Open Peer Review

Invited Reviewers

1 Matthew Stephens, University of Chicago, Chicago, USA

Any reports and responses or comments on the article can be found at the end of the article.

METHOD ARTICLE

Fast effect size shrinkage software for beta-binomial models of allelic imbalance [version 1; peer review: 1 approved with reservations]

Joshua P. Zitovsky\textsuperscript{1}, Michael I. Love\textsuperscript{1,2}

\textsuperscript{1}Department of Biostatistics, University of North Carolina at Chapel Hill, Chapel Hill, NC, 27516, USA
\textsuperscript{2}Department of Genetics, University of North Carolina at Chapel Hill, Chapel Hill, NC, 27514, USA

Abstract

Allelic imbalance occurs when the two alleles of a gene are differentially expressed within a diploid organism, and can indicate important differences in cis-regulation and epigenetic state across the two chromosomes. Because of this, the ability to accurately quantify the proportion at which each allele of a gene is expressed is of great interest to researchers. This becomes challenging in the presence of small read counts and/or sample sizes, which can cause estimates for allelic expression proportions to have high variance. Investigators have traditionally dealt with this problem by filtering out genes with small counts and samples. However, this may inadvertently remove important genes that have truly large allelic imbalances. Another option is to use Bayesian estimators to reduce the variance. To this end, we evaluated the accuracy of three different estimators, the latter two of which are Bayesian shrinkage estimators: maximum likelihood, approximate posterior estimation of GLM coefficients (apeglm) and adaptive shrinkage (ash). We also wrote C++ code to quickly calculate ML and apeglm estimates, and integrated it into the apeglm package. The three methods were evaluated on both simulated and real data. Apeglm consistently performed better than ML according to a variety of criteria, including mean absolute error and concordance at the top. While ash had lower error and greater concordance than ML on the simulations, it also had a tendency to over-shrink large effects, and performed worse on the real data according to error and concordance. Furthermore, when compared to five other packages that also fit beta-binomial models, the apeglm package was substantially faster, making our package useful for quick and reliable analyses of allelic imbalance. Apeglm is available as an R/Bioconductor package at http://bioconductor.org/packages/apeglm.

Keywords

RNA-seq, Allelic imbalance, Allele-specific expression (ASE), Beta-binomial, Shrinkage estimation, Empirical Bayes, Bioconductor, Statistical software
This article is included in the Bioconductor gateway.

Corresponding author: Michael I. Love (milove@email.unc.edu)

Author roles: Zitovsky JP: Data Curation, Formal Analysis, Investigation, Software, Validation, Visualization, Writing – Original Draft Preparation, Writing – Review & Editing; Love MI: Conceptualization, Data Curation, Funding Acquisition, Methodology, Project Administration, Resources, Software, Supervision, Writing – Review & Editing

Competing interests: No competing interests were disclosed.

Grant information: JPZ was supported the National Institutes of Health [R01 HG009125]. MIL was supported by the National Institutes of Health grants [HG009937, R01 MH118349, P01 CA142538 and P30 ES010126]. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Copyright: © 2019 Zitovsky JP and Love MI. This is an open access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

How to cite this article: Zitovsky JP and Love MI. Fast effect size shrinkage software for beta-binomial models of allelic imbalance [version 1; peer review: 1 approved with reservations] F1000Research 2019, 8:2024 (https://doi.org/10.12688/f1000research.20916.1)

First published: 28 Nov 2019, 8:2024 (https://doi.org/10.12688/f1000research.20916.1)
Introduction

Allelic imbalance (AI) occurs when the two alleles of a gene are expressed at different levels in a diploid organism, and the measurement of AI is valuable in elucidating the factors that regulate the expression of genes. For example, for a diploid organism, the allele on one chromosome may have higher or lower expression levels compared to the allele on the other chromosome due to genetic variation in nearby non-coding regulatory sites, a process known as cis-regulation. Allelic imbalance in expression may also be associated with differential epigenetic state of the genomic region across the chromosomes. In some cases, differential allelic expression resulting from differential epigenetic state can be linked to the parent-of-origin of the alleles, a phenomenon known as genetic imprinting.

One challenge currently faced in allelic expression studies is that estimates for allelic expression proportions can be highly variable in the presence of low read counts and/or small sample sizes. Large estimates of allelic proportions in these cases often result from estimation error as opposed to true differences in allelic expression. Though small samples and low counts are a problem for RNA-seq data in general, they are especially problematic when dealing with allele-specific counts. When a subject is heterozygous for a gene at a particular SNP(s), RNA-seq reads that overlap the SNP(s) allow for quantification of the levels of expression from either chromosome. Thus, allelic expression cannot be measured within a gene for subjects that are homozygous for that gene, and the number of samples with allele-specific counts for a gene can be much less than the number of samples in the study. Furthermore, alleles are often differentiated by a single SNP, and RNA-seq reads that do not overlap the SNP cannot be mapped to either allele. For these reasons, the proportion of mRNA-seq reads that are allele-specific can be quite low, depending on both read length and heterozygosity of the subjects. For instance, one study with 2x50 base pair (bp) paired-end reads and 30 million heterozygous SNPs from breast tumors of 550 human subjects found that allele-specific counts made up only 3.4% of RNA-seq reads. Experiments making use of model organism crosses can maximize the number of RNA-seq reads overlapping heterozygous SNPs, for example Raghupathy et al. found in an RNA-seq dataset of a mouse F1 cross that 22% of uniquely mapping reads were allele-specific.

One traditional remedy investigators have used to deal with the challenges of high-variance estimates is to filter out genes that have low counts or small samples. While this does cause the resulting estimates to be more stable and thus representative of true allelic expression proportions, filtering may also remove genes that have true allelic imbalances. Furthermore, the cutoff used to determine what genes to filter out (i.e. how many counts a gene must have for it to not be removed) must be chosen per dataset by the analyst. Another potential remedy to the problem of variable estimates from low-count and low-sample genes is to use Bayesian shrinkage estimators to moderate estimates.

A large number of Bayes estimators have already been developed for allelic expression studies. For instance, MMSEQ uses a Gamma prior on allele-specific transcript abundance to provide isoformal and allelic imbalance estimates that are more accurate and stable in the face of low coverage. Other methods that have used Bayesian approaches to test for AI include those by León-Novelo et al. and Skelly et al. However, these methods can only test overall AI, and cannot test the effects of covariates, such as different groups, on AI. More recently, a method was developed that expanded on that by León-Novelo et al. and was able to estimate AI within groups as well as compare AI between groups. It uses Bayesian shrinkage estimates for its parameters to shrink allelic proportions within groups toward 0.5, overdispersion toward a pre-specified prior mean, and the total counts of both alleles toward a pooled estimate. While the method is more flexible than the other methods listed, it still cannot estimate the effects of continuous covariates on allelic imbalance, nor can it estimate differences in AI between groups while controlling for additional confounding variables. Furthermore, the authors showed their method to be effective in reducing type I and type II error in the face of different sources of bias, the advantage of their method in estimation accuracy itself and in the face of genes with low counts need to be more thoroughly investigated.

