This is a preprint of an article published in *Nucleic Acids Research* Volume 43, Issue 7, article e47 (20 April 2015).

limma powers differential expression analyses for RNA-sequencing and microarray studies

Matthew E. Ritchie [1,3], Belinda Phipson [5], Di Wu [6], Yifang Hu [2], Charity W. Law [7], Wei Shi [2,4] and Gordon K. Smyth [2,3,8]

2 January 2015
Last revised 26 January 2015

(1) Molecular Medicine Division and (2) Bioinformatics Division, The Walter and Eliza Hall Institute of Medical Research, 1G Royal Parade, Parkville, Victoria 3052, Australia; (3) Department of Mathematics and Statistics and (4) Department of Computing and Information Systems, The University of Melbourne, Parkville, Victoria 3010, Australia; (5) Murdoch Childrens Research Institute, Royal Children's Hospital, 50 Flemington Road, Parkville, Victoria 3052, Australia; (6) Department of Statistics, Harvard University, 1 Oxford Street, Cambridge, Massachusetts 02138-2901, USA; (7) Institute of Molecular Life Sciences, University of Zurich, Winterthurerstrasse 190, Zurich 8057, Switzerland; (8) Email: smyth@wehi.edu.au.

Abstract

limma is an R/Bioconductor software package that provides an integrated solution for analyzing data from gene expression experiments. It contains rich features for handling complex experimental designs and for information borrowing to overcome the problem of small sample sizes. Over the past decade, *limma* has been a popular choice for gene discovery through differential expression analyses of microarray and high-throughput PCR data. The package contains particularly strong facilities for reading, normalizing and exploring such data. Recently, the capabilities of *limma* have been significantly expanded in two important directions. First, the package can now perform both differential expression and differential splicing analyses of RNA sequencing (RNA-seq) data. All the downstream analysis tools previously restricted to microarray data are now available for RNA-seq as well. These capabilities allow users to analyse both RNA-seq and microarray data with very similar pipelines. Second, the package is now able to go past the traditional gene-wise expression analyses in a variety of ways, analysing expression profiles in terms of co-regulated sets of genes or in terms of higher-order expression signatures. This provides enhanced possibilities for biological interpretation of gene expression differences. This article reviews the philosophy and design of the *limma* package, summarizing both new and historical features, with emphasis on recent enhancements and features than have not been previously described.

1 Introduction

Gene expression technologies are used frequently in molecular biology research to gain a snapshot of transcriptional activity in different tissues or populations of cells. These profiles are then compared to identify gene expression changes associated with a treatment condition or phenotype of interest. Gene expression studies may be randomized designed experiments in which a biological system is perturbed, for example by a gene knock-out or by applying a specified stressor. Such

experiments are among the most powerful tools in functional genomics, providing insights into normal cellular processes as well as disease pathogenesis. Or they may be observational studies in which different phenotypes are compared, diseased and normal tissue for example or cells from different populations. Such studies are common in cancer research and in the study of cell development. In either case, the study design can range from simple two group comparisons to complex set-ups with several experimental factors varying over multiple levels. Researchers might be interested for example in whether a particular gene facilitates or blocks the action of a particular drug, in which case knock-down and wild-type samples both with and without drug treatment would be profiled. Obervational studies may involve multiple batch effects and covariates that must be accounted for in the analysis.

Despite the complexity, gene expression studies often involve only a small number of biological replicates. The small but complex nature of gene expression studies poses challenging statistical problems and motivates the use of a number of specialized statistical techniques in order to get the most out of each dataset. We have developed the *limma* software over the past decade to provide a framework for analysing gene expression experiments from beginning to end in a flexible and statistically rigorous way.

The *limma* package is a core component of Bioconductor, an R-based open-source software development project in statistical genomics [16, 64]. It has proven a popular choice for the analysis of data from experiments involving microarrays [44, 8], high-throughput PCR [20], protein arrays [35] and other platforms. The package is designed in such a way that, after initial pre-processing and normalization, the same analysis pipeline is used for data from all technologies.

Recently, the capabilities of *limma* have expanded significantly in two important directions. First, the package can now perform both differential expression (DE) and differential splicing analyses of RNA sequencing (RNA-seq) data [32, 68]. All the downstream analysis tools previously restricted to microarray data are now available for RNA-seq as well. These capabilities allow users to analyse both RNA-seq and microarray data with very similar pipelines. Second, the package is now able to go past the traditional gene-wise expression analyses in a variety of ways, analysing expression profiles in terms of co-regulated sets of genes or in terms of higher-order expression signatures [32]. This provides enhanced possibilities for biological interpretation of gene expression differences.

This article reviews the philosophy and design of the *limma* package, summarizing both new and historical features, with emphasis on recent enhancements and features that have not been previously described. The article outlines *limma*'s functionality at each of the main steps in a gene expression analysis, from data import, pre-processing, quality assessment and normalization, through to linear modelling, differential expression and gene signature analyses.

2 Statistical principles

2.1 Overview

limma integrates a number of statistical principles in a way that is effective for large-scale expression studies. It operates on a matrix of expression values, where each row represents a gene or some other genomic feature relevant to the current study and each column corresponds to an RNA sample. On one hand, it fits a linear model to each row of data and takes advantage of the flexibility of such models in various ways, for example to handle complex experimental designs and to test very flexible hypotheses. On the other hand, it leverages the highly parallel nature of genomic data to borrow strength between the gene-wise models, allowing for different

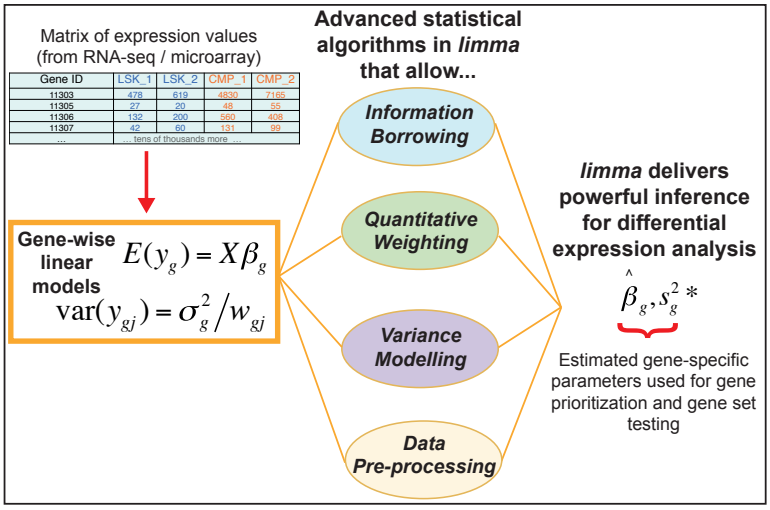

Figure 1: Schematic of the major components that are central to any *limma* analysis. For each gene g, we have a vector of gene expression values (y_g) and a design matrix, X that relates these values to some coefficients of interest (β_g). The *limma* package includes statistical methods that (i) facilitate information borrowing using empirical Bayes methods to obtain posterior variance estimators (s_g^{2*}), (ii) incorporate observation weights (w_{gj} where j refers to sample) to allow for variations in data quality, (iii) allow variance modelling to accommodate technical or biological heterogeneity that may be present and (iv) pre-processing methods such as variance stabilization to reduce noise. These methods all help improve inference at both the gene and gene set level in small experiments.

levels of variability between genes and between samples, and making statistical conclusions more reliable when the number of samples is small. All the features of the statistical models can be accessed not just for gene-wise expression analyses but also for higher level analyses of gene expression signatures. Figure 1 depicts the linear model and highlights the statistical principles employed in a typical *limma* analysis.

2.2 Linear models analyse complete experiments together

The hallmark of the *limma* approach is the use of linear models to analyse entire experiments as an integrated whole rather than making piece-meal comparisons between pairs of treatments. This has the effect of sharing information between samples. Analysing the data as a whole also allows us to model correlations that may exist between samples due to repeated measures or other causes. This kind of analysis would not be feasible were the data partitioned into subsets and analysed as a series of pairwise comparisons.

Linear models permit very general analyses. Researchers can adjust for the effects of multiple experimental factors or can adjust for batch effects. The linear model might include time course effects or regression splines. The linear model could even include the expression values themselves of one or more genes as covariates, allowing researchers to test for inter-gene dependencies. Linear models allow researchers to test very flexible hypotheses, not just simple comparisons between groups but also interaction effects or more complex customized comparisons.

2.3 Shared global parameters link gene-wise models

A separate model is fitted for each gene, but the gene-wise models can be linked by global parameters or global hyper-parameters. The use of global parameters is a simple means of

sharing information between genes that can be used even for the smallest experiments, because the global parameters can be estimated from the entire dataset involving all the genes at once. This strategy allows the gene-wise models to incorporate such things as correlations between duplicate probes for the same gene, or correlations between related RNA samples, or variations in quality between the RNA samples.

2.4 Empirical Bayes borrows information between genes

The highly parallel nature of gene expression experiments lends itself to a particular class of statistical methods, called parametric empirical Bayes, that borrow information between genes in a dynamic way [14, 40]. The fact that the same linear model is fitted to each gene allows us to borrow strength between genes in order to moderate the residual variances [63]. The estimated variance for each gene then becomes a compromise between the gene-wise estimator, obtained from the data for that gene alone, and the global variability across all genes, estimated by pooling the ensemble of all genes. This has the effect of increasing the effective degrees of freedom with which the gene-wise variances are estimated. It was an innovation of the *limma* package to show that exact small-sample inference could be conducted using the empirical Bayes posterior variance estimators [63]. This approach has proven particularly advantageous in experiments with small sample sizes, ensuring that inference is reliable and stable even when the number of replicates is small.

