GigaScience, 7, 2018, 1–9

doi: 10.1093/gigascience/giy059
Advance Access Publication Date: 26 May 2018
Technical Note

TECHNICAL NOTE

zUMIs - A fast and flexible pipeline to process RNA sequencing data with UMIs

Swati Parekh[*,†,‡], Christoph Ziegenhain[†,§], Beate Vieth, Wolfgang Enard and Ines Hellmann[*]

Anthropology and Human Genomics, Department of Biology II, Ludwig-Maximilians University,
Grosshaderner Str. 2, 82152 Martinsried, Germany

[*] **Correspondence address.** Ines Hellmann. E-mail: hellmann@bio.lmu.de http://orcid.org/0000-0003-0588-1313 and Swati Parekh. E-mail: sparekh@age.mpg.de https://orcid.org/0000-0002-4826-1651
[†] Contributed equally.
[‡] Present address: Max Planck Institute for Biology of Ageing, 50931 Cologne, Germany.
[§] Present address: Department of Cell & Molecular Biology, Karolinska Institutet, 171 77 Stockholm, Sweden.

Technical Note

Abstract

Background: Single-cell RNA-sequencing (scRNA-seq) experiments typically analyze hundreds or thousands of cells after amplification of the cDNA. The high throughput is made possible by the early introduction of sample-specific bar codes (BCs), and the amplification bias is alleviated by unique molecular identifiers (UMIs). Thus, the ideal analysis pipeline for scRNA-seq data needs to efficiently tabulate reads according to both BC and UMI. **Findings:** *zUMIs* is a pipeline that can handle both known and random BCs and also efficiently collapse UMIs, either just for exon mapping reads or for both exon and intron mapping reads. If BC annotation is missing, *zUMIs* can accurately detect intact cells from the distribution of sequencing reads. Another unique feature of *zUMIs* is the adaptive downsampling function that facilitates dealing with hugely varying library sizes but also allows the user to evaluate whether the library has been sequenced to saturation. To illustrate the utility of *zUMIs*, we analyzed a single-nucleus RNA-seq dataset and show that more than 35% of all reads map to introns. Also, we show that these intronic reads are informative about expression levels, significantly increasing the number of detected genes and improving the cluster resolution. **Conclusions:** *zUMIs* flexibility makes if possible to accommodate data generated with any of the major scRNA-seq protocols that use BCs and UMIs and is the most feature-rich, fast, and user-friendly pipeline to process such scRNA-seq data.

Keywords: single-cell RNA-sequencing; *digital gene expression*; *unique molecular identifiers*; *pipeline*

Introduction

The recent development of increasingly sensitive protocols allows for the generation of RNA-sequencing (RNA-seq) libraries of single cells [1]. The throughput of such single-cell RNA-seq (scRNA-seq) protocols is rapidly increasing, enabling the profiling of tens of thousands of cells [2, 3] and opening exciting possibilities to analyze cellular identities [4, 5]. As the required amplification from such small starting amounts introduces substantial amounts of noise [6], many scRNA-seq protocols incorporate unique molecular identifiers (UMIs) to label individual cDNA molecules with a random nucleotide sequence before amplification [7]. This enables the computational removal of amplification noise and thus increases the power to detect expression differences between cells [8, 9]. To increase the throughput, many protocols also incorporate sample-specific bar codes (BCs) to label all cDNA molecules of a single cell with a nucleotide sequence before library generation [10]. This allows for early pooling, which further decreases amplification noise [6]. Addition-

Received: 17 October 2017; **Revised:** 16 March 2018; **Accepted:** 15 May 2018

Table 1: Features of available UMI pipelines for the quantification of gene expression data.

Name	Reference	Open source	Quality filter	UMI collapsing	Mapper	BC detection	Intron	Down-sampling	Compatible UMI library protocols
Cell Ranger	[2]	yes	BC+UMI	Hamming distance	STAR	A	no	yes	[2]
CEL-seq	[15]	yes	BC+UMI	Identity only	bowtie2	WL	no	no	[15, 46]
dropEst	[16]	yes	BC	Frequency-based	TopHat2 or Kallisto	WL,top-n,EM	yes	no	[2, 13, 19]
Drop-seq-tools	[13]	no	BC+UMI	Hamming distance	STAR	WL,top-n	no	no	[13, 15, 17]
scPipe	[47]	yes	BC+UMI	Hamming distance	subread	WL,top-n	no	no	[13, 17, 18, 46]
umis	[14]	yes	BC	Frequency-based	Kallisto	WL,top-n,EM	no	no	[2, 13, 17–19, 46, 48]
UMI-tools	[25]	yes	BC+UMI	Network-based	BWA	WL	no	no	[17, 19]
zUMIs	This work	yes	BC+UMI	Hamming distance	STAR	A,WL,top-n	yes	yes	[2, 3, 12, 13, 15, 17, 18, 21, 46, 48]

We consider whether the pipeline is open source, has sequence quality filters for cell BCs and UMIs, mappers, UMI-collapsing options, options for BC detection (A, automatically infer intact BCs; WL, extract only the given list of known BCs; top-n, order BCs according the number of reads and keep the top n BCs; EM, merge BCs with given edit distance), whether it can count intron mapping reads, whether it offers a utility to make varying library sizes more comparable via downsampling, and finally with which RNA-seq library preparation protocols is it compatible

ally, for cell types such as primary neurons, it has been proven to be more feasible to isolate RNA from single nuclei rather than whole cells [11, 12]. This decreases mRNA amounts further so that it has been suggested to count intron mapping reads originating from nascent RNAs as part of single-cell expression profiles [11]. However, the few bioinformatic tools that process RNA-seq data with UMIs and BCs have limitations (Table 1). For example, the Drop-seq-tools is not an open source [13]. While Cell Ranger is open, it is exceedingly difficult to adapt the code to new or unknown sample BCs and other library types. Other tools are specifically designed to work with one mapping algorithm and focus mainly on transcriptome references [14, 15]. Furthermore, the only other UMI-RNA-seq pipeline providing the utility to also consider intron mapping reads, dropEst [16], is only applicable to droplet-based protocols. Here, we present *zUMIs*, a fast and flexible pipeline that overcomes these limitations.

