Juggling offsets unlocks RNA-seq tools for fast scalable differential usage, aberrant splicing and expression analyses.

Alexandre Segers[1,2], Jeroen Gilis[1,3,4], Mattias Van Heetvelde[2,5], Elfride De Baere[2,5], Lieven Clement[1,4*]

[1*]Department of Applied Mathematics, Computer Science and Statistics, Ghent University, Ghent, Belgium.
[2]Center for Medical Genetics Ghent, Ghent University and Ghent University Hospital, Ghent, Belgium.
[3]Data Mining and Modeling for Biomedicine, VIB Flemish Institute for Biotechnology, Ghent, Belgium.
[4]Bioinformatics Institute Ghent, Ghent University, Ghent, Belgium.
[5]Department of Biomolecular Medicine, Ghent University, Ghent, Belgium.

*Corresponding author(s). E-mail(s): lieven.clement@ugent.be;
Contributing authors: alexandre.segers@ugent.be; jeroen.gilis@ugent.be; mattias.vanheetvelde@ugent.be; elfride.debaere@ugent.be;

Abstract

RNA-sequencing (RNA-seq) is increasingly used to diagnose patients with rare diseases by prioritising genes with aberrant expression and/or splicing. State-of-the-art methods for detecting aberrant expression and splicing, however, are extremely slow. The latter, also discard much information because they only use junction reads to infer aberrant splicing. In this contribution, we show that replacing the offset for library size unlocks conventional bulk RNA-seq workflows for fast and scalable differential usage, aberrant splicing and expression analyses. Our method, saseR, is several orders of magnitude faster than the state-of-the-art methods and dramatically outperforms these in terms of sensitivity and specificity for aberrant splicing, while being on par with these inferring differential usage

and aberrant expression. Finally, our framework is also very flexible and can be used for all applications that involve the analysis of proportions of short- or long RNA-seq read counts.

1 Introduction

An estimated 350 million people worldwide suffer from rare diseases [1, 2]. For these individuals, a diagnostic rate of 15-75% is currently achieved by using whole exome sequencing (WES) and whole genome sequencing (WGS) to identify the underlying pathogenic variants [3–7]. Although most of these are within coding regions, there is increasing evidence that the diagnostic rate can be further improved by discovering variants in intronic regions that mostly lead to aberrant splicing and in other non-coding regions that contribute to impaired transcriptional regulation [8]. Because the effect of these variants is difficult to identify using WGS only [9], WGS is increasingly complemented with RNA-sequencing (RNA-seq) to improve the diagnostic rate by identifying aberrant expression, missplicing or mono-allelic expression [2, 9–12].

The detection of aberrant expression and missplicing, however, is not possible with default bulk RNA-seq workflows. Indeed, testing for differential expression by comparing each sample against the rest of the cohort is statistically invalid. With this respect, OUTRIDER [13] and FRASER [14] have disrupted the field by providing formal count-based outlier tests that pick up aberrant expression and splicing respectively, while automatically controlling for latent confounders. But, their approach is slow and FRASER is discarding a lot of useful information as it only focuses on junction reads to discover aberrant splicing. To overcome the computational burden of OUTRIDER, OutSingle [15] was developed, which uses the procedure of Gavish and Donoho [16] to determine the optimal rank of the matrix decomposition when denoising matrices. This avoids the need of OUTRIDER's hyperparameter optimisation, which reduces computational time by several orders of magnitude. OutSingle assumes

gene-expression to be log-normal distributed, which, although obtaining similar performance and better computational time compared to OUTRIDER, does not account for the heteroscedasticity and discrete nature of the count distribution of RNA-seq data. Moreover, OutSingle can also not correct for known confounders.

Here, we argue that conventional bulk methods can be unlocked for detecting aberrant expression and splicing. More specifically, they can be used for estimating the mean and dispersion of the negative binomial (NB) distribution upon including latent factors in the model, which can subsequently be plugged into the NB distribution to perform count-based outlier tests. The ASpli tool [17] is an important starting point for developing workflows for aberrant splicing detection. ASpli complements the differential splicing analysis by also conducting hypothesis tests on exonic and intronic bin reads next to junction reads, which has been shown to boost the power. However, its parameter estimation is based on edgeR's diffSpliceDGE [18] which performs worse than DEXSeq [19], while the latter scales poorly to the large cohorts in rare disease studies (Fig. 1, Supplementary Fig. 1) [20]. To overcome this, we have developed satuRn for differential transcript and exon usage [20]. satuRn is fast and scalable as it uses a quasi-binomial model that directly models the proportion of transcript (or exon) counts on the total gene count. But, satuRn's estimation approach is based on quasi-likelihood that does not provide a full distribution, which renders it useless for outlier discovery.

In this contribution, we show how juggling offsets can effectively unlock conventional bulk RNA-seq workflows for fast and scalable differential usage and aberrant splicing analyses. Indeed, by replacing the conventional offset for library size in DESeq2 or edgeR transcript or exon level analyses by the logarithm of the total gene count, the parameters of the mean model enable us to directly estimate the average transcript or exon usage, respectively. We further develop workflows on different ASpli counts, i.e. bin and junction counts, combined with the appropriate offsets to infer aberrant

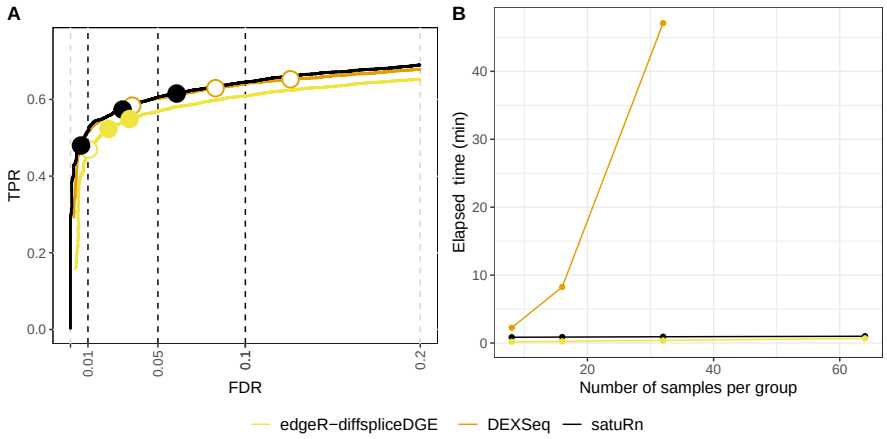

Fig. 1 Comparison of current state-of-the-art methods for assessing differential transcript usage. True positive rate (TPR) versus false discovery rate (FDR) curves for detecting differential transcript usage (panel A) and the computational time relative to the number of samples (panel B) for edgeR-diffspliceDGE, DEXSeq and satuRn in a two-group comparison. Figure adapted from Gilis et al. [20].

splicing. We also provide an unbiased and fast algorithm for parameter estimation to assess aberrant expression and splicing that scales better to the large number of latent covariates that are typically needed in studies on rare disease with large cohorts. In simulation and real case studies we show how our framework, saseR (Scalable Aberrant Splicing and Expression Retrieval), vastly outperforms existing state-of-the-art tools as DEXSeq, OUTRIDER, OutSingle and FRASER in terms of computational speed and scalability. More importantly, they also dramatically boost the performance for aberrant splicing (cf. FRASER) while maintaining a similar performance for differential usage (cf. DEXSeq) and aberrant expression detection (cf. OUTRIDER, OutSingle).