In recent years, the empirical Bayes procedures of *limma* have been enhanced in two important ways. First, the global variance estimate can now incorporate a mean-variance trend [59, 47, 24]. This is important because many gene expression technologies produce data that is less reliable at lower intensities or abundances. Second, the relative weighting of the gene-wise and global variance estimators no longer needs to be the same for all genes. This allows a sophisticated robust empirical Bayes procedure in which hyper-variable genes are identified and treated separately [46, 47]. Both of these enhancements improve statistical power and accuracy by improving the modelling of the global characteristics of the data in a more flexible way.

2.5 Quantitative weights allow for unequal quality

Another unique feature of *limma* is the ability to incorporate quantitative weights into all levels of the statistical analysis, from normalization, to linear modelling and gene set testing. Weights can be applied to genes or to RNA samples or to individual expression values. Weights can be used to give more emphasis to control probes during normalization, or can be used to down-weight measurements or samples that are less reliable in a gene expression analysis. The weights can be preset based on external quality information, or may be estimated from the expression data itself. The use of weights increases power to detect differentially expressed genes, and having a model based approach avoids the need for ad-hoc decisions on which observations or samples to filter out [54].

2.6 RNA-seq and sequence data

All the downstream analysis features of *limma* are available for RNA-seq and other sequence count data, as well as for data from microarrays and other platforms. Traditionally, RNA-seq data requires specialized software based on the negative binomial or similar distributions [57]. *limma* however is able to analyse RNA-seq read counts with high precision by converting counts to the log-scale and estimating the mean-variance relationship empirically (Figure 3A) and incorporating it into precision weights. The precision weights are computed using the `voom`

function and incorporated into the analysis of log-transformed RNA-seq counts using the same downstream commands as microarrays. The resulting pipeline gives comparable performance to the best of the negative binomial based software packages but with greater speed and reliability for large datasets [52, 24]. Additionally, and conveniently, only minimal pipeline changes are required when switching between analyses for RNA-seq and microarray experiments within *limma*. This also means that the same statistical tests with the same format of results and graphical displays are available for both data types.

2.7 Variance models allow for unequal variability

Expression values often show some degree of heteroscedasticity, either because there is a relationship between abundance and measurement precision, or because some treatment conditions are more heterogeneous than others. For example, tumours might be more variable than normal tissue. Concern for such effects has prompted some researchers to filter out low intensity observations or to use Welch's t-test for DE between two groups instead of classical pooled t-tests. The use of weights and the ability to model global parameters allow *limma* to incorporate unequal variances in a number of ways. One way is through estimating a mean-variance trend, which can either be incorporated into the empirical Bayes procedure as mentioned above or used to generate observation weights [24]. A recent development is the ability to estimate precision weights associated with treatment groups or more generally with any given set of covariates. More generally again, the mean-variance trend can be estimated in a treatment-specific way, combining the two types of heteroscedasticity mentioned above. These approaches allow *limma* to model unequal variances even for experiments with a small number of RNA samples. Importantly, they accommodate unequal variances without compromising the linear modelling and empirical Bayes framework of the package.

2.8 Using sets of genes to represent higher-level expression signatures

In recent years, the linear modelling capabilities of *limma* have been extended to higher-level expression signature analyses involving co-regulated sets of genes. The idea is to use a set of genes, together with their log-fold-changes, to represent the transcriptional signature of a biological process or cell type. One way that this is done is by rotation technology, which permits statistical significance to be tested for sets of genes for any linear model contrast. A particular feature of rotation tests is the ability to incorporate prior information about the direction and strength with which each gene is expected to contribute to the statistical signature. In this way, *limma* provides a uniquely flexible means to relate new expression datasets to previous results collated from earlier experiments, taking into account for example the fold-change and direction of change for each gene in the earlier experiment.

A closely related statistical approach implemented in *limma* is to fit global covariance models, either to estimate correlations between genes or to estimate the relatedness between the DE profiles resulting from difference comparisons. These new analyses are described briefly later in this article.

2.9 Pre-processing methods preserve information

Microarray expression data is measured as intensities, which need to be background corrected and normalized before any statistical analysis can be conducted. *limma* includes a range of background correction and normalization procedures suitable for different types of DNA microarrays or protein arrays. Notable are the maximum likelihood implementation of the normal-

exponential convolution model for background correction [62] and the implementation of lowess curves and normalization using quantitative weights. The guiding principle in the pre-processing steps is to preserve information, avoiding missing values or inflated variances [55]. Normalized intensities are offset from zero before transforming to the log-scale to avoid missing values or large variances. Offsets in a range of moderate values have been shown to achieve an effective compromise between noise and bias [61].

2.10 Mean-difference plots

Measuring expression in multiple RNA samples produces columns of correlated expression values, which are highly correlated because they are measured on the same set of genes or genomic features. It has long been established in the biomedical literature that the level of agreement between correlated variables can be usefully examined by plotting differences vs means. Such a plot is called a Bland-Altman plot [36] or a Tukey mean-difference plot [10]. Indeed the concept of DE can be viewed as a measure of disagreement between expression measures for the same genes in different samples. Mean-difference plots were introduced to the two-color microarray literature by Dudoit *et al.* [13] and to the single-channel literature by Bolstad *et al.* [6], who called them MA-plots. *limma* generalized the concept of an MA-plot in two ways. First, the idea was extended to apply to sets of single-channel expression values. In this case, the plot is used to compare each sample to the average of all other samples. A virtual array is constructed by averaging the log-expression value for all the samples other than the sample of interest, and then a mean-difference plot is made between the single array and the virtual array. Second, the idea was extended to apply to the fitted model objects. In this case, the plot compares the log-fold-changes for a chosen contrast versus the average log-expression values of each gene across all samples. In effect, this plots a coefficient of the linear model versus an overall mean intercept parameter. These ideas were part of the original *limma* package submitted to Bioconductor in 2003.

2.11 Parametric modelling versus permutation methods

It is worth mentioning what *limma* doesn't do, which is permutation or re-sampling-based inference. Permutation is frequently useful in large-scale studies when the aim is to compare two groups. However permutation has a number of disadvantages which make it unattractive for assessing differential expression in experiments with complex designs. If permutation is applied only to samples involved in two treatment conditions to be compared, then the typically small number of replicates is a severe limitation that will result in low power to detect differences. If permutation is applied to all the samples in a multi-factor experiment, then the composite null hypothesis being testing is an uninteresting one, and the power to reject it may be highly dependent on the existence of differential expression between treatment conditions which are not of primary interest. In other words, permutation cannot be tuned to test specific null hypotheses of interest in a designed experiment. Even more importantly, permutation assumes that all samples are independent and identically distributed under the null hypothesis, and these assumptions are frequently, usually perhaps, unrealistic. In addition, permutation is potentially misleading when the samples are correlated or of unequal precision. In other words, permutation is unable to accommodate blocking structures or quality weights. In small, complex experiments, the potential compromises involved in modelling expression values using parametric distributions, which can never be perfectly correct, are outweighed by the gains in precision and accuracy by modelling the variance structure more realistically.

3 Pre-processing RNA-seq and other sequencing data

Figure 2 provides an overview of the functions available at each stage of a gene expression analysis. The first step is to import expression data into the R session.

limma accepts RNA-seq data in the form of a matrix of read counts, with rows for genomic features and columns for RNA samples. Alternatively it can accept a `DGEList` object from the *edgeR* package. The genomic regions are often genes or exons, but could in principle be any genomic feature of interest. In this article, the regions will usually be called genes for simplicity of terminology. The read counts are processed by the `voom` function in *limma* to convert them into $\log_2$ counts per million (logCPM) with associated precision weights. The logCPM values can be normalized between samples by the `voom` function or can be pre-normalized by adding normalization factors within *edgeR*.

Raw read counts are assembled outside *limma* using tools such as featureCounts [29], HTSeq-counts [1] or RSEM [27]. The authors of this article find the Subread [28] and featureCounts pipeline particularly convenient because it is fast, acccurate [68] and can be run from the R prompt using the *Rsubread* package. The data input to *limma* should be counts, rather than popular expression summaries such as reads-per-kilobase-per-million (RPKM), so that *limma* can estimate the appropriate mean-variance relationship. The `voom` output can be converted to RPKM values for convenience of interpretation, by subtracting log-gene-lengths, but this should be done after running `voom` rather than before.

After running `voom`, downstream analysis for RNA-seq data is the same as for any other technology. For example, RNA-seq data can be explored using boxplots or mean-difference plots, similarly to single-channel microarray data. More detail is given about this in the following sections.

4 Preprocessing microarray data

4.1 Reading or importing data

For DNA or protein microarrays, importing expression data often involves reading output files created by an image analysis program. Alternatively, a data frame of expression values may be read from a file or data might be directly imported as an R object.

The main *limma* function to read image output files is `read.maimages`. This function directly supports formats written by many different image analysis programs including GenePix, Agilent Feature Extraction, ArrayVision, BlueFuse, ImaGene, QuantArray and SPOT (Table 1). It also supports the Stanford Microarray Database format. Output in other formats can be read if the appropriate column names are supplied. Two-color and single-channel data are both supported. Illumina files need special treatment: output from Illumina's GenomeStudio can be read by `read.ilmn` if exported as a text file or by `read.idat` if in binary format.

Probe annotation is also automatically read if contained in the image output files, or can be read separately. `readGAL` supports the GenePix Gene Array List format. `read.maimages` includes the ability to generate spot quality weights according to any user-specified rule based on any information found in the image output files.

limma includes many possibilities for using or highlighting different types of control probes. The functions `readSpotTypes` and `controlStatus` are provided to conveniently classify probes based on text found in the input files. The status of each probe is automatically carried through to appropriate downstream functions.

