

## Detection of aberrant events in RNA sequencing data

Vicente A. Yépez<sup>1,2,\*</sup>, Christian Mertes<sup>1,\*</sup>, Michaela F. Müller<sup>1</sup>, Daniela S. Andrade<sup>1</sup>, Leonhard Wachutka<sup>1</sup>, Laure Frésard<sup>3</sup>, Mirjana Gusic<sup>4,5</sup>, Ines Scheller<sup>1</sup>, Patricia F. Goldberg<sup>1</sup>, Holger Prokisch<sup>4,5</sup>, Julien Gagneur<sup>1,2,\*</sup>

<sup>1</sup> Department of Informatics, Technical University of Munich, Munich, Germany

<sup>2</sup> Quantitative Biosciences Munich, Department of Biochemistry, Ludwig-Maximilians Universität, Munich, Germany

<sup>3</sup> Department of Pathology, School of Medicine, Stanford University, Stanford, CA, USA

<sup>4</sup> Institute of Human Genetics, Helmholtz Zentrum München, Neuherberg, Germany

<sup>5</sup> Institute of Human Genetics, Technical University of Munich, Munich, Germany

\*Corresponding authors

**RNA sequencing (RNA-seq) has emerged as a powerful approach to discover disease-causing gene regulatory defects for individuals affected with a genetically undiagnosed rare disorder. Pioneer studies have shown that RNA-seq could increase diagnostic rates over DNA sequencing alone by 8% to 36 % depending on disease entities and probed tissues. To accelerate the adoption of RNA-seq among human genetic centers, detailed analysis protocols are now needed. Here, we present a step-by-step protocol that instructs how to robustly detect aberrant expression, aberrant splicing, and mono-allelic expression of a rare allele in RNA-seq data using dedicated statistical methods. We describe how to generate and assess quality control plots and interpret the analysis results. The protocol is based on DROP (Detection of RNA Outliers Pipeline), a modular computational workflow that integrates all analysis steps, can leverage parallel computing infrastructures, and generates browsable web page reports.**

**Keywords:** RNA-seq, outlier detection, aberrant expression, aberrant splicing, mono-allelic expression, rare disorder, workflow

## Introduction

Whole-exome sequencing (WES) and whole-genome sequencing (WGS) are increasingly used to identify disease-causing genetic variants of individuals affected with a genetically undiagnosed rare disorder<sup>1-3</sup>. Depending on the disease entity, analysis of WES or WGS achieves a diagnosis rate of only up to 50%<sup>2-5</sup>. One of the main challenges in these DNA-based molecular diagnostics approaches is the interpretation of the non-coding variants. Many of these

variants may have a regulatory role in controlling RNA abundance or splicing. However, computational predictions of their regulatory effects from DNA sequence alone remain uncertain<sup>6</sup>. To advance diagnostics, RNA sequencing (RNA-seq) has emerged as a complementary tool because it directly probes gene expression, thereby providing functional data supporting the clinical interpretation of variants. Moreover, RNA-seq can reveal genes with a regulatory defect, even in the absence of candidate regulatory variants, which is particularly important in the situation where WES instead of WGS has been performed. Pilot studies systematically using RNA-seq have obtained increased diagnostic rates over DNA-sequencing alone for a variety of rare disorders ranging between 8% and 36%<sup>7–11</sup>.

As the first studies that systematically used RNA-seq for the genetic diagnosis of rare diseases were published only two years ago<sup>7,8</sup>, the field is new and lacks established workflows. Here, we provide a step-by-step protocol that instructs how to use RNA-seq data in this context by supporting the detection of aberrant events. Specifically, we provide detailed instructions to identify aberrant expression, aberrant splicing, and mono-allelic expression (MAE) of a rare allele by starting from Binary Sequence Alignment Map (BAM)<sup>12</sup> and Variant Call Format (VCF) files<sup>13</sup> (Table 1). The protocol covers a number of quality control steps and plots that evaluate the counting and fitting procedures of each pattern. In order to control sample mixing, the protocol includes a validation step comparing variants called from RNA and DNA. Lastly, we describe how to analyze individual samples using the results and objects derived from the pipeline.

**Table 1** | File formats

File format	Extension	Description
(Binary) sequence alignment map <sup>12</sup>	.bam	A binary compressed file containing sequencing read (typically from DNA or RNA-seq) alignments.
FASTA	.fa	A text file containing nucleotide sequences.
General Transfer Format	.gtf	A text file containing different gene annotations (e.g. genomic position, type of, transcripts, exons).
Serialized R data	.Rds (.rds, .RDS)	A binary compressed file containing a single R object.
Tab separated value	.tsv	A human-readable text file with columns separated by tabs.
Variant call	.vcf	A file containing sequence variants including their genomic

format <sup>13</sup>		positions, quality score, and other annotation.
----------------------	--	---

## Application of the protocol

The aim of this protocol is to provide a one-in-all computational workflow starting from VCF and BAM files to call and visualize all forms of aberrant gene expression that have been currently used for rare disease diagnosis purposes. It is based on DROP (Detection of RNA Outliers Pipeline), a computational workflow that automatizes and standardizes all the steps needed to obtain the results from raw input files. The workflow uses the Snakemake framework<sup>14</sup> (Box 1) to guarantee reproducible results. Overall, our workflow can be easily implemented and provides a comprehensive collection of tools enabling the detection of outliers using RNA-seq.

To showcase the workflow, we have created a dataset based on 100 samples from the Geuvadis dataset<sup>15</sup> (Table S1), which is accessible without restriction under <https://www.internationalgenome.org/data-portal/data-collection/geuvadis>. We refer to this dataset as the test dataset. DROP, including a small demo dataset of 10 samples and chromosome 21 only, is publicly available at <https://github.com/gagneurlab/drop>. All the plots, results and analyses of the test dataset can be found here: [https://i12g-gagneurweb.in.tum.de/public/paper/drop\\_analysis/webDir/html/drop\\_analysis\\_index.html](https://i12g-gagneurweb.in.tum.de/public/paper/drop_analysis/webDir/html/drop_analysis_index.html).

The DROP pipeline integrates recent tools leading to increased diagnostic rates. In our pilot study (Kremer et al.<sup>7</sup>), we found the *MCOLN1* gene to be borderline significant for aberrant splicing and expression, and the *CLPP* gene not to be significant for aberrant expression using our previous approaches (DESeq2<sup>16</sup> to detect aberrant expression and LeafCutter<sup>17</sup> to detect aberrant splicing). Re-analysing the same dataset with DROP, both of those cases are now called significant.

## Advantages and limitations

Currently, only one computational workflow for RNA-seq based diagnostics of rare disorders has been published<sup>9</sup>. That pipeline includes read alignment, read counting and quality controls of the counts, as well as functions to filter and annotate variants and prioritize candidates. The main advantages of using DROP is that we automatize most of the steps to detect rare RNA events. DROP integrates the specialized methods OUTRIDER<sup>18</sup> and FRASER<sup>19</sup> and adds a module to

test for MAE. Additionally, it provides quality control through dedicated plots to evaluate the different steps and VCF-BAM matching. DROP also overlaps candidate genes with phenotypes (encoded as HPO terms<sup>20</sup>). All the results and plots are rendered as HTML pages. DROP is designed with a modular architecture, such that the aberrant expression, aberrant splicing, and MAE modules can be executed without depending on each other. This architecture allows new modules with different functionalities to be added. Another advantage of DROP is that it uses the workflow management framework Snakemake<sup>14</sup> to run and monitor the execution of the pipeline (Box 1). By using Snakemake, DROP ensures that the results reflect the latest scripts and input data files, while avoiding unnecessary executions of scripts lying upstream or parallel to the modifications.

Another advantage of DROP is the way in which it manages analysis groups. Users might not want to merge all the available samples into one analysis, but instead create subsets. These subsets could contain, for example, samples extracted from the same tissue or belonging to the same disease entity. A sample can belong to one or multiple groups, which are specified in the sample annotation.. Each sample is processed only once on each module, thus saving space and computational resources. Then, the intermediate outputs are merged in their respective group(s), and a separate analysis is performed for each group.

RNA-seq is not the first tier diagnostic tool. Its power is in the detection of functional relevant consequences of DNA variants affecting gene expression levels and splicing. In this way it complements DNA sequencing by supporting interpretation of variants of uncertain significance and reprioritizing variants overlooked by DNA sequencing. RNA-seq has some inherent limitations in diagnostics (the highest reported diagnosis rate increases after WES using RNA-seq are 35%<sup>8</sup> and 36%<sup>10</sup>). For instance, the disease-causing variant(s) (e.g. missense) might not have any effect on the transcript. Another issue is that the gene is not expressed in the available tissue. In order to overcome this, researchers can investigate a priori whether genes are expressed in a certain tissue by using public datasets like GTEx<sup>21</sup> or the Expression Atlas<sup>22</sup>. Moreover, PAGE<sup>10</sup> can give insight on which tissues express the largest subset of a given set of genes. Regarding splicing, MAJIQ-CAT allows to test the similarity in splicing events of a group of genes in clinically-accessible tissues with respect to other tissue(s)<sup>23</sup>. Finally, fibroblasts can be transdifferentiated into other cell lines which better reflect the affected tissue of the studied disease, thus allowing to find aberrant events missed by the original fibroblasts<sup>10</sup>.

In order to confidently detect outliers, a large sample size is needed. The power analysis done in OUTRIDER concluded that the cohort should contain at least 50-60 samples<sup>18</sup>, and in FRASER at least 30 samples<sup>19</sup>. This limitation can be overcome by integrating control samples from publicly available cohorts like GEUVADIS<sup>15</sup> or GTEx<sup>21</sup> (See Dataset Design).

Assuming that local pipelines are established for read alignment and variant calling, DROP takes as input BAM and VCF files (Table 1) and does not provide a built-in solution for those steps. However, this protocol provides instructions for those steps as well (Materials). Furthermore, we implement only one method to compute the results of each type of aberrant expression pattern. Nevertheless, the user can modify the pipeline to apply any other method to them.

