BIOINFORMATICS ORIGINAL PAPER

Vol. 29 no. 17 2013, pages 2146–2152
doi:10.1093/bioinformatics/btt350

Gene expression

Advance Access publication July 2, 2013

Data-based filtering for replicated high-throughput transcriptome sequencing experiments

Andrea Rau[1,2,*], Mélina Gallopin[1,2], Gilles Celeux[3] and Florence Jaffrézic[1,2]

[1]INRA, UMR1313 Génétique animale et biologie intégrative, 78352 Jouy-en-Josas, France, [2]AgroParisTech, UMR1313 Génétique animale et biologie intégrative, 75231 Paris 05, France and [3]Inria Saclay - Île-de-France, 91405 Orsay, France

Associate Editor: Inanc Birol

ABSTRACT

Motivation: RNA sequencing is now widely performed to study differential expression among experimental conditions. As tests are performed on a large number of genes, stringent false-discovery rate control is required at the expense of detection power. Ad hoc filtering techniques are regularly used to moderate this correction by removing genes with low signal, with little attention paid to their impact on downstream analyses.

Results: We propose a data-driven method based on the Jaccard similarity index to calculate a filtering threshold for replicated RNA sequencing data. In comparisons with alternative data filters regularly used in practice, we demonstrate the effectiveness of our proposed method to correctly filter lowly expressed genes, leading to increased detection power for moderately to highly expressed genes. Interestingly, this data-driven threshold varies among experiments, highlighting the interest of the method proposed here.

Availability: The proposed filtering method is implemented in the `R` package `HTSFilter` available on Bioconductor.

Contact: andrea.rau@jouy.inra.fr

Supplementary information: Supplementary data are available at *Bioinformatics* online.

Received on November 20, 2012; revised on May 24, 2013; accepted on June 17, 2013

1 INTRODUCTION

During the past 5 years, next-generation high-throughput sequencing (HTS) technology has become an essential tool for genomic and transcriptomic studies. In particular, the use of HTS technology to directly sequence the transcriptome, known as RNA sequencing (RNA-seq), has revolutionized the study of gene expression by opening the door to a wide range of novel applications. Unlike microarray data, which are continuous, RNA-seq data represent highly heterogeneous counts for genomic regions of interest (typically genes) and often exhibit zero-inflation and a large amount of overdispersion among biological replicates; as such, a great deal of methodological research (e.g. Anders and Huber, 2010; Dillies *et al.*, 2012; Robinson *et al.*, 2010) has recently focused on appropriate normalization and analysis techniques that are adapted to the characteristics of RNA-seq data; see Oshlack *et al.* (2010) for a review of RNA-seq technology and analysis procedures.

*To whom correspondence should be addressed.

As with data arising from previous technologies, such as microarrays or serial analysis of gene expression, HTS data are often used to conduct differential analyses. In recent years, several approaches for gene-by-gene tests using gene-level HTS data have been proposed, with the most popular making use of Poisson (Wang *et al.*, 2010), overdispersed Poisson (Auer and Doerge, 2011) or negative binomial distributions (Anders and Huber, 2010; Robinson *et al.*, 2010). Because a large number of hypothesis tests are performed for gene-by-gene differential analyses, the obtained *P*-values must be adjusted to address the fact that many truly null hypotheses will produce small *P*-values simply by chance; to address this multiple testing problem, several well-established procedures have been proposed to adjust *P*-values to control various measures of experiment-wide false positives, such as the false-discovery rate. Although such procedures may be used to control the number of false positives that are detected, they are often at the expense of the power of an experiment to detect truly differentially expressed (DE) genes, particularly as the number of genes in a typical HTS dataset may be in the thousands or tens of thousands. To reduce this impact, several authors in the microarray literature have suggested the use of data filters to identify and remove genes that appear to generate an uninformative signal (Bourgon *et al.*, 2010) and have no or little chance of showing significant evidence of differential expression; only hypotheses corresponding to genes that pass the filter are subsequently tested, which in turn tempers the correction needed to adjust for multiple testing.

In recent work, Bourgon *et al.* (2010) advocate for the use of *independent data filtering*, in which the filter and subsequent test statistic pairs are marginally independent under the null hypothesis, and the dependence structure among tests remains largely unchanged pre- and post-filter, ensuring that post-filter *P*-values are true *P*-values. For such an independent filter to be effective, it must be positively correlated with the test statistic under the alternative hypothesis; indeed, it is this correlation that leads to an increase in detection power after filtering. In addition, Bourgon *et al.* demonstrate that non-independent filters for which dependence exists between the filter and test statistic (e.g. making use of condition labels to filter genes with average expression in at least one condition less than a given threshold), can in some cases lead to a loss of control of experiment-wide error rates.

