 'empiric',
R+                    cohort_sizes = c(3, 3), previous_outcomes = outcomes,
R+                    beta_sd = 1, refresh = 0)
```

This function uses many of the same parameters as `stan_crm`. The `previous_outcomes` parameter describes the path the trial has already taken; each future path starts from here. It can be omitted to reflect a trial that has not yet started. The `cohort_sizes` parameter controls the number and size of future cohorts to consider. Each cohort of n may result in 0, 1, 2,..., or n patients experiencing DLT. Thus calculating transitions for two future cohorts of three patients, as above, yields 16 possible paths.

At the end of each cohort, there is a dose selection decision. By default, CRM models choose the dose with posterior mean probability of toxicity closest to the target. However, in practice, trialists may want to tailor this decision.

Custom dose selection is supported in `crm_dtps` via the `user_dose_func` parameter. For instance, investigators may want to prevent the skipping of untested doses in escalalation, or build in the facility to stop a trial early if there is sufficient evidence that a reference dose is excessively toxic. Both of these design aspects were used in a recently published dose-escalation trial in acute myeloid leukaemia (Craddock, Slade, De Santo, Wheat, Ferguson, Hodgkinson, Brock, Cavenagh, Ingram, Dennis, Malladi, Siddique, Mussai, and Yap 2019) and they are provided in `trialr` by the `careful_escalation` function. It prevents skipping in escalation by recommending, where appropriate, the next highest dose to the maximum that has yet been used. For instance, if the default behaviour is to escalate from dose 2 to dose 4 without having tested dose 3, `careful_escalation` will instead recommend dose 3. Additionally, it returns `NA` if a reference dose is detected to be too toxic and this is interpreted by `crm_dtps` as the sign that this trial path should stop. Users are free to provide any delegate function via `user_dose_func` that takes a `dose_finding_fit` as the sole parameter and returns either an integer dose-level or `NA`. We re-calculate those paths using our custom dose function:

```
R> paths2 <- crm_dtps(skeleton = skeleton, target = target, model = 'empiric',
R+                    cohort_sizes = c(3, 3), previous_outcomes = outcomes,
R+                    user_dose_func = function(x) {
R+                        careful_escalation(x, tox_threshold = target + 0.1,
R+                                                certainty_threshold = 0.7,
R+                                                reference_dose = 1)
R+                    }, beta_sd = 1, seed = 123, refresh = 0)
```

Here, a pathway will stop if ever

$$Pr(\pi_T(x_1) > 0.35|X) > 0.7, \tag{1}$$

where $\pi_T(x)$ is the posterior probability of DLT at dose x.

The paths are shown in Figure 2. Code to recreate this graph appears in the Appendix. We see that the subsequent path `2TTT 1TTT` yields the understandable decision to stop, but that every other path selects a dose. We will only advise as high as dose 4 if no DLTs are seen in the next 6 patients. This level of foresight and transparency underpins our motivation to use dose pathways, particularly in a visual manner.
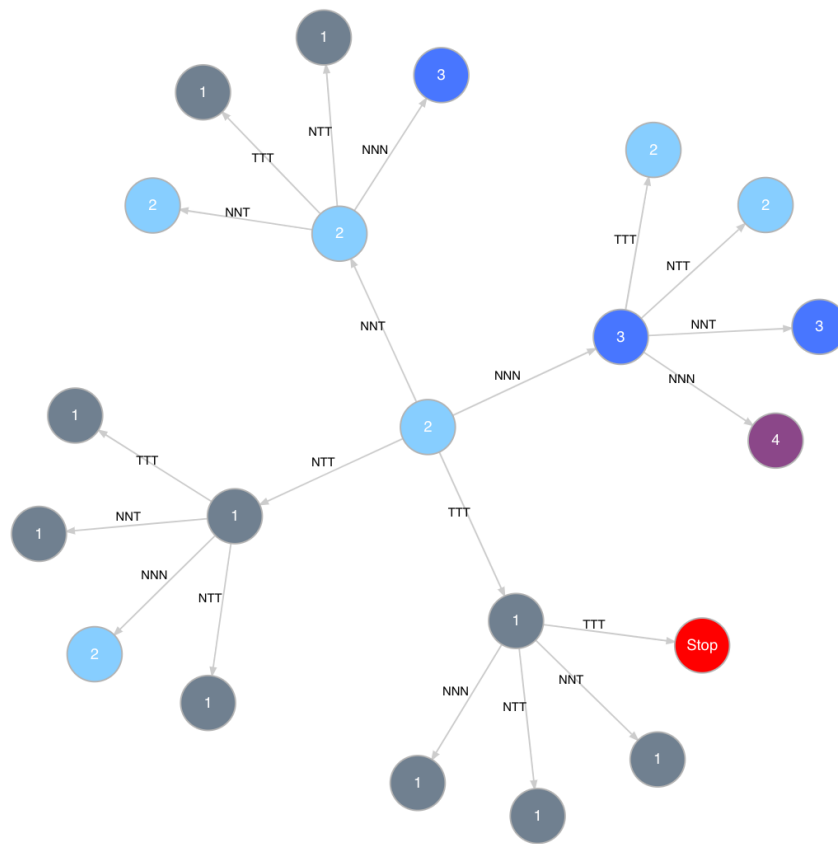
Figure 2: Dose transition pathways for the next two cohorts of three patients in a dose-finding trial using a CRM design.

The returned object, `paths2`, has type `dose_finding_paths` and contains model fits at each
node on each pathway, as well as information on the tree structure. This is perhaps best
viewed as a `tibble`:

```
R> library(tibble)
R> paths2_df <- as_tibble(paths2)
R> paths2_df %>% head()
```

```
# A tibble: 6 x 8
  .node .parent .depth outcomes next_dose fit       parent_fit dose_index
  <dbl>   <dbl>  <dbl> <chr>        <dbl> <list>    <list>     <list>
1     1      NA      0 ""               2 <crm_fit> <NULL>     <dbl [5]>
2     2       1      1 NNN              3 <crm_fit> <crm_fit>  <dbl [5]>
3     3       2      2 NNN              4 <crm_fit> <crm_fit>  <dbl [5]>
4     4       1      1 NNT              2 <crm_fit> <crm_fit>  <dbl [5]>
5     5       4      2 NNN              3 <crm_fit> <crm_fit>  <dbl [5]>
6     6       1      1 NTT              1 <crm_fit> <crm_fit>  <dbl [5]>
```

Each node has a single row in this table. The variables that define the tree structure are
prefixed with a period. Whilst this tall format is useful for some purposes, like creating
graphs like Figure 2, dose-transition pathways are perhaps most intuitively tabulated in a
wide format. That task is accomplished by the `spread_paths` function, that will join `.node`
to `.parent` at each `.depth` to create a wide view of DTPs. For example,

```
R> spread_paths(paths2_df %>% select(-fit, -parent_fit, -dose_index))
```

```
# A tibble: 16 x 6
   outcomes0 next_dose0 outcomes1 next_dose1 outcomes2 next_dose2
   <chr>          <dbl> <chr>          <dbl> <chr>          <dbl>
 1 ""                 2 NNN                3 NNN                4
 2 ""                 2 NNN                3 NNT                3
 3 ""                 2 NNN                3 NTT                2
 4 ""                 2 NNN                3 TTT                2
 5 ""                 2 NNT                2 NNN                3
 6 ""                 2 NNT                2 NNT                2
 7 ""                 2 NNT                2 NTT                1
 8 ""                 2 NNT                2 TTT                1
 9 ""                 2 NTT                1 NNN                2
10 ""                 2 NTT                1 NNT                1
11 ""                 2 NTT                1 NTT                1
12 ""                 2 NTT                1 TTT                1
13 ""                 2 TTT                1 NNN                1
14 ""                 2 TTT                1 NNT                1
15 ""                 2 TTT                1 NTT                1
16 ""                 2 TTT                1 TTT               NA
```

In this view, paths can be read horizontally. This confirms the dose choices shown in Figure 2. Had we retained `fit` in the frame that was passed to `spread_paths`, there would also be columns `fit0`, `fit1` and `fit2` in the table above. These functions create a rich and flexible method for working with paths in CRM trials. We demonstrate a similar approach in the next section with EffTox.

Fitting TITE-CRM models is also supported in **trialr**, again through the `stan_crm` function. The outcome syntax used above, however, ceases to be succinct and useful when each patient is associated an individual weight. Thus, when fitting the TITE-CRM, we provide vectors for the doses fit, the binary toxicity outcomes observed, and the weight given to each observation. For instance, in the example below we calculate the next dose in a trial that has partially evaluated four patients at dose 3, as described by Cheung (2011, p.124). No patient has yet experienced DLT but they are between 28 and 73 days through a 126-day evaluation window.

