METHOD ARTICLE

REVISED Fast effect size shrinkage software for beta-binomial models of allelic imbalance [version 2; peer review: 3 approved with reservations]

Joshua P. Zitovsky [1], Michael I. Love [1,2]

[1]Department of Biostatistics, University of North Carolina at Chapel Hill, Chapel Hill, NC, 27516, USA
[2]Department of Genetics, University of North Carolina at Chapel Hill, Chapel Hill, NC, 27514, USA

v2 **First published:** 28 Nov 2019, **8**:2024
https://doi.org/10.12688/f1000research.20916.1
Latest published: 14 Dec 2020, **8**:2024
https://doi.org/10.12688/f1000research.20916.2

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 estimators 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 pseudocounts or Bayesian estimators to reduce the variance. To this end, we evaluated the accuracy of four different estimators, the latter two of which are Bayesian shrinkage estimators: maximum likelihood, adding a pseudocount to each allele, 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 four methods were evaluated on two simulations and one real data set. Apeglm consistently performed better than ML according to a variety of criteria, and generally outperformed use of pseudocounts as well. Ash also performed better than ML in one of the simulations, but in the other performance was more mixed. Finally, when compared to five other packages that also fit beta-binomial models, the *apeglm* package was substantially faster and more numerically reliable, 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.

Open Peer Review

Reviewer Status ? ? ?

	Invited Reviewers		
	1	**2**	**3**
version 2 (revision) 14 Dec 2020			
version 1 28 Nov 2019	? report	? report	? report

1. **Matthew Stephens**, University of Chicago, Chicago, USA
University of Chicago, Chicago, USA

2. **Jarad Niemi**, Iowa State University, Ames, USA
Ignacio Alvarez-Castro, University of the Republic, Montevideo, Uruguay

3. **Ernest Turro**, University of Cambridge, Cambridge, UK
University of Cambridge, Cambridge, UK

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

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 (michaelisaiahlove@gmail.com)

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.

How to cite this article: Zitovsky JP and Love MI. **Fast effect size shrinkage software for beta-binomial models of allelic imbalance [version 2; peer review: 3 approved with reservations]** F1000Research 2020, **8**:2024 https://doi.org/10.12688/f1000research.20916.2

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

REVISED **Amendments from Version 1**

Many changes to the original manuscript were made, all of which are detailed in our notes to reviewers. Here we summarize only the most substantial modifications. Additional details about the methodology of the different estimators and their differences have been added to the methods section of the main manuscript and the supplementary methods section. The intercepts of our simulations are now drawn from a standard normal distribution, whereas before they were drawn from the estimates from intercept-only GLMs fitted to the mouse dataset. We no longer stratify mean absolute error by true effect size in the normal simulation and instead stratify results by total gene counts and maximum likelihood (ML) estimate magnitude. We investigated the impact of infinite ML estimates on results, and whether the performance of the different methods improve when using pseudo-counts. Real data results have been changed to focus on more qualitative assessments (e.g. extent of shrinkage and confidence/credible interval width) which do not assume knowledge of ground truth. Moreover, rather than just looking at intercept-only models, we now also compare the different methods on real data when using more complicated design matrices. We have added results regarding the numerical accuracy of our package compared to other beta-binomial GLM-fitting packages. All problems with reproducibility in the previous code have been addressed. The abstract and conclusions have been changed to more accurately summarize differences in apeglm's and ash's performance.

Any further responses from the reviewers can be found at the end of the article

Introduction

Allelic imbalance (AI) occurs when the two alleles of a gene are expressed at different levels in a diploid organism, and its measurement 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. AI 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 estimators 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 at a particular SNP within an exon of a gene, RNA-seq reads that overlap the SNP allow for quantification of the levels of expression from either allele[1]. 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 RNA-seq reads that are allele-specific can be quite low, depending on read length and heterozygosity of the subjects. For instance, one study with 2x50 base pair (bp) paired-end reads and examining 30 million SNPs from 550 human subjects found that allele-specific counts made up 3.4% of RNA-seq reads[2]. On the other hand, experiments making use of model organism crosses can maximize the number of RNA-seq reads overlapping heterozygous SNPs. For example, Raghupathy *et al.*[3] 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 estimators 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 AI. One can think of this as akin to achieving higher specificity in detecting large true effect sizes at the cost of sensitivity. 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 traditional remedy has involved adding a pseudocount to each allele prior to estimation. As we will show, however, Bayesian shrinkage estimators offer advantages in moderating estimates.

A large number of Bayes estimators have already been developed for allelic expression studies. For instance, MMSEQ[4] uses a Gamma prior on allele-specific transcript abundance to provide AI estimates that are more accurate in the face of low coverage. Other methods that have used Bayesian approaches to test for AI include those by Leòn-Novelo *et al.* 2014[5] and Skelly *et al.*[6] Leòn-Novelo *et al.* 2018 expanded on the work of Leòn-Novelo *et al.* 2014 and developed a method that can to estimate AI within groups as well as compare AI between groups[7]. It uses Bayes estimators 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 more flexible than its predecessor, their method still does not allow for arbitrary design matrices (e.g. it cannot estimate the effects of continuous covariates on AI), and performance evaluations mainly focused on type I and type II error, not estimation accuracy. Since the original publication of our work, a Bayesian method for ASE has been developed which does allow for arbitrary design matrices[8]. The method by Alvarez-Castro and Niemi 2019 models counts of each allele with over-dispersed Poisson regression models and places empirical Bayesian priors on both the regression and overdispersion parameters.

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 try to 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[9–13]. Many of these methods directly involve or can easily be applied to generalized 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. For instance, Turro E, Astle WJ and Tavaré S[13] uses their method to assess imprinting by including haplotype information in the design matrix. Moreover, many of these methods have flexible generalized linear model specifications, and assessing allelic imbalance is as simple as changing the likelihood and link function appropriately (e.g. from a Poisson or negative binomial likelihood and log-link to a binomial or beta-binomial likelihood and logit link). We focus on the latter approach.

To this end, we look at four different estimation methods and their performance on data sets with small-to-moderate numbers of samples: maximum likelihood (ML), adding a pseudocount to each allele and sample, approximate posterior estimation of GLM coefficients (apeglm)[12] and adaptive shrinkage (ash)[11]. ML estimators are based on estimating effects by modelling allele-specific counts with a beta-binomial GLM. Apeglm and ash are Bayesian shrinkage estimators which shrink likelihood-based estimates toward zero (ash can additionally handle non-likelihood-based estimates, such as quasi-likelihood estimates). Our results found that apeglm performed better than ML across a variety of metrics, making it robust and reliable when dealing with small sample sizes. Ash also performed better than ML in some metrics, though in other metrics results were more mixed. In addition to evaluating the performance of apeglm on allelic count data, we also introduced new source code for the `apeglm` package to improve computational performance for fitting 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 faster running times, greater scalability with the number of covariates, and better numerical reliability.

The methods and performance benchmarks we focus on here address issues stemming from low-count genes and small sample sizes. There are other important concerns in allele-specific analysis of short read RNA-seq datasets, such as reference allele bias, but we do not address such problems here and the methods discussed cannot directly account for them. Our simulation does not involve reference allele bias, and the RNA-seq study we examine took specific measures to avoid reference allele bias. For methods and analysis concerns involving reference allele bias, see Turro *et al.*[4] and Castel *et al.*[1]

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 (ML) estimation with the likelihood described below, approximate posterior estimation of GLM coefficients (apeglm) and adaptive shrinkage (ash). All analyses was done using R version 4.0.2[14]. The first two methods mentioned are implemented in the apeglm v.1.11.2 package, while the last is implemented in the ashr v.2.2.47 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 (AI), 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 ($1 \le g \le G$), 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, $1 \le i \le I$. Investigators may designate the first and second alleles of a gene as the paternal and maternal alleles or as the alternate and reference alleles, for example. It is assumed that $Y_{ig} \overset{ind}{\sim} \text{BetaBin}(n_{ig}, p_{ig}, \phi_g)$ where n_{ig} is the equal to the total counts of both alleles for the i-th subject, p_{ig} is the probability of reads belonging to the first allele of the i-th subject, and ϕ_g 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 a read for a particular gene belongs to a particular allele for a particular sample as the 'allelic proportion' for that particular allele and sample. The beta-binomial distribution models proportions that exhibit more variance than what would typically be observed under a binomial distribution (this additional variance is called the overdispersion), and is the typical distribution used for modeling allelic proportions. In this case the overdispersion *parameter* ϕ is inversely related to the actual overdispersion, and $\phi \to \infty$ implies variance no larger than what would be seen in a binomial distribution. n_{1g}, ..., n_{Ig} are assumed to be fixed and known. As the beta-binomial probability mass function has multiple forms and parameterizations, we specify our parametrization as:

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

where B specifies the beta function. Furthermore, let $\mathbf{x}_i$ be the i-th row of the design matrix $\mathbf{X}$ (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 $p_{ig}=[1+\exp(-\mathbf{x}_i^T\boldsymbol{\beta}_g)]^{-1}$, or equivalently $\text{logit}(p_{ig})=\mathbf{x}_i^T\boldsymbol{\beta}_g$, where $\boldsymbol{\beta}_g = (\beta_{1g}, ..., \beta_{Kg})^T$ is a vector of coefficients representing the effect sizes for the predictors in the design matrix. For ML estimation, $\boldsymbol{\beta}_g$ is estimated via ML. Constrained optimization is used for the nuisance parameter ϕ_g with a minimum of 0.01 and a maximum of 5000 for the computational performance and numerical accuracy benchmarks and a minimum of 1 and a maximum of 500 for the estimation performance benchmarks. The user can specify the minimum and maximum as desired. The lower constraint is used for numerical stability as the evaluated probability mass function is degenerate for $\phi_g = 0$ and the upper constraint is used so that genes with no overdispersion do not have infinite estimated values of ϕ_g, Details can be found in the 'Estimating Overdispersion while Coefficients are Fixed' section of the Supplementary Methods section[15]. We found that using a range beyond a minimum of 1 and a maximum of 500 led to only very small, clinically meaningless differences in the coefficients, and we only went beyond this range to demonstrate our package's potential numerical accuracy and computational robustness to larger overdispersion ranges. Standard errors and confidence intervals are calculated based on the asymptotic normal distribution of the ML estimators.

Apeglm shrinks the effects of one chosen covariate at a time, across all genes[12]. It does this by assuming a zero-mode Cauchy prior distribution for the effects of one of the predictors. Due to its heavy tails, a Cauchy prior has a tendency to shrink truly large effect sizes less and in a differential gene expression context was shown to produce estimates with lower error and better ranking by size compared to a Normal prior[12]. For estimating the effect of the j-th predictor in our model, where $j \in \{1, ..., K\}$ is chosen by the user, we have:

$$Y_{ig} \mid \boldsymbol{\beta}_g \overset{ind}{\sim} \text{BetaBin}(n_{ig}, p_{ig}, \phi_g), \text{ for all } 1 \le i \le I, 1 \le g \le G$$

$$p_{ig} = \frac{1}{1+\exp(-\mathbf{x}_i^T\boldsymbol{\beta}_g)}, \text{ for all } 1 \le i \le I, 1 \le g \le G$$

$$\beta_{jg} \overset{iid}{\sim} \text{Cauchy}(0, \gamma_j), \text{ for some } 1 \le j \le K \text{ and all } 1 \le g \le G$$

The scale parameter of the Cauchy prior, γ_j, is estimated by pooling information across genes (see 'Estimating the Scale of the Cauchy Prior' section of the Supplementary Methods[15]). Covariates other than the j-th covariate do not have their effect sizes shrunk, and instead we simply impose a wide and very weakly informative normal prior on their effect sizes (see 'Estimating Coefficients while Overdispersion is Fixed' section of the Supplementary Methods[15]). Apeglm then provides Bayesian shrinkage estimates based on the mode of the resulting log-posterior of $\boldsymbol{\beta}_g$. 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. Posterior standard errors and credible intervals are calculated using a Laplace approximation to the posterior (we will use CIs to abbreviate both confidence intervals and credible intervals moving forward).

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 and/or high-dimensional data sets with beta-binomial GLMs, we wrote fast C++ code for calculating ML and apeglm shrinkage estimates of beta-binomial regression coefficients. We also changed the source code to speed up computation of the posterior standard errors (though such computations were still done in R) and prevent convergence issues. Details can be found in the 'Estimating Coefficients while Overdispersion is Fixed' section and the 'Additional Technical Steps' section of the Supplementary Methods[15]. Finally, while we focus on optimizing performance and evaluating accuracy when a beta-binomial likelihood and Cauchy prior is used, we should note that the *apeglm* package can actually work with any custom likelihood function and any kind of generalized Student's t prior (of which our default prior is a special case with zero mode and 1 degree of freedom).

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[11]. Suppose again that one is interested in the effect sizes of the j-th predictor, $\boldsymbol{\beta}_{j\cdot} = (\beta_{j1}, ..., \beta_{jG})$, where $1 \le j \le K$. Ash takes as input a vector of estimated effects $\hat{\boldsymbol{\beta}}_{j\cdot} = (\hat{\beta}_{j1}, ..., \hat{\beta}_{jG})$ (whether derived by ML estimation or some other method) and corresponding estimated standard errors $\boldsymbol{\sigma}_{\beta j\cdot} = (\sigma_{\beta j1}, ..., \sigma_{\beta jG})$. 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[16]. For all $1 \le g \le G$, it is assumed that $\hat{\beta}_{jg} \mid \beta_{jg} \sim N(\beta_{jg}, \sigma_{\beta_{jg}})$ and that $\beta_{jg} \overset{iid}{\sim} h_j$, where h_j is some unimodal, zero-mode prior distribution. h_j is estimated from the vector of estimates $\boldsymbol{\beta}_{j\cdot}$ using mixtures of uniforms and a point-mass at zero, a choice guided by the fact that any unimodal distribution can be approximated as a mixture of uniforms with arbitrary accuracy[11,17,18]. The posterior is $\beta_{jg} \mid \hat{\beta}_{jg} \sim N(\beta_{jg}, \sigma_{\beta jg}) \times h_j$, and ash provides Empirical Bayes shrinkage estimates using the mean of the posterior. As ash uses the posterior mean, the point estimates will have minimum mean square error over the posterior. Genes with larger standard errors for their ML estimators 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. Finally, it is worth noting that while the original paper and default *ashr* implementation use a normal approximation for the likelihood of estimates, and while we focus on this implementation here, the *ashr* package can alternatively work with a generalized Student's t approximation, that is, $\hat{\beta}_{jg} \mid \beta_{jg} \sim t_{df}(\beta_{jg}, \sigma_{\beta_{jg}})$ (of which the Gaussian is a special case with infinite degrees of freedom). Posterior standard errors

and CIs are calculated directly from the tail probabilities of the estimated posterior.