Several *ad hoc* data filters for RNA-seq data have been used in recent years, including filtering genes with a total read count smaller than a given threshold (Sultan *et al.*, 2008) and filtering genes with at least one zero count in each experimental condition (Bottomly *et al.*, 2011); however, selecting an arbitrary threshold

value to filter genes in this way does not account for the overall sequencing depth or variability of a given experiment. One exception to these *ad hoc* filters is the work of Ramsköld *et al.* (2009), in which a comparison between expression levels of exonic and intergenic regions was used to find a threshold for detectable expression above background in various human and mouse tissues, where expression was estimated as Reads Per Kilobase per Million mapped reads (RPKM) (Mortazavi *et al.*, 2008). The threshold of 0.3 RPKM identified in this work has in turn been applied to several other studies (e.g. Cánovas *et al.*, 2010; Łabaj *et al.*, 2011; Sam *et al.*, 2011). However, to our knowledge, although filters for read counts are routinely used in practice, little attention has been paid to the choice of the type of filter or threshold used or its impact on the downstream analysis.

In this article, we propose a novel data-based procedure to choose an appropriate filtering threshold based on the calculation of a similarity index among biological replicates for read counts arising from replicated high-throughput transcriptome sequencing data. This technique provides an intuitive data-driven way to filter RNA-seq data and to effectively remove genes with low constant expression levels. Our proposed filtering threshold may be useful in a variety of applications for RNA-seq data, including differential expression analyses, clustering and co-expression analyses, and network inference.

2 METHODS

2.1 Types of filters used for RNA-seq data

Data filters are routinely used in practice for differential analyses of RNA-seq data. Most such filters are applied to data that have been normalized in some way, rather than directly to the raw counts, to account for systematic inter-sample biases typical of RNA-seq data, e.g. differences in library size (Anders and Huber, 2010; Robinson and Oshlack, 2010) or GC content (Hansen *et al.*, 2012; Risso *et al.*, 2011). In particular, the Trimmed M-Means library size normalization (Robinson and Oshlack, 2010) and the normalization included in the `DESeq` Bioconductor package (Anders and Huber, 2010) have been found to be robust methods to correct for library size biases, even in the presence of widely different library compositions (Dillies *et al.*, 2012).

We consider two broad categories of filters for RNA-seq data based on the filtering criterion used: mean-based filters and maximum-based filters. We note that although variance-based filters are routinely used for microarray data (Bourgon *et al.*, 2010), they have not been applied to RNA-seq data; this is likely due to the small number of replicates available in most RNA-seq datasets (and thus, the difficulty in obtaining accurate estimates of per-gene variances) and the fact that the variance is assumed to be a function of the mean under a negative binomial model.

2.1.1 Mean-based filters In mean-based filters, genes with mean normalized counts across all samples less than or equal to a pre-specified cutoff are filtered from the analysis. Some authors (Sultan *et al.*, 2008) have also proposed filtering genes with a total read count less than or equal to a given threshold s; we note that this is equivalent to mean-based filters for threshold s divided by the number of samples.

In addition to normalized counts, we also consider mean-based filters for the RPKM (Mortazavi *et al.*, 2008) measure, which was initially proposed to simultaneously normalize RNA-seq data for biases due to library size and gene length. However, we note that it has been shown that counts, rather than RPKM values, are preferable for the differential analysis of RNA-seq data (Oshlack and Wakefield, 2009). For this reason, after filtering genes with a RPKM mean filter, raw counts are used for the subsequent differential analysis. A comparison of differential analysis methods developed for counts and RPKM values is beyond the scope of this work.

2.1.2 Maximum-based filters In maximum-based filters, genes with maximum normalized counts across all samples less than or equal to a pre-specified threshold are filtered from the analysis. As aforementioned, in addition to normalized counts, we also consider maximum-based filters for RPKM values, which we refer to as a RPKM maximum filter.

A generalization of the maximum-based filter has also been proposed in the `edgeR` analysis pipeline (Robinson *et al.*, 2010) based on counts per million (CPM), calculated as the raw counts divided by the library sizes and multiplied by one million. Genes with a CPM value less than a given cutoff (e.g. 1 or 100) in more samples (ignoring condition labels) than the size of the smallest group are subsequently filtered from the analysis. To distinguish this approach from the other maximum-based filters, we refer to this strategy as a CPM filter.

We note that maximum-based filters are not independent filters as described by Bourgon *et al.* (2010); in particular, for extremely large filtering thresholds, maximum-based filters do not guarantee control of the Type I error rate if P-values are computed using the pre-filter null distribution. For the threshold values typically used in practice (e.g. based on a quantile, or the data-based threshold proposed later in the text), this is usually not a concern (see Supplementary Fig. S28). Although it may be difficult to verify that conditional and unconditional P-value distributions coincide for real data, it may be useful to examine histograms of each (e.g. as shown in Fig. 3).

2.2 A data-based threshold for maximum-based filters

For each of the filter types previously defined, a biologically pertinent cutoff (or alternatively, number of genes to be filtered) must be chosen; in practice, arbitrary thresholds are routinely used with little or no discussion of their impact on the downstream analysis. To address this issue, we propose a data-based choice for the threshold to be used in maximum-based filters. The main idea underlying this choice is to identify the threshold that maximizes the filtering similarity among replicates, i.e. one where most genes tend to either have normalized counts less than or equal to the cutoff in all samples (i.e. filtered genes) or greater than the cutoff in all samples (i.e. non-filtered genes).