```
R> fit <-stan_crm(skeleton = c(0.05, 0.12, 0.25, 0.40, 0.55), target = 0.25,
R+                doses_given = c(3, 3, 3, 3),
R+                tox = c(0, 0, 0, 0),
R+                weights = c(73, 66, 35, 28) / 126,
R+                model = 'empiric', beta_sd = sqrt(1.34),
R+                seed = 123, refresh = 0)
R> fit$recommended_dose


[1] 4
```

We receive confirmation that the model advocates starting the next patient on dose 4.

3.3. EffTox

The EffTox model addresses the need for a dose-finding design that incorporates both efficacy and toxicity outcomes. This is useful in modern oncology treatments like small molecules and immunotherapies where, compared to chemotherapy, there is a weaker rationale for the assumption that higher doses are associated with higher probabilities of success.

The probability model for EffTox described above uses six parameters. The extra complexity naturally means it is more challenging to select priors. Thall *et al.* (2014) describe an algorithm that selects hyperparameters for weakly informative normal priors on α, β, γ and ζ so that the expected prior probabilities of efficacy and toxicity at each dose match investigator-elicited values. The priors for η and ψ are fixed at $N(0, 0.2)$ and $N(0, 1)$ respectively.

For example, expecting efficacy probabilities (0.2, 0.4, 0.6, 0.8, 0.9) and toxicity probabilities (0.02, 0.04, 0.06, 0.08, 0.10) in a five-dose example, Thall *et al.* (2014) choose the parameter priors in Table 2. **trialr** does not currently implement the algorithm to generate these hyperparameters but will do in future. In the meantime, the official EffTox software implements the algorithm.

Figures 3 and 4 show that the prior views on the location of the optimal dose do not change materially for modest changes in the values of the standard deviation hyperparameters. In each, the value of the non-varying parameters takes the value listed in Table 2. We see in each panel of each plot that dose-level 5 is overwhelmingly expected to have the highest utility score. As the standard deviation hyperparameters increase, more probability is placed on dose

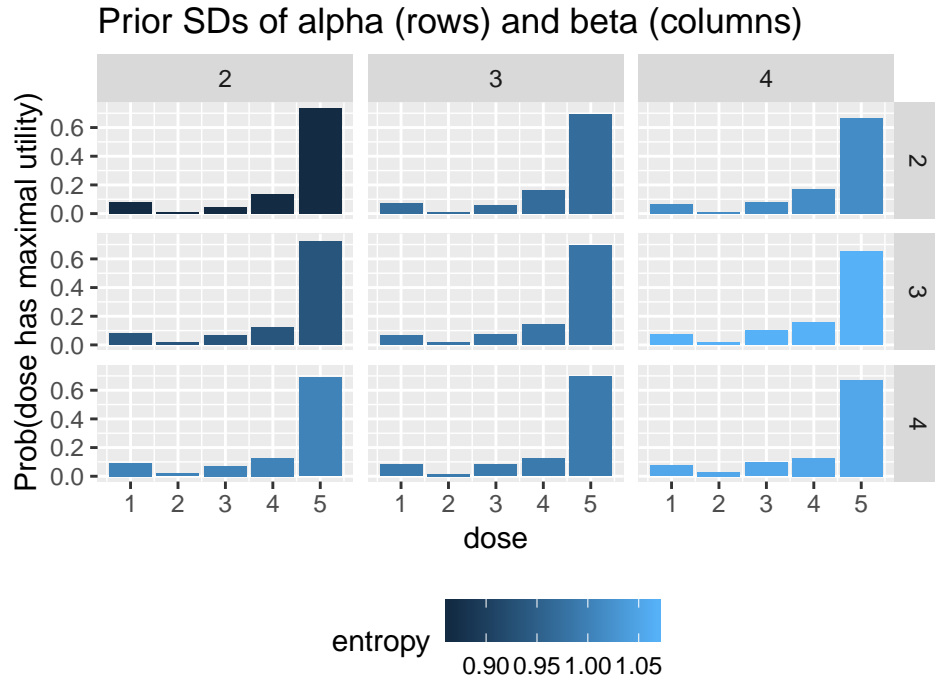| Parameter	Mean	Standard deviation
α	-7.9593	3.5487
β	1.5482	3.5018
γ	0.7367	2.5423
ζ	3.4181	2.4406
η	0	0.2
ψ	0	1

Table 2: Normal prior hyperparameters used by Thall *et al.* (2014).

Figure 3: The prior standard deviations on alpha and beta in EffTox have relatively weak effects on the prior beliefs on the location of optimal dose.
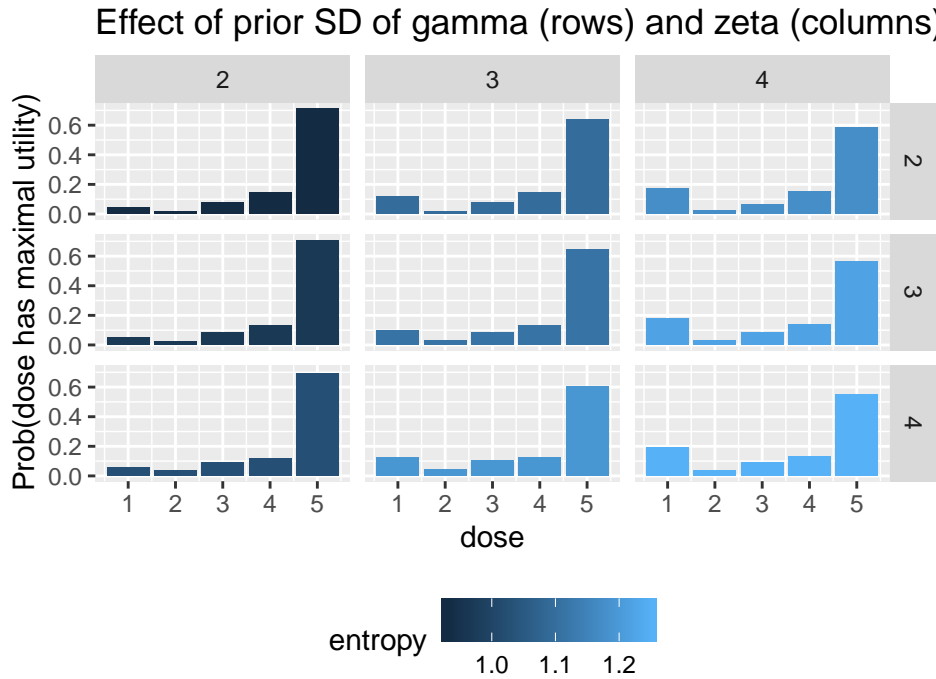
Figure 4: The prior standard deviations on gamma and zeta in EffTox have relatively weak effects on the prior beliefs on the location of optimal dose.

1 being optimal. The probabilities of doses 2 to 4 barely change. The prior probabilities that each dose is optimal are driven primarily by the location of the parameter priors rather than the scale.

By way of illustration, let us continue with Thall *et al.*'s parameterisation. The authors seek the most attractive dose of a drug from (1, 2, 4, 6.6, 10) mcL/kg that satisfies both

$$Pr(\pi_E(x_i) > 0.5|X) > 0.1 \tag{2}$$

and

$$Pr(\pi_T(x_i) < 0.65|X) > 0.1 \tag{3}$$

where the neutral utility contour intersects the hinge points $(\pi_E, \pi_T) = (0.5, 0)$, $(1, 0.65)$ and $(0.7, 0.25)$. Let us fit this model to the outcomes 1NNN 2ENN:

```
R> outcomes <- '1NNN 2ENN'
R>
R> fit <- stan_efftox(outcome_str = outcomes,
R+                    real_doses = c(1.0, 2.0, 4.0, 6.6, 10.0),
R+                    efficacy_hurdle = 0.5, toxicity_hurdle = 0.3,
R+                    p_e = 0.1, p_t = 0.1, eff0 = 0.5, tox1 = 0.65,
R+                    eff_star = 0.7, tox_star = 0.25,
R+                    alpha_mean = -7.9593, alpha_sd = 3.5487,
R+                    beta_mean = 1.5482, beta_sd = 3.5018,
```