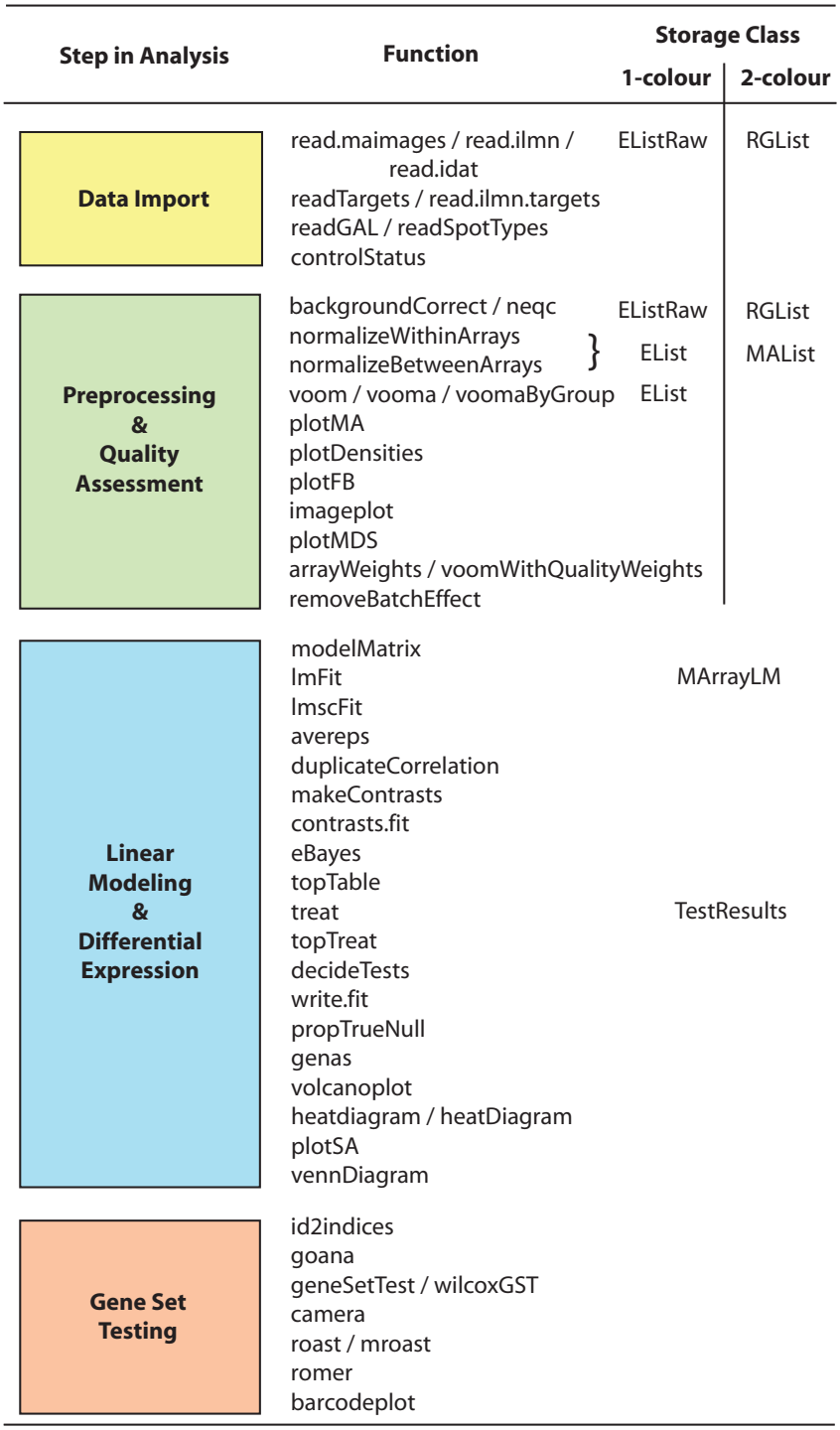

Figure 2: The *limma* workflow. The diagram shows the main steps in a gene expression analysis, along with individual functions that might be used and the corresponding classes used to store data or results. Online documentation pages are available both for each individual function and for each major step.

Table 1: Standard microarray data formats handled by *limma*. Data output by the following software can be read-in using `read.maimages` or `read.ilmn`. *limma* can read files in other formats, provided the user provides the names of the columns containing foreground and background intensities.

Software	**Vendor**	**Channels**
Agilent Feature Extraction Software	Agilent Technologies	1/2
ArrayVision	GE Healthcare	1/2
BlueFuse	BlueGnome	1/2
GenePix	Molecular Devices	1/2
BeadScan/GenomeStudio	Illumina Inc	1
ImaGene	BioDiscovery	1/2
QuantArray	PerkinElmer Life Sciences	1/2
ScanArray Express	PerkinElmer Life Sciences	1/2
SMD	Stanford	1/2
Spot	CSIRO	1/2

The function `readTargets` is provided to read information about the RNA samples or *targets*. This information typically includes information about the treatment conditions and experimental design.

limma can accept data objects containing expression data from other Bioconductor packages. It can accept `marrayNorm` objects from the *marray* package, `PLMset` objects from the *affyPLM* package, `vsn` objects from the *vsn* package, or objects of any class inheriting from `ExpressionSet`. Alternatively, expression data can be supplied as a numeric matrix. Expression values can be image intensities or normalized log-expression values.

4.2 Background correction

When array images are read, it is usual to read both foreground and background intensities for each probe. The background intensities can be used to derive an estimate of the ambient intensity affecting each probe. Removing this non-specific signal from the foreground intensity of each probe is called background correction and it is typically the first step in processing microarray images. Simply subtracting background from foreground intensities is too heavy-handed [55]. The *limma* `backgroundCorrect` function offers a range of more sophisticated alternatives, most unique to the package. These include a method based on a convolution of normal distributions [21] and a normal-exponential (*normexp*) convolution [55] with different options for parameter estimation [62]. The `plotFB` function plots foreground against background intensities for each array and is useful for choosing an appropriate correction method.

Illumina BeadChip data again benefits from special treatment. The `nec` function implements normexp background correction for Illumina BeadArrays making special use of the control probes that are specific to these arrays [61].

The `propexpr` function compares intensities to those of negative control probes to estimate the total proportion of probes on each array that correspond to expressed genes [60]. This provides an estimate of the size of the transcriptome in each sample and is useful for deciding how many probes to filter from downstream analyses.

4.3 Normalization

Before meaningful comparisons can be made between treatment conditions in a designed experiment, it is critical that the expression values are normalized so that all the samples are as far as possible on the same measurement scale. The purpose of normalization is to remove systematic effects due to technical differences between the assays unassociated with the biological differences of interest. Different technology platforms introduce different biases and so require different normalization methods. The `normalizeWithinArrays` function normalizes data from two-color microarrays by aligning the two channels for each array. A popular method is to remove intensity-dependent dye-biases and spatial artifacts from M-values (log-intensity ratios) using locally weighted regression (loess) [78]. The `normalizeBetweenArrays` function aligns expression values between samples for one-color microarrays and other single channel platforms using methods such as quantile normalization or cyclic loess [6]. Both functions provide a range of different normalization methods suitable for different platforms. `normalizeBetweenArrays` also implements separate channel normalization methods for two-color arrays [79, 67]. *limma* is the only software to allow the use of quantitative weights in loess normalization [66], giving it the ability to downweight less reliable probes or to give higher priority to control probes or housekeeping genes. The latter ability has been exploited for normalizing assays when the proportion of differentially expressed genes may be high, for example boutique arrays [43], miRNA arrays [72], PCR arrays, protein arrays or protein mass spectrometry. Other enhancements include the ability to replace the loess curve with a spline curve that has high robustness breakdown properties, and the ability to apply empirical Bayes moderation to the spline curves for multiple regions within the same array (robust-spline normalization).

The `neqc` function implements quantile normalization for Illumina BeadArrays making special use of the control probes specific to these arrays [61].

All the between-arrays normalization methods are accessible for RNA-seq data from within the `voom` function. Alternatively, `voom` has the ability to respect normalization factors computed outside of *limma* by methods such as trimmed mean of M-values [58] or conditional quantile normalization [18].

4.4 Graphical exploration of data quality

Diagnostic plots allow the user to visually inspect data from a designed experiment in order to identify potential quality problems, such as degraded samples, or problems that arise due to array handling or sample processing. Such displays may also reveal systematic biases that should be removed prior to downstream analysis. Figure 3 presents examples from three different plotting functions. Plots for individual arrays include the foreground-background plots mentioned above (`plotFG`), and image plots which can reveal inconsistencies across the array surface (`imageplot`).

Mean-difference plots which show intensity-dependent trends in the log-ratios of two-color arrays (`plotMA`, Figure 3B). The `plotMA` function can show similar plots for single channel data. In this case, the mean-difference plot is constructed by comparing the log-expression values for that sample compared with the mean of all other samples. The `plotMA` function makes it simple to highlight particular subsets of probes or genes, for example control probes. Control probes are automatically highlighted if they have previously been identified using `controlStatus` (Figure 3B).

The distribution of expression values can be compared between samples using box plots or density plots (`plotDensities`). The latter is particularly useful when considering separate channel analyses of two-color arrays.

