Differential Expression Analysis of Complex RNA-seq Experiments Using edgeR*

Yunshun Chen, Aaron T. L. Lun and Gordon K. Smyth

31 January 2014

Abstract

This article reviews the statistical theory underlying the edgeR software package for differential expression of RNA-seq data. Negative binomial models are used to capture the quadratic mean-variance relationship that can be observed in RNA-seq data. Conditional likelihood methods are used to avoid bias when estimating the level of variation. Empirical Bayes methods are used to allow gene-specific variation estimates even when the number of replicate samples is very small. Generalized linear models are used to accommodate arbitrarily complex designs. A key feature of the edgeR package is the use of weighted likelihood methods to implement a flexible empirical Bayes approach in the absence of easily tractable sampling distributions. The methodology is implemented in flexible software that is easy to use even for users who are not professional statisticians or bioinformaticians. The software is part of the Bioconductor project.

This article describes some recently implemented features. Loess-style weighting is used to improve the weighted likelihood approach, and an analogy with quasi-likelihood is used to estimate the optimal weight to be given to the empirical Bayes prior. The article includes a fully worked case study with complete code.

1 Introduction

With the dramatic drop in sequencing costs provided by the Next Generation sequencing technologies in past few years, RNA-seq has now supplanted microarrays as the technology of choice for genome level expression profiling of RNA samples [17, 28, 24]. RNA-seq data is typically summarized by counting the number of sequence reads that map to genomic features of interest [10]. In this article we will

*Please cite as: Chen, Y., Lun, A.T.L., and Smyth, G.K. (2014). Differential expression analysis of complex RNA-seq experiments using edgeR. In: *Statistical Analysis of Next Generation Sequence Data*, Somnath Datta and Daniel S Nettleton (eds), Springer, New York, pages 51–74.

assume that the aim is to conduct a gene-level analysis, but similar analyses could be done for exons or exon-junctions or other genomic constructs. One very common problem is to use the read counts to identify genes that are differentially expressed between experimental conditions.

This article reviews the statistical theory underlying the edgeR software package [20] for differential expression analysis of RNA-seq data. Rigorous and statistically powerful analysis of RNA-seq data requires careful attention to a number of issues. The read counts are discrete integers that show strong mean-variance relationships. Different genes show different levels of variability, but the number of replicate samples from which variability is estimated can be very small indeed. Meanwhile, experiments may involve complex experimental designs with multiple treatment factors and other experimental variables.

edgeR uses negative binomial based models to capture the quadratic mean-variance relationship that can be observed in RNA-seq data, and to distinguish between biological and technical sources of variation [15]. By technical variation, we mean that associated with the sequencing technology whereas biological variation refers to changes in expression levels between experimental subjects. Information is shared between genes to estimate biological variation reliably even when the number of replicates is very small [23]. Conditional likelihood methods are used to avoid bias when estimating the level of variation [23, 15]. Empirical Bayes methods are used to allow gene-specific variation estimates while borrowing information between genes [22, 15]. A key feature of the edgeR package is the use of weighted likelihood methods to implement a flexible empirical Bayes approach in the absence of easily tractable sampling distributions. Finally, generalized linear models are used to accommodate arbitrarily complex designs, and the conditional likelihood and empirical Bayes procedures are generalized to work in this context [15].

This article also describes some recent additions to the package, not previously described in published form. In particular, loess-style weighting is used to improved the weighted likelihood approach, and an analogy with quasi-likelihood [11] is used to estimate the optimal weight to be given to the empirical Bayes prior. The article includes a fully worked case study.

The edgeR package is part of the Bioconductor project [7]. Some advanced numerical algorithms are used to ensure reliable convergence of the iterative algorithms, and some of the core code has been implemented in C++ for speed and numerical stability. The package can be installed from the Bioconductor website `http://www.bioconductor.org`.

Table 1: Table of read counts for a simple RNA-seq experiment with four samples. Each column corresponds to a sample from a mouse with a wild-type or mutant genotype. Each row corresponds to a gene in the mouse genome. Each entry is set at the number of reads mapped to a particular gene in a particular sample. The sum of counts in each column is the library size for the corresponding sample.

	Wild-type		Mutant	
	Sample 1	Sample 2	Sample 3	Sample 4
Gene 1	24	31	76	59
Gene 2	0	3	7	2
Gene 3	1988	1125	3052	2450
Gene 4	5	0	0	1
...	...	...	...	...
Total	22341961	20739175	15669423	23711320

2 The Negative Binomial Model

2.1 Summarizing an RNA-seq Experiment with a Count Matrix

In a typical RNA-seq experiment, purified RNA is converted to cDNA and sequenced on one of the high-throughput platforms. Millions of short 'read' sequences ranging from 25 to 300 base pairs in length are generated from one (single-end) or both (paired-end) ends of the cDNA fragments. These sequences must be aligned (or *mapped*) to a reference genome or transcriptome. Summarization is then performed by counting the number of reads mapped to known genomic features such as genes or exons. For simplicity, we will refer to these features as 'genes' although any genomic interval can be used. This results in a table of read counts for tens of thousands of genes across a number of samples. These samples are associated with a variety of treatment conditions that we want to compare.

Table 1 shows an example of the matrix of read counts for a very simple RNA-seq experiment. The dataset consists of two groups (wild-type and mutant), each of which contains samples from two mice, i.e., two biological replicates. After sequencing, reads for each sample are mapped to the mouse genome and summarized into gene-level counts. The final RNA-seq expression profile is represented by a table of read counts for tens of thousands of genes in all four mice samples (Table 1). The aim of this experiment is to identify differentially expressed genes between wild-type and mutant mice.

In this article, the total number of genes is denoted by G and the total number

of samples is denoted by n. Hence, the table of read counts from an RNA-seq experiment is a $G \times n$ matrix of non-negative integers. We refer to the set of read counts for a sample as a *library* and the total number of reads in the library as the *library size*. For a particular gene g, let y_{gi} denote the read count in the ith sample. The expected value of y_{gi} given the experimental conditions and the sequencing depths is then

$$E(y_{gi}) = \mu_{gi} = \lambda_{gi} \cdot N_i, \tag{1}$$

where N_i is the library size and λ_{gi} is the expected proportion of reads mapped to gene g in the ith sample.

In the above example, we have $\lambda_{g1} = \lambda_{g2} = \lambda_g^W$ and $\lambda_{g3} = \lambda_{g4} = \lambda_g^M$ where λ_g^W and λ_g^M are the expected proportion of reads mapped to gene g in the wild-type and the mutant groups, respectively. Then, the aim of the differential expression analysis is to test

$$H_0 : \lambda_g^W = \lambda_g^M \qquad \text{against} \qquad H_1 : \lambda_g^W \neq \lambda_g^M, \tag{2}$$

for each gene $g = 1, 2, ..., G$.

2.2 Distinguishing Technical from Biological Variation

Two levels of variation can be distinguished in any RNA-seq experiment. First, there is the basic variability in the expression level of each gene from one biological sample to another, even when the experimental conditions have not been changed. Second, because expression levels can never be measured perfectly, there is a certain level of technical variation arising from measurement error. RNA-seq provides the possibility of disentangling these two sources of variation. Unlike microarrays, RNA-seq can do this without technical replicates of the same RNA samples, because the level of technical variation from sequencing is of a predictable nature.

