GigaScience

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

--Manuscript Draft--

Manuscript Number:	GIGA-D-17-00271R1
Full Title:	zUMIs - A fast and flexible pipeline to process RNA sequencing data with UMIs
Article Type:	Technical Note
Funding Information:	Deutsche Forschungsgemeinschaft (SFB1243 - A15) — Dr. Ines Hellmann Deutsche Forschungsgemeinschaft (SFB1243 - A14) — Prof. Wolfgang Enard
Abstract:	Single cell RNA-seq (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 barcodes (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. zUMIs is such a pipeline, it can handle both known and random BCs and also efficiently collapses UMIs, either just for exon mapping reads or for both exon and intron mapping reads. Another unique feature of zUMIs is the adaptive downsampling function, that facilitates dealing with hugely varying library sizes, but also allows to evaluate whether the library has been sequenced to saturation. zUMIs flexibility allows to accommodate data generated with any of the major scRNA-seq protocols that use BCs and UMIs. To illustrate the utility of zUMIs, we analysed a single-nucleus RNA-seq dataset and show that more than 35% of all reads map to introns. We furthermore show that these intronic reads are informative about expression levels, significantly increasing the number of detected genes and improving the cluster resolution.
Corresponding Author:	Ines Hellmann Ludwig-Maximilians-Universitat Munchen Fakultat fur Biologie Martinsried, GERMANY
Corresponding Author Secondary Information:	
Corresponding Author's Institution:	Ludwig-Maximilians-Universitat Munchen Fakultat fur Biologie
Corresponding Author's Secondary Institution:	
First Author:	Swati Parekh
First Author Secondary Information:	
Order of Authors:	Swati Parekh Christoph Ziegenhain Beate Vieth Wolfgang Enard Ines Hellmann
Order of Authors Secondary Information:	
Response to Reviewers:	In particular, both reviewers feel that some of your results that have been achieved by simulation need to be backed up with an analysis of real data (reviewer 1, #2; reviewer 2, #6). I also agree with reviewer 2 that it is important to compare the performance of (parts of) your pipeline with existing tools that perform steps in the zUMI pipeline. --- AUTHOR RESPONSE: Thank you for the useful comments. We have now backed up the comparison of UMI collapsing approaches by analysis of real data. To this end, we have added descriptive

statistics plots in a new Figure 2. This data also shows how well gene expression estimates correspond between published pipelines and zUMIs. Furthermore, we have added a Table showing presence or absence of important features in existing tools and zUMIs.

Reviewer 2's comments regarding technical and biological sources of variation (#2 in the report) is another crucial point that needs careful consideration when you are preparing a revised submission.

AUTHOR RESPONSE:
We have added a deeper analysis of the DroNc-seq dataset to show more in detail the biological relevance of adding Intronic counts (see detailed answer in the Response to Reviewers) below. Additionally, we show that the Intronic counts are not artifacts by sampling fake random Intronic reads and showing that this actually decreases cluster resolution (for more details see response to Reviewer 1, comment 3).

It is important that the description of your methods allows full reproducibility - please include missing details, as outlined by our reviewers.

AUTHOR RESPONSE:
We have added a detailed Methods section in the paper to carefully describe the datasets and analysis strategies.

Reviewer #1: Parekh and coworkers introduce a pipeline to process high throughput scRNA-seq data consisting of cell barcodes and UMIs. This is an open-source software that also supports features such as reads downsampling and Intron counting - the latter is important for single nucleus RNA-seq data. Overall, this is a useful study and I would like to support its publication, but the current manuscript could greatly benefit with additional biological analysis (related to Fig 4) that would prompt users to take notice. I would like to support its publication, contingent on the authors addressing the following comments.

1.The pipeline appears to collapse only identical UMIs. We have found in our experience that this can lead to overcounting of transcripts, and that it is necessary to collapse UMIs mapping to the same gene in the same cell within an edit distance of 1. I would be curious to see how this impacts the number of transcripts detected per cell.