To define this filtering similarity, we begin with some notation. Let y_{gj} represent the observed normalized read count (e.g. after scaling raw counts by library size) for gene g in sample j, and let $\mathcal{C}(j)$ represent the experimental condition of sample j, with $g \in \{1, \ldots, G\}$ and $j \in \{1, \ldots, J\}$. Typically, in the context of differential analyses, the number of conditions is equal to 2. We denote the full vector of read counts in a given sample as $\mathbf{y}_j$. We now wish to define a *similarity index* between a pair of replicates within the same condition $\{(\mathbf{y}_j, \mathbf{y}_{j'}) : \mathcal{C}(j) = \mathcal{C}(j')\}$ after binarizing the data for a fixed cutoff s (1 if $y_{gj} > s$ and 0 otherwise). We note that a variety of similarity indices have been proposed since the early 1900s; however, in a comparison among a set of similarity indices (see the Supplementary Materials), we found the Jaccard index (Jaccard, 1901) to be simple, natural and easy to interpret for the analysis of HTS data. This index is defined as follows:

$$J_s(\mathbf{y}_j, \mathbf{y}_{j'}) = \frac{a}{a+b+c} \tag{1}$$

where a, b and c are defined in Table 1. We note that $J_s(\mathbf{y}_j, \mathbf{y}_{j'})$ takes on values from 0 (dissimilar) to 1 (similar). Because multiple replicates and/or conditions are typically available in HTS experiments, we extend the definition of the pairwise Jaccard index in Equation (1) to a global

Table 1. Definition of the constants used to calculate the Jaccard similarity index for a pair of samples j and j' and a given threshold s

	Sample j	
	Normalized counts $>s$	Normalized counts $\leq s$
Sample j'		
Normalized counts $>s$	a	b
Normalized counts $\leq s$	c	d

Note: The constant a represents the number of genes with normalized counts greater than s in both samples j and j' and so on.

Jaccard index by averaging the indices calculated over all pairs in each condition:

$$J_s^\star(\mathbf{y}) = \text{mean}\left\{J_s(\mathbf{y}_j, \mathbf{y}_{j'}) : j<j' \text{ and } \mathcal{C}(j) = \mathcal{C}(j')\right\}. \quad (2)$$

Using the global Jaccard index defined in Equation (2) as a measure of similarity, we now wish to identify the cutoff $s^\star$ for normalized counts that corresponds to the greatest similarity possible among replicates, i.e. the value of s corresponding to the maximum value of the global Jaccard index:

$$s^\star = argmax_s J_s^\star(\mathbf{y}). \quad (3)$$

In practice, for the calculation of the data-based global filtering threshold in Equation (3), we calculate the value of the global Jaccard index in Equation (2) for a fixed set of threshold values and fit a loess curve (Cleveland, 1979) through the set of points; the value of $s^\star$ is subsequently set to be the maximum of these fitted values.

Once the data-driven filter threshold for normalized counts $s^\star$ has been identified, the subsequent steps to be taken may change for different applications. To perform an analysis of differential expression between two experimental conditions, we propose using this threshold $s^\star$ in a maximum-based filter, as defined in Section 2.1.2; in the following, we refer to this technique as the *Jaccard filter*.

2.3 The Bioconductor package `HTSFilter`

The proposed filtering method is implemented in the `HTSFilter` package, currently available as part of the Bioconductor project (Gentleman *et al.*, 2004) within the statistical environment `R` (R Development Core Team, 2009). The `HTSFilter` package is compatible with a variety of data classes and analysis pipelines, including `matrix` and `data.frame` objects, the S4 class `CountDataSet` in the `DESeq` pipeline (Anders and Huber, 2010) and the S3 class `DGEList` in the `edgeR` pipeline (Robinson *et al.*, 2010). A package vignette describes the use of the `HTSFilter` package within each of these pipelines.

3 RESULTS

In the following, we apply the normalization approach proposed by Anders and Huber (2010) for mean- and maximum-based filters, although other types of normalization may be appropriate for some data. For gene-by-gene comparisons between two conditions, we illustrate the use of the proposed filter in conjunction with the model proposed in the `DESeq` Bioconductor package (version 1.8.3), which has been developed to model count data with a small number of replicates in the presence of overdispersion (Anders and Huber, 2010); P-values are adjusted for multiple testing using the Benjamini–Hochberg procedure (Benjamini and Hochberg, 1995) to control the false-discovery rate. We note that the filtering method proposed here may also be used in conjunction with other popular methods, e.g. `edgeR` (Robinson *et al.*, 2010). See the Supplementary Materials for additional discussion of the normalization and statistical testing methods used in this work.

In practice, there may be some question about the appropriate point in the analysis pipeline to apply data filters: Should normalized data first be filtered, then normalization factors re-estimated and the model fit (i.e. mean and dispersion parameters estimated)? Should normalization factors and model parameters be estimated based on the full data, and the data filtered only at the end of the analysis pipeline? The difference between the two options is non-trivial, particularly as the differential analysis approaches implemented in the `DESeq` and `edgeR` packages both borrow information across genes (whether all or only those passing the filter) to obtain per-gene parameter estimates. In this work, we present results based on the application of filters applied as late in the pipeline as possible, i.e. after library size and dispersion parameter estimation; a more detailed discussion of this issue is included in the Supplementary Materials.