Let π_{gi} be the fraction of all cDNA fragments in the ith sample that originate from gene g. This can be viewed as the true unobserved expression level of gene g in individual sample i. Given π_{gi} and the library size N_i, the expected count is $E(y_{gi}|\pi_{gi}) = \pi_{gi}N_i$. The read counts for any given gene are usually considered to follow a Poisson law under repeated sequencing runs of the same RNA sample [14], so it is reasonable to suppose that $\text{var}(y_{gi}|\pi_{gi}) = \pi_{gi}N_i$ also. This represents technical variability associated with the sequencing technology.

Let us further suppose that π_{gi} varies between biological replicates in such a way that the coefficient of variation (CV) remains constant for any given gene. This implies that $E(\pi_{gi}) = \lambda_{gi}$ and $\text{var}(\pi_{gi}) = \phi_g\lambda_{gi}^2$, where ϕ_g is the squared CV and λ_{gi} is the population mean proportion for gene g given the experimental conditions applied to sample i. The unconditional variance of y_{gi} can then be derived as

$$\text{var}(y_{gi}) = E_\pi[\text{var}(y|\pi)] + \text{var}_\pi[E(y|\pi)] = \mu_{gi} + \phi_g\mu_{gi}^2 \tag{3}$$

where $\mu_{gi} = \lambda_{gi}N_i$ is the population mean of y_{gi}. Dividing both sides by μ_{gi}^2 gives

$$\text{CV}^2(y_{gi}) = 1/\mu_{gi} + \phi_g \tag{4}$$

The first term is the squared CV of y_{gi} given π_{gi} and the second is the squared CV of π_{gi}. In other words,

$$\text{Total CV}^2 = \text{Technical CV}^2 + \text{Biological CV}^2 \, . \tag{5}$$

This partition of CV^2 into technical and biological components was first derived by [15].

We call $\phi_g^{1/2}$ the biological coefficient of variation (BCV). BCV represents the coefficient of variation with which the true abundance of the gene varies between replicate RNA samples. It represents the CV that would remain between biological replicates if sequencing depth could be increased indefinitely. Note that the technical CV decreases as the size of the counts increases whereas the BCV does not. Thus, the BCV is likely to be the dominant source of uncertainty for high-count genes. Reliable estimation of the BCV is therefore crucial for realistic assessment of differential expression in RNA-seq experiments.

2.3 Generalized Linear Models Accommodate Complex Experiments

Generalized linear models (GLMs) are an extension of classical linear models to non-normally distributed response data [18, 16]. We use GLMs to accommodate complex experimental designs with multiple explanatory factors. GLMs allow the responses to follow any linear exponential family of probability distributions, and each distribution family is characterized by it mean-variance relationship. In our case, the quadratic mean-variance relationship shown above in Equation 3 determines the negative binomial distribution family for read counts. We assume therefore that

$$y_{gi} \sim \text{NB}(\mu_{gi}, \phi_g) \, , \tag{6}$$

where μ_{gi} is the mean and ϕ_g is now the negative binomial dispersion parameter. The assumption of negative binomial variation for y_{gi} is equivalent to assuming that the true gene abundances π_{gi} follow a gamma distributional law across replicate RNA samples.

We use a log-linear model to represent the influence of the treatment conditions and the library sizes on the expected count sizes for any gene. Recall that μ_{gi} is the product of the expression proportion λ_{gi} and the library size. We suppose that λ_{gi} can be represented by a log-linear model,

$$\log \lambda_{gi} = x_i^T \beta_g, \tag{7}$$

where x_i is a covariate vector indicating the treatment conditions applied to sample i and β_g is a vector of regression coefficients by which the covariate effects are mediated for gene g. It follows that

$$\log \mu_{gi} = x_i^T \beta_g + \log N_i. \tag{8}$$

Gathering the covariate vectors x_i into a design matrix X, the vector of linear predictors for gene g is the matrix product $X\beta_g$. The standard GLM method would use Fisher-scoring to estimate the parameter vector β_g. This is usually successful but can fail to converge for some datasets. edgeR enhances the usual Fisher-scoring algorithm with a Levenberg damping modification to ensure that the sequence of iterations converges for all genes and all datasets [15]. The modified algorithm forces a reduction in the residual deviance at each iteration. The sequence of deviances is monotonic and bounded, and so always converges unless floating point inaccuracies intervene first.

In the simple example shown in Section 2.1, the design matrix might take the form

$$X = \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 1 \\ 1 & 1 \end{pmatrix} . \tag{9}$$

In that case the first regression coefficient β_{g1} would represent the log-expression proportion in the wild-type group and the second coefficient β_{g2} would represent the log-fold change in expression in the mutant group relative to wild-type. In the notation of Section 2.1, $\beta_{g1} = \log \lambda_g^W$ and $\beta_{g2} = \log(\lambda_g^M / \lambda_g^W)$. The hypothesis of interest in this example is

$$H_{0g} : \beta_{g2} = 0 \qquad \text{against} \qquad H_{1g} : \beta_{g2} \neq 0, \tag{10}$$

and this hypothesis is tested for all genes.

edgeR provides the ability to test whether any contrast of the regresssion coefficients equal to zero. Specifically, one can test the null hypothesis $H_0 : c^T \beta_g = 0$ where c is an arbitrary contrast vector. By default, hypotheses are tested using the usual asymptotic chisquare approximation to the likelihood ratio statistic, although edgeR also offers two more conservative F-test approximations as alternative options.

3 Empirical Bayes Dispersion Estimation

3.1 Overview

Accurate estimation of the dispersion parameter ϕ in the negative binomial model is vital for fitting GLMs and assessing differential expression. Given that an RNA-seq dataset often has a small number of samples, traditional univariate estimators of ϕ tend to perform poorly [23]. Maximum likelihood estimators (MLEs) in particular tend to underestimate dispersion parameters because they make no adjustment for the fact that the mean is estimated from the same data [23].

The differential expression analysis of an RNA-seq experiment with a one-way layout was studied by Robinson and Smyth [22, 23] who proposed a quantile-adjusted conditional maximum likelihood method for dispersion estimation. This

approach is available in edgeR via the `estimateTagwiseDisp` function, but is restricted to experiments with a one-way layout, i.e., to experiments with only one experimental factor.

In this chapter, we will focus on the general case in which RNA-seq experiments may involve multiple treatment conditions and blocking variables. Dispersion estimation for complex experimental designs was studied by McCarthy *et al.* [15]. Their method is based on the idea of an adjusted profile likelihood proposed by Cox and Reid [4].

3.2 Cox-Reid Adjusted Profile Likelihood

For general RNA-seq experiments with multiple factors, negative binomial dispersions are estimated using the Cox-Reid (CR) adjusted profile likelihood method [4, 15]. The CR method is based on the idea of approximate conditional likelihood which reduces to residual maximum likelihood (REML). Briefly, REML removes the effect of nuisance parameters which allows unbiased estimation of the dispersion. This accounts for all systematic sources of variation in the model.

For the purpose of estimating the dispersion, ϕ_g is the parameter of interest whereas the regression coefficients β_g and the means μ_{gi} are nuisance parameters. One condition of the CR method is that the nuisance parameters are assumed to be orthogonal to the parameter of interest, i.e., the Fisher information matrix must be block diagonal [4]. It can be shown that orthogonality between β_g and ϕ_g follows here from the fact that ϕ_g appears only in the variance function and not in the mean of the negative binomial GLMs [26].