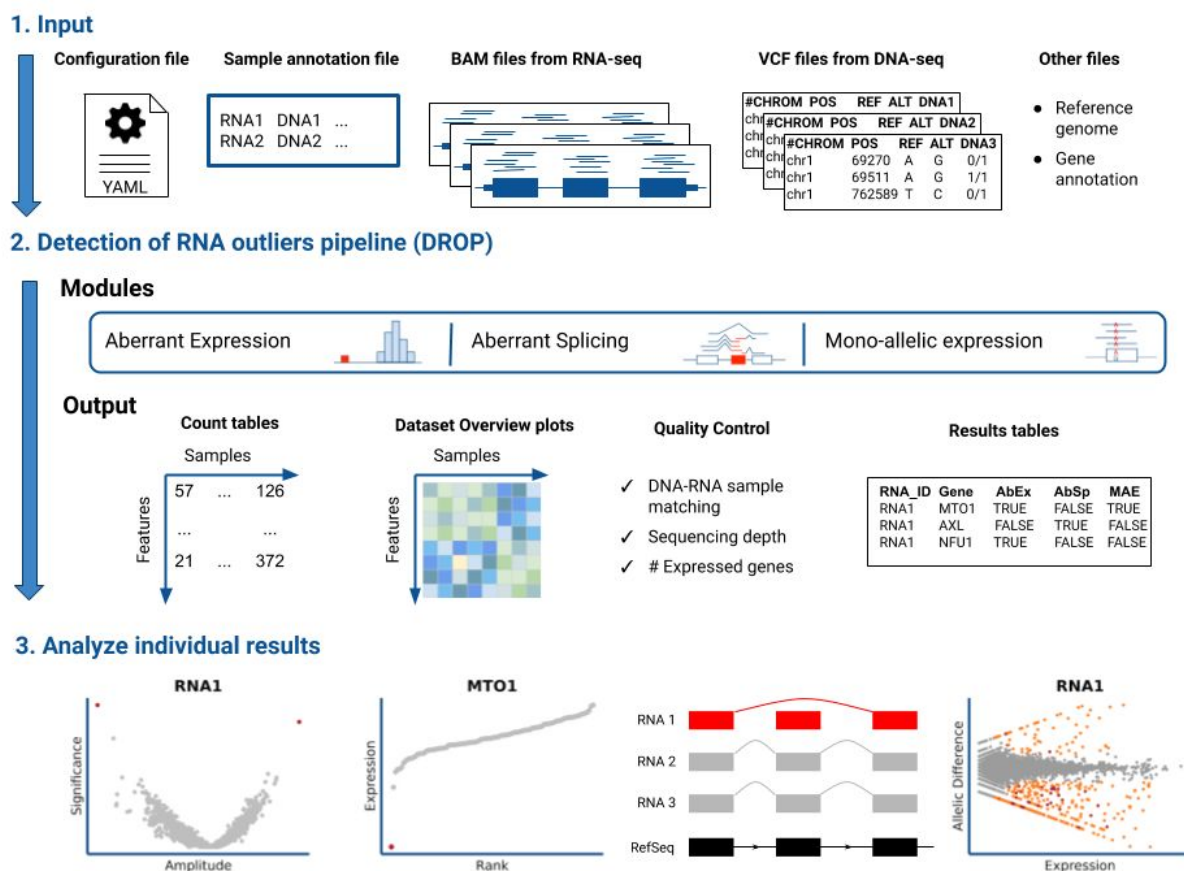
### Box 1. Workflow Management Systems used in DROP

**Snakemake** (<https://snakemake.readthedocs.io/en/stable>) is a workflow management system to create data analyses guaranteeing reproducibility, automation, and scalability<sup>36</sup>. Snakemake workflows consist of rules that describe how to create output files from the respective input files. These output files are created by defining instructions in shell, Python or R code. Rule dependencies are determined bottom-up by matching file names with the support of automatically inferred named wildcards in the rules. Every time the pipeline is executed, Snakemake computes the dependencies among all scripts and data files. Then, it checks if either a data file or a script were modified or added. If a script was added or modified, Snakemake will execute it, together with the downstream steps. If a data file was added or modified, Snakemake will execute any script using it, and all the downstream scripts in cascade. Moreover, Snakemake allows the usage of multiple scheduling systems or parallel backends (e.g., SGE, SLURM, multi core) to efficiently use HPC systems and run executions in parallel by automatically determining parallel parts in the workflow.

**wBuild** (<https://wbuild.readthedocs.io>) is a framework that automatically creates Snakemake dependencies, pipeline rules based on R markdown scripts and compiles the analysis results into a navigable HTML page. All information needed such as input, output, number of threads, and even Python code, is specified in a YAML header inside the R script file, thereby keeping code and dependencies together.

# Workflow

Our workflow is composed of three main steps: i) preparing the input data, ii) fitting the models and extracting the results from aberrant expression, aberrant splicing, and MAE, and iii) analyzing the individual results (Fig. 1). The input data are BAM files, VCF files, a sample annotation table, a configuration file containing the pipeline's parameters, a human reference genome (FASTA) file, and a gene annotation (gtf) file (Table 1). DROP then generates intermediate files (e.g., read count matrices), final results, and produces HTML pages for convenient visualization using the framework wBuild (Box 1). The generated files are stored as either tab-separated text (tsv) files or binary R data (Rds) files, depending on their content. See Table 1 for file format overview. Finally, users can access the objects and results in order to plot and analyze the samples and genes of interest.



**Fig. 1 | Workflow overview.** As input, DROP requires a configuration file, a sample annotation file, BAM files from RNA-seq, and VCF files. DROP processes then for each module the input data and generates count tables, overview plots (e.g. sample covariation heatmap), quality

control plots, and result tables. Lastly, users can perform case-by-case analysis with the help of different visualizations.

## Aberrant expression

Aberrantly expressed genes, or expression outliers, refer to genes with an expression level aberrantly higher or lower in a sample with respect to other samples. Calling expression outliers in individuals affected with a rare disorder has been successfully used to identify disease-causing genes and to identify or confirm potential disease-causing variants<sup>7,9,10</sup>.

To detect aberrant expression, we first compute an RNA-seq read count matrix by counting for each sample the sequencing reads that fully overlap each gene (Methods). Then, we apply OUTRIDER<sup>18</sup>, a specialized statistical method to detect expression outliers that computes significance level for extreme read count values while controlling for hidden confounders<sup>18</sup>. We extract *P* values, *z* scores and fold changes for each sample–gene combination from the OUTRIDER returned objects. After correcting for multiple testing, there is typically a handful of outlier genes per sample at a false discovery rate of 0.05. For example, there is a median of 1 outlier per sample in the test dataset, consistent with observations from other datasets<sup>7,18</sup>. If a sample has more than 0.1% of expressed genes as outliers, we refer to it as an aberrant sample. Pinpointing a candidate disease-causing genes among the tens to hundreds of expression outliers of an aberrant sample is difficult. Aberrant samples should be carefully analyzed to find a possible explanation for their high number of outliers, e.g., in case of contamination, or if the sample belongs to a different population (tissue, ethnicity). A biological cause is not to be excluded. For instance, disruption of the function of a transcription factor may cause aberrant expression of many downstream genes.

Other methods to detect aberrant expression consist of applying cutoffs on *z* scores<sup>9,10</sup> computed after regressing out contributions of hidden confounders estimated by PEER<sup>24</sup> or SVA<sup>9</sup>. However, simulations and enrichment analysis of rare variants among expression outliers have shown that OUTRIDER outperformed these methods<sup>18</sup>.

## Aberrant splicing

Aberrant splicing is a major cause of Mendelian disorders<sup>25,26</sup>. Despite progresses in predicting aberrant splicing from sequence<sup>27,28</sup>, the task remains difficult because splicing involves a complex set of cis-regulatory elements that are not yet fully understood. Moreover, some variants affecting splicing cannot be detected by WES for being located in intronic sequences

not covered by WES kit probes. Aberrant splicing events can instead be advantageously detected from RNA-seq data<sup>7–11,29–32</sup>.

The aberrant splicing module considers split reads, which are reads aligning to separate locations of the same chromosome and strand, and therefore supportive of an exon-exon junction. This is done independently of exon annotation, allowing for calling splicing events de novo. In addition, reads spanning the exon-intron boundary from both donor and acceptor sites are counted to enable intron retention detection. The counts are converted into two intron-centric metrics, percent-spliced-in (PSI,  $\psi$ ) and splicing efficiency ( $\theta$ ), which are calculated for both the donor  $D$  (5' splice site) and acceptor  $A$  (3' splice site) sites of every intron<sup>33</sup> (Methods). The  $\theta_5$  and  $\theta_3$  values are fitted together, hence we jointly refer to them as  $\theta$ . Junctions expressed at low levels are filtered out (Methods). Then, for each of the metrics  $\psi_5$ ,  $\psi_3$ , and  $\theta$ , we apply the specialized statistical method to detect splicing outliers, FRASER, which uses a denoising autoencoder to control for latent confounders and fits a beta-binomial distribution on every intron or splice site<sup>19</sup>.  $P$  values, z scores and  $\Delta\psi_5$ ,  $\Delta\psi_3$ , and  $\Delta\theta$  values are then computed from the beta-binomial fits, where the  $\Delta$  values are defined as the difference between the observed and the expected values. FRASER then reports gene-level  $P$  values as well as splice site-level  $P$  values. Similar to expression outliers, FRASER typically returns a handful of aberrant splicing events per sample that are significant at a FDR <0.10. For example, the median number of outlier events in the test dataset are 2 for  $\psi_5$ , 2 for  $\psi_3$ , and 5 for  $\theta$ .

Three other methods have been used to detect aberrant splicing, which are (i) an adaptation of the differential splicing test LeafCutter<sup>17</sup> used in the Kremer et al. study<sup>7</sup>, (ii) a cutoff based approach used in the studies of Cummings et al.<sup>8</sup> and Gonorazky et al.<sup>10</sup>, and (iii) a z score based method used in the Frésard et al. study<sup>9</sup>. The first one constructs intron clusters and tests for differential usage between one sample and all others, instead of aberrant splicing events. Moreover, it does not control for sample covariation. The second one defines aberrant splicing events as novel introns in genes with enough reads in the affected individual but not (or almost not) appearing in a control cohort, after applying local normalization. The two main caveats are that it depends on arbitrary cut-offs and may fail to recognize aberrant splicing events in weak splice sites<sup>34</sup>. The third one does correct for covariation, but uses a z score approach which does not offer any control for false discovery rate and can be inaccurate in splice sites with low reads. Having in mind these limitations, we have opted for FRASER<sup>19</sup>, which not only addresses all those issues, but also detects intron retention.



## Mono-allelic expression

MAE refers to the expression of a single allele out of the two alleles of a gene. When assuming a recessive mode of inheritance, single heterozygous rare variants are not prioritized after DNA sequencing. However, mono-allelic expression of such a single heterozygous rare variant in an affected individual, which could be due to genetic or epigenetic silencing of the other allele, is consistent with a recessive mode of inheritance. Therefore, detecting MAE of a rare variant has helped diagnosing rare disorders<sup>7,9,10,31,35,36</sup>. Rare disorders can also arise due to *de novo* mutations in haploinsufficient genes<sup>37,38</sup>, in which case MAE of either the reference or alternative allele can help highlight those genes.

Detecting mono-allelically expressed genes relies on counting the reads aligned to each allele at genomic positions of heterozygous variants. Several methods have been developed to detect MAE in the context of rare diseases, among which are the one described by Kremer et al.<sup>7</sup>, and more recently ANEVA-DOT<sup>35</sup>. On the one hand, Kremer et al. used a negative binomial test with a fixed dispersion for all genes. On the other hand, ANEVA-DOT implements a binomial-logit-normal test with gene-specific variance  $V^G$ , with the caveat that due to insufficient training data, estimates of  $V^G$  have been computed so far for only 4,962 genes (in median), depending on the tissue of interest<sup>35</sup>. As using ANEVA-DOT would result in losing more than half of the tested genes, we opted for the training data-independent negative binomial test (Methods). In the test dataset, we found 364 heterozygous SNVs to be mono-allelically expressed, 79 of which were toward the alternative allele, and 3 of which were rare (MAX AF  $\leq$  0.001).

