edgeR 4.0: powerful differential analysis of sequencing data with expanded functionality and improved support for small counts and larger datasets

Yunshun Chen[1,2], Lizhong Chen[1,3], Aaron T. L. Lun[4], Pedro L. Baldoni[1,3] and Gordon K. Smyth[1,5,*]

[1]Bioinformatics Division, WEHI, Parkville, VIC 3052, Australia
[2]ACRF Cancer Biology and Stem Cells Division, WEHI, Parkville, VIC 3052, Australia
[3]Department of Medical Biology, The University of Melbourne, Parkville, VIC 3010, Australia
[4]Computational Sciences, Genentech Inc, 1 DNA Way, South San Francisco, CA 94080, USA
[5]School of Mathematics and Statistics, The University of Melbourne, Parkville, VIC 3010, Australia
[*]To whom correspondence should be addressed. Email: smyth@wehi.edu.au

Abstract

edgeR is an R/Bioconductor software package for differential analyses of sequencing data in the form of read counts for genes or genomic features. Over the past 15 years, edgeR has been a popular choice for statistical analysis of data from sequencing technologies such as RNA-seq or ChIP-seq. edgeR pioneered the use of the negative binomial distribution to model read count data with replicates and the use of generalized linear models to analyse complex experimental designs. edgeR implements empirical Bayes moderation methods to allow reliable inference when the number of replicates is small. This article announces edgeR version 4, which includes new developments across a range of application areas. Infrastructure improvements include support for fractional counts, implementation of model fitting in C++, and a new statistical treatment of the quasi-likelihood pipeline that improves accuracy for small counts. The revised package has new functionality for differential methylation analysis, differential transcript expression, differential transcript and exon usage, testing relative to a fold-change threshold and pathway analysis. This article reviews the statistical framework and computational implementation of edgeR, briefly summarizing all the existing features and functionalities but with special attention to new features and those that have not been described previously.

Introduction

Next Generation Sequencing (NGS) has revolutionized biomedical research over the past 15–20 years. RNA-seq has become the standard technology for profiling gene and transcript expression [1, 2] while other technologies such as ChIP-seq, ATAC-seq, CUT&Tag, BS-seq and Hi-C allow high-resolution exploration of the molecular mechanisms by which expression is regulated [3].

edgeR is an R software package for differential analyses of data arising from NGS or similar technologies in the form of sequence read counts for genes or genomic features [4, 5, 6]. It is particularly designed to detect genes or features that have changed abundance levels between experimental conditions or cell types. edgeR pioneered the use of negative binomial (NB) generalized linear models (GLMs) to model read counts in genomic research [7, 8]. edgeR implements a range of novel statistical methods including methods for borrowing information between genes, a strategy that is essential for genomic experiments with small sample sizes [4, 7, 9, 10]. It has become the underlying analysis engine for a wide range of sequencing technologies including ChIP-seq [11], Hi-C [12], bisulfite sequencing [13] and even proteomics [14, 15, 16]. The use of an explicit probabilistic count distribution allows edgeR to make meaningful inferences even for very low counts and provides a distinction with normal-based methods such as limma [17].

The edgeR package has undergone a number of major revisions since it was first released as part of the Bioconductor project in 2008 [18]. The original edgeR v1 pipeline (now called the "classic" pipeline) used exact conditional likelihood to achieve unbiased estimation of the NB-dispersion, exact NB tests to make pairwise comparisons between groups, and weighted likelihood empirical Bayes to borrow strength between genes [4, 5, 6]. These innovative statistical approaches allowed edgeR to achieve stable and reliable results even for experiments with very small numbers of biological replicates.

Full GLM functionality was introduced to edgeR in September 2010, allowing edgeR to model arbitrarily complex experiments including multiple treatment factors, batch effects and continuous covariates. All the original functionality was transferred to the GLM context, with Cox-Reid approximate conditional inference replacing the exact conditional likelihoods and likelihood ratio tests replacing the exact NB tests [7]. The edgeR GLM pipeline was released as edgeR v2 in 2011.

The second major revision was the introduction of quasi-likelihood (QL) methods in January 2012 [19, 20, 10, 21]. The QL model added a second dispersion parameter, the QL-dispersion, which increased edgeR's ability to model technical as well as biological sources of variability. Another key advantage was that the QL-dispersions could be estimated by applying limma's parametric empirical Bayes (EB) procedures to the genewise GLM deviances, which in turn enabled edgeR to leverage some of limma's exact small-sample theory [22]. The amount of EB moderation applied to the genewise dispersions could be optimized for the specific data at hand [9]. The GLM likelihood ratio tests could be replaced by quasi-F-tests, which allow for the uncertainty with which the genewise dispersions are estimated and thereby provide rigorous control of the false discovery rate (FDR) even for small sample sizes [19]. The edgeR QL pipeline was released as part of edgeR v3 in 2012.

This article announces edgeR version 4, which was released in October 2023 and includes new developments across a range of application areas. The revised package implements two fundamental changes that improve edgeR's treatment of small counts and affect most analyses going forward. The first is a continuous generalization of the NB distribution that allows edgeR to accept fractional counts without rounding [23]. The second is a major revision of the QL pipeline based on improvements to the classical statistical theory underlying GLMs. The revision ensures unbiased QL-dispersion estimates even when the read counts are very small and dramatically reduces computation time for large datasets with many samples. Meanwhile, substantial parts of the edgeR package have been rewritten in C++ to increase speed and reduce memory usage.

The package also has new functionality for differential methylation analysis [13], differential transcript expression [23], differential exon usage, differential transcript usage, testing relative to a fold-change threshold and pathway analysis.

This article reviews the statistical framework and computational implementation of edgeR, briefly summarizing all the existing features and functionalities but with special attention to the new features and those that have not been described previously.

Materials and methods

Summary of the edgeR distributional model

Log-linear models.

The input to edgeR is a matrix of sequence read counts with rows corresponding to genomic features and columns to biological samples. The rows often represent genes, but can correspond to almost any meaningful genomic feature including transcripts, exons, exon-exon junctions, genomic regions, methylation sites or DNA-DNA interactions.

Write μ_{gi} for the expected number of reads assigned to feature g in sample i. edgeR assumes that μ_{gi} can be modelled as a log-linear model

$$\log \mu_{gi} = x_i^T \beta_g + \log L_i$$

where x_i is a covariate vector specifying the experimental conditions applied to sample i, β_g is a coefficient vector that captures the experimental effects and log-fold-changes, and L_i is the effective library size (sequencing depth) for sample i. From a mathematical point of view, the purpose of edgeR is to test hypotheses about β_g.

Technical and biological variation produce a quadratic mean-variance relationship.

Write y_{gi} for the actual number of reads assigned to feature g in sample i. edgeR assumes that technical replicates produced from the same RNA or DNA sample (and with the same library size) would result in quasi-Poisson repeatability with variance $\mathrm{var}(y_{gi}) = \sigma_g^2 \mu_{gi}$. This represents the measurement error associated with sequencing and read alignment of a single sample. In ideal circumstances, the Poisson dispersion σ_g^2 is close to 1 [24], but can be much greater than 1 in the presence of PCR duplication or if reads are probabilistically assigned to features [23]. Single-cell data and transcript-level quantification are two application areas for which Poisson dispersions larger than 1 are common.

edgeR also assumes that the true abundance of feature g varies between biological replicates according to a gamma distribution with squared coefficient of variation equal to ψ_g. The read count distribution therefore follows a mixture distribution across biological replicates with a quadratic mean-variance relationship of the form

$$\mathrm{var}(y_{gi}) = \sigma_g^2 \mu_{gi} + \psi_g \mu_{gi}^2$$

where σ_g^2 and ψ_g represent technical and biological variation respectively[7, 23]. In edgeR terminology, σ_g^2 is the quasi-likelihood dispersion (QL-dispersion) and ψ_g is the squared biological coefficient of variation (SBCV).

The mean-variance relationship is usually rewritten as the variance function for a quasi-NB GLM,

$$\mathrm{var}(y_{gi}) = \sigma_g^2 \left(\mu_{gi} + \phi_g \mu_{gi}^2 \right)$$

where $\phi_g = \psi_g / \sigma_g^2$ is the NB dispersion parameter. If $\sigma_g^2 = 1$, then the counts y_{gi} can be considered to follow a NB distribution. If $\sigma_g^2 \neq 1$, then y_{gi} is distributed as a gamma mixture of quasi-Poisson distributions and quasi-NB methods are used.

Empirical Bayes.

edgeR implements two empirical Bayes estimation strategies, a weighted likelihood empirical Bayes strategy for estimating the NB dispersions, and a limma-style parametric empirical Bayes strategy for estimating the QL-dispersions. The edgeR's classic and GLM pipelines assume the QL-dispersions are equal to 1 and use weighted likelihood empirical Bayes to estimate feature-specific NB-dispersions. The more recent QL edgeR pipelines set the NB-dispersions to global (trended or constant) values and then apply limma-style empirical Bayes to estimate feature-specific QL-dispersions.

Divided counts.