The Cox-Reid adjusted profile likelihood (APL) for ϕ_g is the penalized log-likelihood, i.e.,

$$\mathrm{APL}_g(\phi_g) = \ell(\phi_g; y_g, \hat{\beta}_g) - \frac{1}{2}\log\det(\mathcal{I}_g), \tag{11}$$

where y_g is the vector of counts for gene g, $\hat{\beta}_g$ is the estimated coefficient vector, ℓ is the log-likelihood function and $\mathcal{I}_g$ is the Fisher information of β_g evaluated at $\hat{\beta}_g$ and ϕ_g.

Note that the $\hat{\beta}_g$ is the MLE of β_g given ϕ_g. Thus, $\hat{\beta}_g$ is also a function of ϕ_g. This means that the log-likelihood ℓ can be considered as a profile likelihood ℓ_p which depends only on ϕ_g, i.e., $\ell(\phi_g; y_g, \hat{\beta}_g) = \ell_p(\phi_g; y_g)$. Similarly, the adjustment term $\mathcal{I}_g$ can be treated as a function of ϕ_g. Maximization of $\mathrm{APL}_g(\phi_g)$ can then be used to obtain an estimate for ϕ_g.

3.3 Weighted Likelihood Empirical Bayes

The empirical Bayes method is one of the most powerful tools in data analysis. The aim is to estimate the prior distribution from the data and then apply the standard Bayesian approach to obtain posterior estimates. Empirical Bayes estima-

tion has been shown to outperform classical maximum likelihood estimates for high dimensional problems [6, 5, 25].

The cost of RNA-seq experiments often limits RNA-seq studies to only a small number of replicate libraries. This makes it difficult to obtain reliable dispersion estimates. The situation is further complicated by the fact that different genes may have different dispersions. For microarray data, this problem has been solved by applying an empirical Bayes strategy [25] where information is shared across genes or probes to stabilize the gene-wise variance estimates. It is tempting to apply a similar approach to RNA-seq data. Unfortunately, the direct empirical Bayes approach to stabilize the dispersion estimates is not applicable in the case of RNA-seq data since there is no conjugate prior distribution for the negative binomial dispersion ϕ.

One way to approximate the empirical Bayes strategy is to use a weighted likelihood. It can be shown that an empirical Bayes estimator is equivalent to an estimate obtained by maximizing a weighted likelihood function on a set of observations [27, 3]. This result provides an opportunity to implement an approximation of the empirical Bayes method for RNA-seq data.

Common Dispersion The simplest approach of sharing information between genes is to assume that all genes share a same dispersion value ϕ, which is called the *common dispersion* [23, 15]. It can be estimated by maximizing the common APL, which is defined as

$$\mathrm{APL}_C(\phi) = \frac{1}{G}\sum_{g=1}^{G}\mathrm{APL}_g(\phi), \tag{12}$$

where G is the total number of genes in the dataset.

The common APL can be considered as a special weighted likelihood in which the likelihoods for each gene have equal weights. Hence, all genes contribute equally to the estimation of this common dispersion. A common dispersion can be estimated in edgeR via the `estimateGLMCommonDisp` function.

Trended Dispersion The common dispersion approach is almost certainly too simple. It is far more likely that some genes have larger or smaller dispersion values than other genes. It has been found in many RNA-seq datasets that genes with lower expression level tend to have larger dispersions, and vice versa. Hence, it is reasonable to assume that the dispersion values depend on the gene-wise expression levels and can be modelled by a mean-dispersion trend [1]. In edgeR, the dispersion values obtained from the mean-dispersion trend are referred to as the *trended dispersion*, and in principle genes with the same expression level (or the same mean) should have the same trended dispersion.

The trended dispersion can also be estimated by the weighted likelihood approach. Given an RNA-seq dataset, the overall expression level of each gene is

calculated as an average across all samples and expressed as an average log count-per-million (logCPM) using the `aveLogCPM` function. This average is computed by a simple GLM, taking into account the common dispersion and the library sizes. Then, all the genes are sorted according to their average logCPM values. For a particular gene g, a locally shared APL denoted $\text{APL}_{S_g}(\phi_g)$ is formed by averaging the APLs of the set of genes, denoted C_g, that are nearest to gene g in average logCPM. By default, the neighbourhood set C_g is chosen to contain at least 25% of all genes, and the proportion is automatically increased if the total number of genes in the dataset is small. This ensures that each set C_g contains enough genes (and hence sufficient information) to represent the dispersion trend locally.

A graduated weighting approach was used to account for the relevance in expression level between gene g and other genes in the set C_g. The weight for the APL of gene a in C_g, denoted w_a, is determined by the tricube function, i.e.,

$$w_a = (1 - |x_a|^3)^3, \tag{13}$$

where $-1 < x_a < 1$ represents the scaled difference in average logCPMs for genes g and a. In other words, the closer the expression levels of genes g and a are, the smaller $|x_a|$ will be, and thus the larger w_a will be. This process can be repeated for all the genes in the set to obtain

$$\text{APL}_{S_g}(\phi_g) = \frac{\sum_{a \in C_g} w_a \cdot \text{APL}_a(\phi_g)}{\sum_{a \in C_g} w_a}, \tag{14}$$

as the locally shared APL for gene g. This is equivalent to fixing ϕ to a constant, fitting a loess curve of degree 0 through those $\text{APL}_a(\phi)$ for $a = 1, 2, ..., G$, and using the fitted value as the final value of the locally shared APL at ϕ for each gene. The trended dispersion for gene g can then be estimated by maximizing $\text{APL}_{S_g}(\phi_g)$.

Gene-specific Dispersion The trended dispersion approach would be sufficient if the true dispersions followed the mean-dispersion trend and genes with the same expression level had identical dispersion. This however is rarely true for real datasets and in practice dispersions are gene-specific. An individual dispersion therefore should be estimated for each individual gene, yet we are faced with the problem that the data from a single gene are often insufficient for reliable estimation of this dispersion. We need therefore a method that allows each gene to have its own dispersion estimate while still gaining information from the other genes. This can be achieved by an empirical Bayes approach that combines individual and shared information to obtain stable dispersion estimators. Such an approach has the effect of squeezing the genewise dispersions towards a pooled estimate, resulting in more stable inference when the number of samples is small.