One of the main differences between apeglm and ash is that of assumptions. Apeglm assumes a fully-specified parametric model for the data likelihood and a specified generalized Student's t prior and maximizes the resulting posterior from the full data. Ash only assumes the likelihood can be approximated with a normal distribution (or at least a generalized Student's t distribution) and that the prior is zero-mode: By default, it uses a Gaussian approximation to the likelihood based on initial parameter estimates and associated standard errors, and the form of a zero-mode prior is determined empirically from the data and estimated in a nonparametric fashion via a mixture of uniforms. Though in large-sample settings the normal approximation will typically be close to the true likelihood (due to the Central Limit Theorem), there could be meaningful differences between the normal approximation and the true likelihood in small-sample settings. In these settings, if apeglm's assumed likelihood is a good match, then it could benefit in avoiding approximation of the likelihood. However, if the specified likelihood is wrong, the likelihood approximation by ash could be more accurate. Moreover, as ash only assumes the prior is zero-mode and estimates it with a universal approximator, it is far more flexible and thus able to adapt to situations where a Cauchy prior (or generalized Student's t prior more broadly) is inappropriate, such as if the true distribution of point estimates are non-symmetric or light-tailed. Furthermore, by only requiring a vector of arbitrary point estimates, ash can also work with estimates not derived by ML estimation, such as quasi-likelihood estimates.

In our software, we take several measures to preventing convergence issues and non-finite estimates even when the true ML estimate is non-finite[15]. However, even with our imposed constraints, our optimization procedure can give ML estimates that are quite large (e.g. >7) if the genes have truly infinite ML estimates (e.g. one allele having zero count for all samples while the other allele has positive counts). In theory, these estimates could adversely affect estimation of the prior by ash and apeglm and lead to suboptimal shrinkage. To investigate the sensitivity of ash and apeglm to these estimates and possible solutions, we explored two possible remedies. First, we considered adding a pseudocount to each sample and for each allele prior to ML estimation. Second, we considered filtering out genes with Truly infinite ML estimates prior to ML estimation.

Datasets and simulations

We compared the three estimation methods using the data set from the allelic expression study by Crowley *et al.*[19,20] 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 following the TruSeq RNA Sample Preparation v2 protocol to generate 100-bp paired-end reads. To assure that the mice all had the same alleles, we chose one genotype to focus on, namely the genotype resulting from the cross with CAST/EiJ and PWK/PhJ. The resulting data set, which we will refer to for the remainder of this paper as the 'mouse data set', had 24 mice, 12 of each sex and 12 of each parent of origin, and each mouse had nearly the same nuclear genetic composition as a result of the cross.

To evaluate the estimators on estimating effect sizes of predictors when the truth is known, we first fit an intercept-only beta-binomial model on each gene for the mouse data set. $\boldsymbol{\phi} = [\phi_g]$ is the vector of ML estimates of the overdispersion parameter from each model. 8 mice were then selected from the data set, 2 of each sex and parent of origin combination. Denote $\mathbf{N}_{I\times G} = [n_{ig}]$ as the matrix of total ASE counts for the 8 mice. Finally, a matrix of counts from one of the alleles $\mathbf{Y}_{I\times G} = [y_{ig}]$ was simulated for a sample size of 4 vs. 4, where Y_{ig} was simulated from $\text{BetaBin}(n_{ig}, p_{ig}, \phi_g)$, $logit(p_{ig}) = \beta_{0g} + \beta_{1g}x$, β_{0g} and β_{1g} were both simulated from a standard normal distribution independently, and x splits the mice into two groups of size four ($x = 1$ if a mouse is in the first group and 0 otherwise). Samples were drawn from the beta-binomial distribution using the emdbook v1.3.12 package[21]. We refer to this simulation throughout the paper as the 'normal simulation', reflecting the distribution of the true effect sizes.

A second simulation was also performed that was similar in setup to the first, but with modifications to the distribution of β_{1g} and ϕ_g. In many studies, the effect sizes of a predictor will be zero for all but a handful of genes. Thus, β_{1g} was simulated from $t_3/10$ (a Student's t-distribution with 3 degrees of freedom scaled by 1/10), which gave effects mostly close to zero, but with moderate and large effects occasionally appearing (Supplementary Figure 1[15]). Furthermore, the distribution of ϕ_g from the mouse data appeared to be a mixture of two distributions: Genes without overdispersion had an obvious point mass at 500 with 70% proportion, and the remaining 30% of the genes could be modelled somewhat well by an exponential distribution with mean $\mu = 179$ (Supplementary Figure 2[15]). To get more over-dispersed allele-specific counts, ϕ_g was simulated from $0.5\text{Exp}(\mu = 89) + 0.5(500)$, a mixture distribution where one component was exponential with a mean of 89 and had 50% proportion, and the other component was a point mass at 500 and had 50% proportion. We refer to this simulation throughout the paper as the 'Student's t simulation', again reflecting the distribution of the true effect sizes. Note that these two simulations assume a data generating process, specifically the same data generating process as our assumed likelihood.

The estimators were then evaluated on real data with the focus on estimating mean, or gene-wide, AI. From the mouse data set, random samples of size 6 were drawn, and this process was repeated 10 times to reduce the impact of sampling variability. 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).

Estimating the intercept in an intercept-only model for each gene is equivalent to estimating overall AI for each gene. As the truth is unknown for real data, and as we don't have enough samples to estimate the truth with high accuracy in an independent held-out sample (there are only 24 mice in the mice dataset overall), we focus on more qualitative comparisons and metrics that don't require the truth, such as the degree of shrinkage and CI width.

Finally, to demonstrate the flexibility afforded by apeglm, we fit a model to the entire mouse data set (all mice) with two binary variables as well as an interaction between them, and investigated the effects of ash and apeglm for shrinking the interaction term. Let Y_{ig} denote the counts for the first allele for sample i and gene g, $1 \leq i \leq 24$, $1 \leq g \leq G$. For all genes and all samples in the mouse data set, we fit the model $Y_{ig} \sim \text{BetaBin}(n_{ig}, p_{ig}, \phi_g)$ where n_{ig} is the total gene expression counts of sample i, gene g, ϕ_g is unknown (estimated), $p_{ig} = \beta_0 + \beta_{1g}SEX_i + \beta_{2g}POE_i + \beta_{3g}SEX_i \times POE$, $SEX_i = I(\text{mouse } i \text{ is female})$ determines the sex effect and $POE_i = I(\text{mouse } i \text{ has strain CAST/EiJ as mother})$ determines the parent-of-origin effect. The interaction effect β_{3g} was shrunk.

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[22]. Briefly, we have $\mathbf{Y}_{100\times5000} = [y_{ig}]$ as our simulated count matrix for one allele with associated total count matrix $\mathbf{N}_{100\times5000} = [n_{ig}]$ where rows are samples and columns are genes, $y_{ig} \sim \text{BetaBin}(n_{ig}, p_g, \phi_g)$, $\phi_g \sim U(0, 1000)$, $p_g \sim N(.5, 0.5^2)$, $n_{ig} \sim \text{NB}(\mu_g, 1/\theta_g)$, and μ_g, θ_g are based on the `airway` data set by Himes *et al.*[23] 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 was $\mathbf{X}_{100\times K} = [\mathbf{1}\ \mathbf{x}_1\ \ldots\ \mathbf{x}_{K-1}]$, where $\mathbf{x}_j$ 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 $\mathbf{X}_{100\times4} = (\mathbf{1}, \mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3) = [x_{ij}]$, where $\mathbf{x}_1 = (1, 0, 1\ldots, 1, 0)^T$ separates the samples into two equally sized groups and $x_{i2}, x_{i3} \sim N(0, 1)$. $\mathbf{x}_1$ is the covariate whose effect size estimates are shrunk.

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 from the simulations and the real data analyses involving intercept-only models , so that all samples could be assumed independent and identically distributed for all genes.

To determine whether sex or parent effects were significant, beta-binomial GLMs were estimated for each gene by ML, 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 the ML estimate to zero. We define a gene as (noticeably) shrunk if shrinkage exceeds 0.1, and substantially or most shrunk if shrinkage is greater than $\max(1, |\hat{\beta}_{\text{ML}}|/4)$. For instance, if an apeglm estimate for a gene is 0.15 closer to zero than the ML estimate, 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[24] were used to determine which estimation method could best find the most important genes (the genes with the largest effect size). For an estimation method, CAT 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 each of the design matrices posited in our computation simulations, computational performance of apeglm estimation was compared between the old and new `apeglm` code. From `apeglm v1.11.2`, 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[22]. Computational performance of ML estimation was also compared between our improved `apeglm` package and the following packages: `aod v1.3.1`[25], `VGAM v1.1.3`[26], `aods3 v0.4.1.1`[27], `gamlss v5.2`[28] and `HRQoL v1.0`[29]. Computational performance was evaluated using the `microbenchmark v1.4.7` package[30] for estimation of a single gene (we used the *`microbenchmark`* function and set *`times=20L`*) 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, ML estimation paired with optimal filtering criteria was also assessed via CAT. 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 ML estimation 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. For each rule and threshold, the ML estimates were calculated and concordance among the top 50, 100, 200, 300, 400 and 500 genes were averaged. We will refer to the rule and threshold that had the best concordance as the 'optimal filtering rule'.

Results

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 ML with pseudocounts successfully shrunk erroneously large estimates and had improved performance over maximum likelihood without pseudocounts. Ash had lower estimation error than ML, but estimation error was still higher than that of apeglm and ML with pseudocounts, and CAT performance was worse than ML (see Table 1 and Figure 1).

Table 1. Performance Metrics for Normal Simulation. ML: Maximum Likelihood, apeglm: Approximate Posterior Estimation of Generalized Linear Model Coefficients, ash: Adaptive Shrinkage.

Performance Metric	ML	Apeglm	Ash	ML+Pseudo	Apeglm+Pseudo	Ash+Pseudo
MAE	0.208	0.187	0.196	0.183	0.196	0.201
MAE (apeglm-shrunk genes)	0.626	0.501	0.552	0.487	0.561	0.577
MAE (ash-shrunk genes)	0.557	0.447	0.496	0.407	0.471	0.507
MAE (counts<Q1)	0.482	0.399	0.413	0.384	0.419	0.411
MAE (counts>Q1, |MLE|>2)	0.374	0.31	0.424	NA	NA	NA
Coverage Probability for 95% CI	0.949	0.94	0.924	0.935	0.911	0.9
Average Interval Width for 95% CI	1.109	0.862	0.813	0.795	0.751	0.736

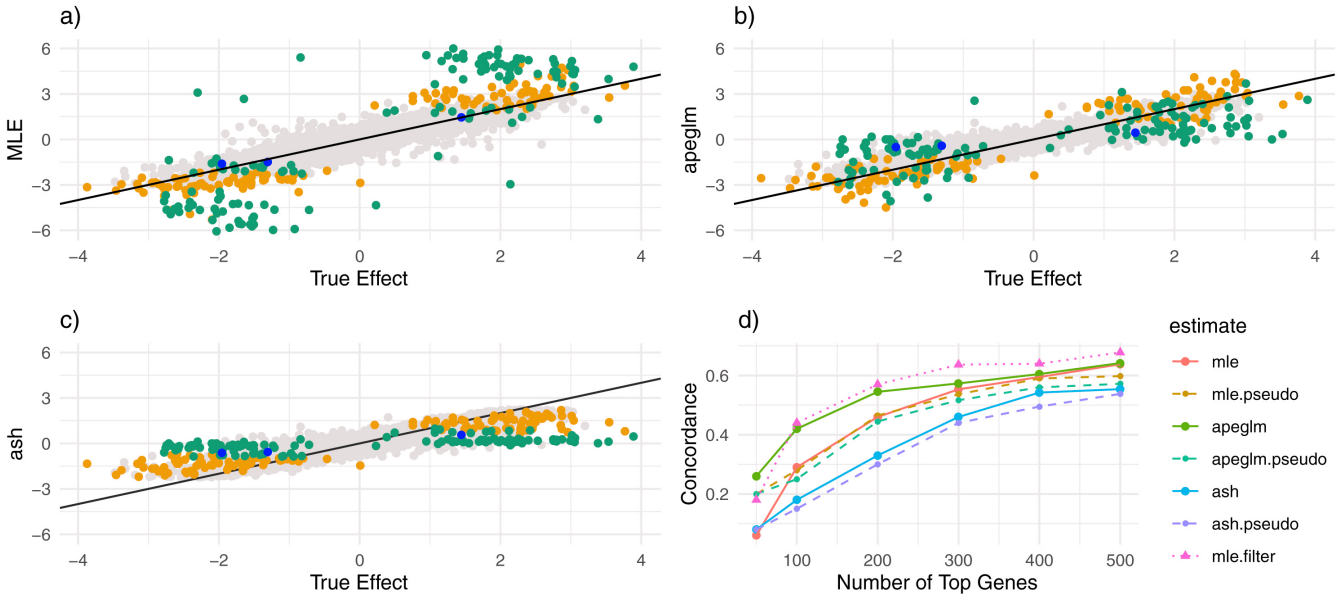

Figure 1. Estimate vs. truth and CAT Plots for normal simulation. a) Estimate vs. truth plot for ML estimation. Blue points represent genes substantially shrunk by apeglm only, orange points represent genes substantially shrunk by ash only and green points represent genes substantially shrunk by both ash and apeglm. **b**) estimate vs. truth plots for apeglm. **c**) estimate vs. truth plots for ash. **d**) CAT plot for the three methods with and without pseudocounts as well as for ML estimation after filtering. CAT: Concordance at the Top, ML estimation: Maximum Likelihood Estimation, apeglm: Approximate Posterior Estimation of Generalized Linear Model Coefficients, ash: Adaptive Shrinkage.