## VCF-BAM matching

A crucial step when performing multi-omics is to ascertain that all assays performed on samples obtained from the same individual correspond. Therefore, we designed an algorithm to match the variants derived from DNA and RNA sequencing based on the ideas proposed by t' Hoen et al.<sup>39</sup> and Lee et al.<sup>40</sup> (Methods). We achieved this by comparing the BAM files from RNA-Seq with the VCF files from DNA sequencing at predefined genomic positions of variants that are not in linkage disequilibrium. The proportion of variants derived from the same individual that match in the DNA and RNA has to be significantly higher than the one from different individuals, but will not reach 100% due to MAE<sup>39</sup>. The algorithm is applied not only to the annotated VCF-BAM samples, but to all combinations in order to find other possible unannotated matches.

In the test dataset, the median of the proportion of matching genotypes of samples from the same individual is 0.98 compared to 0.58 in case of non matching samples. In the case of family members, we expect a value in-between these. All of the 100 RNA samples from the test dataset match their corresponding DNA, as expected since it is a quality-controlled public dataset. This procedure is part of the MAE module, but can also be run independently. We provide a VCF file containing the set of genomic positions to be tested in our resource web server.

## Detection of RNA Outliers Pipeline

To automatize and standardize the generation of the results, we developed the computational workflow DROP. DROP consists of three independent modules (Fig. 1). It is built using R and Python on top of the workflow manager frameworks Snakemake<sup>14</sup> and wBuild (Box 1). DROP offers a parallelized backend through Snakemake to execute the same processes on different samples at the same time, thus making effective use of computer clusters. It runs on the command line. Detailed instructions for installation, editing the input files and executing the pipeline can be found in the DROP documentation. The functions used throughout the workflow are explained in detail in the Methods section.

Here we used a subset of 100 samples from the Geuvadis dataset to test our pipeline and generate the plots. We expect many groups to implement and benefit from our workflow, as using RNA-seq for rare disease diagnosis is growing in popularity. Finally, as the source code is open source and the design of our pipeline is flexible, we encourage the community to expand the pipeline with new modules, e.g. by adding other multi-omics modules, prediction models, or calling variants on RNA-seq data.

## Dataset design

In general, samples originating from the same tissue and that were prepared and sequenced in a similar way should be analyzed as separate groups. Power analyses have suggested analyzing groups of at least 50 samples for aberrant expression<sup>18</sup> and at least 30 samples for aberrant splicing<sup>19</sup>. In this section we discuss whether it is advisable to combine samples from different cohorts, if the original sample size is smaller than the recommended values. When combining samples, they must be aligned using the same reference genome build, aligner and parameters, before processing them jointly with DROP.

One strategy is to include samples originating from the same tissue but from another cohort. Simulations detailed below indicate that this setting leads to an increased, yet manageable, list of reported outliers (less than 10 fold larger in these simulations), with no strong loss of sensitivity (less than 30%). We simulated the effect of merging RNA-seq data from one diagnostic lab with RNA-seq data with a public resource (GTEx). The diagnostic lab samples were the Kremer samples, which are derived from skin fibroblast cells<sup>7</sup>. We used the GTEx samples derived from skin not sun exposed as external samples. The samples from both cohorts were sequenced non-strand-specifically and aligned to the hg19 genome build in the original studies. To investigate the effect on calling expression outliers, we simulated 30 heterogeneous datasets, each with sample size equal to the one of the original Kremer dataset (n=119). Each heterogeneous dataset consisted of 102 randomly picked samples from GTEx and the 17 samples of the Kremer dataset with a confirmed pathogenic protein truncation variant that leads to aberrant expression (probably through nonsense-mediated decay). OUTRIDER could not correct these 30 simulated heterogeneous datasets as effectively as it could correct the original Kremer dataset. Suboptimal OUTRIDER correction is evident from larger correlation values and the 17 Kremer samples clustering together after correction (heatmaps, Fig. S1), as well as larger deviations between observed read counts and OUTRIDER estimates (larger post-correction biological coefficient of variation, Fig. S2a). Moreover, the number of outliers per sample of the 17 Kremer samples increased by around 8 fold in the heterogeneous setting (median of 23 outliers per sample) compared to the original setting (median of 3 outliers per sample) using the recommended OUTRIDER FDR cutoff of 0.05. Importantly, more than 70% of those 17 pathogenic outliers were detected on median (Fig. S2b,c). With an FDR cutoff of 0.3, more than 80% of the 17 pathogenic outliers are detected, but at a cost of reporting 67 outliers in median per sample. Heterogeneous datasets were also simulated for investigating aberrant splicing calling, using the 13 Kremer samples with a confirmed pathogenic splicing defect. More than 75% of those 13 splicing pathogenic outliers were detected in the heterogeneous datasets, at a cost of obtaining 34 outliers in median per sample, instead of 14 in the Kremer dataset, using the recommended FRASER FDR cutoff of 0.1 (Fig. S2d,e). Loosening the FDR cutoff to 0.5 did not recover more true positives (Fig S2e). Altogether, this analysis indicates that combining small cohorts with samples from public resources appears to recover pathogenic aberrant expression events at enough sensitivity and specificity to be useful for diagnostics.

Expression and splicing patterns are known to differ across tissues<sup>19,21</sup>. In order to test the effect of combining samples from different tissues, we combined 100 GTEx samples from whole blood with 100 samples from either skin not sun exposed, skeletal muscle, cerebellum (brain) or liver. Even though the number of expression and splicing outliers did not increase when combining blood with other tissues with respect to blood alone, only ~50% of expression outliers (and 30% of splicing outliers) found in blood were recovered in the combined datasets (Fig. S3). Therefore, we conclude that it is better not to merge samples from different tissues.

Sequencing costs can be reduced by higher multiplexing, yielding a lower sequencing depth per sample. To investigate the effect of sequencing depth, we downsampled reads from the 17 expression true positives (and 13 splicing true positives) to sum to a total sequencing depth of ~30 million reads and merged them with the rest of the Kremer dataset which has a median depth of ~86 million reads (Fig. S4a). At a lower depth, less expression and splicing outliers per sample were detected, retrieving 88% of the 17 pathogenic expression outliers, but only 46% of the 13 pathogenic splicing outliers (Fig. S4b-d). Calling splicing outliers relies on split reads which requires a higher sequence depth than calling expression outliers.

We do not recommend merging strand- with non-strand-specific samples. Reads that overlap two genes lying on different strands will be assigned to both genes if the sequencing was done non-strand-specifically, while only to the correct gene if it was done strand-specifically.

Diagnostic labs may encounter further situations we could not investigate (e.g., merging polyA selection with ribo-depleted RNA samples, or FFPE tissues with fresh frozen tissues). Generally, after running OUTRIDER and FRASER, we recommend investigating the sample correlation heatmaps (Fig. S1). If the autoencoder was able to successfully remove the covariation, they should look similar to figures 2d, S1b and S1f. On the contrary, if the autoencoder fails, the heatmaps will look like figures S1d and S1h.

## Other uses of RNA-seq in diagnostics of rare diseases

Besides the three strategies already mentioned, RNA-seq data provide other valuable information. First, variants can be called from BAM files derived from RNA-seq. This has limited value when WGS is available, but is useful in the situation where only WES is available by better covering the UTRs and revealing deep intronic variants of aberrantly-created exons<sup>8,10</sup>. Second, variant phasing can be done using RNA-seq data. Due to splicing, variants can be phased over longer distances than WES or WGS. Phasing is useful in the clinical setting by

allowing to distinguish between compound heterozygotes and variants on the same allele<sup>40</sup>. Third, gene fusion can be detected from transcriptome. Gene fusion happens when, due to chromosomal translocation, inversion, deletion, or duplication, genetic material from different genes are merged and transcribed together. Even though gene fusion has been used more extensively in cancer<sup>41,42</sup>, it has also been successfully applied to diagnose patients with congenital<sup>43</sup> and rare diseases<sup>44</sup>.

Pointers to specialized programs and pipelines for these tasks are provided in the Procedure section and in the Online Resources section. Integration of these and other tools as DROP modules could be considered in the future.

## Expertise needed

The installation, creation of a Python environment and set up of the config file should be done preferably by either a bioinformatician or a system administrator. Afterwards, the protocol can be followed by any scientist.

## Materials

### BAM files from RNA-seq data

This workflow focuses on the analysis of the sequencing data rather than on their generation and preprocessing. In this subsection we explain how the BAM files (Table 1) should be generated to serve as input for DROP. BAM files are created by aligning FASTA files derived from RNA-seq to a reference genome. We recommend using the aligner STAR<sup>36</sup> with the default parameters and `twopassMode = 'Basic'` to detect novel splice junctions. We strongly recommend performing quality control with tools like MultiQC<sup>41</sup>. The BAM files contain reads that will be used in all of the modules to generate the read count matrices. They do not need to be in the same folder, as the location is specified in the sample annotation table. The BAM files must be sorted by position and indexed. This can be achieved using the commands `sort -o` and `index` from SAMtools<sup>34</sup>. Under default settings, DROP does not require the BAM files to contain any particular tag or field. However, if specific options are included in the `ScanBamParam` parameter in the config file, then the BAM files must include the required tags or fields (refer to DROP's documentation).

## VCF files from either WES or WGS

VCF files (Table 1) are generated through calling variants on a BAM file. Therefore, the FASTA files derived from DNA sequencing must be aligned to a reference genome first (one possible tool is the Burrows-Wheeler Aligner<sup>39</sup>). BAM files must be sorted by position and indexed. Afterwards, variants can be called by following, for example, the GATK guidelines<sup>43,44</sup>. This will generate either one VCF file per DNA sample, or a single VCF file containing multiple samples. DROP can handle either option. VCF files must be compressed and indexed. This can be achieved by using the `bgzip -c` and `tabix -p` commands from Tabix<sup>45</sup>, respectively. They do not need to be stored in the same folder, as the location is specified in the sample annotation table. We further recommend annotating the VCF file(s) with the Variant Effect Predictor<sup>32</sup> following their tutorial (Online resources). This will add information such as variant consequences (e.g. missense, stop-gained) and allele frequencies. The genome build used to align the files derived from DNA sequencing must be the same as the one used to align the files derived from RNA sequencing.

## Human reference genome file

In order to compute the allelic counts, a human reference genome (FASTA) file is needed. Preferably, the file should be the same that was used to align the samples FASTA files. Otherwise, the user can download it from different sources, as long as the genome build (either GRCh37 (hg19), or GRCh38 (hg38)) matches the BAM files. For example, for build 19 <https://hgdownload.soe.ucsc.edu/goldenPath/hg19/bigZips/>, hg19.fa.gz; and for 38 <https://hgdownload.soe.ucsc.edu/goldenPath/hg38/bigZips/latest/>, hg38.fa.gz, from UCSC<sup>46</sup>. Finally, an index (.fai) file must be created in the same directory where the FASTA file is located using the function `faidx` from SAMtools<sup>12</sup>.