AUTHOR RESPONSE:
zUMIs now also offers the option to collapse UMIs based on Hamming distance and add a plot comparing the number of UMIs/cell for 4 different approaches (updated Figure 1). We also extended the text accordingly:

“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 the two zUMIs UMI-collapsing options to the recommended directional adjacency approach implemented in UMI-tools [15], using our in-house example dataset (see Methods). zUMIs identity collapsing yields nearly identical UMI counts per cell as UMI-tools, while Hamming distance yields increasingly fewer UMIs/cell with increasing sequencing depth (Figure 2C). Smith et al. [15] suggest that edit distance collapsing without considering the relative frequencies of UMIs might indeed overreach and over-collapse 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 reduced to one, the minimal number after collapsing (Figure 2D).
However, note that the above described differences are minor. By and large, there is good agreement between UMI counts obtained by UMI-tools [15], the Drop-seq

pipeline [24] and zUMIs. The correlation between gene-wise counts of the same cell is > 0.99 for all comparisons (Figure 2B). In light of this, we would consider the > 3 times higher processing speed of zUMIs a decisive advantage (Figure 2A)“

2.The authors use simulations to describe the impact of Intron counting on differential expression. I would instead like to see this on real data. In particular I would like to see examples of "before/after" plots (e.g. violin plots/heatmaps) of specific genes that (1) were called out as differentially expressed (DE) but no longer are once introns are incorporated, (2) the reverse of (1), and (3) those that remain DE but with significantly different statistical significances.

AUTHOR RESPONSE:
To analyse real data we use the DroNc-seq data from Habib et al. (2017). We analyse the log2 fold changes (LFC) for the groups that were split up more when using Exon+Intron counting (see the new Figure 4F) and we added a description of our findings to the main text.

“Following the Seurat pipeline to cluster cells [30, 31], we find that using Exon+Intron counts discriminates 28 clusters, while we could only discriminate 19 clusters using Exon counts (Figure 4A+B). We then continue to further characterize the 7 clusters that were further subdivided by the addition of Intron counts (Figure 4D). First, we identify differentially expressed (DE) genes between the newly formed clusters. If we count only Exon reads there appears on average only 10 genes to be DE between the sub-groups, while Exon+Intron counting yields ~10x more DE genes, thus corroborating the signal found with clustering. The log2-fold changes estimated for the 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 (Figure 4F).“

3.I am also not convinced of the result claiming more clusters when introns are included. What is the evidence that these clusters are not spurious? The detection of additional clusters is not evidence enough that these are real. It would be useful to show a heatmap demonstrating that there is true, biologically significant differential expression between the novel clusters detected by the Intron counting.

AUTHOR RESPONSE:
We now added plots with the numbers of DE genes distinguishing the newly split clusters for both Exon+Intron as well as Exon counting (Figure 4D). Note that with the number of DE genes also the more informative marker genes such as the example in Figure 4E are detected. Thus, even though we do not fully understand the biological meaning of the more fine grained clusters, we are confident that they are indeed based on a biological signal from the RNA-seq data. Additionally, we demonstrate this, by sampling randomly from the distribution of intronic counts and adding to the exonic counts. The resulting extra-noise in fact leads to a lower number of clusters detected: 19 Exon 28 Intron+Exon 7 fake Intron+Exon (see plot in attached “Additional File 1”).

4.For completeness, I would like if the authors could include a section comparing their "exon-counts" matrix with the count matrix produced by either the cellranger or Drop-seq-tools for datasets that have been classically analyzed by the latter methods. This would produce some confidence in the base reproducibility of the methods. If on the other hand, zUMIs produces a different Exon count matrix, then the authors must explain why this is the case.