3.1 Description of data

We applied our proposed Jaccard index filter, in addition to the alternative filter types described earlier in the text, on the following data:

- **Sex-specific expression of liver cells in human.** Sultan *et al.* (2008) obtained high-throughput transcriptome sequencing data from a human embryonic kidney and a B cell line, with two biological replicates each. The raw read counts and phenotype tables were obtained from the ReCount online resource (Frazee *et al.*, 2011).
- **Differential striatal expression between inbred mouse strains**. Bottomly *et al.* (2011) performed RNA-seq experiments for 10 biological replicates of the C57BL/6J inbred mouse strain and 11 for the DBA/2J strain, and the results were compared with those arising from two different microarray platforms. The raw read counts and phenotype tables were obtained from the ReCount online resource (Frazee *et al.*, 2011).
- **MiTF repression in a human melanoma cell line**. Strub *et al.* (2011) obtained HTS data to compare gene expression in a melanoma cell line expressing the Microphtalmia Transcription Factor to one in which small interfering RNAs were used to repress Microphtalmia Transcription Factor, with three biological replicates in each group. The raw read counts and phenotype tables are available in the Supplementary Materials of Dillies *et al.* (2012).
- **Simulated data**. To investigate the effect of the various filtering methods on downstream results, we developed a simulation framework as described in Section 3.3.

For the three real datasets (described in further detail in Supplementary Table S1), gene annotations for *Mus musculus* (NCBIM37) and *Homo sapiens* (GRCh37.p7) were obtained

from Ensembl version 67 (Birney *et al.*, 2004) using the Biomart tool (Kasprzyk *et al.*, 2004), and the length of each gene in base pairs was calculated. In the Bottomly, Sultan and Strub data, ~4, 2 and 5% of the genes, respectively, had been retired from Ensembl; for these genes, RPKM-based filters were not used. Readers wishing to examine the complete analyses in detail may find a Sweave document containing commented `R` code for the analyses of each of the datasets in the Supplementary Materials.

3.2 Comparison of filters on real data

For the three real datasets, differential analyses were performed using a negative binomial model (Anders and Huber, 2010) for unfiltered data and after filtering the data using the techniques described earlier in the text. For the Jaccard index filter, the global Jaccard index in Equation (2) was calculated for a range of threshold values s, yielding a largely unimodal distribution of values for all datasets considered (Fig. 1 and Supplementary Figs S11 and S18). We also note that the data-driven threshold values $s^\star$ identified for the Bottomly, Sultan and Strub data were not equal; in the case of the Bottomly data, the threshold for normalized counts was found to be 14.7, whereas this threshold was found to be 11.5 for the Sultan data and 103.5 for the Strub data. These differences in filtering threshold among experiments are due to both sequencing depth and variability within the data; in particular, experiments with greater sequencing depth will tend to have higher filtering thresholds, and those with greater variability will tend to have lower filtering thresholds. Among the data considered in our study, the Strub data have the highest sequencing depth (1.5×10^8) coupled with low intra-condition variability (minimum correlation among replicates equal to 0.98), and they also have the highest threshold considered here. On the other hand, the Sultan data have a much lower sequencing depth (1.8×10^6) and thus have a much lower threshold.

For each dataset, the Jaccard filter was applied with the corresponding data-based threshold calculated earlier in the text. For the alternative mean- and maximum-based filters for normalized counts and RPKM values, cutoffs were chosen based on the 15% quantile of the respective criterion. For the CPM filter, as suggested in the `edgeR` pipeline Robinson *et al.* (2010), genes with a CPM value <1 in more samples than the size of the smallest group are subsequently filtered from the analysis.

Among the filters considered, it may immediately be seen in Figure 2 that in the Strub data, with the exception of the CPM and Jaccard filters, most of the filters considered here appear to be ineffective as they are largely unable to filter genes with low levels of expression and small log-fold changes; a similar phenomenon may be observed in the Bottomly and Sultan data (Supplementary Figs S7 and S14). Although these techniques are thus unlikely to (incorrectly) filter truly DE genes, the number of statistical tests is not markedly reduced, and as such the power to detect differential expression will remain largely unchanged as compared with the unfiltered data. With the exception of the RPKM-based filters, all are able, to some extent, to identify and remove genes contributing to a peak of raw P-values close to one (Fig. 3 and Supplementary Figs S13 and S20), a phenomenon due to the discretization of P-values for small counts; indeed, histograms of raw P-values following the application of most filters appear to be roughly uniformly distributed under the null hypothesis. In addition, we note that the proposed Jaccard filter is able to more effectively remove genes with moderate log base means and small log fold changes (see Fig. 2). Similar conclusions may be drawn from the volcano plots shown in Supplementary Figures S8, S15 and S22.