```
R+                      gamma_mean = 0.7367, gamma_sd = 2.5423,
R+                      zeta_mean = 3.4181, zeta_sd = 2.4406,
R+                      eta_mean = 0, eta_sd = 0.2,
R+                      psi_mean = 0, psi_sd = 1,
R+                      seed = 123, refresh = 0)
R> fit

  Patient Dose Toxicity Efficacy
1       1    1        0        0
2       2    1        0        0
3       3    1        0        0
4       4    2        0        1
5       5    2        0        0
6       6    2        0        0

  Dose N ProbEff ProbTox ProbAccEff ProbAccTox Utility Acceptable ProbOBD
1    1 3  0.0506 0.00685      0.003      0.997  -0.911      FALSE  0.0158
2    2 3  0.2700 0.00356      0.130      1.000  -0.466       TRUE  0.0075
3    3 0  0.7287 0.01399      0.798      0.988   0.434       TRUE  0.0862
4    4 0  0.8667 0.05261      0.911      0.940   0.648      FALSE  0.1893
5    5 0  0.9121 0.11791      0.938      0.861   0.637      FALSE  0.7013


The model recommends selecting dose-level 3.
The dose most likely to be the OBD is 5.
Model entropy: 0.88
```

We see that dose-level 3 is recommended for the next cohort. Even though doses 4 and 5 have higher estimated utility scores, they are not recommended because dose 3 has not yet been tested. `ProbAccEff` and `ProbAccTox` estimate the probability that the estimated rates of efficacy and toxicity are the desirable side of 0.5 and 0.65, respectively. Dose 1 is not acceptable because `ProbAccEff` is less than 0.1. The utility values implied by the posterior mean probabilities of efficacy and toxicity are shown in Figure 5, created using the command:

```
R> efftox_contour_plot(fit)
```

Using the posterior samples of the utility scores at each dose, it is possible to estimate the probability that each dose is superior to the others. The following superiority matrix shows the implied probability that the dose represented by the column has greater utility than the dose represented by the row:

```
R> efftox_superiority(fit)

        1       2       3       4       5
1      NA 0.98125 0.97800 0.97350 0.96800
2 0.01875      NA 0.97575 0.96075 0.93150
3 0.02200 0.02425      NA 0.89200 0.81300
4 0.02650 0.03925 0.10800      NA 0.70825
5 0.03200 0.06850 0.18700 0.29175      NA
```
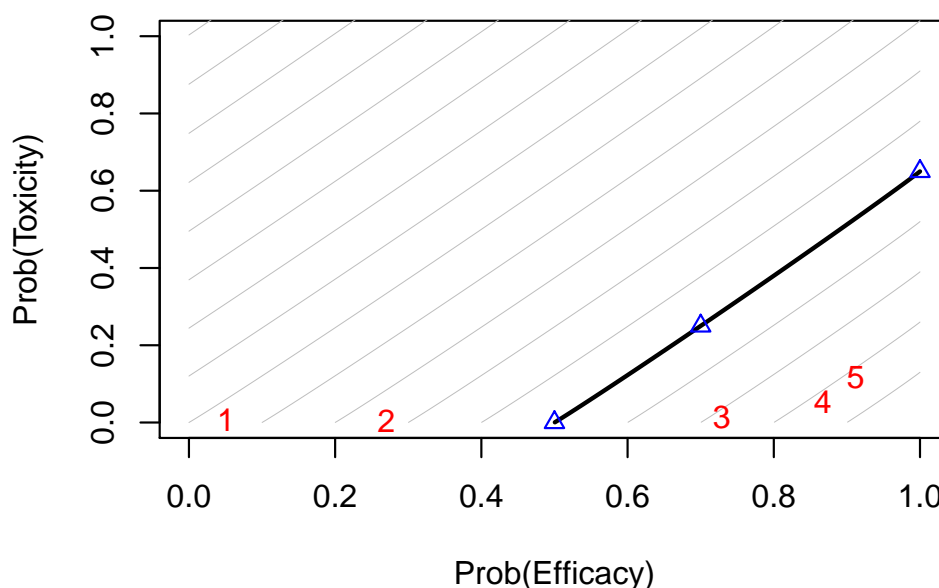
Figure 5: Utility contour plot for the described EffTox example after observing 1NNN 2ENN.

Thus, we can be quite sure that dose 4 has higher utility than doses 1 to 3, but we are rather more unsure about whteher it has higher utility than dose 5. This is confirmed by Figure 5.

trialr also supports the calculation of dose-transition pathways for EffTox. The number of events that patients may experience is now four (E, T, B or N), compared to the two events (T or N) that were possible in the CRM examples. As such, the number of distinct pathways that can be taken increases very quickly. For instance, the number of outcome combinations for a cohort of three is 20, so that two cohorts of three generate 400 pathways. This is too many to represent on a graph. To illustrate, we consider the next cohort of three patients in our trial above:

```
R> paths <- efftox_dtps(cohort_sizes = c(3), previous_outcomes = outcomes,
R+                      real_doses = c(1.0, 2.0, 4.0, 6.6, 10.0),
R+                      efficacy_hurdle = 0.5, toxicity_hurdle = 0.3,
R+                      p_e = 0.1, p_t = 0.1, eff0 = 0.5, tox1 = 0.65,
R+                      eff_star = 0.7, tox_star = 0.25,
R+                      alpha_mean = -7.9593, alpha_sd = 3.5487,
R+                      beta_mean = 1.5482, beta_sd = 3.5018,
R+                      gamma_mean = 0.7367, gamma_sd = 2.5423,
R+                      zeta_mean = 3.4181, zeta_sd = 2.4406,
R+                      eta_mean = 0, eta_sd = 0.2,
R+                      psi_mean = 0, psi_sd = 1,
R+                      next_dose = 3, seed = 123, refresh = 0)
```

Again, the returned object has type `dose_finding_paths` so, as with the CRM examples, we can invoke `as_tibble` and `spread_paths` on `paths`. The associated `fit` and `parent_fit` objects are each of type `efftox_fit`. These can be used with `map`-like functions from **purrr** (Henry and Wickham 2019) for flexible analysis of DTP objects:

```
R> library(tidyr)
R> library(purrr)
R>
R> as_tibble(paths) %>%
R+   filter(.depth > 0) %>%
R+   mutate(prob_obd = map(fit, 'prob_obd'),
R+          parent_prob_obd = map(parent_fit, 'prob_obd')) %>%
R+   select(outcomes, dose_index, prob_obd, parent_prob_obd) %>%
R+   unnest %>%
R+   mutate(prob_obd_delta = prob_obd - parent_prob_obd) %>%
R+   filter(dose_index == 5)
```