Findings

zUMIs is a pipeline to process RNA-seq data that were multiplexed using cell BCs and also contain UMIs. Read-pairs are filtered to remove reads with low-quality BCs or UMIs based on sequence and then mapped to a reference genome (Fig.1). Next, *zUMIs* generates UMI and read count tables for exon and exon+intron counting. We reason that very low input material such as from single nuclei sequencing might profit from including reads that potentially originate from nascent RNAs. Another unique feature of *zUMIs* is that it allows for downsampling of reads before collapsing UMIs, thus enabling the user to assess whether a library was sequenced to saturation or whether deeper sequencing is necessary to depict the full mRNA complexity. Furthermore, *zUMIs* is flexible with respect to the length and sequences of the BCs and UMIs, supporting protocols that have both sequences in one read [2, 3, 12, 13, 15, 17, 18] as well as protocols that provide UMI and BC in separate reads [19–21]. This makes *zUMIs* the only tool that is easily compatible with all major UMI-based scRNA-seq protocols.

Implementation and Operation

Filtering and mapping

The first step in our pipeline is to filter reads that have low-quality BCs according to a user-defined threshold (Fig.1). This step eliminates the majority of spurious BCs and thus greatly reduces the number of BCs that need to be considered for counting. Similarly, we also filter low-quality UMIs.

The remaining reads are then mapped to the genome using the splice-aware aligner `STAR` [22]. The user is free to customize mapping by using the options of `STAR`. Furthermore, if the user wishes to use a different mapper, it is also possible to provide *zUMIs* with an aligned bam file instead of the fastq file with the cDNA sequence, with the sole requirement that only one mapping position per read is reported in the bam file.

Transcript counting

Next, reads are assigned to genes. In order to distinguish exon and intron counts, we generate two mutually exclusive annotation files from the provided gtf, one detailing exon positions, the other introns. Based on those annotations, `Rsubread featureCounts` [23] is used to first assign reads to exons and afterward to check whether the remaining reads fall into introns, in other words, if a read is overlapping with intronic and exonic sequences, it will be assigned to the exon only. The output is then read into R using `data.table` [24], generating count tables for UMIs and reads per gene per BC. We then collapse UMIs that were mapped either to the exon or intron of the same gene. Note that only the processing of intron and exon reads together allows for properly collapse of UMIs that can be sampled from the intronic as well as from the exonic part of the same nascent mRNA molecule.

Per default, we only collapse UMIs by sequence identity. If there is a risk that a large proportion of UMIs remains under-collapsed due to sequence errors, *zUMIs* provides the option to collapse UMIs within a given Hamming distance. We compare

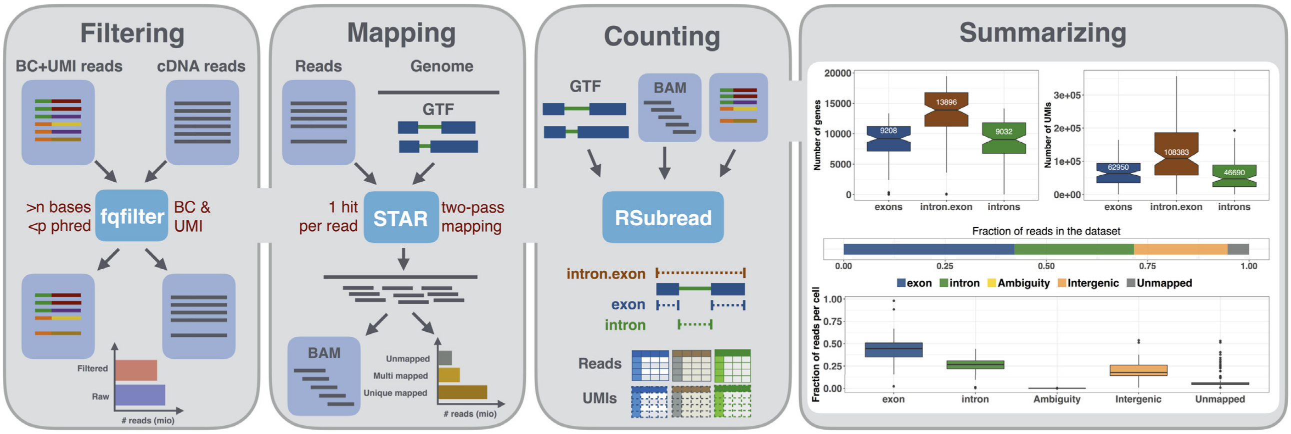

Figure 1: Schematic of the zUMIs pipeline. Each of the gray panels from left to right depicts a step of the *zUMIs* pipeline. First, fastq files are filtered according to user-defined bar code (BC) and unique molecular identifier (UMI) quality thresholds. Next, the remaining cDNA reads are mapped to the reference genome using STAR. Gene-wise read and UMI count tables are generated for exon, intron, and exon+intron overlapping reads. To obtain comparable library sizes, reads can be downsampled to a desired range during the counting step. In addition, *zUMIs* also generates data and plots for several quality measures, such as the number of detected genes/UMIs per BCe and distribution of reads into mapping feature categories.