## Gene annotation file

In order to count the reads from the BAM files in the aberrant expression module, a gene annotation file is required. DROP was developed using the 'main annotation' gtf file from the release 29 of the GENCODE annotation<sup>47</sup>. Nevertheless, newer releases already exist and will continue to appear. As gtf files from other sources can have other columns and with different names, DROP might not be able to parse them. Therefore, to avoid problems, we recommend

users to download the latest release from GENCODE, with the right genome build (<https://www.gencodegenes.org/human/>).

## Hardware

A multi-core computer with at least 16 cores and 64 GB RAM is suggested, but depends heavily on the analyzed dataset (e.g., sample size). For developing and testing DROP on the 100 samples from the Geuvadis dataset, a Scientific Linux 7.7 based server with 32 physical cores and 256 GB memory was used.

## Software

- A Linux based operating system
- DROP (<https://github.com/gagneurlab/drop>)
- Python, version 3.6.7 or higher
  - We recommend using a virtual environment such as conda (<https://docs.conda.io/>) for making sure that the correct versions are installed
  - Pip version 19.1 or higher
- R (<https://www.r-project.org/>), version 3.5 or higher
  - Bioconductor (<https://bioconda.github.io/>) and devtools need to be installed. Please make sure that the library paths are writable.
  - The required packages are listed in the DROP repository and an installation script is provided.
  - An integrated development environment (IDE) like RStudio (<https://www.rstudio.com/products/rstudio/download/>) is recommended.
- GATK<sup>43</sup> (<https://software.broadinstitute.org/gatk/>).
- SAMtools<sup>12</sup> version 1.7 or higher and Tabix<sup>45</sup> can be both installed from HTSLib. (<https://www.htslib.org/download/>).
- BCFtools<sup>48</sup>. Use the latest version from its github page (<https://github.com/samtools/bcftools>).
- Integrative Genomics Viewer (IGV)<sup>49</sup> (<https://software.broadinstitute.org/software/igv/>)
- Graphviz (<https://www.graphviz.org/>)
- Pandoc (<https://pandoc.org/>)

# Procedure

## Configuration of the workflow

1. Define a directory where the analysis code will be. Go to that directory and open a terminal. Set up a new DROP-based project by executing `drop init`. This will create a config file, a Snakefile and a Scripts directory for analysis that can be modified by the user.
2. Adjust the keys of the config.yaml file, which follows the YAML format<sup>50</sup>. All non-file keys contain default values that can but do not have to be changed. A detailed description of parameters and file names can be found in the DROP documentation.
3. Create the sample annotation table. Each row corresponds to a unique pair of RNA and DNA samples derived from the same individual. An RNA assay can belong to one or more DNA assays, and vice-versa. If so, they must be specified in different rows. The required columns are RNA\_ID, RNA\_BAM\_FILE and DROP\_GROUP, plus other module-specific ones. Save it in the tab-separated values format (Table 1). The column order does not matter. Also, it does not matter where it is stored, as the path is specified in the config file. An example is shown in Table S1. Further instructions and examples can be found in the DROP documentation. ?TROUBLESHOOT
  - RNA\_ID: unique identifier from an RNA assay.
  - RNA\_BAM\_FILE: absolute path of the BAM file derived from RNA-seq (e.g. /home/project1/Data/RNA1.bam, and not RNA1.bam). A BAM file can only belong to one RNA\_ID, and vice-versa.
  - DROP\_GROUP: the analysis group(s) that the RNA assay belongs to. Multiple groups must be separated by commas (e.g., blood,WES,all). We recommend doing a different analysis for each tissue as gene expression and splicing can be tissue-specific.

MAE module specific columns:

- DNA\_ID: unique identifier from a DNA assay.
- DNA\_VCF\_FILE: absolute path to the corresponding VCF file (e.g., /home/project1/Data/sample1.vcf.gz, and not sample1.vcf). The DNA\_ID has to



match the ID inside the VCF file. In case a multi-sample VCF is used, write the file name for each sample.

The following columns describe the RNA-seq experimental set-up. They affect the counting procedures of the aberrant expression and splicing modules. For a detailed explanation, refer to the documentation of HTSeq<sup>51</sup>.

- **PAIRED\_END**: either TRUE or FALSE, depending on whether the sample comes from paired-end RNA-seq or not.
- **STRAND**: either yes, no or reverse: “no” means that the sequencing was not strand-specific; “yes” that it was strand-specific, and the first read in the pair is on the same strand as the feature and the second read on the opposite strand; and “reverse” that these rules are reversed.

The following columns describe how the counting should be performed in the aberrant expression module. We recommend setting the count mode to IntersectionStrict to consider only reads that fully align within the gene, and the count overlaps to TRUE to detect as many genes as possible. Refer to the documentation of HTSeq<sup>51</sup> for details.

- **COUNT\_MODE**: either “Union”, “IntersectionStrict” or “IntersectionNotEmpty”.
- **COUNT\_OVERLAPS**: either TRUE or FALSE, depending on whether reads overlapping different regions are allowed and counted.

The following column is optional:

- **HPO\_TERMS**: comma-separated phenotypes encoded as HPO (Human Phenotype Ontology) terms (e.g., HP:0001479,HP:0005591).

4. Execute `snakemake sampleAnnotation`. The command will parse and check the provided sample annotation, and create the annotation overview webpage (Table 2). The annotation overview contains information regarding the sample annotation and different sanity checks. Go through it carefully and, if detected, correct any errors in the sample annotation file. Check if the number of samples of each DROP group are correct. Also, if values were provided in the HPO\_TERMS column, this command will create a text file containing all the genes that overlap the HPO terms of each sample (Table 2).
5. Execute `snakemake -n`. It will display all the jobs that must be performed. The full pipeline can already be run by executing `snakemake <options>`. Nevertheless, we recommend to run it by modules. For a detailed explanation of the “options”, check the DROP documentation.

## The aberrant expression module

6. To run the aberrant expression module execute `snakemake <options> aberrantExpression`. This step is computationally heavy and can take several hours. This command counts the reads for each sample (Methods), performs the OUTRIDER fit, extracts the results, and creates two HTML reports with different plots for counting and results (Fig. S5a). This is performed for each annotation and group combination defined in the `geneAnnotation` parameter (config file) and in the `DROP_GROUP` column (annotation file), respectively. Different intermediate and final objects are saved locally (Table 2) and can be used for further analyses.

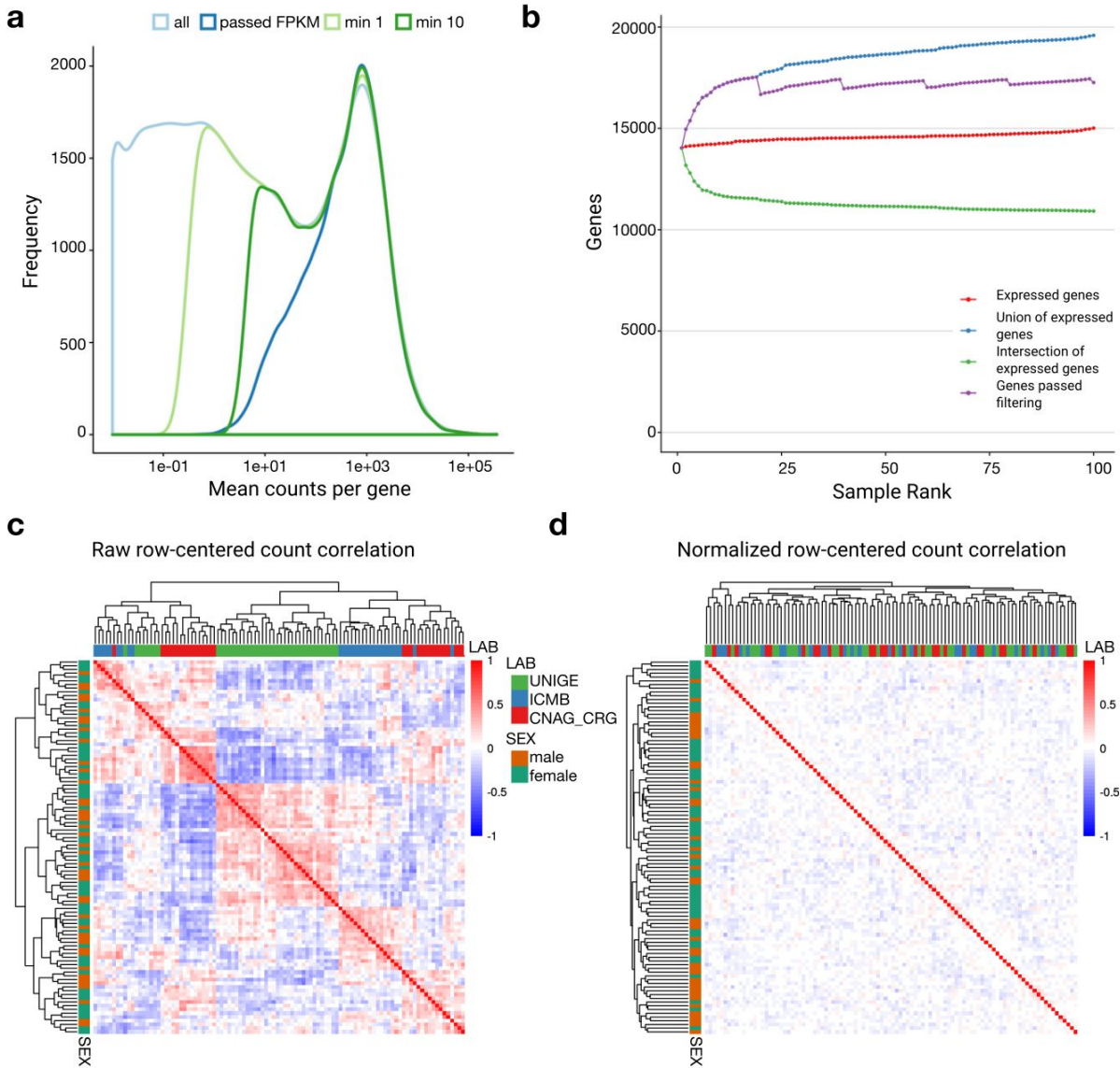
**Table 2** | Files created by DROP and their locations. The variables `<root>`, `<htmlOutputPath>`, and `<geneAnnotation>` are defined in the config file. The variable `<group>` corresponds to the `DROP_GROUP` column from the sample annotation.