```
# A tibble: 20 x 5
   outcomes dose_index prob_obd parent_prob_obd prob_obd_delta
   <chr>         <int>    <dbl>           <dbl>          <dbl>
 1 BBB               5    0.136           0.701         -0.565
 2 BBE               5    0.061           0.701         -0.640
 3 BBN               5    0.176           0.701         -0.525
 4 BBT               5    0.316           0.701         -0.385
 5 BEE               5    0.142           0.701         -0.559
 6 BEN               5    0.21            0.701         -0.491
 7 BET               5    0.164           0.701         -0.538
 8 BNN               5    0.323           0.701         -0.378
 9 BNT               5    0.274           0.701         -0.427
10 BTT               5    0.397           0.701         -0.305
11 EEE               5    0.677           0.701         -0.0245
12 EEN               5    0.756           0.701          0.0550
13 EET               5    0.223           0.701         -0.478
14 ENN               5    0.788           0.701          0.0872
15 ENT               5    0.312           0.701         -0.389
16 ETT               5    0.280           0.701         -0.421
17 NNN               5    0.728           0.701          0.0272
18 NNT               5    0.329           0.701         -0.372
19 NTT               5    0.242           0.701         -0.460
20 TTT               5    0.287           0.701         -0.415
```

In the example above for instance, although dose-level 5 is initially selected by the model as that most likely to be the OBD, that view can change dramatically based on the outcome of the next cohort of three, which will be treated at dose 3. This demonstrates the lack of information currently in the model in this situation, particularly the sensitivity of the estimated identity of the OBD to incidence of any toxicity.

The left panel of Figure 6 shows the paths for the next cohort of three. After only nine patients, there are three paths that advocate stopping. These are all paths that see considerable toxicity in cohort 3. The right panel of Figure 6 shows the paths generated by the next two cohorts of one patient. Code to create these graphs appears in the Appendix.

The function `efftox_dtps` also accepts a custom dose selection function via the `user_dose_func` parameter. The `fit` object recorded at each node in `paths` has type `efftox_fit`, a subclass
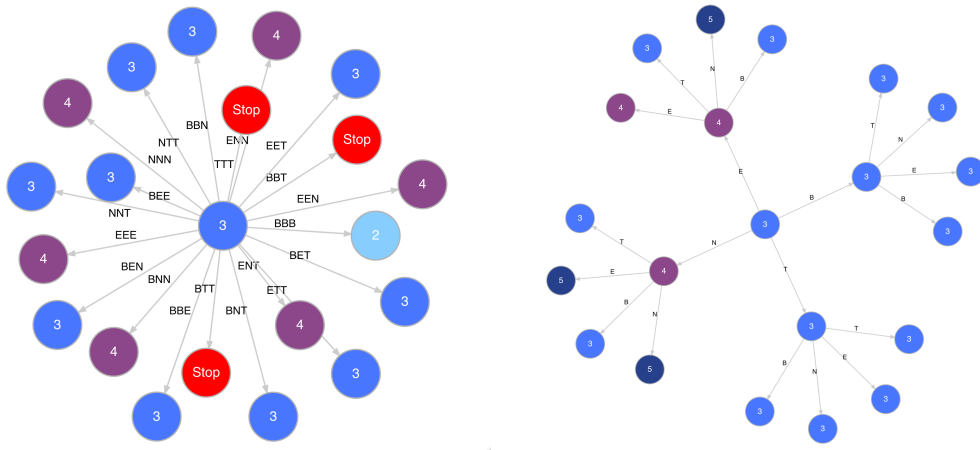
Figure 6: Dose pathways for the next cohort of three patients (left) and next two cohorts of one patient (right) in an EffTox trial that has already seen '1NNN 2ENN'.

of `dose_finding_fit`. The `user_dose_func` delegate should take an `efftox_fit` as the sole parameter and return an integer dose-level for the next patient, or `NA` to advocate stopping the trial, mirroring the behaviour of the CRM example.

3.4. Augmented Binary

To demonstrate the two-period Augmented Binary method for single arm trials, we will fit it to a dataset drawn from one of the simulation scenarios used by Wason and Seaman (2013). First, let us consider priors on the parameters.

We will consider two sets of priors to demonstrate the effect they have on the posterior. The prior distributions in the first set are relatively informative:

```
R> informative_priors <- list(alpha_mean = 0, alpha_sd = 0.1,
R+                            beta_mean = 0, beta_sd = 0.1,
R+                            gamma_mean = 0, gamma_sd = 0.1,
R+                            sigma_mean = 0, sigma_sd = 0.5,
R+                            omega_lkj_eta = 1,
R+                            alpha_d1_mean = 0, alpha_d1_sd = 0.5,
R+                            gamma_d1_mean = 0, gamma_d1_sd = 0.25,
R+                            alpha_d2_mean = 0, alpha_d2_sd = 0.5,
R+                            gamma_d2_mean = 0, gamma_d2_sd = 0.25)
```

For the covariance matrix Σ, half-normal priors with parameters `sigma_mean` and `sigma_sd` are used for the diagonal elements, and an LKJ prior is specified on the associated correlation matrix, Ω. The parameter $\Omega_\eta = 1$ generates uniform beliefs on the correlation between y_1 and y_2. Values > 1 would reflect an expectation that correlations near 0 are more likely, and vice-versa.

In constrast, the priors in the second set are relatively diffuse:

```
R> diffuse_priors <- list(alpha_mean = 0, alpha_sd = 1,
```
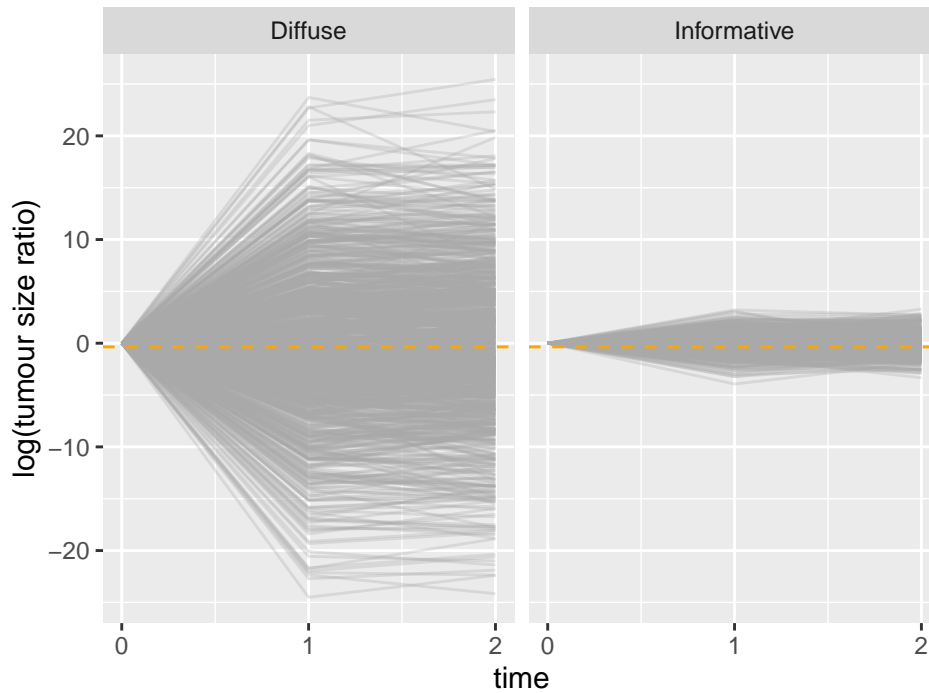
Figure 7: Prior predictive tumour size ratios generated by two sets of prior parameters in the two-period AugBin model.

```
R+                      beta_mean = 0, beta_sd = 1,
R+                      gamma_mean = 0, gamma_sd = 1,
R+                      sigma_mean = 0, sigma_sd = 1,
R+                      omega_lkj_eta = 1,
R+                      alpha_d1_mean = 0, alpha_d1_sd = 1,
R+                      gamma_d1_mean = 0, gamma_d1_sd = 1,
R+                      alpha_d2_mean = 0, alpha_d2_sd = 1,
R+                      gamma_d2_mean = 0, gamma_d2_sd = 1)
```

Unit normal distributions may not immediately be regarded as diffuse until we consider their effect on the generated outcome distributions. Prior predictive outcomes generated by each set of priors are shown in Figures 7 and 8. Code to create these figures are included in the Appendix.

Figure 7 shows the sampled log-tumour-size ratios with respect to baseline. We see that the diffuse priors generate a far wider set of outcomes. In fact, log-tumour-size ratios near 20 imply tumour growth that is many orders of magnitude greater than what is feasible. Growth to this extent would imply that the patient could increase their volume thousands of times over. The baseline tumour size are sampled from U(5, 10) cm so that the expected tumour size in each set of priors is 7.5cm. Under the informative priors, the expected tumour size at each post-baseline assessment is 12.5cm. Under each set, the chances of tumour growth under the experimental treatment is roughly 50%.

Figure 8 shows the prior predictive probabilities of non-shrinkage failure generated by two prior sets at the interim timepoint. Once again, we see that extreme outcomes are generated
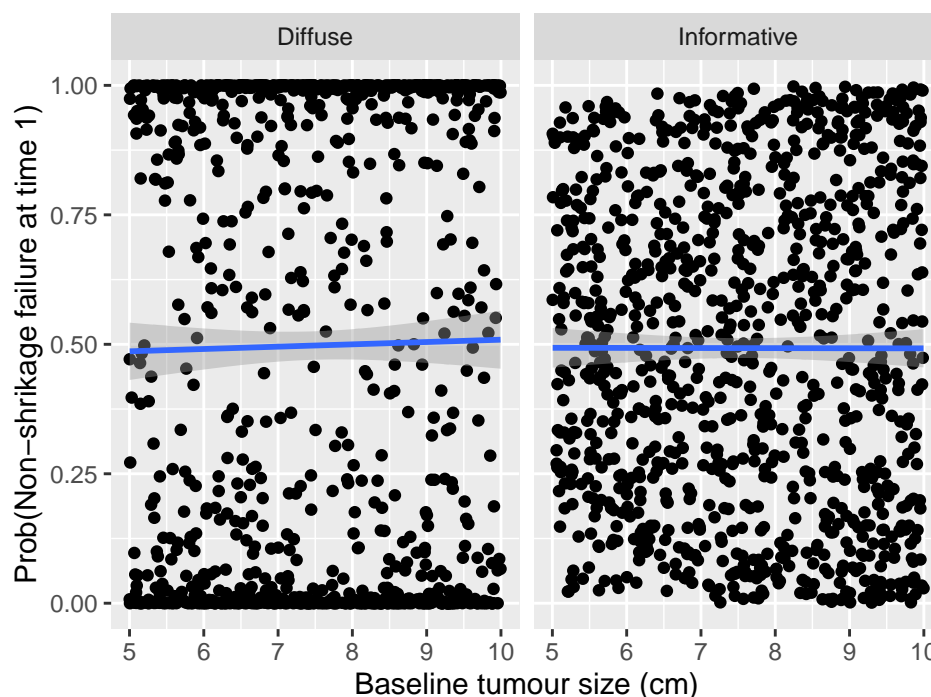
Figure 8: Prior predictive probabilities of non-shrinkage failure generated by two sets of prior parameters in the two-period AugBin model.

far too readily by the diffuse priors. In contrast, the informative priors generate probabilities that are roughly uniform and not associated with baseline tumour size.

To demonstrate the AugBin model, we will fit it to a dataset randomly sampled using the parameterisation described by Wason and Seaman (2013) in the first line of their Table 1. We have parameters:

```
R> N <- 50
R> sigma <- 1
R> delta1 <- -0.356
R> mu <- c(0.5 * delta1, delta1)
R> Sigma = matrix(c(0.5 * sigma^2, 0.5 * sigma^2,
R+                  0.5 * sigma^2, sigma^2), ncol = 2)
R> alphaD <- -1.5
R> gammaD <- 0
```

That is, there are 50 patients, the expected tumour shrinkage at time 2 is $\exp(-0.356) - 1 = 30\%$, and the chances of non-shrinkage failure at each stage are $\text{logit}^{-1}(-1.5) = 18\%$. These parameters generate data:

```
R> set.seed(123456)
R> y <- MASS::mvrnorm(n = N, mu, Sigma)
R> z0 <- runif(N, min = 5, max = 10)
R> z1 <- exp(y[, 1]) * z0
```

```
R> z2 <- exp(y[, 2]) * z0
R> d1 <- rbinom(N, size = 1, prob = gtools::inv.logit(alphaD + gammaD * z0))
R> d2 <- rbinom(N, size = 1, prob = gtools::inv.logit(alphaD + gammaD * z1))
R> tumour_size = data.frame(z0, z1, z2) # cm
R> non_shrinkage_failure <- data.frame(d1, d2)
```

The values in `tumour_size` reflect (z_1, z_2, z_3) measured in cm.

To fit this data to the two-period, single-arm model, referred to as `2t-1a` in **trialr**, we run the command:

```
R> fit_diffuse <- stan_augbin(tumour_size, non_shrinkage_failure,
R+                            model = '2t-1a', prior_params = diffuse_priors,
R+                            seed = 123, refresh = 0)
R> fit_inf <- stan_augbin(tumour_size, non_shrinkage_failure,
R+                          model = '2t-1a', prior_params = informative_priors,
R+                          seed = 123, refresh = 0)
```

The object returned by `stan_fit` has type `augbin_fit`.

The main benefit that Wason & Seaman identify for their method is the increase in efficiency that comes from analysing the continuous tumour size variable rather than dichotomising the information into a binary outcome. They compare widths of 95% confidence intervals for the probability of success under their method and the binary dichotomisation, using 30% shrinkage at time 2 and no non-shrinkage failure as the criteria for success. We can recreate this type of analysis in **trialr**. Focusing on the model fit using the diffuse priors:

```
R> pred_diffuse <- predict(fit_diffuse, y2_upper = log(0.7))
R> pred_diffuse %>% head()
```

```
# A tibble: 6 x 8
      id    z0    z1 prob_success_samp prob_success lower upper ci_width
   <int> <dbl> <dbl> <list>                   <dbl> <dbl> <dbl>    <dbl>
1      1  5.99  16.0 <dbl [4,000]>            0.277 0.176 0.385    0.208
2      2  5.66  4.81 <dbl [4,000]>            0.223 0.141 0.318    0.177
3      3  9.32  5.59 <dbl [4,000]>            0.269 0.165 0.385    0.220
4      4  6.16  3.16 <dbl [4,000]>            0.218 0.136 0.316    0.180
5      5  5.57  10.2 <dbl [4,000]>            0.250 0.161 0.352    0.191
6      6  7.08  11.1 <dbl [4,000]>            0.275 0.186 0.374    0.188
```

We see that the estimated probabilities of success are approximately 25%, and vary slightly depending on the baseline and interim tumour size measurements. The `prob_success_samp` column contains the posterior samples of the probability of success for each patient. By default, `lower` and `upper` contain posterior 2.5% and 97/5% quantiles of `prob_success` but this can be changed by specifying the desired quantile probabilities using the `probs` argument.

We can also make inferences by analysing the dichotomised binary outcome using the **binom** (Dorai-Raj 2014) package:
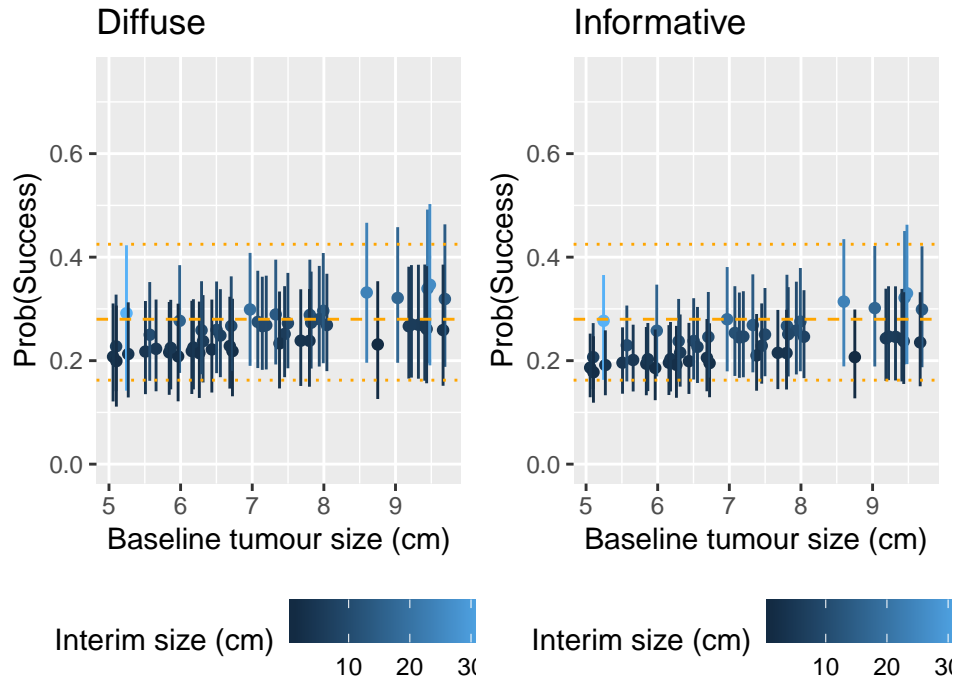
Figure 9: Posterior predictions from the AugBin model to 50 simulated patients under two different sets of priors.