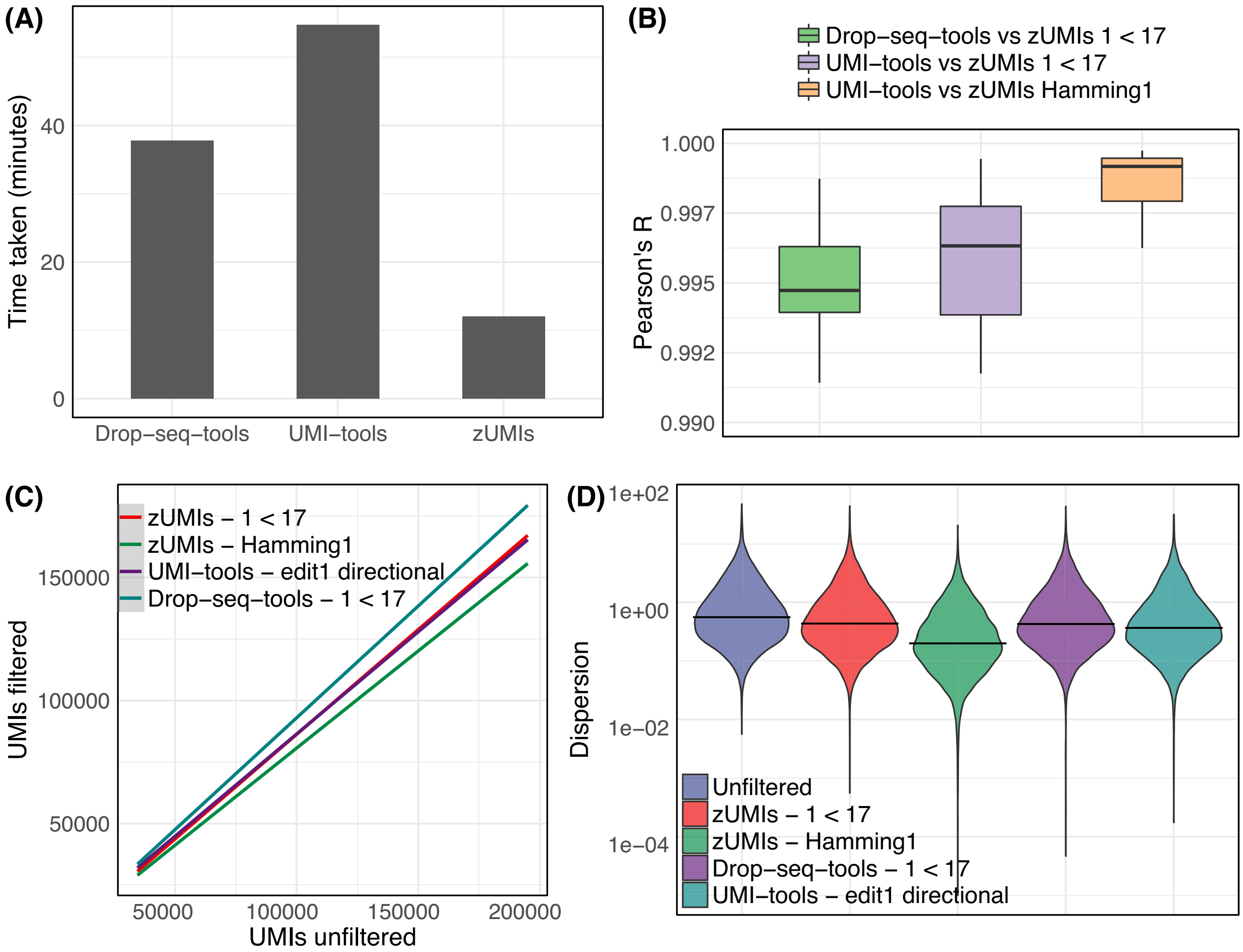

Figure 2: Comparison of different UMI collapsing methods. We compared Drop-seq-tools and UMI-tools with zUMIs using our HEK dataset (227 mio reads). **(A)** Run time to count exonic UMIs using zUMIs (hamming distance = 0), UMI-tools ("unique" mode) and Drop-seq-tools (edit distance = 0). **(B)** Box plots of correlation coefficients of gene-wise UMI counts of the same cell generated with different methods. UMI counts generated using zUMIs (quality filter "1 base under phred 17" or hamming distance = 1) were correlated to UMI counts generated using Drop-seq-tools (quality filter "1 base under phred 17") and UMI-tools ("directional adjacency" mode). **(C)** Comparison of the total number of UMIs per cell derived from different counting methods to "unfiltered" counts. **(D)** Violin plots of gene-wise dispersion estimates with different quality filtering and UMI collapsing methods.

the two *zUMIs* UMI-collapsing options to the recommended directional adjacency approach implemented in UMI-tools [25] using our in-house example dataset (see Methods section). *zUMIs* identity collapsing yields nearly identical UMI counts per cell as UMI-tools, while Hamming distance yields increasingly fewer UMIs per cell with increasing sequencing depth (Fig.2C). Smith et al [25] suggest that edit distance collapsing without considering the relative frequencies of UMIs might indeed overreach and overcollapse the UMIs. We suspect that this is indeed what happens in our example data, where we find that gene-wise dispersion estimates appear suspiciously truncated as expected if several counts are unduly reduce to one, the minimal number after collapsing (Fig.2D).

However, note that the above-described differences are minor. By and large, there is good agreement between UMI counts obtained by UMI-tools [25], the Drop-seq pipeline [13], and *zUMIs*. The correlation between gene-wise counts of the same cell is >0.99 for all comparisons (Fig. 2B). In light of this, we consider the >3 times higher processing speed of *zUMIs* to be a decisive advantage (Fig.2A).

Cell BC selection

In order to be compatible with well-based and droplet-based scRNA-seq methods, *zUMIs* needs to be able to deal with known as well as random BCs. As default behavior, *zUMIs* infers which BCs mark good cells from the data (Fig.3A, 3B). To this end, we fit a k-dimensional multivariate normal distribution using the R-package `mclust` [26, 27] for the number of reads/BC, where k is empirically determined by mclust via the Bayesian information criterion. We reason that only the kth normal distribution with the largest mean contains BCs that identify reads originating from intact cells. We exclude all BCs that fall in the lower 1% tail of this kth normal distribution to exclude spurious BCs. The HEK dataset used here contains 96 cells with known BCs and *zUMIs* identifies 99 BCs as intact, including all the 96 known BCs. Also, for the single-nucleus RNA-seq from Habib et al. [12], *zUMIs* identified a reasonable number of cells; Habib et al. report 10,877 nuclei and zUMIs identified 11,013 intact nuclei. However, we recommend to always check the elbow plot generated by zUMIs (Fig.3B) to confirm that the cutoff used by zUMIs is valid for a given dataset. In cases where the number of BCs or BC sequences are known, it is preferable to use this information. If *zUMIs* is either given the number of expected BCs or is provided with a list of BC sequences, it will use this information and forgo automatic inference.

Downsampling

scRNA-seq library sizes can vary by orders of magnitude, which complicates normalization [28, 29]. A straight-forward solution for this issue is to downsample overrepresented libraries [30]. *zUMIs* has an built-in function for downsampling datasets to a user-specified number of reads or a range of reads. By default, *zUMIs* downsamples all selected BCs to be within three absolute deviations from the median number of reads per BC (Fig.3C). Alternatively, the user can provide a target sequencing depth, and *zUMIs* will downsample to the specified read number or omit the cell from the downsampled count table if fewer reads were present. Furthermore, *zUMIs* also allows the user to specify a multiple target read number at once for downsampling. This feature is helpful if the user wishes to determine whether the RNA-seq library was sequenced to saturation or whether further sequencing would increase the number of detected genes or UMIs enough to justify the extra cost. In our HEK-cell example dataset, the number of detected genes starts leveling off at 1 million reads. Sequencing double that amount would only increase the number of detected genes from 9,000 to 10,600 when counting exon reads (Fig.3D). In line with previous findings [8, 14], the saturation curve of exon+intron counting runs parallel to the one for exon counting, both indicating that a sequencing depth of 1 million reads per cell is sufficient for these libraries.