File	Module	Path
Sample annotation overview	---	<code>&lt;htmlOutputPath&gt;/Scripts_Pipeline_SampleAnnotation.html</code>
Analysis scripts	all	<code>&lt;project&gt;/Scripts/</code>
Merged counts table	Ab. Exp.	<code>&lt;root&gt;/processed_data/aberrant_expression/&lt;geneAnnotation&gt;/outrider/&lt;group&gt;/total_counts.tsv</code>
OUTRIDER full results table	Ab. Exp.	<code>&lt;root&gt;/processed_results/aberrant_expression/&lt;geneAnnotation&gt;/outrider/&lt;group&gt;/OUTRIDER_results_all.Rds</code>
OUTRIDER significant-only results table	Ab. Exp.	<code>&lt;root&gt;/processed_results/aberrant_expression/&lt;geneAnnotation&gt;/outrider/&lt;group&gt;/OUTRIDER_results.tsv</code>
Fitted OUTRIDER dataset (ods) object	Ab. Exp.	<code>&lt;root&gt;/processed_results/aberrant_expression/&lt;geneAnnotation&gt;/outrider/&lt;group&gt;/ods.Rds</code>
Aberrant expression module webpage	Ab. Exp.	<code>&lt;htmlOutputPath&gt;/aberrant-expression-pipeline_index.html</code>
Aberrant expression analysis webpage	Ab. Exp.	<code>&lt;htmlOutputPath&gt;/Scripts_AberrantExpressionAnalysis_Overview.html</code>

Fitted FRASER dataset (fds) object	Ab. Spl.	<root>/<processed_data>/aberrant_splicing/datasets/savedObjects/<group>/fds-object.RDS
FRASER significant-only results table	Ab. Spl.	<root>/<processed_data>/aberrant_splicing/results/{dataset}_results.tsv
Aberrant splicing module webpage	Ab. Spl.	<htmlOutputPath>/aberrant-splicing-pipeline_index.html
Aberrant splicing analysis webpage	Ab. Spl.	<htmlOutputPath>/Scripts_AberrantSplicingAnalysis_Overview.html
MAE individual results	MAE	<root>/processed_results/mae/samples{DNA_ID}--{RNA_ID}_res.Rds
MAE full results table	MAE	<root>/processed_results/mae/<group>/MAE_results_all_<geneAnnotation>.tsv.gz
MAE significant-only results table	MAE	<root>/processed_results/mae/<group>/MAE_results_<geneAnnotation>.tsv
MAE module webpage	MAE	<htmlOutputPath>/mae-pipeline_index.html
MAE analysis webpage	MAE	<htmlOutputPath>/Scripts_MAEAnalysis_Overview.html

7. To look at the counting outcome, open the aberrant expression module webpage (Table 2). Go to the Counting tab. It contains one link per analysis group. Click on the analysis group you want to visualize. It contains the following quality control plots:
  - a. Number of reads counted per sample.
  - b. Percentage of reads counted per sample (Fig. S5b).
  - c. Size factors per sample (Fig. S5c).
  - d. Size factors vs read count ratio.
  - e. Raw read count distribution (Fig. 2a).
  - f. Number of expressed genes (Fig. 2b). A total of 17,259 genes passed the filter in the test dataset.

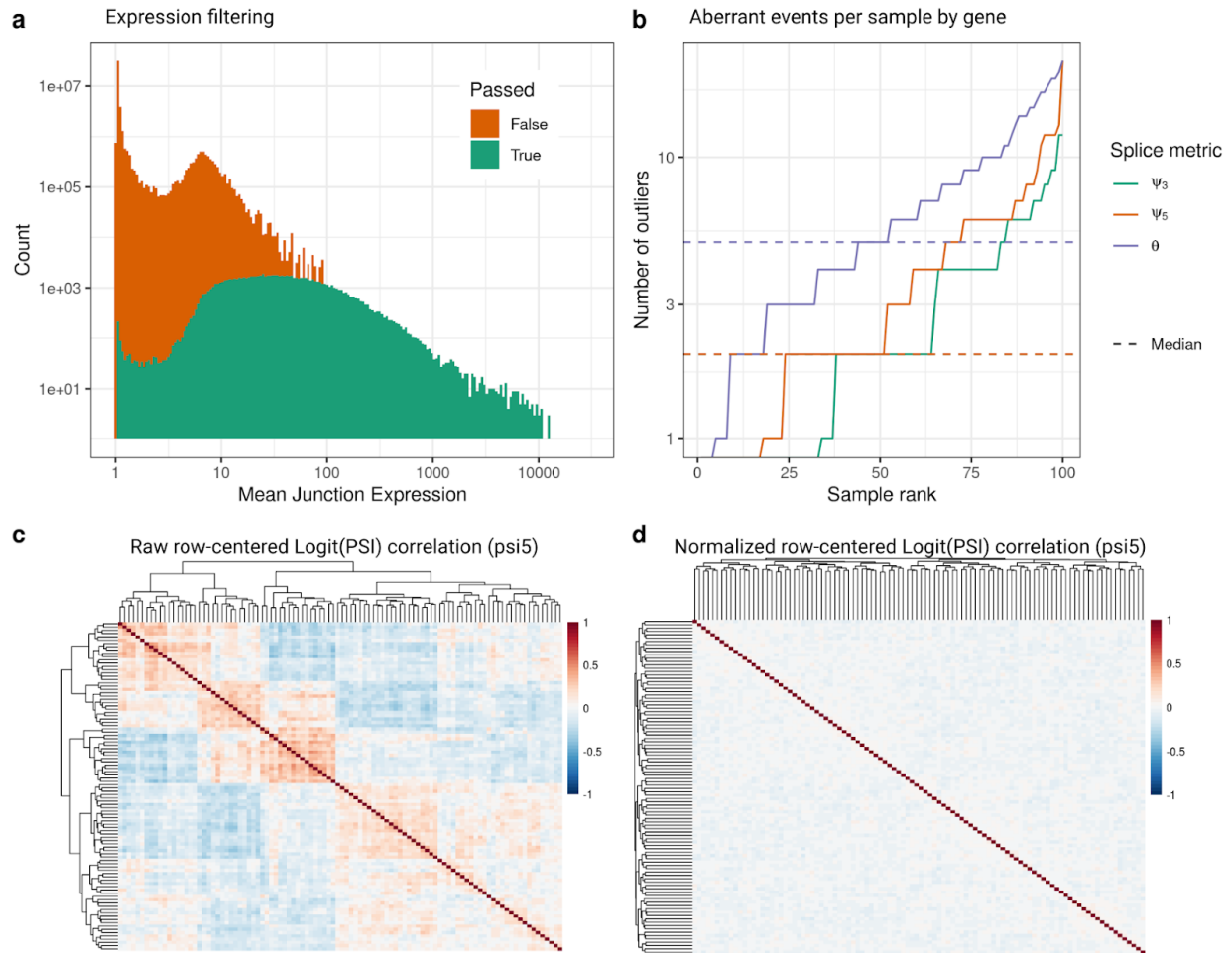
8. To investigate the results, open the aberrant expression module webpage. Go to the OUTRIDER tab. It contains one link per analysis group. Click on the analysis group you want to visualize. It contains the following:
  - a. Fit of the encoding dimension plot (Fig. S5d).
  - b. Number of aberrant genes per sample plot (Fig. S5e).
  - c. Heatmaps showing the sample correlation before (Fig. 2c) and after the correction (Fig. 2d). The clustering should disappear after the correction.
  - d. Heatmaps showing the log row-centered gene expression of the 50 most variable genes before and after the correction. This is useful to identify the genes that drive the clustering of samples.
  - e. Gene-wise biological coefficient of variation<sup>52</sup> plot before and after the correction.
  - f. Table with aberrant samples.
  - g. Searchable results table with overlapping HPO terms (if specified) and links to download both the full and subsetted results tables.



**Fig. 2 | The Aberrant Expression Module.** **a**, Histogram of raw read count distribution colored by different filtering strategies: all, minimum 1 count in 5% of the samples, minimum 10 counts in 5% of the samples, and minimum 1 (or user-defined) FPKM in 5% of the samples. **b**, Number of expressed genes cumulative across all samples. Colors represent the union of all detected genes (blue), genes that passed the FPKM filter as a group (violet), genes that are expressed on each sample (red), and the intersection of expressed genes (green). **c**, Heatmap of the correlation of row-centered log-transformed read counts between samples before and after **(d)** the autoencoder correction including the samples' sexes and the laboratories where they were sequenced. Before correction, samples cluster by laboratory.

## The aberrant splicing module

9. To run the aberrant splicing module execute `snakemake <options> aberrantSplicing`. This step is also computationally heavy and can take several hours. It creates a FRASER dataset object that contains all the intron-centric metrics and results, an HTML page with overview plots, and the splicing outlier results table (Fig. S6a).
10. Open the aberrant splicing module webpage. Go to the Counting tab. Click on the analysis group you want to visualize. It contains the following:
  - a. Introns and splice sites that passed the filter
  - b. Filter expression plot (Fig. 3a)
  - c. Filter variability plot
11. Open the aberrant splicing module overview webpage. Go to the FRASER tab. Click on the analysis group you want to visualize. It contains the following:
  - a. Finding best encoding dimension plot for  $\psi_3$  (Fig. S6b),  $\psi_5$  (Fig. S6c), and  $\theta$ .
  - b. Number of aberrant  $\psi_3$ ,  $\psi_5$  and  $\theta$  events per sample plot (Fig. 3b).
  - c. Heatmaps (sample vs. sample) before (Fig. 3c) and after correction (Fig. 3d).
  - d. Searchable results table for all metrics combined aggregated by gene, with overlapping HPO terms (if specified).



**Fig. 3 | The aberrant splicing module.** **a**, Histogram showing the junctions that passed the filter (Methods). **b**, Number of aberrant splicing events per sample by gene colored by metric. **c**, Heatmap of the correlation of row-centered logit-transformed read count ratios between samples before and after **(d)** the correction.