2 Results

In this manuscript, we will first use saseR to detect simulated aberrant expression events, using our fast parameter estimation and count-based outlier tests. Next, we introduce adapted offsets in conventional bulk RNA-seq tools, edgeR [18] and DESeq2 [21], to unlock them for different applications. For differential splicing, for instance, we suggest to use the log-transformed total read counts of the gene to which the feature

belongs as an offset, so that the mean model parameters get an interpretation in terms of usage, i.e. the log-transformed feature read count relative to the total read count of its corresponding gene. An overview of different counts and offsets for interesting applications is given in Table 1. We then continue with benchmarking saseR against FRASER [14, 22] to detect aberrant splicing. Finally, we assess the performance of saseR to detect real aberrant events in a case study.

Table 1 Bulk RNA-seq tools can be unlocked for different applications by carefully selecting the input data and offsets for the negative binomial model framework.

Input data (y_{ij})	Offsets (o_{ij})	Application
Gene counts	Conventional offsets for library size	Differential gene expression or aberrant expression
Exon and intron bin counts	Log of total count for gene	Differential usage or aberrant splicing
Junction counts	Log of total count for gene	Differential usage or aberrant splicing
Transcript counts	Log of total count for gene	Differential transcript usage or aberrant splicing
Allele specific counts	Log of total count over all alleles	Differential allele usage or aberrant allele usage

2.1 Detection of aberrant expression

To benchmark aberrant expression detection, the GTEx [23] and Kremer [10] datasets are used. Only suprapubic skin cells were retained from the GTEx data, originating from healthy deceased donors. The Kremer dataset contains fibroblast cell lines from patients diagnosed with mitochondrial diseases. Outliers are randomly simulated in these datasets with a frequency of 10^{-3}. The performance of saseR is benchmarked against OutSingle [15] and OUTRIDER [13]. Similar to the default releases of OutSingle and OUTRIDER, saseR is run without controlling for known confounders. We include two OUTRIDER workflows, (1) OUTRIDER-Autoencoder using a negative

binomial autoencoder to estimate and control for latent factors, and (2) OUTRIDER-PCA performing a principal component analysis on log-transformed counts, which is computationally more efficient but does not account for the count properties of the data.

Figure 2 shows the area under the precision-recall curve (AUC) for each sample, the overall precision-recall curve and the computational time on the GTEx data with simulated aberrant expression outliers. saseR, OutSingle and OUTRIDER-Autoencoder have a similar performance for detecting simulated aberrant expression outliers, and slightly out-compete OUTRIDER-PCA. Strikingly, saseR and OutSingle are much faster than OUTRIDER-Autoencoder and OUTRIDER-PCA, while saseR is an additional factor two faster than OutSingle. Indeed, by using the Gavish and Donoho threshold method [16] saseR and OutSingle by default do not require hyperparameter optimisation to select the number of latent factors. saseR can also be run with a similar hyperparameter optimisation as OUTRIDER to select the number of latent factors (see Supplementary Fig. 2). This shows that, saseR with hyperparameter optimisation has similar performance compared to its fast default workflow, while remaining much faster than OUTRIDER-Autoencoder. Although saseR's computational time with hyperparameter optimisation is slower than OUTRIDER-PCA, Supplementary Fig. 3 shows that the computational time to run a single analysis for a certain number of latent factors, genes and samples is faster for saseR. This indicates that the increased computational complexity is probably related to the outlier injection scheme required for hyperparameter optimisation, i.e. NB versus Gaussian outliers, respectively. Supplementary Fig. 3 also shows the benefits of saseR's fast estimation procedure. Indeed, parameter estimation with edgeR [18] does not scale well with increasing number of latent factors or large design matrices. saseR, however, by default considers a quadratic variance structure, which reduces each Newton-Raphson interation to a matrix multiplication and thus scales well towards large design matrices.

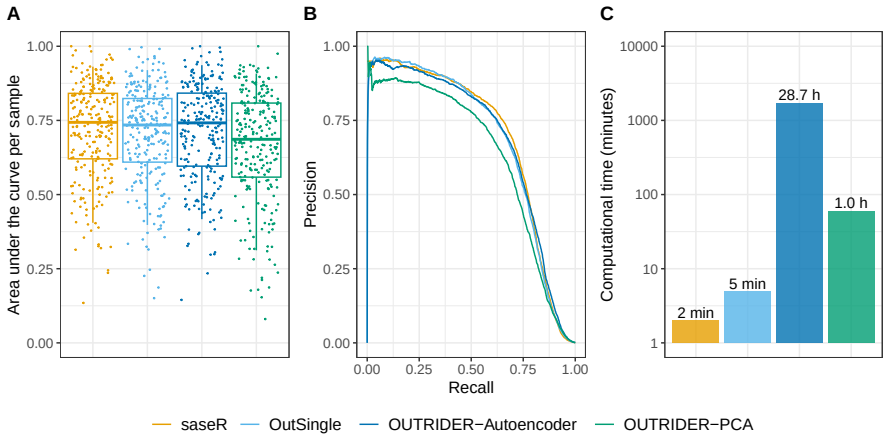

Fig. 2 Benchmark of aberrant expression detection. Comparison of performance to detect simulated expression outliers in the GTEx dataset based on area under the precision-recall curve per sample (panel A), the precision-recall curve (panel B) and the computational time (panel C). Four methods are benchmarked: saseR, OutSingle, OUTRIDER-Autoencoder and OUTRIDER-PCA. Simulated outliers were injected according to the gene-specific marginal distribution, only taking into account DESeq2 size factors for normalisation. The whiskers of the boxplots in panel A correspond to the 5th and 95th quantile.

Similar results are observed when analysing the Kremer dataset with simulated outliers (Supplementary Fig. 4).

