DTU tools literature

Below we provide a brief description of each of the DTU methods from the literature that were included in the performance benchmarks of this paper. For further details, we refer to the respective original publications. Note that all methods were run in R version 3.6.1 using their respective default settings.

DEXSeq

DEXSeq[1] (R package version 1.32.0) takes as input a transcript-level expression matrix Y_{ti}, with T transcripts (rows) and n samples or cells (columns). Next, a matrix of complementary counts C_{ti} is calculated, which defines how many reads map to any of the other transcripts of the same gene as respective transcript t in cell i. DEXSeq then augments the original expression matrix Y_{ti} by concatenating it with the complementary counts C_{ti}, hence doubling the number of columns of the original count matrix. A negative binomial generalized linear model (GLM) is fitted to each transcript in the augmented count matrix as follows

$$\begin{cases} \{Y_{ti}, C_{ti}\} \sim NB(\mu_{ti}, \phi_t) \\ \log(\mu_{ti}) \sim \eta_{ti} \\ \eta_{ti} \sim \boldsymbol{X}_i^T \boldsymbol{\beta}_t \, . \end{cases}$$

In the specification of the GLM, $\boldsymbol{X}_i^T$ corresponds to row i of design matrix **X**, which defines a covariate pattern that (i) links the transcript-level count matrix to the complementary counts through sample-level intercepts, and (ii) specifies the design of the experiment. Inference on DTU is obtained by testing an interaction effect that assesses if the log fold change between transcript t and all other transcripts in its corresponding gene changes between the conditions of interest (e.g. treatment) with a likelihood ratio test. It is important to note that the estimation of sample-level intercepts is required because of the concatenation of the two count matrices. As a consequence, DEXSeq scales quadratically with the number of samples or cells in the data. The lack of scalability is thus inherent to the parametrization of DEXSeq, putting a severe burden on the utility of DEXSeq for DTU analysis in large datasets, as displayed in Figure 1.

DoubleExpSeq

DoubleExpSeq[2] (R package version 1.1) assumes a double binomial distribution for each transcript. The double binomial distribution is a member of the double exponential family of distributions described by Efron[3], which are extensions of one-parameter exponential family distributions that allow for a more flexible variance structure through introduction of an additional dispersion parameter. DoubleExpSeq adopts a bespoke empirical Bayes procedure for computing shrinkage estimates of the dispersion parameter of the double binomial distribution. The double binomial models the log-odds of drawing a particular transcript t from the pool of transcripts in the corresponding gene g across samples. The intercept thus has an interpretation of a log-odds and the remaining mean model parameter(s) are log-odds ratios, which may thus be interpreted in terms of differential transcript usage. The significance of the mean model parameter(s) are tested using a likelihood ratio test. Importantly, the current implementation of DoubleExpSeq does not allow for modeling multifactorial designs and cannot make use of parallel computing.

DRIMSeq

DRIMSeq[4] (R package version 1.14.0) assumes that the transcript-level expression counts marginally follow a Dirichlet multinomial distribution (DM), where the Dirichlet conjugate prior is used to account for overdispersion with respect to the multinomial distribution. The most important consequence of treating transcript expression as a realization of a multinomial distribution, is that the correlations between expression of transcripts derived from the same gene are directly accounted for. In the DRIMSeq framework, the total count for a gene is considered fixed, and the quantity of interest is the change in proportion of each transcript within a gene between groups of samples or cells. More specifically, DRIMSeq uses a likelihood ratio test to determine if the transcript ratios of a gene, which are modelled by the multinomial, are different between conditions of interest.

Limma diffsplice

Limma diffsplice (limma, R package version 3.42.2) is a built-in functionality described in the current user's guide of the limma Bioconductor R package[5]. Limma was originally devised for analyzing microarray data but can also be used for RNA-Seq data with the limma-voom method[6]. Limma-voom fits a linear model to the log-transformed (normalized) transcript-level count matrix, while adjusting for heteroskedasticity via weighted regression, where the observation weights are computed from the observed variance-mean relationship. Limma diffsplice then uses a series of t-tests to assess DTU at the transcript level by comparing the log-fold change in expression of transcript *t* with the average log-fold change in the expression of all transcripts belonging to the same gene as transcript *t*.