## Mono-allelic Expression Module

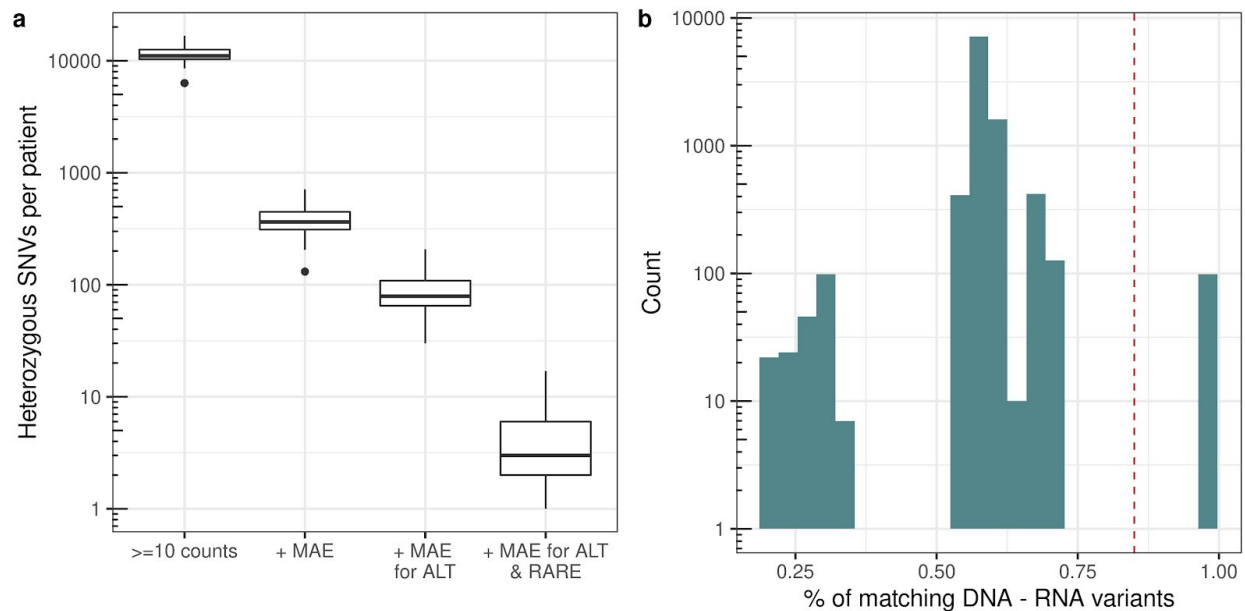
12. To run the MAE module execute `snakemake <options> mae`. This creates the allelic counts, aggregated results table (Methods) and the overview webpage (Fig. S7). The results for each sample contain the counts, the nominal and the adjusted  $P$  values, and the alternative allele ratio ( $\text{altRatio} = \#ALT / (\#ALT + \#REF)$ ) of each variant. Moreover, the results are annotated with the minor allele frequencies from gnomAD<sup>53</sup> (for each of the African, American, East Asian and Non-Finnish European populations, and the maximal frequency among them), and a boolean column “rare” if the allele frequency was below the corresponding cut-off specified in the config file. This step is computationally heavy and can take several hours.

13. Open the MAE overview webpage (Table 2). Go to the MAE tab. It contains different summarizing information:
  - a. Cascade plot (Fig. 4a) showing the number of heterozygous SNVs for the four cumulative filter criteria: at least 10 reads, mono-allelically expressed for either the reference or alternative allele, for the alternative only, and rare.
  - b. Searchable results table containing only the significant events with overlapping HPO terms (if specified). To subset for rare variants, simply filter the “rare” column.
  - c. Links to download full and significant-only results tables.

## VCF - BAM matching

14. Download the test VCF file (qc\_vcf\_1000G.vcf.gz) and its index file from the DROP resource webpage (Online resources). Its location is specified in the config file.
15. To run the VCF - BAM file matching, execute `snakemake <options> sampleQC`.  
This step is computationally heavy and can take several hours.
16. Open the MAE overview webpage (Table 2). Go to the QC tab. It contains:
  - a. Histogram of the percentage of matching genotypes between DNA and RNA (Fig. 4b).
  - b. Table containing the samples that were annotated to match, but actually do not (empty if there are no mismatches).
  - c. Table containing the samples that were not annotated to match, but actually do (empty if there are no such cases).





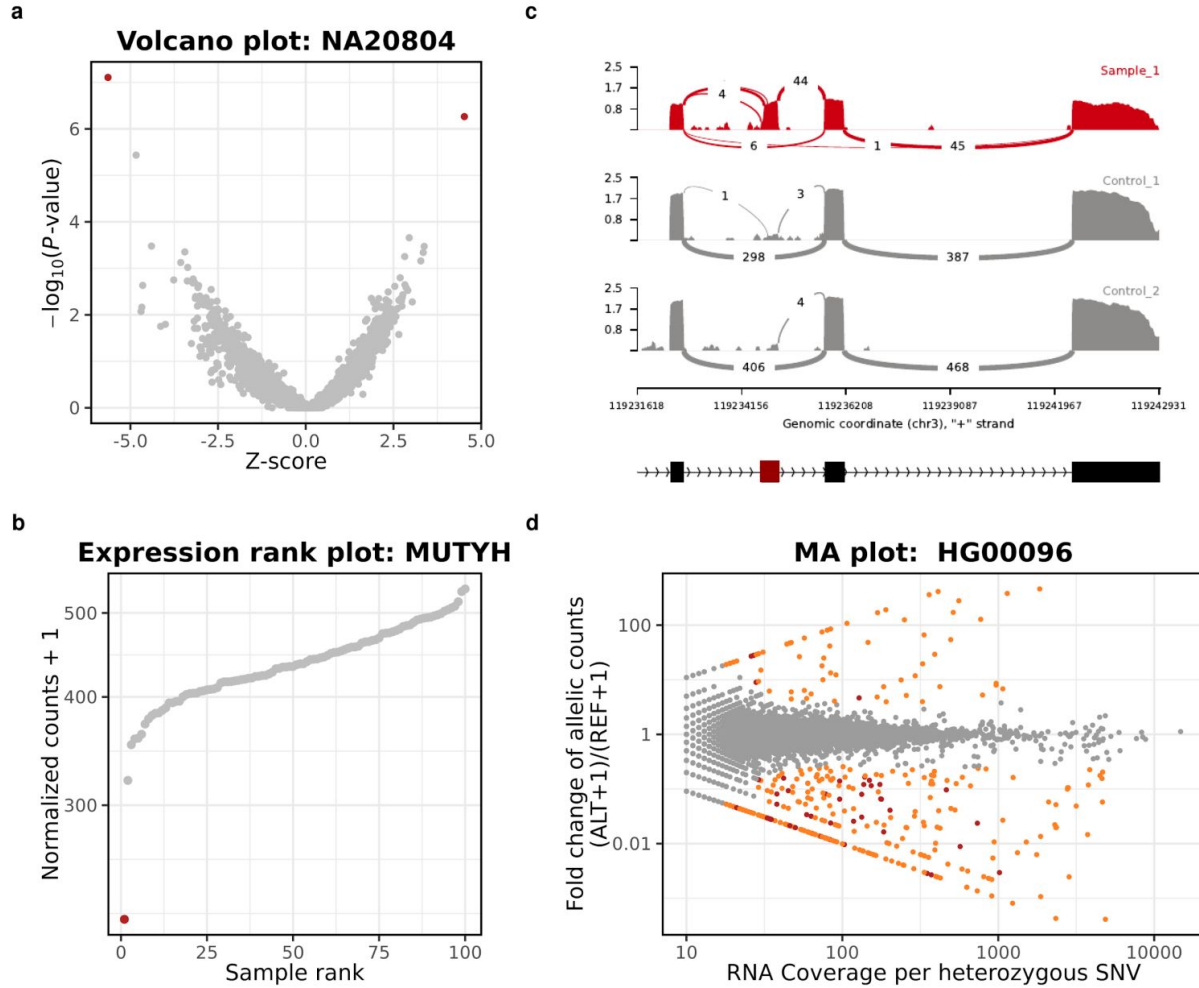
**Fig. 4 | Mono-allelic Expression Module.** **a**, Cascade plot showing the number of heterozygous SNVs with at least 10 reads (median=11,532), that are mono-allelically expressed (median=363), mono-allelically expressed toward the alternative allele (median=77) and rare (median=3). The lower and upper limits of the boxes correspond to the first and third quartiles, respectively. Whiskers extend to the most extreme value, no further than 1.5x the interquartile range. **b**, Histogram (y-axis in log scale) of the proportion of matching genotypes from DNA and RNA when comparing all sample combinations. The median of matching samples is 0.98 compared to 0.58 of non-matching samples. One sample had a very low rate of matching with the rest of samples (median=0.26).

## Exploration of outlier results

17. Execute `snakemake <options>`. This will run all the analysis scripts and create a webpage for each of them with case-specific plots.
18. Open the aberrant expression analysis webpage (Table 2). It contains the following:
  - a. Links to the overview webpages, and the paths of the OUTRIDER dataset objects and results tables.
  - b. Volcano plot ( $-\log_{10}(P \text{ value})$  vs. z score) of the first sample from the results table. Figure. 5a shows a typical scenario in which the expression of most of the genes of a sample does not significantly deviate from the rest of the population with the exception of a handful of genes clearly distinguishable from the other ones.
  - c. Expression plot (normalized counts across all samples) of the first gene from the results table (Fig. 5b). This plot not only shows if the gene of interest is aberrantly expressed in a sample, but also the magnitude. In this case, the expression of

gene *MUTYH* lies in a range of 350-500 counts for all samples except for one, where it is around 200 counts, thus implying a ~50% reduction.

- d. Expected vs. observed counts plot of the same gene across all samples (Fig. S8a). This plot emphasizes the need to correct the raw reads, as samples with low reads can anyway have a low expected value after the OUTRIDER fit.



**Fig. 5 | Analysis plots.** **a**, Negative log-transformed nominal  $P$  values (y-axis) versus z scores (x-axis) derived from the expression of all genes of sample NA20804 showing one over- and one underexpression outlier (red). **b**, Normalized counts (y-axis) of gene *MUTYH* of all samples (x-axis) reflecting one underexpression outlier (red). **c**, Representative sashimi plot showing the creation of a new exon for Sample 1 which is skipped by the other two samples. The height represents the coverage in  $\log_{10}$  RPKM and the number of the arcs the junction's coverage. On the bottom, the corresponding gene model is depicted (in red the cryptic exon). **d**, Fold change between the counts of the alternative and reference alleles (y-axis) compared against the total coverage of each heterozygous SNV (with at least 10 reads) of sample HG00096, highlighted by significance (orange), and significance and rarity (red).

19. Open the aberrant expression analysis script (Table 2) in your preferred editor (e.g., RStudio). It contains the code that generated the analysis webpage. Modify the code by specifying a sample and gene of interest and recreate the plots from the previous step by executing the different lines of code. Alternatively, save the script and execute `snakemake Scripts_AberrantExpressionAnalysis_Overview_R` to generate an updated analysis webpage.
20. Open the aberrant splicing analysis webpage (Table 2). It contains the following:
  - a. Links to the overview webpages, and the paths of the FRASER dataset objects and results tables.
  - b. Volcano plot ( $-\log_{10}(P \text{ value})$  vs.  $\Delta\psi$ ) of the first sample from the results table (Fig. S8b).
  - c. Expression plot (junction count versus total junction coverage) for  $\psi_3$  of the first junction from the results table across all samples (Fig. S8c). In this example, for most of the samples, the count of the junction was the same as the total count of the corresponding acceptor. Nevertheless, in one sample, the total count for the acceptor is around 3 times higher than the tested junction. This means that most of its spliced reads come from another donor not present in any other sample.
  - d. Observed vs. predicted counts plot for  $\psi_3$  of the same junction across all samples (Fig. S8d).
  - e. Quantile-quantile plot of observed vs. expected  $P$  values for  $\psi_3$  of the same junction to evaluate the fit (Fig. S8e).
21. Open the aberrant splicing analysis script (Table 2) in RStudio. Once again, go through the different lines of code and modify them to specify the sample and junction of interest. Recreate the plots. Run `snakemake Scripts_AberrantSplicingAnalysis_Overview_R` to display the plots in the analysis webpage. These plots indicate if an event was an outlier by visualizing all samples. Nevertheless, sashimi plots provide a better visualization of the nature of the splicing aberration.
22. To create a sashimi plot, load the BAM file of the sample of interest and also other BAM files that do not have this splicing outlier in IGV<sup>49</sup> (File > Load from File, then select the BAM file). Go to the genomic position of the junction of interest obtained from the FRASER results table. Adjust the zoom to cover the entire region of interest and create a

sashimi plot by doing right click > Sashimi Plot. The bars represent the coverage for each alignment track, and the arcs the splice junctions connecting exons with the number of reads split across the junction. Therefore, sashimi plots can be used for both qualitative and quantitative analysis of transcripts. High-quality sashimi plots (Fig. 5c) can be achieved using the MISO package<sup>51</sup>. We provide an example script, config and gff file needed to generate a MISO sashimi plot in the DROP github webpage.