Output and statistics

zUMIs outputs three UMI and three read count tables: gene-wise counts for traditional exon counting, one for intron and one for exon+intron counts. If a user chooses the downsampling option, six additional count tables per target read count are provided. To evaluate library quality, *zUMIs* summarizes the mapping statistics of the reads. While exon and intron mapping reads likely represent mRNA quantities, a high fraction of intergenic and unmapped reads indicates low-quality libraries. Another measure of RNA-seq library quality is the complexity of the library, for which the number of detected genes and the number of identified UMIs are good measures (Fig.1). We processed 227 million reads with *zUMIs* and quantified expression levels for exon and intron counts on a Unix machine using up to 16 threads, which took less than 3 hours. Increasing the number of reads increases the processing time approximately linearly, where filtering, mapping, and counting each take up roughly one third of the total time (Fig.3E). We also observed that the peak random access memory usage for processing datasets of 227, 500, and 1,000 million pairs was 42 Gb, 89 Gb, and 172 Gb, respectively. Finally, *zUMIs* could process the largest scRNA-seq dataset reported to date with around 1.3 million brain cells and 30 billion read-pairs generated with 10xGenomics Chromium (see Methods section) on a 22-core processor in only 7 days.

Intron counting

Recently, it has been shown that intron mapping reads in RNA-seq likely originate from nascent mRNAs and are useful for gene expression estimates [31, 32]. Additionally, novel approaches leverage the ratios of intron and exon mapping reads to infer information on transcription dynamics and cell states [33]. To address this new aspect of analysis, *zUMIs* also counts and collapses intron-only mapping reads as well as intron and exon mapping reads from the same gene with the same UMI. To assess the information gain from intronic reads to estimate gene expression levels, we analyzed a publicly available DroNc-seq dataset from mouse brain ([12]; see Methods section). For the ~11,000 single nuclei of this dataset, the fraction of intron mapping reads of all reads goes up to 61%. Thus, if intronic reads are considered, the mean number of detected genes per cell increases from 1,041 for exon counts to 1,995 for exon+intron counts. Next, we used the resulting UMI count tables to investigate whether exon+intron counting improves the identification of cell types, as suggested by Lake et al. [11]. The validity and accuracy of counting introns for single-nucleus sequencing methods has recently been demonstrated [34]. Following the Seurat pipeline to cluster cells [35, 36], we find that using exon+intron counts discriminates 28 clusters, while we could only discriminate 19 clusters using exon counts (Fig.4A, 4B). The larger number of clusters is not simply due to the increase in the counted UMIs and genes. When we permute the intron counts across cells and add them to the exon counts, the added noise actually reduces the number of identifiable clusters (Fig.4E).

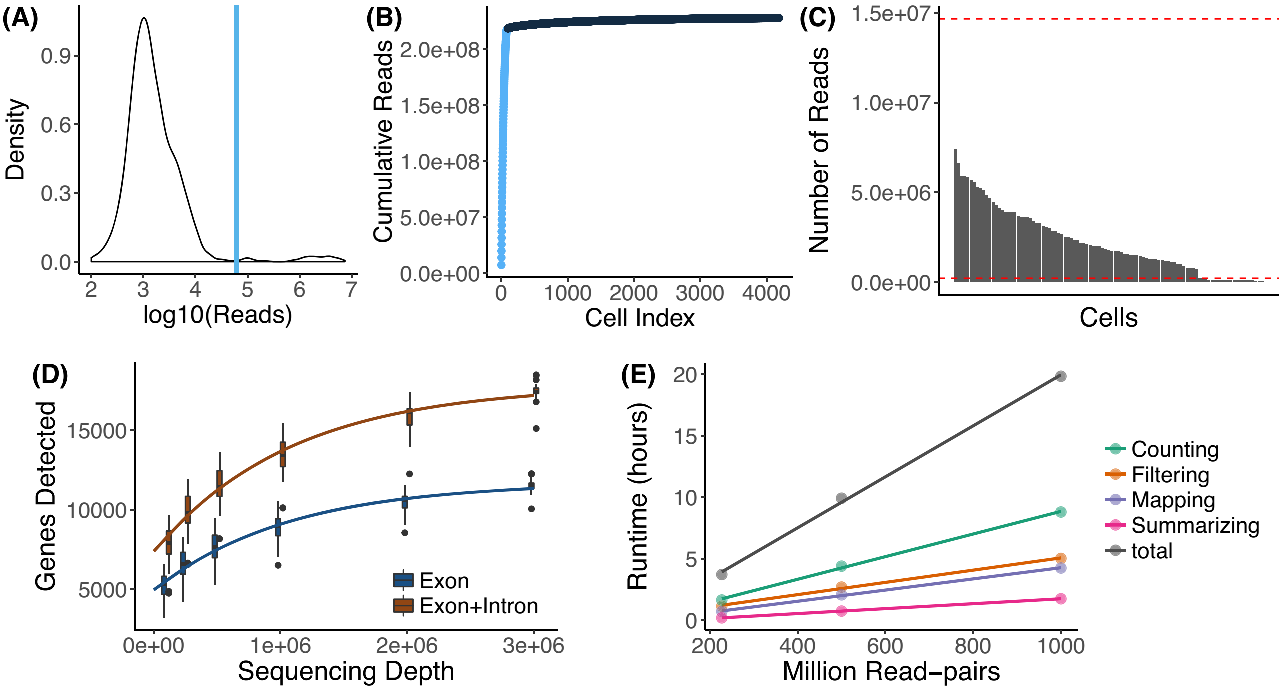

Figure 3: Utilities of zUMIs. Each of the panels shows the utilities of *zUMIs* pipeline. The plots from A–D show the results from the example HEK dataset used here. **(A)** The plot shows a density distribution of reads per BC. Cell BCs with reads right of the blue line are selected. **(B)** The plot shows the cumulative read distribution in the example HEK dataset where the BCs in light blue are the selected cells. **(C)** The bar plot shows the number of reads per selected cell BC with the red lines showing upper and lower median absolute deviation (MAD) cutoffs for adaptive downsampling. Here, the cells below the lower MAD have very low coverage and are discarded in downsampled count tables. **(D)** Cells were downsampled to six depths from 100,000 to 3,000,000 reads. For each sequencing depth, the genes detected per cell are shown. **(E)** Runtime for three datasets with 227, 500, and 1,000 million read-pairs. The runtime is divided in the main steps of the *zUMIs* pipeline as follows: filtering, mapping, counting, and summarizing. Each dataset was processed using 16 threads ("-p 16").