Though gene expression read counts are typically larger than allele-specific counts and can be measured for all subjects, the uncertainty of estimates in the presence of low counts and/or low sample sizes is still an issue. Thus, several shrinkage estimators for log fold changes in gene expression have also been developed which shrink estimates that are only large due to the variance of the estimator and leave unchanged estimates that are likely to be large due to true expression changes.

Many of these methods directly involve or can easily be applied to linear models, which provide great flexibility in the kinds of study designs that can be treated and hypotheses that can be investigated. Though these methods were originally developed for improving accuracy and stability of log fold change estimates in gene expression, several can be directly applied or at least easily extended to estimating the effects of covariates on allelic expression proportions.

To this end, we look at three different estimation methods and their performance on data sets with small-to-moderate numbers of samples: maximum likelihood (ML), approximate posterior estimation of GLM coefficients (apeglm) and adaptive shrinkage (ash). ML estimates are based on estimating effects by modelling allele-specific counts with a beta-binomial GLM. Apeglm and ash are Bayesian shrinkage estimators which shrink maximum likelihood-based estimates toward zero. Our results show that while apeglm is not always the best method, it always performs better than ML and never performs much worse than ash for most metrics, making it the most robust and reliable when dealing with small sample sizes in our analysis. We also introduced new source code for the apeglm package to improve computational performance for beta-binomial GLMs, and compared our improved package to other R packages that can also fit beta-binomial GLMs. As the apeglm package can calculate both ML and Bayesian shrinkage estimates, our improvements can be used even by those
who wish not to use shrinkage estimators. Compared to other R packages, we show that apeglm with our improved code gives better running times and greater scalability with the number of covariates.

**Methods**

**Estimation methods**

We evaluated three estimation methods on their ability to estimate allelic expression proportions (or equivalently, the effects of covariates on allelic expression proportions): maximum likelihood estimation (ML estimation or MLE) with the likelihood described below, approximate posterior estimation of GLM coefficients (apeglm) and adaptive shrinkage (ash).

All analyses were done using R version 3.5.1. The first two methods mentioned are implemented in the apeglm v.1.7.5 package, while the last is implemented in the ash v.2.2.32 package. When using the ash function in the latter package, we set the method parameter equal to “shrink”. While there are many Bayesian estimation methods that can be used to quantify allelic imbalance, these allow for arbitrary design matrices. For instance, these methods can estimate differences in AI between groups while controlling for, or allowing interactions with, multiple additional variables, and can estimate the effects of continuous variables on AI.

For the $g$-th gene ($1 \leq g \leq G$), a beta-binomial GLM was fit to model allele-specific counts as follows. Let $Y_{ig}$ be the read counts of the first of the two alleles (which allele is designated as the first allele is arbitrary) for the $i$-th subject, $1 \leq i \leq I$. Investigators may designate the first and second alleles of a gene as the paternal and maternal alleles or as the alternate and reference alleles. It is assumed that $Y_{ig} \sim \text{BetaBin}(n_{ig}, p_{ig}, \phi)$, where $n_{ig}$ is the equal to the total counts of both alleles for the $i$-th subject, $p_{ig}$ is the probability of counts belonging to the first allele of the $i$-th subject, and $\phi$ is the overdispersion parameter. For the remainder of this paper, we will refer to the total allele-specific counts for both alleles of a particular gene and for a particular sample as the ‘total counts’ for that gene and sample. Furthermore, we will refer to the probability that counts for a particular gene belong to a particular allele for a particular sample as the ‘allelic proportion’ for that particular allele and sample. In this case, $\phi \rightarrow \infty$ implies no overdispersion beyond what would be seen in a binomial distribution and $\phi \rightarrow 0$ implies increasing variance. $n_{ig}, ..., n_{ig}$ are assumed to be fixed and known.

As the beta-binomial probability density function has multiple forms and parameterizations, we specify our parametrization as:

$$f(y; n, p, \phi) = \binom{n}{y} B(y + \phi p, n - y + \phi(1 - p))$$

where $B$ specifies the beta function. Furthermore, let $x_{ij}$ be the $i$-th row of the design matrix $X$ (matrix where columns are vectors of covariates of interest). Potential predictors include disease status for association studies, parent of origin for imprinting studies, and the presence of a SNP for eQTL linkage studies. We also assume that $p_{ig} = \left[1 + \exp(-x_{ij} \beta_j)\right]^{-1}$, or equivalently $\logit(p_{ig}) = x_{ij} \beta_j$, where $\beta_j = (\beta_{j1}, ..., \beta_{jK})^T$ is a vector of coefficients representing the effect sizes for the predictors in the design matrix. For ML estimation, $\beta_j$ is estimated via maximum likelihood. Constrained optimization is used for the nuisance parameter $\phi$, with a maximum of 500, so that genes with no overdispersion have finite estimated values of $\phi$.

Apeglm additionally assumes a zero-centered Cauchy prior distribution for the effects of one of the predictors. For estimating the effect of the $j$-th predictor in our model, where $1 \leq j \leq K$ is chosen by the user, and for the $g$-th gene, we have:

$$Y_{ig} | \beta_j \sim \text{BetaBin}(n_{ig}, p_{ig}, \phi)$$

$$p_{ig} = \frac{1}{1 + \exp(-x_{ij} \beta_j)}$$

$$\beta_j \sim \text{Cauchy}(0, \gamma_j)$$

Apeglm shrinks the effect of one chosen predictor at a time, across all genes. The scale parameter of the Cauchy prior, $\gamma_j$, is estimated by pooling information across genes. The posterior distribution of $\beta_j$ is the product of the above Cauchy prior and beta-binomial likelihood, and apeglm provides Bayesian shrinkage estimates based on the mode of the posterior as well as standard errors. Genes with lower expression, smaller numbers of heterozygous subjects and higher dispersion in allelic proportions will have flatter likelihoods, which will lead to the prior having more influence and shrinkage being greater. Furthermore, if the ML estimates are tightly clustered about zero, the estimated scale parameter of the Cauchy prior will be smaller. This will lead to more peakedness in the prior and also cause shrinkage to be greater.