Baldoni *et al.* [23] introduced the new concept of divided counts. If a reliable estimate of σ_g^2 is available, then one can form the divided counts $z_{gi} = y_{gi} / \sigma_g^2$. The divided counts have mean

$$E(z_{gi}) = \nu_{gi} = \mu_{gi} / \sigma_g^2$$

and variance

$$\mathrm{var}(z_{gi}) = \nu_{gi} + \psi_g \nu_{gi}^2,$$

showing that the divided counts have the same variance function as a NB distribution with the same SBCV as the original data but with the technical over-dispersion removed. The divided counts have reduced library sizes compared to the original counts, showing that the QL-dispersions can be interpreted as reducing the effective information content of each count. In the presence of QL-dispersion, the counts have the same information content as the divided counts without QL-dispersion.

Divided counts are currently used by edgeR only for transcript-level expression analyses, when testing for differential transcript expression or differential transcript usage. The strategy is potentially useful however whenever extra information is available about the QL-dispersions additional to the read counts themselves.

Observation-level weights and library sizes.

edgeR allows users to specify library sizes for individual observations if desired, which generalizes the log-linear model to

$$\log \mu_{gi} = x_i^T \beta_g + \log L_{gi}$$

where L_{gi} is the observation-specific effective library size. In edgeR terminology, the matrix of $\log L_{gi}$ values is the offset matrix. Observation-specific library sizes can be used to implement non-linear normalization in edgeR [25, 26, 27, 12, 11] or to allow for transcript length effects [28].

edgeR v4 also allows observation-specific dispersions by

$$\mathrm{var}(y_{gi}) = \frac{\sigma_g^2}{w_{gi}} \left(\mu_{gi} + \phi_g \mu_{gi}^2 \right)$$

where the w_{gi} are known GLM weights. The weights can be used to down-weight outlier samples or observations [29].

Example datasets

Example edgeR diagnostic plots are shown in Figure 2. The MDS plot in panel (a) is generated from the RNA-seq profiles of 882 human breast tumour samples from The Cancer Genome Atlas (TCGA) [30]. Each sample is classified as one of the five breast cancer intrinsic subtypes based on PAM50 gene signatures [31]. The data was downloaded from the TCGA Data Portal (https://tcga-data.nci.nih.gov) in the form of genewise read counts and processed as previously described [32]. Panels (b), (c) and (d) display a BCV plot, a QL-dispersion plot, and a mean-difference (MD) plot respectively from the RNA-seq differential expression analysis described by Chen *et al.* [10]. The RNA-seq data for this analysis is from Fu *et al.* [33]. Panel (e) shows a differential splicing plot for the *Foxp1* gene. The *Foxp1* exons that are differentially used under the comparison are highlighted in the plot. The RNA-seq data for the differential exon usage analysis is from Fu *et al.* [34].

Results

Statistical principles

Differential analyses.

edgeR is designed to conduct differential analyses of sequence read counts obtained from next generation sequencing (NGS) or similar technologies [6]. A dataset consists of a matrix of read counts where rows represent genomic features such as genes, exons, transcripts, genomic intervals, methylation sites or DNA-DNA interactions, and columns are samples associated with different biological groups or treatment conditions. In the following, we will refer to the genomic features as "genes", with the understanding that the methodology is applicable to any genomic feature for which read counts can be obtained. The fact that edgeR focuses on differential results rather than on quantification of abundance means that edgeR can focus directly on statistical analysis of the read counts. In most analyses there is no need to adjust the read counts for expression biases such as gene length, GC content or mappability because, under the null hypothesis of no differential expression, such biases affect all samples equally and therefore cancel out of differential analyses.

The negative binomial distribution captures biological variations.

edgeR assumes that the read counts follow NB distributions, an assumption that can be justified by a mixture model in which the true expression of each gene varies between biological replicates with constant coefficient of variation and the measurement error for individual samples follows a Poisson law [7, 23]. The use of an explicit count distribution allows meaningful probability calculations even for very small counts. The NB distribution implies a quadratic mean-variance relationship in which technical variation dominates for small counts and biological variation dominates for large counts. This assumption has been validated by extensive data analyses showing that NGS datasets do show the sort of quadratic mean-variance relationship that the NB distribution implies. The NB dispersion parameter estimates the coefficient of variation of the true expression levels. The square-root of the NB dispersion is called the biological coefficient of variation (BCV) in edgeR [7].

Generalized linear models handle complex experiments.

Another advantage of the NB distribution is that it belongs to the family of distributions for which generalized linear modelling is possible [8]. The NB GLM framework adopted in edgeR provides great flexibility for analysing complex multifactor experiments. Different experimental conditions and batch effects can be easily handled by the design matrix. It enables time-course analyses by incorporating a spline curve across different time points into the design [35]. Very general analyses are possible, for example gene-gene correlations can be detected by adding log-expression values for a target gene as a covariate column in the design matrix.

Unbiased estimation of the NB-dispersion.

Obtaining an unbiased estimator of the NB-dispersion is not straightforward because regular maximum likelihood estimation is biased leading to under-estimated dispersions [5]. Unbiased estimation requires the likelihood be adjusted for estimation of the linear model parameters by conditioning on the linear model estimators, a process analogous to REML for normal-based models [36]. edgeR originally implemented an exact conditional likelihood approach, which was very effective but was applicable only to relatively simple oneway experiments with multiple groups but no blocks or covariates [5]. In edgeR v2, Cox-Reid adjusted profile likelihood (APL) was adopted under the more flexible GLM framework [37, 7]. This allows unbiased estimation of the NB-dispersions even for analyses with complex experimental designs [7].

Borrowing information by weighted likelihood empirical Bayes.

Conditional likelihood can be used to obtain genewise dispersion estimates, but simple genewise estimation is not reliable for genomic datasets, which typically have many genes or genomic features but relatively few biological replicates. Given the parallel nature of genomic data, whereby the same log-linear model is fitted in parallel to a large number of genes, the accuracy of dispersion estimate for each individual gene can be greatly improved by adopting an empirical Bayes strategy to borrow strength between genes. Empirical Bayes variance estimation has proved extremely effective for the limma package [38, 22], but a similar parametric empirical Bayes approach is not available for the NB dispersion because there is no conjugate prior distribution for the NB dispersion parameter. Instead, a weighted likelihood approach was implemented in edgeR to produce an approximate empirical Bayes strategy [4, 7, 39, 9]. The weighted likelihood approach has the advantage that it makes no assumptions about the shape of the prior distribution but instead adapts to the data at hand. Under the NB GLM framework, a common NB-dispersion for all the genes can be estimated by maximizing the total APLs of all the genes with equal weights. To account for the fact that genes with lower expression level tend to have larger dispersions, trended NB dispersions are introduced and estimated by maximizing the locally shared APL formed by genes with similar expression levels. Finally, a gene-specific NB-dispersion can be obtained by maximizing the weighted APLs formed by the individual APL of that gene and its locally shared APL with a proper weight. The weight represents the amount of prior information borrowed from the neighbouring genes. The final gene-specific NB-dispersion estimate can be considered as a compromise between the estimates obtained from the data for that gene alone and from the other neighbouring genes. This approach has proven to be highly effective for stabilizing dispersion estimates when the sample size is small. Initially, a preset weighting was used for the prior distribution [7]. Later, a QL strategy using the deviances was used to optimize the prior weight, leveraging limma's hyperparameter estimation [9]. A variation of the weighted likelihood empirical Bayes algorithm that is more robust to observational outliers was also implemented [29].

Quasi-likelihood.

edgeR v3 introduced a QL generalization of the NB model [19, 10]. In edgeR's QL pipeline, the NB-dispersions are used to model the global trend in biological variation while the QL-dispersions are used to accommodate gene-specific variability. The first step is to estimate the global NB-dispersion trend, same as for the weighted likelihood approach described above. The QL-dispersions are then estimated by the same empirical Bayes moderation strategy as in limma but with the GLM residual deviances in place of the residual variances used by limma [22]. As in limma, the posterior QL-dispersion estimates are moderated towards a trended prior with prior degrees of freedom (df) estimated from the data as part of the empirical Bayes procedure.

The precision of the posterior QL-dispersion estimators is summarized by their posterior df, which is made up of the prior df plus the residual df. Quasi-F-tests are then used to test hypotheses with the posterior df entering as the residual df. The quasi-F-tests provide more rigorous error rate control than other analysis methods that do not fully reflect the uncertainty with which the QL-dispersions are estimated [19, 40]. QL has become the default recommended pipeline in edgeR because of its robustness and very reliable FDR control.

Hypothesis testing for general contrasts.

edgeR v1 was only able to make pairwise comparisons between treatment groups, but later versions of edgeR can test very general null hypotheses specified by any linear contrast of the linear model coefficients. This provides enormous flexibility for users. For example, one might compare one group to the average of other groups simply by specifying suitable contrast weights for the means of the different groups. Tests can be conducted for a single contrast or for several contrasts at a time, leading to an analysis of deviance test on several df [8, 10]. For example, one can conduct an ANOVA-like test for differences between several groups by specifying any set of contrasts that distinguish the groups. The edgeR GLM functions glmFit and glmLRT conduct likelihood ratio tests [7]. The edgeR QL functions glmQLFit and glmQLFTest conduct quasi F-tests where the likelihood ratio statistic is the numerator of the F-statistic and the QL-dispersion is the denominator [8, 10].