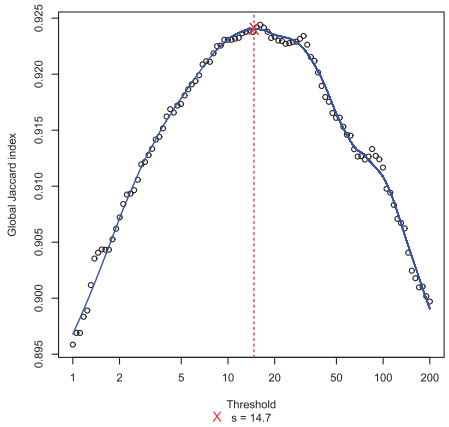

Fig. 1. Global Jaccard index for the Bottomly data calculated for a variety of threshold values for normalized counts, with a loess curve (solid line) superposed and data-driven threshold value (dotted line) equal to $s^\star = 14.7$

It is also of interest to consider the effect of each filter on the number of DE genes identified at various levels of expression; in Figure 4 and Supplementary Figures S5 and S12, we note that in all datasets, the Jaccard filter leads to more discoveries at all but weak levels of expression (i.e. mean expression <10), with this difference being particularly marked for moderate levels of expression (i.e. mean expression greater than 50). We note that a large number of the missed discoveries for the Jaccard filter at low levels of expression correspond to genes with zero read counts in one condition and a small number of read counts in the other; for example, in the Sultan data, 50.3% of the 449 discoveries among genes with mean normalized read counts <10 had zero read counts in one of the two conditions. Among the other filter types, the CPM filter appears to come closest to the Jaccard filter, with the remaining filters performing similarly.

In considering the overlap of genes filtered using each method (Supplementary Figs S9, S16 and S23), it is clear that a large number of genes may be filtered, regardless of the technique used. However, the Jaccard index is better able to filter a large number of weakly expressed genes in all three of the datasets considered here, leading to a more moderate correction for multiple testing; the direct consequence of this is a larger number of discoveries at moderate-to-high levels of expression. To determine whether this advantage is due to the filtering type (i.e. maximum) or the threshold used for each (i.e. using the 15% quantile or the data-based threshold identified by the global Jaccard index), we consider a set of simulation studies in the next section.

3.3 Simulated data

Data were simulated using a negative binomial model, with parameters chosen based on the Bottomly, Sultan and Strub

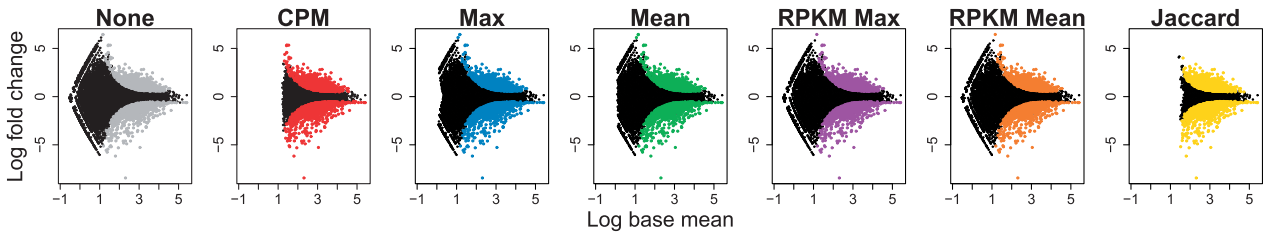

Fig. 2. Log mean expression versus log fold change values for the Strub data. For each filter, genes identified as non-DE are drawn in black, those identified as DE are drawn in color (online) or grey (print), and those filtered from the differential analysis are omitted from the plot. From left to right, the filters are as follows: none, CPM, maximum, mean, RPKM maximum, RPKM mean and maximum using the global Jaccard index threshold

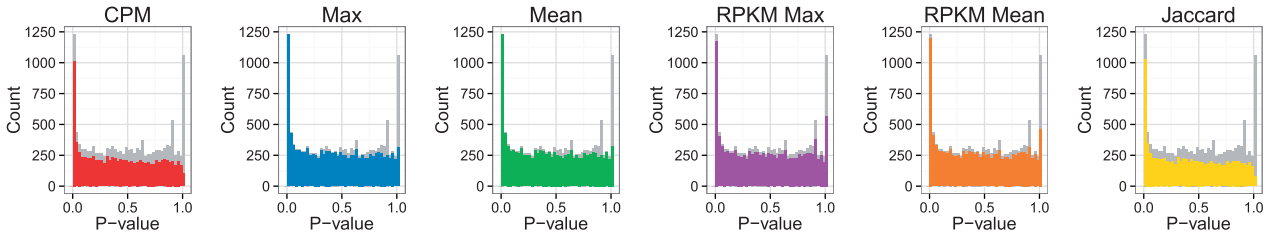

Fig. 3. Histograms of raw *P*-values from a differential analysis of the Bottomly data for a variety of filter types. Histograms in the background represent the raw *P*-values from a differential analysis of the Bottomly data using unfiltered data; histograms in the foreground represent the raw *P*-values from a differential analysis of the data filtered with various filter types. Figure made using the `ggplot2` package (Wickham, 2009)