The original apeglm package estimated regression coefficients using C++ for negative binomial GLMs, while GLMs with other likelihoods, such as the beta-binomial, were fit completely in R. To improve scalability for large data sets with beta-binomial GLMs, we wrote fast C++ code for calculating maximum likelihood and apeglm shrinkage estimates of beta-binomial regression coefficients. We also changed the source code to speed up computation of the standard errors (though such computations were still done in R) and prevent convergence issues. Details can be found in the Supplementary Methods section.

Ash is a general Empirical Bayes shrinkage estimator for hypothesis testing and measuring uncertainty in a vector of effects of interest, such as a set of log fold changes in gene expression between biological conditions. Suppose again that one is interested in the effect sizes of the $j$-th predictor, $\beta_j = (\beta_{j1}, ..., \beta_{jK})$, where $1 \leq j \leq K$. Ash takes as input a vector of ML estimated effects $\hat{\beta}_j = (\hat{\beta}_{j1}, ..., \hat{\beta}_{jK})$ and corresponding
estimated standard errors $\sigma_{\hat{\beta}_g} = (\sigma_{\hat{\beta}_1}, \ldots, \sigma_{\hat{\beta}_G})$. Here we take the estimated standard errors to be the true standard errors as suggested in the original methodology for ash, though the developers of ash have recently proposed an extension to their method that allows for random errors. For all $1 \leq g \leq G$, it is assumed that $\hat{\beta}_g | \beta_g, \sigma_{\beta_g} \sim N(\beta_g, \sigma_{\beta_g})$ and that $\beta_g \sim h_g$, where $h_g$ is some unimodal, zero-mode prior distribution. $\hat{h}_g$ is estimated from the ML estimates using mixtures of uniforms and a point-mass at zero, a choice guided by the author’s claim that any unimodal distribution can be approximated as a mixture of uniforms with arbitrary accuracy. The posterior is $\beta_g | \hat{\beta}_g, \sigma_{\beta_g} \sim N(\beta_g, \sigma_{\beta_g}) \times h_g$, and ash provides Empirical Bayes shrinkage estimates using the mean of the posterior as well as standard errors. Genes with larger standard errors for their ML estimates will have a flatter likelihood that will be less impactful on the estimation. Thus, estimates for these genes will be shrunk more. Like apeglm, ash can only shrink estimates for one covariate at a time.

Datasets and simulations

We compared the three estimation methods using the data set from the allelic expression study by Crowley et al. The study took mice from three divergent inbred strains (CAST/EiJ, PWK/PhJ and WSB/EiJ) and performed a diallel cross. The data set contains ASE counts for 72 mice and 23,297 genes in the resulting cross, with 12 mice of each possible parent combination (e.g. CAST/EiJ as mother and PWK/PhJ as father is one parent combination, and PWK/PhJ as mother and CAST/EiJ as father is another), and an equal number of males and females within each parent combination. Sequencing was performed with the Illumina HiSeq 2000 platform to generate 100-bp paired-end reads and following the TruSeq RNA Sample Preparation v2 protocol. To assure that the mice all had the same preparation v2 protocol. To assure that the mice all had the same

The estimators were then evaluated on real data with the focus on estimating mean, or gene-wide, allelic imbalance. From the mouse data set, random samples of size 6 were drawn, and this process was repeated 100 times. We will refer to these samples throughout the paper as the ‘random subsamples’. For each random subsample, the ML, apeglm and ash estimates of intercept-only models were calculated for the genes (where the intercept term was shrunk), and the MLE of the held-out 18 mice was taken to be the truth. Estimating the intercept in an intercept-only model for each gene is equivalent to estimating overall allelic imbalance for each gene.

Additional simulations were conducted for evaluating computational performance of our improvements to apeglm, to see how well they would scale to larger and more complicated data sets. Allele-specific counts were simulated in a similar manner as the apeglm vignette. Briefly, we have $\mathbf{Y}_{100\times5000} = [y_{ij}]$ as our simulated count matrix for one allele with associated total count matrix $\mathbf{N}_{100\times5000} = [n_{ij}]$ where rows are samples and columns are genes, $y_{ij} \sim \text{BetaBin}(n_{ij}, \mu, \phi)$, $\theta \sim U (0, 1000)$, $p_r \sim N(.5, .05)$, $n_{ij} \sim \text{NB}(\mu, 1/\phi)$, and $\mu, \phi$ are based on the airway data set by Himes et al. To see how well our improvements scaled with increasing numbers of covariates, the data were split multiple times into differing numbers of groups of approximately equal size, where the number of groups ranged from 2 to 10. With K groups, the design matrix $\mathbf{X}_{100\times K} = [1 \ x_1 \ldots x_K]$, where $\mathbf{x}_i$ is an indicator variable for the $(j + 1)$-th group, or a row vector whose $i$-th element is 1 if the $i$-th sample is in the $(j + 1)$-th group and 0 otherwise. A simulation was also conducted to see how well apeglm would work with continuous predictors. This time, $\mathbf{Y}$ and $\mathbf{N}$ was kept the same, but with the design matrix $\mathbf{X}_{100\times K} = [1, x_1, x_2, \ldots, x_K]$, where $\mathbf{x}_K = (1, 0, 1, 0, \ldots)$ separates the samples into two equally sized groups and $x_2, x_3 \sim N (0, 1)$.
Data processing

Genes where at least three samples did not have at least 10 counts were removed, which we considered minimal filtering that shouldn’t decrease statistical power. Genes without at least one count for both alleles across all individuals were removed. Genes with a marginally significant sex or parent effect were removed, so that all samples could be assumed independent and identically distributed for all genes. Genes were removed from the mouse data set prior to conducting random sampling from the data set or simulations.

To determine whether sex or parent effects were significant, beta-binomial GLMs were estimated for each gene by maximum likelihood, with a design matrix that included a sex effect (an indicator that was 1 if male and 0 if female), a parent-of-origin effect (an indicator that was 1 if the mother was the CAST/EiJ strain and 0 if the father was the CAST/EiJ strain) and an interaction term. For each gene, if the p-value for the sex, parent-of-origin or interaction effect was less than 0.1, the effect was deemed marginally significant for that gene.

Technical details of evaluations

For each gene, we define the shrinkage score as movement from MLE to zero. We define a gene as (noticeably) shrunk if shrinkage exceeds 0.1, and substantially or most shrunk if shrinkage is greater than max \( |\hat{\beta}_{MLE} - \beta| / 4 \). For instance, if an apeglm estimate for a gene is 0.15 closer to zero than the MLE, then the shrinkage score is 0.15 and the gene is noticeably shrunk but not substantially shrunk by apeglm.