Note, that saseR can also include known covariates to estimate the mean, although for the GTEx dataset this does not yield better performance to detect simulated outliers based on the conditional distribution (Supplementary Fig. 5). This functionality is not available for OutSingle.

2.2 Detection of differential usage

Here, we show that the analysis of differential usage (DU), i.e. changes in relative abundance of transcripts/exons/introns within the same gene, can also be done with canonical bulk RNA-seq tools when using the logarithm of the total gene count as an offset. To assess the performance of these novel workflows, we add them to the benchmark of Gilis et al. [20]. Panels A and B in Fig. 3 show the performance to pick up DU using true positive rate (TPR) versus the false discovery rate (FDR) plots for both bulk- and single-cell RNA-seq (scRNA-seq) datasets, respectively. Panel C

shows the computation time in function of the number of samples. DEXSeq, edgeR-diffspliceDGE, satuRn, and our novel edgeR and DESeq2 workflows with adapted offsets are included in our comparison. Note, that the computational time for DEXSeq is not included in panel C because it was several orders of magnitudes slower, which we already have shown in Figure 1. The computational time of our novel edgeR and DESeq2 workflows are in line with the other methods. edgeR is even slightly faster than satuRn, but, it remains slightly slower than edgeR-diffspliceDGE. The performances of edgeR and DESeq2 using adapted offsets to detect DU on bulk RNA-seq data are comparable to DEXSeq and satuRn, and outperform edgeR-diffspliceDGE. On scRNA-seq data, however, satuRn still outperforms all other methods. DEXSeq performs slightly better than edgeR with adapted offsets, closely followed by edgeR-diffspliceDGE and DESeq2 with adapted offsets. Note, however, that this comparison only involved 20 vs 20 cells as DEXSeq does not scale to the data volumes in real scRNA-seq datasets (see Figure 1).

2.3 Detection of aberrant splicing

In this section we show how intron, exon and junction counts can be modeled with bulk RNA-seq tools to pick up aberrant splicing. Indeed, when using the logarithm of the total gene count as an offset, the mean model parameters again get an interpretation in terms of usages. To benchmark the performance of our workflows, aberrant splicing outliers are injected in the lymphoblastoid cell lines from the healthy patients of the Geuvadis [24] dataset, using the RSEM simulator [25]. Again, outliers are injected with a frequency of 10^{-3}. We evaluate different workflows for saseR: (1) saseR-bins using bin read counts (exon and intron) and the logarithm of the total gene counts as offset, (2) saseR-junctions using junction reads and the logarithm of the sum of the junction read counts per gene as offset, and (3) saseR-ASpli using junction reads and as offset the logarithm of the sum of the junction reads that have at least one splice site in common,

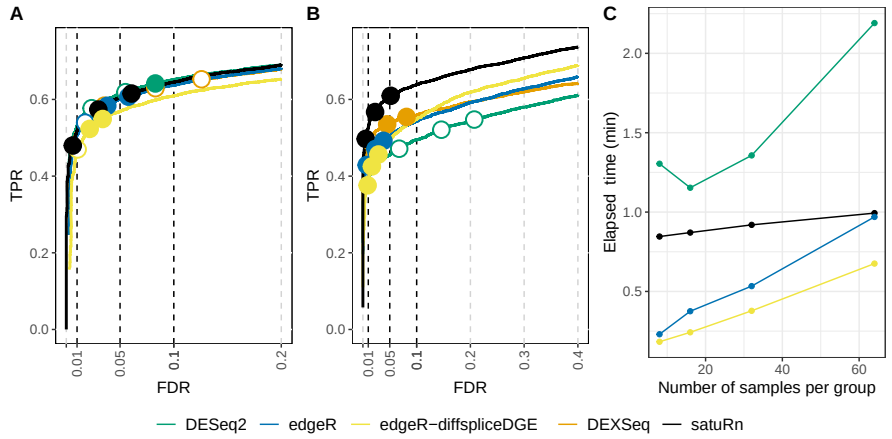

Fig. 3 Benchmark of differential usage detection. Comparison of performance for differential transcript usage between satuRn, DEXSeq, edgeR-diffspliceDGE, and edgeR and DESeq2 with adapted offsets. The performance of the methods is compared on basis of true positive rate (TPR) versus false discovery rate (FDR) curves for a 5 vs 5 comparison on bulk RNA-seq data (panel A), for a 20 vs 20 comparison on scRNA-seq data (panel B) and the computational time relative to the number of samples (panel C). The three circles on each TPR-FDR curve represent the working points when the FDR level is set at nominal levels of 1%, 5% and 10%, respectively. The circles are filled if the empirical FDR is equal or below the imposed FDR threshold. Note, that the computational time of DEXSeq is not shown in panel C because it is several orders of magnitude larger than for the other tools, which was already illustrated in Figure 1.

i.e. based on the ASpli junction cluster. These three saseR workflows are benchmarked against FRASER [14], which can use different autoencoders to control for confounders, i.e. a beta-binomial autoencoder (FRASER-Autoencoder), a PCA encoder and beta-binomial decoder (FRASER-BB-Decoder) and PCA (FRASER-PCA). All comparison are based on the prioritisation of genes in which outliers were injected. A gene p-value was obtained by using the minimal p-value of all features belonging to that gene.

Note that the results in the main paper are based on FRASER's novel Intron Jaccard Index [22], which combines the former aberrant donor, acceptor and intron retention metrics [14]. We refer to this method as FRASER 2.0. The hyperparameter optimisation of FRASER was done with PCA to reduce computational time, which is its default setting. Results with FRASER's donor and acceptor metrics are included in Supplementary Information.

Figure 4 shows the AUC for each sample, the overall precision-recall curve and the computational time for saseR-bins, saseR-junctions, saseR-ASpli and FRASER 2.0. All saseR workflows outperform these of FRASER 2.0. saseR-bins and saseR-junctions have the best power, followed by saseR-ASpli. The precision-recall curve shows that the precision of FRASER 2.0 never reaches high levels, even at a low recall of simulated outliers.

Remarkably, FRASER-PCA 2.0 performs better in our benchmark than its BB-Decoder and Autoencoder variants. This was not observed using the FRASER donor and acceptor metrics (Supplementary Fig. 6). But, the performance of these older methods never reaches these of FRASER-PCA, let alone those of saseR. To rule out convergence issues as the cause for the lack of performance of the FRASER 2.0 BB-decoder and autoencoder methods, we removed the junctions for which the decoder matrix did not converge. This, however, did not improve the results considerably (Supplementary Fig. 7).