Supporting fractional counts.

edgeR was designed for analysing digital gene expression data, which usually comes in the format of integer counts, and edgeR assumes that read counts follow NB distributions, which has a discrete probability mass function defined on the non-negative integers. Tools that count reads overlapping genomic features such as featureCounts [41] or HTSeq [42] do indeed produce integer counts. However, the read counts can be fractional when the number of sequence reads originating from a gene or transcript is estimated in a probabilistic manner. RSEM, kallisto and Salmon are examples of software tools that output estimated gene and transcript counts that are not integers [43, 44, 45].

To handle fractional data without the need for rounding, a continuous generalization of the NB distribution has been implemented in edgeR. All the binomial coefficients of the NB probability mass function were replaced with gamma functions resulting in a continuous probability density function (PDF). A normalizing constant is required to ensure the density integrates to 1. There is no closed-form for the normalizing constant in the PDF, but it can be shown numerically that the normalizing constant is very close to 1 and is nearly constant with respect to the mean and dispersion parameters. The mean and variance of the continuous NB are also well approximated by the traditional mean and variance formulas for the NB. The continuous version of NB distribution allows for fractional counts while matching exactly with the discrete NB distribution when the counts are integers.

Correcting quasi-likelihood for bias.

Traditional quasi-likelihood statistical theory relies an a chisquare approximation to the GLM deviances, which can be justified by a saddlepoint approximation [39, 8]. The chisquare approximation to NB deviances is excellent when the NB-dispersion is relatively small and the counts are relatively large. Unfortunately, for single-cell data, the NB-dispersions are often large and the counts are small. For genes with very low counts, average count below 1 for example, the chisquare approximation underestimates the QL-dispersions.

In edgeR v3 we already implemented an adjustment for the residual df when some of the fitted values were exactly zero [21]. In edgeR v4, we have implemented a more comprehensive approach. We drop the use of the saddlepoint approximation and instead approximate the unit deviances by scaled chisquare random variables on fractional df. The df and the scaling factor are chosen to match the first two moments (the mean and variance) of the unit deviances exactly, given the fitted mean and the NB-dispersion for each observation. This approach yields very nearly unbiased QL-dispersion estimators even for fitted values close to zero and for quite large NB-dispersions. The new approach makes wide-reaching changes to the package because it produces different residual deviances and fractional residual df.

Larger datasets.

The edgeR v4 QL pipeline also allows edgeR to analyse larger datasets than previously. The new QL pipeline allows the estimateDisp function to be bypassed resulting in substantially reduced computation time for large datasets. A oneway multi-group analysis with 1000 samples and 10000 genes takes about 30 seconds on a laptop computer including dispersion estimation, GLM fitting and testing for differential expression. The same data with a more complex design matrix takes less than two minutes, depending on the number of predictors.

Analysing next generation sequencing data

Standard analysis steps.

edgeR offers a complete differential analysis pipeline from read counts to lists of differential genes and pathways. The standard steps in most analyses include data import, filtering out lowly expressed genes, normalization, data exploration, dispersion estimation, fitting GLMs and hypothesis testing. edgeR implements one or more functions for each of these steps (Figure 1).

Data import.

The minimum data that edgeR needs to conduct an analysis is a numeric matrix of counts and a factor or covariate distinguishing the samples. For most users, the first step is to assemble the read counts and annotation into a DGEList object, which is edgeR's data container. If users already have a numeric matrix in R. then the DGEList function converts it to a DGEList. If users have imported the counts and gene annotation from a text file into an data-frame using base R, then DGEList will process the data-frame and attempt to distinguish the counts from the annotation columns. edgeR's readDGE function will collate read count files written by Subread-featureCounts [41] or by HTSeq-count [42], both of which write one file per sample. Alternatively, the featureCounts2DGEList function imports output from Rsubread's featureCounts function [46] directly without the need to write intermediate files to disk.

10X Genomics Chromium is the most popular platform for single-cell RNA-seq. edgeR's read10X function reads output from the 10X Genomics Cellranger pipeline [47] and assembles the UMI counts into an integer matrix in R with accompanying cellular barcodes and gene symbols. Seurat is an extremely popular R package for statistical analysis of scRNA-seq [48]. edgeR's Seurat2PB function imports data from Seurat and aggregates it to pseudo-bulk form suitable for an edgeR analysis [49].

Salmon [45] and kallisto [44] have become very popular for quantifying RNA-seq data because of their speed and ability to estimate transcript-level expression. edgeR's catchSalmon and catchKallisto functions import output from Salmon or kallisto, read in transcript counts and estimate the assignment uncertainty for each transcript.

Bismark is a popular software tool for read mapping and methylation calling of bisulfite sequencing (BS-seq) data [50]. edgeR's readBismark2DGE function reads Bismark coverage files and collates the methylated and unmethylated read counts from multiple files into a DGEList object [13].

Filtering low count features.

Genes with very low read counts are filtered before downstream analysis. Filtering is done partly because very low expressed genes are of little biological interest and because NB-dispersion estimation is unreliable for such genes. The main reason however is that genes with too few reads cannot achieve statistical significance even if they truly are differentially abundant. Keeping such genes in the downstream analysis increases the amount of multiple testing and decreases statistical power without any compensating advantage. edgeR's filterByExpr function is used to keep only those genes or features that have sufficient counts to achieve statistical significance when meaningful differential abundance is present.

Effective library sizes and the offset matrix.

edgeR's normLibSizes function estimates an effective library size (ELS) for each sample. It estimates a normalization factor for each sample, which multiplies the raw library size to become the ELS. TMM (trimmed mean of M-values) [51] is the default normalization method but a new TMMwsp method (TMM with singleton pairing) has been added to provide more robust behaviour for sparse data with many zeros.

The log-ELSs become GLM offsets in downstream edgeR analyses. The offset matrix can also be set directly, which provides a way to reflect biases at the observation level that arise from factors such as GC content or gene length. edgeR supports custom normalization procedures implemented by external packages such as cqn [26], EDAseq [25] or tximport [28], by importing the appropriate offset matrix. edgeR's scaleOffset function ensures that the scale of externally defined offset matrices are consistent with library sizes while preserving the normalization results from those external methods.

Model fitting and differential analysis.

edgeR defines linear models in the same way as limma [52]. Users create a design matrix appropriate for their experimental design using standard base R functions. edgeR has the flexibility to work with any full-rank design matrix with any number of covariates or factor effects.

In edgeR v4, one can proceed directly from normalization to glmQLFit to fit GLMs and glmQLFTest to test hypotheses. In edgeR v4, the glmQLFit function estimates both NB and QL dispersions automatically.

In earlier edgeR versions, the NB-dispersions are first estimated by estimateDisp [9]. Then one proceeds to exactTest in the classic pipeline, to glmFit followed by glmLRT in the GLM pipeline, or to glmQLFit followed by glmQLFTest in the QL pipeline.

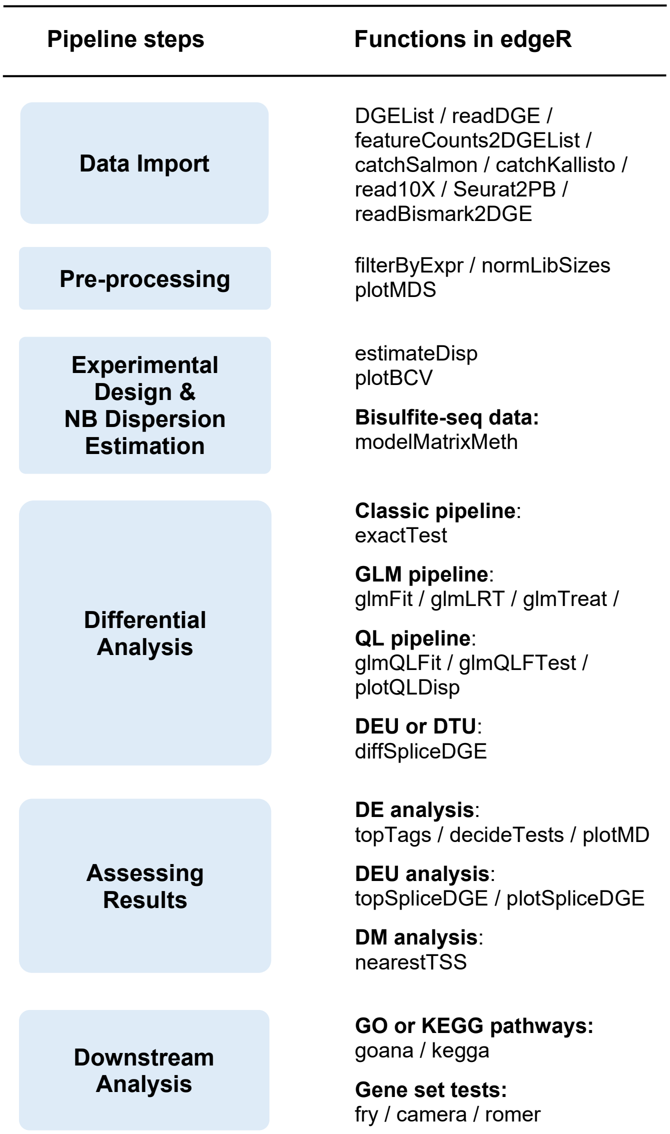
The functions glmFit and glmQLFit shrink the estimated log-fold-changes slightly towards zero to avoid infinite values and to improve prediction accuracy. edgeR implements the concept of a "prior count" to quantify the prior information determining the amount of shrinkage [53].