AUTHOR RESPONSE:
Unfortunately, we could not run the cellranger pipeline on our example dataset, because it does not allow to freely specify cell barcodes. Instead we compare to the Drop-seq and the UMI-tools pipelines. We generally find a high correlation between the number of UMIs per gene detected in a cell. The slight discrepancies between zUMIs

and the Drop-seq-tools are due to how reads are associated with genes. For example zUMIs does not count ambiguously mapped reads, i.e. reads that overlap with multiple genes, while Drop-seq counts them for all genes.
UMI-tools on the other hand also uses featureCounts for read association, however their recommended method to collapse UMIs by directional adjacency with edit distance 1 differs from the options in zUMIs. Here, our newly added feature of collapsing UMIs Hamming distance yields as expected the most similar counts. These results are now included in Figure 2C.

5.In Figure 4, the authors show to show a confusion matrix to compare how clusters in A map to clusters in B. Also for those clusters that multi-map (i.e. those resolved by intron-Exon mapping but not by exon-mapping alone), is there biologically meaningful differential expression? Some examples of specific cell types and their gene expression differences in A vs B would be very informative.

AUTHOR RESPONSE:
We added an example for a subsplit of a mainly GABAergic cluster that also has significantly DE Marker gene for Pvalb GABAergic neurons when considering Intron+Exon counts in Figure 4E and discuss this in the main text:

“Having a closer look at cluster 7, it was split into a bigger (7) and a smaller cluster (24) using exon+intron counting (Figure 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, suggesting that the split of cluster 7 might be biological meaningful (Figure 4E).”

Reviewer #2: Review of "zUMIs - A fast and flexible pipeline to process RNA sequencing data with UMIs"

Summary:
Parekh et al. describe a computational pipeline to preprocess single-cell RNA-seq data that contains UMIs and cell barcodes. The main components of the pipeline include sequence quality filtering of UMIs and barcodes, a wrapper to call the mapping software STAR, selection of cell barcodes, and downsampling of reads to lower library size. While other tools exist that perform all of these steps either all together or individually for one or more platforms, the novelty of zUMIs is that it performs all of these steps at once for data from any UMI platform. Such a tool would likely be useful for the single-cell community, however many methodological details are missing. In addition the manuscript could benefit from additional comparison to existing tools.

The authors also argue that in general quantification of gene expression should incorporate intron-mapping reads, a task which is enabled by the use of their software. However, I have reservations about the evidence upon which this conclusion is based.

AUTHOR RESPONSE:
We want to clarify that we do not wish to claim that counting introns is a good idea in general. However, we argue that for extremely sparse datasets such as generated by single nuclei sequencing, having Intron counts is better than losing even more genes. We hope that we could make this clearer in the text, as such:

“Furthermore, we think that although noisy, the large number of additionally detected genes makes Exon+Intron counting worthwhile for extremely sparse data.”

I have identified several issues that the authors should address in order to improve the manuscript, which are detailed below and divided into major (of critical importance) and minor (to improve clarity) categories.

Major Comments:
1.Methodological details are missing throughout. The method should be described more completely and clearly, and any analyses or comparisons should be made reproducible. For example:
*What differential expression method was used in the simulation study to compare UMItools and zUMI?
*What options were used with powsimR in the simulation study?
*How is the k-dimensional multivariate normal distribution fit in the cell barcode selection step?
*How is k determined in the cell barcode selection step?
*How was data simulated for the Intron evaluation?
*What options were used in applying the Seurat pipeline to cluster cells?

AUTHOR RESPONSE:
We added a methods section (Page:3-4) that includes subsections for (1) data generation of the HEK dataset as well as data processing of other used datasets, (2) the powsimR simulations and (3) the use of the Seurat pipeline. The passage about the Cell-Barcode selection was changed in the main text (Page:2). We hope to have made our barcode selection clearer in the main text.

“To this end, we fit a k-dimensional multivariate normal distribution using the R-package mclust [25, 26] for the number of reads/BC, where k is empirically determined by mclust via the Bayesian Information Criterion (BIC). We reason that only the kth normal distribution with the largest mean contains barcodes that identify reads originating from intact cells.”