```
R> pred_binary <- binary_prob_success(fit_diffuse, y2_upper = log(0.7),
R+                                     methods = 'exact')
R> pred_binary
```

```
  method  x  n mean     lower     upper   ci_width
1  exact 14 50 0.28 0.1623106 0.4249054 0.2625948
```

We see that there were 14 responses in 50 patients, yielding a response rate of 28%. Again, the default width of this frequentist confidence interval is 95%. The average decrease in the widths of the 95% uncertainty intervals is:

```
R> mean(pred_diffuse$ci_width) / pred_binary$ci_width - 1
```

```
[1] -0.2262706
```

The reduction of approximately 24% in this single simulated iteration is higher than the 16.5% average reduction reported by Wason & Seaman comparing AugBin to the regular binary method following a simulation study.

These inferences for both AugBin fits are shown in Figure 9. The points show the estimated probability of success, and the vertical bars the 95% posterior credible intervals. The orange lines show the estimates from the ordinary binary approach, with the dashed line showing the mean and the dotted lines the 95% confidence interval boundaries.

We see a slight positive relationship in estimated success probability and baseline tumour size. This is a random manifestation in the sampled dataset. We see that the intervals are

generally narrower with the AugBin method, particularly so under the informative priors. The average credible interval width under the informative priors is 36% narrower than that from the binary method. This suggests the strong effect the priors are having.

Comparing the two sets of fitted values from the AugBin models, we see that the estimates are generally a few percent lower under the informtive priors. In fact, a simulation study with 1000 iterations reveals that the estimated probabilities of success under these informative priors are biased downwards by 4.5% on average, and that the coverage of 95% posterior credible intervals is only 71.7%. In contrast, simulations under the diffuse priors reveal downward bias of approximately 1.5% and 95% posterior interval coverage of 92.7%, much closer to the nominal value. Code to run the simulation study appears in the GitHub repository of this article at `https://github.com/brockk/trialr-jss`

As an R-user might expect, the `predict` function allows the specification of `newdata` to obtain predictions for custom cases:

```
R> predict(fit_diffuse, newdata = data.frame(z0 = 5:10, z1 = 4:9),
R+          y2_upper = log(0.7))

# A tibble: 6 x 8
      id    z0    z1 prob_success_samp prob_success lower upper ci_width
   <int> <int> <int> <list>                   <dbl> <dbl> <dbl>    <dbl>
1      1     5     4 <dbl [4,000]>            0.210 0.124 0.312    0.188
2      2     6     5 <dbl [4,000]>            0.228 0.149 0.320    0.171
3      3     7     6 <dbl [4,000]>            0.247 0.166 0.336    0.171
4      4     8     7 <dbl [4,000]>            0.265 0.177 0.362    0.185
5      5     9     8 <dbl [4,000]>            0.282 0.180 0.394    0.214
6      6    10     9 <dbl [4,000]>            0.298 0.177 0.436    0.260
```

Function parameters also allow users to specify different shrinkage thresholds for response at interim and final timepoints.

4. Discussion

The **trialr** package exploits the power of Stan to offer full Bayesian inference to clinical trialists and researchers. We have illustrated this using the CRM and EffTox designs for conducting dose-finding trials. One of the cornerstones of our approach is using posterior samples for intuitive and flexible inference. For instance, having access to posterior samples in a utility-maximising design like EffTox allowed us to directly calculate the probabilities of superiority of the doses. Without posterior samples, that calculation would have been more challenging.

The other core tenets of our approach include using modern `tidyverse` (Wickham 2017) classes and methods and relying on visualisation to convey insights. By using **tidybayes** (Kay 2019), posterior samples are available in tidy formats, convenient for further analysis via **dplyr** (Wickham, François, Henry, and Müller 2019) and **purrr** (Henry and Wickham 2019) functions, as we have demonstrated in several examples. We have incldued plots using **ggplot2** (Wickham 2016) and graphs using **DiagrammeR** (Iannone 2018).

We have demonstrated functions and classes dedicated to calculating dose transition pathways in dose-finding trials. Our approach works not just for CRM and EffTox but for general phase

I and phase I/II designs. In future releases of **trialr**, we will provide the same funcionality for the Escalation with Overdose Control design (Tighiouart and Rogatko 2010) and the seamless phase I/II design of Wages and Tait (2015). Calculating dose paths for Wages & Tait's design may initially seem questionable because their method uses adaptive randomisation to allocate patients to doses inferred to be tolerable. If the dose recommended is random, what point is there in considering future paths? Although the dose is random, the way that future randomisation probabilities are adapted in response to outcomes will still be of interest. Our use of `tibble` allows that a vector may be the item of interest in a DTP analysis. For example, it would be possible to analyse adaptive randomisation probabilities in Wages & Tait's design in a manner similar to how we analysed above changes in the probability of a dose being the OBD.

We have not presented methods for conducting dose-finding simulation studies. Though these would be simple to perform using **trialr**, users may be put off by the computational overhead of performing MCMC sampling. For the purposes of illustration, let us assume that model fitting by MCMC sampling takes 1 second, and that simulations would investigate performance of a dose-finding trial using 30 patients, evaluated in cohorts of three. Each simulated replicate would require 10 model fits, taking 10 seconds. A study of 1000 replicates would take over 2.5 hours. By time alone, this is vastly inferior to functions that conduct simulations using non-MCMC methods such as those offered by **dfcrm** (Cheung 2013). Nevertheless, simulations remain a key tool for clinical trialists to justify the use of such adaptive designs, and will be developed in **trialr** in future.

We also demonstrated in this manuscript the AugBin model of Wason and Seaman (2013) for single-arm trials. The authors also include a two-arm version suitable for randomised trials. The authors also latterly introduced versions of the model suitable for trials with an arbitrary number of post-baseline assessments. These variants are not currently in **trialr** but they will be added in the future, supporting the same generic functions.

We investigated two sets of priors with the AugBin model. The diffuse priors we used could hardly be described as generative or representing any reasonable person's beliefs because they yield prior predictive outcomes that are plainly implausible. However, our informative priors that are broadly generative of plausible outcomes a-priori, exhibit possibly undesirable numerical performance, with relatively large bias and low interval coverage. Knowing how to set priors for this Bayesian version of the AugBin model is an area worthy of further research.

Despite being an R package, the Stan implementations of the models in **trialr** can be used with **StataStan** and **PyStan** to fit the models in Stata and Python respectively.

trialr was first added to CRAN in Nov-2017 and is available at `https://CRAN.R-project.org/package=trialr`. Source code is available on GitHub at `https://github.com/brockk/trialr`. The source for this article is available at `https://github.com/brockk/trialr-jss`.

5. Acknowledgements

presentated at StanCon, Helsinki, 2018. The author extends to them their gratitude.

6. Appendix

6.1. Reproducing Figures in CRM sections

The following code creates Figure 2. The command `render_graph` displays the graph in the RStudio Viewer.

```
R> library(tibble)
R> library(dplyr)
R> paths2_df <- as_tibble(paths2)
R>
R> library(DiagrammeR)
R> # DiagrammeR requires data.frames of nodes and edges to create a graph.
R> paths2_df %>%
R>   transmute(id = .node,
R>             type = NA,
R>             label = case_when(
R>               is.na(next_dose) ~ 'Stop',
R>               TRUE ~ next_dose %>% as.character()),
R>             shape = 'circle',
R>             fillcolor = case_when(
R>               next_dose == 1 ~ 'slategrey',
R>               next_dose == 2 ~ 'skyblue1',
R>               next_dose == 3 ~ 'royalblue1',
R>               next_dose == 4 ~ 'orchid4',
R>               next_dose == 5 ~ 'royalblue4',
R>               is.na(next_dose) ~ 'red'
R>             )
R>   ) -> nodes_df
R> paths2_df %>%
R>   filter(!is.na(.parent)) %>%
R>   select(from = .parent, to = .node, label = outcomes) %>%
R>   mutate(rel = "leading_to") -> edges_df
R> graph <- create_graph(nodes_df = nodes_df, edges_df = edges_df)
R> render_graph(graph)
```

6.2. Reproducing Figures in EffTox sections

The following code creates Figures 3 and 4.