For all pipelines, the number of differential genes at any given FDR threshold can be shown by the decideTests function and the top differential genes can be displayed by the topTags function.

Testing differences relative to a fold-change threshold.

The glmLRT or glmQLFTest functions can be replaced by glmTreat if one wants to test differential abundance relative to a higher fold-change threshold.

The standard edgeR analysis identifies differential expression based on statistical significance regardless of how small the difference might be. There are circumstances where researchers are interested in studying genomic features of which the expression levels change by a certain amount. A few *ad hoc* approaches have been used to select features with large fold-changes. For example, some studies applied a fold-change cut-off and then ranked all the genes above that fold-change



Pipeline steps	Functions in edgeR
Data Import	DGEList / readDGE / featureCounts2DGEList / catchSalmon / catchKallisto / read10X / Seurat2PB / readBismark2DGE
Pre-processing	filterByExpr / normLibSizes plotMDS
Experimental Design & NB Dispersion Estimation	estimateDisp plotBCV **Bisulfite-seq data:** modelMatrixMeth
Differential Analysis	**Classic pipeline**: exactTest **GLM pipeline**: glmFit / glmLRT / glmTreat / **QL pipeline**: glmQLFit / glmQLFTest / plotQLDisp **DEU or DTU**: diffSpliceDGE
Assessing Results	**DE analysis**: topTags / decideTests / plotMD **DEU analysis**: topSpliceDGE / plotSpliceDGE **DM analysis**: nearestTSS
Downstream Analysis	**GO or KEGG pathways:** goana / kegga **Gene set tests:** fry / camera / romer

Figure 1: The edgeR workflow. The diagram shows the main steps in an edgeR analysis. Individual functions involved in each step are shown on the right.

threshold by p-value. There were also some cases where genes were first chosen according to a p-value cut-off and then sorted by their fold-changes. However, these *ad hoc* approaches tend to prioritise low expression genes and can lead to loss of FDR control.

To assess differential expression relative to a threshold in a statistically rigorous way, the TREAT method was developed under the limma empirical Bayes framework [54, 22]. It adopts a re-centred moderated t-statistic to provide an upper bound of the type I error rate. This approach yields an easily computable conservative p-value for testing against a fold-change threshold. We adopted the idea of TREAT and extended it to the edgeR NB GLM framework. Unlike TREAT, the edgeR approach computes the expectation rather than the maximum value of the type I error rate as the p-value of the test, a refinement that increases the statistical power of the test while still controlling the FDR. This method is implemented in the glmTreat function in edgeR, and it can be used under both the likelihood ratio test (LRT) and the QL F-test pipelines.

Gene set enrichment analysis.

It is often helpful to interpret differential results in terms of higher-order biological processes. The limma package [17] provides a number of functions to facilitate this interpretation in terms of standardized gene annotation or gene sets. The goana and kegga functions determine the overlap of a list of differential genes with categories in the Gene Ontology (GO) database [55] or with pathways in the Kyoto Encyclopedia of Genes and Genomes (KEGG) [56] database. The goana and kegga functions evaluate the significance of the overlaps with hypergeometric tests, optionally adjusting for any power trend bias associated with average expression or with gene length [57]. Alternatively, differential analyses can be conducted directly for pre-defined gene sets such as the Molecular Signatures Database collections [58] or gene sets from previous independent studies. The roast, mroast and fry functions perform self-contained tests to assess whether the majority of the genes in a set are differentially expressed across the comparison of interest [59]. The camera function performs a competitive gene set test to find gene sets that are highly ranked relative to other genes in terms of differential expression [60]. The romer function conducts a gene set enrichment analysis analogous to the GSEA approach [61]. The functions all achieve rigorous FDR control even in the presence of inter-gene correlations [59, 60].

edgeR v4 includes S3 methods for all the above gene set functions. The goana and kegga methods operate on an DGEExact or DGELRT object produced by exactTest, glmLRT or glmQLFTest. They automatically extract differentially expressed genes under a given significance threshold and perform generalized hypergeometric tests for enrichment of GO terms or KEGG pathways in the list of differentially expressed genes.

edgeR's roast, mroast, fry, camera and romer methods operate on DGEList data objects and use a z-score strategy to translate negative binomial counts into normal deviates suitable for limma analyses. The functions fit NB GLMs using the null hypothesis design matrix to the count data and convert the counts to their z-score equivalents using a quantile-quantile transformation with the estimated means and dispersions. edgeR supports integer or non-integer data in computing the z-score equivalents. For integer values, a continuity correction is applied by splitting the probability mass of each integer in two. It computes the mid-p tail probability of the given quantile and then converts it to the standard normal deviate with the same cumulative probability distribution value [62]. Non-integer values are handled by interpolation. In the edgeR context, the fry function is always preferred over roast or mroast. In the limma context, fry is an analytic approximation to mroast that is faster and yields higher resolution p-values. In edgeR, the z-score transformation ensures that the distributional assumptions of fry are satisfied exactly.

Transcript-level differential expression.

edgeR versions v1-v3 were designed for gene-level analyses of RNA-seq, but edgeR v4 adds the ability to conduct differential expression analyses for individual transcripts (isoforms) of each gene using output from either Salmon [45] or kallisto [44]. Transcript quantifications are inherently more uncertain than gene-level read counts because of ambiguous assignment of RNA fragments to isoforms [28]. Whereas sequence reads can usually be assigned unambiguously to a gene, reads are very often compatible with multiple transcripts for that gene, particularly for genes with many isoforms. This read to transcript ambiguity (RTA) causes technical overdispersion in the read counts and the amount of overdispersion depends on the degree of overlap of each transcript with other transcripts rather than on its expression. The edgeR gene-level differential expression pipeline does not perform optimally on transcript counts because the RTA disrupts the mean-variance relationship normally observed for gene level RNA-seq data and therefore interferes with the efficiency of the empirical Bayes dispersion estimation procedures. However, the RTA dispersion can be modelled elegantly using overdispersed Poisson distributions in accordance with edgeR's QL model.

Salmon and kallisto have the ability to repeatedly resample each RNA-seq library to generate bootstrap samples, and this resampling provides a means to estimate technical variability. Alternatively, Salmon can generate Gibbs samples from the posterior distribution, which can be used in the same way as bootstrap samples. The edgeR functions catchSalmon or catchKallisto read the Salmon or kallisto outputs and use the bootstrap or Gibbs samples to estimate a QL-dispersion arising from RTA for each transcript. The QL-dispersions can then be divided out of the transcript counts, as described in Methods, leading to divided counts that can be analysed by edgeR's gene-level software tools with full statistical efficiency [23]. The QL-dispersions estimate the variance-inflation induced by RTA and scale down the transcript counts so that the resulting library sizes reflect their true precision. The divided counts follow the traditional NB mean-variance relationship so that standard methods designed for gene-level differential expression analyses can be applied without further modification. The divided counts are not integers but are handled by edgeR's continuous generalization of the NB distribution.

The divided transcript counts can also be input to edgeR's diffSpliceDGE function for a differential transcript usage analysis, as described in the next section.

Differential exon usage.

RNA-Seq has been proven to be a very powerful tool in studying alternative splicing where exons are differentially combined or skipped, resulting in multiple protein isoforms encoded by a single gene. One way to detect alternative splicing events is to test for differential exon usage (DEU), which allows fast detection of potentially differentially spliced genes without identifying the actual isoforms present in different groups.

In edgeR, DEU analysis can be performed using the diffSpliceDGE function. It takes an input of read counts at the exon level and compares the change of the expression level of each exon to the change of the expression level of the gene containing that exon under a certain comparison. NB GLMs are fitted to exon counts and exon-level NB-dispersions are estimated, as would be done for a gene-level analysis but at the exon level.

Depending on the pipeline of choice, statistical tests can be performed using either likelihood ratio tests or quasi-likelihood F-tests to obtain a test statistics and a p-value for each exon. The diffSpliceDGE function offers two different ways to provide inferences at the gene level. The first approach is to combine the exon level test statistics for all the exons within a gene and then to conduct a gene level test. The second approach is to combine the exon level p-values using Simes' method for all the exons within a gene to get a p-value for that gene [63]. The first method favours genes in which a number of exons are differentially spliced. The Simes' method, on the other hand, is likely to be more powerful for picking up genes of which only a minority of the exons are differentially spliced.

shRNA-seq screens.

edgeR can be used to analyse data from CRISPR-Cas9 and shRNA-seq genetic screens to identify the change of single guide RNA (sgRNA) or short hairpin RNA (shRNA) in the selected cell population relative to a control population [64]. Given a list of sample index sequences and sgRNA- or shRNA-specific sequences from an amplicon sequencing screen, the edgeR processAmplicons function obtains read counts for each sgRNA/shRNA in the screen across all samples. This is done by counting the number of times each sample index and sgRNA/shRNA combination could be matched in reads from the input fastq files. These read counts are then organized and stored in a DGEList object for a standard edgeR down-stream DE analysis, which identifies a list of sgRNAs/shRNAs that are differentially expressed between the groups. To interpret the results at gene level, multiple sgRNAs/shRNAs that target the same gene are grouped together and treated as a 'set'. Gene set testing methods such as camera and fry can then be applied to summarize the results at gene level [59, 60].