To ensure a fair comparison, we also assess the impact of the filtering strategies of saseR and FRASER. The performances shown in Fig. 4 are solely based on the outliers that were included in its corresponding output, and filtered outliers are thus ignored. Alternatively, we assess the performance by enforcing all methods to use the same set of outliers. On the one hand, we consider the union of all outliers in the output of all methods. When an outlier is filtered for a specific workflow its p-value was set at 1. The results for this analysis remain very similar. The power of saseR-bins and FRASER remained more or less the same. The power of saseR-junctions and saseR-ASpli reduced slightly because they filtered more outliers. However, they still largely outperform FRASER (Supplementary Fig. 8). On the other hand, we also considered the intersection of the outliers in all methods, which did not alter the results (Supplementary Fig. 9).

Finally, we assess the performance when injecting outliers in the Kremer dataset with FRASER's method. When injecting outliers based on jaccard counts, saseR-junctions still largely outperforms FRASER-PCA 2.0, and FRASER with donor and acceptor metrics (Supplementary Fig. 10). saseR-bins and saseR-ASpli could not be benchmarked on the Kremer dataset, as only junction reads are publicly available. Note, that it is also possible to include junction counts and jaccard offsets in saseR. This, however, does not lead to consistent results. It dramatically outperforms FRASER-PCA 2.0 in the Geuvadis benchmark and reaches similar performance as saseR-junctions (Supplementary Fig. 11), but has a slightly lower power in the Kremer benchmark compared to FRASER-PCA 2.0, which are both outperformed by saseR-junctions (Supplementary Fig. 12).

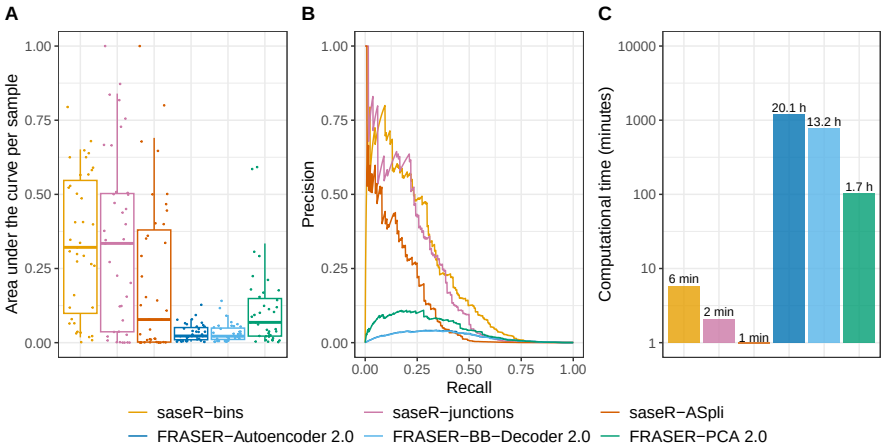

Fig. 4 Benchmark of aberrant splicing detection. Comparison of performance to detect RSEM simulated splicing outliers in the Geuvadis dataset based on area under the precision-recall curve per sample (panel A), the precision-recall curve (panel B) and the computational time (panel C). saseR, with bin reads (saseR-bins), with junction reads and the logarithm of the total junction read counts per gene as offset (saseR-junctions), and with junction reads and the logarithm of the total junctions read count per ASpli junction cluster (saseR-ASpli) are benchmarked against FRASER 2.0 workflows, which use the Intronic Jaccard Index, i.e. one with an autoencoder (FRASER-Autoencoder 2.0), a beta-binomial decoder matrix (FRASER-BB-Decoder 2.0) and PCA (FRASER-PCA 2.0). Note, that PCA is always used for hyperparameter optimisation, which is FRASER 2.0's default to reduce computational time. The whiskers of the boxplots in panel A correspond to the 5th and 95th quantile.

Interestingly, saseR considering bins, junction or ASpli junctions is also much faster than FRASER, even when considering hyperparameter optimisation (Supplementary

Fig. 13). Moreover, Supplementary Fig. 14 shows that our fast estimation procedure with 50 iterations is as good as the slower edgeR implementation. But, the additional speed gain of restricting the number of iterations cannot be justified as it seems to lead to a sub-optimal precision-recall.

2.4 Case study: Kremer dataset

saseR, OutSingle, OUTRIDER and FRASER are compared to detect aberrant expression and splicing events in real rare disease cases from the Kremer dataset. Again, only saseR-junctions is used for prioritising genes with aberrant splicing, because no BAM files are publicly available for the Kremer dataset. We assess if these methods can discover the novel genes discussed by Kremer et al. [10], Brechtmann et al. [13] and Mertes et al. [14], as well as the list of disease related genes with aberrant splicing that are reported in the FRASER paper [14]. Although some gene variants (*ALDH18A1* and *MCOLN1*) are known to be related to mono-allelic expression, these were picked up by Kremer et al. [10] and are also considered here. The rank of the p-value of the disease-related genes that were validated in a specific patient are shown in Table 2. Every line corresponds to a validated disease-related gene in a different patient.

saseR aberrant expression, OutSingle and OUTRIDER show similar performance to prioritise the previously reported disease-related genes. Only *TAZ* and *COASY* are not easily detected by the four methods and OUTRIDER-PCA can also not prioritise *SFXN4*.

When assessing aberrant splicing with saseR-junctions and FRASER 2.0, it can be observed that the ranking of saseR-junctions is better in picking up the previously reported disease-related genes than FRASER 2.0 with the PCA and Autoencoder method. The most remarkable differences are *TAZ* and *TALDO1*, which are prioritised by saseR-junctions while they are missed by the FRASER 2.0 workflows. When using FRASER with donor and acceptor metrics (Supplementary Table 1) the prioritisation

Table 2 Detection of disease-related genes. Prioritisation based on the rank of the p-values for disease-related gene within diagnosed patients using saseR, OutSingle, OUTRIDER AUTO (autoencoder), OUTRIDER PCA (principal component analysis), saseR-junctions, FRASER 2.0 AUTO and FRASER 2.0 PCA.

Gene	Aberrant expression				Aberrant splicing		
	saseR	OutSingle	OUTRIDER		saseR	FRASER 2.0	
			AUTO	PCA	junctions	AUTO	PCA
MGST1	2	3	3	9	5840	3688	4324
TIMMDC1	1	1	1	1	2	1	2
TIMMDC1	1	2	1	3	7	9	6
CLPP	1	1	1	1	1	1	1
TAZ	2994	2065	1717	772	1	1029	1153
TANGO2	1	1	1	1	2783	831	801
TALDO1	1	1	1	1	9	87	37
SFXN4	2	2	5	100	5	53	4
COASY	81	77	212	122	4	7	4
PANK2	12	14	9	6	3	2	2
ALDH18A1	1	1	1	7	3985	1958	1470
MCOLN1	1	1	1	1	3	1	1

of most disease-related genes is worse. Also, a saseR workflow with junction counts and jaccard offsets leads to suboptimal rankings compared to saseR-junctions and FRASER 2.0 (Supplementary Table 1).