We continue to further characterize the seven clusters that were subdivided by the addition of intron counts (Fig.4D). First, we identify DE genes between the newly formed clusters. If we count only exon reads, there appear to be, on average, only 10 DE genes between the subgroups, while exon+intron counting yields ∼10 times more DE genes, thus corroborating the signal found with clustering. The log2-fold changes of those additional DE genes estimated with either counting strategy are generally in good agreement; especially large log2-fold changes are detected with both exon and exon+intron counting (Fig.4F). Genes that are detected as DE in only one of our counting strategies have small log2-fold changes, and there are more of these small changes detected using exon+intron counting.

Detecting more genes naturally increases the chance to also detect more informative genes. Here, we cross-reference the gene list with marker genes for transcriptomic subtypes detected for major cell types of the mouse brain [37] and find that ∼5% of the additional genes are also marker genes, which corresponds well to the general frequency of marker genes among the detected genes (4%). In the same vein, we also detect proportionally more DE genes with exon+intron counting compared to exon counting. Thus, including introns simply allows us to better detect present transcripts, while leaving the proportions of interest unaltered. Having a closer look at cluster 7, it was split into a bigger (7) and a smaller cluster (24) using exon+intron counting (Fig.4A-C), we find one marker gene (Il1rapl2) to be DE between the subclusters using exon+intron counting, while Il1rapl2 had only spurious counts using exon counts. Il1rapl2 is a marker for transcriptomic subtypes of GABAergic Pvalb-type neurons [37], suggesting that the split of cluster 7 might be biologically meaningful (Fig.4E).

In order to evaluate the power gained by exon+intron counting in a more systematic way, we perform power simulations using empirical mean and dispersion distributions from the largest and most uniform cluster (∼1,500 cells) [9]. For a fair comparison, we include all detected genes, which is equivalent to the number of genes detected with exon+intron counting. Also, since we call a gene detected as soon as one count is associated, exon counting is necessarily a subset of exon+intron. Thus, there are, on average, 4 times more genes in the lowest expression quantile for exon counting than for exon+intron counting (Fig.4H). For those genes, expression is too spurious to be used for differential expression analysis; for exon+intron counting, we have, on average, 60% power to detect a DE gene in the first mean expression bin with a well-controlled false discovery rate (FDR) (Fig.4G). In summary, the increased power for exon+intron counting and probably also the larger number of clusters are due to better detection of lowly expressed genes. Furthermore, we think that although potentially noisy, the large number of additionally detected genes makes exon+intron counting worthwhile, especially for single-nuclei sequencing techniques that are enriched for nuclear nascent RNA transcripts, such as DroNc-seq [12]. Additionally, exon+intron counting may help in extracting as much information as possible from low coverage data as generated in the context of high-throughput cell atlas efforts (e.g., 10,000–20,000 reads/cell [38, 39]. Last, users should always exclude the possibility of intronic reads stemming from genomic DNA contamination in the library preparation by confirming low inter-