2.One major conclusion of the paper is that incorporation of intron-mapping reads significantly improves cluster resolution. It is perhaps not surprising that including the Intron counts results in a higher mean number of genes detected, but the authors conclude that since more clusters are also found that this means the additional reads are biologically meaningful. Unfortunately, the authors have not provided any evidence that this is the case. The fact that more clusters are seen says nothing about the difference between technical and biological sources of variation. If these additional clusters also corresponded to some independently measured biological covariate, the argument would have basis.

AUTHOR RESPONSE:
While we do not wish to claim that counting of intron-mapping reads is recommended in all cases of scRNA-seq, we do think it is valid and helpful for extremely sparse datasets such as the DroNc-seq data from Habib et al. (2017). We now provide detailed analyses of differences between newly formed subclusters using Exon+Intron counting. We find not only more genes, but also more significantly differentially expressed genes between subclusters when using Exon+Intron UMI data (Figure 4D). Furthermore, log2 fold changes (LFC) for the groups that were split up more when using Exon+Intron counting corresponded well to the Exon-only LFC (see the new Figure 4F). Additionally, we illustrate the biological relevance of subclusters found with Exon+Intron data by the example of the transcriptomic subtypes of GABAergic Pvalb-type Neurons marked by Il1rapl2 expression. We have added this evidence to the ‘Intron Counting’ section and included methodological details in the appropriate Methods sections.
Lastly, we have excluded the possibility of Intron-mapping reads being spurious by sampling fake intronic reads and attempting cluster identification (see response to Reviewer 1, point 3).

3.Many central conclusions of the article were made based on an analysis of a dataset of 96 cells that is never described. It is referred to as "the HEK dataset" throughout the manuscript, but no citation, details of data generation, or description of the experimental design is given.

AUTHOR RESPONSE:
We added this information to the new Method section (Page:3-4).

4.Several open-source tools exist that perform many of the steps in the zUMI pipeline [1, 2, 3]. It would be nice to see how these perform in comparison to zUMI.

AUTHOR RESPONSE:
While several tools exist that can perform some of the steps of the zUMIs pipeline, none of them provides a comprehensive combination as zUMIs. We have added a Table to compare available features of six other pipelines geared towards scRNA-seq data with UMIs. The tool “UMI-Reducer” with reference [2] suggested by the reviewer was omitted because it seemed like a tool geared towards one specific application outside of single-cell RNA-seq. Furthermore, “UMI-Reducer” only de-duplicates UMIs with the same mapping position, which would be inappropriate for scRNA-seq protocols that fragment after preamplification, such as SCRB-seq.
Furthermore, we added a comparison of the count-tables produced by zUMIs, Drop-seq-tools and UMI-tools and generally find very good correspondence (see response to Reviewer 1).

5.The conclusion that a UMI distance filter (using UMI-tools) is unnecessary is only based on a single simulated dataset of up to 90 cells per condition. It is also based on a single metric (power to identify differentially expressed genes in simulated data). If we are only interested in differential expression analyses, this might be a reasonable metric. However to be widely applicable to the analysis of single cell RNA-seq, the authors should consider additional metrics such as replicate reproducibility, number of detected genes, etc. The authors should also consider additional datasets.

AUTHOR RESPONSE:
We substantially extended our comparison of different UMI-collapsing method. In Fig. 2 B,C, we also compare the correlation of gene expression values and numbers of detected UMIs per cell between various different filtering methods and find that there is generally a high consensus among all UMI collapsing methods in our HEK example dataset. An analysis of the DroNc-seq data gave basically the same results (see plot in attached “Additional File 1”).
Furthermore, we added the possibility to collapse UMIs with a specified Hamming-distance to zUMIs, giving users more choice over UMI filtering. All these new analysis are also described in the section “Transcript Counting” of the main text.

