Received 26 April 2016; | Revised 6 June 2016; | Accepted 6 June 2016

DOI: xxx/xxxx

ARTICLE TYPE

Bayesian Reproducibility

Author One*[1] | Author Two[2,3] | Author Three[3]

[1]Org Division, Org Name, State name, Country name
[2]Org Division, Org Name, State name, Country name
[3]Org Division, Org Name, State name, Country name

Correspondence
*Corresponding author name, This is sample corresponding address. Email: authorone@gmail.com

Present Address
This is sample for present address text this is sample for present address text

Summary

KEYWORDS:
keyword1, keyword2, keyword3, keyword4

```
knitr::opts_chunk$set(message = FALSE, warning = FALSE, cache=TRUE)
mu<-I<-sd<-overall<-NULL

library(knitr)
library(ggplot2)
library(ggthemes)
require(reshape2)

## Loading required package: reshape2

library(doParallel)

## Loading required package: foreach
## Loading required package: iterators
## Loading required package: parallel

library(R2jags)

## Loading required package: rjags
## Loading required package: coda
## Linked to JAGS 4.2.0
## Loaded modules: basemod,bugs
##
## Attaching package: 'R2jags'
## The following object is masked from 'package:coda':
##
##    traceplot
```

```
library(random)
require(gridExtra)

## Loading required package: gridExtra
```

1 | INTRODUCTION

P-values have been on of the reasons behind lack of reproducibility in scientific discoveries and especially in replicated studies and multiple testing[1]. This problem is especially prevalent in Genome-Wide Association Studies (GWAS), where estimated effects have upward bias and often fail to replicate in validation studies. This phenomenon is known as the winner's curse[2]. To account for this discrepancy, previous studies perform two analyses: one with all the data together, and one only using the validation site data. However, this approach is based on the underlying assumption that the association found in discovery sites is true, which is problematic for multiple testing applications such as genome-wide association studies. Furthermore, if there is a true effect, leaving out the discovery data, which could be a large portion of the total dataset, reduces power.

One example of the winner's curse in action is the analysis of the association between single nucleotide polymorphisms (SNPs) in the p53 protein, which is needed for cell growth and DNA repair, and invasive ovarian cancer. Three independent discovery studies focused on TP53 polymorphisms and risk of ovarian cancer: the North Carolina Ovarian Cancer Study (NCOCS), the Mayo Clinic Case-Control Study (MAYO), and the Polish Ovarian Cancer Study (POCS). These were restricted to non-Hispanic white women with newly diagnosed, histologically confirmed, primary invasive epithelial ovarian cancer and to non-Hispanic white controls. 23 SNPs were genotyped in total, with some overlap between sites. Ten other sites contributed data: the Australian Ovarian Cancer Study (AOCS) and the Australian Cancer Study (ACS) presented together as AUS, the Family Registry for Ovarian Cancer (FROC, presented as STA), the Hawaiian Ovarian Cancer Study (HAW), the Malignant Ovarian Cancer Study Denmark (MALOVA), the New England Case-Control Study (NEC), the Nurses' Health Study (NHS), SEARCH Cambridge (SEA), the Los Angeles County Case-Control Study of Ovarian Cancer (LAC-CCOC, presented here as USC), the University of California at Irvine study (UCI), and the United Kingdom Ovarian Cancer Population Study (UKOPS, presented here as UKO). The combined data set (discovery and replication) comprised 5,206 white, non-Hispanic invasive epithelial ovarian cancer cases, of which 2,829 were classified as serous invasive ovarian cancer, and 8,790 white non-Hispanic controls. Analysis was restricted to white, non-Hispanic invasive serous ovarian cancer cases and white, non-Hispanic controls.

Mixed effect SNP-at-a-time analysis of 5 SNPs that were chosen for replication resulted in associations between 2 SNPs and serous invasive cancer[3]. Only one of these was strongly supported to be associated in a follow-up analysis using Multi-level Inference for SNP Association (MISA), which employs Bayesian Model Averaging and Bayes Factors for selection[4]. However, most recent studies with added data have not found evidence of association between any TP53 SNPs and cancer[5].

The aim of this project is twofold: to explore ways in which discovery findings can be reported to avoid the winner's curse, and to combine discovery results with validation data in a coherent manner accounting for the selection effect.

After a review of existing literature, we propose three approaches to adress this: a fully Bayesian model, a conditional likelihood prior for discovery site data, and a Bayes Factor approximation to the probability of association. The performance of these methods is tested on normal simulations, and then on hierarchical simulations split into discovery and validation "sites". All three proposed methods provide improvements over naive models. Finally, the models are used to reanalyze the tp53 SNPs; none of the SNPs are found to be significant under any model.

2 | LITERATURE REVIEW

Misuse of p-values and lack of reproducibility in scientific discoveries have been a cause for concern in the scientic world, leading to proposals of new ways to define significance. Benjamin et. al have shown that the Bayes factor equivalents for commonly used p-values only correspond to "weak" evidence in the Bayes factor characterization[1]. They suggest reducing the p-value threshold in studies with less power, but acknowledge that hypothesis testing with thresholding is still an issue. Another approach proposes two calibrations of the p-value: as the lower bound of the Bayes factor under any alternative hypothesis, and as a posterior probability of the type 1 error in a Bayesian framework[6].

This problem has become a major issue in replicated studies, an effect known as the "winner's curse"[2] or the Beavis effect[7]. Zollner and Pritchard first define this in the context of genome-wide association scans (GWAS), which use stringent thresholds for significance, resulting in inflated effect sizes after selection, especially since these are calculated with the same data. Thus, replication studies underestimate the sample size necessary and do not have enough power to detect an effect. They suggest a conditional-likelihood based method to address this issue, proposing a computational algorithm to maximize over the the likelihood of the parameters conditional on the significance association at level α, which results in less biased coefficient estimates (albeit with larger variance) and sample size estimates centered at the true value[2].

Zhong and Prentice also propose a similar method, but use a different parametrization and an asymptotic approximation instead of a computational one to find the estimators, which is more computationally efficient[8]. Ghosh et al. also define an approximate conditional likelihood, and propose two more estimators (other than the MLE): the mean of the (normalized) conditional likelihood, which can be interpreted as a posterior mean of the parameters under a flat prior, and a "compromise" estimator which is the average of the mean and MLE[9]. The combination estimator proves to have the most stable MSE accross the range of true values for the parameters. Their approach only requires summary statistics, so they further apply it to published datasets. The results are similar for the three conditional likelihood approaches.

Another method proposed to create bias-reduced estimates uses bootstrap re-sampling to correct for both the thresholding effect and the ranking effect, which is not addressed in the conditional likelihood methods because of the difficulty of specifying joint likelihoods for correlated variables[10]. By using a sample-split approach, the detection and estimation datasets can be virtually independent. This is repeated multiple times in order to reduce variance in the results. The main drawback of this approach is its computational intensity.

Several authors have also proposed shrinkage-based methods in the effect detection step. Bacanu and Kendler use a soft threshold method to scale statistics such that their sum of squares do not overestimate the true mean and then find "suggestive" signals in a GWAS context by setting a threshold. This method does not address the winner's curse directly, but provides a subset of the genome which can be futher analyzed or used in future studies[11]. Bigdelli et al. propose shrinking coefficient estimates by drawing a comparison between "winner's curse adjustments" for effect sizes and multiple testing approaches for p-values, since both are used on the tail of their respective distributions. Their method transforms False Discovery Rate (FDR) adjusted p-values into the corresponding Z-score and uses that as the estimator[12]. Both Bigdelli and Bacanu assume the data is normally distributed. Storey and Tibshirani, on the other hand, propose to adjust the value used for significance testing rather than the coefficients, choosing the FDR value as an alternative to the p-valu e[13].

Multiple Bayesian methods have also been proposed: Xu et al use a Bayesian approach to a logistic regression, selecting a spike and slab prior for the mean and an inverse gamma prior for the variance[14]. A beta prior for the proportion of each component in the prior, and the hyperparameters were estimated empirically. They also propose a Bayesian Model Average approach, which they recommend for instances with little prior information. Their results show that the Bayesian models had smaller variance than conditional likelihood methods, but still do not address the "ranking effect"[10], or implement a fully Bayesian approach because of the dependence on the threshold α.

Ferguson et al propose an Empirical Bayes approach, which estimate the prior density distribution with the data[15]. This is a nonparametric estimate, but still depends on other specifications such as the number of bins, type of splines, etc. Using the empirical prior, the posterior is then calculated, from which the estimate and pseudo-Bayesian credible intervals are derived by considering the 5% and 95% points. This method resulted in better estimates in the higher density regions, but performed worse than conditional likelihood methods on the tails. Thus, the authors propose a combined method, which calculates both the empirical Bayes and the conditional likelihood confidence intervals, and picks the shortest one. One possible problem with this approach is the use of non-HPD intervals, which could change the tail behavior.

The Bayesian framework is also applied to power calculations specifically, defining "Bayesian power" as the marginal probability of finding significance in a replicated study given the original and the data. In this paper, a spike and slab prior is also used, but the hyperparameters are estimated empirically. The resulting power estimators are improved, but lead to downwards bias in the effect size[16].

3 | MODELS

Following the structure of the motivating example, consider a binary dataset describing an event such as whether or not someone has ovarian cancer. This data is collected across different sites, which may have different sampling procedures (as well as simply different populations). The mean effect for each site can be thought of as normally distributed around a global effect if this exists. This is the alternative hypothesis. The null hypothesis is that there is no global effect. This also means that there must not be a site effect. We are interested in two things: whether or not there is a global effect (hypothesis testing), and what the effect's size is (inference). Let Y_{ij} be the observed data. The j index corresponds to the site to which the observation belongs.

$$P(Y_{ij} = 1|\beta_j) = \mathsf{logit}^{-1}(\beta_j) \tag{1}$$

$$\beta_j|\mu, \sigma^2, H_1 \sim \mathsf{N}(\mu, \sigma^2) \tag{2}$$

$$\beta_j|H_0 = 0 \tag{3}$$

In the case where we have multiple sites' information, we can use this hierarchical model, but if we only have data from one discovery site, then the site effect becomes meaningless, and we can use the following model:

$$P(Y_i = 1|\mu) = \mathsf{logit}^{-1}(\mu) \tag{4}$$

We propose three different Bayesian approaches:

1. A fully Bayesian mixed effects hierarchical model that can jointly perform significance testing and effect estimation. By combining the testing and estimate steps, we can overcome the winner's curse and account for the uncertainty that arises when selecting an SNP. This model builds on Xu et al., and Jiang et al., which introduce the spike-and-slab prior, with the addition of random effects to account for heterogeneity between sites.