The problem with directly applying the empirical Bayes approach is that there is no conjugate prior for the negative binomial dispersion ϕ_g. Thus, a weighted likelihood method has been proposed to approximate the empirical Bayes strategy

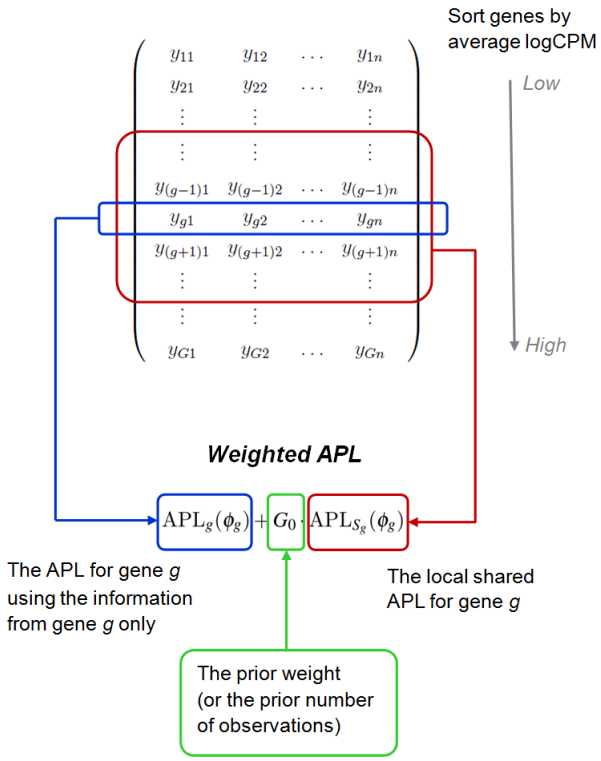


Figure 1: Genes are sorted by their expression level. The gene-specific dispersion for a particular gene g is estimated by maximizing the weighted APL, i.e., the weighted average between the gene-wise APL and the locally shared APL. The weight assigned to the locally shared likelihood is denoted by G_0 which can be interpreted as the prior number of observations.

for RNA-seq count data [22, 15]. To estimate the gene-specific dispersion, the weighted APL for a particular gene g is constructed as

$$\mathrm{APL}_{W_g}(\phi_g) = \mathrm{APL}_g(\phi_g) + G_0 \cdot \mathrm{APL}_{S_g}(\phi_g), \tag{15}$$

where $\mathrm{APL}_g(\phi_g)$ is the gene-wise APL using the information from gene g only, $\mathrm{APL}_{S_g}(\phi_g)$ is the locally shared APL for gene g, and G_0 is the weight assigned to the $\mathrm{APL}_{S_g}(\phi_g)$. The gene-specific dispersion ϕ_g is then estimated by maximizing $\mathrm{APL}_{W_g}(\phi_g)$. This weighted APL approach is described in Figure 1.

In empirical Bayes terms, the locally shared APL, $\mathrm{APL}_{S_g}(\phi_g)$, can be interpreted as the prior distribution for ϕ_g, and the $\mathrm{APL}_g(\phi_g)$ as the likelihood from the direct observed data. This means that the $\mathrm{APL}_{W_g}(\phi_g)$ can be interpreted as the posterior distribution for ϕ_g, which is a compromise between the prior and the observation. In the weighed likelihood approach, the prior distribution for ϕ_g can be thought of as arising from prior observations on a set of G_0 genes. Hence, the prior weight G_0 is referred to as the *prior number of observations*.

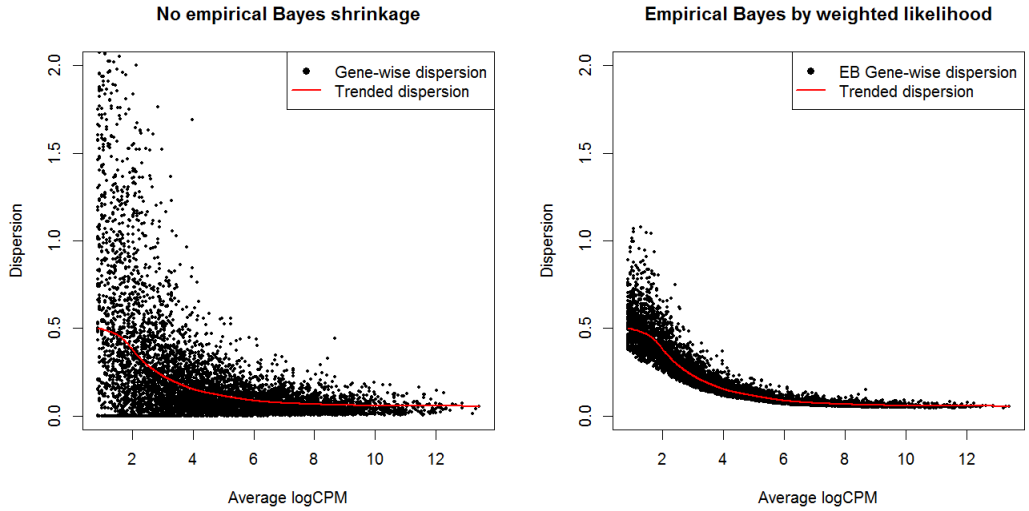


Figure 2: The empirical Bayes shrinkage by weighted likelihood on simulated data. The plot on the left shows the dispersion estimates without empirical Bayes shrinkage. For each gene, the gene-wise dispersion estimate is obtained using the information of that gene only. The plot on the right shows the gene-wise dispersion estimates after empirical Bayes shrinkage. Gene-wise dispersion estimates are squeezed towards the dispersion trend which represents the use of prior information.

The optimal choice for G_0 depends on the variability of the dispersions. Large values are best when the dispersions are a constant for all the genes or they closely follow the mean-dispersion trend. Smaller values are recommended when the dispersions are more variable among different genes. If $G_0 = 0$, no information is borrowed from other genes. This means that the gene-specific dispersion for a particular gene is purely estimated from its gene-wise APL. If G_0 is set to be infinitely large, information from that individual gene will be ignored. This means that the gene-specific dispersion will be fully determined by its locally shared APL such that the result will be the same as the trended dispersion. This information borrowing strategy can be viewed as shrinking individual dispersion estimates towards the dispersion trend (Figure 2) where the value of G_0 represents the amount of shrinkage.

3.4 Estimating Prior Weight

As mentioned previously, there is no conjugate prior for the genewise dispersion parameters. This means that there is no automatic estimation for the prior number of observations G_0. Thus, an alterative approach must be used. To account for the fact that more samples result in more gene-wise information, we write G_0 as

$$G_0 = \frac{d_0}{d_g}, \tag{16}$$

where d_0 is the *prior degrees of freedom* and d_g is the (known) residual degrees of freedom for gene g. The prior degrees of freedom represents the precision of the prior and does not depend on the total number of samples. The prior degrees of freedom can also be viewed as a measure of the consistency of the genewise dispersions. If the dispersions tend to be very gene-specific, then d_0 should be small and the prior will be vague. If the genewise dispersions tend to be consistent, i.e., close to the global trend, then d_0 should be large making the prior very informative. Once we estimate the d_0, we can easily calculate the prior weight G_0 in the weighted likelihood to obtain the best estimator for ϕ_g.

One way to estimate the prior degrees of freedom under the GLM framework is to use a quasi-likelihood in which the uncertainty of the variance can be absorbed into an overdispersion parameter. In GLM theory, the variance function $V(\mu)$ uniquely specifies a probability distribution such as the Poisson or negative binomial distribution. The quasi-likelihood variance function can then be written as

$$\text{var}(y_{gi}) = \sigma_g^2 \cdot V(\mu_{gi}), \tag{17}$$

where σ_g^2 is a factor that we will call the *quasi-dispersion parameter*. Note that the quasi-likelihood function is not a log-likelihood corresponding to any actual probability distribution. Instead, it can be used to describe a function that has similar properties to a log-likelihood function.