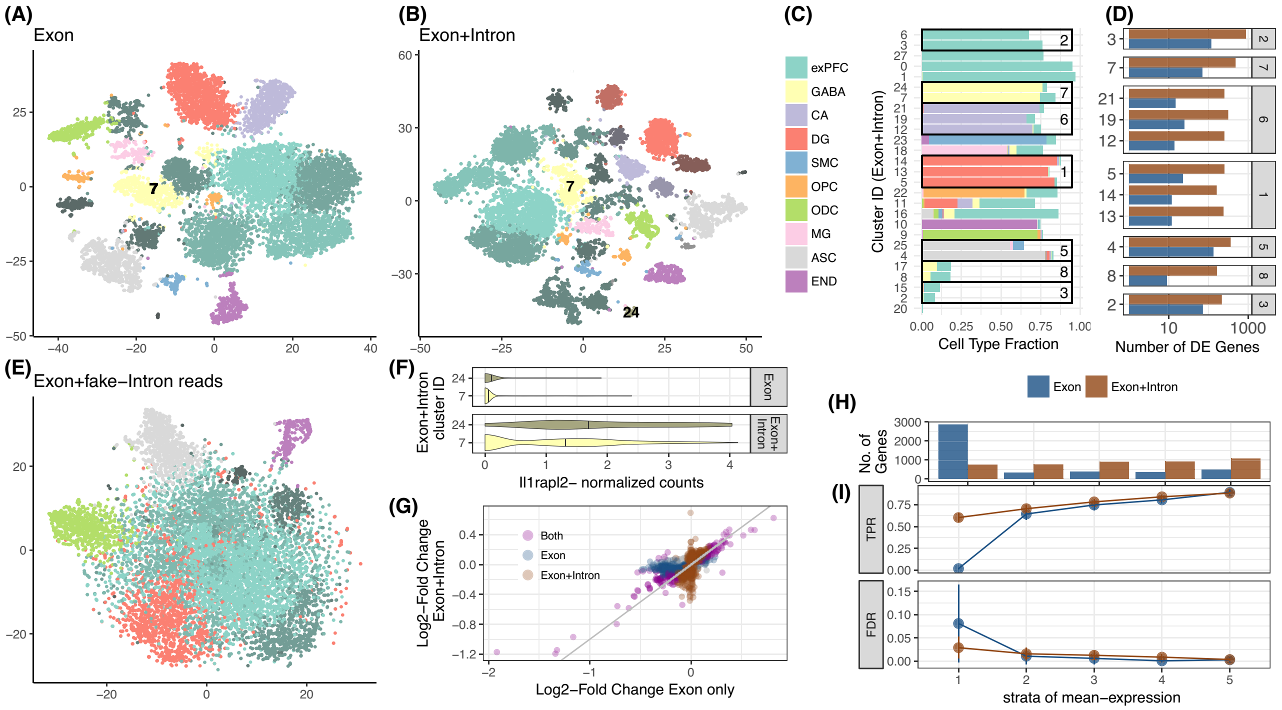

Figure 4: Contribution of intron reads to biological insights. We analyzed published single-nucleus RNA-seq data from mouse prefrontal cortex (PFC) and hippocampus [12] to assess the utility of counting intron in addition to exon reads. We processed the raw data with *zUMIs* to obtain expression tables with exon reads as well as exon+intron reads and then used the R-package Seurat [35, 36] to cluster cells. With exon counts, we identified 19 clusters **(A)**, and with exon+intron counts we identified 27 clusters **(B)**. Clusters are represented as t-SNE plots and colored according to the most frequent cell-type assignment in the original article [12]: glutamatergic neurons from the prefrontal cortex (exPFC), GABAergic interneurons (GABA), pyramidal neurons from the hippocampal CA region (CA), granule neurons from the hippocampal dentate gyrus region (DG), astrocytes (ASC), microglia (MG), oligodendrocytes (ODC), oligodendrocyte precursor cells (OPC), neuronal stem cells (NSC), smooth muscle cells (SMC) and endothelial cells (END). Different shades of those clusters indicate that multiple clusters had the same major cell type assigned. If we randomly sample counts from the intron data and add them to the exon counting, the noise reduces the number of clusters and the Seurat pipeline can only identify 9–11 clusters **(E)**. The composition of each cluster based on exon+intron is detailed in panel **(C)**, and cells that were not assigned a cell type in [12] are displayed as empty. The boxes mark the clusters that were not split when using exon data only. For example, cluster 7 from exon counting, which mainly consists of GABAergic neurons, was split into clusters 7, 24 (506, 66 cells) when using exon+intron counting. In **(D)**, we show the numbers of genes that were differentially expressed (DE) (limma p-adj <0.05) between the clusters found only with exon+intron counts. The panel numbers represent the exon counting cluster numbers and the y-axis the exon+intron counting cluster number. The log2-fold changes corresponding to these contrasts are also used in **(G)**. Among the genes that were additionally detected to be DE by exon+intron counting was the marker gene Il1rapl2 (limma p-adj $= 10^{-5}$). In **(F)**, we present a violin plot of the normalized counts for Il1rapl2 in cells of the GABAergic subclusters 7 and 24. Log2-fold changes calculated with exon+intron counts correlate well with exon counts **(G)**. Note that for exon counting only, half as many genes could be evaluated as for exon+intron counting and thus only half of the exon+intron genes are depicted in **(G)**. Large $\log_2$ fold changes (LFCs) are found to be significant with both counting strategies (purple points are close to the bisecting line). We conducted simulations based on mean and dispersion measured using exon cluster 0 (1,616 cells, ~90% exPFC). In **(I)** we show the expected true positive rate and the false discovery rate for a scenario comparing 300 vs 300 cells. Results for exon and exon+intron counting were stratified into five quantiles according to the mean expression of genes, where stratum 1 contains lowly expressed genes and stratum 5 the most highly expressed genes. The numbers of genes falling into each of the bins using exon+intron and exon counting are depicted in **(H)**.

genic mapping fractions using the statistics output provided by *zUMIs*.

Conclusion

zUMIs is a fast and flexible pipeline for processing raw reads to obtain count tables for RNA-seq data using UMIs. To our knowledge, it is the only open source pipeline that has a BC and UMI quality filter, allows intron counting, and has an integrated downsampling functionality. These features ensure that *zUMIs* is applicable to most experimental designs of RNA-seq data, including single-nucleus sequencing techniques, droplet-based methods where the BC is unknown, as well as plate-based UMI-methods with known BCs. Finally, *zUMIs* is computationally efficient, user-friendly, and easy to install.

Methods

Analyzed RNA-seq datasets

HEK293T cells were cultured in DMEM high glucose with L-glutamine (Biowest) supplemented with 10% fetal bovine serum (Thermo Fisher) and 1% penicillin/streptomycin (Sigma-Aldrich) in a 37°C incubator with 5% carbon dioxide. Cells were passaged and split every 2 or 3 days. For single-cell RNA-seq, HEK293T cells were dissociated by incubation with 0.25% Trypsin (Sigma-Aldrich) for 5 minutes at 37°C. The single-cell suspension was washed twice with phosphate-buffered saline, and dead cells were stained with Zombie Yellow (Biolegend) according to the manufacturer’s protocol. Single cells were sorted into DNA LoBind 96-well polymerase chain reaction (PCR) plates (Eppendorf) containing lysis buffer with a Sony SH-800 cell sorter in 3-drop purity mode using a 100-µmnozzle. Next, single-cell RNA-seq libraries were constructed from one 96-well plate using a slightly modified version of the mcSCRB-seq protocol. Reverse transcription was performed as described previously [40], with the only change being the use of KAPA HiFi HotStart enzyme for

PCR amplification of cDNA. Resulting libraries were sequenced using an Illumina HiSeq1500 with 16 cycles in Read 1 to decode cell BCs (6 bases) and UMIs (10 bases) and 50 cycles in Read 2 to sequence into the cDNA fragment, obtaining ~227 million reads. Raw fastq files were processed using *zUMIs*, mapping to the human genome (hg38) and Ensembl gene models (GRCh38.84).

In addition, we analyzed data from 1.3 million mouse brain cells generated on the 10xGenomics Chromium platform [2]. Sequences were downloaded from the National Center for Biotechnology Information Sequence Read Archive under accession number SRP096558. The data consist of 30 billion read-pairs from 133 individual samples. In these data, read 1 contains 16 bp for the cell BC and 10 bp for the UMI and read 2 contains 114 bp of cDNA. *zUMIs* was run using default settings, and we allowed 7 threads per job for a total of up to 42 threads on an Intel Xeon E5-2699 22-core processor.

Finally, we obtained mouse brain DroNc-seq read data [12] from the Broad Institute Single Cell Portal [41]. This dataset consists of ~1,615 million read-pairs from ~11,000 single nuclei. Read 1 contains a 12 bpcell BC and a 8 bpUMI and read 2 60 bpof cDNA.

The two mouse datasets were mapped to genome version mm10 and applying Ensembl gene models (GRCm38.75).