Concordance at the top (CAT) plots were used to determine which estimation method could best find the most important genes (the genes with the greatest allelic imbalance or largest effect size). For an estimation method, concordance at the top takes the top genes according to the true ranking and compares it to the top genes according to the estimates, where the top genes are the genes with the largest true or estimated effect sizes in absolute value. For instance, a concordance at the top 10 of 90% means that the top 10 genes according to the estimation method and the top 10 genes according to the truth agree for 9 out of 10 genes.

For evaluating the performance of the three methods in estimating intervals, we calculated normality-based 95% confidence and credible intervals (both of which we will abbreviate as CIs) of the ML and apeglm estimators using their standard errors, or intervals based on the Laplace approximation of the likelihood and posterior. Such normality-based intervals are the default and suggested method for the apeglm package. Credible intervals in the ashr package were calculated from directly estimating tail probabilities of the posterior.

For each of the design matrices posited in our computation simulation, computational performance of apeglm estimation was compared between the old and new apeglm code. From apeglm v1.7.5, we set the method parameter equal to "betabinCR" to run the new C++ code, and set the log.lik parameter equal to a beta-binomial log-likelihood function to run the old code from before our improvements were introduced (version 1.6.0 of the package). Details can be found in the vignette. Computational performance of ML estimation was also compared between our improved apeglm package and the following packages: aod v1.3.12, VGAM v1.1.22, aods3 v0.42, gamlss v5.1.24 and HRQoL v1.0.25. Computational performance was evaluated using the microbenchmark v1.4.6 package for estimation of a single gene and elapsed time for estimation of all 5000 genes, on a 2012 15-inch MacBook Pro with an Intel Core i7-3720QM processor.

Determining the optimal filtering rule

In addition to comparing the three estimation methods described above, maximum likelihood estimation paired with optimal filtering criteria was also assessed via concordance at the top. CAT was chosen over other benchmark metrics, such as mean absolute error, as the different number of genes after filtering would make comparisons between filtered MLE and the three unfiltered methods biased. Furthermore, as we were primarily interested in whether a good filtering rule even existed, the true ranking of genes was used to determine the filtering rule. We looked at three rules: 1) removing genes where less than half the samples had a minimum total count threshold, 2) removing genes where less than all the samples had a minimum total count threshold, and 3) removing genes where the sum of total counts across samples was less than a certain threshold. For the remainder of the paper, we will refer to the sum of total counts across samples as the ‘summed counts’ of a gene. For each rule, various different thresholds were looked at: \( \{0, 10, ..., 200\} \) were potential thresholds for rule 1, \( \{0, 10, ..., 100\} \) were potential thresholds for rule 2, and \( \{0, 50, ..., 1000\} \) were potential thresholds for rule 3.

We will refer to the rule and threshold that had the best concordance as the ‘optimal filtering rule’.

Results

Standard normal simulation

We began by looking at a simulation where allelic counts came from known beta-binomial distributions and effect sizes came from a standard normal distribution. In this simulation, apeglm and ash successfully shrunk erroneously large estimates and reduced estimation error, particularly for genes that were noticeably shrunk (see Table 1 and Figure 1).

All three estimation methods gave similar mean absolute error (MAE), as many genes did not differ much between the methods (Table 1). In exploring the behavior of shrinkage estimators, we were most interested in genes where shrinkage was high, and thus where estimates would be much closer to or much farther from the truth for one estimation method than for another. Thus, in addition to overall MAE, we also calculated MAE among genes that were noticeably shrunk by apeglm and genes that were noticeably shrunk by ash, to determine whether there was substantial improvement on average when apeglm or ash did noticeably shrink a gene. Among genes that
were shrunk by apeglm, apeglm decreased the mean absolute error by 18.1%, and among genes that were shrunk by ash, ash decreased the mean absolute error by 21.1%. Moreover, from Figure 1a–c, it can be seen that apeglm shrunk most ML estimates that were inflated, bringing them closer to the truth, and mostly left truly large effects alone. Ash also shrunk ML estimates that were inflated, including some inflated estimates missed by apeglm. However, ash also had a tendency to incorrectly and excessively shrink: some genes with estimates close to the truth were severely shrunk, and several genes with truly large effects were shrunk to zero. Because of this tendency to over-shrink, ash performed worse among genes with large effects than among genes with small effects. For instance, among genes with effect sizes greater than two in absolute value, ash estimates had a higher mean absolute error than the MLE.

Ash and apeglm also performed better than the MLE in determining the most important genes, where concordance at the Top was higher regardless of the number of genes being considered (Figure 1d). Apeglm performed slightly better than ash in concordance at the top 100 genes, but otherwise they performed about the same. Concordance at the top for the MLE was optimized when filtering out genes with summed counts less than 350. Using this filtering, we were able to get CAT results better than that of the shrinkage estimates, even if only by a very small amount. Thus, for this simulation, it was possible to outperform both apeglm and ash with filtering alone.

---


<table>
<thead>
<tr>
<th>Performance Metric</th>
<th>MLE</th>
<th>Apeglm</th>
<th>Ash</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean Absolute Error</td>
<td>0.152</td>
<td>0.142</td>
<td>0.141</td>
</tr>
<tr>
<td>Mean Absolute Error (apeglm-shrunk genes)</td>
<td>0.498</td>
<td>0.408</td>
<td>0.397</td>
</tr>
<tr>
<td>Mean Absolute Error (ash-shrunk genes)</td>
<td>0.469</td>
<td>0.38</td>
<td>0.37</td>
</tr>
<tr>
<td>Mean Absolute Error (</td>
<td>effect size</td>
<td>2)</td>
<td>0.221</td>
</tr>
<tr>
<td>Coverage Probability for 95% CI</td>
<td>0.951</td>
<td>0.951</td>
<td>0.949</td>
</tr>
<tr>
<td>Average Interval Width for 95% CI</td>
<td>0.761</td>
<td>0.697</td>
<td>0.685</td>
</tr>
</tbody>
</table>

---

**Figure 1. Truth vs. estimate and CAT Plots for normal simulation.**

- **a)** truth vs. estimate plot for MLE. Blue points represent genes substantially shrunk by apeglm only, orange points represent genes substantially shrunk by ash only and green points represent genes substantially shrunk by both ash and apeglm.
- **b)** truth vs. estimate plots for apeglm.
- **c)** truth vs. estimate plots for ash.
- **d)** CAT plot for the three methods as well as for MLE after filtering. CAT: Concordance at the Top, MLE: Maximum Likelihood Estimation, apeglm: Approximate Posterior Estimation of Generalized Linear Model Coefficients, ash: Adaptive Shrinkage.
(provided that the true ranking of genes was known, and used to determine the optimal filtering rule).