Following [11], we assume that the prior distribution for σ_g^2 is a scaled inverse χ^2-distribution with degrees of freedom d_0 and scaling factor $s_0^2 d_0$, i.e.,

$$\sigma_g^2 \sim s_0^2 \cdot \frac{d_0}{\chi^2_{d_0}}, \tag{18}$$

where s_0^2 can be considered as a prior mean for the quasi-dispersion. Our aim is to estimate d_0, which represents the precision of the prior distribution for σ_g^2.

Write D_g for the residual deviance of the generalized linear model fitted to the read counts for gene g. The mean residual deviance

$$s_g^2 = \frac{1}{d_g} D_g \tag{19}$$

is an estimator of σ_g^2. It can be shown [3] using the saddlepoint approximation [8] that the mean deviance s_g^2 follows approximately a χ^2-distribution with degrees of freedom d_g and scaling factor σ_g^2/d_g , i.e.,

$$s_g^2 | \sigma_g^2 \sim \sigma_g^2 \cdot \frac{\chi^2_{d_g}}{d_g}. \tag{20}$$

To make this approximation more accurate, a special calculation is required for the residual degrees of freedom d_g when some of the fitted values are exactly zero. In particular, we ensure that any experimental condition for which the counts are all

zero does not contribute to d_g. This is because such counts will have fitted values exactly zero and will make zero contribution to the residual deviance regardless of the value of the dispersion. This calculation is a refinement on the procedure of Lund *et al.* [11], and serves to make s_g^2 more nearly unbiased for σ_g^2 in the presence of zero counts.

The values of s_0^2 and d_0 can be estimated from the marginal distribution of s_g^2, which is scaled F-distribution,

$$s_g^2 \sim s_0^2 \cdot F_{d_g,d_0}, \tag{21}$$

where F_{d_g,d_0} denotes the F-distribution with degrees of freedom d_g and d_0 [25, 11]. Estimators of s_0^2 and d_0 can then be obtained by the method of moments [25].

In the main edgeR analysis pipeline, the quasi-likelihood is used only to estimate d_0. We assume that it is reasonable to use the same d_0 for empirical Bayes estimation of the negative binomial dispersions ϕ_g as for the quasi-dispersions σ_g^2. This allows us to calculate the prior weight G_0 required for Equation 15 from Equation 16 using d_g and the quasi-likelihood estimate for d_0.

4 Case Study: Transcriptional Program Regulation by IRF4

4.1 Experimental Design

We now demonstrate by way of a case study how the statistical theory in Sections 2 and 3 is applied in practice to analyze RNA-seq datasets. The case study includes the complete R code used to undertake the analysis. The data are from a study on the transcription factor IRF4 [13]. In the study, it was found that IRF4 regulated the expression of key molecules required for the aerobic glycolysis of effector T cells and was essential for the clonal expansion and maintenance of effector function of antigen-specific CD8+ T cells [13].

One part of this study was to identify the transcriptional program regulated by IRF4 during the TCR affinity-driven population expansion of CD8+ T cells. To investigate this, T cells were harvested from Irf4$^{+/+}$ wild-type or Irf4$^{-/-}$ knock-out mice. The knock-out mice have a mutation which prevents the Irf4 gene from producing a viable protein. T cells were stimulated with high-affinity peptides (N4) or low-affinity peptides (V4). RNA was extracted from the cells and profiled using RNA-seq.

The study can be viewed as a 2×2 factorial experiment with 2–3 replicates for each combination of IRF4 and affinity peptide conditions. There are 9 RNA samples in all. As is usual for an edgeR analysis, we start with experimental information about each RNA sample contained in a data frame called `targets`. The data frame was created using a spreadsheet and read into R using `readTargets`. It contains

the two experimental factors, Genotype and Treatment, as well as the identifier for each sample on the public ENA repository:

```
> targets
        ENA      Label Genotype Treatment
1 SRR953136 WT.N4.rep1       WT        N4
2 SRR953137 WT.N4.rep2       WT        N4
3 SRR953138 WT.V4.rep1       WT        V4
4 SRR953139 WT.V4.rep2       WT        V4
5 SRR953140 KO.N4.rep1       KO        N4
6 SRR953141 KO.N4.rep2       KO        N4
7 SRR953142 KO.N4.rep3       KO        N4
8 SRR953143 KO.V4.rep1       KO        V4
9 SRR953144 KO.V4.rep2       KO        V4
```

The aim is to detect genes that are differentially expressed (DE) between different conditions.

4.2 Mapping Reads to the Mouse Genome

The RNA samples were sequenced on an Illumina HiSeq 2000 at the Australian Genome Reseach Facility. Paired end sequencing was used, and reads were 100 bases long. This means that the first and last 100 bases of each RNA fragment were sequenced. Fragments were up to about 600 bases long in total.

The raw sequence reads are available either in SRA format from the Gene Expression Omnibus (`http://www.ncbi.nlm.nih.gov/geo`) as series GSE49929 or in FastQ format from the European Nucleotide Archive (ENA) (`http://www.ebi.ac.uk/ena`) as series SRP028864. We analyse here gzipped FastQ files downloaded from ENA. There are a total of 11 samples under ENA series SRP028864, the first 9 of which are analyzed here.

We start with a data frame of file names in R:

```
> files
               Forward              Reverse           SAM
1 SRR953136_1.fastq.gz SRR953136_2.fastq.gz SRR953136.sam
2 SRR953137_1.fastq.gz SRR953137_2.fastq.gz SRR953137.sam
3 SRR953138_1.fastq.gz SRR953138_2.fastq.gz SRR953138.sam
4 SRR953139_1.fastq.gz SRR953139_2.fastq.gz SRR953139.sam
5 SRR953140_1.fastq.gz SRR953140_2.fastq.gz SRR953140.sam
6 SRR953141_1.fastq.gz SRR953141_2.fastq.gz SRR953141.sam
7 SRR953142_1.fastq.gz SRR953142_2.fastq.gz SRR953142.sam
8 SRR953143_1.fastq.gz SRR953143_2.fastq.gz SRR953143.sam
9 SRR953144_1.fastq.gz SRR953144_2.fastq.gz SRR953144.sam
```

Each row corresponds to an RNA sample. The first column gives the name of the file containing the sequences of the forward strand ends of the RNA fragments. The

second column gives the name of the file containing the reverse strand reads.

The paired reads were mapped to the mouse genome using the Subread aligner [9]. The aligner uses the reads from both ends of each fragment to locate the fragment on the genome.

```
> library(Rsubread)
> align("mm9", readfile1=files$Forward, readfile2=files$Reverse,
+       "gzFASTQ", output_file=files$SAM, tieBreakQS=TRUE)
```

This code also uses an index (`"mm9"`) of the mouse genome. The index was created from the NCBI37/mm9 (July 2007) build of the mouse genome using the `buildindex` command of the subread package [9]. The mm9 index file can be downloaded from the Subread website `http://subread.sourceforge.net`.