2. A conditional likelihood model that can take into account the probability of finding a significant result in the discovery sites when estimating effect size. This model incorporates the conditional likelihood introduced by Zollner and Pritchard as well as Zhong and Prentice, and Ghosh et al.)[2,8,9] and incorporates it into the Bayesian hierarchical framework.

3. A Bayes factor-based model that uses an upper bound on the Bayes factor from the discovery sites, which is more reliable than the p-value[1] to quantify the uncertainty of the significant result.

While the first approach is truly Bayesian and requires all the data, the second and third can be used as long as the sufficient statistics (MLE, SE, p-value, α) are available.

3.1 | Fully Bayesian Model

Let $\delta_a(x)$ be the Dirac delta function: $\delta_a(x) = 1$ for $x = a$ and $\delta_a(x) = 0$ otherwise. Then, $P(\mu|H_0) = \delta_0(\mu)$, and $P(\mu|H_1) = N(0, 1)$ or some other diffuse prior. We can define a hyperparameter ξ such that $P(H_1) = \xi$. This gives rise to a latent variable drawn from a Bernoulli(ξ), which is equivalent to selecting H_1 or H_0.

Site means $\beta_j|\mu, \sigma^2, H_1 \sim N(\mu, \sigma^2)$, and $\beta_j|H_0 = 0$. In this case the prior for $\mu|H_1$ was chosen to be a Cauchy distribution. The prior for σ was a truncated Cauchy, with support only on the positive real line. The prior for ξ, the probability of the alternative, was a Beta distribution.

The complete model is as follows:

$$\beta_j|H_1 \sim \mathsf{N}(\mu, \sigma_\beta^2) \tag{5}$$

$$\mu|H_1 \sim \mathsf{Cauchy}(0, \sigma_\mu^2) \tag{6}$$

$$\mu, \beta_j|H_0 = 0 \tag{7}$$

$$\sigma_\beta \sim \mathsf{Cauchy}^+(0, \sigma_\sigma^2) \tag{8}$$

$$H \sim \mathsf{Bernoulli}(\xi) \tag{9}$$

$$\xi \sim \mathsf{Beta}(a, b) \tag{10}$$

```r
cond.likelihood<- function(ybar, mu, SE, q=1.96){
  dnorm(ybar, mu, SE)/
    (pnorm(-q*SE, mu, SE) +
       1-pnorm(q*SE, mu, SE))
}
```

```
p = 0.00325
LOR = log(1.65)
q = -qnorm(p/2)
# 1.21-2.25
SE =  (log(2.25 ) - LOR)/1.96
#exp(LOR + c(-1, 1)*1.96*SE)
BF = 1/(-exp(1)*p*log(p))
mu <- seq(-.2,1.25,.01)

p<-ggplot(data = data.frame(maxy = dnorm(.4, .4,.2)/.6, x = mu, y = dnorm(mu, .4,.2), xl=0, xu=0,yl=0,yu=
  geom_line(aes(x,y/maxy))+geom_segment(aes(x=xl,y=yl,xend=xu,yend=yu))+
  labs(title= "Mixture Model with Point Mass at 0",x=expression(mu),y="density")+
  theme_minimal() + scale_colour_few()
```

If we only consider discovery data from one site, this model becomes the same as the one proposed by Xu et al. with slightly different priors.

There is no difference between discovery and validation sites in the Bayesian framework. Even considering them separately, one could consider the posterior distributions of the parameters given only discovery site data as the priors given the validation data, which would result in exactly the same results.

3.2 | Conditional Likelihood Model

In this case, the results from the discovery sites are used as a prior for the validation data analysis, which is why only the sufficient statistics are needed.

Given the discovery sites' MLE and SE, we can use the CLT and definition of MLE to state that $\mathrm{MLE}_i \sim N(\beta_i, \mathrm{SE}_i)$. Let B indicate that the data is significant at the level α. The conditional likelihood is

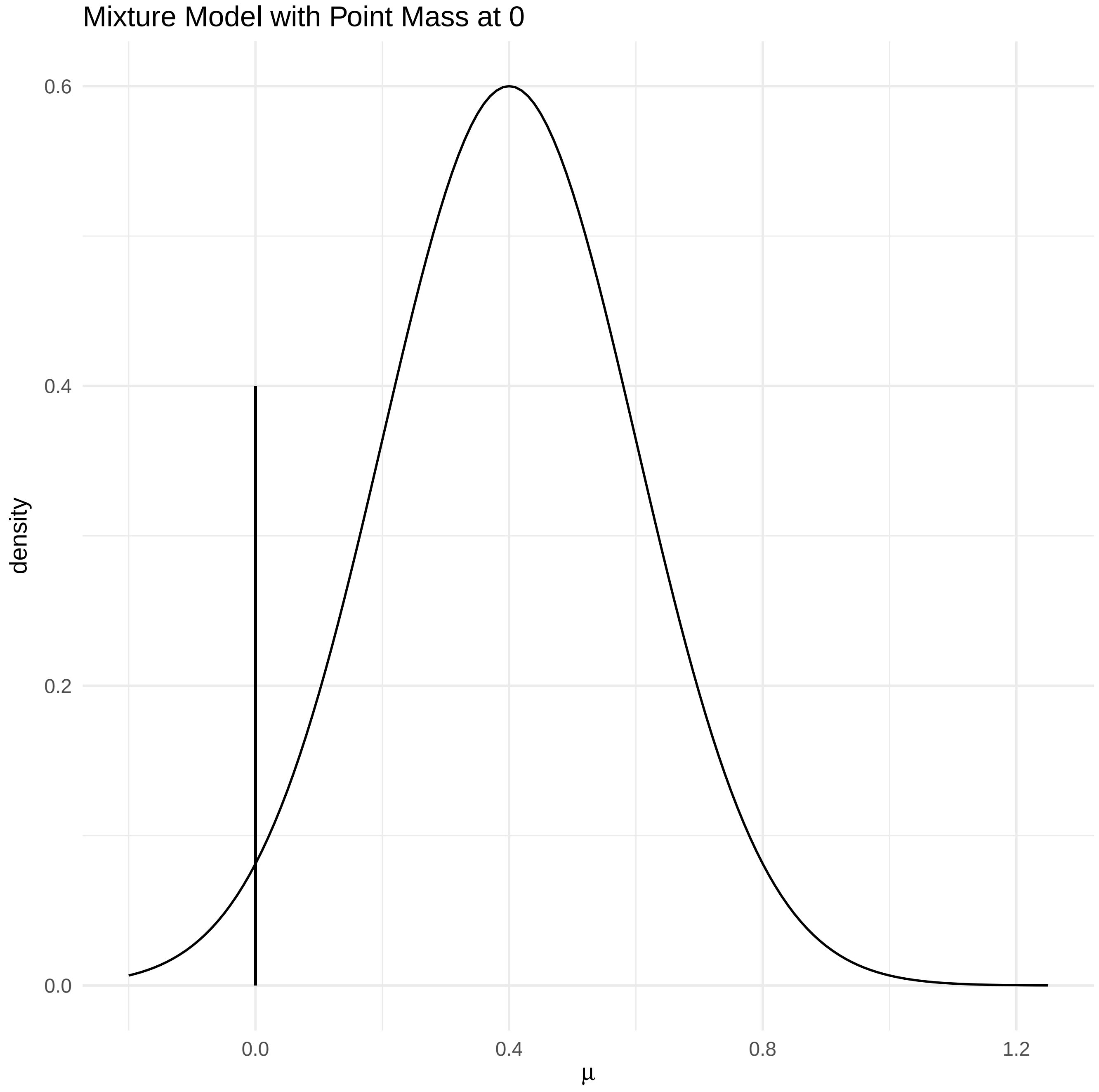
$$L(\mu|B) = \frac{p(Y|\mu)p(B|Y,\mu)}{p(B|\mu)} = \frac{p(Y|\mu)}{\int_{\text{significant Y}} p(t|\mu)dt} \tag{11}$$

Conditioning on finding a significant estimate using a Normal approximation,

$$p(\mathrm{MLE}_i|B, \beta_i, \mathrm{SE}_i, q_i) = \frac{\phi(\mathrm{MLE}_i, \beta_i, \mathrm{SE}_i)}{\Phi(-q_i, \beta_i, \mathrm{SE}_i) + 1 - \Phi(q_i, \beta_i, \mathrm{SE}_i)} \tag{12}$$

where $\phi(x, \beta_i, \sigma)$ is the pdf of a normal distribution with mean β_i and variance σ^2, and $\Phi(x, \beta_i, \sigma)$ is the cdf of the same distribution. The value of q_i is $\Phi^{-1}(1 - \frac{\alpha}{2}, 0, \mathrm{SE}_i)$, where α is the power of the test (i.e. p-values that are smaller than α are considered significant). This is cutoff for an MLE value to be considered significant. Let the conditional likelihood of MLE_i be denoted as $\mathrm{CL}(\beta_i, \mathrm{SE}_i, q_i)$.

```
CLdata<-data.frame(mu, cond.likelihood(LOR, mu, SE, q = -qnorm(0.005/2)),
                   cond.likelihood(LOR, mu, SE, q = -qnorm(0.05/2)),
                   cond.likelihood(LOR, mu, SE, q = -qnorm(0.01/2)),
                   dnorm(LOR, mu, SE))
colnames(CLdata)<- c("mu", "CL.005","CL.05","CL.01","uncond")
melted <- melt(CLdata, id.vars=c("mu"))
p<- ggplot(melted, aes(mu, value, color = variable)) + geom_line(aes(group=variable)) +
  labs(title = "Conditional vs. Unconditional Likelihood",
       x = expression(mu), y = "Likelihood")+  theme_minimal()+
  theme(legend.justification=c(1,1), legend.position=c(1,1)) +
  scale_colour_few(name="", labels=c(expression(paste(alpha," = .005")),
                                     expression(paste(alpha," = .05")),
                                     expression(paste(alpha," = .01")),
                                     "unconditional"))
```