6.It is not clear how the simulation parameters in the comparison to UMI-tools directly relate to the UMI quantification. Specifically, estimating the mean and dispersion of the processed data and then using these as the basis for a simulated dataset seems pretty far removed from the observed UMI counts. The authors should also investigate differences in differential expression analysis of the actual data (not simulated data). They could also generate a simulated null comparison by randomly permuting sample labels. The same comments hold for the second simulation (evaluating Intron count inclusion).

AUTHOR RESPONSE:
We removed the simulations from the description of UMI-collapsing methods and focus our reporting on the descriptive statistics suggested by the reviewer (Figure 2 & section “Transcript counting”).

Minor Comments:
1.The results of the simulation evaluating Intron usage are summarized broadly in the text, but the specific results are not shown. For example what does "power to detect

differentially expressed genes was similar for the Exon and Exon+Intron counts" mean? How similar? What were the values?

AUTHOR RESPONSE:
This is now better described in the main text (Page 3 Passage:Intron Counting) along with specific settings for the powsimR package listed in the method section. Additionally, power simulation results are shown in Figure 4 with the true positive rate (TPR) and false discovery rate (FDR) shown for 5 stratas of gene expression (Figure 4G). Furthermore, we display the number of genes per stratum for Exon and Exon+Intron counting (Figure 4H).

2.The pipeline requires the user to specify many parameters for each step, however the implementation is run with one command. This means that if a user wants to change a single parameter in one of the later steps, they would still have to rerun the entire pipeline, wasting time and computational resources. It would be useful if the pipeline could alternatively be run as a series of individual steps so that the same exact steps don't need to be carried out multiple times in these situations.

AUTHOR RESPONSE:
This feature is implemented as “-w” option. One can invoke zUMIs at any step, eg to just re-run the counting of gene expression the user can give “-w counting”.

3.In the cell barcode selection step, the authors state that they remove "all barcodes that fall in the lower 1% tail of this distribution." What is the justification for this? What does this correspond to in practice? This threshold should also be denoted in Figure 3A.

AUTHOR RESPONSE:
The blue line in figure 3A corresponds to the calculated read cut-off. The normal distribution identified by mclust with the highest mean number of reads contains actual cell barcodes. Thus, setting the read cut-off to the lower 1% of this distribution is an empirical value that gives good correspondence to the known cell-barcodes for the HEK dataset (cut-off value: 52634 reads/barcode) and gave similarly good results for the DroNc-seq data analysed here. Still, in practice we recommend to always look at the elbow-plots output by zUMIs (Figure 3B). This will show whether our empirical cut-off was also valid for the dataset at hand.

4.What are the practical guidelines for downsampling? How should it be used in practice to normalize for sequencing depth?

AUTHOR RESPONSE:
We found the downsampling function extremely useful for method comparisons as we showed in our previous study (Ziegenhain et al. 2017). This also allows to evaluate whether the single cell libraries were sequenced to saturation (Figure 3D). For normalization purposes, the built-in MAD cut-offs as indicated by the dashed red lines in Figure 3C should be sufficient.

5.In the documentation online, section on cell barcode selection (here: https://github.com/sdparekh/zUMIs/wiki/Cell-barcodes-selection), Figure A is contradictory to Figure 3A in the manuscript. Specifically, the online documentation says "cells left to the blue line are selected" and the manuscript says "cell barcodes with reads above the blue line are selected."

AUTHOR RESPONSE:

	This was indeed a mistake and we corrected it on GitHub. --- 6.As a main advantage of zUMIs is the ability to apply on any UMI platform, the documentation should clearly state how to use the software in each case. Currently, this is unclear, as for example in the case of the "-c" option the wiki on GitHub (https://github.com/sdparekh/zUMIs/wiki/Usage) states that "For STRT-seq/InDrops give this as 1-n where n is your first cell barcode(-f) length." But it also states in the very next line "For InDrops give this as 1-n where n is the total length of cell barcode(e.g. 1-22)," which is contradictory to what the previous line states about InDrops. --- AUTHOR RESPONSE: This was indeed a mistake and we corrected it on GitHub. --- References: [1] Luyi Tian, Shian Su, Daniela Amann-Zalcenstein, Christine Biben, Shalin H. Naik, Matthew E. Ritchie. scPipe: a flexible data preprocessing pipeline for single-cell RNA-sequencing data. bioRxiv 175927; doi: https://doi.org/10.1101/175927 [2] Serghei Mangul, Sarah Van Driesche, Lana S. Martin, Kelsey C. Martin, Eleazar Eskin. UMI-Reducer: Collapsing duplicate sequencing reads via Unique Molecular Identifiers. bioRxiv 103267; doi: https://doi.org/10.1101/103267 [3] Viktor Petukhov, Jimin Guo, Ninib Baryawno, Nicolas Severe, David Scadden, Maria G. Samsonova, Peter V. Kharchenko. Accurate estimation of molecular counts in droplet-based single-cell RNA-seq experiments. bioRxiv 171496; doi: https://doi.org/10.1101/171496
Additional Information:	
Question	**Response**
Are you submitting this manuscript to a special series or article collection?	No
Experimental design and statistics Full details of the experimental design and statistical methods used should be given in the Methods section, as detailed in our Minimum Standards Reporting Checklist. Information essential to interpreting the data presented should be made available in the figure legends. Have you included all the information requested in your manuscript?	Yes
Resources A description of all resources used, including antibodies, cell lines, animals and software tools, with enough information to allow them to be uniquely identified, should be included in the Methods section. Authors are strongly encouraged to cite Research Resource Identifiers (RRIDs) for antibodies, model	Yes

organisms and tools, where possible. Have you included the information requested as detailed in our Minimum Standards Reporting Checklist?	
Availability of data and materials All datasets and code on which the conclusions of the paper rely must be either included in your submission or deposited in publicly available repositories (where available and ethically appropriate), referencing such data using a unique identifier in the references and in the “Availability of Data and Materials” section of your manuscript. Have you have met the above requirement as detailed in our Minimum Standards Reporting Checklist?	Yes

GigaScience, 2017, 1–9

doi: 10.1101/153940
Technical Note

TECHNICAL NOTE

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

Swati Parekh[1,*,†], Christoph Ziegenhain[1,†], Beate Vieth[1], Wolfgang Enard[1] and Ines Hellmann[1,*]

[1]Anthropology & Human Genomics, Department of Biology II, Ludwig-Maximilians University, 82152 Martinsried, Germany

[*]parekh@bio.lmu.de; hellmann@bio.lmu.de
[†]Contributed equally.

Abstract

Single cell RNA-seq (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 barcodes (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. *zUMIs* is such a pipeline, it can handle both known and random BCs and also efficiently collapses UMIs, either just for Exon mapping reads or for both Exon and Intron mapping reads. Another unique feature of *zUMIs* is the adaptive downsampling function, that facilitates dealing with hugely varying library sizes, but also allows to evaluate whether the library has been sequenced to saturation. *zUMIs* flexibility allows to accommodate data generated with any of the major scRNA-seq protocols that use BCs and UMIs. To illustrate the utility of *zUMIs*, we analysed a single-nucleus RNA-seq dataset and show that more than 35% of all reads map to Introns. We furthermore show that these intronic reads are informative about expression levels, significantly increasing the number of detected genes and improving the cluster resolution. **Availability:** https://github.com/sdparekh/zUMIs

Key words: Single-Cell RNA-Sequencing, Digital Gene Expression, Unique Molecular Identifiers, Pipeline

Introduction

The recent development of increasingly sensitive protocols allows to generate RNA-seq libraries of single cells [1]. The throughput of such single-cell RNA-sequencing (scRNA-seq) protocols is rapidly increasing, enabling the profiling of tens of thousands of cells [2, 3] and opening exciting possibilities to analyse cellular identities [4, 5]. As the required amplification from such low 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 barcodes (BCs) to label all cDNA molecules of a single cell with a nucleotide sequence before library generation [10, 2]. This allows for early pooling, which further decreases amplification noise [6]. Additionally, 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 open source [10]. While Cell Ranger is open, it is exceedingly difficult to adapt the code to new or unknown sample barcodes and other library types. Other tools are specifically designed to work with one mapping algorithm and focus mainly on tran-

Compiled on: March 16, 2018.
Draft manuscript prepared by the author.