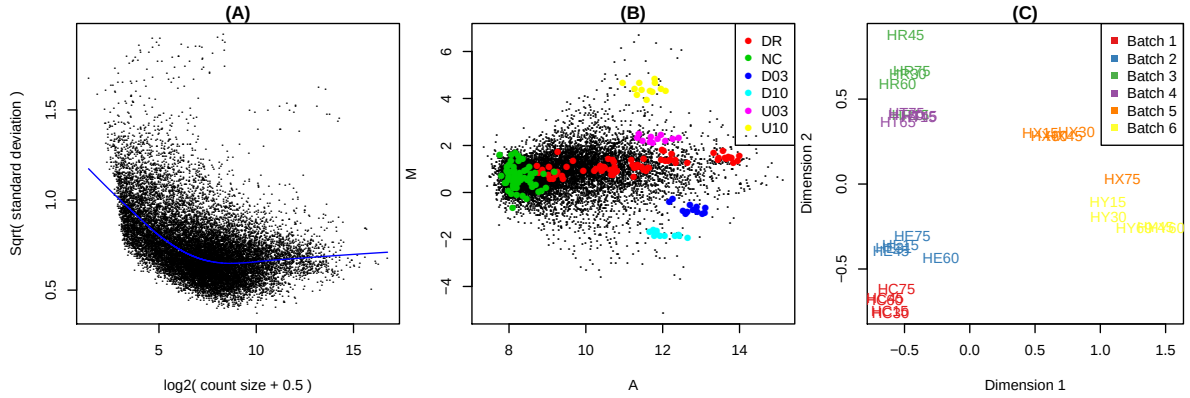

Figure 3: Example diagnostic plots produced by *limma*. **(A)** Plot of variability versus count size for RNA-seq data, generated by `voom` with `plot=TRUE`. This plot shows that technical variability decreases with count size. Total variability asymptotes to biological variability as count sizes increases. **(B)** Mean-difference plot produced by the `plotMA` function for a two-color microarray. The plot highlights negative (NC), constant (DR) and differentially expressed (D03, D10, U03, U10) spike-in controls. Regular probes are non-highlighted. **(C)** Multidimensional scaling (MDS) plot of a set of 30 microarrays, generated by `plotMDS`. All arrays are biologically identical and the plot reveals strong batch effects. Distances represent leading log2-fold changes between samples.

5 Finding differentially expressed genes

5.1 Exploration of sample relationships

After the pre-processing steps described above, the next major analysis stage is to identify differentially expressed genes. It is advisable to begin the DE analysis with a plot that visualizes the relative differences in transcriptional profile between the samples. The `plotMDS` function uses multi-dimensional scaling to plot differences in expression profiles between different samples (Figure 3C). Distances between samples on the plot represent *leading fold change*, which is defined as the root-mean-square average of the log-fold-changes for the genes best distinguishing each pair of samples. This provides a type of unsupervised clustering of the samples. It is useful for examining how different are the profiles produced by different experimental factors and for identifying unexpected patterns, such as batch effects, that should be adjusted for during the linear model analysis. This helps guide the construction of the design matrix used for the linear modelling below.

The `plotRLDF` function provides a supervised plot of the samples that shows whether the expression data can distinguish a set of known groups. The function implements a regularized version of linear discriminant functions.

The `removeBatchEffect` function can be used to remove systematic variation due to batches or other covariates prior to plotting the data, so that the effect of treatments can be better seen.

5.2 Linear modelling

The core component of the *limma* package is the ability to fit gene-wise linear models to gene expression data in order to assess DE [63]. The basic idea is to estimate log-ratios (for two-channel data) or log-intensities (for single-channel data) between two or more target RNA samples simultaneously.

Each analysis begins with a matrix of expression levels, with probes/genes/exons in the rows

and different samples (biological/technical replicates) in the columns. The linear modelling is performed in a row-wise fashion, with regression coefficients and standard errors either directly estimating the comparisons of interest or via contrasts to obtain test-statistics for gene ranking that can be further summarized at the gene set level to perform gene signature/pathway-level ranking.

The flexibility of the linear modelling approach allows almost any experimental design to be handled. Experiments with two or more groups, factorial and time-course designs, and internal controls such as dye-swaps can all be modelled and summarized using the `lmFit` function. Where appropriate, nuisance variables such as batch and dye effects can also be modelled. Models can be fit robustly or by least squares. Once a linear model is fitted, the `makeContrasts` function can be used to form a contrast matrix. The fitted model object and contrast matrix are used by `contrasts.fit` to compute $\log_2$-fold-changes and t-statistics for the contrasts of interest. This allows all possible pairwise comparisons between treatments to be made.

The `plotSA` function provides a useful diagnostic plot of the linear model fit, plotting gene-wise residual standard deviations against average log-expression. This allows mean-variance trends to be readily identified, should they exist.

5.3 Quality weights and heteroscedasticity

limma is the only package that allows variations in quality to be handled in a graduated way via quantitative weights. Both observation-level [56, 64, 24] and sample-specific weights [54] can be used in an analysis. For microarray data, the `arrayWeights` function estimates relative array variances, which are converted to weights which can be used in the linear model analysis to down-weight observations from less reliable arrays. Probe and array weights can be easily combined by multiplying them together, and when used appropriately, have been demonstrated to increase power to detect DE [56, 54].

For RNA-seq data, the `voomWithQualityWeights` function combines observation-level and sample-specific weights for use in the subsequent linear modelling.

5.4 Blocking and random effects

limma includes a unique strategy for incorporating the fact that observations or samples may be correlated. The strategy is similar to fitting a random effects model, with the difference that all genes are constrained to share the same intrablock correlation. The `duplicateCorrelation` function is used to estimate the consensus correlation. The correlation structure is then incorporated into the linear model fit and hence into all tests for DE. Originally the idea was used to estimate the correlation between replicate copies of the same probe on a microarray [65]. The correlation strategy preserves more information than simply averaging the replicate probe copies. More generally, the same idea is also used to model the correlation between related RNA samples, for example repeated measures on the same individual or RNA samples collected at the same time.

5.5 Separate channel analysis of two-color microarrays

Two-color microarrays are traditionally analysed in terms of log-ratios between the two channels hybridized to each probe. *limma* also provides the possibility of analysing two-color microarrays as if they were single channel microarrays with two separate samples hybridized to each physical array. This provides a very powerful type of analysis in which intensities can be directly compared between microarrays. The pairing of the red and green channels from each array is kept

track of by estimating the correlation between the two channels hybridized to each probe [67]. This type of separate channel analysis uses the `intraspotCorrelation` and `lmscFit` functions.

All the linear model fits, whether using `lmFit` or `lmscFit`, produce a fitted model object with the same structure. The same fitted model applies regardless of whether correlations have been estimated, whether robust regression or least squares has been used, or whether quality weights have been included. This consistency allows the same framework for DE to be used for all experimental designs and platform technologies.

5.6 Testing for differential expression

An empirical Bayes framework to borrow information between genes when estimating the variances is implemented in the `eBayes` function. Gene-wise variances are squeezed towards the common or trended variance, which reduces the number of false positives for genes with very small variances and improves power to detect DE for genes with larger variances. *limma* includes a robustified shrinkage strategy that allows for gene-wise shrinkage factors to be estimated. This ensures unusually large variances are not squeezed too heavily, reducing the chance that they will appear statistically significant; while more consistently expressed genes are squeezed more severely towards the common variance. This robust strategy offers the benefits of shrinkage to the majority of the genes, while negating the effects of outliers.

For each coefficient in the linear model or contrast, empirical Bayes moderated t-statistics and their associated p-values (or log-odds of DE) are generally used to assess the significance of the observed expression changes. Moderated F-statistics which combine the t-statistics for all contrasts into an overall test of significance for each gene can also be used.

When one has a particular cut-off for log-fold-change in mind, the `treat` function can be used to test whether the $\log_2$-fold-change is greater than a threshold rather than merely different to zero [37]. This can be effective for prioritizing results that are biologically as well as statistically significant.

limma provides a number of options to adjust tests for multiple testing. Users can control either the family-wise type I error rate or the false discovery rate [5]. As well as the usual control for multiple testing across multiple genes, *limma* is the only software package to provide methods for error rate control across multiple contrasts and genes simultaneously. For individual tests, multiple testing can be applied using the `topTable` function. The `decideTests` function gives access to the full range of options.

To visualize the results of a DE analysis for single or multiple contrasts, *limma* provides a number of plotting options. Figure 4 shows three such displays: a volcano plot showing the DE results from a single condition, a Venn diagram showing the number of differentially expressed genes in multiple experimental conditions and a barcode enrichment plot highlighting a particular gene signature in a DE analysis ranked by moderated t-statistics.

Another useful plot is produced by `plotMA`, which plots estimated log-fold-changes against mean log-expression for each gene. This allows the magnitude of changes to be visualized in the context of overall expression level, see for example Figure 5C of Liu *et al.* [32].