Finally, an analysis with saseR using hyperparameter optimisation to determine the number of latent factors returns similar results as our default workflow with the Gavish and Donoho [16] threshold (Supplementary Table 2).

3 Discussion

In this contribution we developed saseR, a framework that unlocks bulk RNA-seq tools for fast and scalable differential splicing, aberrant splicing and expression analysis. Our key idea is to use specific RNA-seq counts in conjunction with well-chosen offsets to facilitate the proper interpretation of the mean model parameters for a specific application, e.g. gene counts and conventional offsets to correct for library size for the expression based analyses; exon or junction counts with respectively the exon or junction count per gene as an offset for usage and splicing based analysis; amongst others.

Upon parameter estimation, the conventional bulk RNA-seq inference framework can then be used when one wants to infer differential expression or usage. When the aim is to infer aberrant expression or splicing, the estimated mean model and dispersion parameters are simply plugged into the negative binomial distribution to obtain the corresponding quantile to assess how extreme the observed count is for each feature in each sample.

Hence, our approach has the advantage of providing a single, unified framework to infer a wide range of applications. Moreover, in contrast to current state-of-the-art methods for aberrant splicing that only consider junction reads, it is also future-proof to novel sequencing-based technologies and applications, such as transcript counts with long-read sequencing and allele specific expression, amongst others, as long as the quantification can be recasted in specific feature counts in conjunction with proper offsets.

For aberrant splicing applications, saseR outperforms the current state-of-the-art method FRASER with donor, acceptor and Jaccard metrics [14, 22] both in terms of outlier detection as well as in computational complexity. saseR allows for different count inputs, such as bin read counts and junction read counts. This improves upon FRASER, which only uses junction read counts and discards much information on aberrant splicing that is also present in short-read RNA-seq exon and intron bin reads. With saseR we still provide a separate junction read count workflow, because bin read counts are less suited to pick up novel splice sites. We convincingly showed saseR's superior performance in our simulation studies on aberrant transcript splicing outliers simulated with RSEM [25] as well as on junction outliers simulated with FRASER's jaccard outlier injection scheme, and in our case study that focuses on prioritising disease-related genes in the Kremer dataset [10].

For aberrant expression and differential usage on bulk RNA-seq data, the performance of the saseR workflows is at least on par with current state-of-the-art methods,

such as OutSingle [15] and OUTRIDER [13] for aberrant expression, and DEXSeq [19] and satuRn [20] for differential usage analysis. However, saseR dramatically outperforms the existing methods in terms of computational time and/or flexibility to formulate the mean model.

The poor scalability of DEXSeq stems from modelling both the counts for a specific feature and the other counts for the same gene in one model, which requires the introduction of blocking factor for each sample to address the within sample correlation. These sample specific intercepts, therefore, leads to an explosion of the design matrix with increasing sample size. By normalising each feature with a well-chosen offset, e.g. the logarithm of the total gene count per sample to which the feature belongs, the mean model parameters also get the interpretation of a ratio without having to estimate a sample specific model parameter, which vastly improves the scalability. This approach can also be motivated theoretically due to the well known approximation of a multinomial model by a Poisson model with the total counts as an offset, which is extended towards a negative binomial distribution in the presence of overdispersion.

For aberrant detection and aberrant splicing only an unbiased estimator for the mean model and dispersion parameters is required, which are subsequently plugged into the negative binomial distribution for outlier discovery. Therefore, saseR introduces a fast algorithm by assuming a quadratic variance structure when estimating the mean model parameters, which reduces each Newton-Raphson iteration to matrix multiplication. We show that this parameter estimator remains unbiased even when the variance structure is misspecified. We then use the mean model parameter estimates in edgeR's *estimateDisp* function to estimate the negative binomial dispersion. This approach improves the scalability dramatically for large cohort studies of rare diseases, which often require many latent factors to be included when estimating the mean model. We have shown that our fast algorithm for parameter estimation has

equal performance to detect aberrant splicing events compared to edgeR, while improving the computational time by orders of magnitude for large studies. This approach, however, cannot be used for differential analyses because a misspecification of the variance structure renders the downstream inference invalid.

A further improvement upon OUTRIDER and FRASER is the use of Gavish and Donoho threshold [16] to determine the number of latent factors to include in the mean model, which avoids the computational intensive hyperparameter optimisation. Salkovic et al. [15] already introduced this idea in the context of aberrant expression detection. However, they considered RNA-seq read counts to be log-normal distributed, ignoring heteroscedasticity and the count nature of the data. Moreover, their method also lacks the flexibility to specify the mean model structure and cannot be used for other applications. We also implemented the option to select the number of latent factors in saseR with a similar hyperparameter optimisation as OUTRIDER and FRASER and showed that the performance for outlier detection remained very similar to the fast Gavish and Donoho threshold method.

In conclusion, we developed a novel and very flexible framework saseR for fast and scalable analysis for differential usage, aberrant splicing and aberrant expression that dramatically outperforms state-of-the-art methods in terms of computational complexity. Interestingly, it also boosts the performance to detect aberrant splicing in rare diseases. Moreover, our approach has the advantage that it provides a unified workflow for many applications. Indeed, the user only has to change the input towards the proper count matrix and offsets for their specific application, which makes it generally applicable, user-friendly, and future-proof for current and novel sequencing-based technologies and applications.

4 Methods

We first introduce the framework of saseR, and explain how it can infer aberrant expression, aberrant splicing and differential splicing using adapted offsets in the negative binomial framework. Next, we show how we control for unknown confounders, and develop a novel algorithm for parameter estimation that scales to large design matrices. We conclude with an overview of the datasets and our benchmarking protocols.

4.1 Detection of aberrantly expressed genes

Conventional bulk RNA-seq tools for differential analysis use a negative binomial framework to estimate the mean expression for each feature [18, 21] which can be formulated as:

$$\begin{cases} y_{ij} \sim NB(\mu_{ij}, \theta_j) \\ \log(\mu_{ij}) = \eta_{ij} \\ \eta_{ij} = \boldsymbol{x_i^T}\boldsymbol{\beta_j} + o_{ij}, \end{cases} \tag{1}$$

with y_{ij} the observed count for feature j of sample i, μ_{ij} the sample specific mean of feature j, θ_j the negative binomial dispersion parameter for feature j, η_{ij} the linear predictor of feature j for sample i, $\boldsymbol{x_i^T}$ the covariate pattern for sample i, $\boldsymbol{\beta_j}$ a vector with the corresponding model parameters for feature j, and o_{ij} an offset for feature j in sample i to normalise for differences in library size between samples. Note, that conventional bulk RNA-seq tools by default consider the same offset for all features. However, in user defined workflows they allow the user to specify different offsets for each feature, which we will exploit in this paper.