Key Points

- zUMIs processes UMI-based RNA-seq data from raw reads to count tables in one command.
- Unique features of zUMIs:
  - Automatic cell barcode selection
  - Adaptive downsampling
  - Counting of Intron mapping reads for gene expression quantification
- zUMIs is compatible with all major UMI-based RNA-seq library protocols.

scriptomes [13, 14]. Furthermore, the only other UMI-RNA-seq pipeline providing the utility to also consider Intron mapping reads, dropEst [15], is only applicable to droplet-based protocols. Here, we present *zUMIs*, a fast and flexible pipeline that overcomes these limitations.

Findings

Here we describe *zUMIs*, a pipeline to process RNA-seq data that were multiplexed using cell barcodes 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 (Figure 1). Next, *zUMIs* generates UMI and read count tables for Exon and Exon+Intron counting. We reason that especially 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 [16, 17, 10, 14, 3, 2, 12] as well as protocols that provide UMI and BC in separate reads [18, 19, 20]. 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 (Figure 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` [21]. 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` [22] is used to first assign reads to Exons and afterwards 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` [23], 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 to properly collapse 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 undercollapsed due to sequence errors, *zUMIs* provides the option to collapse UMIs within a given Hamming distance. We compare the two *zUMIs* UMI-collapsing options to the recommended directional adjacency approach implemented in UMI-tools [24], using our in-house example dataset (see Methods). *zUMIs* identity collapsing yields nearly identical UMI counts per cell as UMI-tools, while Hamming distance yields increasingly fewer UMIs/cell with increasing sequencing depth (Figure 2C). Smith et al. [24] suggest that edit distance collapsing without considering the relative frequencies of UMIs might indeed overreach and over-collapse 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 (Figure 2D).

However, note that the above described differences are minor. By and large, there is good agreement between UMI counts obtained by UMI-tools [24], the Drop-seq pipeline [10] and *zUMIs*. The correlation between gene-wise counts of the same cell is > 0.99 for all comparisons (Figure 2B). In light of this, we would consider the > 3 times higher processing speed of *zUMIs* a decisive advantage (Figure 2A).

Cell Barcode 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 barcodes mark good cells from the data (Figure 3 A,B). To this end, we fit a k-dimensional multivariate normal distribution using the R-package mclust [25, 26] for the number of reads/BC, where k is empirically determined by mclust via the Bayesian Information Criterion (BIC). We reason that only the kth normal distribution with the largest mean contains barcodes that identify reads originating from intact cells. We exclude all barcodes that fall in the lower 1% tail of this kth normal-distribution to exclude spurious barcodes. The HEK dataset used in this paper contains 96 cells with known barcodes and *zUMIs* identifies 99 barcodes as intact, including all the 96 known barcodes. 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, if the number of barcodes or barcode sequences are known, it is preferable to use this information. In the case that *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 [27, 28]. A straight-forward solution for this issue is to downsample over-represented libraries [29]. *zUMIs* has an inbuilt function for downsampling datasets to a user-specified number of reads or a range of reads. By default, *zUMIs* downsamples all selected barcodes to be within three absolute deviations from the median number of reads per barcode (Figure 3 C). 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 less reads were present. Furthermore, *zUMIs* also allows to specify 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 of at one million reads, sequencing double that amount would only increase the number of detected genes from 9,000 to 10,600, when counting Exon reads (Figure 3D). In line with previous findings [8, 13], the saturation curve of Exon+Intron counting runs parallel to the one for Exon counting, both indicating that a sequencing depth of one 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, 6 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 (Figure 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 barely 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 (Figure 3 E). We also observe that the peak RAM usage for processing datasets of 227, 500 and 1000 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) on a 22-core processor in only 7 days.