Differential methylation analysis.

Bisulfite sequencing (BS-seq) has become the gold-standard technology for studying DNA methylation [65]. A number of software tools, such as Bismark, have been developed to align BS-seq reads to a reference genome and count the number of C-to-T conversions at each CpG locus in each sample [50]. Typical downstream analyses often involve identifying differentially methylated CpG loci or regions between different experimental conditions.

Since the output count matrix from methylation calling software contains both methylated and unmethylated Cytosines at each CpG locus, the structure of BS-seq data is analogous to paired-samples RNA-seq data. This allows the differential methylation analysis to be conducted using existing edgeR pipelines developed for RNA-seq differential expression analyses [13]. In particular, edgeR models both methylated and unmethylated counts as NB distributed, and the variation between replicate samples is captured by the NB dispersion parameter. One key difference between BS-seq and paired-samples RNA-seq analysis is that the pair of libraries that hold the methylated and unmethylated read counts from each sample are treated as a unit, and hence share the same library size. A special design matrix constructed by the modelMatrixMeth function in edgeR includes individual sample effects accounting for read coverage plus group-specific coefficients representing the log-ratio of methylated to unmethylated reads for each group. The subsequent analysis is identical to any other edgeR analysis giving methylation analysts access to the full downstream capabilities of the edgeR package.

Likelihood ratio or quasi-F tests can then be performed to identify CpG loci at which the proportion of methylated reads are significantly different between the groups [13]. Given the fact that unmethylated CpGs are often enriched in gene promoters, a gene oriented differential methylation analysis can be conducted by aggregating the methylated and unmethylated CpG counts in gene promoter regions. This improves the statistical power of differential methylation analysis and provides an interpretation at gene level.

The edgeR methylation approach was originally developed for reduced representation BS-seq but has been extended to whole-genome BS-seq and shown to outperform competing methods [66].

The same edgeR analysis strategy as for BS-seq can be applied to any type of sequencing application where the sequence reads are classified into two classes at each genomic loci and the aim is to test for changes in the relative proportions of the two classes. In unpublished work by the authors, for example, this approach has been used to test for changes in haplotype-specific expression.

Single-cell pseudo-bulk analysis.

Single-cell technology is revolutionizing the field of biomedical research. A number of R packages and software tools have been developed for analysing scRNA-seq data in the past few years [67, 68, 48]. Now as the technology evolves and the cost reduces, people are able to perform single cell experiments with biological replicates. Accounting for biological variation between replicates is crucial for single-cell differential expression analyses so that the results are not driven by particular samples.

One popular approach to solve this problem is the pseudo-bulk method, where read counts of the cells within the same cluster and from the same sample are aggregated together to form pseudo-bulk samples [49]. The edgeR differential expression analysis pipeline can be applied to pseudo-bulk samples for identifying marker genes of each cell cluster.

The combination of pseudo-bulk and edgeR pipeline has been examined in a simulation study, and it gives the best performance compared with other existing methods for single-cell differential expression analysis in the presence of replicate samples [49]. This combination was also applied to a recent large cohort single cell study, and it successfully identified marker genes of different cell populations while accounting for the biological variation between tissue specimens [69].

Graphic exploration

MDS plot.

edgeR offers diagnostic plots for data exploration (Figure 2). One of the most commonly used plots is the multi-dimensional scaling (MDS) plot, which visualizes the similarity between all the samples on a two-dimensional scatter plot (Figure 2a). The top (500 by default) genes with largest standard deviations across the samples are used for computing the principal coordinates. The distances on the MDS plot represent the leading log2-fold-changes, which are defined as the root-mean-square average of the top largest log2-fold-changes between each pair of samples. The percentage variation explained is also calculated and returned for the selected dimensions. MDS plot is an unsupervised visualization method that provides a easy and fast way to check the relationship of all the samples in a data set.

limma's removeBatchEffect can be used in conjunction with an MDS plot to display treatment or group effects after adjusting for batch effects or covariates.

Dispersion plots.

edgeR models the read counts using NB distribution for each gene across all the samples. Dispersion parameter of the NB distribution can be interpreted as the variability between biological replicates [7]. In a standard analysis pipeline, edgeR provides three different types of NB dispersion estimates for each gene. The first one is a common dispersion, which is a global dispersion estimate for all genes. The second one is a trended dispersion where the dispersion is predicted from the abundance of a gene. The last one is a gene-specific dispersion for each individual gene estimated by the empirical Bayes information sharing strategy. These dispersion estimates can be visualized in a BCV plot generated by the plotBCV function in edgeR (Figure 2b). The y-axis of the BCV plot displays square-root NB dispersion, which can be interpreted as biological coefficient of variation (BCV).

Under the QL framework, the gene-specific variability in addition to the overall biological variability is measured by the QL dispersion. The estimates of QL dispersion can be visualized in a QL-dispersion plot generated by the plotQLDisp function in edgeR (Figure 2c). It displays a global QL mean-dispersion trend, the raw QL dispersion estimates, as well as the squeezed estimates after the empirical Bayes moderation.

MD plot.

A mean-difference plot (MD plot) is a very useful tool for assessing the differences between samples or groups. To examine each individual sample in a data set, an MD plot can be constructed by comparing the log-expression values of all the genes in that sample with the mean of those in all other samples. The skewness of the log-ratios on an MD plot would suggest that a normalization is required to correct for the compositional biases. MD plots can also be useful tools to spot biases that are typically observed in DNA-based sequencing experiments, such as immunoprecipitation efficiency biases and also more complex trended biases for which the magnitude of the systematic differences between samples will change with the average abundance [11]. When a differential expression analysis is performed, an MD plot can be generated to show the relationship between the log fold changes and the average log-CPM values for all the genes under that comparison (Figure 2d). This allows a quick identification of outlier genes that change substantially under the comparison.

Splicing plot.

edgeR detects alternative splicing events by testing for differential exon usage (DEU) using the diffSpliceDGE function. The results from diffSpliceDGE can be visualized in a differential splicing plot generated by the plotSpliceDGE function in edgeR (Figure 2e). In a splicing plot, relative log-fold changes by exons for the specified gene are plotted for all the exons within the gene from left to right in the order of their genomic locations. Exons that are significantly differentially used are highlighted in red. This easy and simple visualization approach provides a quick way to examine the differences of gene structure between the groups.

Computational efficiency

C++ implementation.

edgeR typically fits NB GLMs to tens of thousands of genes in a given dataset. For some applications, the number of genomic features may in the hundreds of thousands, even millions are possible. The vector of GLM parameters needs to be estimated for each individual gene using an iterative computational algorithm. The iteration must reliably converge and provide accurate estimates for every gene. When the NB-dispersions are estimated, the genewise GLMs need to be refitted repeatedly with different candidate values for the dispersions. The efficiency and reliability of the iterative GLM fitting is therefore a core factor in determining the overall computational load of the package.

The classical algorithm used for GLM fitting is an iteratively reweighted least squares iteration that is equivalent to Fisher-scoring [8]. The basic algorithm is adequate for classical univariate applications but is not sufficiently reliable in the edgeR context where even a small percentage of convergence failures would be problematic.

The original GLM version of edgeR was written purely in R. A novel GLM algorithm was implemented using a simplified approximation to the Fisher information matrix and a line search strategy to ensure convergence [7]. To achieve acceptable computational speed, the R implementation was vectorized so that the NB GLMs were fitted for all the genes simultaneously. In edgeR v4, the GLM fitting is implemented in C++. If the experiment design can be transformed to a oneway layout, then unmodified Fisher scoring is used which, in this scenario, is fast and reliable. For other designs, Fisher-scoring is implemented with a Levenberg-Marquardt modification to ensure convergence [70, 71]. The direct C++ implementation increases speed, improves the accuracy of the final estimates, and allows observational weights to be supported.

Other edgeR components rewritten in C++ include the Cox-Reid APL, the GLM deviances, loess curves and maximizing the interpolant as part of the weighted likelihood empirical Bayes.

CompressedMatrix class saves memory.

When fitting the genewise GLMs, the offsets $\log L_{gi}$, weights w_{gi} and NB dispersions ϕ_g all enter into the calculations and are treated as known values. The offsets are often the same for every gene but can be observation-specific. The weights are usually constant $(w_{gi} = 1)$ but could be sample-specific or observation-specific. The NB-dispersions are typically gene-specific but might be constant and could in principle be observation-specific. In other words, each of these quantities could be a constant, a row vector, a column vector, or a matrix. It is convenient to represent each of these quantities as a gene×sample matrix in R, but doing so is memory inefficient if the same numerical values are repeated over genes or samples or both.

In edgeR v4, a simple but elegant object class called CompressedMatrix has been introduced to make these matrices more efficient. A CompressMatrix object is subsettable in R as a matrix but, internally, stores only the minimum information required to construct the matrix from the repeat structure. The minimum information can be a constant, a row vector, a column vector, or a complete matrix. This simple but efficient representation saves considerable memory for most analyses when the offsets, weights and NB-dispersion are passed to C++ or stored in the fitted model DGEGLM object.

User interface

Object-oriented programming.