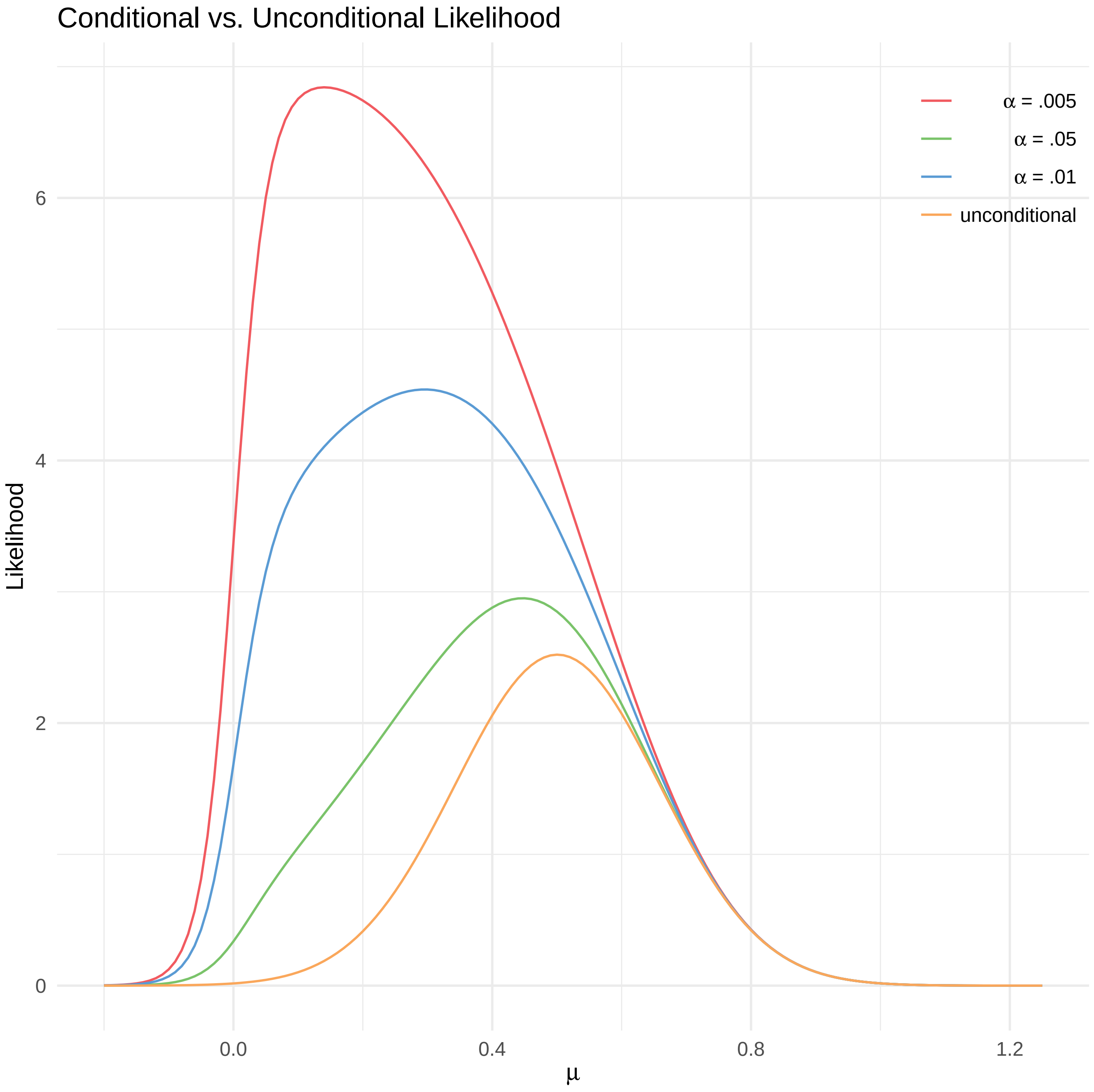
We can see that as α decreases (i.e. the tests are more strict), the likelihood becomes more skewed towards 0.

The conditional likelihood of $\beta_j, j \in$ discovery becomes the posterior of this variable if we use a uniform prior, since the likelihood will just be multiplied by one.

In the discovery-only case, this is enough to create credible intervals for μ by sampling from the posterior (as opposed to maximizing the conditional likelihood or approximating the surface).

In the hierarchical setting, the posteriors for the discovery sites are used as the priors for the validation data; that is, $p(\mu|\beta_i, \mathrm{MLE}_i, \mathrm{SE}_i, q_i, i \in \text{discovery})$ is the prior for the hierarchical model using validation data.

The updated model is:



$$\beta_j|\mu,\sigma_\beta^2 \sim \mathsf{N}(\mu,\sigma_\beta^2), j \in \text{validation} \tag{13}$$

$$\text{MLE}_j|\beta_j,\text{SE}_j,q_j \sim \mathsf{CL}(\beta_j,\text{SE}_j,q_j), j \in \text{discovery} \tag{14}$$

$$\sigma_\beta \sim \mathsf{Cauchy}^+(0,\sigma_\sigma^2) \tag{15}$$

Note that the selection uncertainty is somewhat accounted for through the conditional likelihood, but there is no measure of this uncertainty. By using the discovery MLEs, we are already assuming that there is a nonzero effect.

3.3 | Bayes Factor Model

The discovery data can be used not only in estimating the distribution of the size of a preestablished effect (μ), but in estimating the distribution of the probability of the effect itself ($\xi = P(H_1)$). To make this model easily generalizable, we use the upper bound on the Bayes Factor

$$BF_{H_1:H_0} = \frac{L(\bar{Y}|H_1)}{L(\bar{Y}|H_0)} \leq \frac{1}{-eplog(p)} \tag{16}$$

where p is the p-value from the discovery data[6]. This is a "best-case scenario" of how much evidence there is from data given a particular p-value. Since this value is fixed given the discovery data, we can then consider the "posterior"" probability of true association ξ given the discovery p-value as a transformation of ξ, which is parametrized with prior Beta(.5,.5). Let o be the prior odds $\frac{1-\xi}{\xi}$.

$$\xi' = \frac{P(H_1) * L(Y|H_1)}{P(H_0) * L(Y|H_0) + P(H_1) * L(Y|H_1)} = \frac{o * BF_{H_1}}{1 + o * BF_{H_1}} \tag{17}$$

Then ξ' can be used in the fully Bayesian model, but only with the validation data.

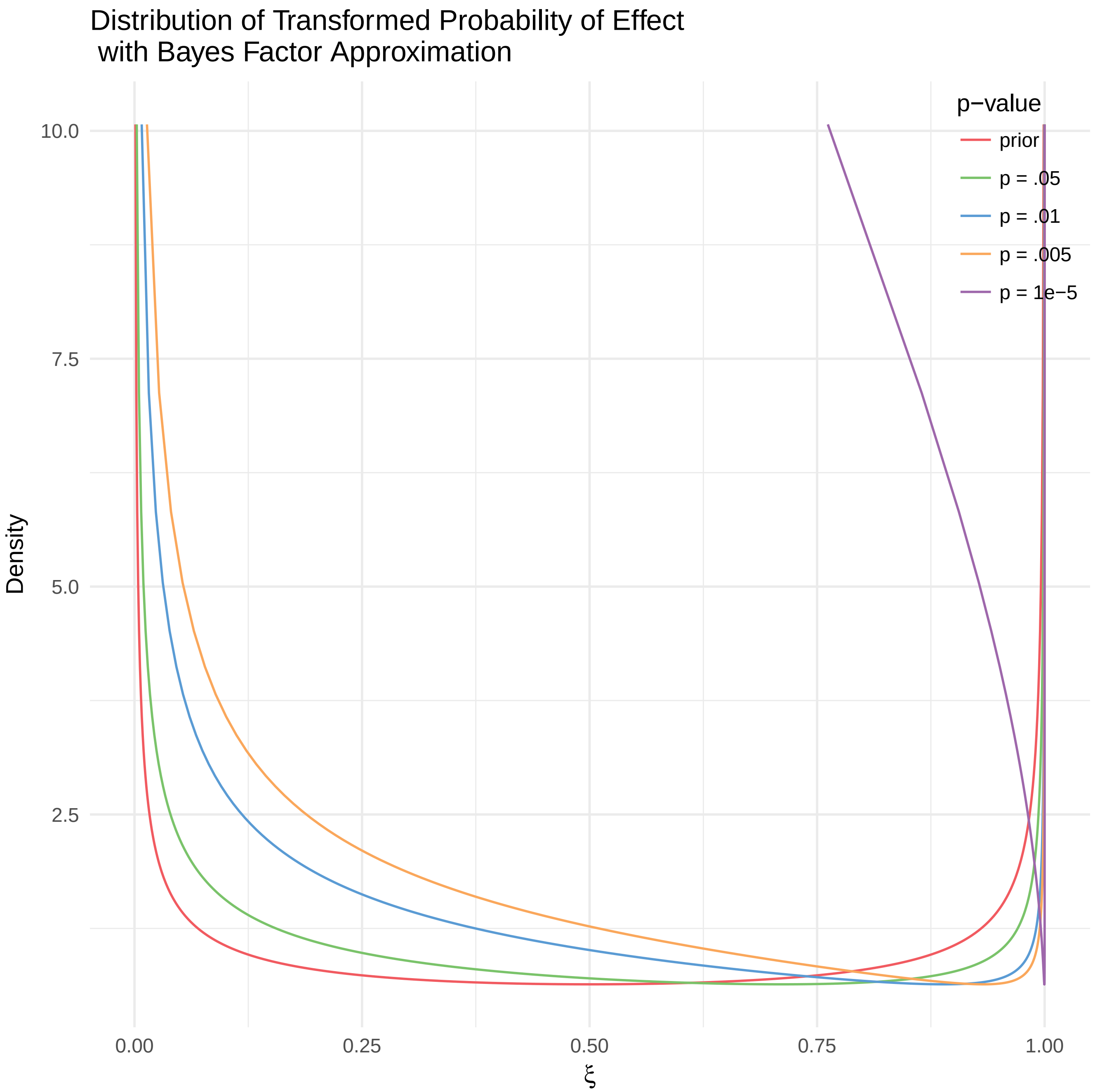
```
pind.dens<-function(post.ind,p){
  BF<- 1/(-exp(1)*p*log(p))
  prior.odds<-post.ind/(1-post.ind)/BF
  pind<- prior.odds/(1+prior.odds) #a/0
  #eplogp is BF h0/h1
  return(pind)
  }
postind.dens<-function(pind, p=.0035){
  prior.odds<- (pind)/(1-pind) #a/0
  BF<- 1/(-exp(1)*p*log(p)) #eplogp is BF h0/h1
  post.ind<-prior.odds*BF/(1+prior.odds*BF)
  return(post.ind)
}
pind<- seq(.001, .999,by=.001)

BFapproxdata<-data.frame(pind, postind.dens(pind, .05),postind.dens(pind, .01),
                         postind.dens(pind, .005),postind.dens(pind, 1e-5),
                   dbeta(pind,.5,.5))
colnames(BFapproxdata)<-c("prior", "p = .05", "p = .01", "p = .005", "p = 1e-5","xi")
p<- ggplot(data=melt(BFapproxdata,id.vars = "xi"), aes(y=xi))+
  geom_line(aes(x=value, group=variable,color=variable))+
  labs(title = "Distribution of Transformed Probability of Effect\n with Bayes Factor Approximation",
       x = expression(xi), y = "Density")+  theme_minimal()+
  theme(legend.justification=c(1,1), legend.position=c(1,1)) +
  scale_colour_few(name="p-value")
```

In this case, the discovery data will have an effect on the amount of zero-valued global effects sampled because it skews the distribution to the right. However, since in this specific model we do not use the effect estimates from the discovery data, we essentially throw away any information regarding the size of the effect. These can be added in future models to better utilize the discovery data.