With regard to the extent of shrinkage, both apeglm and ash mainly exhibited shrinkage for genes that had very low counts (Supplementary Figure 3'). This is not too surprising for this particular simulation, as after filtering out lowly-expressed genes, the remaining ML estimates were much closer to the truth (Supplementary Figure 4'). When comparing shrinkage scores between apeglm and ash, we found that there was a clear upward shift of shrinkage scores for ash (Supplementary Table 1'), further showing that ash had more extreme shrinkage than apeglm for this dataset. Though all three methods gave intervals that were similar in coverage probability, average interval width was smaller for apeglm and ash compared to the MLE (Table 1).

**Student’s t Simulation**

We also investigated the performance of the estimators when most of the effect sizes were close to zero and overdispersion was large. Here the shrinkage estimates had even more marked improvement over the ML estimates (see Table 2 and Figure 2).

Apeglm improved mean absolute error by 49.5% among all genes, and by 65.4% among noticeably shrunk genes specifically (Table 2). Ash improved mean absolute error by 52.2% among all genes and by 65.8% among noticeably shrunk genes specifically. These improvements were greater than that seen from the standard normal simulation. Figure 2a–c show that ash successfully shrunk inflated ML estimates closer to the truth while leaving truly large effects mostly unchanged.

### Table 2. Performance metrics for Student’s t Simulation.

<table>
<thead>
<tr>
<th>Performance Metric</th>
<th>MLE</th>
<th>Apeglm</th>
<th>Ash</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean Absolute Error</td>
<td>0.186</td>
<td>0.094</td>
<td>0.089</td>
</tr>
<tr>
<td>Mean Absolute Error (apeglm-shrunk genes)</td>
<td>0.353</td>
<td>0.122</td>
<td>0.113</td>
</tr>
<tr>
<td>Mean Absolute Error (ash-shrunk genes)</td>
<td>0.336</td>
<td>0.126</td>
<td>0.115</td>
</tr>
<tr>
<td>Coverage Probability for 95% CI</td>
<td>0.924</td>
<td>0.937</td>
<td>0.941</td>
</tr>
<tr>
<td>Average Interval Width for 95% CI</td>
<td>0.873</td>
<td>0.45</td>
<td>0.456</td>
</tr>
</tbody>
</table>

**Figure 2.** Truth vs. estimate and CAT Plots for Student’s t Simulation.  

- **a)** truth vs. estimate plot for MLE. Orange points represent genes substantially shrunk by ash only and green points represent genes substantially shrunk by both ash and apeglm. All genes substantially shrunk by apeglm were shrunk by practically the same amount or more by ash.  
- **b)** truth vs. estimate plots for apeglm.  
- **c)** truth vs. estimate plots for ash.  
- **d)** CAT plot for the three methods as well as for ML after filtering. MLE: Maximum Likelihood Estimation, apeglm: Approximate Posterior Estimation of Generalized Linear Model Coefficients, ash: Adaptive Shrinkage.
Apeglm brought many inflated ML estimates closer to the truth as well, but not as many as ash.

Concordance at the top was better for the shrinkage estimates than for the ML estimates, regardless of the number of top genes in question. Furthermore, similar to mean absolute error, the improvements seen from the shrinkage estimates over the MLE was larger than those seen from the standard normal simulation. Ash performed better than apeglm in concordance at the top 50 and 100 genes, though performance was similar when looking at a larger number of genes. Concordance at the top for the MLE was optimized when filtering out genes where less than half the samples had at least 110 counts. Though this improved CAT by quite a lot, performance was still much lower than apeglm and ash Thus, unlike in the standard normal simulation, the performance in CAT obtained by shrinkage could not be matched with filtering, even when using the true gene ranking to determine the optimal filtering rule.

In this simulation, due to the increased overdispersion, there were many effects that were overestimated or underestimated by ML, even among genes with large counts. Because of this, both ash and apeglm exhibited shrinkage for effects across the dynamic range of summed counts, as opposed to only shrinking effects with small counts (Supplementary Figure 5). Together with the true vs. estimate plots, this shows that both apeglm and ash can correctly shrink falsely large effects even when the summed counts are large. Ash had larger shrinkage scores than apeglm on average, indicating that ash tended to shrink estimates more than apeglm (Supplementary Table 2). All methods had coverage slightly less than nominal (95%), ranging from 92 to 94%. However, both apeglm and ash had half the average interval width compared to maximum likelihood, despite both having slightly higher coverage rates.

We also conducted simulations similar to that of the standard normal and Student’s t, but with 5 vs. 5 samples. Like the 4 vs. 4 case, both apeglm and ash had lower average estimation error and higher concordance at the top than the MLE (results not shown).

Sampling from the mouse dataset
To evaluate performance on real data, we took 100 random subsamples of 6 mice from the mouse data set and averaged various performance metrics across the random subsamples. Similar to the simulations, apeglm appeared to improve estimation accuracy and shrink erroneously large genes. Ash, on the other hand, appeared to perform worse than the MLE according to mean absolute error, concordance at the top and interval coverage (see Table 3 and Figure 3).

Among genes that were shrunk by apeglm, mean absolute error was 14.2% lower on average for apeglm (Table 1). On the other hand, average MAE rose by 34.1% for ash among genes that were shrunk. Moreover, the minimum MAE obtained by ash across all 100 random subsamples was larger than the maximum MAE obtained by apeglm (results not shown). From Figure 3a–c, ash appears to be over-shrinking, and some of the genes with the largest held-out effect estimates were shrunk to zero. Though some genes also appeared to have been incorrectly or overly shrunk by apeglm, apeglm mainly was observed to shrink genes with inflated estimates and over-shrinkage was normally less severe when it occurred.

Both apeglm and MLE had universally higher concordance at the top than ash (Figure 1d). While apeglm performed slightly better than the MLE in concordance at the top 50 genes, performance was identical when looking at larger numbers of genes. We found that any filtering only decreased concordance at the top, as many top genes had low counts (i.e. the optimal filtering rule was no filtering). The most likely reason for this is that for each random subsample, we are treating the MLE of the held-out set as the truth. Thus, estimation error in the face of low-count genes would affect the held-out effect estimates and bias CAT results to some degree, even though the held-out sets have larger numbers of samples and performance metrics are averaged over many random subsamples. However, because genes with very large held-out effect estimates are more likely to have low counts, metrics that average across all genes, such as mean absolute error, would not be biased as much by estimation error.