The number of reads (forward and reverse) varies from 12 million to 19 million for each sample. For this dataset, the proportion of reads successfully mapped to the genome was more than 99% for all samples. This suggests good quality RNA samples and successful alignment:

```
> propmapped(file$SAM)
        Samples NumTotal NumMapped PropMapped
1 SRR953136.sam 13164036  13089886      0.994
2 SRR953137.sam 13007946  12932901      0.994
3 SRR953138.sam 12919854  12849910      0.995
4 SRR953139.sam 12334822  12262014      0.994
5 SRR953140.sam 12454324  12370667      0.993
6 SRR953141.sam 18595382  18487656      0.994
7 SRR953142.sam 19119234  19008197      0.994
8 SRR953143.sam 13217130  13125153      0.993
9 SRR953144.sam 13273338  13200580      0.995
```

4.3 Fragment Counts for Each Gene

Now we compute a table of genewise counts. This is a two-step process. First the mapped reads are converted into mapped RNA fragments. A pair of forward and reverse reads is considered to represent an RNA fragment whenever they map to compatible nearby locations on the genome. The fragment is then assigned to a gene whenever the fragment overlaps at least one exon of the gene. This computation is done by the `featureCounts` function of the Rsubread package [10]:

```
> fc <- featureCounts(files$SAM, isPairedEnd=TRUE)
```

By default, the function uses RefSeq annotation from the National Center for Biotechnology Information (NCBI) giving the start and end positions of each exon [19]. The output is a matrix of counts, one row for each NCBI Entrez Gene identifier and one column for each RNA sample.

4.4 Creating a DGEList Object

The edgeR package stores data in a simple list-based data object called a DGEList. edgeR provides a range of generic functions and methods for such data objects, but they can at the same time be manipulated like ordinary lists in R. The main components of a DGEList object are a matrix of integer counts, a data frame of sample information and an optional data frame of gene annotation.

```
> library(edgeR)
> y <- DGEList(counts=fc$counts, group=targets$Genotype)
> colnames(y) <- targets$Label
```

There are entries for 26310 genes and 9 samples:

```
> dim(y)
[1] 26301     9
```

Note the application of standard generic functions `colnames` and `dim` which have methods defined for DGEList objects. Many other generic functions in R that are applicable to matrices or data frames also have methods for `DGEList` objects.

The library sizes are automatically computed by `DGEList` as the total number of assigned RNA fragments for each sample. The number of mapped fragments is slightly less than half the total number of mapped reads shown in Section 4.2, and the number of fragments assigned to genes is about 80% of that.

```
> y$samples
           group lib.size norm.factors
WT.N4.rep1    WT  5038159            1
WT.N4.rep2    WT  4966457            1
WT.V4.rep1    WT  5026320            1
WT.V4.rep2    WT  4665370            1
KO.N4.rep1    KO  4703442            1
KO.N4.rep2    KO  6975408            1
KO.N4.rep3    KO  7271163            1
KO.V4.rep1    KO  4726829            1
KO.V4.rep2    KO  4995218            1
```

Many edgeR functions will accept an ordinary matrix of counts, but a DGEList object is more convenient because it automatically collates a variety of related information. For example, subsetting the above DGEList object `y` by column would automatically subset both the counts and the sample information at the same time.

4.5 Filtering and Normalization

Genes with counts that are all zero or all very low are usually not of interest in a differential expression analysis for two reasons. The biological reason is that a gene must be expressed at some minimal level before it is likely to be translated into a protein or to be biologically important. The statistical reason is that very low counts

provide little statistical information to distinguish between the null and alternative hypotheses. In this particular dataset, we consider a gene to be expressed at a reasonable level in a sample if its count-per-million (CPM) value is above 1, which is equivalent to having 5–7 fragments in that sample. A gene is kept in the analysis if it is sufficiently expressed (CPM $>$ 1) in at least two samples:

```
> CPM <- cpm(y)
> keep <-  rowSums(CPM > 1) >= 2
> y <- y[keep, ]
```

The filtering rule doesn't use the experimental design information, yet will keep any gene that is expressed in both the samples for any combinations of genotype and treatment condition.

After filtering, there are 12347 genes remaining and most of the counts are greater than zero:

```
> dim(y)
[1] 12347     9
> head(y$counts)
          WT.N4.rep1 WT.N4.rep2 WT.V4.rep1 WT.V4.rep2 KO.N4.rep1
27395            305        291        430        499        599
18777            510        527        653        642        404
21399            333        361        445        608        424
108664           194        124        230        281        264
12421            326        355        158        210        193
100504079         15         15          3         10         23
          KO.N4.rep2 KO.N4.rep3 KO.V4.rep1 KO.V4.rep2
27395            702        895        785        671
18777            888        724        585        544
21399            710        806        771        572
108664           398        444        334        340
12421            388        263        175        237
100504079         36         10          5         10
```

It is also useful to compute relative scaling factors for the libraries by

```
> y <- calcNormFactors(y)
> y$samples
           group lib.size norm.factors
WT.N4.rep1    WT  5038159        1.033
WT.N4.rep2    WT  4966457        1.013
WT.V4.rep1    WT  5026320        0.964
WT.V4.rep2    WT  4665370        0.986
KO.N4.rep1    KO  4703442        1.009
KO.N4.rep2    KO  6975408        1.015
KO.N4.rep3    KO  7271163        1.039
```

```
KO.V4.rep1    KO  4726829        0.931
KO.V4.rep2    KO  4995218        1.016
```

The `calcNormFactors` function returns the DGEList data argument back with only the norm.factors changed. The scaling factors here represent compositional differences between the shape of the count distributions for the samples. The normalization factors multiply to unity. Factors below 1 indicate that an excessive number of fragments have been assigned to a small number of very highly expressed genes in that library, meaning that less sequencing depth is available for the remaining genes [21].

4.6 Gene Annotation

The summarized counts from Rsubread include Entrez Gene IDs as rownames. The Entrez IDs link to gene-specific information from the NCBI database [12]. To get more details such as gene symbol and chromosome number, we use the annotation file 'Mus_musculus.gene_info' obtained from the NCBI website (`ftp://ftp.ncbi.nih.gov/gene/DATA/GENE_INFO/Mammalia`).

```
> anno <- read.delim(file="Mus_musculus.gene_info", header=FALSE, skip=1)
```

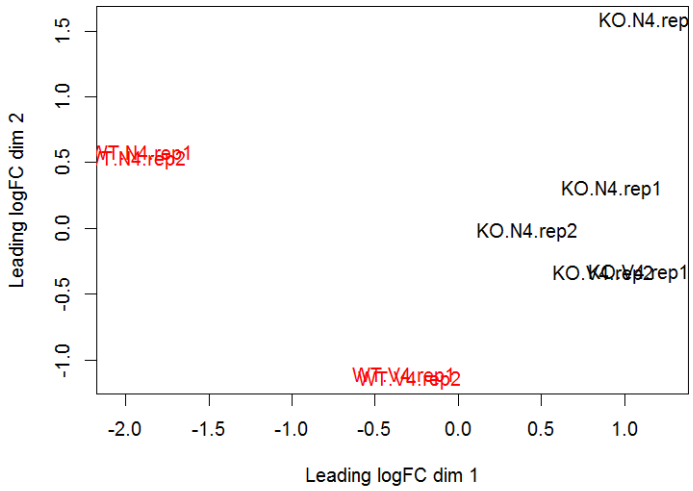
We add selected annotation columns to the DGEList object:

```
> m <- match(rownames(y), anno[,2])
> y$genes <- anno[m, c(2,3,7)]
> colnames(y$genes) <- c("GeneID", "Symbol", "Chr")
> head(y$genes)
      GeneID  Symbol Chr
7060   27395  Mrpl15   1
4165   18777  Lypla1   1
5899   21399   Tcea1   1
24191 108664 Atp6v1h   1
625    12421  Rb1cc1   1
```

4.7 Data Exploration

A multiple dimensional scaling (MDS) plot can be used to check the dissimilarities among the samples:

```
> plotMDS(y, col=as.numeric(targets$Genotype))
```



`plotMDS` is a generic function defined in the limma package with a method defined for DGEList objects. The distance between each pair of samples is calculated as the *leading fold change*, defined as the root-mean-square of the largest 500 $\log_2$-fold changes between that pair of samples. Samples are well separated by the genotype condition (i.e., IRF4 wild-type and knock-out) in the first dimension. A separation by the affinity peptide level (N4 and V4) is also observed in the second dimension. All the replicates are close to each other except for the ones in the IRF4 knock-out (KO) with high-affinity peptides (N4).