Comparing apeglm and ash to ML first, we note that all three estimators have similar overall mean absolute error 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 mean absolute error, we also calculated mean absolute error 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. Shrinkage is defined as movement from ML to zero, and a gene is considered '(noticeably) shrunk' by apeglm or ash if the apeglm or ash shrinkage exceeds 0.1 and 'substantially shrunk' if shrinkage exceeds max $(1,|\hat{\beta}_{\mathrm{ML}}|/4)$. Among genes that were noticeably shrunk by apeglm, apeglm decreased the mean absolute error by 20%, and among genes that were shrunk by ash, ash decreased the mean absolute error by 20.8%. Moreover, among genes whose total counts were less than the 1st quartile (and where shrinkage would be the most apparent), mean absolute error decreased by 17.2% and 14.3% for apeglm and ash respectively. We can also see that apeglm had lower mean absolute error than ash, both overall and across these three categories (apeglm-shrunk, ash-shrunk and low-count genes). As shrinkage can be considered a complex function of observed data statistics, stratifying by shrinkage does not use the true data-generating distribution and provides a fair comparison.

From Figure 1a–c, it can be seen that apeglm shrunk most ML estimates that were inflated (i.e. much larger in magnitude than the corresponding true effect sizes), and mostly left truly large effects alone. Ash also shrunk ML estimates that were inflated, including some estimates less severely inflated that were missed by apeglm. However, ash also had a tendency to shrink more excessively, and that quite a few genes with truly large effects were shrunk to zero. This agrees with Supplementary Table 1[15], which compares quantiles of shrinkage between apeglm and ash and illustrates a clear upward shift of shrinkage for ash. Moreover, from Figure 2a–b, we can see that both apeglm and ash exhibited more shrinkage for genes with low counts and severely shrunk genes with low counts and large estimates. However, ash also severely shrunk large ML estimates for genes with larger counts, even though these genes were more likely to have truly large effects. From Table 1, we see that among genes with high counts and large ML estimates, ash performed worse than ML estimation. All of this suggests that ash might be over-shrinking (that is, shrinking too much) for this data-generating distribution.

Adding a pseudocount to each sample and for each allele prior to ML estimation greatly improved accuracy of the ML estimates. Overall, the mean absolute error for adding pseudocounts followed by maximum likelihood was lower than ash and slightly lower than apeglm. However, adding pseudocounts prior to apeglm or ash actually made performance worse. From Figure 2c–d, it can be seen that adding pseudocounts alone lead to noticeably smaller MLEs, and this shrinkage was substantial for genes with low counts.

Apeglm greatly outperformed ML estimation in determining the set of genes with the largest effect sizes, where concordance

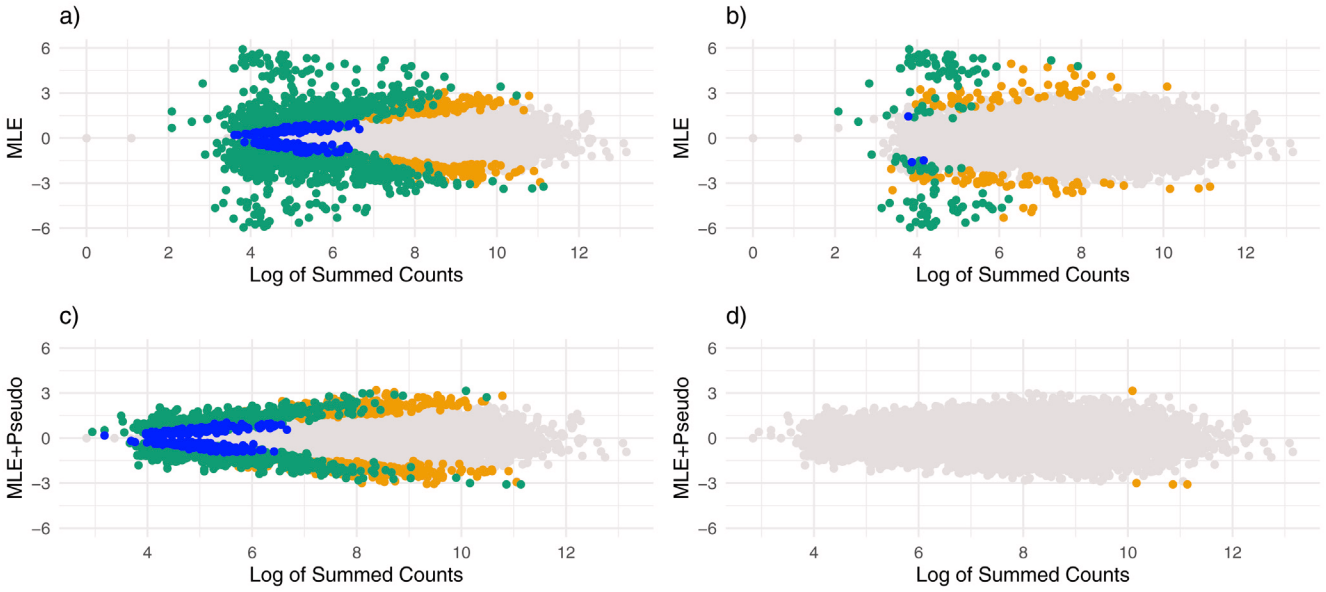

Figure 2. MA Plots for normal simulation. Estimates of effect size vs. log of summed counts. Each point represents a gene and the x-axis gives the logarithm of the gene's summed counts. For the top plots (**a** and **b**) the y-axis gives the ML estimates without using pseudocounts, and for the bottom plots (**c** and **d**) the y-axis gives the ML estimates after adding pseudocounts. Points are colored by whether they were noticeably shrunk on the left (**a** and **c**), and whether they were severely shrunk on the right (**b** and **d**). Blue points represent genes noticeably or substantially shrunk by apeglm only, orange points represent genes noticeably or substantially shrunk by ash only and green points represent genes noticeably or substantially shrunk by both ash and apeglm.

at the top (CAT) was higher regardless of the number of genes being considered (Figure 1d). Though adding pseudocounts improved CAT performance of the ML estimates, the improvement was modest and did not rival performance of the apeglm estimates. Ash performed worse than both apeglm and ML estimation, perhaps due in part to the potential over-shrinking discussed in the previous paragraph. Like for mean absolute error, pseudocounts improved CAT performance for ML estimation but made apeglm and ash CAT performance worse.

In addition to adding pseudocounts, we also attempted to filter out the 114 genes (out of about 10,000) with truly infinite ML estimates. The resulting changes did not alter conclusions: apeglm still had the best CAT performance, and ash still had the worst (Supplementary Figure 3)[15]. Finally, we tried filtering out genes with small counts, with the cutoff based on that which improved CAT performance the most (dubbed 'mle.filter' in the CAT plot above). This led to the ML estimates having slightly better CAT performance than other methods on average. Thus, for this simulation, it was possible to beat other methods using filtering alone (by a small margin), provided we used knowledge of the underlying data-generating distribution to decide the filtering rule.

The coverage probability for intervals estimated by apeglm, ash and ML with pseudocounts were 1%, 2% and 1.5% lower than nominal, respectively, but average interval width was also 22.3%, 26.3% and 28.3% narrower than the likelihood-based intervals without pseudocounts (Table 1). Adding pseudocounts prior to using apeglm and ash led to lower coverage and interval width was not that different from those obtained by using pseudocounts and maximum likelihood.

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 both apeglm and ash gave marked improvement over the ML estimates, while the improvement from adding pseudocounts was only slight (see Table 2 and Figure 3).

Apeglm improved mean absolute error by 52.9% among all genes, and by 66.1% among noticeably shrunk genes specifically (Table 2). Ash improved mean absolute error by 56.3% among all genes and by 68.1% among noticeably shrunk genes specifically. Adding pseudocounts slightly improved mean absolute error for the ML estimates, but estimation error was still not nearly as low as apeglm and ash, and again combining pseudocounts with ash and apeglm did not lead to better performance than using ash or apeglm without pseudocounts.

Shrinkage patterns between ash and apeglm were much more similar here than in the normal simulation. Figure 3a–c show that both apeglm and ash shrunk inflated ML estimates similarly while leaving truly large effects mostly unchanged, though there a few inflated estimates that were missed by apeglm but shrunk by ash. Supplementary Table 2[15] compares quantiles of shrinkage between apeglm and ash, and we can see that for this simulation the difference in shrinkage quantiles between apeglm and ash is quite small (though a paired Wilcoxon signed-rank test still concluded that ash had greater shrinkage on average with $p<0.001$). Furthermore, both ash and apeglm exhibited shrinkage for effects across the dynamic range of summed counts (Figure 4a–b). This is not too surprising, as due to the increased overdispersion, there were more effects that were overestimated by ML, even among genes with large counts.

CAT performance was much better for apeglm and ash than for the ML estimates with and without pseudocounts, regardless of the number of top genes in question, with ash performing better than apeglm (particularly for the top 50 genes). Adding pseudocounts only slightly improved CAT performance for the ML estimates and did not improve performance for apeglm and ash. Moreover, filtering out genes with small counts (with the cutoff based on that which improved CAT performance the most) did not lead to nearly as good CAT performance for apeglm and ash. Thus, for this simulation, it was not possible to beat apeglm using filtering alone, even when using the true data-generating distribution to decide the filtering rule.

Both apeglm and ash had half the average interval width compared to ML despite also having higher coverage rates. Adding a pseudocount to each allele and sample followed by ML estimation did not lead to the same improvement in interval

Table 2. Performance metrics for Student's t Simulation. MAE: Mean Absolute Error, MLE: Maximum Likelihood, apeglm: Approximate Posterior Estimation of Generalized Linear Model Coefficients, ash: Adaptive Shrinkage.

Performance Metric	ML	Apeglm	Ash	MLE+Pseudo	Apeglm+Pseudo	Ash+Pseudo
MAE	0.208	0.098	0.091	0.182	0.096	0.09
MAE (apeglm-shrunk genes)	0.375	0.127	0.114	0.318	0.125	0.113
MAE (ash-shrunk genes)	0.361	0.129	0.115	0.308	0.127	0.114
MAE (counts<Q1)	0.364	0.112	0.104	0.272	0.108	0.103
Coverage probability for 95% CI	0.921	0.938	0.942	0.924	0.936	0.942
Average Interval Width for 95% CI	0.975	0.462	0.471	0.864	0.454	0.454

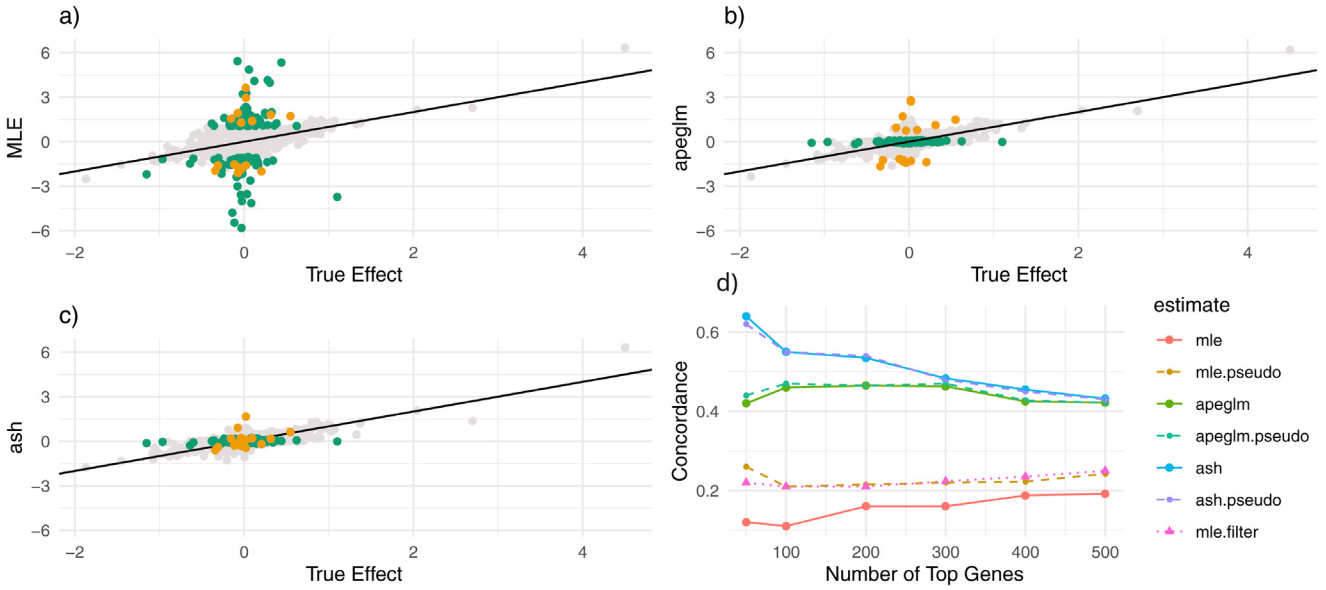

Figure 3. Estimate vs. truth and CAT Plots for Student's t Simulation. a) estimate vs. truth plot for ML estimation. Orange points represent genes substantially shrunk by ash only and green points represent genes substantially shrunk by both ash and apeglm. All genes substantially shrunk by apeglm were shrunk by practically the same amount or more by ash. **b**) estimate vs. truth plots for apeglm. **c**) estimate vs. truth plots for ash. **d**) CAT plot for the three methods without and with pseudocounts. CAT: Concordance at the Top, ML: Maximum Likelihood, apeglm: Approximate Posterior Estimation of Generalized Linear Model Coefficients, ash: Adaptive Shrinkage.