3.4 | Prior Specifications

The choice of Cauchy priors for μ and σ^2_β is based on simulation results. Both priors had mean zero and variance 1, based on the usual range of effect sizes in GWAS. The hyperparameters for ξ were set to $a = b = \frac{1}{2}$. This distribution has a U-shape so that it favors extreme probabilities (0 or 1) more heavily than the values between them.



The normal approximation for the conditional likelihood model was chosen for its simplicity and because of the large sample sizes of GWAS, which allow for CLT assumptions.

For the Bayes Factor model, $a = b = \frac{9}{10}$. This is because for small p-values, the transformation of ξ can be very extreme. For a GWAS p-value $p = 10^{-7}$ and $\xi \sim \text{Beta}(.5, .5)$, $P(\xi' \leq 0.5) = 4.74 \times 10^{-11}$. For the flatter prior: $\xi \sim \text{Beta}(.9, .9)$, $P(\xi' \leq 0.5) = 1.24 \times 10^{-6}$.

The Bayes Factor model is extremely sensitive to the choice of prior as well as to the p-value. While the skew is appropriate for this particular prior, it would not necessarily make sense with a flat or informative prior.

3.5 | Methods

Models were fit using R2jags and in the simpler cases, with original Metropolis Hastings algorithms. To specify distributions that are not part of the R2jags library, such as the conditional likelihood, we use the "ones trick", which is implement by creating artificial observations of a Bernoulli variable. Consider a prior for θ that is proportional to $\pi(\theta)$. If we set that Bernoulli variable "ones" is equal to 1 with probability $\pi(\theta)$, create an observation "ones"$= 1$, and set a uniform prior for θ, then we are effectively creating a "posterior" for theta that is proportional to $\pi(\theta)$ as intended.

Each JAGS model was run with the default settings: 3 chains, 2000 iterations, and 1000 burn-in samples. JAGS model functions for the hierarchical simulations can be found in the supplement.

All computed credible intervals are HPD (highest posterior density) intervals. A 95% HPD interval is the 95% of the sampled values with the highest density. HPD intervals are guaranteed to be the shortest intervals for that scale (they are not scale-invariant), and can give more reasonable answers for multimodal distributions than quantile-based intervals because they can be disjoint.

Point estimates were calculated used the posterior median, so that these estimates would be invariant to transformations (e.g. log).

4 | NORMAL SIMULATION

This simulation study addresses on the first goal outlined in the introduction: if the data from one site is found to be significant, how can we report this discovery in a way that takes into account the winner's curse?

4.1 | Data Generation

To test the hypothesis $H_0 : \mu = 0$ versus $H_1 : \mu \neq 0$, a fixed proportion (set at 0.5) of null vs. alternative hypotheses are generated. For each hypothesis H_i, let $\mu_i = 0$ in the null scenario and $\mu_i \sim \mathrm{N}(0, 1)$ in the alternative. The data Y_i is generated from a normal distribution with mean μ_i and known variance 1, with sample size 100. If Y_i is not significant at $\alpha = .05$, it is sampled again from the same distribution until the sufficient statistic is significant. This is done in order to properly compare the Bayesian approach with the conditional likelihood, which requires the data to be significant. The Bayes factor model as it is specified cannot be used in this scenario, since it only uses the p-value from the discovery site(s).

4.2 | Conditional Likelihood

The credible intervals were estimated by treating the conditional likelihood as if it were a posterior distribution with an improper prior $p(\mu) = 1$, and obtaining the HPD region covering 95%. Sampling was done through a Metropolis-Hastings algorithm.

4.3 | Posterior Distribution

```
cond.posterior<- function(Y, n.samp){
  cond.likelihood<- function(Y, mu){
    ybar = mean(Y)
    N = length(Y)
    #(abs(ybar-mu)/(sqrt(1/N))>1.96)*
    dnorm(ybar, mu, sqrt(1/N))/
      (pnorm((-1.96*sqrt(1/N)), mu, sqrt(1/N))
       +1-pnorm((1.96*sqrt(1/N)), mu, sqrt(1/N)))
    #not symmetric other than mu = 0
  }
  #x<- seq(-2,2,.01)
  #plot(x, cond.likelihood(Y, x), type="l")
  #metropolis hasting with flat prior
  c=1
  mu <- 0
  MU <- NULL

  for(s in 1:n.samp){
    mu.star <- rnorm(1, mu, c)
    r = cond.likelihood(Y,mu.star)/cond.likelihood(Y,mu)
    if(runif(1)<r){
      mu <- mu.star
    }
    MU <- c(MU, mu)
  }
  #plot(MU, type="l")
  #ggplot(data = data.frame(MU))+geom_density(aes(x=MU))
  (MU)
}
```

In the Bayesian case, the prior was set to the mixture model $p(\mu|\xi) = (1-\xi)\delta_0(\mu) + \xi\phi(\mu)$. In this case, $\xi = 0.5$ is a constant. Note that this is also the true data generating model.

The marginal posterior distribution is

$$P(\mu|Y) = P(H_0|Y)P(\mu|Y, H_0) + P(H_1|Y)P(\mu|Y, H_1) \tag{18}$$

The separate posteriors for μ are:

$$P(\mu|Y, H_0) = \delta_0(\mu) \tag{19}$$

$$P(\mu|Y, H_1) \sim \mathrm{N}(\frac{n}{n+1}\bar{Y}, \frac{1}{n+1}) \tag{20}$$

The posterior for the alternative hypothesis can be calculated using its Bayes factor, BF and the prior odds, $\pi = \frac{(1-\xi)}{\xi}$:

$$P(H_1|Y) = \frac{\pi BF}{1+\pi BF} \tag{21}$$

For this example, the prior odds are 1 (because the probability of $H_1 = \xi = 0.5$). The Bayes factor is

$$BF = \frac{L(\bar{Y}|H_1)}{L(\bar{Y}|H_0)} = \sqrt{n+1} * e^{\frac{n^2}{2(n+1)}(\bar{Y})^2} \tag{22}$$

This result comes from the fact that the marginal likelihood $L(\bar{Y}|H_1) \sim \mathrm{N}(0, \frac{n}{n+1})$.

Putting these pieces together results in the marginal posterior for μ, which can be used to generate samples to calculate credible intervals.

```
getposterior <- function(Y,n.samp, pi = .5){
  n = length(Y)
  odds = (1-pi)/pi
  #bf.approx <- -exp(1)*p*log(p)
  bf <-  (n+1)^(-.5)*exp(n^2*mean(Y)^2/(2*(n+1)))
  alt.prob <- odds*bf/(1+odds*bf)
  #alt.prob.approx <- odds*bf.approx/(1+odds*bf.approx)

  #draws from posterior-flip a coin (ber w prob P(H given Y) and then use that to get draw
  draws = sapply(runif(n.samp), function(x)  {
    ifelse(x<(1-alt.prob), rnorm(1, 0, 0), rnorm(1,mean(Y)*n/(n+1), sqrt(1/(n+1))))
  })
  #ggplot(data = data.frame(draws))+geom_density(aes(x=draws))
  (list(alt.prob=alt.prob, draws=draws))
}

HPD <-function(post, prob  = .95){

  #HPD interval- copied from BAS/coda
  obj <- as.matrix(post)
  vals <- apply(obj, 2, sort)
  if (!is.matrix(vals))
    stop("obj must have nsamp > 1")
  nsamp <- nrow(vals)
  npar <- ncol(vals)
  gap <- max(1, min(nsamp - 1, round(nsamp * prob)))
  init <- 1:(nsamp - gap)
  inds <- apply(vals[init + gap, , drop = FALSE] - vals[init,
                                                        , drop = FALSE], 2, which.min)
  (cbind(vals[cbind(inds, 1:npar)], vals[cbind(inds +
                                                 gap, 1:npar)]))
  #look into this and check about continuity of cdf, etc
}
```