datasets. Briefly, for each dataset, genes with zero counts in one of the two groups and mean <5 were removed (see Supplementary Table S1). Differential analyses were performed on unfiltered data using the `DESeq` Bioconductor package (Anders and Huber, 2010) as described in the Supplementary Materials. After adjusting raw *P*-values for multiple testing using the Benjamini–Hochberg procedure (Benjamini and Hochberg, 1995), we identified 570, 2515 and 2485 DE genes, respectively, at the 5% significance level in the Bottomly, Sultan and Strub data. In addition, the parametric gamma regressions fitted to the per-gene dispersion estimates α and gene expression means μ (Supplementary Fig. S25) were identified for each dataset:

$$\alpha_{\text{Bottomly}}(\mu) = 0.03 + 0.72/\mu, \quad (4)$$

$$\alpha_{\text{Sultan}}(\mu) = 0.01 + 1.23/\mu, \quad (5)$$

$$\alpha_{\text{Strub}}(\mu) = 0.03 + 13.35/\mu. \quad (6)$$

Subsequently, simulation parameters for each dataset were fixed as follows. For genes identified as being DE, means for each condition were set to be the empirically calculated means from each condition from the normalized data; for genes not identified as being DE, means for each condition were both set to be the global mean (across both conditions) from the normalized data. This allows genes to be simulated as DE across the full range of mean expression values (Supplementary Fig. S27). Per-gene dispersion parameters were set to be the fitted values from the regression equations defined in Equations (4)–(6) as a function of the overall mean for each gene; for the simulations based on the Bottomly and Sultan data, dispersion parameters for genes with overall mean expression <20 were fixed to be equal to 10^{-10} to simulate negligible overdispersion, as shot noise appears to dominate biological noise at low expression levels in these data (Supplementary Fig. S26). Once these parameters were fixed for each gene, a negative binomial model was used to simulate 300 individual datasets each for the parameters based on the Bottomly, Sultan and Strub data, with 21 samples (10 in one condition and 11 in the other), 4 samples (2 in each condition) and 6 samples (3 in each condition), respectively. Real lengths corresponding to the genes in each dataset were used for the calculation of RPKM values.

3.4 Comparison of filters on simulated data

To assess performance on simulated data, we focus on the sensitivity of detecting DE genes after each data filter, defined as the proportion of truly DE genes detected among all truly DE genes. In addition, we construct Receiver Operating Characteristic (ROC) curves of each filter, based on the *filtering sensitivity*, defined as the proportion of correctly unfiltered genes (i.e. DE and unfiltered) among all truly DE genes, and the *filtering specificity*, defined as the proportion of correctly filtered genes (i.e. non-DE and filtered) among all non-DE genes.

In Figure 5, we note that the sensitivity to detect DE genes greatly varies among the filtering types for simulations based on the Strub data, as well as among different thresholds within each filtering type; in addition, the RPKM maximum and RPKM mean filters actually lead to lower detection power than unfiltered data. Similar results may be seen for the simulations based on the Bottomly and Sultan data in Supplementary Figure S30. For the simulation setting shown in Figure 5, larger thresholds appear to yield the highest detection sensitivity (i.e. maximum or mean-based filters for normalized counts using the 30% quantile as a threshold). However, this trend is reversed for the other

simulation settings, where smaller cutoffs (i.e. cutoffs of 5 or 15% for the maximum or mean-based filters) lead to higher detection sensitivity. This highlights the difficulty in pre-selecting a fixed threshold for a given filtering method, as well as the advantage of our proposed Jaccard filter. Indeed, the Jaccard filter appears to lead to high detection sensitivity for all simulations with the exception of those based on the Bottomly data; for these data, because many weakly expressed genes were simulated to be DE (see Supplementary Fig. S27), unfiltered data had the highest sensitivity.

To assess the role of the choice of threshold for each filter type, we constructed ROC curves of the filtering sensitivity and specificity over varying cutoffs in Figure 6 and Supplementary Figure S29. We note that the mean and maximum RPKM-based filters tend to have lower filtering sensitivity than the others, and the maximum filters (for both normalized counts and CPM values) tend to be similar; however, for all simulation settings, the maximum filter for normalized counts has a slight advantage over that for CPM values. In other words, regardless of the threshold used for filtering, maximum-based filters appear to more effectively filter non-DE genes than the remaining methods. Finally, we note also that the data-based threshold using the global Jaccard index appears to find a good compromise between filtering sensitivity and filtering specificity.

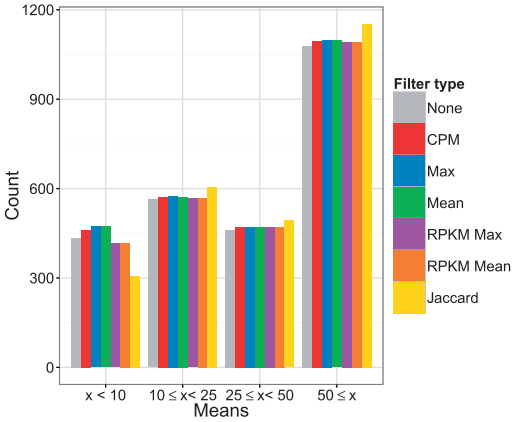

Fig. 4. Number of DE genes detected in the Sultan data, categorized by normalized base mean for each filter type (from left to right in each bin: none, CPM, maximum, mean, maximum RPKM, mean RPKM, and Jaccard). Figure made using the `ggplot2` package (Wickham, 2009)