A large amount of variability in the ML estimates was discernible for genes with low counts (Supplementary Figure 6). Like in the standard normal simulation, the low-count genes were mainly the ones shrunk by apeglm and ash. As the truth vs. estimate plots suggest, ash had larger shrinkage scores than apeglm (indicating more extreme shrinkage), and with the difference in shrinkage between the two methods being larger than in the simulations (Supplementary Table 4). Though apeglm intervals appeared to have smaller coverage than the ML intervals, the difference in coverage was very small, and average interval width was also 26.8% smaller for apeglm than that of maximum likelihood. Ash intervals were slightly more

<table>
<thead>
<tr>
<th>Table 3. Performance metrics averaged across random subsamples.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Performance Metric</strong></td>
</tr>
<tr>
<td>Mean Absolute Error</td>
</tr>
<tr>
<td>Mean Absolute Error (apeglm-shrunk genes)</td>
</tr>
<tr>
<td>Mean Absolute Error (ash-shrunk genes)</td>
</tr>
<tr>
<td>Coverage Probability for 95% CI</td>
</tr>
<tr>
<td>Average Interval Width for 95% CI</td>
</tr>
</tbody>
</table>
narrow than apeglm, with average interval width 32.5% smaller than that of maximum likelihood, but coverage was also lower.

As the mouse data set only had 24 samples, we determined that we didn’t have the sufficient sample size to evaluate our methods on estimating effect sizes of predictors, or models with more than just an intercept term. For instance, even if we wanted to look at the performance of our methods on estimating a group effect with only four samples in each group, each held-out set would only have eight samples in each group. Thus, the ML estimates of the held-out sets would have a lot of variance and could be far from the truth.

Computational performance of Apeglm
To evaluate the computational performance of our package on larger datasets, we simulated allelic counts for 5000 genes and 100 samples, and randomly divided the samples into differing numbers of groups. apeglm with our improvements had very fast running times for both ML and apeglm estimation and scaled well with the number of covariates (see Figure 4 and Figure 5).

Estimation times per gene for ML estimation was substantially faster for apeglm than all other packages (Figure 4). The next best package, aods3, took 5 to 11 times longer than apeglm and did not scale as well with the number of groups. Furthermore, the aods3, gamlss and HRQoL packages occasionally produced errors and could not fit beta-binomial models for all the simulated genes.

For estimating all genes in the simulation via maximum likelihood, apeglm took 24 seconds for two groups and added only 1–2 seconds of computational time for every group added (Figure 5a). The next fastest package that could fit beta-binomial models for all the genes, aod, took seven times longer for two groups and grew 80 times as much for every group added. Comparisons in apeglm estimation between our improved apeglm package and the original package gave similar conclusions. Furthermore, unlike the new apeglm package, which grew roughly linearly with the number of groups in the range we assessed, the order of growth from the original package was not linear: the greater the number of groups already in the model, the greater the computational time increased for adding additional groups. At 10 groups, our improvements made apeglm 27 times faster than aod for ML estimation and 33 times faster than the old package for apeglm estimation. Our improvements also performed quite favorably when fitting beta-binomial models with two groups and two numerical controls. Elapsed time was 31 seconds for ML estimation and 43 seconds for apeglm estimation with the new apeglm package. In contrast, ML estimation took over nine minutes for aod and apeglm estimation took over seven minutes for the old apeglm package. Introducing multicollinearity into the design matrix did not substantially change computational performance for any package (results not shown).

Discussion
Here the performance of three estimators was compared across two simulations and one real dataset of allele specific expression
in mice. Though apeglm was not the best estimator in all cases, it was the most robust and with consistent performance. Apeglm had smaller mean absolute error and greater concordance at the top than the MLE, and was never much worse than ash in these respects. Ash also performed better than the MLE for the simulated data for most metrics, including mean absolute error and concordance at the top. Moreover, ash had higher concordance at the top than apeglm in the Student’s t simulation. However, ash also had a tendency to over-shrink some genes, shrinking some truly large effects close to zero. Furthermore, for the real data set, ash performed worse than the MLE for most metrics, including mean absolute error and concordance at the top, most likely due to over-shrinking of many genes. As performance on the real data set was based on
taking random subsamples of mice and using the MLE of the held-out set as the truth, estimation error of the held-out effect estimates may have biased results. For future research, using larger data sets to analyze apeglm performance than that of Crowley et al. would allow for held-out sets with more samples and thus reduce estimation error of held-out effect size estimates.

The shrinkage estimators compared here typically shrunk only low-count genes, as low-count genes tend to be those with the most uncertain and variable estimates. However, during a simulation where extreme overdispersion and heavy tails of the distribution of true effects were introduced, there were some large-count highly-variable genes that were shrunk as well, showing that ash and apeglm will shrink large-count genes if there is high uncertainty in the estimates. Ash consistently had more extreme shrinkage than apeglm and greater estimation error among genes with truly large effects. Thus, ash would most likely perform best in a situation where most effects were small, such as in the Student’s t simulation.

No method gave confidence or credible intervals with the highest coverage rates for all scenarios. However, across both simulations and analysis of the mouse data, differences in coverage rates between the three methods were small, and coverage rates for apeglm credible intervals in particular were always very close to the interval that had the largest coverage. Furthermore, interval width for apeglm and ash were always smaller than that of maximum likelihood. This suggests that interval estimates from apeglm could be advantageous over those by maximum likelihood. For future research, it would be beneficial to evaluate the accuracy of hypotheses tests based on the estimates or posterior distribution of apeglm using metrics such as type I and type II error. The method of Leôn-Novelo et al. 2018 rejected hypotheses based on credible intervals of its posterior distribution, and if a similar step was taken for apeglm, its narrower intervals and robust coverage could potentially give more powerful hypothesis tests without suffering from inflated type I error.

Our changes to the apeglm package greatly improved computational performance for both ML and apeglm estimation of beta-binomial GLMs, particularly when larger numbers of covariates were involved. Among the R packages that we looked at which could fit beta-binomial models, the new apeglm package was always the fastest for fitting many GLMs in sequence, e.g. across many genes or variant locations. Thus, the new apeglm package is useful for quick and reliable analyses of allelic imbalance even for researchers who wish to only use likelihood-based estimators. Moreover, only coefficient estimates are currently calculated in C++, and even better computational performance would be achieved if overdispersion and standard error calculations were integrated into C++ as well. We are not aware of any other R packages that utilize faster programming languages such as C or C++ to estimate numerous beta-binomial regression models based on large matrices of observed allelic counts. The most similar package we noted was fastglm7, which fits individual quasi-binomial models in C++. While quasi-binomial models also estimate proportions and control for overdispersion, they do so in a different manner and with different assumptions.