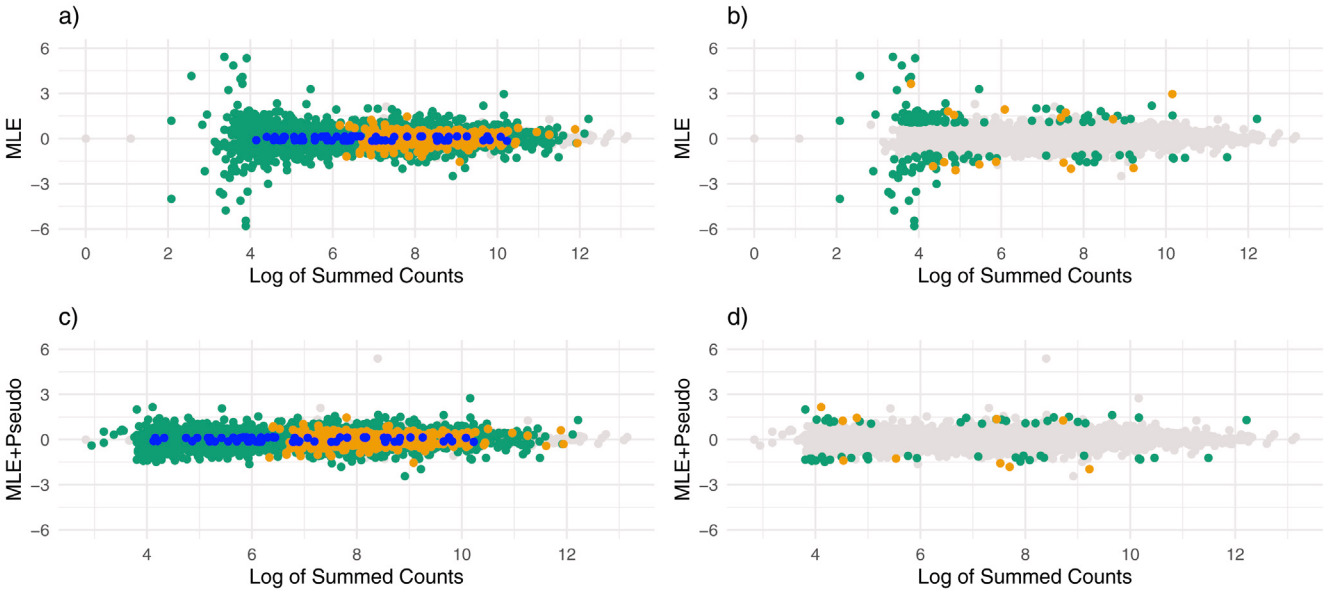

Figure 4. MA Plots for Student's t Simulation. Estimates of effect size over log of summed counts for the Student's t simulation. Each point represents a gene and the x-axis gives the logarithm of the gene's summed counts. For the top plots (**a** and **b**) the y-axis gives the ML estimates without using pseudocounts, and for the bottom plots (**c** and **d**), the y-axis gives the ML estimates after adding pseudocounts. Points are colored by whether there were noticeably shrunk on the left (**a** and **c**), and whether there were severely shrunk on the right (**b** and **d**). Blue points represent genes noticeably or substantially shrunk by apeglm only, orange points represent genes noticeably or substantially shrunk by ash only and green points represent genes noticeably or substantially shrunk by both ash and apeglm.

coverage or width and combining pseudocounts with apeglm or ash did not yield any improvement.

Filtering out genes with truly infinite ML estimates did not change prior estimation of apeglm or ash, as there were only 20 such genes for this simulation.

Sampling from the mouse dataset

To evaluate performance on real data, we first took 10 random subsamples of 6 mice from the mouse data set with replacement and calculated different evaluation metrics for each random subsample. These results are summarized in Table 3 and Figure 5.

Table 3 and Figure 5a are based on all subsamples. For example, with 10 subsamples and ~10,000 genes, the first row of Table 1 is the 75th percentile of the ~100,000 shrinkage scores calculated across all 10 iterations, and each colored histogram in Figure 1a is based on the top 50 genes for each of 10 subsamples or 500 (possibly overlapping) summed counts overall.

Table 3. Summaries of Evaluation Metrics Across the Subsamples. For each gene, we define the "shrinkage score" as the movement from the ML estimate to zero. ML: Maximum Likelihood, apeglm: Approximate Posterior Estimation of Generalized Linear Model Coefficients, ash: Adaptive Shrinkage.

Evaluation Metric	ML	Apeglm	Ash
75th Percentile of Shrinkage Scores	NA	0.022	0.047
90th Percentile of Shrinkage Scores	NA	0.096	0.147
97.5th Percentile of Shrinkage Scores	NA	0.342	0.521
99th Percentile of Shrinkage Scores	NA	0.568	1.141
Median Summed Counts of Top 50 Genes	63	222	248
Average Interval Width for 95% CI	0.567	0.43	0.395

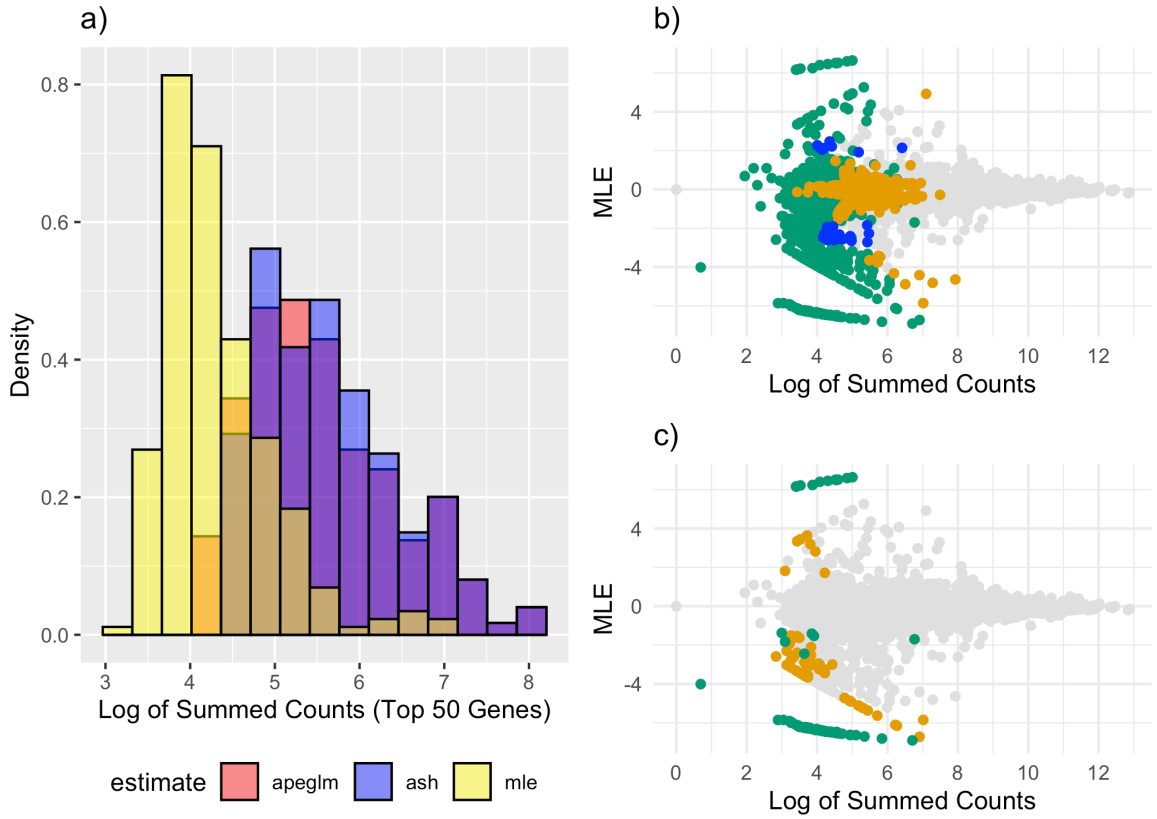

Figure 5. Distribution of Summed Counts for random subsamples. a) Overlapping histograms of log-summed counts for the top 50 genes according to ML (yellow), apeglm (red) and ash (blue), across all 10 subsamples. **b**) MA plot (ML estimates vs. log-summed counts) for one random subsample. Blue points represent genes noticeably shrunk by apeglm only, orange points represent genes noticeably shrunk by ash only and green points represent genes noticeably shrunk by both ash and apeglm. **c**) Same as (**b**) except now points are only colored by whether there was substantial shrinkage, as opposed to whether there was noticeable shrinkage. ML: Maximum Likelihood, apeglm: Approximate Posterior Estimation of Generalized Linear Model Coefficients, ash: Adaptive Shrinkage.

This was done to remove the effect of sampling variability on our results.

From Table 3, we see that percentiles of shrinkage scores were higher for ash than apeglm, particularly across the highest percentiles, indicating that ash was exhibiting more frequent and more severe shrinkage as in the normal simulation. It is also interesting to compare apeglm and ash in which genes they tend to shrink. For example, all of the genes shrunk by apeglm had low counts and/or very large MLEs, and this characterized many of the genes shrunk by ash as well (Figure 5b–c). However, some of the genes shrunk by ash also had both larger counts and smaller ML estimates.

From Table 3 and Figure 5a, we can see that both apeglm and ash had higher counts among their top ranked genes than the top ranked genes by ML. For comparison, the 1st quartile of summed counts of all genes was 507, and thus the distribution of counts for the genes ranked highest by apeglm and ash were more similar to the distribution of counts among all genes. Compared to ML intervals, intervals were 26.1% narrower for apeglm and 32.2% narrower for ash.

We also fit a model with two binary variables and an interaction to all 24 mice, and used ash and apeglm to shrink the interaction term (see Table 4 and Figure 6). Unlike the real data intercept estimates and the estimates from the simulations, the distribution of ML estimates for the interaction effect had a positive mode and skew (sample skewness = 5.21). Apeglm assumes a symmetric distribution for the true effect sizes about zero and ash assumes at least a mode of zero, and it is not clear how much performance for apeglm and ash would degrade if these assumptions were violated. The ML estimates also had larger standard errors than in the simulations and real data intercept models, perhaps because there were three variables in our model instead of one and six mice per sex-POE group. Perhaps relatedly, ash estimates had a few notable differences than from previous analyses. For example, here average intercept width for ash was only 7.8% as large as that of likelihood-based intervals and only 12.8% as large as apeglm intervals. Filtering out infinite ML genes did not substantially change results (see Supplementary Table 3).

Table 4. Summaries of Evaluation Metrics for the Interaction Model. ML: Maximum Likelihood, apeglm: Approximate Posterior Estimation of Generalized Linear Model Coefficients, ash: Adaptive Shrinkage.

Evaluation Metric	ML	Apeglm	Ash
75th Percentile of Shrinkage Scores	NA	0.023	0.188
90th Percentile of Shrinkage Scores	NA	0.134	0.427
97.5th Percentile of Shrinkage Scores	NA	0.697	1.213
99th Percentile of Shrinkage Scores	NA	1.693	2.334
Median Summed Counts of Top 50 Genes	2582	3234	5616
Average Interval Width for 95% CI	1.275	0.795	0.111

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 7 and Figure 8).

Estimation times per gene for ML estimation was substantially faster for `apeglm` than all other packages (Figure 7). The next best package, `aods3`, took 3 to 12 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. Though our previous analysis showed that apeglm estimators often have higher accuracy than that of ML estimators, there are still reasons why one may prefer standard likelihood-based beta-binomial GLMs, such as if the sample size is large or if simplicity or unbiasedness is desired. Moreover, many shrinkage estimation packages like `ash` require a vector of initial ML estimates and standard errors, and one cannot use these methods without a ML-fitting package. It is also worth noting that apeglm estimation is practically as fast as ML estimation in the `apeglm` package, and thus for comparing apeglm estimation speed to ML estimation speed of the other packages, our package is still substantially faster.

For estimating all genes in the simulation via ML, `apeglm` took 24 seconds for two groups and added only 1–2 seconds of computational time for every group added (Figure 8a). 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).

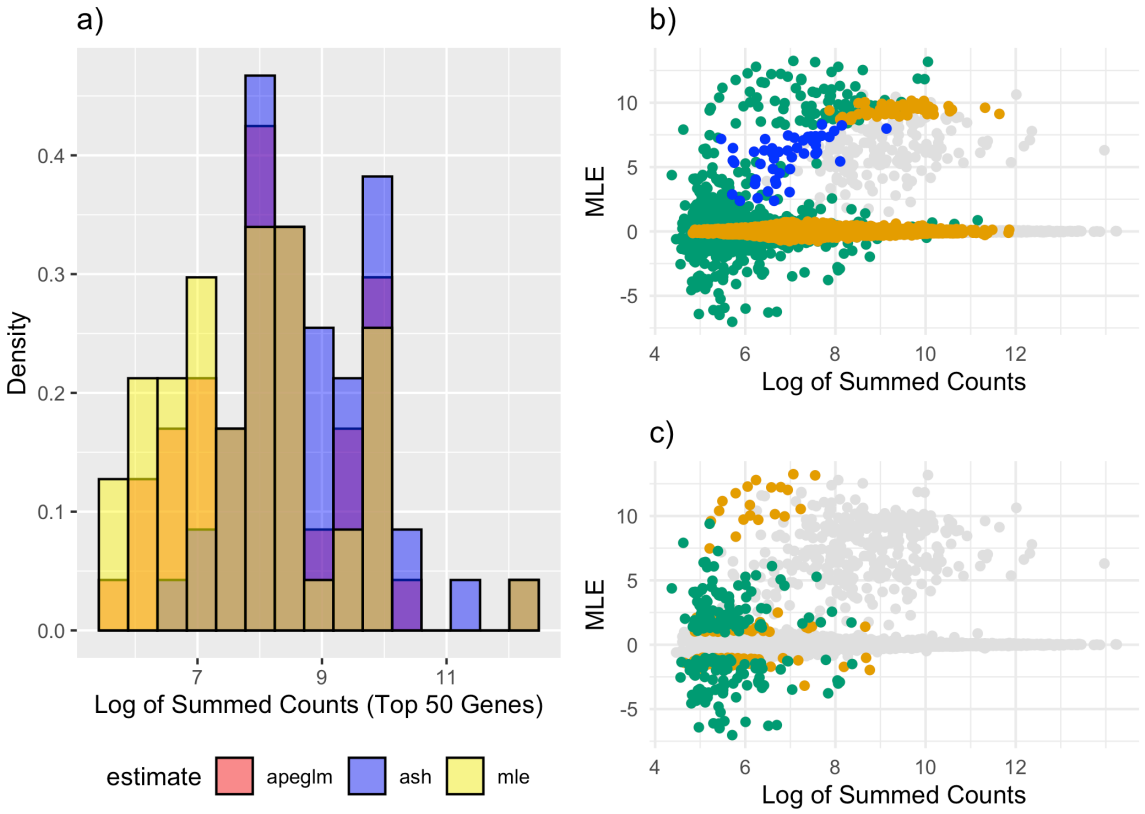

Figure 6. Distribution of Summed Counts for Interaction Model. a) Overlapping histograms of log-summed counts for the top 50 genes (genes with the largest interaction effect sizes) according to ML (yellow), apeglm (red) and ash (blue). **b)** MA plot (ML estimates vs. log-summed counts) for one random subsample. Blue points represent genes noticeably shrunk by apeglm only, orange points represent genes noticeably shrunk by ash only and green points represent genes noticeably shrunk by both ash and apeglm. **c)** Same as (b) except now points are only colored by whether there was substantial shrinkage, as opposed to whether there was noticeable shrinkage. ML: Maximum Likelihood, apeglm: Approximate Posterior Estimation of Generalized Linear Model Coefficients, ash: Adaptive Shrinkage.