4 CONCLUSIONS AND DISCUSSION

Data filtering has proven to be of great practical importance for the differential analysis of high-throughput microarray and RNA-seq data by identifying and removing genes with uninformative signal before testing. In recent years, many *ad hoc* procedures have been used to filter RNA-seq data, such as

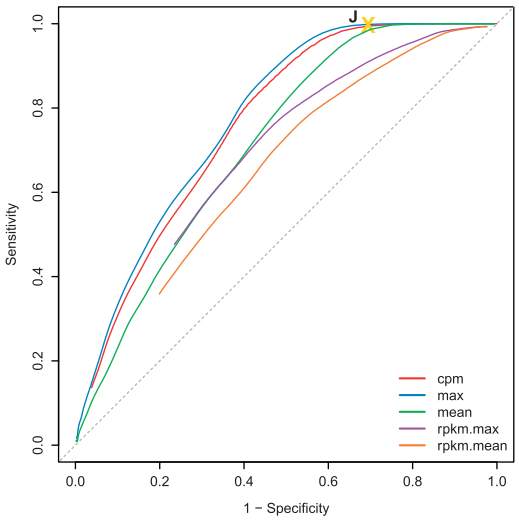

Fig. 6. ROC curves (averaged over 300 datasets) for the filtering performance on simulated data based on the Sultan data for the CPM, maximum, mean, maximum RPKM and mean RPKM filters over a range of cutoffs. The yellow cross labeled with a 'J' corresponds to the filtering sensitivity and specificity for the data-based threshold chosen via the global Jaccard index

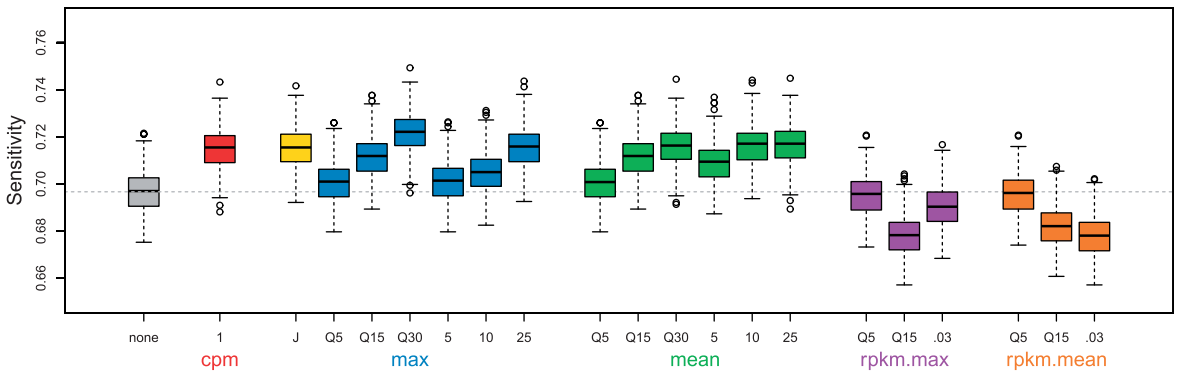

Fig. 5. Sensitivity (over simulated 300 datasets) to detect DE genes for a variety of filter types and cutoffs, with simulation parameters based on the Strub data. None, no filter; CPM, genes with a CPM less than one in more than half the samples are filtered; Max, maximum-based filter, using the threshold based on the Jaccard index (J), quantiles (5, 15 and 30%), or values (5, 10, 25); Mean, mean-based filter, using the threshold based on quantiles (5, 15 and 30%), or values (5, 10, 25); RPKM.max, maximum RPKM filter, using the threshold based on quantiles (5 and 15%) or the value 0.3; RPKM.mean, maximum RPKM filter, using the threshold based on quantiles (5 and 15%) or the value 0.3

filtering genes with a total or mean normalized read count less than a specified threshold. However, despite its impact on the downstream analyses, no clear recommendations have yet been provided concerning the choice of filtering technique.

Among the filter types considered here, we have found that filters using the maximum normalized count appear to be best able to correctly filter genes with low levels of expression and little evidence of differential expression. In addition, we have proposed a method to calculate a data-driven and non-prefixed filtering threshold value for normalized counts from replicated RNA-seq data, based on the global Jaccard similarity index. In particular, our proposed filtering technique was found to remove from the analysis a large number of genes with little or no chance of showing evidence of differential expression, and therefore to increase detection power at moderate-to-high levels of expression through a moderation of the correction for multiple testing. As such, we recommend that genes with a normalized count value less than this data-driven threshold in all samples be filtered from subsequent differential analyses. We emphasize that the data-driven threshold value may vary greatly among RNA-seq experiments owing to differences in sequencing depth and intra-condition variability (see Supplementary Fig. S31 for the data-based thresholds calculated on three additional RNA-seq datasets); as such, the threshold value must be recalculated for each dataset of interest.