5.7 Testing for differential splicing

The linear model framework of *limma* is extended to test very easily for differential splicing events when exon-level expression data is available. The data can be either from an exon microarray or from RNA-seq data summarized at the exon level. In either case, the approach is based on fitting linear models to the exon-level expression data. The approach can relate differential exon

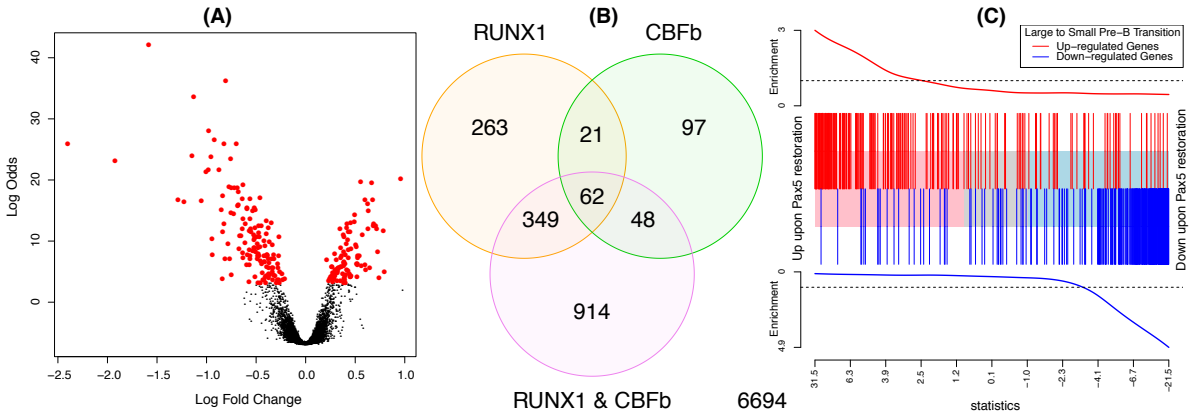

Figure 4: Example plots displaying results from DE or gene set analyses. **(A)** Volcano plot showing fold changes and posterior odds of DE for a particular comparison (RUNX1 over-expression vs wild-type in this case), generated by `volcanoplot`. Probes with $p < 0.00001$ are highlighted in red. **(B)** Venn diagram showing overlap in the number of DE genes for three comparisons from the same study as (A), generated by the `vennDiagram` function. **(C)** Gene set enrichment plot produced by `barcodeplot`. The central bar orders differentially expressed genes by significance from up to down upon Pax5 restoration in an RNA-seq experiment [32]. The vertical bars mark genes that are induced (red) or repressed (blue) upon the transition from large cycling pre-B cells to small resting pre-B cells during normal B cell development according to the published literature [19] The plot shows a strong positive concordance between Pax5 restoration and the large to small cell transition. The `roast` function can be used to assign statistical significance to this correlation.

usage to continuous as well as categorical predictors or to any contrast in a linear model. The test is conducted by the `diffSplice` function and results are displayed using `plotSplice` and `topSplice`. The `plotExons` function is also useful for exploring exon expression for individual genes. This approach is considerably faster than alternative approaches to differential splicing, making large-scale surveys of differential exon usage feasible.

6 Higher-level analyses

Linear modelling of gene expression data provides the ideal platform from which to attack larger functional genomic questions related to gene-wise independence or interaction and the decomposition of gene signatures into distinct molecular pathways. This section describes higher-level analyses involving multiple genes.

6.0.1 Estimating the proportion of true null hypotheses:

First we consider DE from a genomic point of view. The DE analyses described so far identify individual differentially expressed genes according to a FDR criterion. However, for most studies, there are likely to be false negatives: truly differentially expressed genes that are not detected as differentially expressed because the study did not have enough statistical power to identify them with confidence.

The `propTrueNull` function estimates the number of truly differentially expressed genes that remain to be identified. Mathematically, it estimates the proportion of true null hypotheses in a collection of hypothesis tests given a vector of p-values. In a gene expression study, it estimates the proportion of non-differentially expressed genes, out of all tested, for any contrast in the

linear model. The function implements a number of different methods. The default is based on averaging local false discovery rates across the p-values [46]. Other methods are the histogram method of [41, 42], the convex decreasing density estimate of [22], and a very simple estimate based on averaging the p-values.

6.1 Genuine association of gene expression profiles

Gene expression experiments typically involve a number of different treatment conditions. A question that often arises is this: to what extent do two different treatments produce similar or different expression profiles? One way to address this question is to count the overlap in differentially expressed genes from the two treatments. This approach however is too crude. It is very sensitive to the significance cut-off used to identify differentially expressed genes, has little chance of success in situations where power to detect differentially expressed genes is relatively low, and is subject to technical biases when both treatments are compared back to the same control samples. The `genas` function [46] addresses these problems. It tests whether two different contrasts in a linear model affect the same genes in similar or different ways, adjusting for biases, without needing to apply a significance cut-off for assessing DE. More technically, it estimates the true biological correlation between the $\log_2$-fold-changes of two different contrasts. By biological correlation, we mean the correlation that would exist between log-fold-changes if they could be measured perfectly without any statistical error. `genas` is based on a bivariate generalization of the empirical Bayes model that is used to assess DE in *limma*. This method is particularly powerful for gaining insight into commonly affected gene pathways when the changes are small but consistent. For instance, applying `genas` to a microarray study looking at the relationship between polycomb repressor complex (PRC) 1 and PRC2 facilitated the discovery of the opposing roles of these two complexes [34]. This relationship would have been missed if the analysis were restricted to the statistically significant genes from each contrast alone.

6.2 Gene set testing

Gene set analyses assess the overall significance of a set of co-regulated genes. Each gene set is chosen to represent a particular molecular pathway or some other biological process of interest. Gene sets are defined by gene annotation external to the current expression study, for example from Gene Ontology (GO) database [3] or from previous expression studies. For gene sets defined by previous studies, the genes may optionally be annotated with the direction and magnitude of expression changes in the earlier experiment. In this way, a gene set may contain genes both positively and negatively associated with the molecular pathway that it represents.

A range of options to look for expression changes in an *a priori* defined pathway or gene set are available in the *limma* package (i.e., `wilcoxGST`, `goana`, `camera`, `roast` and `romer`). There are two major types of gene set tests: competitive tests and self-contained tests. In competitive tests, the different gene sets are pitted against one another. In self-contained tests, only the genes in the gene set are considered and whether they are associated with the sample groups [17].

If the primary interest is to use Gene Ontology (GO [3]) terms as gene sets, the `goana` function is available. It applies a generalized hypergeometric test for enrichment of each GO term in the up and down differentially expressed genes from a linear model fit. It can also test for over-representation of GO terms in one or more sets of genes. `goana` includes the ability to adjust for gene length or abundance biases in DE detection in a similar way to the *goseq* package

[80].

Another simple approach is implemented in the `geneSetTest`/`wilcoxGST` functions which perform rank-based tests (as used in Michaud *et al.* [38]). Such tests assess how highly ranked a group of genes is in a particular contrast relative to other genes using a test statistic for DE. They assume the expression level of each gene is independent of the other genes. Ignoring positive correlation between genes in a gene set tends to cause optimistic p-values in competitive gene set tests.

More sophisticated competitive tests that take into account dependence of the genes in the linear modelling framework have also been implemented via `camera` [75, 74]. `Camera` is a variance adjusted mean-rank testing method. The variance of the summary statistics would be higher than estimated, if the (mostly) positive average inter-gene correlation in a gene set is considered. In `camera`, a variance inflation factor is computed and used to adjust the variance of the summary statistics. This avoids overly optimistic p-values in the test results.

The `roast` (rotation gene set tests) method tests the self-contained null hypotheses of whether a given gene set is differentially expressed [73]. Instead of permutation, it uses rotation, which is a smoothed version of permutation suitable for linear models [23]. For genes in the set, the residual space orthogonal to the nuisance parameters in the linear model is randomly rotated to generate the distribution of the parameter of interest under the null hypothesis [23]. Unlike permutation of samples, rotation does not change the correlation between genes which ensures that `roast` holds its size. By using the full residual degrees of freedom in linear models, `roast` can provide higher power in small complex gene expression experiments. It accommodates any correlations, blocking structure or weights used in the original linear model. As a self-contained test, `roast` tends to have greater power than an equivalent competitive test. The `mroast` function applies this procedure to multiple gene sets in series. Although `roast` can be used to test a large number of gene sets, `roast` is most useful for linking different datasets by finding similarities in gene expression patterns using gene weights from other differential expression analyses [31, 30, 4, 74]. Potential applications for `roast` include those where the set might not be made up of genes, e.g., exon-level expression analyses to test whether any exon of a given gene is differentially expressed.

Gene set enrichment analysis (GSEA) is an approach which correlates a large database of co-regulated gene sets with respect to a microarray or RNA-seq data set [39, 69]. `romer` implements a GSEA of a battery of gene sets similar in motivation to Subramanian *et al.* [69] but designed for use with linear models. It is a competitive test, in that the different gene sets are pitted against one another. Set-level test statistics are calculated as the mean of the moderated t-statistics (or mean from the top 50% ranked genes). Like `roast`, it uses rotation [23] to calculate p-values, and can be used with any linear model with some level of replication. It takes into account the large number of moderate gene expression changes we observe, and can perform competitive gene set tests using any collection of gene sets, such as the Broad Institute's Molecular Signatures database (MSigDB) [69]. Therefore, modest changes in individual genes are less relevant, and instead changes across gene sets are detected.

Both `camera` and `romer` are competitive tests and ideal for testing large numbers of gene sets. However, when there are many differentially expressed genes that are not in the gene set or many differentially expressed genes overall, the power of `camera` and `romer` will decrease. It is known that the DE status of genes outside the gene set will not impact on the significance of a self-contained test, so the power of `romer` can be higher than `camera` in this situation [34, 2].

To perform `romer`, `mroast` or `camera` gene set tests with a database of gene sets, we need to know the indices of the gene sets in the expression data matrix. A simple way to match between gene identifiers in the gene set to the expression matrix is to use the function `id2indices`. To map

a set of gene symbol aliases to the set of official gene symbols, the functions `alias2SymbolTable` or `alias2Symbol` can be used. The *limma* authors maintain mouse and human versions of the MSigDB collections [69] in R format that can be conveniently used with these functions (`http://bioinf.wehi.edu.au/software/MSigDB`).

`Camera`, `roast` and `romer` all operate on the pre-processed gene expression data matrix. If only the statistics or p-values for individual genes are available, for example from a genome-wide association study, then the `geneSetTest`/`wilcoxGST` functions can be used. These methods can also accommodate observational and/or sample-specific weights in the gene set test.

The ranks of genes from a particular gene signature in a given data set can be visualized using the `barcodeplot` function (Figure 4C), which highlights the genes from a set using vertical bars, with a smoother line showing the (tri-cube) moving average of the ranks from the genes in the set. The `barcodeplot` function can optionally display varying weights for different genes, for example log-fold-changes from a previous experiment.