Based on previous work, there are several ways in which the apeglm methodology could potentially be improved for allelic expression studies. For instance, while our extension of apeglm estimated overdispersion by MLE, the original methodology for apeglm as applied to negative binomial GLMs utilized Bayesian estimates for overdispersion as well as for regression coefficients. Introducing a prior for beta-binomial overdispersion that pools information across genes may lead to better estimation and inference of regression coefficients. We also assumed that the total allele-specific counts were fixed and known. Allowing such quantities to be random, as in the method by Leôn-Novelo et al. 2018, may lead to better inference as well. Adjusting for read mapping biases and ambiguities (Leôn-Novelo et al. 20147; Leôn-Novelo et al. 2018; Raghupathy et al. 2018) could also lead to better estimates when such biases and quantification uncertainty are present. Lastly, though here we focused on beta-binomial GLMs, a wide variety of statistical models can be used for ASE, from quasi-binomial24 to Poisson-lognormal models25.

Data availability

Underlying data


This project contains the following underlying data:

- fullGeccoRnaDump.csv

This file contains the Crowley et al. mouse dataset which was obtained from http://csbio.unc.edu/gecco/data/fullGeccoRnaDump.csv.gz. We uploaded the dataset to Zenodo on the authors’ behalf with their permission, due to the fact that the original dataset is not currently hosted in a stable repository.

The dataset from this repository is available under the terms of the Creative Commons Attribution 4.0 International license (CC-BY 4.0).

Extended data


This project contains the following extended data:

- Supplementary Methods.pdf (Contains the mathematical and algorithmic details of how the apeglm package estimates beta-binomial coefficient effect sizes by maximum likelihood and apeglm, including the steps taken to improve computational performance, increase numerical stability and prevent convergence issues)

- Supplementary Figures and Tables.pdf (Contains supplementary figures 1–6 and supplementary tables 1–3. These figures and tables were referenced and described in the main body of the article)
Data are available under the terms of the CC-BY 4.0 license.

Software availability

The software from this repository is available under the terms of the GNU General Public License v3.0 (GPL-3).

Zenodo: Source Code for Zitovsky and Love 2019. http://doi.org/10.5281/zenodo.3404669^{12}. This repository contains the R scripts used to run the analyses described in this article and generate all of its figures. All figures associated with this paper, including figures present in the main article and supplementary figures, were generated as separate png and .eps files and can also be found in this repository. The R scripts can be found under the ‘Code’ folder while the figures can be found under the ‘Figures’ folder.

Material from this repository are available under the terms of the GPL-3 license.

apeglm is available as part of the Bioconductor project^{33} at http://bioconductor.org/packages/apeglm. The vignette^{38} and manual provide detailed information on how to use the package.

Acknowledgements
We thank Anqi Zhu and Joseph G. Ibrahim of the Department of Biostatistics at UNC Chapel Hill for their contributions to the conceptualization and development of the original apeglm methodology, and Rob Patro for useful discussions.

References
Open Peer Review

Current Peer Review Status: ?

Version 1

Reviewer Report 17 December 2019

https://doi.org/10.5256/f1000research.23018.r57281

© 2019 Stephens M. This is an open access peer review report distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Matthew Stephens
1 Department of Statistics, University of Chicago, Chicago, IL, USA
2 Department of Human Genetics, University of Chicago, Chicago, IL, USA

Summary:

The paper presents new implementations of shrinkage methods for beta binomial models, implemented in the R software package apeglm. One potential application of these models is estimating allele-specific biases in various sequencing-based assays (and differences in bias between groups), and the paper focuses on this application.

The performance of the shrinkage methods is assessed via simulation and real data analysis (using performance on hold-out data as a performance metric), and the shrinkage methods implemented here are found to be competitive with another shrinkage approach (adaptive shrinkage, ash), and consistently outperform the mle. The new implementations are also shown to be computationally faster than existing implementations (e.g., aod or the previous version of apeglm).

The paper is generally well written, and carefully done, with some exceptions I note later. The new implementations seem likely to be useful in a range of applications. Certainly the use of shrinkage methods in these types of applications is to be encouraged, and I congratulate the authors for leading the way on this. I hope they will find my report helpful in revising their work.

I was instructed "Please indicate clearly which points must be addressed to make the article scientifically sound." I believe points 2-4 below are most important to address to make the article scientifically sound.

1. A note on differences between the shrinkage methods:

One thing that I felt was missing from the paper was a qualitative summary of how the two shrinkage methods used here differ from one another. Both are a form of Empirical Bayes shrinkage, but they use different prior families, different likelihoods, and different point estimate strategies: apeglm uses a Cauchy prior, with beta-binomial likelihood, and posterior mode point estimate; whereas ash uses a more flexible unimodal prior (which includes Cauchy as a special case), a normal approximation to the likelihood, and uses a posterior mean point estimate. So the trade-off here is that ash is using an approximate
likelihood, but a more flexible prior and arguably a more principled point estimate (posterior mean is optimal under mean squared error).

I think many readers might benefit from this "high-level" summary of the differences.

Another important point, which will come up later, is that when using ash the user has a choice of how to make the normal approximation. Specifically ash requires the user to provide point estimates (beta-hat) and standard errors (s-hat), with the goal that beta-hat approx \sim N(beta, s-hat), where beta is true value that is being estimated.

So there is not only one way to apply ash to a problem, but many different ways depending on the choice of point estimate beta-hat. The mle is one natural choice, but in this application there can be problems with infinite mles; see 2. below.

2. On dealing with infinite mles:

To explain the issue with infinite mles, consider first a simple binomial experiment X \sim Bin(n,p) in which we observe X=0. Then the mle for p is 0, and the mle for theta:=log(p/(1-p)) is -Infinity. Similarly, if X=n the mle for theta is Infinity. Also, in both cases, the standard error for theta is infinite. The same issue arises in the more complex beta-binomial models considered here.

Essentially if all the reads in an experiment show the same allele then the mle for the allelic bias parameter (on the logit scale) is +-Infinity. This could happen due to low coverage, but it could also happen at high coverage sites if the allelic bias is very strong.