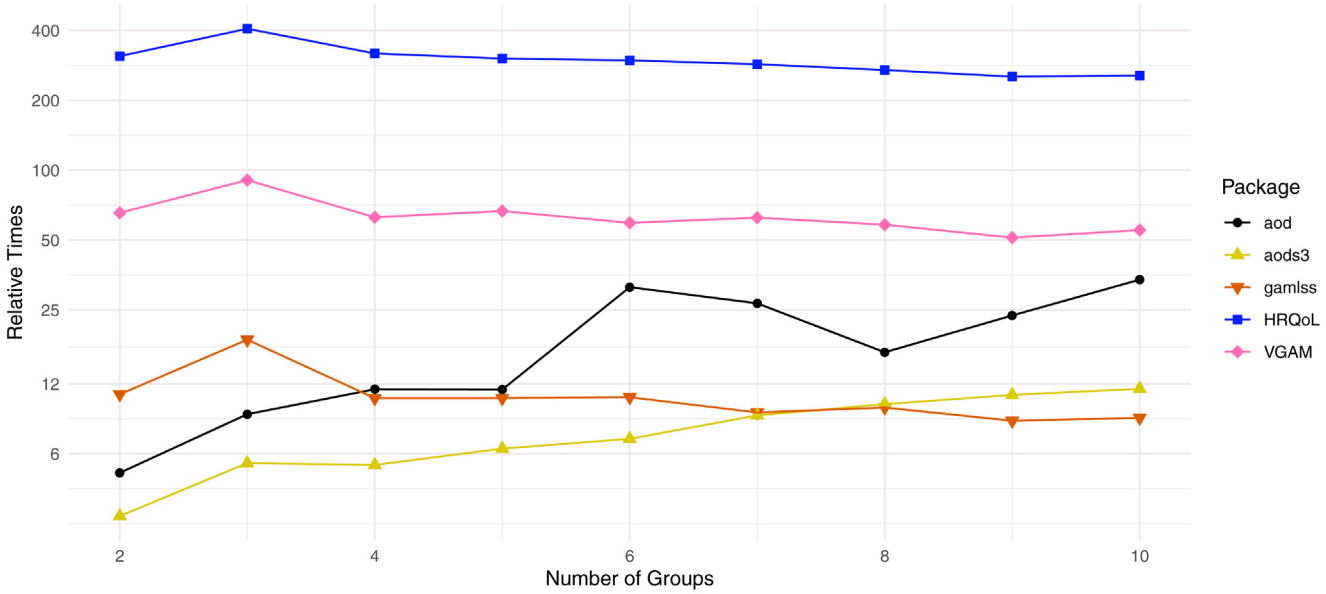

Figure 7. Comparisons in estimation time for one gene. Relative Times are defined as the fold changes in computation time relative to the *apeglm* package for the same number of groups. For instance, *aods3* takes about 6 times longer than *apeglm* to fit a beta-binomial GLM to one gene with two groups, and about 12 times longer than *apeglm* to fit such a model with three groups.

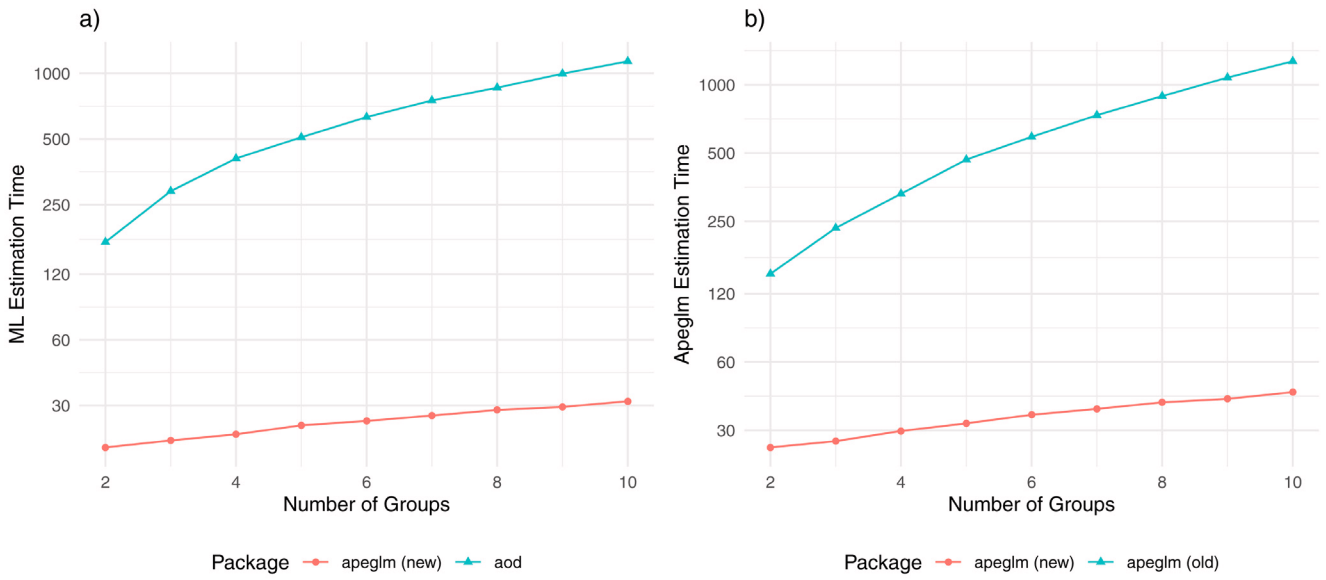

Figure 8. Comparisons in estimation time for all genes. a) computational time of ML estimation (in seconds) for the `apeglm` and `aod` packages by the number of groups (covariates). **b**) computational time of apeglm estimation for the new and old `apeglm` packages by number of groups (covariates).

In addition to looking at computational performance, we also compared the numerical accuracy and reliability of our package to the packages in Figure 7 using the same simulation. `Gamlss`, `aods3` and `HRQoL` failed to converge and produced errors with relatively high frequency and estimates of `aod` and `VGAM` tended to have lower log-likelihoods than those of `apeglm`. Moreover, while `aod` did successfully converge for this computational simulation, it failed to converge (gave infinite estimates associated with log-likelihoods of negative infinity) when extreme overdispersion and smaller sample sizes were introduced into the simulation. On the other hand, our package converged in all evaluations, including across all simulations and real data analyses discussed here and in the Supplementary Methods[15]. We imposed wide artificial caps on the sample-specific probabilities to prevent our package from producing errors and giving non-finite solutions even when the dataset of interest contains genes that exhibit all counts of zero for one allele across all samples, while positive counts for the other allele. For further details on numerical accuracy, see 'Evaluating Numerical Accuracy' section of the Supplementary Methods[13].

Discussion

Here the performance of four estimators was compared across two simulations and one real dataset of allele specific expression in mice. The performance of the point estimates of apeglm was robust and consistent: across both simulations, apeglm had lower mean absolute error and higher concordance at the top than ML and had either the best or second-best estimation and CAT performance. Ash performed universally better than ML for the Student's t simulation, but for the normal simulation its CAT performance was worse and its MAE was lower among genes with high counts. Conversely, use of pseudocounts and filtering performed comparatively similarly to apeglm in the normal simulation, but performed much worse than both apeglm and ash in the Student's t simulation.

Apeglm and ash 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 by both methods as well, showing that ash and apeglm will shrink large-count genes if there is high uncertainty in the estimates. Ash consistently shrunk genes more than apeglm: the quantiles of shrinkage scores for ash were always higher than the corresponding quantiles for apeglm, and genes with high counts were more likely to be shrunk by ash.

No method gave confidence or credible intervals with the highest coverage rates for all scenarios. However, across both simulations, 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 much smaller than that of ML. This suggests that interval estimates from apeglm and ash could have similar utility to and be advantageous over those by ML. For future research, it would be beneficial to evaluate the accuracy of Bayesian or frequentist hypotheses tests based on the estimates or posterior distribution of apeglm and ash using metrics such as

type I and type II error. The method of Leòn-Novelo *et al*. 2018[7] 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 the fastest for fitting many GLMs in sequence, e.g. across many genes or variant locations. It also had the best convergence in practice on datasets we evaluated: solutions had the highest likelihood on average, and it was one of the only packages that never produced errors or failed to converge, even in the face of extreme data dispersion and large ML estimates. The ML estimates were also indirectly capped to avoid non-finite solutions. For typical real datasets of reasonable size, it is common to have at least a few genes that exhibit counts of zero for one allele for all samples. Therefore, not only is `apeglm` substantially faster than the other packages, but it is also numerically accurate and reliable. Thus, the new `apeglm` package is useful for quick and reliable analyses of AI even for researchers who wish to only use likelihood-based estimators. Currently, only coefficient estimates are calculated in C++, and even better computational performance would be achieved if overdispersion and posterior standard error calculations were integrated into C++ as well. We are not aware of any other R packages made at the time of this article's publication that utilize fast 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 `fastglm`[31], 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 ML estimation, 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 Leòn-Novelo *et al*. 2018, may lead to better inference as well. Adjusting for read mapping biases and ambiguities (Leòn-Novelo *et al*. 2014[5]; Leòn-Novelo *et al*. 2018[7]; Raghupathy *et al*. 2018[3]) 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-binomial[32] to Poisson-lognormal models[8].

Data availability

Underlying data

Zenodo: RNA-seq Dataset from Crowley *et al*. 2015. http://doi.org/10.5281/zenodo.3404689[20].

This project contains the following underlying data:

- fullGeccoRnaDump.csv

This file contains the Crowley *et al.* mouse dataset which was was obtained from http://csbio.unc.edu/gecco/data/fullGeccoRnaDump.csv.gz[19,33]. 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.

Extended data

Zenodo: Supplementary Material for Zitovsky and Love 2019. http://www.doi.org/10.5281/zenodo.4033010[15].

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 and reports results on its numerical accuracy)
- Supplementary Figures and Tables.pdf (Contains supplementary figures 1–3 and supplementary tables 1–3. These figures and tables were referenced and described in the main body of the article)

Software availability

Zenodo: Apeglm v1.11.2 Source Code. http://www.doi.org/10.5281/zenodo.4033035[34]. This repository contains the source code for the version of the `apeglm` package used in this paper.

Zenodo: Source Code for Zitovsky and Love 2019. http://www.doi.org/10.5281/zenodo.4033007[35]. 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 or .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.

`apeglm` is available as part of the Bioconductor project[36] at http://bioconductor.org/packages/apeglm. The vignette[22] 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. We thank the reviewers for their helpful comments and suggestions on the original manuscript.

References

1. Castel SE, Levy-Moonshine A, Mohammadi P, *et al.*: **Tools and best practices for data processing in allelic expression analysis.** *Genome Biol.* 2015; **16**(1): 195.
PubMed Abstract | Publisher Full Text | Free Full Text

2. Sun W, Hu Y: **Mapping of Expression Quantitative Trait Loci Using RNA-seq Data.** In: Somnath Datta and Dan Nettleton, editors, *Statistical Analysis of Next Gen- eration Sequencing Data*. Springer International Publishing, Switzerland. 2014; 145–168.
Publisher Full Text

3. Raghupathy N, Choi K, Vincent MJ, *et al.*: **Hierarchical analysis of RNA-seq reads improves the accuracy of allele-specific expression.** *Bioinformatics.* 2018; **34**(13): 2177–84.
PubMed Abstract | Publisher Full Text | Free Full Text

4. Turro E, Su SY, Gonçalves Â, *et al.*: **Haplotype and isoform specific expression estimation using multi-mapping RNA-seq reads.** *Genome Biol.* 2011; **12**(2): R13.
PubMed Abstract | Publisher Full Text | Free Full Text

5. León-Novelo LG, McIntyre LM, Fear JM, *et al.*: **A flexible Bayesian method for detecting allelic imbalance in RNA-seq data.** *BMC Genomics.* 2014; **15**(1): 920.
PubMed Abstract | Publisher Full Text | Free Full Text

6. Skelly DA, Johansson M, Madeoy J, *et al.*: **A powerful and flexible statistical framework for testing hypotheses of allele-specific gene expression from RNA-seq data.** *Genome Res.* 2011; **21**(10): 1728–37.
PubMed Abstract | Publisher Full Text | Free Full Text

7. León-Novelo LG, Gerken AR, Graze RM, *et al.*: **Direct Testing for Allele-Specific Expression Differences Between Conditions.** *G3 (Bethesda).* 2018; **8**(2): 447–460.
PubMed Abstract | Publisher Full Text | Free Full Text

8. Alvarez-Castro I, Niemi J: **Fully Bayesian Analysis of Allele-Specific RNA-seq Data**. *Math Biosci Eng.* 2019; **16**(6): 7751–770.
PubMed Abstract | Publisher Full Text

9. Love MI, Huber W, Anders S: **Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2.** *Genome Biol.* 2014; **15**(12): 550.
PubMed Abstract | Publisher Full Text | Free Full Text