7 User-interface

7.1 Object-oriented programming

A simple but appropriate object-oriented paradigm provides users with a consistent analysis interface that is very easy from a user point of view. Many *limma* functions are generic or operate appropriately on objects of different classes. *limma* defines a number of classes which have been tailored to handle both microarray and RNA-seq data. The philosophy has been to define simple list-based data objects that can be easily explored and manipulated by users, in the same style as familiar, long-standing core functions in R such as `lm` and `glm`.

For raw intensity data, the classes `'RGList'` and `'EListRaw'` are used to store two-color and single-channel data respectively. These objects are often created using the function `read.maimages` and contain the raw values from the image analysis output files along with probe annotation information.

Normalized data is stored in `'MAList'` or `'EList'` objects. Normalized two-color data is converted from red and green intensities, R and G, into M and A-values, which hold the log-ratio and average log-intensity values for each spot. Single channel data is background corrected and $\log_2$ transformed and stored in an `'EList'` object. For RNA-seq data, the `voom` transformed matrix of gene/exon counts is also stored in an `'EList'` object.

The next major classes store output from a DE analysis. `'MArrayLM'` objects store the result of fitting gene-wise linear models to the normalized intensities or log-ratios. Objects of this class are created by the `lmFit` and `eBayes` functions. After running `decideTests`, an object of class `'TestResults'` stores the results of testing a set of contrasts equal to zero for each probe/gene.

All of these data classes obey many analogies with matrices. In the case of `'RGList'`, `'MAList'`, `'EListRaw'` and `'EList'`, rows correspond to probes/genes and columns to different samples. In the case of `'MarrayLM'` and `'TestResults'` rows correspond to unique probes/genes and the columns to linear model coefficients or contrasts. The standard R functions `summary`, `dim`, `length`, `ncol`, `nrow`, `dimnames`, `rownames`, `colnames` have methods for each of these classes. Objects of any of these classes may also be subsetted. Multiple data objects may be combined by rows to add extra probes, or by columns to add extra arrays.

Furthermore all of these classes may be coerced to be of class matrix using `as.matrix`, although this entails loss of information. Fitted model objects of class `'MArrayLM'` can be coerced to class `data.frame` using `as.data.frame` in R. The first five classes belong to the virtual class

'LargeDataObject' for which a show method is defined to display the leading rows of a large vector, matrix or data.frame.

7.2 Computational efficiency

The *limma* package is implemented primarily in R [50] and includes some C code to speed up computationally intensive steps. At every stage, effort has been expended to achieve high numerical reliability and efficiency. The memory requirements are linear in the number of genes and the number of samples. Most estimation procedures finish in a few seconds on a standard desktop computer and virtually all in less than a minute.

7.3 Availability

The *limma* software is freely available online as part of the Bioconductor project (http://www.bioconductor.org). More than 120 other Bioconductor packages make use of *limma* (as of March 2014). The *limma* package is used as a building block or as the underlying computational engine by a number of software projects designed to provide user-interfaces for gene expression data analysis including *limmaGUI* [71], *affylmGUI* [70], WebArray [76], RACE [49], CarmaWEB [51], Goulphar [26], MAGMA [53], Asterias [12], GenePattern [11], GEO2R (http://www.ncbi.nlm.nih.gov/geo/geo2r), the EBI expression atlas [45], Guide [9] and Degust (http://www.vicbioinformatics.com/degust).

7.4 Documentation

The *limma* package provides three levels of documentation. First, each function has its own documentation page that concisely but completely specifies the input data, options and output format of the function. Similarly, each data class has a documentation page explaining all the required and optional components of objects of that class. Care is taken to adhere to the same standards and style that users will be familiar with from help pages in the base R packages.

Second, a series of more general subject help pages serve to link together functions and classes that are used for related purposes. The subject pages cover the topics of (1) introduction, (2) classes, (3) reading data, (4) background correction, (5) normalization, (6) linear models, (7) individual channel analysis of two-colour data, (8) hypothesis testing for linear models, (9) diagnostics and quality assessment, (10) gene set tests and (11) RNA-seq.

Third, the package comes with an extensive user's guide of over 120 pages, available from the drop-down menu in Windows or alternatively launched by the limmaUsersGuide command. The user's guide gives detailed advice on how to analyse a variety of common study designs. It also includes ten fully-worked case studies for which full data and code is provided.

Users who need more help or advice are invited to post questions to the Bioconductor support site (https://support.bioconductor.org). Questions are usually answered promptly, either by the authors or by other members of the Bioconductor community. The support site archives answers to many common questions, including many queries about experimental design and setting up appropriate design matrices.

8 Conclusions

This article has summarized the current features of the widely used, open source *limma* package for gene expression analysis. This software provides an integrated data analysis solution, using

advanced computational algorithms to deliver reliable performance on large data sets and object-oriented ideas to represent expression data and simplify the user interface. New functionality is continually being added as model refinements and new use cases arise.

Although originally developed with microarray data in mind, the development of the voom methodology unlocks the majority of analysis methods for use on RNA-seq data, such as random effects modelling and gene set testing. As with any data analysis problem, the appropriate combination of methods to use will depend upon the biological question, platform used (microarray/RNA-seq) and experimental design.

Being R-based, reports of *limma* analyses can be compiled using Sweave [25] or *knitr* [77] and provided along with the raw data in a compendium to promote reproducible research in genomics [15].

Applications of *limma*'s linear modelling strategy beyond the intended analysis of gene expression data have been made in a variety of applications, including the analysis of data from Nuclear Magnetic Resonance spectroscopy, PCR (including Nanostring), quantitative proteomics [7], DNA methylation arrays and comparative ChIP-seq [33].

As the cost of collecting genome-wide profiles continues to fall, we expect the popularity of this approach to continue to grow, with new applications in the analysis of single cell gene expression data, CRISPR/Cas9 knock-out screens and methylation analysis [48].

Funding

This research was supported by NHMRC Project grant 1050661 (MER, GKS), Project Grant 1023454 (GKS, MER, WS), Program grant 1054618 (GKS), Victorian State Government Operational Infrastructure Support and Australian Government NHMRC IRIISS.

Acknowledgements

We are grateful to our colleagues who have been actively involved in the *limma* project over the years, including James Wettenhall, Jeremy Silver, Davis McCarthy, Natalie Thorne, Aaron Lun, Alicia Oshlack, Ken Simpson, Yang Liao, Yunshun Chen and Carolyn de Graaf. The basic design of the `RGList` and `MAList` classes for two-color microarrays was based on similar objects defined by the *sma* package written by Yee-Hwa (Jean) Yang. The *limma* package has benefited from many other people who have made suggestions, reported bugs or contributed code including Kemal Akat, Naomi Altman, Gioia Altobelli, James Arnold, Alain Bateman, Ido Ben-Zvi, Henrik Bengtsson, Lourdes Peña Castillo, Dongseok Choi, Marcus Davy, Simon de Bernard, Ramon Diaz-Uriarte, Lars Eijssen, Pär Engström, Anthoula Gaigneaux, Robert Gentleman, Guido Hooiveld, Wolfgang Huber, Derek Janszen, William Kenworthy, Axel Klenk, Kevin Koh, Erik Kristiansson, Mette Langaas, Michael Lawrence, Gildas Le Corguille, Gregory Lefebvre, Philip Lijnzaad, Andrew Lynch, James MacDonald, Martin Maechler, Aaron Mackey, Steffen Moeller, Duarte Molha, Florian Nigsch, Ashley Ng, Mario Novkovic, Ron Ophir, Francois Pepin, Sergi Sayols Puig, Hubert Rehrauer, Davide Risso, Mark Robinson, Ken Simpson, Dario Strbenac, Laurentiu Adi Tarca, Dan Tenenbaum, Gregory Theiler, Ryan C. Thompson, Joern Toedling, Michael Turewicz, Björn Usadel, Chris Wilkinson, Beth Wilmot, Jean Yee Hwa Yang and John Zhang.

References

[1] Anders, S., Pyl, P.T., and Huber, W. (2015). HTSeq—a Python framework to work with high-throughput sequencing data. *Bioinformatics* 31, 166–169.

[2] Artmann, S., Jung, K., Bleckmann, A., and Beissbarth, T. (2012). Detection of simultaneous group effects in microRNA expression and related target gene sets. *PLoS One* 7, e38365.

[3] Ashburner, M., Ball, C.A., Blake, J.A., Botstein, D., Butler, H., Cherry, J.M., Davis, A.P., Dolinski, K., Dwight, S.S., Eppig, J.T., et al. (2000). Gene Ontology: tool for the unification of biology. *Nature Genetics* 25, 25–29.

[4] Asselin-Labat, M., Vaillant, F., Sheridan, J., Pal, B., Wu, D., Simpson, E., Yasuda, H., Smyth, G., Martin, T., Lindeman, G., and Visvader, J. (2010). Control of mammary stem cell function by steroid hormone signalling. *Nature* 465, 798–802.

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

[6] Bolstad, B.M., Irizarry, R.A., Åstrand, M., and Speed, T.P. (2003). A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. *Bioinformatics* 19, 185–193.

[7] Brusniak, M.Y., Bodenmiller, B., Campbell, D., Cooke, K., Eddes, J., Garbutt, A., Lau, H., Letarte, S., Mueller, L.N., Sharma, V., Vitek, O., Zhang, N., Aebersold, R., and Watts, J.D. (2008). Corra: Computational framework and tools for LC-MS discovery and targeted mass spectrometry-based proteomics. *BMC Bioinformatics* 9, 542.