Power simulations and DE analysis

We evaluated the power to detect differential expression with the help of the powsimR package [9]. For the DroNc-seq dataset, we estimated the parameters of the negative binomial distribution from one of the identified clusters, namely, cluster 0, compromising 1,500 glutamatergic neuronal cells from the prefrontal cortex (Fig.4D). Since we detect more genes with exon+intron counting (4,433 compared to 1,782), we included this phenomenon in our read count simulation by drawing mean expression values for a total of 4,433 genes. This means that the table includes sparse counts for the exon counting. Log_2-fold changes were drawn from a gamma distribution with shape equal to 1 and scale equal to 2. In each of the 25 simulation iterations, we draw an equal sample size of 300 cells per group and test for differential expression using limma-trend [42] on $\log_2$ counts per million (CPM) values with scran [43] library size correction. The true positive rate and FDR are stratified over the empirical mean expression quantile bins.

For the differential expression analysis between clusters, we use the same DE estimation procedure as in the simulations: scran normalization followed by limma-trend DE-analysis (c.f. [44]).

Cluster identification

After processing the DroNc-seq data [12] with zUMIs as described above, we cluster cells based on UMI counts derived from exons only and exons+introns reads using the Seurat pipeline [35, 36]. First, cells with fewer than 200 detected genes were filtered out. The filtered data were normalized using the LogNormalize function. We then scale the data by regressing out the effects of the number of transcripts and genes detected per cell using the ScaleData function. The normalized and scaled data are then used to identify the most variable genes by fitting a relationship between mean expression (ExpMean) and dispersion (LogVMR) using the FindVariableGenes function. The identified variable genes are used for principle component analysis, and the top 20 principle components are then used to find clusters using graph-based clustering as implemented in FindClusters. To illustrate that the additional clusters found by counting exon+intron reads are not spurious, we use intron-only UMI counts from the same data to add to the observed exon-only counts. More specifically, to each gene we add scran-size factor-corrected intron counts from the same gene after permuting them across cells. We assessed the cluster numbers from 100 such permutations.

Comparison of UMI collapsing strategies

In order to validate *zUMIs* and compare different UMI collapsing methods, we used the HEK dataset described above. We ran *zUMIs* (1) without quality filtering, (2) filtering for onebase under Phred 17, and (3) collapsing similar UMI sequences within a hamming distance of 1. To compare with other available tools, we ran the same dataset using the Drop-seq-tools version 1.13 [13] and quality filter “1 base under Phred 17” without edit distance collapsing. Last, the HEK dataset was used with UMI tools [25] in (1) “unique” and (2) “directional adjacency” mode with edit distance set to 1. Also, we compared the output of *zUMIs* from the DroNc-seq dataset when using default parameters (“1 base under Phred 20”) to UMI-tools in (1) “unique,” (2) “directional adjacency,” and (3) “cluster” settings. For each setting and tool combination, we compared per-cell/per-nuclei UMI contents in a linear model fit.

Availability of source code and requirements

- Project name: zUMIs
- Project home page: https://github.com/sdparekh/zUMIs
- Operating system(s): UNIX
- Programming language: shell, R, perl
- Other requirements: STAR >= 2.5.3a, R >= 3.4, Rsubread >= 1.26.1, pigz >= 2.3 & samtools >= 1.1
- License: GNU GPLv3.0
- Research Resource Identification Initiative ID: SCR_016139

Availability of supporting data

All data that were generated for this project were submitted to GEO under accession GSE99822. An archival copy of the source code and test data are available via the *GigaScience* repository GigaDB [45].

Abbreviations

BC: barcode; DE: differentially expressed; FDR: false discovery rate; MAD: median absolute deviation; PCR: polymerase chain reaction; PFC: prefrontal cortex; scRNA-seq: single-cell RNA sequencing; UMI: unique molecular identifier.

Competing interests

The authors declare that they have no competing interests.

Funding

This work has been supported by the Deutsche Forschungsgemeinschaft (DFG) through SFB1243 subprojects A14/A15.

Author contributions

S.P. and C.Z. designed and implemented the pipeline. B.V. tested the pipeline and helped in power simulations. All authors contributed to writing the manuscript.