23. Open the MAE analysis webpage (Table 2). It contains the following:
  - a. Links to the overview webpages, and the paths of the results per sample.
  - b. MA plot (fold change vs. RNA coverage) of the first sample from the results table, colored by significance and rarity (Fig. 5d). As expected, a lower proportion of variants are mono-allelically expressed towards the alternative than towards the reference, and from them only a handful are rare.
  - c. Alternative vs. reference allele coverage plot of the same sample (Fig. S8f), which gives a similar overview as the previous plot.
24. Open the MAE analysis script (Table 2) in RStudio. Once again, go through the different lines of code and modify them to specify the sample of interest. Note that the format of the identifier is {RNA\_ID}--{DNA\_ID}. Recreate the plots. Run `snakemake Scripts_MAEAnalysis_Overview_R` to display the plots in the analysis webpage.
25. Inspect the mono-allelically expressed variants in IGV. Load the BAM file of interest in IGV. If available, load the corresponding WES or WGS BAM file, and other BAM files as controls. Go to the position of the mono-allelically expressed variant (from the MAE results table) and confirm the detected mono-allelic expression.

## Adding or editing input data

26. Every time a new input file is added or edited, execute `snakemake -n`. DROP will automatically identify the steps that need to be run to obtain the new results. Run each of the modules and analysis scripts as done before. Be sure to include any new sample in the sample annotation table. Alternatively, execute `snakemake <options>` to run the full pipeline. Changes in the config file and in the sample annotation (except for added rows) are not recognized by DROP. Therefore, executing `snakemake -n` will reflect no new jobs to be performed. A way to force one of the modules to be executed is

by running `snakemake --forcerun <module>`. Refer to the documentation for more details.

## Other uses of RNA-seq data

27. Call variants in RNA-seq data by following the “RNA-seq short variant discovery (SNPs + Indels)” workflow from GATK (Online resources). The input are BAM files from RNA-seq and the output are VCF files that can later be annotated and used to find disease-causal variants. This step adds little value when WGS is also available.
28. Phase variants using phASER<sup>54</sup> (Online resources). The input are BAM files from RNA-seq and VCF files from DNA sequencing and the output is a phased VCF file.
29. Obtain gene fusion calls using Manta<sup>55</sup> (Online resources). The input are BAM files from RNA-seq and a reference genome. The output is a text file containing the genomic coordinates of the fused events, as well as the reads supporting them. Manta can discover fused genes from both different and same chromosomes.

## Troubleshooting

Step	Problem	Possible reason(s)	Solution(s)
5	The sample annotation does not load properly	There were problems with parsing special values	1. Avoid special characters 2. The DROP_group column has comma separated values, so be sure to save the sample annotation with tab-separated values (tsv). 3. Go through the sample annotation html report and try to find any inconsistency.
7	The gene counts are very low	Be sure about the RNA-seq protocol of the samples to determine whether they were paired-end, strand-specific and to which strand the first read aligns to	Load the BAM files in IGV and color them by first-of-pair strand. Go to a gene that lies on the forward strand. Scroll over the reads and note the “Pair orientation” value. Set the STRAND column from the sample annotation to: - “no” if the reads have multiple colors - “yes” if the “Pair orientation” is

			F1R2 - “reverse” if the “Pair orientation” is F2R1
7	Very few genes are counted	- A gtf file from a different genome build was used. - Too low coverage.	- Be sure to provide a gtf file with the same build (hg 19 or hg38) as used for aligning the BAM files. - Check the coverage of the BAM files.
All	Any other problem during the installation or running of the different modules, feature that could be added	Even though we have tested DROP in numerous data sets, there might still be problems that we have not encountered yet.	Go to DROP’s github webpage and click on the ‘Issues’ tab. Scroll through the issues to check if someone else had the same problem already. If not, create a new issue by clicking on ‘New issue’. Copy the error and the step in which it occurred, together with a description of the dataset. We will reply as soon as possible.
All	The results were not as expected	This could be due to many factors such as combining samples sequenced using different protocols, bad choice of cutoffs or other parameters, or a problem with the pipeline.	Make an issue in DROP’s github following the steps described in the previous point.

## Timing

The timings below are provided for the test dataset (composed of 100 samples) with 20 available CPU cores and 96 GB RAM. However, the performance highly depends on the dataset size and the number of available CPU cores and RAM.

- Installation: 3 h max if all dependencies must be installed.
- Creating the sample annotation: ~2 h depending on the experimental design and annotations at hand.
- Modifying the keys of the config file: 15 min.
- Aberrant expression module: ~30 min counting/sample/core, ~1 h merging, filtering, fitting and generating the results. The counting needed 4 GB/sample. The step that required the most memory was the OUTRIDER fit with ~55 GB.

- Aberrant splicing module: ~5 min counting split reads per sample using 3 cores, ~3 min counting non-split reads per sample using 3 cores, ~1 h for merging, computing  $\psi$  and  $\theta$ , filtering, fitting and generating the results. The counting of the split reads needed 8 GB/sample. The step that required the most memory was computing z scores and  $P$  values with ~80 GB.
- MAE module: ~2 h counting/sample/core, 1 h for generation of all results. Running the negative binomial test needed 11 GB/sample, which was also the step that required the most memory.
- Sample QC module: similar to the MAE module.

## Anticipated results

One can expect a handful of expression outliers per sample. In the test dataset the median number of expression outliers is 1 and the 90th percentile of samples had 7 expression outliers (Fig. S5e), in line with previous reports<sup>7,18</sup> and other in-house analyses. These expression outliers should be interpreted according to their fold changes. A strong down-regulation (fold-changes < 0.2) probably implies impaired gene function. Fold-changes of weaker amplitudes should further be investigated. In particular a fold-change of 0.5 often reflects loss of expression of one allele. Inspecting the MAE results may further reveal mono-allelic expression of a rare variant harbored by the other allele<sup>7,10</sup>.

One can also expect a handful of splicing outliers per sample. These aberrant splicing events should be visualized as sashimi plots, for instance using the genome browser IGV, to detect the nature of the splicing defect including intron retention, exon truncation, exon elongation, or exon skipping. The next step is to look for direct splice site and near splice site variants that can explain the defect. However, other variants such as synonymous and deep intronic, have also been described to activate new splice sites<sup>9</sup> potentially originating from cryptic exons<sup>7,8,10,11</sup>.

Analysis of RNA-seq data delivers candidate genes whose aberrant expression or splicing may be contributing to the clinical presentation of the patient. Analysis of the gene fusion results can further provide candidate genes. If provided in the sample annotation, DROP also outputs whether the outlier genes overlap with the phenotypes. Further prioritization involves adding specific disease gene lists, or general ones such as OMIM<sup>56</sup>. Promising candidate genes need to be further investigated for potential pathogenic variants. RNA-seq informs about consequences of genetic variants but not necessarily allows pinpointing the causal variant.

Nonetheless, in some instances a splice defect is directly explained by a variant in the splice site. Also, underexpression outliers can be explained by nonsense mediated decay triggered by premature termination codons caused by stop-gain variants or frameshift variants. In this respect the value of RNA-seq is to support prioritization of candidate genes which need further inspection or functional validation of an already candidate variant. In any case, for the molecular diagnostic interpretation we refer to the ACMG standards and guidelines for the interpretation of sequence variants<sup>57</sup>.

## Acknowledgements

We thank Carla Andrade for helping with the DROP logo, as well as the members of the Gagneur Lab and Prokisch Lab for input. V.A.Y., C.M., H.P., and J.G. were supported by the EU Horizon2020 Collaborative Research Project SOUND (633974). The Bavaria California Technology Center supported C.M through a fellowship. The German Bundesministerium für Bildung und Forschung (BMBF) supported the study through the e:Med Networking funds AbCD-Net (FKZ 01ZX1706A to V.A.Y., C.M., and J.G.), the German Network for Mitochondrial Disorders (mitoNET; 01GM1113C to H.P.), and the E-Rare project GENOMIT (01GM1207 to H.P.). A fellowship through the Graduate School of Quantitative Biosciences Munich supports V.A.Y.. 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 June 12, 2017, under accession number dbGaP: phs00424.v6.p1.

## Author contributions

V.A.Y., C.M., M.F.M., and J.G. participated in the design of the workflow. V.A.Y., C.M., M.F.M., D.S.A., I.S., and P.F.G. contributed to the computational pipeline. L.F. implemented the candidate prioritization pipeline. L.W. designed and implemented wBuild. V.A.Y. and J.G. wrote the manuscript with the help of L.F, D.S.A., M.G., and I.S.. C.M., H.P., and J.G. supervised the research. All authors revised the manuscript.

## Competing interests

The authors declare no competing interests.



## Online resources

DROP documentation: <https://drop-rna.readthedocs.io/en/latest/>

DROP in github: <https://github.com/gagneurlab/DROP>

DROP resource folder: [https://i12g-gagneurweb.in.tum.de/public/paper/drop\\_analysis/resource/](https://i12g-gagneurweb.in.tum.de/public/paper/drop_analysis/resource/)

DROP test dataset analysis:

[https://i12g-gagneurweb.in.tum.de/public/paper/drop\\_analysis/webDir/html/drop\\_analysis\\_index.html](https://i12g-gagneurweb.in.tum.de/public/paper/drop_analysis/webDir/html/drop_analysis_index.html)

DROP Kremer dataset analysis:

[https://i12g-gagneurweb.in.tum.de/public/paper/drop\\_analysis/kremer/kremer/kremer\\_index.html](https://i12g-gagneurweb.in.tum.de/public/paper/drop_analysis/kremer/kremer/kremer_index.html)

FRASERR in Bioconductor:

<http://bioconductor.org/packages/release/bioc/html/FRASER.html> Geuvadis dataset:

<https://www.internationalgenome.org/data-portal/data-collection/geuvadis>

Human Phenotype Ontology: <https://hpo.jax.org/>

HTSeq documentation: [https://htseq.readthedocs.io/en/release\\_0.11.1/count.html](https://htseq.readthedocs.io/en/release_0.11.1/count.html)

Manta: <https://github.com/Illumina/manta/>

OUTRIDER in Bioconductor:

<https://bioconductor.org/packages/release/bioc/html/OUTRIDER.html>

phASER: <https://github.com/secastel/phaser>

Snakemake documentation: <https://snakemake.readthedocs.io/en/stable/>

Variant effect predictor tutorial:

[https://uswest.ensembl.org/info/docs/tools/vep/script/vep\\_tutorial.html](https://uswest.ensembl.org/info/docs/tools/vep/script/vep_tutorial.html)

wBuild documentation: <https://wbuild.readthedocs.io/en/latest/>

YAML documentation:

[https://docs.ansible.com/ansible/latest/reference\\_appendices/YAMLSyntax.html](https://docs.ansible.com/ansible/latest/reference_appendices/YAMLSyntax.html)

## References

1. Bamshad, M. J. *et al.* Exome sequencing as a tool for Mendelian disease gene discovery. *Nat. Rev. Genet.* **12**, 745–755 (2011).
2. Yang, Y. *et al.* Clinical Whole-Exome Sequencing for the Diagnosis of Mendelian Disorders. *N. Engl. J. Med.* **369**, 1502–1511 (2013).
3. Taylor, J. C. *et al.* Factors influencing success of clinical genome sequencing across a broad spectrum of disorders. *Nat. Genet.* **47**, 717–726 (2015).