10. Landau W, Niemi J, Nettleton D: **Fully Bayesian analysis of RNA-seq counts for the detection of gene expression heterosis.** *J Am Stat Assoc.* 2018; **114**(526): 610–621.
PubMed Abstract | Publisher Full Text | Free Full Text

11. Stephens M: **False discovery rates: a new deal.** *Biostatistics.* 2017; **18**(2): 275–94.
PubMed Abstract | Publisher Full Text | Free Full Text

12. Zhu A, Ibrahim JG, Love MI: **Heavy-tailed prior distributions for sequence count data: removing the noise and preserving large differences.** *Bioinformatics.* 2018; **35**(12): 2084–2092.
PubMed Abstract | Publisher Full Text | Free Full Text

13. Turro E, Astle WJ, Tavaré S: **Flexible analysis of RNA-seq data using mixed effects models.** *Bioinformatics.* 2014; **30**(2): 180–188.
PubMed Abstract | Publisher Full Text

14. R Core Team: **R: A Language and Environment for Statistical Computing.** R Foundation for Statistical Computing, Vienna, Austria. 2018.
Reference Source

15. Zitovsky JP, Love MI: **Supplementary Material for Zitovsky and Love 2019.** (Version v1.2). *Zenodo.* 2020.
http://www.doi.org/10.5281/zenodo.4033010

16. Lu M, Stephens M: **Empirical Bayes Estimation of Normal Means, Accounting for Uncertainty in Estimated Standard Errors.** 2019; arXiv:1901.10679.
Reference Source

17. Khintchine AY: **On Unimodal Distributions.** *Izv Nauchno-Issled Inst Mat Mekh Tomsk Gos Univ.* 1938; **2**: 1–7.

18. Shepp L: **Symmetric Random Walk.** *Transactions of the American Mathematical Society.* 1962; **104**(1): 144–153.
Publisher Full Text

19. Crowley JJ, Zhabotynsky V, Sun W, *et al.*: **Analyses of allele-specific gene expression in highly divergent mouse crosses identifies pervasive allelic imbalance.** *Nat Genet.* 2015; **47**(4): 353–360.
PubMed Abstract | Publisher Full Text | Free Full Text

20. Crowley JJ, Zitovsky JP, Love MI: **RNA-seq Dataset from Crowley et. al. 2015.** (Version v1.0). *Zenodo.* 2019.
http://www.doi.org/10.5281/zenodo.3404689

21. Bolker B: **emdbook: Ecological Models and Data in R.** In: R package version 1.3.12. 2020.
Reference Source

22. Zhu A, Ibrahim JG, Love MI: **Effect Size Estimation with Apeglm.** Bioconductor. 2020.
Reference Source

23. Himes BE, Jiang X, Wagner P, *et al.*: **RNA-Seq transcriptome profiling identifies CRISPLD2 as a glucocorticoid responsive gene that modulates cytokine function in airway smooth muscle cells.** *PLoS One.* 2014; **9**(6): e99625.
PubMed Abstract | Publisher Full Text | Free Full Text

24. Irizarry RA, Warren D, Spencer F, *et al.*: **Multiple-laboratory comparison of microarray platforms.** *Nat Methods.* 2005; **2**(5): 345–350.
PubMed Abstract | Publisher Full Text

25. Lesnoff M, Lancelot R: **aod: Analysis of Overdispersed Data.** R package version 1.3.1. 2019.
Reference Source

26. Yee TW: **Vector Generalized Linear and Additive Models: With an Implementation in R**. 2019.
Publisher Full Text

27. Lesnoff M, Lancelot R: **aods3: Analysis of Overdispersed Data Using S3 Methods.** R package version 0.4-1.1. 2018.
Reference Source

28. Rigby RA, Stasinopoulos DM: **Generalized Additive Models for Location, Scale and Shape.** *J R Stat Soc C-Appl.* 2005; **54**(3): 507–54.
Publisher Full Text

29. Dae-Jin L, Najera-Zuloaga J, Arostegui I: **HRQoL: Health Related Quality of Life Analysis.** R package version 1.0. 2017.
Reference Source

30. Mersmann O: **microbenchmark: Accurate Timing Functions.** R package version 1.4-7. 2019.
Reference Source

31. Huling J: **fastglm: Fast and Stable Fitting of Generalized Linear Models using RcppEigen.** R package version 0.0.1. 2019.
Reference Source

32. McVicker G, van de Geijn B, Degner JF, *et al.*: **Identification of genetic variants that affect histone modifications in human cells.** *Science.* 2013; **342**(6159): 747–749.
PubMed Abstract | Publisher Full Text | Free Full Text

33. Crowley JJ, *et al.*: **Gene Expression in the Collaborative Cross**. (and Others). 2015. [Data set].

34. Zhu A, Zitovsky J, Ibrahim J, *et al.*: **Apeglm v1.11.2 Source Code (Version v1.2)**. *Zenodo.* 2020.
http://www.doi.org/10.5281/zenodo.4033035

35. Zitovsky JP, Love MI: **Source Code for Zitovsky and Love 2019 (Version v1.5)**. *Zenodo.* 2020.
http://www.doi.org/10.5281/zenodo.4033007

36. Huber W, Carey VJ, Gentleman R, *et al.*: **Orchestrating high-throughput genomic analysis with Bioconductor.** *Nat Methods.* 2015; **12**(2): 115–121.
PubMed Abstract | Publisher Full Text | Free Full Text

Open Peer Review

Current Peer Review Status: ? ? ?

Version 1

Reviewer Report 10 February 2020

https://doi.org/10.5256/f1000research.23018.r57280

?

Ernest Turro
[1] Department of Hematology, University of Cambridge, Cambridge, UK
[2] MRC Biostatistics Unit, University of Cambridge, Cambridge, UK

This paper has two components:

1) An advance in computational efficiency for estimating beta-binomial regression coefficients with shrinkage. The authors have produced a C++ implementation of the inference code previously written in R. Both versions of the code are implemented in the apeglm R package.

2) An application of this new implementation of their method to the task of inferring allele-specific expression (ASE) and an assessment of its statistical performance in relation to two alternative approaches (ash and MLE).

As the authors start the paper by discussing ASE, rather than computational inference for shrinkage models, it is not immediately apparent that the innovation presented in this paper is computational rather than statistical. Distinguishing these two components clearly would make it more readily apparent that the paper does not present a novel statistical method.

The modelling of ASE has important facets that the authors do not discuss in the introduction (page 3) but which other (uncited) methods have addressed. For example, in a given sample, a gene may contain multiple heterozygous variants (potentially with uncertain phasing of alleles). Each heterozygous variant could overlap different sets of isoforms, each of which may have different levels of ASE. This phenomenon is modelled by the MMDIFF method (Turro *et al*, 2014, Bioinformatics[1]), for example. The authors should acknowledge this (unmodelled) complication in ASE and explain how they summarise allele-specific count data across multiple variants (e.g., SNPs or indels, which are possibly unphased) within genes to obtain the count pairs modelled by the beta-binomial shrinkage estimators.

The authors have performed several simulation studies and an analysis of a real ASE dataset. Both shrinkage estimators outperform MLE in the simulation studies. However, apeglm and MLE do approximately equally well in the real data set and both outperform ash by a significant margin. In

addition, filtering of genes with low allele-specific read counts improves the MLE in the simulation studies but it does not do so in the real data analysis. This discordance demonstrates that the real data are very dissimilar from the simulated data. Although I don't think a major rewrite is warranted, if the authors could demarcate the computational advance (which can be demonstrated by simulation studies that are not representative of ASE, as the authors have done) from the specific application to ASE (using a real data set and perhaps a more faithful simulation study), the striking difference in performance shown in Figures 1-3 would be less incongruous.

In the introduction, the inability of other methods to model the effects of continuous covariates or estimate differences in allelic imbalance between groups (this is not the case though, see MMDIFF) is highlighted and contrasted with the proposed method. However, the authors' own analysis of real data only uses an intercept model. It would be desirable to demonstrate the flexibility afforded by the proposed approach.

In the assessment of statistical performance using the real data set, the MLEs obtained from the held-out data are treated as truth, even though earlier in the paper the authors demonstrate that MLEs have a particularly high mean absolute error. Presumably, this is the case (for genes with relatively low counts) even when the sample size is 18. The authors should consider alternative measures of performance that do not have this drawback.

Minor comments:

- p3: "estimates for allelic expression proportions can be highly variable" - estimates are fixed, the authors should write "estimators".
- p3: a cancer dataset may not be the best choice of example to refer to the proportion of genes with allele-specific reads, due to the prevalence of somatic mutations.
- p3: when discussing filtering as a "remedy" perhaps explain that this achieves a boost in specificity at the cost of power.
- p3: "the most robust and reliable when dealing with small sample sizes" - this part of the sentence does not follow from the previous part, as there is no mention of ash's inadequacy.
- p3: "also introduced new source code" - it is not clear what the "also" refers to.
- p4: "the probability that counts for a particular gene belong to a particular allele" should be changed to "the probability that a read for a particular gene belongs to a particular allele" as the total "counts" will not be assigned to an allele as a block (the total counts derive from a heterogeneous mixture of reads from the two different alleles).
- p4: more information should be given about how the scale parameter of the Cauchy prior is "estimated by pooling information across genes".
- p4: the placement of the \cdot indexing the bold face beta is unusual, as the j subscript corresponds to the first rather than the second index.
- p9: rerunning the simulation study with 4 v 4 samples having run it with 5 v 5 samples

seems unnecessary, as such a small change in sample size is unlikely to alter the conclusions.

- p9: "Figure 1d" should read "Figure 3d".

References

1. Turro E, Astle WJ, Tavaré S: Flexible analysis of RNA-seq data using mixed effects models. *Bioinformatics*. 2014; **30** (2): 180-8 PubMed Abstract | Publisher Full Text

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

Partly

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: Biostatistics, genomics.

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.

Author Response 01 Nov 2020

Josh Zitovsky, University of North Carolina at Chapel Hill, Chapel Hill, USA

This paper has two components:

1) An advance in computational efficiency for estimating beta-binomial regression coefficients with shrinkage. The authors have produced a C++ implementation of the inference code previously written in R. Both versions of the code are implemented in the apeglm R package.

2) An application of this new implementation of their method to the task of inferring allele-specific expression (ASE) and an assessment of its statistical performance in relation to two alternative approaches (ash and MLE).

As the authors start the paper by discussing ASE, rather than computational inference for shrinkage models, it is not immediately apparent that the innovation presented in this paper is computational rather than statistical. Distinguishing these two components clearly would make it more readily apparent that the paper does not present a novel statistical method.

We feel that the manuscript title referencing "software", the abstract mentioning "we evaluated the accuracy of three different estimators" and "We also wrote C++ code to quickly calculate ... apeglm estimates", the citation of the apeglm publication in the Introduction ("To this end, we look at three different estimation methods... approximate posterior estimation of GLM coefficients (apeglm)[11]"), and the note about the software in the Introduction ("We also introduced new source code for the apeglm package") make it clear that the apeglm shrinkage method is not proposed as novel in this manuscript.

The modelling of ASE has important facets that the authors do not discuss in the introduction (page 3) but which other (uncited) methods have addressed. For example, in a given sample, a gene may contain multiple heterozygous variants (potentially with uncertain phasing of alleles). Each heterozygous variant could overlap different sets of isoforms, each of which may have different levels of ASE. This phenomenon is modelled by the MMDIFF method (Turro *et al*, 2014, Bioinformatics[1]), for example. The authors should acknowledge this (unmodelled) complication in ASE and explain how they summarise allele-specific count data across multiple variants (e.g., SNPs or indels, which are possibly unphased) within genes to obtain the count pairs modelled by the beta-binomial shrinkage estimators.

We thank the reviewer for bringing up this concern. Here we have focused exclusively on observed allelic counts, ignoring uncertainty of reads that align to both alleles and aggregation of read information across SNPs within a gene. Such data could feasibly be acquired with longer reads that are approaching the transcript length, but in general we agree this as a limitation of our manuscript. We have now added the following to our manuscript to address this unmodelled complication:

"The methods and performance benchmarks we focus on here address issues stemming from low-count genes and small sample sizes. There are other important concerns in allele-specific analysis of short read RNA-seq datasets, such as reference allele bias, but we do not address such problems here and the methods discussed cannot directly account for them. Our simulation does not involve reference allele bias, and the RNA-seq study we examine took specific measures to avoid reference allele bias. For methods and analysis concerns involving reference allele bias, see Turro et. al.[4] and Castel et. al.[1]."

The authors have performed several simulation studies and an analysis of a real ASE dataset. Both shrinkage estimators outperform MLE in the simulation studies. However, apeglm and MLE do approximately equally well in the real data set and both outperform

ash by a significant margin. In addition, filtering of genes with low allele-specific read counts improves the MLE in the simulation studies but it does not do so in the real data analysis. This discordance demonstrates that the real data are very dissimilar from the simulated data. Although I don't think a major rewrite is warranted, if the authors could demarcate the computational advance (which can be demonstrated by simulation studies that are not representative of ASE, as the authors have done) from the specific application to ASE (using a real data set and perhaps a more faithful simulation study), the striking difference in performance shown in Figures 1-3 would be less incongruous.

We thank the reviewer for pointing out that the simulation and real data results may have been seen as contradicting each other. Based on concerns voiced by other reviewers and our own investigations, we have determined that the issue is not in the simulations, but rather in the real data analyses. Specifically, when benchmarking our methods on the real data set, we had treated the ML estimates from a held-out set as the truth, but as the held-out set only contains 18 samples, the inherent instability and estimation variance present in ML estimators could still present an issue in the accuracy of these estimates. In other words, it may not be reasonable to expect that these ML estimates are close to the true effect sizes, and treating them as such could bias results in favor of ML estimates and against ash (as ash estimates are further from the MLE than apeglm on average). The real data analyses now have been changed to focus more on qualitative comparisons where the truth need not be known (e.g. extent of shrinkage, estimation variance, etc.), and we have largely left estimation accuracy assessments to the simulations. With these changes in place, the simulation and real data results are no longer incongruous.