[8] Caiazzo, M., Dell'Anno, M.T., Dvoretskova, E., Lazarevic, D., Taverna, S., Leo, D., Sotnikova, T.D., Menegon, A., Roncaglia, P., Colciago, G., Russo, G., Carninci, P., Pezzoli, G., Gainetdinov, R.R., Gustincich, S., Dityatev, A., and Broccoli, V. (2011). Direct generation of functional dopaminergic neurons from mouse and human fibroblasts. *Nature* 476, 224–227.

[9] Choi, J. (2013). Guide: a desktop application for analysing gene expression data. *BMC Genomics* 14, 688.

[10] Cleveland, W.S. (1993). *Visualizing Data*. AT&T Bell Laboratories, Murray Hill, New Jersey. ISBN 0-9634884-0-6.

[11] De Groot, P.J., Reiff, C., Mayer, C., and Muller, M. (2008). NuGO contributions to GenePattern. *Genes and Nutrition* 3, 143–146.

[12] Diaz-Uriarte, R., Alibes, A., Morrissey, E.R., Canada, A., Rueda, O.M., and Neves, M.L. (2007). Asterias: integrated analysis of expression and aCGH data using an open-source, web-based, parallelized software suite. *Nucleic Acids Research* 35, W75–80.

[13] Dudoit, S., Yang, Y.H., Callow, M.J., and Speed, T.P. (2002). Statistical methods for identifying differentially expressed genes in replicated cDNA microarray experiments. *Statistica Sinica* 12, 111–140.

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

[15] Gentleman, R. (2005). Reproducible research: a bioinformatics case study. *Statistical Applications in Genetics and Molecular Biology* 4, Article 2.

[16] 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., and Zhang, J. (2004). Bioconductor: open software development for computational biology and bioinformatics. *Genome Biology* 5, R80.

[17] Goeman, J.J. and Bühlmann, P. (2007). Analyzing gene expression data in terms of gene sets: methodological issues. *Bioinformatics* 23, 980–7.

[18] Hansen, K.D., Irizarry, R.A., and Zhijin, W. (2012). Removing technical variability in RNA-seq data using conditional quantile normalization. *Biostatistics* 13, 204–216.

[19] Hoffmann, R., Seidl, T., Neeb, M., Rolink, A., and Melchers, F. (2002). Changes in gene expression profiles in developing b cells of murine bone marrow. *Genome Res* 12, 98–111.

[20] Hubert, F.X., Kinkel, S.A., Crewther, P.E., Cannon, P.Z.F., Webster, K.E., Link, M., Uibo, R., O'Bryan, M.K., Meager, A., Forehan, S.P., Smyth, G.K., Mittaz, L., Antonarakis, S.E., Peterson, P., Heath, W.R., and Scott, H.S. (2009). Aire-deficient C57BL/6 mice mimicking the common human 13-base pair deletion mutation present with only a mild autoimmune phenotype. *The Journal of Immunology* 182, 3902–3918.

[21] Kooperberg, C., Fazzio, T.G., Delrow, J.J., and Tsukiyama, T. (2002). Improved background correction for spotted DNA microarrays. *Journal of Computational Biology* 9, 55–66.

[22] Langaas, M., Lindqvist, B., and Ferkingstad, E. (2005). Estimating the proportion of true null hypotheses, with application to DNA microarray data. *Journal of the Royal Statistical Society Series B* 67, 555–572.

[23] Langsrud, Ø. (2005). Rotation tests. *Statist Comput* 15, 53–60.

[24] Law, C.W., Chen, Y., Shi, W., and Smyth, G.K. (2014). Voom: precision weights unlock linear model analysis tools for RNA-seq read counts. *Genome Biology* 15, R29.

[25] Leisch, F. (2002). Sweave: Dynamic generation of statistical reports using literate data analysis. In W. Härdle and B. Rönz, editors, *Compstat 2002 - Proceedings in Computational Statistics*, pages 575–580. Physica Verlag, Heidelberg. ISBN 3-7908-1517-9.

[26] Lemoine, S., Combes, F., Servant, N., and Le Crom, S. (2006). Goulphar: rapid access and expertise for standard two-color microarray normalization methods. *BMC Bioinformatics* 7, 467.

[27] Li, B. and Dewey, C. (2011). RSEM: accurate transcript quantification from RNA-Seq data with or without a reference genome. *BMC Bioinformatics* 12, 323.

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

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

[30] Lim, E., Vaillant, F., Wu, D., Forrest, N., Pal, B., Hart, A., Asselin-Labat, M., Gyorki, D., Ward, T., Partanen, A., et al. (2009). Aberrant luminal progenitors as the candidate target population for basal tumor development in BRCA1 mutation carriers. *Nature Medicine* 15, 907–913.

[31] Lim, E., Wu, D., Pal, B., Bouras, T., Asselin-Labat, M., Vaillant, F., Yagita, H., Lindeman, G., Smyth, G., and Visvader, J. (2010). Transcriptome analyses of mouse and human mammary cell subpopulations reveal multiple conserved genes and pathways. *Breast Cancer Research* 12, R21.

[32] Liu, G.J., Cimmino, L., Jude, J.G., Hu, Y., Witkowski, M.T., McKenzie, M.D., Kartal-Kaess, M., Best, S.A., Tuohey, L., Liao, Y., Shi, W., Mullighan, C.G., Farrar, M.A., Nutt, S.L., Smyth, G.K., Zuber, J., and Dickins, R.A. (2014). Pax5 loss imposes a reversible differentiation block in B progenitor acute lymphoblastic leukemia. *Genes & Development* 28, 1337–1350.

[33] Lun, A.T.L. and Smyth, G.K. (2014). De novo detection of differentially bound regions for ChIP-seq data using peaks and windows: controlling error rates correctly. *Nucleic Acids Research* 42, e95.

[34] Majewski, I.J., Ritchie, M.E., Phipson, B., Corbin, J., Pakusch, M., Ebert, A., Busslinger, M., Koseki, H., Hu, Y., Smyth, G.K., et al. (2010). Opposing roles of polycomb repressive complexes in hematopoietic stem and progenitor cells. *Blood* 116, 731–739.

[35] Mannsperger, H.A., Gade, S., Henjes, F., Beissbarth, T., and Korf, U. (2010). Rppanalyzer: Analysis of reverse-phase protein array data. *Bioinformatics* 26, 2202–2203.

[36] Martin, B.J. and Altman, D.G. (1986). Statistical methods for assessing agreement between two methods of clinical measurement. *The Lancet* 327, 307–310.

[37] McCarthy, D.J. and Smyth, G.K. (2009). Testing significance relative to a fold-change threshold is a TREAT. *Bioinformatics* 25, 765–771.

[38] Michaud, J., Simpson, K., Escher, R., Buchet-Poyau, K., Beissbarth, T., Carmichael, C., Ritchie, M., Schütz, F., Cannon, P., Liu, M., et al. (2008). Integrative analysis of runx1 downstream pathways and target genes. *BMC genomics* 9, 363.

[39] Mootha, V.K., Lindgren, C.M., Eriksson, K.F., Subramanian, A., Sihag, S., Lehar, J., Puigserver, P., Carlsson, E., Ridderstrale, M., Laurila, E., Houstis, N., Daly, M.J., Patterson, N., Mesirov, J.P., Golub, T.R., Tamayo, P., Spiegelman, B., Lander, E.S., Hirschhorn, J.N., Altshuler, D., and Groop, L.C. (2003). PGC-1alpha-responsive genes involved in oxidative phosphorylation are coordinately downregulated in human diabetes. *Nat Genet* 34, 267–73.

[40] Morris, C.N. (1983). Parametric empirical Bayes inference: theory and applications. *Journal of the American Statistical Association* 78, 47–55.

[41] Mosig, M.O., Lipkin, E., Khutoreskaya, G., Tchourzyna, E., Soller, M., and Friedmann, A. (2001). A whole genome scan for quantitative trait loci affecting milk protein percentage in

Israeli-Holstein cattle, by means of selective milk DNA pooling in a daughter design, using an adjusted false discovery rate criterion. *Genetics* 157, 1683–1698.

[42] Nettleton, D., Hwang, J.G., Caldo, R.A., and Wise, R.P. (2006). Estimating the number of true null hypotheses from a histogram of p values. *Journal of Agricultural, Biological, and Environmental Statistics* 11, 337–356.

[43] Oshlack, A., Emslie, D., Corcoran, L., and Smyth, G. (2007). Normalization of boutique two-color microarrays with a high proportion of differentially expressed probes. *Genome Biology* 8, R2.

[44] Peart, M., Smyth, G., Van Laar, R., Bowtell, D., Richon, V., Marks, P., Holloway, A., and Johnstone, R. (2005). Identification and functional significance of genes regulated by structurally different histone deacetylase inhibitors. *Proceedings of the National Academy of Sciences of the United States of America* 102, 3697–3702.

[45] Petryszak, R., Burdett, T., Fiorelli, B., Fonseca, N.A., Gonzalez-Porta, M., Hastings, E., Huber, W., Jupp, S., Keays, M., Kryvych, N., McMurry, J., Marioni, J.C., Malone, J., Megy, K., Rustici, G., Tang, A.Y., Taubert, J., Williams, E., Mannion, O., Parkinson, H.E., and Brazma, A. (2014). Expression Atlas update—a database of gene and transcript expression from microarray and sequencing-based functional genomics experiments. *Nucleic Acids Research* 42, D926–932.

[46] Phipson, B. (2013). *Empirical Bayes modelling of expression profiles and their associations.* Ph.D. thesis, Department of Mathematics and Statistics, University of Melbourne.

[47] Phipson, B., Lee, S., Majewski, I.J., Alexander, W.S., and Smyth, G.K. (2013). Empirical Bayes in the presence of exceptional cases, with application to microarray data. Technical report, Bioinformatics Division, Walter and Eliza Hall Institute of Medical Research, Melbourne, Australia.