References

1. Sandberg R. Entering the era of single-cell transcriptomics in biology and medicine. Nat Methods 2014;**11**(1):22–4.
2. Zheng GXY, Terry JM, Belgrader P, et al. Massively parallel digital transcriptional profiling of single cells. Nat Commun 2017;**8**:14049.
3. Rosenberg AB, Roco CM, Muscat RA, et al. Single-cell profiling of the developing mouse brain and spinal cord with split-pool barcoding. Science 2018;**360**(6385):176–82.
4. Wagner A, Regev A, Yosef N. Revealing the vectors of cellular identity with single-cell genomics. Nat Biotechnol 2016;**34**(11):1145–60.
5. Regev A, Teichmann SA, Lander ES, et al. The Human Cell Atlas. Elife, 2017; 6.
6. Parekh S, Ziegenhain C, Vieth B, et al. The impact of amplification on differential expression analyses by RNA-seq. Sci Rep 2016;**6**:25533.
7. Kivioja T, Vähärautio A, Karlsson K, et al. Counting absolute numbers of molecules using unique molecular identifiers. Nat Methods 2012;**9**(1):72–4.
8. Ziegenhain C, Vieth B, Parekh S, et al. Comparative analysis of single-cell RNA sequencing methods. Mol Cell 2017;**65**(4):631–43.e4.
9. Vieth B, Ziegenhain C, Parekh S, et al. powsimR: power analysis for bulk and single cell RNA-seq experiments. Bioinformatics 2017;**33**(21):3486–3488.
10. Ziegenhain C, Vieth B, Parekh, et al. Quantitative single-cell transcriptomics. Brief Funct Genomics 2018; doi:10.1093/bfgp/ely009.
11. Lake BB, Ai R, Kaeser GE, et al. Neuronal subtypes and diversity revealed by single-nucleus RNA sequencing of the human brain. Science 2016;**352**(6293):1586–90.
12. Habib N, Avraham-Davidi I, Basu A, et al. Massively parallel single-nucleus RNA-seq with DroNc-seq. Nat Methods 2017;**14**(10):955–8.
13. Macosko EZ, Basu A, Satija R, et al. Highly parallel genome-wide expression profiling of individual cells using nanoliter droplets. Cell 2015;**161**(5):1202–14.
14. Svensson V, Natarajan KN, Ly LH, et al. Power analysis of single-cell RNA-sequencing experiments. Nat Methods 2017;**14**(4):381–7.
15. Hashimshony T, Senderovich N, Avital G, et al. CEL-Seq2: sensitive highly-multiplexed single-cell RNA-seq. Genome Biol 2016;**17**(1):77.
16. Petukhov V, Guo J, Baryawno N, et al. Accurate estimation of molecular counts in droplet-based single-cell RNA-seq experiments. bioRxiv 2017;p. 171496.
17. Soumillon M, Cacchiarelli D, Semrau S, et al. Characterization of directed differentiation by high-throughput single-cell RNA-seq. bioRxiv 2014.
18. Jaitin DA, Kenigsberg E, Keren-Shaul H, et al. Massively parallel single-cell RNA-seq for marker-free decomposition of tissues into cell types. Science 2014;**343**(6172):776–9.
19. Klein AM, Mazutis L, Akartuna I, et al. Droplet barcoding for single-cell transcriptomics applied to embryonic stem cells. Cell 2015;**161**(5):1187–1201.
20. Zilionis R, Nainys J, Veres A, et al. Single-cell barcoding and sequencing using droplet microfluidics. Nat Protoc 2017;**12**(1):44–73.
21. Hochgerner H, Lönnerberg P, Hodge R, et al. STRT-seq-2i: dual-index 5' single cell and nucleus RNA-seq on an addressable microwell array. Sci Rep 2017;**7**(1):16327.
22. Dobin A, Davis CA, Schlesinger F, et al. STAR: ultrafast universal RNA-seq aligner. Bioinformatics 2013;**29**(1):15–21.
23. Liao Y, Smyth GK, Shi W. featureCounts: an efficient general purpose program for assigning sequence reads to genomic features. Bioinformatics 2014;**30**(7):923–30.
24. Dowle M, Srinivasan A. data.table: Extension of 'data.frame." 2017, https://CRAN.R-project.org/package=data.table, r package version 1.10.4.
25. Smith TS, Heger A, Sudbery I. UMI-tools: modelling sequencing errors in unique molecular identifiers to improve quantification accuracy. Genome Res. 2017.
26. Fraley C, Raftery AE. Model-based clustering, discriminant analysis, and density estimation. J Am Stat Assoc 2002;**97**(458):611–31.
27. Fraley C, Raftery AE Enhanced Model-Based Clustering, Density Estimation and Discriminant Analysis Software: MCLUST. , J. Classification , 2003, **20**, 263–286.
28. Vallejos CA, Risso D, Scialdone A, et al. Normalizing single-cell RNA sequencing data: challenges and opportunities. Nat Methods 2017;**14**(6):565–71.
29. Evans C, Hardin J, Stoebel DM. Selecting between-sample RNA-seq normalization methods from the perspective of their assumptions. Brief Bioinform 2017.
30. Grün D, van Oudenaarden A. Design and analysis of single-cell sequencing experiments. Cell 2015;**163**(4):799–810.
31. Hendriks GJ, Gaidatzis D, Aeschimann F, et al. Extensive oscillatory gene expression during *C. elegans* larval development. Mol Cell 2014;**53**(3):380–92.
32. Gaidatzis D, Burger L, Florescu M, et al. Analysis of intronic and exonic reads in RNA-seq data characterizes transcriptional and post-transcriptional regulation. Nat Biotechnol 2015;**33**(7):722–9.
33. La Manno G, Soldatov R, Hochgerner H, et al. RNA velocity in single cells. bioRxiv 2017;p. 206052.
34. Lake BB, Codeluppi S, Yung YC, et al. A comparative strategy for single-nucleus and single-cell transcriptomes confirms accuracy in predicted cell-type expression from nuclear RNA. Sci Rep 2017;**7**(1):6031.
35. Satija R, Farrell JA, Gennert D, et al. Spatial reconstruction of single-cell gene expression data. Nat Biotechnol 2015;**33**(5):495–502.
36. Butler A, Satija R. Integrated analysis of single cell transcriptomic data across conditions, technologies, and species. bioRxiv 2017;p. 164889.
37. Tasic B, Menon V, Nguyen TN, et al. Adult mouse cortical cell taxonomy revealed by single cell transcriptomics. Nat Neurosci 2016;**19**(2):335–46.
38. The Tabula Muris Consortium, Quake SR, Wyss-Coray T, et al. Single-cell transcriptomic characterization of 20 organs and tissues from individual mice creates a Tabula Muris. bioRxiv 2018;p. 237446.
39. Han X, Wang R, Zhou Y, et al. Mapping the mouse cell atlas by microwell-seq. Cell 2018;**172**(5):1091–1107.e17.
40. Bagnoli JW, Ziegenhain C, Janjic A, et al. mcSCRB-seq: sensitive and powerful single-cell RNA sequencing. bioRxiv 2017;p. 188367.
41. Broad Institute Single Cell Portal. https://portals.broadinstitute.org/single_cell/study/dronc-seq-single-nucleus-rna-seq-on-mouse-archived-brain.

42. Law CW, Chen Y, Shi W, et al. voom: precision weights unlock linear model analysis tools for RNA-seq read counts. Genome Biol 2014;**15**(2):R29.
43. Lun ATL, McCarthy DJ, Marioni JC. A step-by-step workflow for low-level analysis of single-cell RNA-seq data with Bioconductor. F1000Res 2016;**5**.
44. Soneson C, Robinson MD. Bias, robustness and scalability in single-cell differential expression analysis. Nat Methods 2018;**15**(4):255–61.
45. Parekh S, Ziegenhain C, Vieth B, et al. Supporting data for 'zUMIs - A fast and flexible pipeline to process RNA sequencing data with UMIs.' GigaScience Database 2018;http://dx.doi.org/10.5524/100447.
46. Grün D, Kester L, van Oudenaarden A. Validation of noise models for single-cell transcriptomics. Nat Methods 2014;**11**(6):637–40.
47. Tian L, Su S, Amann-Zalcenstein D, et al. scPipe: a flexible data preprocessing pipeline for single-cell RNA-sequencing data. bioRxiv 2017;p. 175927.
48. Islam S, Zeisel A, Joost S, et al. Quantitative single-cell RNA-seq with unique molecular identifiers. Nat Methods 2014;**11**(2):163–6.