Upon parameter estimation, differentially expressed features are prioritised by testing on a single mean model parameter or a linear combination of model parameters that corresponds with the research hypothesis of interest. However, conventional hypothesis testing for prioritising aberrant expression in the context of rare diseases is invalid

because the condition of each subject is typically caused by another feature. Therefore, the problem reduces to outlier detection and aberrant features can be prioritised by assessing how extreme the quantile of an observed feature is given its estimated distribution. For RNA-seq data, the negative binomial distribution is typically used and the quantile can be transformed to a kind of "two sided p-value" [13]

$$p_{ij} = 2 \times min\big(0.5, \sum_{1}^{y_{ij}} P(y_{ij}|\mu_{ij}, \theta_j), 1 - \sum_{1}^{y_{ij}-1} P(y_{ij}|\mu_{ij}, \theta_j)\big), \tag{2}$$

which can be estimated using conventional bulk RNA-seq tools such as edgeR [18]. Due to the discrete nature of the distribution, the p-values have to be restricted to be at most 1. These two-sided p-values can be used to rank the genes according to their magnitude of aberrant expression. Note, that a distribution is needed to compute these quantiles, so we cannot resort to quasi-likelihood based workflows.

4.2 Differential usage and aberrant splicing

For differential usage or aberrant splicing, one has to estimate the relative abundance or proportion for a certain exon, intron or transcript relative to the total expression of all features mapping to a particular gene. To overcome this, we developed our satuRn [20] tool, which uses quasi-binomial likelihood. However, with quasi-likelihood only the first two moments of the distribution are modelled, which renders it useless for outlier detection. Similar to FRASER, we could resort to the beta-binomial distribution to address heteroscedasticity in the binomial counts. However, fitting a beta-binomial is slow.

Alternatively, DEXSeq [19] could be used, which models the counts of specific feature and the other counts of all features that map to the same gene using a negative binomial model. But, the DEXSeq approach introduces a subject specific intercept to account for the correlation between the counts and the other counts as they are measured for the same subject, which leads to an explosion of the size of the design

matrix in studies with many subjects. Therefore, DEXSeq does not scale to the large data compendia that are typically used in the context of rare diseases.

We argue that similar results to DEXSeq can be obtained using a negative binomial model with an offset for the total count of all features that map to a gene. Indeed, the mean model parameters in Equation 1 then also get an interpretation in terms of the log-ratio relative to the total count for a gene, which unlocks bulk RNA-seq tools for differential usage and aberrant splicing applications. To avoid taking the logarithm of 0, a pseudo-count of 1 is added to gene counts that are equal to 0, as well as to their corresponding feature count of 0.

Table 1 in the results section gives an overview for interesting applications that can be modeled using different kinds of input count and offset combinations.

For each application, saseR will use different counts and offsets in its workflow, i.e. workflows for intron and exon bin counts as well as junction counts are developed for aberrant splicing. For the former we will use the logarithm of the total count over all bins that map to a gene as an offset. For the latter we will consider two workflows with different offsets, one with the logarithm of the sum over all junction counts that map to a gene, the other with an offset derived of ASpli [17] junction clusters, which do not require prior annotation. These junction clusters correspond to all junctions that have at least a splice site in common and are needed to infer novel/unknown splice sites.

4.3 Correction for latent confounders

When performing differential or aberrant event detection in large data compendia one typically has to account for unknown confounders. To correct for these latent factors, several algorithms have been developed, e.g. [26, 27]. In saseR, we use RUV [27] that adopts a singular value decomposition on the deviance residuals to estimate these latent confounders, which are subsequently incorporated in the negative binomial model as covariates.

To estimate the optimal number of latent factors, two different approaches were used. On the one hand, we implement the optimal hard threshold for singular values of Gavish and Donoho [16], which was adopted recently by Salkovic et al. [15] in the context of aberrant expression. As opposed to Salkovic et al. who assume the RNA-seq counts to be log-normal distributed and therefore do not account for the heteroscedastic nature of counts, our method explicitly models the data using negative binomial models. On the other hand, we also implement a similar approach to OUTRIDER [13] and FRASER [14, 22]: corrupted counts are injected in the data to estimate the number of latent factors. The corrupted counts replace original counts with a frequency of 10^{-2}. Then, a grid search is performed that varies the number of latent factors, and the discovery of the corrupted counts is then evaluated. The number of latent factors that obtains the highest area under the precision-recall curve is then used for the final analysis on the original data. Note, that a grid search implies that this approach will by much slower than the strategy from Gavish and Donoho [16] because the NB-model has to be fitted for each grid point. For details on the simulation of corrupted counts for aberrant expression and splicing, we refer the reader to Supplementary Information.

4.4 Fast parameter estimation for large design matrices

The mean model parameters of the negative binomial model are commonly estimated using a Newton-Raphson algorithm. It iteratively solves

$$\boldsymbol{\beta}_j^{k+1} = \boldsymbol{\beta}_j^k + (\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{W}_{\boldsymbol{j}}\boldsymbol{X})^{-1}\boldsymbol{X}^{\boldsymbol{T}}\frac{\partial \boldsymbol{\eta}_{\boldsymbol{j}}}{\partial \boldsymbol{\mu}_{\boldsymbol{j}}}\boldsymbol{W}_{\boldsymbol{j}}(\boldsymbol{y}_{\boldsymbol{j}} - \boldsymbol{\mu}_{\boldsymbol{j}}),$$

with $\boldsymbol{W}_{\boldsymbol{j}}$ a $n \times n$ feature-specific diagonal weight matrix with elements

$$w_{ii} = \frac{\partial \mu_{ij}}{\partial \eta_{ij}} Var(y_{ij})^{-1} \frac{\partial \mu_{ij}}{\partial \eta_{ij}}.$$

For the detection of aberrant events, and mainly for aberrant expression, the required number of latent factors to obtain optimal power can be large as large data compendia are typically used for this purpose. For large design matrices $\boldsymbol{X}$, the computation of the Newton-Raphson equation does not scale well and becomes slow. Therefore, we introduce a method for parameter estimation that is fast, scalable with large design matrices and provides an unbiased estimator of the mean model parameters.