```
R> library(tibble)
R> library(purrr)
R> library(dplyr)
```

```
R> library(tidyr)
R> library(ggplot2)
R>
R> fit_thall_2014 <- function(outcomes, alpha_sd, beta_sd, gamma_sd, zeta_sd) {
R>   stan_efftox(outcome_str = outcomes,
R>               real_doses = c(1.0, 2.0, 4.0, 6.6, 10.0),
R>               efficacy_hurdle = 0.5, toxicity_hurdle = 0.3,
R>               p_e = 0.1, p_t = 0.1, eff0 = 0.5, tox1 = 0.65,
R>               eff_star = 0.7, tox_star = 0.25,
R>               alpha_mean = -7.9593, alpha_sd = alpha_sd,
R>               beta_mean = 1.5482, beta_sd = beta_sd,
R>               gamma_mean = 0.7367, gamma_sd = gamma_sd,
R>               zeta_mean = 3.4181, zeta_sd = zeta_sd,
R>               eta_mean = 0, eta_sd = 0.2,
R>               psi_mean = 0, psi_sd = 1,
R>               seed = 123, refresh = 0)
R> }
R>
R> expand.grid(
R>   alpha_sd = c(2, 3, 4),
R>   beta_sd = c(2, 3, 4),
R>   gamma_sd = 2.5423,
R>   zeta_sd = 2.4406
R> ) %>%
R>   as_tibble() %>%
R>   mutate(fit = pmap(list(alpha_sd, beta_sd, gamma_sd, zeta_sd),
R>                     fit_thall_2014, outcomes = ''),
R>          series = rownames(.)) -> prior_fits1
R>
R> prior_fits1 %>%
R>   mutate(
R>     dose = map(fit, 'dose_indices'),
R>     prob_obd = map(fit, 'prob_obd'),
R>     entropy = map_dbl(fit, 'entropy')
R>   ) %>%
R>   select(-fit) %>%
R>   unnest %>%
R>   ggplot(aes(x = dose, y = prob_obd, fill = entropy)) +
R>   geom_col() +
R>   facet_grid(~ alpha_sd ~ beta_sd) +
R>   labs(y = 'Prob(dose has maximal utility)',
R>        title = 'Effect of prior SD of alpha (rows) and beta (columns)') +
R>   theme(legend.position = 'bottom')

R> expand.grid(alpha_sd = 3.5487,
R>             beta_sd = 3.5018,
R>             gamma_sd = c(2, 3, 4),
```

```
R>              zeta_sd = c(2, 3, 4)
R> ) %>%
R>   as_tibble() %>%
R>   mutate(fit = pmap(list(alpha_sd, beta_sd, gamma_sd, zeta_sd),
R>                     fit_thall_2014, outcomes = ''),
R>         series = rownames(.)) -> prior_fits2
R>
R> prior_fits2 %>%
R>   mutate(
R>     dose = map(fit, 'dose_indices'),
R>     prob_obd = map(fit, 'prob_obd'),
R>     entropy = map_dbl(fit, 'entropy')
R>   ) %>%
R>   select(-fit) %>%
R>   unnest %>%
R>   ggplot(aes(x = dose, y = prob_obd, fill = entropy)) +
R>   geom_col() +
R>   facet_grid(~ gamma_sd ~ zeta_sd) +
R>   labs(y = 'Prob(dose has maximal utility)',
R>        title = 'Effect of prior SD of gamma (rows) and zeta (columns)') +
R>   theme(legend.position = 'bottom')
```

6.3. Reproducing Figures in AugBin sections

The following code produce Figures 7.

```
R> set.seed(123)
R> diffuse_prior_pred_data <- do.call(prior_predictive_augbin_2t_1a,
R>                             append(diffuse_priors,
R>                                    list(num_samps = 1000)))
R> inf_prior_pred_data <- do.call(prior_predictive_augbin_2t_1a,
R>                         append(informative_priors,
R>                                list(num_samps = 1000)))
R>
R> library(stringr)
R> library(tidyr)
R>
R> # Visualise tumour sizes
R> bind_rows(
R>   diffuse_prior_pred_data %>% mutate(Prior = 'Diffuse'),
R>   inf_prior_pred_data %>% mutate(Prior = 'Informative')
R> ) %>%
R>   select(id, Prior, y0, y1, y2) %>%
R>   gather(assessment, y, -id, -Prior) %>%
R>   mutate(time = str_extract(assessment, '\\d+') %>% as.integer()) %>%
R>   ggplot(aes(x = time, y = y)) +
```

```
R>   geom_line(aes(group = id), alpha = 0.3, col = 'darkgray') +
R>   geom_hline(yintercept = log(0.7), col = 'orange', linetype = 'dashed') +
R>   scale_x_continuous(breaks = 0:2) +
R>   labs(y = 'log(tumour size ratio)') +
R>   facet_wrap(~ Prior)
```

The following code produce Figures 8.

```
R> bind_rows(
R>   diffuse_prior_pred_data %>% mutate(Prior = 'Diffuse'),
R>   inf_prior_pred_data %>% mutate(Prior = 'Informative')
R> ) %>%
R>   ggplot(aes(x = z0, y = prob_d1)) +
R>   geom_point() + geom_smooth(method = 'gam') +
R>   labs(x = 'Baseline tumour size (cm)',
R>        y = 'Prob(Non-shrikage failure at time 1)') +
R>   facet_wrap(~ Prior)
```

The following code produces Figure 9.

```
R> pred_inf <- predict(fit_inf, y2_upper = log(0.7))
R> pred_diffuse <- predict(fit_diffuse, y2_upper = log(0.7))
R> pred_binary <- binary_prob_success(fit_diffuse,
R>                                     y2_upper = log(0.7),
R>                                     methods = 'exact')
R>
R> pred_diffuse %>%
R>   ggplot(aes(x = z0, y = prob_success, col = z1)) +
R>   geom_point() +
R>   geom_linerange(aes(ymin = lower, ymax = upper)) +
R>   geom_hline(yintercept = pred_binary$mean, col = 'orange',
R>              linetype = 'dashed') +
R>   geom_hline(yintercept = pred_binary$lower, col = 'orange',
R>              linetype = 'dotted') +
R>   geom_hline(yintercept = pred_binary$upper, col = 'orange',
R>              linetype = 'dotted') +
R>   ylim(0, 0.75) +
R>   labs(x = 'Baseline tumour size (cm)', y = 'Prob(Success)',
R>        col = 'Interim size (cm)', title = 'Diffuse') +
R>   theme(legend.position = 'bottom') -> aubgin_diffuse_plot
R>
R> pred_inf %>%
R>   ggplot(aes(x = z0, y = prob_success, col = z1)) +
R>   geom_point() +
R>   geom_linerange(aes(ymin = lower, ymax = upper)) +
R>   geom_hline(yintercept = pred_binary$mean, col = 'orange',
R>              linetype = 'dashed') +
```