This issue appears to arise in the data analyses used to produce Figure 3 (I did not check whether it arises in the simulations). In Figure 3 there appear many mles (y axis) taking values near +- (5 to 6); however, my brief investigations of the data suggested that most of these likely correspond to genes where all the reads come from one allele, and so the mle is actually +-Infinity as above. (That these infinite mles are computed to be near +6 is presumably due to an issue with the numerical maximization method used to compute the mle.)

I suspect that the problems with ash observed in Fig 3 stem from this issue: the mle for these situations where all the reads come from one allele are very unstable, and have a very large standard error (technically infinite, although for numeric reasons finite values are used) and these large standard errors cause these mles to be shrunk excessively.

A simple fix for this problem, and one I suggest the authors try, is to add a pseudo-count (say 1, or 0.5) to the counts for "each" allele in the data before computing "mles" and corresponding standard errors. Pseudo-counts are commonly used to improve stability of mles in this type of situation. Indeed, adding pseudo-counts can be viewed as a simple kind of shrinkage method, so it seems reasonable to compare the more sophisticated EB methods with the simple pseudo-count method. For most genes the point estimates and standard errors will be very little affected by the addition of a small pseudo-count; but for the problematic genes with infinite mle the pseudo-count will stabilize the point estimate and reduce the standard error. I suspect entering the stabilized estimates + standard errors into ash will greatly reduce the problems observed with use of the mles in Figure 3.

(Incidentally, Xing, Carbonetto and Stephens arXiv:1605.07787 encounter a closely-related issue when using ash to smooth Poisson data; they solved this using a slightly different approach that is conceptually similar to adding a pseudo-count.)
3. Subsetting results based on shrinkage amounts and “true” values:

In several places the paper reports error measures on subsets of the results. For example, in Table 1 lines 2-4 involve subsets of results chosen based on the true effect size or shrinkage amount (which depends on the true effect). Although tempting, this type of result is hard to interpret. For example, even the optimal shrinkage rule (i.e. the one that uses the correct prior, likelihood and loss function) may not perform uniformly better than the mle on subsets that are chosen in this way. Thus the sentence on p7 (“For instance, among genes with effect sizes greater than two...”) may also be true for the optimal shrinkage rule, and so does not constitute direct evidence for “overshrinkage”. (I agree there is overshrinkage, but this is not the right way to show it). Comparisons like p9 (“Among genes that were shrunk...”), which stratify by the amount of shrinkage, have the same problem because the amount of shrinkage depends on the true value and not only on the observed value.

It is much cleaner and easier to interpret results if they are subsetted based on the *observed* effect (mle), rather than the true effect. This is because the optimal shrinkage rule is still optimal for *any subset chosen based only on the observed data*. (For this reason you could also subset based on other features of the observed data, like total allele count.) For example, if a method is worse than the mle for the subset of results where the mle is >4 then this is indeed evidence of a problem of some kind.

4. Computation: speed vs accuracy:

When comparing with other methods/implementations there should be some assessment not only of speed, but of accuracy of the different implementations (meaning the accuracy with which they optimize the log-likelihood, rather than the accuracy of the point estimates). Fast answers are easy if you do not care about accuracy....

E.g. I suggest boxplots of loglik(method) - loglik(apeglm-new) for each method, to show that the apeglm-new solution is consistently as high in log-likelihood as other methods (or nearly so). Are there convergence criteria decisions to be made that might affect the trade-off between speed and accuracy?

5. Reproducibility:

I congratulate the authors on making all their code and data available. After a few tweaks to the code I was able to run the code used to produce Figures 1-3. However, my version of Fig 3 looked different from the one in the paper - my figure had different colors and some points seemed to be missing on my figure. I do not know the reasons for this.

Reproducibility would have been made easier by avoiding the use of absolute file paths. I also suggest not defining functions that operate on global variables (e.g. subsetCalculations = function(sub){...}) since they are more likely to lead to reproducibility problems.

I was unable to run the code to perform the computation time comparisons (Figure 4), since it errored out. Again I do not know the reason, but it could be due to differences in the package versions I used compared with the authors. I did not have time to troubleshoot this.

6. Miscellaneous other comments:
For Table 3, I think it should be noted that the coverage probability is expected to be <0.95 because you are looking at how often the interval covers the "estimate" in the larger dataset, and not the "true" value. This makes it a hard to compare the methods here because it isn't clear what the right coverage is.

p12: "ash would most likely perform best in a situation where most effects were small". I don't see any evidence for this here (e.g. in the normal simulation ash performs fine) and indeed no reason to expect it to be true a priori. I think this statement should be removed.

7. Minor comments:
   - p3: "When a subject is heterozygous for a gene at a particular SNP"; this wording seemed awkward to me.
   - p3: "... making it the most robust and reliable when dealing with small sample sizes"; this conclusion ("making it") seemed not to follow directly from the first part of the sentence.
   - p4: "Apegglm shrinks the effect of one predictor at a time": I think this sentence might work better at the start of the paragraph, before specifying the prior used.
   - p5: "guided by the author's claim": this is not just a claim, it is a theorem dating back to the 1950s (see original paper for citations).
   - p5: diallel typo?
   - p5: use of beta for the mean of the exponential distribution is confusing as beta is already used elsewhere.
   - p9: "We also conducted..." This did not seem worth reporting to me. The difference in sample size (5 vs 5 instead of 4 vs 4) is too small to expect that the results would be very different.
   - p9: In the paragraph "Both apeglm and MLE..." the acknowledgement that comparing against CAT in a hold-out set is potentially problematic is a bit buried in the middle of the paragraph. It would seem better to acknowledge this up front. Given the problems with CAT acknowledged here I suggest removing that figure (Fig 3d) or moving to an Appendix.

   - Figure 5: this should have a y axis that starts at 0.

References

Is the rationale for developing the new method (or application) clearly explained?
Yes

Is the description of the method technically sound?
Yes

Are sufficient details provided to allow replication of the method development and its use by others?
Partly

If any results are presented, are all the source data underlying the results available to ensure full reproducibility?
Yes

Are the conclusions about the method and its performance adequately supported by the findings presented in the article?
Partly

**Competing Interests:** No competing interests were disclosed.

**Reviewer Expertise:** Bayesian statistics; statistical genetics

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard, however I have significant reservations, as outlined above.

---

The benefits of publishing with F1000Research:

- Your article is published within days, with no editorial bias
- You can publish traditional articles, null/negative results, case reports, data notes and more
- The peer review process is transparent and collaborative
- Your article is indexed in PubMed after passing peer review
- Dedicated customer support at every stage

For pre-submission enquiries, contact research@f1000.com