In particular, we replace the NB variance structure, $Var(Y_{ij}) = \mu_{ij} + \theta_j \mu_{ij}^2$ by a quadratic variance structure

$$Var(Y_{ij}) = \phi_j \mu_{ij}^2,$$

together with the log link function

$$log(\mu_{ij}) = \eta_{ij},$$

which reduces the diagonal elements of weight matrix $\boldsymbol{W_j}$ in each Newton-Raphson iteration to a feature-specific constant,

$$w_{ii} = \mu_{ij} \frac{1}{\phi_j \mu_{ij}^2} \mu_{ij} = \frac{1}{\phi_j},$$

implying that the parameter estimator

$$\begin{aligned} \boldsymbol{\beta}_j^{k+1} &= \boldsymbol{\beta}_j^k + (\boldsymbol{X^T} \frac{1}{\phi_j} \boldsymbol{X})^{-1} \boldsymbol{X^T} \frac{1}{\phi_j} \frac{1}{\boldsymbol{\mu_j}} \boldsymbol{y_j} - \boldsymbol{\mu_j} \\ &= \boldsymbol{\beta}_j^k + (\boldsymbol{X^T X})^{-1} \boldsymbol{X^T} \frac{(\boldsymbol{y_j} - \boldsymbol{\mu_j})}{\boldsymbol{\mu_j}}, \end{aligned}$$

no longer involves feature-specific matrices, and shows that each iteration is reduced to a fast matrix multiplication. In case of misspecification of the variance structure, the estimator remains unbaised

$$\begin{aligned} E[\hat{\boldsymbol{\beta}}_j] &= \boldsymbol{\beta}_j + (\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{X})^{-1}\boldsymbol{X}^{\boldsymbol{T}}\frac{(E[\boldsymbol{y}_j] - \boldsymbol{\mu}_j)}{\boldsymbol{\mu}_j} \\ &= \boldsymbol{\beta}_j + (\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{X})^{-1}\boldsymbol{X}^{\boldsymbol{T}}\frac{(\boldsymbol{\mu}_j - \boldsymbol{\mu}_j)}{\boldsymbol{\mu}_j} \\ &= \boldsymbol{\beta}_j. \end{aligned}$$

Upon the estimation of the mean model parameters, the feature-specific dispersion θ_j can then be calculated using standard bulk RNA-seq tools, such as edgeR [18].

Note, that our fast parameter estimator can only be used for aberrant detection, which only requires an unbiased estimator of the mean and dispersion. Indeed, the estimators are then plugged into the NB distribution and no further inference is required on the parameter estimators themselves. Differential analysis, however, involves statistical hypothesis tests on (contrasts of) the mean model parameters, and the misspecification of the variance could lead to incorrect standard errors on (contrasts of) the mean model parameters and thus to incorrect inference for these applications.

4.5 Data

Three different datasets are used in this work. First, similar to [13], gene expression read counts were downloaded from GTEx portal (version V6P counted with RNA-SeQC v1.1.8) [23]. Only sequencing reads from suprapubic skin cells were retained and samples with a lower RNA integrity number than 5.7 were removed. Genes were kept when having at least 1 fragment per kilobase of transcript per million mapped reads for 5% of the samples. This filtering was done with DESeq2 [21]. Also, genes were only

kept when having at least 1 or more read counts in 25% of the samples. Second, gene, junction and intron-exon boundary reads were downloaded from Zenodo [10, 28] for the Kremer dataset, which contains samples suspected to suffer from Mendelian diseases. The gene expression read counts were filtered in the same way as the GTEx dataset, while the junction and intron-exon boundary reads were filtered using the standard filtering of FRASER [14]. Third, FASTQ-files of 39 samples from the Geuvadis dataset [24] were used (ERR188023-ERR188062, of which ERR188032 was removed due to errors with alignment), which were aligned to the protein coding genes of the of the GRCh38.p13 primary genome assembly [29] using STAR (version 2.7.10b) [30].

The benchmarks on differential usage were performed by using the benchmark framework described in the satuRn paper [20], to which we refer the reader for more details.

4.6 Outlier simulation

4.6.1 Aberrant expression

Different methods to detect aberrant expression were benchmarked by injecting artificial outliers in the GTEx and Kremer datasets. The value of the outlier was determined by using a quantile of the negative binomial distribution, specified by a gene-specific mean and dispersion parameter. These parameters are obtained by performing a negative binomial regression of the read counts with a linear predictor with only an intercept and an offset with sample specific size factors obtained by DESeq2 [21]. The quantile used in the benchmarks correspond to the quantile of Z=3 in the standard normal distribution. Both over- and underexpression outliers were injected. This is conceptually similar to the artificial outlier injection of OUTRIDER [13]. However, we use proper counts from a negative binomial instead of a log-normal approximation.

4.6.2 Aberrant splicing

To simulate aberrantly spliced genes, the read counts corresponding to a transcript within that gene should be increased, while the read counts corresponding to another transcript should be decreased. As this cannot be done starting from a exon or junction count matrix, RSEM (version 1.3.0) [25] is used to simulate these outliers in a similar way as Soneson et al. [31] did for differential usage benchmarks. First, RSEM estimates transcripts per million expression values in each sample. Then, from randomly selected genes, the two most expressed transcripts were further used. Candidate transcripts were only considered if the total gene expression is greater than 100, the two transcripts both have an expression proportion larger than 10%, and the difference between both expression proportions is larger than 30%. The expected expression proportions of these two transcripts are then switched to simulate aberrant splicing outliers. Next, based on these expected proportions, a Dirichlet distribution is used to simulate new transcript counts per million. FASTQ-files were simulated based on these transcript counts per million, using the same library sizes as the original samples. These files were again aligned using STAR (version 2.7.10b) [30].

5 Data availability

No data were generated for this study. The GTEx v6p dataset is available through dbGaP (accession number: phs000424.v6.p1) at https://gtexportal.org/home. Gene, junctions, and intron-exon boundary read counts from the Kremer dataset [10] were downloaded from Zenodo (https://zenodo.org/record/4271599 [28]). FASTQ files from the Geuvadis dataset [24] were downloaded from https://www.ncbi.nlm.nih.gov/sra. The data from Gilis et al. [20], used for the differential usage benchmarking are available at https://doi.org/10.5281/zenodo.6826603 [32]. The scripts used to simulate aberrant expression and splicing are available on our companion GitHub repository for this paper: https://github.com/statOmics/saseRPaper/.

6 Code availability

All code to reproduce the analyses, figures, and tables in the paper is available on our companion GitHub repository https://github.com/statOmics/saseRPaper/. saseR with vignettes for the different workflows will be submitted to Bioconductor.