4. Lionel, A. C. *et al.* Improved diagnostic yield compared with targeted gene sequencing panels suggests a role for whole-genome sequencing as a first-tier genetic test. *Genet. Med.* **20**, 435–443 (2018).
5. Chong, J. X. *et al.* The Genetic Basis of Mendelian Phenotypes: Discoveries, Challenges, and Opportunities. *Am. J. Hum. Genet.* **97**, 199–215 (2015).
6. Cooper, G. M. Parlez-vous VUS? *Cold Spring Harb. Lab. Press* 1423–1426 (2015) doi:doi/10.1101/gr.190116.115.
7. Kremer, L. S. *et al.* Genetic diagnosis of Mendelian disorders via RNA sequencing. *Nat. Commun.* **8**, (2017).
8. Cummings, B. B. *et al.* Improving genetic diagnosis in Mendelian disease with transcriptome sequencing. *Sci. Transl. Med.* **9**, 12 (2017).
9. Frésard, L. *et al.* Identification of rare-disease genes using blood transcriptome sequencing and large control cohorts. *Nat. Med.* **25**, 911–919 (2019).
10. Gonorazky, H. D. *et al.* Expanding the Boundaries of RNA Sequencing as a Diagnostic Tool for Rare Mendelian Disease. *Am. J. Hum. Genet.* **104**, 466–483 (2019).
11. Lee, H. *et al.* Diagnostic utility of transcriptome sequencing for rare Mendelian diseases. *Genet. Med.* (2019) doi:10.1038/s41436-019-0672-1.
12. Li, H. *et al.* The Sequence Alignment/Map format and SAMtools. *Bioinformatics* **25**, 2078–2079 (2009).
13. Danecek, P. *et al.* The variant call format and VCFtools. *Bioinformatics* **27**, 2156–2158 (2011).
14. Köster, J. & Rahmann, S. Snakemake--a scalable bioinformatics workflow engine. *Bioinformatics* **28**, 2520–2522 (2012).
15. Lappalainen, T. *et al.* Transcriptome and genome sequencing uncovers functional variation

- in humans. *Nature* **501**, 506–511 (2013).
16. Anders, S. & Huber, W. Differential expression analysis for sequence count data. *Genome Biol.* **11**, (2010).
  17. Li, Y. I. *et al.* Annotation-free quantification of RNA splicing using LeafCutter. *Nat. Genet.* **50**, 151–158 (2018).
  18. Brechtman, F. *et al.* OUTRIDER: A Statistical Method for Detecting Aberrantly Expressed Genes in RNA Sequencing Data. *Am. J. Hum. Genet.* **103**, 907–917 (2018).
  19. Mertes, C. *et al.* Detection of aberrant splicing events in RNA-Seq data with FRASER. <https://tinyurl.com/FRASER-paper> (2019).
  20. Köhler, S. *et al.* Expansion of the Human Phenotype Ontology (HPO) knowledge base and resources. *Nucleic Acids Res.* **47**, D1018–D1027 (2019).
  21. GTEx Consortium. Genetic effects on gene expression across human tissues. *Nature* **550**, 204–213 (2017).
  22. Papatheodorou, I. *et al.* Expression Atlas: gene and protein expression across multiple studies and organisms. *Nucleic Acids Res.* **46**, D246–D251 (2018).
  23. Aicher, J. K., Jewell, P., Vaquero-Garcia, J., Barash, Y. & Bhoj, E. J. Mapping RNA splicing variations in clinically accessible and nonaccessible tissues to facilitate Mendelian disease diagnosis using RNA-seq. *Genet. Med.* (2020) doi:10.1038/s41436-020-0780-y.
  24. 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).
  25. Scotti, M. M. & Swanson, M. S. RNA mis-splicing in disease. *Nat. Rev. Genet.* **17**, 19–32 (2016).
  26. Singh, R. K. & Cooper, T. A. Pre-mRNA splicing in disease and therapeutics. *Trends Mol.*

- Med.* **18**, 472–482 (2012).
27. Cheng, J. *et al.* MMSplice: modular modeling improves the predictions of genetic variant effects on splicing. *Genome Biol.* **20**, 48 (2019).
  28. Jaganathan, K. *et al.* Predicting Splicing from Primary Sequence with Deep Learning. *Cell* **176**, 535–548.e24 (2019).
  29. Gonorazky, H. *et al.* RNAseq analysis for the diagnosis of muscular dystrophy. *Ann. Clin. Transl. Neurol.* **3**, 55–60 (2016).
  30. Kernohan, K. D. *et al.* Whole-transcriptome sequencing in blood provides a diagnosis of spinal muscular atrophy with progressive myoclonic epilepsy. *Hum. Mutat.* **38**, 611–614 (2017).
  31. Hamanaka, K. *et al.* RNA sequencing solved the most common but unrecognized NEB pathogenic variant in Japanese nemaline myopathy. *Genet. Med.* **21**, 1629–1638 (2019).
  32. Wang, K. *et al.* Whole-genome DNA/RNA sequencing identifies truncating mutations in RBCK1 in a novel Mendelian disease with neuromuscular and cardiac involvement. *Genome Med.* **5**, (2013).
  33. Pervouchine, D. D., Knowles, D. G. & Guigo, R. Intron-centric estimation of alternative splicing from RNA-seq data. *Bioinformatics* **29**, 273–274 (2013).
  34. Kapustin, Y. *et al.* Cryptic splice sites and split genes. *Nucleic Acids Res.* **39**, 5837–5844 (2011).
  35. Mohammadi, P. *et al.* Genetic regulatory variation in populations informs transcriptome analysis in rare disease. *Science* **366**, 351–356 (2019).
  36. Albers, C. A. *et al.* Compound inheritance of a low-frequency regulatory SNP and a rare null mutation in exon-junction complex subunit RBM8A causes TAR syndrome. *Nat. Genet.* **44**, 435–439 (2012).

37. van Haelst, M. M. *et al.* Further confirmation of the MED13L haploinsufficiency syndrome. *Eur. J. Hum. Genet.* **23**, 135–138 (2015).
38. Lindstrand, A. *et al.* Different mutations in *PDE4D* associated with developmental disorders with mirror phenotypes. *J. Med. Genet.* **51**, 45–54 (2014).
39. 't Hoen, P. A. C. *et al.* Reproducibility of high-throughput mRNA and small RNA sequencing across laboratories. *Nat. Biotechnol.* **31**, 1015–1022 (2013).
40. Lee, S. *et al.* NGSCheckMate: software for validating sample identity in next-generation sequencing studies within and across data types. *Nucleic Acids Res.* **45**, (2017).
41. Ewels, P., Magnusson, M., Lundin, S. & Käller, M. MultiQC: summarize analysis results for multiple tools and samples in a single report. *Bioinformatics* **32**, 3047–3048 (2016).
42. Li, H. & Durbin, R. Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics* **25**, 1754–1760 (2009).
43. McKenna, A. *et al.* The Genome Analysis Toolkit: A MapReduce framework for analyzing next-generation DNA sequencing data. *Genome Res.* **20**, 1297–1303 (2010).
44. Van der Auwera, G. A. *et al.* From FastQ Data to High-Confidence Variant Calls: The Genome Analysis Toolkit Best Practices Pipeline. in *Current Protocols in Bioinformatics* 11.10.1-11.10.33 (John Wiley & Sons, Inc., 2013). doi:10.1002/0471250953.bi1110s43.
45. Li, H. Tabix: fast retrieval of sequence features from generic TAB-delimited files. *Bioinformatics* **27**, 718–719 (2011).
46. Haeussler, M. *et al.* The UCSC Genome Browser database: 2019 update. *Nucleic Acids Res.* **47**, D853–D858 (2019).
47. Frankish, A. *et al.* GENCODE reference annotation for the human and mouse genomes. *Nucleic Acids Res.* **47**, D766–D773 (2019).
48. Li, H. A statistical framework for SNP calling, mutation discovery, association mapping and

- population genetical parameter estimation from sequencing data. *Bioinformatics* **27**, 2987–2993 (2011).
49. Robinson, J. T. *et al.* Integrative genomics viewer. *Nat. Biotechnol.* **29**, 3 (2011).
50. Ben-Kiki, O. & Evans, C. YAML Ain't Markup Language (YAML™) Version 1.2. 84.
51. Anders, S., Pyl, P. T. & Huber, W. HTSeq--a Python framework to work with high-throughput sequencing data. *Bioinformatics* **31**, 166–169 (2015).
52. McCarthy, D. J., Chen, Y. & Smyth, G. K. Differential expression analysis of multifactor RNA-Seq experiments with respect to biological variation. *Nucleic Acids Res.* **40**, 4288–4297 (2012).
53. Karczewski, K. J. *et al.* Variation across 141,456 human exomes and genomes reveals the spectrum of loss-of-function intolerance across human protein-coding genes. <http://biorxiv.org/lookup/doi/10.1101/531210> (2019) doi:10.1101/531210.
54. Castel, S. E., Mohammadi, P., Chung, W. K., Shen, Y. & Lappalainen, T. Rare variant phasing and haplotypic expression from RNA sequencing with phASER. *Nat. Commun.* **7**, 12817 (2016).
55. Chen, X. *et al.* Manta: rapid detection of structural variants and indels for germline and cancer sequencing applications. *Bioinformatics* **32**, 1220–1222 (2016).
56. Amberger, J. S., Bocchini, C. A., Schiettecatte, F., Scott, A. F. & Hamosh, A. OMIM.org: Online Mendelian Inheritance in Man (OMIM®), an online catalog of human genes and genetic disorders. *Nucleic Acids Res.* **43**, D789–D798 (2015).
57. Richards, S. *et al.* Standards and guidelines for the interpretation of sequence variants: a joint consensus recommendation of the American College of Medical Genetics and Genomics and the Association for Molecular Pathology. *Genet. Med.* **17**, 405–423 (2015).