The impact of the proposed filtering method has been investigated here in the context of differential analyses. We anticipate that it will also be useful in a variety of other applications, e.g. detecting genes that are specifically expressed in one condition or ubiquitously expressed across several conditions, which is often a crucial biological question. In addition, we anticipate that such filtering will be useful, for example, in co-expression or network reconstruction analyses to remove genes with low constant levels of expression. Finally, we note that although this filter was presented here for the analysis of RNA-seq data, it can readily be applied to other types of replicated HTS data, such as CHIP-seq data.

ACKNOWLEDGEMENTS

The authors thank the three anyonymous reviewers for their valuable comments and suggestions that helped to considerably improve the quality of the manuscript.

Funding: This work was supported by the French National Research Agency (Agence nationale de la recherche) [ANR-09-GENM-006, Biocart project].

Conflict of Interest: none declared.

REFERENCES

Anders,S. and Huber,W. (2010) Differential expression analysis for sequence count data. *Genome Biol.*, **11**, 1–28.

Auer,P.L. and Doerge,R.W. (2011) A two-stage Poisson model for testing RNA-seq data. *Stat. Appl. Genet. Mol. Biol.*, **10**, 1–26.

Benjamini,Y. and Hochberg,Y. (1995) Controlling the false discovery rate: a practical and powerful approach to multiple testing. *J. R. Stat. Soc. Ser. B*, **57**, 289–300.

Birney,E. *et al.* (2004) An overview of ensembl. *Genome Res.*, **14**, 925–928.

Bottomly,D. *et al.* (2011) Evaluating gene expression in C57BL/GJ and DBA/2J mouse striatum using RNA-seq and microarrays. *PLoS One*, **6**, e17820.

Bourgon,R. *et al.* (2010) Independent filtering increases detection power for high-throughput experiments. *PNAS*, **107**, 9546–9551.

Cánovas,A. *et al.* (2010) SNP discovery in the bovine milk transcriptome using RNA-seq technology. *Mamm. Genome*, **21**, 592–598.

Cleveland,W.S. (1979) Robust locally weighted regression and smoothing scatterplots. *J. Am. Stat. Assoc.*, **74**, 829–836.

Dillies,M.A. *et al.* (2012) A comprehensive evaluation of normalization methods for Illumina high-throughput RNA sequencing data analysis. *Brief. Bioinformatics*, [Epub ahead of print, September 17, 2012].

Frazee,A.C. *et al.* (2011) ReCount: a multi-experiment resource of analysis-ready RNA-seq gene count datasets. *BMC Bioinformatics*, **12**, 449.

Gentleman,R.C. *et al.* (2004) Bioconductor: open software development for computational biology and bioinformatics. *Genome Biol.*, **5**, 2004.

Hansen,K.D. *et al.* (2012) Removing technical variability in RNA-seq data using conditional quantile normalization. *Biostatistics*, **13**, 204–216.

Jaccard,P. (1901) Étude comparative de la distribution florale dans une portion des Alpes et des Jura. *Bull. Soc. Vaudoise Sci. Nat.*, **37**, 547–549.

Kasprzyk,D. *et al.* (2004) EnsMart: a generic system for fast and flexible access to biological data. *Genome Res.*, **14**, 160–169.

Łabaj,P.P. *et al.* (2011) RNA-seq precision in quantitative expression profiling. *Bioinformatics*, **27**, i383–i381.

Mortazavi,A. *et al.* (2008) Mapping and quantifying mammalian transcriptomes by RNA-Seq. *Nat. Methods*, **5**, 621–628.

Oshlack,A. *et al.* (2010) From RNA-seq reads to differential expression results. *Genome Biol.*, **11**, 220.

Oshlack,A. and Wakefield,M.J. (2009) Transcript length bias in RNA-seq data confounds systems biology. *Biol. Direct*, **4**, 14.

R Development Core Team. (2009) *R: a Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. http://www.R-project.org (16 July 2013, date last accessed).

Ramsköld,D. *et al.* (2009) An abundance of ubiquitously expressed genes revealed by tissue transcriptome sequence data. *PLoS Comput. Biol.*, **5**, e1000598.

Risso,D. *et al.* (2011) GC-content normalization for RNA-seq data. *BMC Bioinformatics*, **12**, 480.

Robinson,M.D. *et al.* (2010) edgeR: a bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics*, **26**, 139–140.

Robinson,M.D. and Oshlack,A. (2010) A scaling normalization method for differential expression analysis of RNA-seq data. *Genome Biol.*, **11**, R25.

Sam,L.T. *et al.* (2011) A comparison of single molecule and amplification based sequencing of cancer transcriptomes. *PLoS One*, **6**, e17305.

Strub,T. *et al.* (2011) Essential role of microphthalmia transcription factor for DNA eplication, mitosis and genomic stability in melanoma. *Oncogene*, **30**, 2319–2332.

Sultan,M. *et al.* (2008) A global view of gene activity and alternative splicing by deep sequencing of the human transcriptome. *Science*, **15**, 956–960.

Wang,L. *et al.* (2010) DEGseq: an R package for idenetifying differentially expressed genes from RNA-seq data. *Bioinformatics*, **26**, 136–138.

Wickham,H. (2009) *ggplot2: Elegant Graphics for Data Analysis*. Springer, New York.