References

[1] Chial, H. Mendelian genetics: Patterns of inheritance and single-gene disorders. *Nat. Sci. Educ.* **1**, 63 (2008).

[2] Frésard, L. *et al.* Identification of rare-disease genes using blood transcriptome sequencing and large control cohorts. *Nat. Med.* **25**, 911–919 (2019).

[3] Yang, Y. *et al.* Clinical whole-exome sequencing for the diagnosis of mendelian disorders. *N Engl J Med* **369**, 1502–1511 (2013).

[4] Chong, J. X. *et al.* The genetic basis of Mendelian phenotypes: discoveries, challenges, and opportunities. *Am. J. Hum. Genet.* **97**, 199–215 (2015).

[5] Ferreira, C. The burden of rare diseases. *Am. J. Med. Genet. A* **179**, 885–892 (2019).

[6] Turro, E. *et al.* Whole-genome sequencing of patients with rare diseases in a national health system. *Nature* **583**, 96–102 (2020).

[7] Chung, C. C. Y. *et al.* Meta-analysis of the diagnostic and clinical utility of exome and genome sequencing in pediatric and adult patients with rare diseases across diverse populations. *Genet. Med.* **25**, 9 (2023).

[8] French, J. D. & Edwards, S. L. The role of noncoding variants in heritable disease. *Trends Genet.* **36**, 880–891 (2020).

[9] Cummings, B. B. *et al.* Improving genetic diagnosis in Mendelian disease with transcriptome sequencing. *Sci. Transl. Med.* **9**, eaal5209 (2017).

[10] Kremer, L. S. *et al.* Genetic diagnosis of Mendelian disorders via RNA sequencing. *Nat. Commun.* **8**, 15824 (2017).

[11] Murdock, D. R. *et al.* Transcriptome-directed analysis for Mendelian disease diagnosis overcomes limitations of conventional genomic testing. *J. Clin. Investig.* **131**, 1 (2021).

[12] Yepez, V. A. *et al.* Clinical implementation of RNA sequencing for Mendelian disease diagnostics. *Genome Med.* **14**, 38 (2022).

[13] Brechtmann, F. *et al.* OUTRIDER: a statistical method for detecting aberrantly expressed genes in RNA sequencing data. *Am. J. Hum. Genet.* **103**, 907–917 (2018).

[14] Mertes, C. *et al.* Detection of aberrant splicing events in RNA-seq data using FRASER. *Nat. Commun.* **12**, 529 (2021).

[15] Salkovic, E., Sadeghi, M. A., Baggag, A., Salem, A. G. R. & Bensmail, H. OutSingle: a novel method of detecting and injecting outliers in RNA-Seq count data using the optimal hard threshold for singular values. *Bioinformatics* **39** (2023).

[16] Gavish, M. & Donoho, D. L. The optimal hard threshold for singular values is $4/\sqrt{3}$. *IEEE Trans. Inf. Theory* **60**, 5040–5053 (2014).

[17] Mancini, E., Rabinovich, A., Iserte, J., Yanovsky, M. & Chernomoretz, A. ASpli: integrative analysis of splicing landscapes through RNA-Seq assays. *Bioinformatics* **37**, 2609–2616 (2021).

[18] Robinson, M. D., McCarthy, D. J. & Smyth, G. K. edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics* **26**, 139–140 (2009).

[19] Anders, S., Reyes, A. & Huber, W. Detecting differential usage of exons from RNA-seq data. *Genome Res.* **22**, 2008–2017 (2012).

[20] Gilis, J., Vitting-Seerup, K., Van den Berge, K. & Clement, L. satuRn: Scalable analysis of differential transcript usage for bulk and single-cell RNA-sequencing applications. *F1000research* **10** (2021).

[21] Love, M. I., Huber, W. & Anders, S. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biol.* **15**, 550 (2014).

[22] Scheller, I. F., Lutz, K., Mertes, C., Yépez, V. A. & Gagneur, J. Improved detection of aberrant splicing using the Intron Jaccard Index. Preprint at https://www.medrxiv.org/content/early/2023/04/03/2023.03.31.23287997 (2023).

[23] The GTEx consortium *et al.* The Genotype-Tissue Expression (GTEx) pilot analysis: multitissue gene regulation in humans. *Science* **348**, 648–660 (2015).

[24] Lappalainen, T. *et al.* Transcriptome and genome sequencing uncovers functional variation in humans. *Nature* **501**, 506–511 (2013).

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

[26] Stegle, O., Parts, L., Piipari, M., Winn, J. & Durbin, R. Using probabilistic estimation of expression residuals (PEER) to obtain increased power and interpretability of gene expression analyses. *Nat. Protoc.* **7**, 500–507 (2012).

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

[28] Yepez, V. A. Gene expression and splicing counts from the Kremer et al study. Zenodo https://doi.org/10.5281/zenodo.4271599 (2020).

[29] Frankish, A. *et al.* GENCODE 2021. *Nucleic Acids Res.* **49**, D916–D923 (2020).

[30] Dobin, A. *et al.* STAR: ultrafast universal RNA-seq aligner. *Bioinformatics* **29**, 15–21 (2012).

[31] Soneson, C., Matthes, K. L., Nowicka, M., Law, C. W. & Robinson, M. D. Isoform prefiltering improves performance of count-based methods for analysis of differential transcript usage. *Genome Biol.* **17**, 12 (2016).

[32] Gilis, J., Vitting-Seerup, K., Van den Berge, K. & Clement, L. Datasets associated with the publication of the "satuRn" R package. Zenodo https://doi.org/10.5281/zenodo.4439415 (2021).

Acknowledgments. This work was supported by grants from Ghent University Special Research Fund (BOF20/GOA/023) (A.S., E.D.B., L.C., M.V.H.), Research Foundation Flanders (FWO G062219N) (A.S, J.G., L.C.) and (FWO SB fellowship No. 3S037119) (J.G.). E.D.B. is a Senior Clinical Investigator (1802220N) of the FWO.

The Genotype-Tissue Expression (GTEx) Project was supported by the Common Fund of the Office of the Director of the National Institutes of Health, and by NCI, NHGRI, NHLBI, NIDA, NIMH, and NINDS. The data used for the analyses described in this manuscript were obtained from the GTEx Portal on 10/21/22 under accession number phs000424.v6.p1.

Author contributions. A.S. and L.C. conceived and designed the study. A.S. implemented the methods. A.S. and J.G. analyzed the data. A.S. and L.C. wrote the paper. E.D.B., J.G. and M.V.H. contributed to discussions and revisions of the initial draft.

Competing interests. The authors declare no competing interests.

Supplementary information. Supplementary information is provided.