In the introduction, the inability of other methods to model the effects of continuous covariates or estimate differences in allelic imbalance between groups (this is not the case though, see MMDIFF) is highlighted and contrasted with the proposed method. However, the authors' own analysis of real data only uses an intercept model. It would be desirable to demonstrate the flexibility afforded by the proposed approach.

Thank you for bringing the Turro, Astle and Tavaré (2014) paper to our attention. We have added a mention of this paper in the Introduction as an example of a Bayesian method that can deal with allelic counts and arbitrary design matrices, and have removed the sentence that mentioned that methods do not exist to perform Bayesian analysis with arbitrary designs.

Moreover, we agree that it would have been useful to showcase our method on more complicated design matrices to demonstrate the flexibility of our method. To this extent, we have extended our analysis of real data to include an application of apeglm and ash to a model with two binary covariates and an interaction. The results are discussed in the last paragraph of the "Sampling from the mouse dataset" subsection of the "Results" section.

In the assessment of statistical performance using the real data set, the MLEs obtained from the held-out data are treated as truth, even though earlier in the paper the authors demonstrate that MLEs have a particularly high mean absolute error. Presumably, this is the case (for genes with relatively low counts) even when the sample size is 18. The authors should consider alternative measures of performance that do not have this drawback.

We agree that treating the held-out MLEs as the truth is problematic and have changed the

analyses of our real data set so that results do not depend on knowledge of the truth. See our previous response detailing this issue.

Minor comments:

- p3: "estimates for allelic expression proportions can be highly variable" - estimates are fixed, the authors should write "estimators".
  This typo has been corrected.
- p3: a cancer dataset may not be the best choice of example to refer to the proportion of genes with allele-specific reads, due to the prevalence of somatic mutations.

We now clarify that the TCGA dataset referenced here only used the normal breast tissue samples, not the tumor samples.

- p3: when discussing filtering as a "remedy" perhaps explain that this achieves a boost in specificity at the cost of power.
  We have added this explanation as suggested.
- p3: "the most robust and reliable when dealing with small sample sizes" - this part of the sentence does not follow from the previous part, as there is no mention of ash's inadequacy.

We have changed this part of the sentence from "the most robust and reliable" to just "robust and reliable".

- p3: "also introduced new source code" - it is not clear what the "also" refers to.
  We have changed this sentence to make it more clear.
- p4: "the probability that counts for a particular gene belong to a particular allele" should be changed to "the probability that a read for a particular gene belongs to a particular allele" as the total "counts" will not be assigned to an allele as a block (the total counts derive from a heterogeneous mixture of reads from the two different alleles).
  We have made the suggested change.
- p4: more information should be given about how the scale parameter of the Cauchy prior is "estimated by pooling information across genes".

We have added the mathematical details regarding how the scale parameter is estimated in the Supplementary Methods section.

- p4: the placement of the \cdot indexing the bold face beta is unusual, as the j subscript corresponds to the first rather than the second index.

We have made notational changes so that the \cdot appears after the j subscript and not before

- p9: rerunning the simulation study with 4 v 4 samples having run it with 5 v 5 samples seems unnecessary, as such a small change in sample size is unlikely to alter the conclusions.

  Another reviewer made a similar comment, and so this result has been removed.
- p9: "Figure 1d" should read "Figure 3d".

  The typo has been corrected.

Competing Interests: No competing interests were disclosed.

Reviewer Report 04 February 2020

https://doi.org/10.5256/f1000research.23018.r58251

?

Jarad Niemi
Department of Statistics, Iowa State University, Ames, IA, USA
Ignacio Alvarez-Castro
University of the Republic, Montevideo, Uruguay

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

Yes.

- In our work a key issue is bias of allele reads toward a reference genome as explained in Sun and Hu (2014).[1] The authors should mention if this bias is relevant for the applications in this manuscript and, if yes, how the methods deal with the bias.

- The introduction argues against eliminating low count genes, yet the manuscript says "Genes where at least three samples did not have at least 10 counts were removed...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." Why the contradiction?

Is the description of the method technically sound?

No.

- While the writing is clear, we generally found the order of content confusing. For example, normal-based CI construction should be explained immediately after point estimation and before competing methods, simulation details, and method comparison metrics. We also found there was a lack of details, some of which was in the Supplementary Material but seemed like it should be included in the main manuscript.

In addition, we have outlined concerns below:

Major concerns:

- It isn't clear how MAE or CI coverage are calculated for the real data. For real data the truth is not known and therefore MAE and coverage cannot be calculated the way they can for the simulated data. Are you calculating MAE and coverage relative to the data? You comment "we are treating the MLE of the held-out set as the truth". Why? The simulation studies

seemed to show this is a relatively poor estimate of the truth.

Minor concerns:

- Please provide some statements for why a beta-binomial model is assumed as opposed to alternative model assumptions, e.g. binomial, normal, Poisson.

- We assume you are assume conditional independence in your beta-binomial likelihood and in your Cauchy distribution for the regression coefficients. If so, this should be stated explicitly, e.g. using "ind" above the tilde.

- How often is \phi_g estimated to be 500? How important is the value 500? Is this user specifiable in the package?

- It is unclear what is meant by "standard error" in the statement "apeglm provides Bayesian shrinkage estimates based on the mode of the posterior as well as standard errors." Is this the posterior standard deviation? Is it the (asymptotic) standard deviation of the estimator?

- The manuscript states "The scale parameter of the Cauchy prior, \gamma_j, is estimated by pooling information across genes". How exactly is this computed?

- It seems odd to have the Supplementary Material on a site other than F1000. We're disappointed that the Estimation Procedure in the Supplementary Material is not included in the main body of the manuscript as this seems to be key to the methodology. If not included in the main manuscript, perhaps more specific references, say to equation numbers, could be included in the main manuscript.

- We don't understand the statement "Like apeglm, ash can only shrink estimates for one covariate at a time." Isn't the assumed hierarchical distribution a joint hierarchical distribution, albeit assuming independence, for all regression coefficients? If so, then isn't it jointly shrinking all the estimates? Or is the procedure a step-wise procedure where MLEs are shrunk one-at-time?

- It is unclear why a Cauchy distribution is chosen. While a Cauchy distribution has the appealling property that it does not shrink large signals (very much), it generally does little shrinkage to small signals compared to alternative estimators, e.g. Bayesian LASSO (10.1198/016214508000000337,10.1093/biomet/asp047)[2,3], horseshoe (10.1093/biomet/asq017)[4], point-mass priors (10.1080/01621459.1993.10476353)[5]. In our applications, the true distribution of these regression coefficients often has a large spike around 0 which would suggest using a distribution with more mass than a Cauchy near 0.

- The statement "where 1 <= j <= K is chosen by the user" is confusing. Does the user specify which predictors have a Cauchy distributions? What exactly is the user choosing?

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

Partly.

- One reason to provide code and data are to ensure ability to replicate even if the text is

insufficient. So, ensuring the code is able to be run will provide sufficient details.

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

Yes.

- We also applaud the authors for making their code and data available.

  Reviewer 1 addressed this and we did not attempt to evaluate this further.

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

Partly.

In the abstract, the article claims:

1. "Apeglm consistently performed better than ML[E] according to a variety of criteria, including mean absolute error (MAE) and concordance at the top (CAT)."

   Table 1 and 2 provide supporting evidence for the claim that apeglm has lower MAE than MLE for a variety of simulation scenarios.

   Figures 1d and 2d shows apeglm and ash having similar CAT and ahead of the non-filtered MLE approach.

   It might be helpful to point out that ash, another shrinkage estimator, also consistently performs better than the MLE.

2. "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."

   We guess Figures 1a-c and 2a-c as well as line 4 in Table 1 were the evidence for this comment, but we find these figures extremely hard to interpret. The comment in the text is that "some genes with estimates close to the truth were severely shrunk, and several genes with truly large effects were shrunk to zero.", but it isn't clear that this is undesirable. Just because the truth is non-zero doesn't mean that the data randomly generated from this truth should suggest a non-zero result.

   With this being said, we would not be surprised about ash shrinking large signals more than apeglm since the Cauchy distribution (used in apeglm) will shrink large signals less than a normal distribution (used in ash) will, but, as Reviewer 1 points out, there are differences in likelihood and estimation procedure between these two methods which make understanding why differences occur more difficult.

3. "2hen 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."

Figure 4 provides the computational cost comparison and seems to show that apeglm is faster than aod, aods3, gamlss, HRQoL, and VGAM under the tested scenario. An alternative version of this figure would provide the ratio of runtimes for these other methods compared to apeglm. While the current version allows for an understanding of the computation time involved, the main purpose of the figure is in comparison of times.

It does seem a bit odd that the authors compared these packages for computation but not for accuracy. In addition, why is ash not included in this comparison?

Other:

Minor issues:

- Once you've defined an acronym, just use it, e.g. CAT.
- Be consistent with acronyms: choose ML or MLE and stick with it.
- Figure 5 seems unnecessary since an argument in this manuscript is to use "shrinkage" estimators rather than un-shrunk MLEs.
- An updated reference for 29. Alvarez-Castro is 10.3934/mbe.2019389[6]
- The beta-binomial is a discrete random variable and thus it has a probability mass function rather than a probability density function.

References

1. Sun W, Hu Y: Mapping of Expression Quantitative Trait Loci Using RNA-seq Data. 2014. 145-168 Publisher Full Text
2. Park T, Casella G: The Bayesian Lasso. *Journal of the American Statistical Association*. 2008; **103** (482): 681-686 Publisher Full Text
3. Hans C: Bayesian lasso regression. *Biometrika*. 2009; **96** (4): 835-845 Publisher Full Text
4. Carvalho C, Polson N, Scott J: The horseshoe estimator for sparse signals. *Biometrika*. 2010; **97** (2): 465-480 Publisher Full Text
5. George E, McCulloch R: Variable Selection via Gibbs Sampling. *Journal of the American Statistical Association*. 1993; **88** (423): 881-889 Publisher Full Text
6. Alvarez-Castro I, Niemi J: Fully Bayesian analysis of allele-specific RNA-seq data.*Math Biosci Eng*. 2019; **16** (6): 7751-7770 PubMed Abstract | Publisher Full Text

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

Yes

Is the description of the method technically sound?

No

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

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

Author Response 01 Nov 2020

Josh Zitovsky, University of North Carolina at Chapel Hill, Chapel Hill, USA

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

Yes.

- In our work a key issue is bias of allele reads toward a reference genome as explained in Sun and Hu (2014).1 The authors should mention if this bias is relevant for the applications in this manuscript and, if yes, how the methods deal with the bias.

Reference allele bias is indeed a potential problem when dealing with allelic counts from RNA-seq. However, the methods we benchmark in the manuscript cannot directly deal with such bias. Our simulation does not involve reference allele bias, and the RNA-seq study we examine took specific measures to avoid reference allele bias. We apologize for not clarifying this before and have added a paragraph at the end of the Introduction explaining these points.

- The introduction argues against eliminating low count genes, yet the manuscript says "Genes where at least three samples did not have at least 10 counts were removed...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." Why the contradiction?

W*hen filtering is done to remove genes with a high variance estimated allelic ratio, it is usually done with a threshold greater than e.g. 10 total counts per gene / one count per allele. Increased filtering may result in a loss of statistical power, when the optimal filtering rule is not known. Our minimal filtering was performed such that the metrics (e.g. error and ranking concordance) represent features for which there is some minimally detectable signal across alleles.*

Removing genes with a significant sex or parent effect was done for the purposes of performance analysis, as our analysis involved fitting intercept-only models. We did not want the extra variability induced from sex and/or parent effects in the set of genes used for evaluation.

Is the description of the method technically sound?

No.

- While the writing is clear, we generally found the order of content confusing. For example, normal-based CI construction should be explained immediately after point estimation and before competing methods, simulation details, and method comparison metrics. We also found there was a lack of details, some of which was in the Supplementary Material but seemed like it should be included in the main manuscript.

We have moved the description of how the methods compute CIs as suggested. Moreover, we have added additional details about the estimation methods in both the main manuscript (under the "Estimation Methods" subsection of the "Methods" section) and the Supplementary Material. For example, in the main manuscript, we added more details regarding apeglm's likelihood and prior, estimation of the overdispersion and qualitative differences between apeglm's and ash's methodologies. In the Supplemental material, we added more details regarding estimation of the overdispersion, estimation of the scale of the Cauchy prior and the numerical accuracy of our package.

In addition, we have outlined concerns below:

Major concerns:

- It isn't clear how MAE or CI coverage are calculated for the real data. For real data the truth is not known and therefore MAE and coverage cannot be calculated the way they can for the simulated data. Are you calculating MAE and coverage relative to the data? You comment "we are treating the MLE of the held-out set as the truth". Why? The simulation studies seemed to show this is a relatively poor estimate of the truth.

We thank the reviewer for noting this drawback in our initial submission. Initially, our choice to use the MLE of the held-out set as the truth came from the fact that the ML estimators are consistent and asymptotically efficiency estimators of the regression parameters, and thus if the held-out sets are sufficiently large, the ML estimates will be very close to the truth. However, the held-out set only consists of 18 samples, which in practice may be too small to be useful. We agree with your concerns that many of the same problems of ML estimators that we address in our manuscript, such as instability in the presence of low information, would still be present in the held-out sets. After thinking about this more and conducting additional analysis, we came to the conclusion that even when using as many as 24 samples, the ML estimates are not close enough to the truth for some genes and using them as the truth may bias results.

As a result, we have rewritten the real data analysis section to focus on qualitative assessments that do not require knowledge of the truth, such as differences in nature and extent of shrinkage between apeglm and ash and on estimation variance. Accuracy assessments have been largely left to simulations, where the true parameter values are known. Relatedly, we have changed the simulations so that the intercept is simulated from a standard normal distribution, as opposed to being drawn from ML estimates of intercept-only models fit to the genes of the real data set. The reason for this is similar: we have no reason to believe that the intercept ML estimates are close to the true intercepts, and upon investigation, we found that the distribution of ML estimates had several properties that would not realistically be demonstrated by a distribution of true effect sizes.

Minor concerns:

- Please provide some statements for why a beta-binomial model is assumed as opposed to alternative model assumptions, e.g. binomial, normal, Poisson.