An edgeR analysis consists of a number of distinct steps (Figure 1). Each of the major steps in the pipeline — (1) data import, (2) model fitting, (3) statistical testing and (4) gene ranking — produces a classed R object that can be input to functions downstream in the pipeline.

Data import produces an object of class ‘DGEList’, which stores the read counts and associated information (Figure 3a). The essential components of a ‘DGEList’ object are the matrix of raw counts and a data-frame containing the sample information. Other optional components include gene annotation and a design matrix.

Normalization adds library-size normalization factors or offsets to the DGEList object and dispersion estimation adds NB dispersion estimates.

Model fitting by glmFit or glmQLFit produces an object of class ‘DGEGLM’, which stores the generalized linear model fits and dispersion estimates (Figure 3b). The DGEGLM object includes GLM deviances, coefficient estimates, and fitted values, plus all the parameters used in the fitting process. The object produced by glmQLFit contains all the same information as glmFit plus addition components storing the QL-dispersion estimation.

Testing for differential abundance produces either a 'DGEExact' object if exact NB tests were used or a 'DGELRT' object if GLMs were used (Figure 3b). The object includes a data-frame table containing the genewise log-fold changes, average log-CPMs, test statistics and p-values for the comparison specified. The 'DGELRT' output object preserves most of the components of the input 'DGEGLM' object but drops the read counts if they were present.

Finally, a 'TopTags' object containing a ranked gene list is created by the topTags function. The 'TopTags' object contains a sorted data-frame of genewise tests results including adjusted p-values that adjust for multiple testing. The default adjustment method controls the FDRs for a ranked list using the Benjamini and Hochberg method [72].

All of these data classes obey many analogies with matrices. For 'DGEList', rows correspond to genes/features and columns to different samples. For 'DGEGLM', rows correspond to genes/features and columns correspond to linear model coefficients. For 'DGEExact', 'DGELRT' and 'TopTags', the columns correspond to the differential results table. The standard R functions summary, dim, length, ncol, nrow, dimnames, rownames, colnames have methods for each of these classes. DGEList objects can be subsetted by rows and columns as for a matrix. Multiple DGEList objects can be row-binded or column-binded together, again as for matrices. The other edgeR classes can be subsetted by rows, but column subsetting is disallowed because it would not produce a valid object of that class.

All edgeR objects can be coerced to a data-frame using as.data.frame in R. All edgeR classes except TopTags belong to the virtual class 'LargeDataObject' for which a show method is defined to display the leading rows of each component vector, matrix or data.frame.

A design aim of edgeR is that users should be able to easily explore and manipulate all objects produced by edgeR functions and that doing so should require a knowledge of base R only. While all the edgeR objects are formally registered using R's S4 system, the objects are also fully manipulatable as ordinary R lists and the list components are standard base R objects: data-frames, vectors and matrices. This is analogous to the behaviour of core R modelling functions such as lm or glm. At each stage of the edgeR pipeline, functions have S3 object-orientated methods that accept objects from upstream in the pipeline, but also have default methods that accept input in the form of atomic R objects. This allows direct programming access to any stage of the edgeR pipeline from external software tools.

In edgeR v4, support has been added for the SummarizedExperiment object class developed by the Bioconductor core team (`https://bioconductor.org/packages/SummarizedExperiment`) [18]. All relevant edgeR functions now accept raw data in a SummarizedExperiment container that includes a 'counts' assay, making it simpler to access edgeR's functionality in the context of other Bioconductor pipelines using SummarizedExperiment.

Documentation.

The edgeR package is supported by extensive documentation, examples and online help. The package documents 248 exported functions or methods, each of which has a detailed documentation help page including the information of distinct methods for generic functions and various examples. Every help page formally documents the class of each input argument and the class and structure of the output object. edgeR 4.0.0 includes a 139-page User's Guide. It covers a wide range of topics such as the underlying statistical framework of edgeR, normalization, different analysis pipelines, setting up appropriate design matrices and downstream analysis. The edgeR User's Guide also provides 10 individual case studies with complete data source and analysis R code. These fully worked case studies cover different types of analyses that edgeR is capable of, including differential gene expression, differential exon usage and time course analysis of RNA-seq data, as well as differential analysis of data from BS-seq, CRISPR-Cas9 and shRNA-seq genetic screens. Any further questions related to edgeR analysis can be posted to the Bioconductor support site (https://support.bioconductor.org) and will be answered either by the authors or by other Bioconductor community members.

Discussion

The edgeR package has been one of most widely used software tools for the statistical analysis of sequencing read counts over the past 15 years. The package develops and implements advanced statistical methods associated with generalized linear models, conditional likelihood and empirical Bayes for the analysis of count data. edgeR is used as an integrated analysis environment in its own right and is also used as an underlying engine by other packages that analyse specific technologies. At the time of writing (January 2024), 186 downstream Bioconductor packages depend on or suggest edgeR.

This article has summarized the design and capabilities of the edgeR package and has also described the history of the edgeR package over time. edgeR 4.0 introduces further new statistical ideas, improves computational efficiency and extends the range of applications of the package. Statistical innovations include modelling of fractional counts, a more refined modelling of the GLM deviances to achieve more accurate QL-dispersion estimation in small count scenarios and the idea of divided counts to extract out the overdispersion arising from transcript quantification. Implementation of low-level functions in C++ allows edgeR to handle larger datasets more efficiently. New data analyses include transcript-level differential expression, differential exon usage, differential transcript usage, differential methylation analysis, pseudo-bulk analysis of single-cell RNA-seq and hypothesis tests relative to a fold-change threshold. edgeR 4.0 also includes direct support for pathway analysis and gene set enrichment analysis.

Code availability

edgeR software is freely available from `https://bioconductor.org/packages/edgeR`. Instructions for installing edgeR and other Bioconductor packages are given at `https://bioconductor.org/install`. The package source code can be git cloned from `git@git.bioconductor.org:packages/edgeR`. Additional documentation and datasets used in the edgeR User's Guide are available from `https://bioinf.wehi.edu.au/edgeR`.

Data availability

The example datasets shown in this article are publicly available as described in Materials and Methods. The TCGA data is available from `https://tcga-data.nci.nih.gov`. The RNA-seq data analysed by Chen *et al.* [10] and Fu *et al.* [33] is available as GEO series GSE60450. The RNA-seq data analysed by Fu *et al.* [34] is available as GEO series GSE118617.

Acknowledgements

The authors thank all the colleagues who have been actively involved in the edgeR project over the years, including Mark Robinson, Davis McCarthy, Belinda Phipson, Matthew Ritchie, Zhiyin Dai, Oliver Voogd, Yifang Hu, Xiaobei Zhou, Jinming Cheng and Mengbo Li. The authors also appreciate all the suggestions and reported bugs from the Bioconductor community as well as all the edgeR users around the world.

Funding

This work was supported by the Chan Zuckerberg Initiative (grants 2019-207283 and 2021-237445), by the Australian National Health and Medical Research Council (Fellowship 1058892 to GS, Investigator Grant 2025645 to GS, IRIISS to WEHI), by the Australian Medical Research Future Fund

(Investigator Grant 1176199 to YC), and by Victorian State Government Operational Infrastructure Support.

References

[1] Wang,Z., Gerstein,M. and Snyder,M. (2009) RNA-Seq: a revolutionary tool for transcriptomics. *Nature Reviews Genetics*, **10**, 57–63.

[2] Van den Berge,K., Hembach,K.M., Soneson,C., Tiberi,S., Clement,L., Love,M.I., Patro,R. and Robinson,M.D. (2019) RNA sequencing data: hitchhiker's guide to expression analysis. *Annual Review of Biomedical Data Science*, **2**, 139–173.

[3] Cullum,R., Alder,O. and Hoodless,P.A. (2011) The next generation: using new sequencing technologies to analyse gene regulation. *Respirology*, **16**, 210–222.

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

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

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

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

[8] Dunn,P.K. and Smyth,G.K. (2018) *Generalized Linear Models With Examples in R*. Springer-Verlag, New York.

[9] Chen,Y., Lun,A.T.L. and Smyth,G.K. (2014) Differential expression analysis of complex RNA-seq experiments using edgeR. In Datta,S. and Nettleton,D.S. (eds.), *Statistical Analysis of Next Generation Sequence Data*, Springer, New York, pp. 51–74.

[10] Chen,Y., Lun,A.T.L. and Smyth,G.K. (2016) From reads to genes to pathways: differential expression analysis of RNA-seq experiments using Rsubread and the edgeR quasi-likelihood pipeline. *F1000Research*, **5**, 1438.

[11] Lun,A.T.L. and Smyth,G.K. (2016) csaw: a bioconductor package for differential binding analysis of ChIP-seq data using sliding windows. *Nucleic Acids Research*, **44**, e45.

[12] Lun,A.T.L. and Smyth,G.K. (2015) diffHic: a bioconductor package to detect differential genomic interactions in Hi-C data. *BMC Bioinformatics*, **16**, 258.

[13] Chen,Y., Pal,B., Visvader,J.E. and Smyth,G.K. (2017) Differential methylation analysis of reduced representation bisulfite sequencing experiments using edgeR. *F1000Research*, **6**, 2055.