4.4 | Results

```
all <- function(H, N, n.samp, interval,alpha){
  mu <- ifelse(H==0, 0, rnorm(1, 0, 1)) ### normally distributed mu
  Y <- rnorm(N, mu, 1)
  count=0
  while(abs(abs(mean(Y))/sqrt(1/N)-qnorm(1-alpha/2))>interval){ #if Z is not in (1.94, 1.98)
    if(count>1000){ #had to add this bc it wouldn't run
      return (c(H, mu , mean(Y), NA   ,
                NA  , NA  ,
                NA , NA ,
                NA , NA , NA, NA  , NA   ,
                NA , NA, NA, NA ))
    }

    Y <- rnorm(N, mu, 1)
    count<- count+1
  }
  post <- getposterior(Y,  n.samp)
  alt.prob = post$alt.prob
  cred <- HPD(post$draws)
  cred.lower = cred[1]
  cred.upper = cred[2]
  bayes.cov =  (cred.upper>=mu&&cred.lower<=mu)

  cond <-cond.posterior(Y, n.samp)
  conf <- HPD(cond)
  conf.lower = conf[1]
  conf.upper = conf[2]
  freq.cov =  (conf.upper>=mu&&conf.lower<=mu)
  naive.cov <- mean(Y)+1.96*sqrt(1/N)>=mu&&mean(Y)-1.96*sqrt(1/N)<=mu
  expected.cov <- .95*alt.prob+(conf.upper>=0&&conf.lower<=0)*(1-alt.prob)


  Bayes.est = mean(post$draws)
  Bayes.median.est = median(post$draws)
  Bayes.mode.est = as.numeric(names(sort(-table(post$draws)))[1])
  cond.mean.est = mean(cond)
  d <- density(cond)
  cond.mode.est = d$x[which.max(d$y)]

  (c(H, mu , mean(Y), alt.prob  ,
     cred.lower   , cred.upper   ,
     conf.lower , conf.upper ,
     bayes.cov , freq.cov , naive.cov, expected.cov,
     Bayes.est, Bayes.median.est,
     Bayes.mode.est,cond.mean.est,cond.mode.est))

}


N=100; n.samp = 10000; n.sim=1000; pi = .5

results <-data.frame(t(apply(matrix(as.numeric(runif(n.sim)<pi)),1, function(x) all(x, N, n.samp, .2, .05)
colnames(results)<- c("H", "mu" , "Ybar", "alt.prob" ,
        "cred.lower"  , "cred.upper"   ,
        "conf.lower" , "conf.upper" ,
        "bayes.cov" , "freq.cov" ,  "naive.cov", "expected.cov",
        "Bayes.est","Bayes.median.est","Bayes.mode.est",
        "cond.mean.est","cond.mode.est")
save(results, file="normalsim.RData")
```

4.4.1 | Estimators

```
load("normalsim.RData")
results<-na.omit(results)
estimators <- data.frame(Bayes.mean = abs(results$Bayes.est-results$mu),
              naive = abs(results$Ybar-results$mu),
              Bayes.median = abs(results$Bayes.median.est-results$mu),
              Bayes.mode = abs(results$Bayes.mode.est-results$mu),
              cond.mean = abs(results$cond.mean.est-results$mu),
              cond.mode = abs(results$cond.mode.est-results$mu))

p<- ggplot(data = melt(estimators), aes(x=variable, y=value)) + geom_boxplot()+labs(x= "Estimator", y="|Bi

kable(t(c(Bayes.mean = sqrt(sum(estimators$Bayes.mean^2)), Bayes.median= sqrt(sum(estimators$Bayes.median^
```

TABLE 1 RMSE of 100 simulations

Bayes.mean	Bayes.median	Bayes.mode	cond.mean	cond.mode	naive
3.799	4.318	4.318	3.913	3.877	4.982

The conditional likelihood mode (i.e. MLE) has the smallest bias (absolute error) for μ out of the frequentist estimators, while the Bayesian median and mode (which end up being the same) the smallest bias in the Bayesian framework. The RMSE for the Bayesian estimator (mean of the posterior) is the lowest, followed by the conditional mean and mode.

4.4.2 | Credible Intervals

```
p<-ggplot(data = results)+ geom_point(aes(x = conf.upper-conf.lower, y = cred.upper-cred.lower, colour = a
```

The lines mark the $y = x$ line, and the length of naive confidence intervals (which are constant for fixed number of samples) on the x and y axes.

The largest values for the significant statistic also correspond to the largest intervals in both cases. Note that the conditional likelihood credible intervals are almost always larger than the fully Bayesian credible intervals, but still mostly smaller than the naive ones.

4.4.3 | Coverage

The marginal coverage of the conditional likelihood credible interval C is

$$P(\mu \in C|Y) = P(\mu \in C|H_0)P(H_0|Y) + P(\mu \in C|H_1)P(H_1|Y) \tag{23}$$

This will be significantly higher than .95 for the cases in which $0 \in C$, since $P(\mu \in C|H_1) = 0.95$ by definition, and $P(\mu \in C|H_0) = I_{0\in C}$. In this experiment, the expected coverage is 0.98 for intervals with 0, and only 0.38 for those that do not contain 0.

However, conditioning on the alternative hypothesis does not lead to an empirical coverage of 95%.

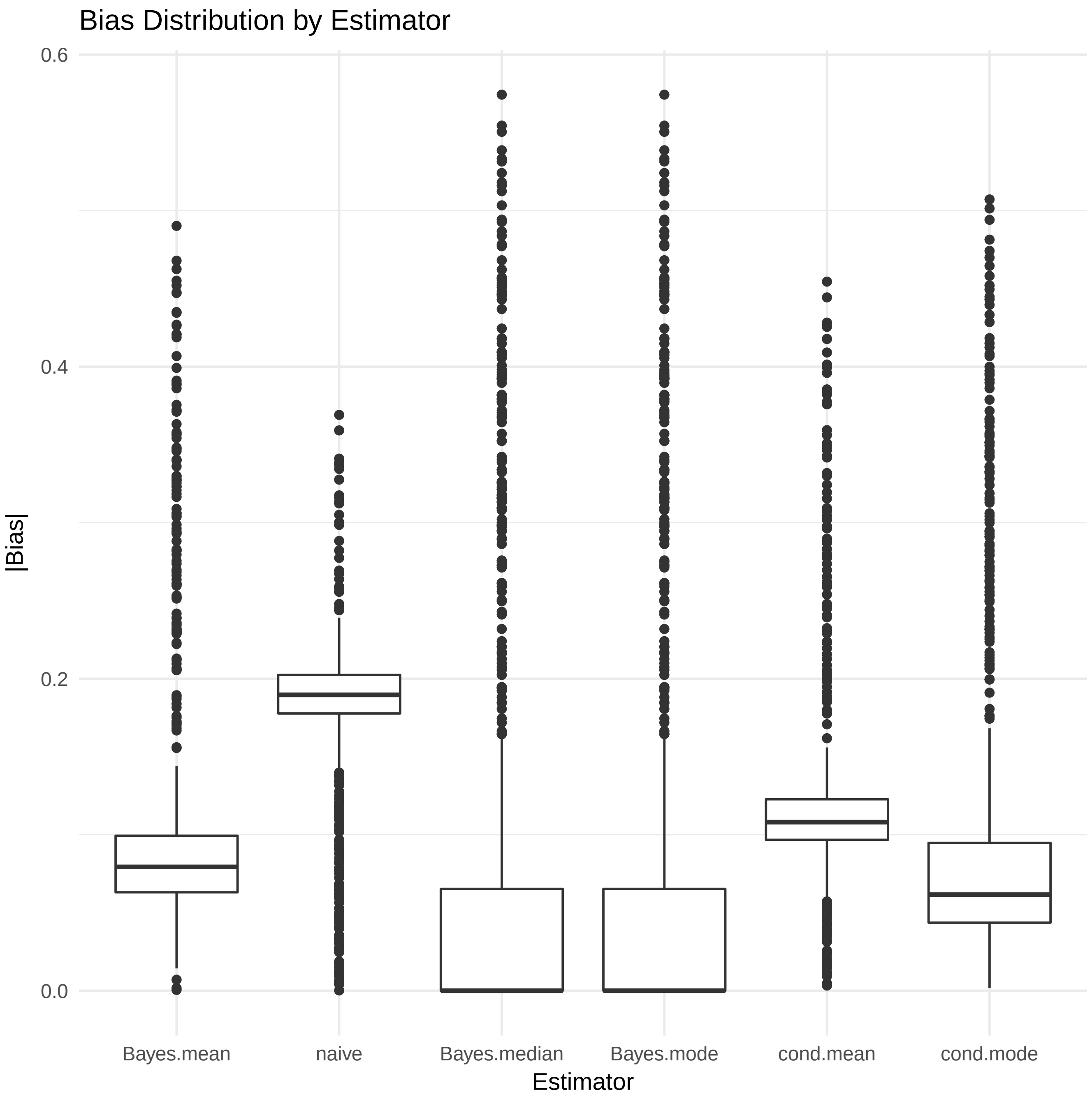
We can see that both methods are still significantly better than the naive one.

```
results <- na.omit(results)
b <- which(results$H==0)
#
# kable(t(data.frame(naive = c(mean(results$naive.cov),mean(results[b,]$naive.cov),mean(results[-b,]$naiv
#         conditional = c(mean(results$freq.cov), mean(results[b,]$freq.cov), mean(results[-b,]$freq.cov),
#         Bayesian = c(mean(results$bayes.cov), mean(results[b,]$bayes.cov), mean(results[-b,]$bayes.cov),
#       col.names = c("Unconditional Coverage","Coverage Conditional on H0", "Coverage Conditional on H1",
#       row.names = TRUE,
#       caption = "Empirical Coverage for 95% Credible Intervals", digits=3)
```

4.4.4 | Hypothesis Testing

Due to the nature of p-values, an $\alpha = 0.05$ corresponds to a marginal posterior probability $P(H_1|Y)$ of only 0.4 for $N = 100$. This means that the 95% credible interval for $\mu|Y$ will contain 0 every time. In terms of hypothesis testing, if we consider the strategy of rejecting the null when the interval does not contain 0, this level for α leads to no rejections.

```
kable(table(results$H,abs(results$Ybar)/sqrt(1/100)<qnorm(1-0.05/2))/dim(results)[1], col.names = c("Do no
```



```
kable(table(results$H,results$conf.upper>=0&results$conf.lower<=0)/dim(results)[1], col.names = c("Do not
```

```
kable(table(results$H,results$cred.upper>=0&results$cred.lower<=0)/dim(results)[1], col.names = c("Do not
```

Despite never rejecting the null, the conditional likelihood and the Bayesian methods both perform better than the naive one in terms of "predicting" accurately. The naive method is especially problematic in that it has a higher Type 1 error (false positives) than true positives OR true negatives in the region of the data.