```
R>    geom_hline(yintercept = pred_binary$lower, col = 'orange',
R>             linetype = 'dotted') +
R>    geom_hline(yintercept = pred_binary$upper, col = 'orange',
R>             linetype = 'dotted') +
R>    ylim(0, 0.75) +
R>    labs(x = 'Baseline tumour size (cm)', y = 'Prob(Success)',
R>        col = 'Interim size (cm)', title = 'Informative') +
R>    theme(legend.position = 'bottom') -> aubgin_inf_plot
R>
R> gridExtra::grid.arrange(aubgin_diffuse_plot, aubgin_inf_plot, ncol = 2)
```

References

Braun TM (2002). "The Bivariate Continual Reassessment Method: Extending the CRM to Phase I Trials of Two Competing Outcomes." *Controlled Clinical Trials*, **23**(3), 240–256. ISSN 01972456. doi:10.1016/S0197-2456(01)00205-7.

Brock K, Billingham L, Copland M, Siddique S, Sirovica M, Yap C (2017). "Implementing the EffTox Dose-Finding Design in the Matchpoint Trial." *BMC Medical Research Methodology*, **17**(1), 112. ISSN 1471-2288. doi:10.1186/s12874-017-0381-x.

Carpenter B, Gelman A, Hoffman M, Lee D, Goodrich B, Betancourt M, Brubaker MA, Li P, Riddell A (2016). "Stan: A Probabilistic Programming Language." *Journal of Statistical Software*, **76**(Ii), 1–32.

Carter S (1973). "Study Design Principles for the Clinical Evaluation of New Drugs as Developed by the Chemotherapy Programme of the National Cancer Institute." In *The Design of Clinical Trials in Cancer Therapy*, pp. 242–289. Editions Scientifique Europe.

Cheung K (2013). *dfcrm: Dose-finding by the continual reassessment method*. R package version 0.2-2, URL https://CRAN.R-project.org/package=dfcrm.

Cheung YK (2011). *Dose Finding by the Continual Reassessment Method*. Chapman & Hall / CRC Press, New York. ISBN 978-1-4200-9151-9.

Cheung YK, Chappell R (2000). "Sequential Designs for Phase I Clinical Trials with Late-Onset Toxicities." *Biometrics*, **56**(4), 1177–1182. ISSN 0006-341X.

Chiuzan C, Shtaynberger J, Manji GA, Duong JK, Schwartz GK, Ivanova A, Lee SM (2017). "Dose-Finding Designs for Trials of Molecularly Targeted Agents and Immunotherapies." *Journal of Biopharmaceutical Statistics*, **27**(3), 477–494. ISSN 1054-3406. doi:10.1080/10543406.2017.1289952.

Craddock C, Slade D, De Santo C, Wheat R, Ferguson P, Hodgkinson A, Brock K, Cavenagh J, Ingram W, Dennis M, Malladi R, Siddique S, Mussai F, Yap C (2019). "Combination Lenalidomide and Azacitidine: A Novel Salvage Therapy in Patients Who Relapse After Allogeneic Stem-Cell Transplantation for Acute Myeloid Leukemia." *Journal of Clinical Oncology*, p. JCO.18.00889. ISSN 0732-183X, 1527-7755. doi:10.1200/JCO.18.00889.

Dorai-Raj S (2014). *binom: Binomial Confidence Intervals For Several Parameterizations*. R package version 1.1-1, URL https://CRAN.R-project.org/package=binom.

Eisenhauer EA, Therasse P, Bogaerts J, Schwartz LH, Sargent D, Ford R, Dancey J, Arbuck S, Gwyther S, Mooney M, Rubinstein L, Shankar L, Dodd L, Kaplan R, Lacombe D, Verweij J (2009). "New Response Evaluation Criteria in Solid Tumours: Revised RECIST Guideline (Version 1.1)." *European Journal of Cancer*, **45**(2), 228–247. ISSN 09598049. doi:10.1016/j.ejca.2008.10.026.

Henry L, Wickham H (2019). *purrr: Functional Programming Tools*. R package version 0.3.2, URL https://CRAN.R-project.org/package=purrr.

Herbst RS, Baas P, Kim DW, Felip E, Pérez-Gracia JL, Han JY, Molina J, Kim JH, Arvis CD, Ahn MJ, Majem M, Fidler MJ, De Castro G, Garrido M, Lubiniecki GM, Shentu Y, Im E, Dolled-Filhart M, Garon EB (2016). "Pembrolizumab versus Docetaxel for Previously Treated, PD-L1-Positive, Advanced Non-Small-Cell Lung Cancer (KEYNOTE-010): A Randomised Controlled Trial." *The Lancet*, **387**(10027), 1540–1550. ISSN 1474547X. doi:10.1016/S0140-6736(15)01281-7.

Iannone R (2018). *DiagrammeR: Graph/Network Visualization*. R package version 1.0.0, URL https://CRAN.R-project.org/package=DiagrammeR.

Iasonos A, Wilton AS, Riedel ER, Seshan VE, Spriggs DR (2008). "A Comprehensive Comparison of the Continual Reassessment Method to the Standard $3 + 3$ Dose Escalation Scheme in Phase I Dose-Finding Studies." *Clinical trials (London, England)*, **5**(5), 465–477. ISSN 1740-7745. doi:10.1177/1740774508096474.

Ji Y, Wang SJ (2013). "Modified Toxicity Probability Interval Design: A Safer and More Reliable Method than the $3 + 3$ Design for Practical Phase I Trials." *Journal of clinical oncology : official journal of the American Society of Clinical Oncology*, **31**(14), 1785–1791. ISSN 15277755. doi:10.1200/JCO.2012.45.7903.

Kay M (2019). *tidybayes: Tidy Data and Geoms for Bayesian Models*. doi:10.5281/zenodo.1308151. R package version 1.0.4, URL http://mjskay.github.io/tidybayes/.

Le Tourneau C, Lee JJ, Siu LL (2009). "Dose Escalation Methods in Phase i Cancer Clinical Trials." *Journal of the National Cancer Institute*, **101**(10), 708–720. ISSN 00278874. doi:10.1093/jnci/djp079.

O'Quigley J, Paoletti X (2003). "Continual Reassessment Method for Ordered Groups." *Biometrics*, **59**(2), 430–440. ISSN 1541-0420. doi:10.1111/1541-0420.00050.

O'Quigley J, Pepe M, Fisher L (1990). "Continual Reassessment Method: A Practical Design for Phase 1 Clinical Trials in Cancer." *Biometrics*, **46**(1), 33–48. ISSN 0006-341X. doi:10.2307/2531628.

O'Quigley J, Zohar S (2006). "Experimental Designs for Phase I and Phase I/II Dose-Finding Studies." *British Journal of Cancer*, **94**(5), 609–613. ISSN 0007-0920. doi:10.1038/sj.bjc.6602969.

Rogatko A, Schoeneck D, Jonas W, Tighiouart M, Khuri FR, Porter A (2007). "Translation of Innovative Designs into Phase I Trials." *Journal of Clinical Oncology*, **25**(31), 4982–4986. ISSN 0732183X. doi:10.1200/JCO.2007.12.1012.

Sabanes Bove D, Yin Yeung W, Palermo G, Jaki T (2018). *crmPack: Object-Oriented Implementation of CRM Designs*. R package version 0.2.9, URL https://CRAN.R-project.org/package=crmPack.

Sweeting M, Mander A, Sabin T (2013). "bcrm: Bayesian Continual Reassessment Method Designs for Phase I Dose-Finding Trials." *Journal of Statistical Software*, **54**(13), 1–26. URL http://www.jstatsoft.org/article/view/v054i13.

Thall P, Cook J (2004). "Dose-Finding Based on Efficacy-Toxicity Trade-Offs." *Biometrics*, **60**(3), 684–693.

Thall P, Cook J, Estey E (2006). "Adaptive Dose Selection Using Efficacy-Toxicity Trade-Offs: Illustrations and Practical Considerations." *Journal of biopharmaceutical statistics*, **16**(5), 623–638. ISSN 1054-3406. doi:10.1080/10543400600860394.

Thall P, Herrick R, Nguyen H, Venier J, Norris J (2014). "Effective Sample Size for Computing Prior Hyperparameters in Bayesian Phase I-II Dose-Finding." *Clinical Trials*, **11**(6), 657–666. ISSN 1740-7745. doi:10.1177/1740774514547397.

Thall PF, Wathen JK, Bekele BN, Champlin RE, Baker LH, Benjamin RS (2003). "Hierarchical Bayesian Approaches to Phase II Trials in Diseases with Multiple Subtypes." *Statistics in Medicine*, **22**(5), 763–780. ISSN 02776715. doi:10.1002/sim.1399.

Tighiouart M, Rogatko A (2010). "Dose Finding with Escalation with Overdose Control (EWOC) in Cancer Clinical Trials." *Statistical Science*, **25**(2), 217–226. ISSN 0883-4237. doi:10.1214/10-STS333. 1011.6479v1.

Wages NA, Tait C (2015). "Seamless Phase I/II Adaptive Design For Oncology Trials of Molecularly Targeted Agents." *Journal of biopharmaceutical statistics*, pp. 1–30. ISSN 1520-5711. doi:10.1080/10543406.2014.920873.

Wason JMS, Seaman SR (2013). "Using Continuous Data on Tumour Measurements to Improve Inference in Phase II Cancer Studies." *Statistics in Medicine*, **32**(26), 4639–4650. ISSN 02776715. doi:10.1002/sim.5867.

Wickham H (2016). *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York. ISBN 978-3-319-24277-4. URL https://ggplot2.tidyverse.org.

Wickham H (2017). *tidyverse: Easily Install and Load the 'Tidyverse'*. R package version 1.2.1, URL https://CRAN.R-project.org/package=tidyverse.

Wickham H, François R, Henry L, Müller K (2019). *dplyr: A Grammar of Data Manipulation*. R package version 0.8.1, URL https://CRAN.R-project.org/package=dplyr.

Yap C, Billingham LJ, Cheung YK, Craddock C, O'Quigley J (2017). "Dose Transition Pathways: The Missing Link between Complex Dose-Finding Designs and Simple Decision-Making." *Clinical Cancer Research*, **23**(24), 7440–7447. ISSN 15573265. doi:10.1158/1078-0432.CCR-17-0582.

Zhang W, Sargent DJ, Mandrekar S (2006). "An Adaptive Dose-Finding Design Incorporating Both Toxicity and Efficacy." *Statistics in Medicine*, **25**(14), 2365–2383. ISSN 02776715. doi:10.1002/sim.2325.

Affiliation:

Kristian Brock
Cancer Research UK Clinical Trials Unit, University of Birmingham
Birmingham, B15 2TT, UK
E-mail: k.brock@bham.ac.uk
URL: https://github.com/brockk/