[14] Shapiro,J.P., Biswas,S., Merchant,A.S., Satoskar,A., Taslim,C., Lin,S., Rovin,B.H., Sen,C.K., Roy,S. and Freitas,M.A. (2012) A quantitative proteomic workflow for characterization of frozen clinical biopsies: laser capture microdissection coupled with label-free mass spectrometry. *Journal of Proteomics*, **77**, 433–440.

[15] Branson,O.E. and Freitas,M.A. (2016) Tag-count analysis of large-scale proteomic data. *Journal of Proteome Research*, **15**, 4742–4746.

[16] Lin,M.H., Wu,P.S., Wong,T.H., Lin,I.Y., Lin,J., Cox,J. and Yu,S.H. (2022) Benchmarking differential expression, imputation and quantification methods for proteomics data. *Briefings in Bioinformatics*, **23**, bbac138.

[17] Ritchie,M.E., Phipson,B., Wu,D., Hu,Y., Law,C.W., Shi,W. and Smyth,G.K. (2015) limma powers differential expression analyses for RNA-sequencing and microarray studies. *Nucleic Acids Research*, **43**, e47.

[18] Gentleman,R.C., Carey,V.J., Bates,D.M., Bolstad,B., Dettling,M., Dudoit,S., Ellis,B., Gautier,L., Ge,Y., Gentry,J. *et al.* (2004) Bioconductor: open software development for computational biology and bioinformatics. *Genome biology*, **5**, 1–16.

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

[20] Lun,A.T.L., Chen,Y. and Smyth,G.K. (2016) It's DE-licious: a recipe for differential expression analyses of RNA-seq experiments using quasi-likelihood methods in edgeR. *Methods in Molecular Biology*, **1418**, 391–416.

[21] Lun,A.T. and Smyth,G.K. (2017) No counts, no variance: allowing for loss of degrees of freedom when assessing biological variability from RNA-seq data. *Statistical Applications in Genetics and Molecular Biology*, **16**, 83–93.

[22] Phipson,B., Lee,S., Majewski,I.J., Alexander,W.S. and Smyth,G.K. (2016) Robust hyperparameter estimation protects against hypervariable genes and improves power to detect differential expression. *Annals of Applied Statistics*, **10**, 946–963.

[23] Baldoni,P.L., Chen,Y., Hediyeh-zadeh,S., Liao,Y., Dong,X., Rithie,M.E., Shi,W. and Smyth,G.K. (2023) Dividing out quantification uncertainty allows efficient assessment of differential transcript expression with edgeR. *Nucleic Acids Research*, **Dec 7**, doi: 10.1093/nar/gkad1167.

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

[25] Risso,D., Schwartz,K., Sherlock,G. and Dudoit,S. (2011) GC-content normalization for RNA-Seq data. *BMC Bioinformatics*, **12**, 480.

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

[27] Risso,D., Ngai,J., Speed,T.P. and Dudoit,S. (2014) Normalization of RNA-seq data using factor analysis of control genes or samples. *Nature Biotechnology*, **32**, 896–902.

[28] Soneson,C., Love,M.I. and Robinson,M.D. (2015) Differential analyses for RNA-seq: transcript-level estimates improve gene-level inferences. *F1000Research*, **4**, 1521.

[29] Zhou,X., Lindsay,H. and Robinson,M.D. (2014) Robustly detecting differential expression in RNA sequencing data using observation weights. *Nucleic Acids Research*, **42**, e91.

[30] Koboldt,D., Fulton,R., McLellan,M., Schmidt,H., Kalicki-Veizer,J., McMichael,J., Fulton,L., Dooling,D., Ding,L., Mardis,E. *et al.* (2012) Comprehensive molecular portraits of human breast tumours. *Nature*, **490**, 61–70.

[31] Parker,J.S., Mullins,M., Cheang,M.C., Leung,S., Voduc,D., Vickery,T., Davies,S., Fauron,C., He,X., Hu,Z. *et al.* (2009) Supervised risk predictor of breast cancer based on intrinsic subtypes. *Journal of clinical oncology*, **27**, 1160.

[32] Nolan,E., Vaillant,F., Branstetter,D., Pal,B., Giner,G., Whitehead,L., Lok,S., Mann,G., Thorne,H., Rohrbach,K. *et al.* (2016) RANK ligand as a potential target for breast cancer prevention in BRCA1-mutation carriers. *Nature Medicine*, pp. 933–939.

[33] Fu,N.Y., Rios,A.C., Pal,B., Soetanto,R., Lun,A.T.L., Liu,K., Beck,T., Best,S.A., Vaillant,F., Bouillet,P. *et al.* (2015) EGF-mediated induction of Mcl-1 at the switch to lactation is essential for alveolar cell survival. *Nature Cell Biology*, **17**, 365–375.

[34] Fu,N.Y., Pal,B., Chen,Y., Jackling,F., Milevskiy,M., Vaillant,F., Capaldo,B., Guo,F., Liu,K.H., Rios,A.C. *et al.* (2018) Foxp1 is indispensable for ductal morphogenesis and controls the exit of mammary stem cells from quiescence. *Developmental Cell*, **47**, 629–644.

[35] Cheng,J., Smyth,G.K. and Chen,Y. (2023) Unraveling the timeline of gene expression: A pseudotemporal trajectory analysis of single-cell RNA sequencing data. *F1000Research*, **12**, 684.

[36] Smyth,G.K. and Verbyla,A. (1996) A conditional likelihood approach to residual maximum likelihood estimation in generalized linear models. *Journal of the Royal Statistical Society. Series B (Methodological)*, **58**, 565–572.

[37] Cox,D.R. and Reid,N. (1987) Parameter orthogonality and approximate conditional inference. *Journal of the Royal Statistical Society: Series B (Methodological)*, **49**, 1–18.

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

[39] Chen,Y. (2013) *Differential expression analysis of complex RNA-seq experiments*. Ph.D. thesis, Department of Medical Biology, University of Melbourne. http://hdl.handle.net/11343/38622.

[40] Burden,C.J., Qureshi,S.E. and Wilson,S.R. (2014) Error estimates for the analysis of differential expression from RNA-seq count data. *PeerJ*, **2**, e576.

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

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

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

[44] Bray,N.L., Pimentel,H., Melsted,P. and Pachter,L. (2016) Near-optimal probabilistic RNA-seq quantification. *Nature biotechnology*, **34**, 525.

[45] Patro,R., Duggal,G., Love,M.I., Irizarry,R.A. and Kingsford,C. (2017) Salmon provides fast and bias-aware quantification of transcript expression. *Nature Methods*, **14**, 417.

[46] Liao,Y., Smyth,G.K. and Shi,W. (2019) The R package Rsubread is easier, faster, cheaper and better for alignment and quantification of RNA sequencing reads. *Nucleic Acids Research*, **47**, e47.

[47] Zheng,G.X., Terry,J.M., Belgrader,P., Ryvkin,P., Bent,Z.W., Wilson,R., Ziraldo,S.B., Wheeler,T.D., McDermott,G.P., Zhu,J. *et al.* (2017) Massively parallel digital transcriptional profiling of single cells. *Nature Communications*, **8**, 14049.

[48] Hao,Y., Stuart,T., Kowalski,M.H., Choudhary,S., Hoffman,P., Hartman,A., Srivastava,A., Molla,G., Madad,S., Fernandez-Granda,C. *et al.* (2023) Dictionary learning for integrative, multimodal and scalable single-cell analysis. *Nature Biotechnology*, pp. 1–12.

[49] Crowell,H.L., Soneson,C., Germain,P.L., Calini,D., Collin,L., Raposo,C., Malhotra,D. and Robinson,M.D. (2020) *muscat* detects subpopulation-specific state transitions from multi-sample multi-condition single-cell transcriptomics data. *Nature Communications*, **11**, 6077.

[50] Krueger,F. and Andrews,S.R. (2011) Bismark: a flexible aligner and methylation caller for Bisulfite-Seq applications. *Bioinformatics*, **27**, 1571–1572.

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

[52] Law,C.W., Zeglinski,K., Dong,X., Alhamdoosh,M., Smyth,G.K. and Ritchie,M.E. (2020) A guide to creating design matrices for gene expression experiments. *F1000Research*, **9**.

[53] Phipson,B. (2013) *Empirical bayes modelling of expression profiles and their associations*. Ph.D. thesis, Department of Mathematics and Statistics, The University of Melbourne. http://hdl.handle.net/11343/38162.

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

[55] 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.

[56] Kanehisa,M. and Goto,S. (2000) KEGG: Kyoto Encyclopedia of Genes and Genomes. *Nucleic Acids Research*, **28**, 27–30.

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

[58] Liberzon,A., Subramanian,A., Pinchback,R., Thorvaldsdóttir,H., Tamayo,P. and Mesirov,J.P. (2011) Molecular signatures database (MSigDB) 3.0. *Bioinformatics*, **27**, 1739–1740.

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

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

[61] 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. *et al.* (2005) Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proceedings of the National Academy of Sciences of the United States of America*, **102**, 15545–15550.

[62] Dunn,P.K. and Smyth,G.K. (1996) Randomized quantile residuals. *Journal of Computational and Graphical Statistics*, **5**, 236–244.

[63] Simes,R.J. (1986) An improved Bonferroni procedure for multiple tests of significance. *Biometrika*, **73**, 751–754.