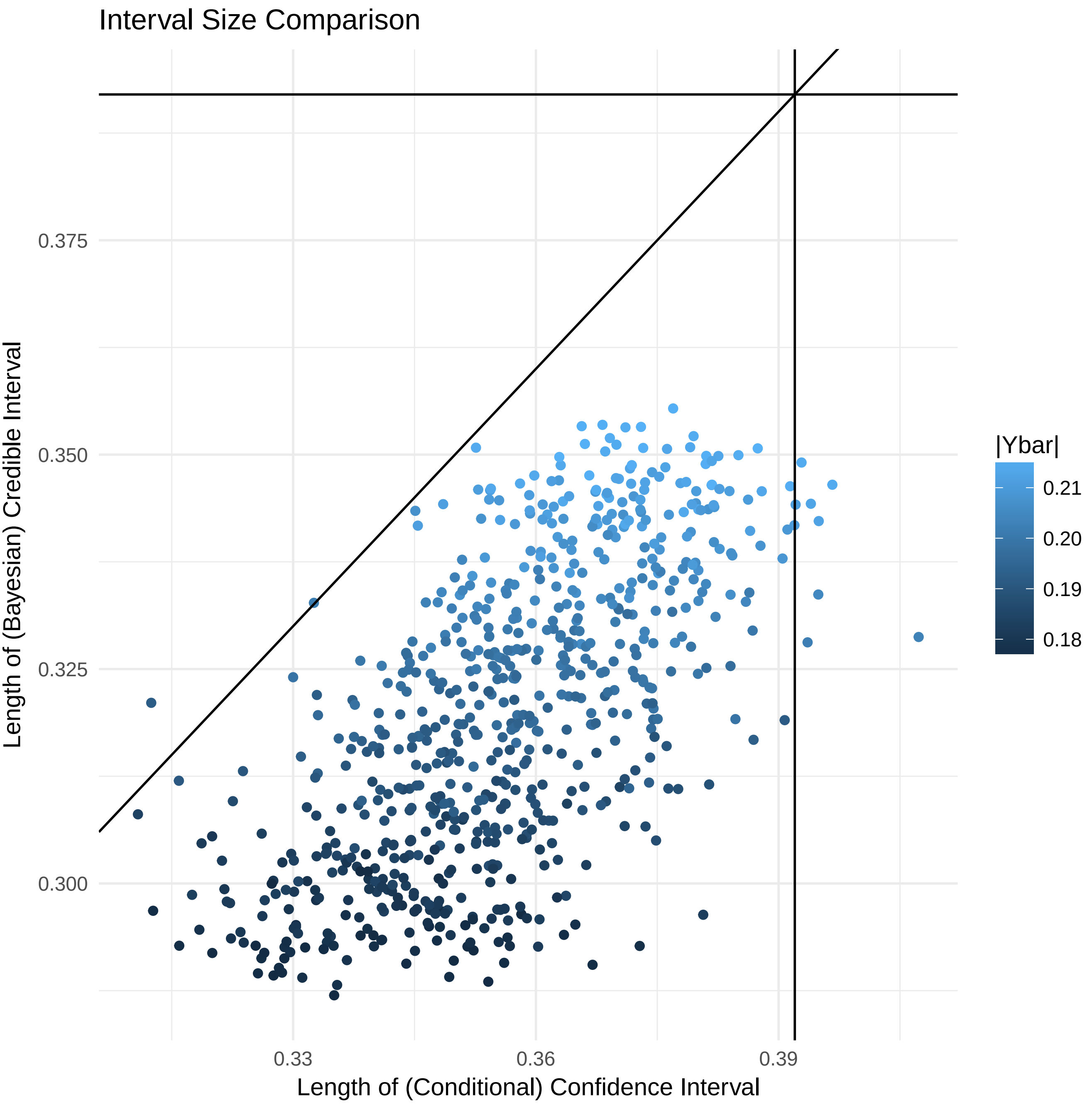


TABLE 2 Naive method

	Do not reject null	Reject null
0	0.317	0.400
1	0.150	0.133

TABLE 3 Conditional Likelihood Method

	Do not reject null
0	0.717
1	0.283

TABLE 4 Bayesian Mixture Model

	Do not reject null
0	0.717
1	0.283

5 | HIERARCHICAL SIMULATIONS

```r
knitr::opts_chunk$set(eval=FALSE)
```

```r
load("p53sim_processeddata.RData")
mu=0.203;sd= 0.05831085 #from glmer estimates using all data and one snp-------redo this
assoc<-c(0,1,1,1,1)
muvec<-c(0,mu,mu,mu,mu)
sdvec<-c(1,.5*sd,sd,2*sd,4*sd)
I = 100
pvals=c(.05,.01,.005,.001, 1e-7)
```

This simulation study aims to deal with the second goal specified in the introduction. After an effect has been discovered, how can data from replication studies be combined with the original? In this scenario, we must account for the heterogeneity between sites; neglecting the uncertainty that comes from replication studies would lead to erroneously confident estimates of significance and effect size.

```r
get.cond.likelihood.data<- function(data, p = 0.00325){
  freq = glm(CaseCon ~ factor(site), data=data,family=binomial, x=T)
  #get "discovery" with smallest p value (that is significant)
  coefs = coef(summary(freq))
  discovery.site = which.min(coefs[,4])
  if(coefs[discovery.site,4]>p){
    return(NULL)
  }

  MLE = coefs[discovery.site,1]
  SE = coefs[discovery.site,2]
  p.value = coefs[discovery.site,4]
  #make new data for model
  exclude <- which(data$site==discovery.site)
  newdata = list(MLE=MLE, SE=SE, n.discovery= 1, zeroes= 0,
                 discovery.sites=discovery.site,
                 CaseCon= data$CaseCon[-exclude],
                 site= data$site[-exclude], n.sites = data$n.sites,
                 q=qnorm(1-p/2),
                 p = coefs[discovery.site,4])
  newdata$J<- length(newdata$CaseCon)
  return(newdata)

}
```

```
sample<- function(assoc, mu, sd, n.sites=7,observations = 1000){
  #assoc is H
  beta.p53 = rnorm(n.sites,mu,sd)*assoc
  Y <-site <- rep(NA, observations*n.sites)
  for(i in 1:n.sites){
    Y[((i-1)*observations+1):(i*observations)]<-rbinom(observations, 1, exp(beta.p53[i])/(1+exp(beta.p53[i
    site[((i-1)*observations+1):(i*observations)]<- rep(i, observations)
  }
  return(list(beta.p53=beta.p53, simdata = list(CaseCon=Y, site=site,  J=n.sites*observations ,n.sites=n.s
}

run.all<- function(assoc,mu, sd, inits,n.sites){
  cond.data = NULL;count=0
  while(is.null(cond.data) & count<1000){
    data <- sample(assoc, mu, sd,n.sites)
    cond.data<-get.cond.likelihood.data(data$simdata)
    count=count+1
  }
  #run jags
  latent.cauchy <- jags(data=data$simdata, inits=inits,
                        parameters.to.save =c("pind", "mu.p53", "sigma.p53","mu.p53.notzero"),
                        model = latent.cauchy.model)
  cond.likelihood<-NULL
  if(!is.null(cond.data)){
    cond.likelihood<- jags(data=cond.data ,inits=inits,
                           parameters.to.save =c("mu.p53", "sigma.p53"),
                           model = cond.likelihood.re.model)
    bf.approx<- jags(data=cond.data, inits=inits,
                     parameters.to.save =c("pind","mu.p53", "sigma.p53","mu.p53.notzero"),
                     model = bf.model)
  }
  original<- jags(data=data$simdata, inits=inits,
                  parameters.to.save =c("mu.p53", "sigma.p53"),
                  model = original.model)

  return(list(latent.cauchy = latent.cauchy,
              cond.likelihood = cond.likelihood,
              bf=bf.approx,
              original= original,
              beta.p53 = data$beta.p53))
}

run.cl.better<- function(assoc,mu, sd, inits, pvals){ #slower but will use the same data so hopefully bet
  cond.data = NULL;count=0
  while(is.null(cond.data) & count<1000){
    data <- sample(assoc, mu, sd)
    cond.data<-get.cond.likelihood.data(data$simdata, min(pvals))
    count=count+1
  }
  if(!is.null(cond.data)){
    cond.likelihood<- lapply(pvals, function(p){
      cond.data$q<-qnorm(1-p/2)
      return(jags(data=cond.data ,inits=inits,
                  parameters.to.save =c("mu.p53", "sigma.p53"),
                  #rerun also with random effect?
                  model = cond.likelihood.re.model))
    })
  }
  return(cond.likelihood)
}
```

5.1 | Data Generation

The data are generated from a hierarchical (i.e. mixed effect) logistic model as discribed in the models section: if truly associated, μ and σ^2 have (fixed) nonzero values;$\beta_j \sim \text{N}(\mu, \sigma^2)$. Otherwise, $\mu = \beta_j = 0, \forall j$.

To try to keep this simulation as close to the real data as possible, a preliminary logistic regression with random slope and random p53 coefficient by site was run. This led to the values of $\mu = -0.2, -0.19, -0.18, -0.17, -0.16, -0.15, -0.14, -0.13, -0.12, -0.11, -0.1, -0.09, -0.08, -0.07, -0.06, -0.05, -0.04, -0.03, -0.02, -0.0$ The value of μ remained fixed through all the simulations, but different values of σ were used to test the sensitivity of the models: σ, $\sigma/2$, 2σ, and 4σ. The number of sites was set to 7, since results using 30 sites were almost identical. Each site had 1000 observations. A total of simulated datasets was created each time.

datasets were also simulated under the null hypothesis. They were fit with the models described previously.

5.1.1 | Finding "Discovery" Sites

In this simulation study, a logistic regression with fixed effects for sites was conducted to find the site with the smallest p-value less than α. If no sites matched this description, the data was resampled until at least one site was viable. The maximum likelihood estimate of this effect and its variance were added as data for the conditional likelihood model, and the p-value was added to the bayes factor approximation model. The observations for this site were then taken out of the data.