4.8 The Design Matrix

We create a design matrix to capture all the experimental information. In this case study, the IRF4 genotype conditions (KO and WT) and the affinity peptide levels (N4 and V4) divide the data into four separate groups. The design matrix can be constructed using the `model.matrix` function as described below.

```
> fac <- paste(targets$Genotype, targets$Treatment, sep=".")
> fac <- factor(fac)
> design <- model.matrix(~0+fac)
> colnames(design) <- levels(fac)
> design
  KO.N4 KO.V4 WT.N4 WT.V4
1     0     0     1     0
2     0     0     1     0
3     0     0     0     1
4     0     0     0     1
5     1     0     0     0
6     1     0     0     0
7     1     0     0     0
```

```
8     0     1     0     0
9     0     1     0     0
attr(,"assign")
[1] 1 1 1 1
attr(,"contrasts")
attr(,"contrasts")$fac
[1] "contr.treatment"
```

We use this simple group-mean parametrization instead of a classic factorial model because it allows contrasts between the groups to be extracted in a simple and transparent way.

4.9 Estimating Dispersions

Now we can proceed to dispersion estimation. The `estimateDisp` function implements the weighted likelihood empirical Bayes strategy described earlier in this chapter. It takes the data object and the design matrix as arguments, and inserts the common, trended and genewise (tagwise) dispersions into the data object:

```
> y <- estimateDisp(y, design)
```

The common dispersion of 0.051 is equivalent to a overall BCV of 23%:

```
> y$common.dispersion
[1] 0.051
```

The gene-specific dispersions vary between 0.024 and 1.1:

```
> summary(y$tagwise.dispersion)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  0.024   0.034   0.046   0.065   0.073   1.100
```

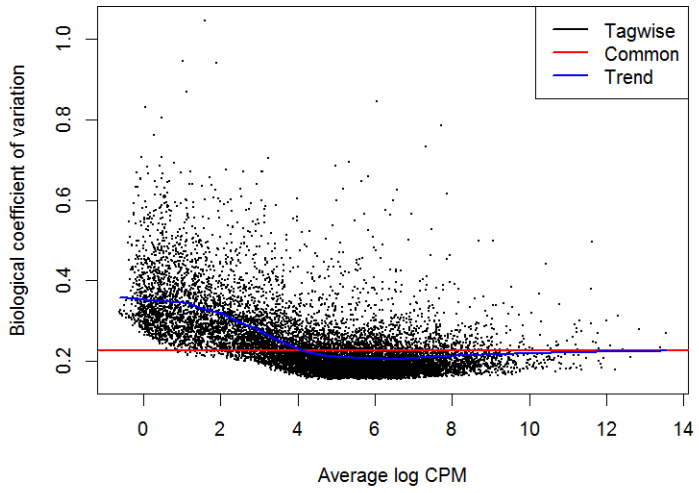
The estimated prior degrees of freedom for this dataset is 6.9:

```
> y$prior.df
[1] 6.9
```

This can be compared to the residual degrees of freedom d_g, which is equal to 5 for most genes in this dataset. The prior degrees of freedom is slightly greater than the residual degrees of freedom, meaning that slightly more weight is being given to the global trend rather than the individual gene when estimating each genewise dispersion.

The BCV plot shows the common, trended and genewise dispersions as a function of average logCPM.

```
> plotBCV(y)
```



Recall that the BCV is the square root of the dispersion. Most of the gene-specific BCVs cluster around the BCV trend, which decreases and then asymptotes to a constant value as the gene expression level increases.

4.10 Detecting Differentially Expressed Genes

In this study, one particular comparison of interest is between IRF4 wild-type (WT) cells stimulated with high-affinity peptide (N4) and WT cells stimulated with low-affinity peptide (V4). To find genes that are DE for this comparison, the first step is to fit genewise negative binomial GLMs using the gene-specific dispersions estimated above:

```
> fit <- glmFit(y, design)
```

Then likelihood ratio statistics are computed for the comparison of interest:

```
> lrt <- glmLRT(fit, contrast=c(0,0,1,-1))
```

Here the contrast argument specifies that the third and fourth groups are to be compared.

The `topTags` function collates results for the most significant genes:

```
> topTags(lrt)
Coefficient:  1*WT.N4 -1*WT.V4
      GeneID  Symbol Chr logFC logCPM    LR   PValue      FDR
1505   13813   Eomes   9 -5.70   7.07 225.7 5.29e-51 6.53e-47
10096  60596 Gucy1a3   3  5.88   4.93 179.9 5.06e-41 3.12e-37
2549   16001   Igf1r   7  3.75   4.88 127.4 1.51e-29 6.20e-26
14239  68404    Nrn1  13  4.16   5.41 109.4 1.30e-25 4.01e-22
30622 236915 Arhgef9   X  5.91   3.38  98.6 3.09e-23 7.62e-20
27600 140795  P2ry14   3 -3.86   4.70  92.9 5.54e-22 9.78e-19
```

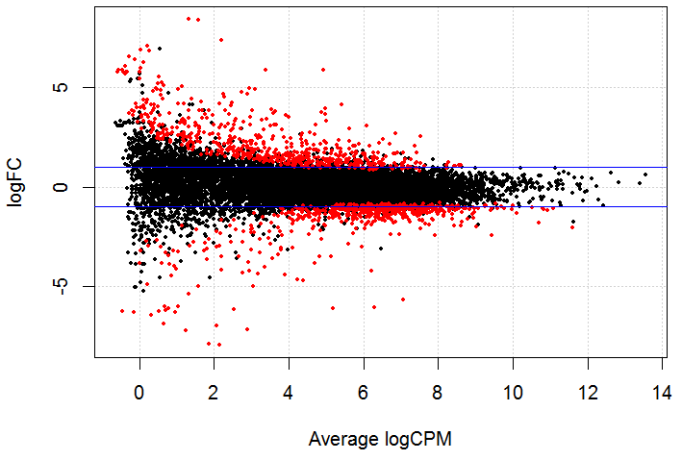
```
3811   18186    Nrp1   8  3.97   4.99  92.9 5.54e-22 9.78e-19
2157   14945    Gzmk  13 -3.40   3.72  85.0 2.94e-20 4.54e-17
35406 380797    Ighd  12  3.70   3.59  83.4 6.56e-20 9.00e-17
34084 320407   Klri2   6  3.83   3.63  78.2 9.35e-19 1.15e-15
```

Local false discovery rates (FDR) are calculated using the Benjamini-Hochberg (BH) method [2]. By default, `topTags` displays the top 10 genes, but can be asked to select any number. By ranking all genes, we can see that there are 1181 genes detected as DE at an FDR cutoff of 1%:

```
> tp <- topTags(lrt, n=Inf)
> sum(tp$table$FDR < 0.01)
[1] 1181
```

A smearplot (a form of MA-plot) can be produced to display the DE results graphically:

```
> DE <- tp$table[tp$table$FDR < 0.01,]$GeneID
> plotSmear(lrt, de.tags=DE, cex = 0.4)
> abline(h=c(-1, 1), col="blue")
```



The axes of the plot correspond to the `logCPM` and `logFC` columns of the results table.