[64] Dai,Z., Sheridan,J.M., Gearing,L.J., Moore,D.L., Su,S., Wormald,S., Wilcox,S., O'Connor,L., Dickins,R.A., Blewitt,M.E. *et al.* (2014) edgeR: a versatile tool for the analysis of shRNA-seq and CRISPR-Cas9 genetic screens. *F1000Research*, **3**, 95.

[65] Laird,P.W. (2010) Principles and challenges of genome-wide DNA methylation analysis. *Nature Reviews Genetics*, **11**, 191–203.

[66] Peters,T.J., Buckley,M.J., Chen,Y., Smyth,G.K., Goodnow,C.C. and Clark,S.J. (2021) Calling differentially methylated regions from whole genome bisulphite sequencing with DMRcate. *Nucleic Acids Research*, **49**, e109–e109.

[67] Lun,A.T.L., McCarthy,D.J. and Marioni,J.C. (2016) A step-by-step workflow for low-level analysis of single-cell RNA-seq data with Bioconductor. *F1000Research*, **5**.

[68] McCarthy,D.J., Campbell,K.R., Lun,A.T. and Wills,Q.F. (2017) Scater: pre-processing, quality control, normalization and visualization of single-cell RNA-seq data in R. *Bioinformatics*, **33**, 1179–1186.

[69] Pal,B., Chen,Y., Vaillant,F., Capaldo,B.D., Joyce,R., Song,X., Bryant,V., Penington,J.S., Di-Stefano,L., Ribera,N.T. *et al.* (2021) A single-cell RNA atlas of human breast spanning normal, preneoplastic and tumorigenic states. *EMBO Journal*, **40**, e107333.

[70] Osborne,M.R. (1992) Fisher's method of scoring. *International Statistical Review*, **60**, 99–117.

[71] Smyth,G.K. (2005) Optimization and nonlinear equations. *Encyclopedia of Biostatistics*, pp. 3174–3180.

[72] 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.

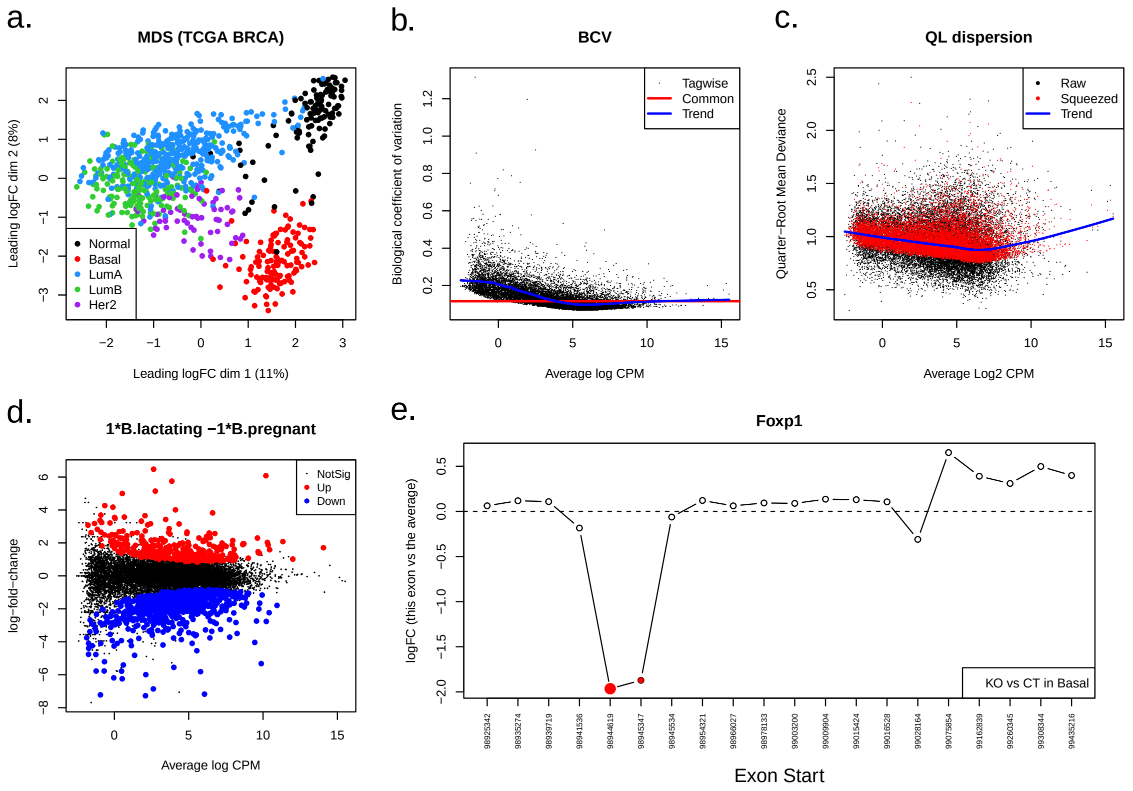


Figure 2: Example diagnostic plots produced by edgeR. **(a)** Multidimensional scaling (MDS) plot of the human TCGA breast cancer data set of 882 individual RNA-seq samples, generated by the plotMDS function in edgeR. Samples are coloured by breast cancer subtype identified by PAM50 signatures. Samples of the same cancer subtype are clustered together on the MDS plot. The proportions of variance explained by the first two dimensions are also shown. **(b)** Scatter plot of the biological coefficient of variation (BCV) against the average abundance of each gene in the mouse mammary gland RNA-seq data set, generated by the plotBCV function in edgeR. The plot shows the square-root estimates of the common, trended and tagwise NB dispersions. **(c)** Scatter plot of the quarter-root QL dispersion against the average abundance of each gene in the mouse mammary gland RNA-seq data set, generated by the plotQLDisp function in edgeR. The “Raw” and “Squeezed” values represent the dispersion estimates before and after the empirical Bayes moderation towards the trend. **(d)** MD plot showing the log-fold change and average abundance of each gene between the lactating and pregnant samples in the basal cell population, generated by the plotMD function in edgeR. Significantly up and down DE genes are highlighted in red and blue, respectively. **(e)** Plot showing relative log-fold changes by exons for the *Foxp1* gene in an RNA-seq experiment where two of the exons were silenced in the KO group. The plot is generated by the plotSpliceDGE function in edgeR. The relative logFC is the difference between the exon’s logFC and the overall logFC for the gene, as computed by the diffSpliceDGE function in edgeR. The significant differentially used exons are highlighted in red. The size of the red dots are weighted by its significance. The start position of each exon is labelled at the bottom.

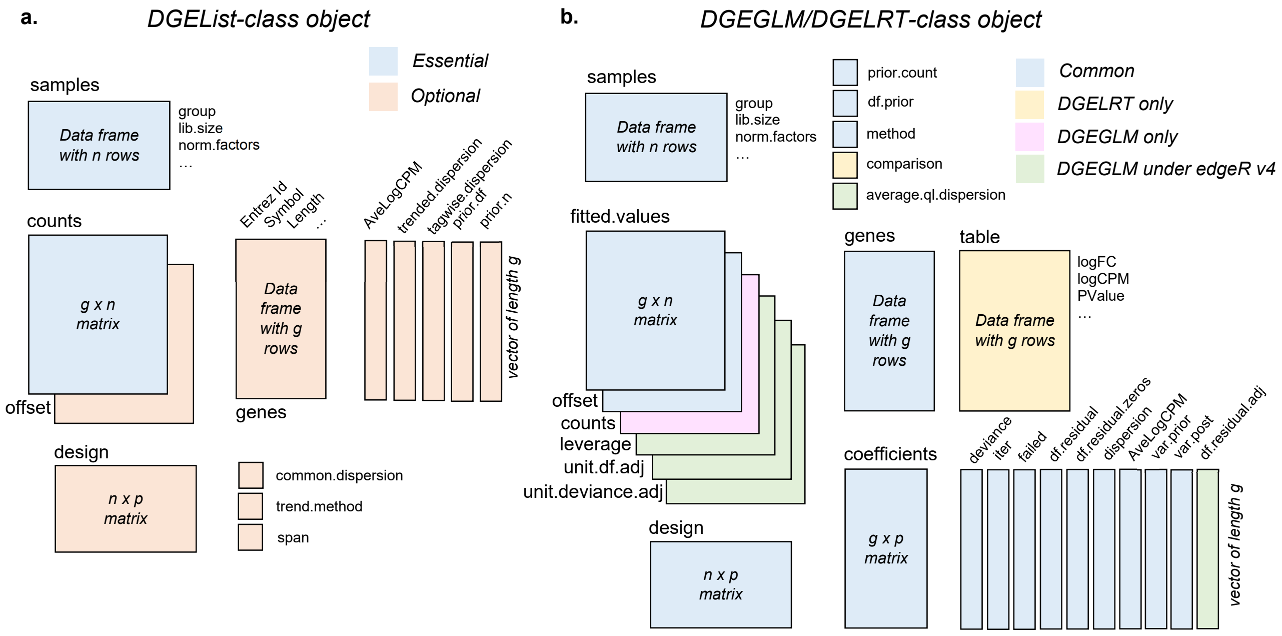


Figure 3: The data structure of **(a)** DGEList-class object, and **(b)** DGEGLM-class and DGELRT-class object.