5.2 | Results

```
cl = makeCluster(8)
registerDoParallel(cl)

system.time({
  allsim<- foreach(j = 1:5) %:%
    foreach ( i = 1:I, .inorder=FALSE, .packages = c("R2jags"))  %dopar% {
      return(run.all(assoc[j],muvec[j],sdvec[j], NULL, 7))
    }
})
save.image()


cl = makeCluster(8)
registerDoParallel(cl)
system.time({
  cl.sim.better.1<-
    foreach ( i = 1:I, .inorder=FALSE, .packages = c("R2jags"))  %dopar% {
      return(run.cl.better(1,mu,sd, NULL,pvals))
    }
})
save.image()

cl = makeCluster(8)
registerDoParallel(cl)
system.time({
  cl.sim.better.4<-
    foreach ( i = 1:I, .inorder=FALSE, .packages = c("R2jags"))  %dopar% {
      return(run.cl.better(1,mu,sd*4, NULL,pvals))
    }
})
save.image()

cl = makeCluster(8)
registerDoParallel(cl)
system.time({
  cl.sim.better.0<-
    foreach ( i = 1:I, .inorder=FALSE, .packages = c("R2jags"))  %dopar% {
      return(run.cl.better(0,0,sd, NULL,pvals[1:4]))
    }
})
save.image()
```

```
getstatsitems<-c("hasmu","haszero","mean","median",
                 "ismultimodal", "intervallength","probneqzero",
                 "hassd","sdmean","sdmedian","sdintlen",
                 "haspind","pindmean","pindmedian","pindg.5")

modelnames<- c("Bayesian",
               "Cond Likelihood","BF Approx", "Original")

getstatsloop<-function(simlist,mu,sd,modelnames){
  I<-length(simlist)
  vec<-sapply(simlist, function(outputlist) sapply(outputlist[-length(outputlist)],getstats,mu,sd))
  return(array(data=vec,dim= c(length(getstatsitems),length(modelnames),I), dimnames=list(getstatsitems,mo
}

getstats<-function(jagsoutput, mu, sd){
  mu.samples<- jagsoutput$BUGSoutput$sims.list$mu.p53
  if(!is.null(jagsoutput$BUGSoutput$sims.list$mu.p53.notzero)){
    mu.samples<-mu.samples*jagsoutput$BUGSoutput$sims.list$mu.p53.notzero
  }
  cred<-HPDM(mu.samples)
  upper<- cred[2]
  lower<- cred[1]
  if(length(cred)>2){
    #assume 2 modes max
    upper <- c(upper,cred[4])
    lower <- c(lower,cred[3])
  }
  sd.samples<-sd.cred<-NA
  if(!is.null(jagsoutput$BUGSoutput$sims.list$sigma.p53)){
    sd.samples <- (jagsoutput$BUGSoutput$sims.list$sigma.p53)
    sd.cred<- HPDM(sd.samples)
  }

  pind.samples<-pind.cred<-NA
  if(!is.null(jagsoutput$BUGSoutput$sims.list$pind)){
    pind.samples<- jagsoutput$BUGSoutput$sims.list$pind
    # pind.cred<- HPDM(pind.samples)
  }
  return(c(hasmu = any(upper>=mu&lower<=mu),
           haszero = any(upper>=0&lower<=0),
           mean = mean(mu.samples),
           median = median(mu.samples),
           ismultimodal=length(cred)>2,
           intervallength = sum(upper-lower),
           probneqzero = sum(mu.samples!=0)/length(mu.samples),
           hassd= sd.cred[2]>=sd&sd.cred[1]<=sd,
           sdmean= mean(sd.samples),
           sdmedian= median(sd.samples),
           sdintlen=sd.cred[2]-sd.cred[1],
           haspind=pind.cred[2]>=0&pind.cred[1]<=1,
           pindmean =mean(pind.samples),
           pindmedian=median(pind.samples),
           # pindintlen=pind.cred[2]-pind.cred[1]
           pindgreater.5=mean(pind.samples>.5)

  ))
}
```

```
sim.0<-getstatsloop(allsim[[1]],0, 1,modelnames)
sim.half<-getstatsloop(allsim[[2]],mu, sd*.5,modelnames)
sim.1<-getstatsloop(allsim[[3]],mu, sd, modelnames)
sim.2<-getstatsloop(allsim[[4]],mu,sd*2,modelnames)
sim.4<-getstatsloop(allsim[[5]],mu,sd*4,modelnames)

sim.cl.stats.better1<-getstatsloop(cl.sim.better.1,mu,sd, pvals)
sim.cl.stats.better4<-getstatsloop(cl.sim.better.4,mu,sd, pvals)
sim.cl.stats.better0<-getstatsloop(cl.sim.better.0 ,0,1, pvals)

save(sim.0, sim.half,sim.1,sim.2,sim.4, sim.cl.stats.better1,sim.cl.stats.better4,sim.cl.stats.better0, fi
```

5.2.1 | RMSE of μ

As expected, the models performed more poorly as σ increased. Out of the three proposed models (compared with the original), the fully bayesian and BF approximation models performed best when there was no true effect (since they were the only ones that that had this option). However, there were some simulated datasets where the bayes factor model estimate was actually nonzero and quite large, which suggests that it is not nearly as reliable as the bayesian model.

At small variances (s/2, s, 2s), the original and bayesian models outperform the others. This is not surprising since the other models only have access to $\frac{6}{7}$ of the data. The Bayesian model actually has a slightly higher lower RMSE than the other models when there is a true association.

```
df1<- data.frame(melt(t(abs(sim.0[4,,])))[,2:3], simulation=0)
simslist<-list(sim.half,sim.1,sim.2,sim.4)
for (i in 1:4){
  df1<-rbind(df1,data.frame(melt(t(abs(simslist[[i]][4,,]-mu)))[,2:3], simulation=i))
}
mseplot = ggplot(data = (df1), aes(x=factor(simulation, labels=c(0,"s/2", "s", "2s", "4s")), y=value)) +
  geom_boxplot(aes(color=Var2))+  theme_minimal() +
  labs(x=expression(paste(sigma^2, " used in simulation")),y="Absolute Error", title=expression(paste("Abs
  scale_colour_few(name="Model") +  theme( legend.justification=c(0,1), legend.position=c(0,1))


rmse<- cbind(sapply(1:4, function(x) sqrt(mean((sim.0[4,x,])^2))),
             sapply(simslist, function(sim)
               sapply(1:4, function(x) sqrt(mean((sim[4,x,]-mu)^2)))))
rownames(rmse)<-modelnames
colnames(rmse)<-c("mu=0","s/2", "s", "2s", "4s")


pindmed<- cbind(sapply(c(1,3), function(x) sqrt(mean((sim.0[14,x,])^2))),
                sapply(simslist, function(sim)
                  sapply(c(1,3), function(x) sqrt(mean((sim[14,x,])^2)))))
rownames(pindmed)<-modelnames[c(1,3)]
colnames(pindmed)<-c("mu=0","s/2", "s", "2s", "4s")

zeromu<- cbind(sapply(c(1,3), function(x) sqrt(mean((sim.0[7,x,])^2))),
               sapply(simslist, function(sim)
                 sapply(c(1,3), function(x) sqrt(mean((sim[7,x,])^2)))))
rownames(zeromu)<-modelnames[c(1,3)]
colnames(zeromu)<-c("mu=0","s/2", "s", "2s", "4s")


df<- data.frame(melt(t(abs(sim.0[6,,])))[,2:3], simulation=0)
simslist<-list(sim.half,sim.1,sim.2,sim.4)
for (i in 1:4){
  df<-rbind(df,data.frame(melt(t(abs(simslist[[i]][6,,])))[,2:3], simulation=i))
}

intlenplot = ggplot(data = (df), aes(x=factor(simulation, labels=c(0,"s/2", "s", "2s", "4s")), y=value)) -
  geom_boxplot(aes(color=Var2))+  theme_minimal() +
  labs(x=expression(paste(sigma^2, " used in simulation")),y="Credible Interval Length", title=expression(
  scale_colour_few(name="Model") +  theme( legend.justification=c(0,1), legend.position=c(0,1))




hasm<-  cbind(sapply(1:4, function(x) mean(sim.0[1,x,])),
              sapply(simslist, function(sim)
  sapply(1:4, function(x) mean(sim[1,x,]))))
rownames(hasm)<-modelnames
colnames(hasm)<-c("mu=0","s/2", "s", "2s", "4s")

has0<-  cbind(sapply(1:4, function(x) mean(sim.0[1,x,])),
              sapply(simslist, function(sim)
  sapply(1:4, function(x) mean(sim[2,x,]))))
rownames(has0)<-modelnames
colnames(has0)<-c("mu=0","s/2", "s", "2s", "4s")
```

```
kable(rmse, caption="RMSE of $\\mu$",digits=3)
```

5.2.2 | Coverage of μ

The conditional likelihood and BF approximation models are the most conservative, with the intervals covering 0 more times than the others for large values of σ. All models have very high coverage in general.

The Bayesian model had the shortest intervals, and the original model had the largest. Thus, even though the coverage and RMSE are around the same, the new models are preferable to the original. This does not apply to the simulation with $\sigma = 4s$, because $4s > \mu$, which leads to more negative site effects. Thus, it makes sense for models to have wider credible intervals for these simulations.

```
kable(hasm, ,format='latex',caption="Proportion of Credible Intervals containing $\\mu$", digits=3)
```

```
kable(has0, ,format='latex',caption="Proportion of Credible Intervals containing 0", digits=3)
```

5.2.3 | Probability of Association ξ

While one would expect the probability of being associated (ξ) to also increase with σ, this was not true for either the fully bayesian model nor the bayes factor approximation one, both of which had consistent posterior estimates of ξ. Similarly, the proportion of nonzero μ samples from the posterior (this is the same as the proportion of times the latent variable $i = 1$), was almost 1 for the truly associated cases, and close to zero for true null. One thing to consider is that under the null hypothesis, the variance across sites would actually be zero, which is why the models identified the association so decisively.