4.11 Session Information

The following output shows the R session and package versions used for this case study:

```
> sessionInfo()
R version 3.0.2 (2013-09-25)
Platform: i386-w64-mingw32/i386 (32-bit)
```

```
locale:
[1] LC_COLLATE=English_Australia.1252
[2] LC_CTYPE=English_Australia.1252
[3] LC_MONETARY=English_Australia.1252
[4] LC_NUMERIC=C
[5] LC_TIME=English_Australia.1252

attached base packages:
[1] splines   stats     graphics  grDevices utils     datasets
[7] methods   base

other attached packages:
[1] locfit_1.5-9.1 edgeR_3.4.0    limma_3.18.3  Rsubread_1.12.6

loaded via a namespace (and not attached):
[1] grid_3.0.2      lattice_0.20-24
```


acknowledgement

Thanks to Wei Shi for providing the fragment counts and aligment code for the IRF4 data, and to Davis McCarthy who programmed the original implementation of the loess local likelihood trend described in Section 3.3.

References

[1] Anders, S., Huber, W.: Differential expression analysis for sequence count data. Genome Biology **11**(10), R106 (2010)

[2] Benjamini, Y., Hochberg, Y.: Controlling the false discovery rate: a practical and powerful approach to multiple testing. Journal of the Royal Statistical Society: Series B **57**, 289–300 (1995)

[3] Chen, Y.: Differential expression analysis of complex RNA-seq experiments. Ph.D. thesis, University of Melbourne (2013)

[4] Cox, D.R., Reid, N.: Parameter orthogonality and approximate conditional inference. Journal of the Royal Statistical Society, Series B **49**, 1–39 (1987)

[5] Efron, B.: Robbins, empirical Bayes and microarrays. The Annals of Statistics **31**(2), 366–378 (2003)

[6] Efron, B., Morris, C.: Stein's estimation rule and its competitors—an empirical Bayes approach. Journal of the American Statistical Association **68**(341), 117–130 (1973)

[7] Gentleman, R.C., Carey, V.J., Bates, D.M., Bolstad, B., Dettling, M., Dudoit, S., Ellis, B., Gautier, L., Ge, Y., Gentry, J., Hornik, K., Hothorn, T., Huber, W., Iacus, S., Irizarry, R., Leisch, F., Li, C., Maechler, M., Rossini, A.J., Sawitzki, G., Smith, C., Smyth, G.K., Tierney, L., Yang, J.Y., Zhang, J.: Bioconductor: open software development for computational biology and bioinformatics. Genome biology **5**(10), R80 (2004)

[8] Jørgensen, B.: The theory of dispersion models. Chapman & Hall, London (1997)

[9] Liao, Y., Smyth, G.K., Shi, W.: The Subread aligner: fast, accurate and scalable read mapping by seed-and-vote. Nucleic Acids Research **41**(10), e108 (2013)

[10] Liao, Y., Smyth, G.K., Shi, W.: featureCounts: an efficient general-purpose read summarization program. Bioinformatics **30**, 923–930 (2014)

[11] Lund, S., Nettleton, D., McCarthy, D., Smyth, G.: Detecting differential expression in RNA-sequence data using quasi-likelihood with shrunken dispersion estimates. Statistical Applications in Genetics and Molecular Biology **11**(5), Article 8 (2012)

[12] Maglott, D., Ostell, J., Pruitt, K., Tatusova, T.: Entrez Gene: gene-centered information at NCBI. Nucleic Acids Res **39**, D52–7 (2011)

[13] Man, K., Miasari, M., Shi, W., Xin, A., Henstridge, D., Preston, S., Pellegrini, M., Belz, G., Smyth, G., Febbraio M Kallies, A.: IRF4 is essential for T cell receptor affinity mediated metabolic programming and clonal expansion of T cells. Nature Immunology **14**, 1155–1165 (2013)

[14] Marioni, J.C., Mason, C.E., Mane, S.M., Stephens, M., Gilad, Y.: RNA-seq: An assessment of technical reproducibility and comparison with gene expression arrays. Genome Res. **18**, 1509–1517 (2008)

[15] McCarthy, D.J., Chen, Y., Smyth, G.K.: Differential expression analysis of multifactor RNA-Seq experiments with respect to biological variation. Nucleic Acids Research **40**(10), 4288–4297 (2012)

[16] McCullagh, P., Nelder, J.A.: Generalized Linear Models, 2nd edition edn. Chapman & Hall/CRC, Boca Raton, Florida (1989)

[17] Mortazavi, A., Williams, B.A., McCue, K., Schaeffer, L., Wold, B.: Mapping and quantifying mammalian transcriptomes by RNA-Seq. Nature Methods **5**(7), 621–628 (2008)

[18] Nelder, J.A., Wedderburn, R.W.M.: Generalized linear models. Journal of the Royal Statistical Society. Series A (General) **135**(3), 370–384 (1972)

[19] Pruitt, K., Tatusova, T., Brown, G., Maglott, D.: NCBI Reference Sequences (RefSeq): current status, new features and genome annotation policy. Nucleic Acids Res **40**, D130–5 (2012)

[20] Robinson, M., McCarthy, D., Smyth, G.: edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. Bioinformatics **26**(1), 139–140 (2010)

[21] Robinson, M.D., Oshlack, A.: A scaling normalization method for differential expression analysis of RNA-seq data. Genome Biology **11**(3), R25 (2010)

[22] Robinson, M.D., Smyth, G.K.: Moderated statistical tests for assessing differences in tag abundance. Bioinformatics **23**(21), 2881–2887 (2007)

[23] Robinson, M.D., Smyth, G.K.: Small-sample estimation of negative binomial dispersion, with applications to SAGE data. Biostatistics **9**(2), 321–332 (2008)

[24] Shendure, J., Aiden, E.L.: The expanding scope of DNA sequencing. Nature Biotechnology **30**(11), 1084–1094 (2012)

[25] Smyth, G.: Linear models and empirical Bayes methods for assessing differential expression in microarray experiments. Statistical applications in genetics and molecular biology **3**(1), Article 3 (2004)

[26] Smyth, G., Verbyla, A.: Adjusted likelihood methods for modelling dispersion in generalized linear models. Environmetrics **10**(6), 695–709 (1999)

[27] Wang, X.: Approximating Bayesian inference by weighted likelihood. Canadian Journal of Statistics **34**(2), 279–298 (2006)

[28] Wang, Z., Gerstein, M., Snyder, M.: RNA-Seq: a revolutionary tool for transcriptomics. Nat. Rev. Genet. **10**(1), 57–63 (2009)