We have added a justification for choosing a beta-binomial distribution to model allelic counts in the second paragraph of the "Estimation Methods" subsection of the "Methods" section.

- We assume you are assume conditional independence in your beta-binomial likelihood and in your Cauchy distribution for the regression coefficients. If so, this should be stated explicitly, e.g. using "ind" above the tilde.

We have made the suggested changes to the notation so that the assumed conditional independence is clearer

- How often is \phi_g estimated to be 500? How important is the value 500? Is this user specifiable in the package?

It is difficult to give an exact frequency, as how often phi is estimated at 500 varies from dataset to dataset. The number of genes in a dataset where no or very little overdispersion is exhibited by the allelic proportions (conditional on the covariates) is roughly the number of times at which phi will be estimated at 500 for the dataset. As phi approaches infinity, the resulting regression parameter MLEs converge to the MLEs from a binomial distribution. We found that with phi=500, the ML estimates are already quite close to the ML estimates from a model with assumption of a binomial distribution, and setting the maximum above 500 led to only very small differences in the coefficients. However, the user can specify a different maximum (and minimum) than that used in this package as desired. Details have been added to the main manuscript and Supplemental Methods regarding our chosen minimum and maximum.

- It is unclear what is meant by "standard error" in the statement "apeglm provides Bayesian shrinkage estimates based on the mode of the posterior as well as standard errors." Is this the posterior standard deviation? Is it the (asymptotic) standard deviation of the estimator?

It is the posterior standard deviation. We clarified this in the second version.

- The manuscript states "The scale parameter of the Cauchy prior, \gamma_j, is estimated by pooling information across genes". How exactly is this computed?

We have added this information in the Supplemental Material section

- It seems odd to have the Supplementary Material on a site other than F1000. We're disappointed that the Estimation Procedure in the Supplementary Material is not included in the main body of the manuscript as this seems to be key to the methodology. If not included in the main manuscript, perhaps more specific references, say to equation numbers, could be included in the main manuscript.

All references to the Supplemental Material have been made more specific, and are now references to the specific section of the Supplemental Material that is relevant.

- We don't understand the statement "Like apeglm, ash can only shrink estimates for one covariate at a time." Isn't the assumed hierarchical distribution a joint hierarchical distribution, albeit assuming independence, for all regression coefficients? If so, then isn't it jointly shrinking all the estimates? Or is the procedure a step-wise procedure where MLEs are shrunk one-at-time?

We apologize if this was not clear in the first version of the manuscript and have added clarifications in the new version of the manuscript and Supplemental Material. In summary, apeglm for allelic counts assumes a Beta-binomial likelihood for all regression coefficients, but it only assumes a Cauchy prior for one regression coefficient at a time (more specifically, the

regression coefficients for only one covariate, across all genes). Thus only one covariate is being "shrunk" at a time. If Bayesian shrinkage of two coefficients was desired (for example), you would have to run apeglm twice: the first time choosing one coefficient, and the second time choosing the other.

- It is unclear why a Cauchy distribution is chosen. While a Cauchy distribution has the appealling property that it does not shrink large signals (very much), it generally does little shrinkage to small signals compared to alternative estimators, e.g. Bayesian LASSO (10.1198/016214508000000337,10.1093/biomet/asp047)2,3, horseshoe (10.1093/biomet/asq017)4, point-mass priors (10.1080/01621459.1993.10476353)5. In our applications, the true distribution of these regression coefficients often has a large spike around 0 which would suggest using a distribution with more mass than a Cauchy near 0.

Our choice of a Cauchy prior was guided by the fact that a Cauchy prior tends to shrink large effect sizes less than other priors, and in a differential expression context was shown to produce estimates with lower error and better ranking be size than competing estimators (see reference 11). We agree that there are situations where a Cauchy prior would not be ideal, if sparsity of estimated coefficients (setting to exactly zero for certain genes) was desired for selection purposes. However apeglm follows and cites the ashr publication in providing the false sign rate (FSR) as a criterion for gene selection. A justification of our choice of a Cauchy prior and the flexibility of our software to handle other priors has also been added into the manuscript.

- The statement "where 1 <= j <= K is chosen by the user" is confusing. Does the user specify which predictors have a Cauchy distributions? What exactly is the user choosing?

This is exactly right: The user is specifying which predictor (singular) is assumed to follow a Cauchy distribution for the purpose of shrinkage estimation. We have tried to make this clearer in the second version of the manuscript. See two responses above.

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

Partly.

- One reason to provide code and data are to ensure ability to replicate even if the text is insufficient. So, ensuring the code is able to be run will provide sufficient details.

We apologize for the reproducibility issues present in the first part of the paper. A detailed explanation of the problems and our fixes was given in our responses to the first reviewer. We believe all previous issues have been fixed and the code should now run without problems (assuming all of the relevant packages are installed and the right package versions are being used).

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

Yes.

- We also applaud the authors for making their code and data available.

  Reviewer 1 addressed this and we did not attempt to evaluate this further.

Please see our response to your concern under "Are sufficient details provided to allow replication

of the method development and its use by others?".

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

Partly.

In the abstract, the article claims:

1. "Apeglm consistently performed better than ML[E] according to a variety of criteria, including mean absolute error (MAE) and concordance at the top (CAT)."

   Table 1 and 2 provide supporting evidence for the claim that apeglm has lower MAE than MLE for a variety of simulation scenarios.

   Figures 1d and 2d shows apeglm and ash having similar CAT and ahead of the non-filtered MLE approach.

   It might be helpful to point out that ash, another shrinkage estimator, also consistently performs better than the MLE.

Due to changes in the simulations (see our response to your "Major Concern" under "Is the description of the method technically sound?"), ash no longer performs better than maximum likelihood universally, though in general it still performs better. The abstract has been changed to accommodate the different results. We believe that our new abstract provides a succinct yet comprehensive and accurate summary of the new results.

1. "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."

   We guess Figures 1a-c and 2a-c as well as line 4 in Table 1 were the evidence for this comment, but we find these figures extremely hard to interpret. The comment in the text is that "some genes with estimates close to the truth were severely shrunk, and several genes with truly large effects were shrunk to zero.", but it isn't clear that this is undesirable. Just because the truth is non-zero doesn't mean that the data randomly generated from this truth should suggest a non-zero result.

   With this being said, we would not be surprised about ash shrinking large signals more than apeglm since the Cauchy distribution (used in apeglm) will shrink large signals less than a normal distribution (used in ash) will, but, as Reviewer 1 points out, there are differences in likelihood and estimation procedure between these two methods which make understanding why differences occur more difficult.

Reviewer 1 voiced similar concerns, and you can see our detailed response to this concern in our responses to the first reviewer. To summarize, we have removed results of mean absolute error stratified by the true effect sizes. We also look more at subsets chosen based only on observed data (e.g. total allele counts and MLE size) to interpret results. We hope our new results are easier to interpret and our conclusions more convincing.

1. "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."

Figure 4 provides the computational cost comparison and seems to show that apeglm is faster than aod, aods3, gamlss, HRQoL, and VGAM under the tested scenario. An alternative version of this figure would provide the ratio of runtimes for these other methods compared to apeglm. While the current version allows for an understanding of the computation time involved, the main purpose of the figure is in comparison of times.

It does seem a bit odd that the authors compared these packages for computation but not for accuracy. In addition, why is ash not included in this comparison?

We have changed the Figure as suggested to better illustrate relative performance of the other packages compared to apeglm. Moreover, we have added comparisons of numerical accuracy to the main manuscript (last paragraph of "Computational performance of apeglm" subsection) and Supplemental Material. Our package is more numerically accurate and reliable than other packages compared. As to why ash is not included in the comparison, this is because ash requires a vector of initial parameter estimates and standard error estimates, and thus to use ash as we do in the manuscript, one has to perform ML estimation first, and then use ash to shrink the estimates. Comparing ash to apeglm or the ML-fitting packages would thus not be a same-to-same comparison.

Other:

Minor issues:

- Once you've defined an acronym, just use it, e.g. CAT.

We have made the suggested changes to the manuscript.

- Be consistent with acronyms: choose ML or MLE and stick with it.

We have made the suggested changes to the manuscript.

- Figure 5 seems unnecessary since an argument in this manuscript is to use "shrinkage" estimators rather than un-shrunk MLEs.

Though our previous analysis showed that apeglm has higher accuracy than ML estimators, there are still reasons why one would prefer likelihood-based beta-binomial GLMs, such as if the sample size is large or if simplicity or unbiasedness is desired. Moreover, many shrinkage estimation packages like ash require a vector of initial ML estimates and standard errors. Finally, apeglm estimation is almost as fast as ML estimation when using the new apeglm package, and thus Figure 5 would be practically the same if we were to compare other packages to apeglm estimation speed instead. We have added this clarification in the "Computational performance of Apeglm" subsection of the "Results" section.

- An updated reference for 29. Alvarez-Castro is 10.3934/mbe.20193896

The reference has been updated.

- The beta-binomial is a discrete random variable and thus it has a probability mass function rather than a probability density function.

In the new manuscript, we refer to the probability function of the beta-binomial as its "probability mass function" as opposed to a "density function"

Competing Interests: No competing interests were disclosed.

Reviewer Report 17 December 2019

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

?

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 (eg 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[1] 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: "Apeglm 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

1. Xing Z, Carbonetto P, Stephens M: Flexible signal denoising via flexible empirical Bayes shrinkage. *arXiv*. 2019. Reference Source

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.

Author Response 01 Nov 2020

Josh Zitovsky, University of North Carolina at Chapel Hill, Chapel Hill, 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 (eg 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.

Thank you for your constructive comments and careful evaluation of our software and analysis. We found your report helpful and have tried our best to address all of your concerns. Point-by-point responses are provided below.

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.

We agree that there are important methodological differences between the methods, and that a high-level summary of these differences would be beneficial to the readers. We have added a paragraph highlighting these differences in the second-to-last paragraph of the "Estimation methods" subsection of the "Methods" section. Among other differences, we highlight the increased flexibility of ash's prior and its ability to handle non-ML estimators. Additional details regarding the methodology of these methods have also been added to the sections where apeglm and ash were initially introduced.

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:1605.077871 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.)

As you suspected, there were indeed genes with “truly infinite” MLEs, but due to numerical reasons, were given finite estimates by the apeglm package. As you suggested, we have now performed additional analyses adding a pseudocount to each allele prior to computing MLEs, and compared the performance of the resulting ML, apeglm and ash estimates to those not involving pseudocounts for the simulations. We also attempted to remove the infinite ML genes prior to analysis. Results can be found in Table 1, Table 2 and Supplementary Figure 3.

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.

Shrinkage in the first manuscript was defined as the movement of apeglm and ash estimates from the MLE toward zero. As apeglm, ash and ML estimates are all functions of the observed data, the degree of shrinkage is also a function of observed data and thus we felt that subsetting by shrinkage was valid. However, we do agree with your concern that subsetting by true effect sizes may cause difficulty in contrasting procedures with each other with respect to the optimal shrinkage rule, and thus have removed results of mean absolute error stratified by the true effect sizes. Moreover, per your suggestion, we have added stratification of MAE by total gene counts and MLE magnitude. We also added MA plots, which illustrates how the amount of shrinkage differs by total gene counts and MLE size (these plots were previously in the Supplemental Material, but have been moved to the main paper).

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?

We agree that an assessment of numerical accuracy is important in showcasing our package, and have adding such assessments in the new version of the manuscript. We focused our analysis of numerical accuracy on genes such that the difference in an estimated coefficient between apeglm and the other packages were non-negligible (above 0.01), and among those genes reported the differences in log-likelihood. A high-level overview of the results is present in the last paragraph of the "Computational Performance of Apeglm" subsection of the "Results" section, and a detailed summary of the results was added to the Supplementary Methods section. Overall, we found that our package is, in addition to its estimation speed, also numerically accurate.

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.

We apologize for the reproducibility issues in the first version of the paper. Briefly, the issues you reported stemmed from two underlying causes: 1) the version of the apeglm package in the devel branch at the time of publication did not match the version used in the manuscript; 2) we accidentally uploaded the wrong scripts to Zenodo. We have now correctly identified the apeglm package version in the manuscript (v1.11.2) and replaced the scripts in Zenodo with the correct ones. All scripts should now run without issues and output the same numbers and plots as shown in the paper. Moreover, we have removed absolute file paths and do not use global variables in our functions (some of the local variables defined within functions might share names with global variables created later on, but our functions no longer call global variables directly).

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.

Due to concerns posed by yourself and other reviewers, we have completely rewritten our analysis of real data to focus on more qualitative results, and have mostly left evaluations of accuracy to the simulations, where the true simulation parameters are known. Among other changes, we do not evaluate or assess coverage probabilities of estimators when analyzing the real data.

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.

We have removed this statement.

7. Minor comments:

- p3: "When a subject is heterozygous for a gene at a particular SNP"; this wording seemed awkward to me.

We have changed the wording to "When a subject is heterozygous at a particular SNP within an exon of a gene"

- 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.

We have changed this from "the most robust and reliable" to just "robust and reliable".

- p4: "Apeglm 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.

We have made the suggested change.

- 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).

Apologies for the confusion. We have changed it from "guided by the author's claim" to "guided by the fact" and have cited both ash and the original 1950's citation.

- p5: diallel typo?

In our original manuscript, we had the term "diallel cross", we did not find a typo.

- p5: use of beta for the mean of the exponential distribution is confusing as beta is already used elsewhere.

We changed the notation for the mean parameter from beta to mu.

- 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.

We have removed this result.

- 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.

Please see our response to your concerns in point #6.

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

Unfortunately, the y-axis for figure 5 of the initial version of the paper (renamed Figure 8 in version 2) is on the log-scale, which means we cannot start it at zero. Using a log scale is necessary due to the very different computational times of the apeglm and aod packages and the difference in how well they scale with increasing numbers of covariates. We considered changing the figure to start the y-axis at a smaller positive number (eg 10, 1, 0.1 etc.) but we ultimately decided against this as the exact cut-point at which to start the y-axis would have been arbitrary and there would have been a large amount of unnecessary white space between the plots and the x-axis (due to the fact that the y-axis is measured on the log scale).

Competing Interests: No competing interests were disclosed.