For the simulations that had no true effect, the Bayes Factor approximation model has much larger median ξ and greater proportion of sampled $H_1$ because the mass of the distribution of ξ is shifted towards 1 by the Bayes Factor transformation. Thus, even though the proportion of $H_1$ is quite low (and the median of μ is 0), the mean of ξ is greater than 0.5.

```
kable(pindmed,caption="Average of Posterior Median $\\xi$", digits=3)
```

```
kable(zeromu,caption="Average Proportion of $H_1$", digits=3)
```

5.2.4 | Sensitivity of Conditional Likelihood Method to Changes in α

The conditional likelihood method with random effects is robust to changes in the level α. To test this, we consider 5 different levels: $0.05, 0.01, 0.005, 0.001, 10^{-7}$. datasets were sampled, for which at least one location was significant at the smallest α level. The conditional likelihood model with random effects and without was then fitted for each level.

```
df<- rbind(data.frame(melt(t(abs(sim.cl.stats.better0[4,,])))[,2:3], simulation=0),
           data.frame(melt(t(abs(sim.cl.stats.better1[4,,]-mu)))[,2:3], simulation=1),
           data.frame(melt(t(abs(sim.cl.stats.better4[4,,]-mu)))[,2:3], simulation=2))

mseplot = ggplot(data = (df), aes(x=factor(simulation, labels=c( "0", "s", "4s")), y=value)) +
  geom_boxplot(aes(color=factor(Var2)))+  theme_minimal() +
  labs(x=expression(paste(sigma^2, " used in simulation")),y="Absolute Error", title=expression(paste("Abs
  scale_colour_few(name="p-value") +  theme( legend.justification=c(0,1), legend.position=c(0,1))


rmse<- cbind(c(sapply(1:4, function(x) sqrt(mean((sim.cl.stats.better0[4,x,])^2))), NA),
             sapply(1:5, function(x) sqrt(mean((sim.cl.stats.better1[4,x,]-mu)^2))),
              sapply(1:5, function(x) sqrt(mean((sim.cl.stats.better4[4,x,]-mu)^2))))
rownames(rmse)<-pvals

colnames(rmse)<-c("0","s", "4s")



df<- rbind(data.frame(melt(t(abs(sim.cl.stats.better0[6,,])))[,2:3], simulation=0),
           data.frame(melt(t(abs(sim.cl.stats.better1[6,,])))[,2:3], simulation=1),
           data.frame(melt(t(abs(sim.cl.stats.better4[6,,])))[,2:3], simulation=2))

intlenplot = ggplot(data = (df), aes(x=factor(simulation, labels=c("0","s","4s")), y=value)) +
  geom_boxplot(aes(color=factor(Var2)))+  theme_minimal() +
  labs(x=expression(paste(sigma^2, " used in simulation")),y="Credible Interval Length", title=expression(
  scale_colour_few(name="p-value") +  theme( legend.justification=c(0,1), legend.position=c(0,1))

kable(rmse,caption="RMSE of $\\mu$", digits=3)
```

This model shows little difference across levels of α.

6 | ANALYSIS OF TP53

6.1 | Models

Each model adjusts for study site, reference age, and personal history of breast cancer. History of breast cancer is treated as a fixed effect, and the rest of the covariates's coefficients are treated as normally distributed random effects.

$$P(Y_{ij} = 1|\beta_j^{site}, \beta_j^{p53}, \beta_j^{age}, \beta^{BC}) = \text{logit}^{-1}(\beta_j^{site} + \beta_j^{p53} * p53_{ij} + \beta_j^{age} * age_{ij} + \beta^{BC} * BC_{ij}) \tag{24}$$

$$\beta_j^{site}|\mu_{site}, \sigma_{site}^2 \sim \text{N}(\mu_{site}, \sigma_{site}^2) \tag{25}$$

$$\beta_j^{age}|\mu_{age}, \sigma_{age}^2 \sim \text{N}(\mu_{age}, \sigma_{age}^2) \tag{26}$$

$$\beta^{\text{BC}}, \mu_{age}, \mu_{site} \sim \text{N}(0, 0.1) \tag{27}$$

$$\sigma_{age}, \sigma_{site} \sim \text{invGamma}(1, 0.05) \tag{28}$$

The p53 variable's site-specific log OR priors were defined using the models described previously.

The fully Bayesian model was fit jointly as well as marginally. Since the results were very similar, the marginal models were used for computational efficiency and clarity of interpretation. Results from the joint analysis can be found in the supplement.

6.2 | Results

The new models shrunk estimates towards 0 for all SNPs. The BF approximation model set all point estimates to zero, as did the fully Bayesian model with the exception of one. This is the same SNP () that was detected as significant in the MISA analysis[4]. However, the credible interval contains 0, so there is not enough evidence for association. Similarly, although the conditional likelihood model does not set any estimates to zero, all its credible intervals contain zero.

The original model has the longest credible intervals for all SNPs except , for which both the Bayesian and Bayes Factor models have large credible intervals, indicating more uncertainty than the original model.

These results are consistent with those from recent GWAS[5], which found no association between any of the TP53 SNPs and cancer.

Below are the full tables of OR by site and SNP.

7 | CONCLUSION

We expand on three ideas from current literature to deal with the winner's curse by reducing the selection effect in initial studies that test for significance, as well as in replicated ones that aim to validate previous discoveries. The fully Bayesian model uses all the data to make inference and uses a binary latent variable to model the true association. This is equivalent to a spike and slab prior, or to model averaging with two possible models: the null and the alternative. The conditional likelihood method uses the likelihood of the estimate conditional on being significant. This is a frequentist approach to selection bias, and it depends on the significance test level as well as the effect estimate and standard error. The conditional likelihood is then used as a prior in the (Bayesian) validation analysis. The Bayes factor approximation method uses an upper bound on the Bayes factor that is only dependent on the p-value to calculate a "best-case scenario" posterior probability of the alternative hypothesis. The distribution that arises from this transformation is used as the prior of the association probability in the validation analysis. This approach also has a frequentist component, since p-values are used, but is a step towards the Bayesian framework since the only function of the p-value is to approximate the Bayes Factor. All models improve upon naive methods in the discovery phase, as shown in the normal simulation study, as well as the validation phase, as shown in the hierarchical simulation study and analysis of p53 data.

One clear advantage of the fully Bayesian model is that it can perform testing and estimation simultaneously. This means all the data is used once, which is why the credible intervals are smaller and the RMSE is lower in the simulations. However, it is not always feasible to implement if the discovery data is unavailable. Furthermore, Bayesian methods are not enough to not guarantee bias correction. They must take into account the selection mechanism (as is done here), and also report the uncertainty that is associated with this. Ignoring the selection effect or reporting only the selected interval can lead to paradoxes, especially for conjugate priors in multivariate inference, as was the case with the original model for the p53 analysis[17].

The conditional likelihood and Bayes factor approximation methods can be used in follow-up studies even when the discovery data is not publicly available. Both provide significant improvements over the naive method. The Bayes factor model has a quasi-testing feature since it accounts for the probability of a true association, but is very sensitive to the p-value as well as the choice of prior, and can be illogical for a prior such as the uniform.

Although the conditional likelihood itself is dependent on the significance test, this prior is only used for the discovery sites, and can be thought of as a posterior distribution under a flat prior. Under a hierarchical model, the global effect is actually unaffected by the α level. This is extremely useful because discoveries that do not have are not significant at the 10^{-7} level can still be used without affecting the results. However, the level α is crucial in the discovery phase. The integration over all significant events is simple to compute in this case, but might not be as simple for other distributions. For example, if one chooses to do Bayesian variable selection the conditional likelihood becomes intractable and must be approximated[18]. In this case, an adaptation of the fully Bayesian model may actually be more computationally feasible.

References

1. Benjamin Daniel J, Berger James O, Johannesson Magnus, et al. Redefine statistical significance. *Nature Human Behaviour.* 2017;.

2. Zöllner Sebastian, Pritchard Jonathan K. Overcoming the winnerâĂŹs curse: estimating penetrance parameters from case-control data. *The American Journal of Human Genetics.* 2007;80(4):605–615.

3. Schildkraut Joellen M., Goode Ellen L., Clyde Merlise A., et al. Single Nucleotide Polymorphisms in the TP53 Region and Susceptibility to Invasive Epithelial Ovarian Cancer. *Cancer Research.* 2009;69(6):2349–2357.

4. Schildkraut Joellen M, Iversen Edwin S, Wilson Melanie A, et al. Association between DNA damage response and repair genes and risk of invasive serous ovarian cancer. *PLoS One.* 2010;5(4):e10061.

5. Phelan Catherine M, Kuchenbaecker Karoline B, Tyrer Jonathan P, et al. Identification of 12 new susceptibility loci for different histotypes of epithelial ovarian cancer. *Nature genetics.* 2017;49(5):680.

6. Sellke Thomas, Bayarri MJ, Berger James O. Calibration of ρ values for testing precise null hypotheses. *The American Statistician.* 2001;55(1):62–71.

7. Xu Shizhong. Theoretical basis of the Beavis effect. *Genetics.* 2003;165(4):2259–2268.

8. Zhong Hua, Prentice Ross L. Bias-reduced estimators and confidence intervals for odds ratios in genome-wide association studies. *Biostatistics.* 2008;9(4):621–634.

9. Ghosh Arpita, Zou Fei, Wright Fred A. Estimating odds ratios in genome scans: an approximate conditional likelihood approach. *The American Journal of Human Genetics.* 2008;82(5):1064–1074.

10. Sun Lei, Dimitromanolakis Apostolos, Faye Laura L, et al. BR-squared: a practical solution to the winnerâĂŹs curse in genome-wide scans. *Human genetics.* 2011;129(5):545–552.

11. Bacanu Silviu-Alin, Kendler Kenneth S. Extracting actionable information from genome scans. *Genetic epidemiology.* 2013;37(1):48–59.

12. Bigdeli T Bernard, Lee Donghyung, Webb Bradley Todd, et al. A simple yet accurate correction for winner's curse can predict signals discovered in much larger genome scans. *Bioinformatics.* 2016;32(17):2598–2603.

13. Storey John D, Tibshirani Robert. Statistical significance for genomewide studies. *Proceedings of the National Academy of Sciences.* 2003;100(16):9440–9445.

14. Xu Lizhen, Craiu Radu V, Sun Lei. Bayesian methods to overcome the winner's curse in genetic studies. *The Annals of Applied Statistics.* 2011;5:201–231.

15. Ferguson John P, Cho Judy H, Yang Can, Zhao Hongyu. Empirical Bayes correction for the Winner's Curse in genetic association studies. *Genetic epidemiology.* 2013;37(1):60–68.

16. Jiang Wei, Yu Weichuan. Power estimation and sample size determination for replication studies of genome-wide association studies. *BMC genomics.* 2016;17(1):19.

17. Dawid AP. Selection paradoxes of Bayesian inference. *Lecture Notes-Monograph Series.* 1994;:211–220.

18. Panigrahi Snigdha, Taylor Jonathan, Weinstein Asaf. Bayesian post-selection inference in the linear model. *arXiv preprint arXiv:1605.08824.* 2016;.

19. Yekutieli Daniel. Adjusted Bayesian inference for selected parameters. *Journal of the Royal Statistical Society: Series B (Statistical Methodology).* 2012;74(3):515–541.

AUTHOR BIOGRAPHY

How to cite this article: Williams K., B. Hoskins, R. Lee, G. Masato, and T. Woollings (2016), A regime analysis of Atlantic winter jet variability applied to evaluate HadGEM3-GC2, *Q.J.R. Meteorol. Soc.*, *2017;00:1–6*.