[48] Phipson, B. and Oshlack, A. (2014). DiffVar: A new method for detecting differential variability with application to methylation in cancer and aging. *Genome Biology* 15, 465.

[49] Psarros, M., Heber, S., Sick, M., Thoppae, G., Harshman, K., and Sick, B. (2005). RACE: Remote Analysis Computation for gene Expression data. *Nucleic Acids Research* 33, W638–43.

[50] R Development Core Team (2014). *R: A language and environment for statistical computing.* R Foundation for Statistical Computing, Vienna.

[51] Rainer, J., Sanchez-Cabo, F., Stocker, G., Sturn, A., and Trajanoski, Z. (2006). CARMAweb: comprehensive R- and Bioconductor-based web service for microarray data analysis. *Nucleic Acids Research* 34, W498–503.

[52] Rapaport, F., Khanin, R., Liang, Y., Pirun, M., Krek, A., Zumbo, P., Mason, C.E., Socci, N.D., and Betel, D. (2013). Comprehensive evaluation of differential gene expression analysis methods for RNA-seq data. *Genome biology* 14, R95.

[53] Rehrauer, H., Zoller, S., and Schlapbach, R. (2007). MAGMA: analysis of two-channel microarrays made easy. *Nucleic Acids Research* 35, W86–90.

[54] Ritchie, M., Diyagama, D., Neilson, J., Van Laar, R., Dobrovic, A., Holloway, A., and Smyth, G. (2006). Empirical array quality weights in the analysis of microarray data. *BMC bioinformatics* 7, 261.

[55] Ritchie, M., Silver, J., Oshlack, A., Holmes, M., Diyagama, D., Holloway, A., and Smyth, G. (2007). A comparison of background correction methods for two-colour microarrays. *Bioinformatics* 23, 2700–2707.

[56] Ritchie, M.E. (2004). *Quantitative quality control and background correction for two-colour microarray data*. Ph.D. thesis, Department of Medical Biology, University of Melbourne.

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

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

[59] Sartor, M.A., Tomlinson, C.R., Wesselkamper, S.C., Sivaganesan, S., Leikauf, G.D., and Medvedovic, M. (2006). Intensity-based hierarchical bayes method improves testing for differentially expressed genes in microarray experiments. *BMC bioinformatics* 7, 538.

[60] Shi, W., De Graaf, C., Kinkel, S., Achtman, A., Baldwin, T., Schofield, L., Scott, H., Hilton, D., and Smyth, G. (2010). Estimating the proportion of microarray probes expressed in an RNA sample. *Nucleic Acids Research* 38, 2168–2176.

[61] Shi, W., Oshlack, A., and Smyth, G. (2010). Optimizing the noise versus bias trade-off for Illumina Whole Genome Expression Beadchips. *Nucleic Acids Research* 38, e204.

[62] Silver, J., Ritchie, M., and Smyth, G. (2009). Microarray background correction: maximum likelihood estimation for the normal–exponential convolution. *Biostatistics* 10, 352–363.

[63] Smyth, G. (2004). Linear models and empirical Bayes methods for assessing differential expression in microarray experiments. *Statistical Applications in Genetics and Molecular Biology* 3, Article 3.

[64] Smyth, G. (2005). Limma: linear models for microarray data. In R. Gentleman, V. Carey, S. Dudoit, R. Irizarry, and W. Huber, editors, *Bioinformatics and computational biology solutions using R and Bioconductor*, pages 397–420. Springer, New York.

[65] Smyth, G., Michaud, J., and Scott, H. (2005). Use of within-array replicate spots for assessing differential expression in microarray experiments. *Bioinformatics* 21, 2067–2075.

[66] Smyth, G. and Speed, T. (2003). Normalization of cDNA microarray data. *Methods* 31, 265–273.

[67] Smyth, G.K. and Altman, N.S. (2013). Separate-channel analysis of two-channel microarrays: recovering inter-spot information. *BMC Bioinformatics* 14, 165.

[68] Su, Z., Labaj, P.P., Li, S., Thierry-Mieg, J., Thierry-Mieg, D., Shi, W., Wang, C., Schroth, G.P., Setterquist, R.A., Thompson, J.F., Jones, W.D., Xiao, W., Xu, W., Jensen, R.V., Kelly, R., Xu, J., Conesa, A., Furlanello, C., Gao, H., Hong, H., Jafari, N., Letovsky, S., Liao, Y., Lu, F., Oakeley, E.J., Peng, Z., Praul, C.A., Santoyo-Lopez, J., Scherer, A., Shi, T., Smyth, G.K., Staedtler, F., Sykacek, P., Tan, X.X., Thompson, E.A., Vandesompele,

J., Wang, M.D., Eli, J.W., Wolfinger, R.D., Zavadil, J., Auerbach, S.S., Bao, W., Binder, H., Blomquist, T., Brilliant, M.H., Bushel, P.R., Cai, W., Catalano, J.G., Chang, C.W., Chen, T., Chen, G., Chen, R., Chierici, M., Chu, T.M., Clevert, D.A., Deng, Y., Derti, A., Devanara, V., Dong, Z., Dopazo, J., Du, T., Fang, H., Fang, Y., Fasold, M., Fernandez, A., Fischer, M., Furió-Tarí, P., Fuscoe, J.C., Gaj, S., Gandara, J., Gao, H., Ge, W., Gondo, Y., Gong, B., Gong, M., Gong, Z., Green, B., Guo, C., Guo, L., Guo, L.W., Hadfield, J., Hellemans, J., Hochreiter, S., Jia, M., Jian, M., Johnson, C.D., Kay, S., Kleinjans, J., Lababidi, S., Levy, S., Li, Q.Z., Li, L., Li, P., Li, Y., Li, H., Li, J., Li, S., Lin, S.M., López, F.J., Lu, X., Luo, H., Ma, X., Meehan, J., Megherbi, D.B., Mei, N., Mu, B., Ning, B., Pandey, A., Pérez-Florido, J., Perkins, R.G., Peters, R., Phan, J.H., Pirooznia, M., Qian, F., Qing, T., Rainbow, L., Rocca-Serra, P., Sambourg, L., Sansone, S.A., Schwartz, S., Shah, R., Shen, J., Smith, T.M., Stegle, O., Stralis-Paves, N., Stupka, E., Suzuki, Y., Szkotnicki, L.T., Tinning, M., Tu, B., van, J., Vela, A., Venturini, E., Walker, S.J., Wan, L., Wang, W., Wang, J., Wang, J., Wieben, E.D., Willey, J.C., Wu, P.Y., Xuan, J., Yang, Y., Ye, Z., Yin, Y., Yu, Y., Yuan, Y.C., Zhang, J., Zhang, K.K., Zhang, W., Zhang, W., Zhang, Y., Zhao, C., Zheng, Y., Zhou, Y., Zumbo, P., Tong, W., Kreil, D.P., Mason, C.E., and Shi, L. (2014). A comprehensive assessment of RNA-seq accuracy, reproducibility and information content by the Sequencing Quality Control Consortium. *Nature Biotechnology* 32, 903–914.

[69] Subramanian, A., Tamayo, P., Mootha, V.K., Mukherjee, S., Ebert, B.L., Gillette, M.A., Paulovich, A., Pomeroy, S.L., Golub, T.R., Lander, E.S., and Mesirov, J.P. (2005). Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proc Natl Acad Sci USA* 102, 15545–50.

[70] Wettenhall, J.M., Simpson, K.M., Satterley, K., and Smyth, G.K. (2006). affylmGUI: a graphical user interface for linear modeling of single channel microarray data. *Bioinformatics* 22, 897–899.

[71] Wettenhall, J.M. and Smyth, G.K. (2004). limmaGUI: a graphical user interface for linear modeling of microarray data. *Bioinformatics* 20, 3705–3706.

[72] Wu, D., Hu, Y., Tong, S., Williams, B.R., Smyth, G.K., and Gantier, M.P. (2013). The use of miRNA microarrays for the analysis of cancer samples with global miRNA decrease. *RNA* 19, 876–888.

[73] Wu, D., Lim, E., Vaillant, F., Asselin-Labat, M., Visvader, J., and Smyth, G. (2010). ROAST: rotation gene set tests for complex microarray experiments. *Bioinformatics* 26, 2176–2182.

[74] Wu, D., Pang, Y., Wilkerson, M.D., Wang, D., Hammerman, P.S., and Liu, J.S. (2013). Gene-expression data integration to squamous cell lung cancer subtypes reveals drug sensitivity. *British Journal of Cancer* 109, 1599–608.

[75] Wu, D. and Smyth, G. (2012). Camera: a competitive gene set test accounting for inter-gene correlation. *Nucleic Acids Research* 40, e133.

[76] Xia, X., McClelland, M., and Wang, Y. (2005). Webarray: an online platform for microarray data analysis. *BMC Bioinformatics* 6, 306.

[77] Xie, Y. (2013). *Dynamic documents with R and knitr*. CRC Press, Boca Raton, FL. ISBN 978-1482203530.

[78] Yang, Y.H., Dudoit, S., Luu, P., Lin, D.M., Peng, V., Ngai, J., and Speed, T.P. (2002). Normalization for cDNA microarray data: a robust composite method addressing single and multiple slide systematic variation. *Nucleic Acids Research* 30, e15.

[79] Yang, Y.H. and Thorne, N.P. (2003). Normalization for two-color cDNA microarray data. In D.R. Goldstein, editor, *Science and Statistics: A Festschrift for Terry Speed*, pages 403–418. Institute of Mathematical Statistics Lecture Notes – Monograph Series, Volume 40.

[80] Young, M., Wakefield, M., Smyth, G., and Oshlack, A. (2010). Gene ontology analysis for RNA-seq: accounting for selection bias. *Genome Biology* 11, R14.