EdgeR diffsplice

EdgeR diffsplice (edgeR, R package version 3.28.1) is a built-in functionality described in the vignettes of the edgeR Bioconductor R package, which was last revisited by Chen et al.[7]. The edgeR diffsplice function fits a negative binomial GLM for each transcript and tests for differential transcript usage by comparing the obtained log-fold changes for each respective transcript within a gene with the log-fold change of the entire gene. If the log-fold change for a certain transcript is significantly different from those of the other transcripts in the gene, it is flagged as differentially used. Note that the negative binomial GLMs can be fit using a canonical likelihood-based approach or using a quasi-likelihood. We adopted the likelihood-based approach as it consistently displayed higher performances (data not shown). In this setting, inference is obtained using a likelihood ratio test.

NBSplice

NBSplice[8] (R package version 1.4.0) fits a negative binomial GLM for each transcript in the dataset. In contrast to e.g. DEXSeq, the mean transcript-level expression (i.e. the mean parameter of the negative binomial model) is taken as the product of the mean gene-level expression value and the observed percentual usage of the transcripts within its corresponding gene. The GLM framework of NBSplice is structured such that DTU between groups of interest can be tested using a likelihood ratio test, where the full model contains an isoform-condition interaction term that is omitted in the null model. Note that in our benchmarks the NB GLM estimation procedure of NBSplice fails to converge when there is a large fraction of zero counts in the data. As a consequence, NBSplice was omitted from the performance benchmarks on single-cell data and from the scalability benchmarks, as the latter also make use of single-cell data.

BANDITS

BANDITS[9] (R package version 1.2.3) adopts a Bayesian hierarchical model with a Dirichlet-multinomial to explicitly model the sample-to-sample variability between biological replicates. In addition to the transcript-level count matrix, equivalence class counts are used as input to the BANDITS algorithm. As described by Bray et al.[10], an equivalence class for a (transcriptomics) read is a multi-set of transcripts associated with that read. As such, an equivalence class represents the transcripts from which a read could have originated. BANDITS leverages the information conveyed by the equivalence class counts to model the uncertainty arising from reads mapping to multiple transcripts. In brief, the allocation of reads to transcripts is treated as a latent variable that is sampled jointly with the parameters of the Dirichlet-multinomial; sampling of these parameters is done with a Markov chain Monte Carlo algorithm. As such, BANDITS allows for modeling the mean relative usage of each transcript within its corresponding gene across samples/cells, while accounting for quantification uncertainty. In addition, BANDITS also accounts for differences in transcript length. Finally, BANDITS tests for DTU (at the transcript level) by performing univariate Wald tests.

References

1. Anders, S., Reyes, A. & Huber, W. Detecting differential usage of exons from RNA-seq data. *Genome Res.* **22**, (2012).
2. Ruddy, S., Johnson, M. & Purdom, E. Shrinkage of dispersion parameters in the binomial family, with application to differential exon skipping. *Ann. Appl. Stat.* **10**, 690–725 (2016).
3. Efron, B. Double exponential families and their use in generalized linear regression. *J. Am. Stat. Assoc.* **81**, 709–721 (1986).
4. Nowicka, M. & Robinson, M. D. DRIMSeq: a Dirichlet-multinomial framework for multivariate count outcomes in genomics. *F1000Research* **5**, 1356 (2016).
5. Smyth, G. K. Linear Models and Empirical Bayes Methods for Assessing Differential Expression in Microarray. *Stat. Appl. Genet. Mol. Biol.* **3**, (2006).
6. Law, C. W., Chen, Y., Shi, W. & Smyth, G. K. voom: precision weights unlock linear model analysis tools for RNA-seq read counts. *Genome Biol.* **15**, (2014).
7. Chen, Y., Mccarthy, D., Ritchie, M., Robinson, M. & Smyth, G. K. edgeR: differential expression analysis of digital gene expression data. *User's Guid.* **https://ww**, (2019).
8. Merino, G. A. & Fernandez, E. A. NBSplice: Negative Binomial Models to detect Differential Splicing. R package. (2019).
9. Tiberi, S. & Robinson, M. D. BANDITS: Bayesian differential splicing accounting for sample-to-sample variability and mapping uncertainty. *Genome Biol.* **21**, 1–13 (2020).
10. Bray, N. L., Pimentel, H., Melsted, P. & Pachter, L. Near-optimal probabilistic RNA-seq quantification. *Nat. Biotechnol.* **34**, 525–527 (2016).