Intron Counting

Assuming that Intron mapping reads originate from nascent mRNAs, *zUMIs* also counts and collapses Intron mapping reads with 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). For the $\sim 11,000$ 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 significantly from 1041 for Exon counts to 1995 for Exon+Intron counts. We then used the resulting UMI count tables to investigate whether Exon+Intron counting improves the identification of cell types, as suggested in [11]. Following the Seurat pipeline to cluster cells [30, 31], we find that using Exon+Intron counts discriminates 28 clusters, while we could only discriminate 19 clusters using Exon counts (Figure 4A+B). We then continue to further characterize the 7 clusters that were subdivided by the addition of Intron counts (Figure 4D). First, we identify differentially expressed (DE) genes between the newly formed clusters. If we count only Exon reads, there appear to be only 10 DE genes on average between the sub-groups, while Exon+Intron counting yields $\sim 10\times$ more DE genes, thus corroborating the signal found with clustering. The log2-fold changes estimated for the 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 (Figure 4F). Genes that are detected as DE in only one of our counting strategies have generally only very 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 [32] and find that $\sim 5\%$ of the additional genes are also marker genes. Having a closer look at cluster 7, it was split into a bigger (7) and a smaller cluster (24) using Exon+Intron counting (Figure 4A-C), we find one marker gene (Il1rapl2) to be DE between the sub-clusters 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 [32], suggesting that the split of cluster 7 might be biological meaningful (Figure 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 (~ 1500 cells) [9]. For a fair comparison, we include genes that were detected in sufficiently many cells for DE-analysis in either Exon or Exon+Intron counting. Thus, there are on average $4\times$ more genes in the lowest expression quantile for Exon counting than for Exon+Intron counting (Figure 4H). For those genes, expression is too spurious to be used for differential expression analysis, while 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 FDR (Figure 4G). In summary, the increased power for Exon+Intron counting and probably also the larger number of clusters is due to a better detection of lowly expressed genes. Furthermore, we think that, although noisy, the large number of additionally detected genes makes Exon+Intron counting worthwhile for extremely sparse data.

Conclusion

zUMIs is a fast and flexible pipeline 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 barcode 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

Analysed 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 % CO2. 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 PBS and dead cells stained with Zombie Yellow (Biolegend) according to the manufacturer's protocol. Single-cells were sorted into DNA LoBind 96-well PCR plates (Eppendorf) containing lysis buffer with a Sony SH-800 cell sorter in 3-drop purity mode using a 100 µm nozzle. 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 [33], 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 barcodes (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).

Furthermore, we anlysed data from 1.3 million mouse brain cells generated on the 10xGenomics Chromium platform [2]. Sequences were downloaded from the NCBI 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 barcode 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 (https://portals.broadinstitute.org/single_cell/study/dronc-seq-single-nucleus-rna-seq-on-mouse-archived-brain). This dataset consists of ~1615 million read pairs from ~ 11,000 single nuclei. Read 1 contains a 12bp cell barcode and a 8bp UMI and read 2 60bp of 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 1500 glutamatergic neuronal cells from the prefrontal cortex (Figure 4D). Since we detect more genes with Exon+Intron counting (4433 compared to 1782), we included this phenomenon also in our read count simulation by drawing mean expression values for a total of 4433 genes. This means that the table includes sparse counts for the Exon counting. Log2 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 [34] on log2 CPM values with scran [35] library size correction. The TPR 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 normaliztion followed by limma-trend DE-analysis.

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 [30, 31]. 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 (PCA) and the top 20 PCs are then used to find clusters using graph based clustering as implemented in 'FindClusters'.

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 1 base 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 [10] and quality filter "1 base under Phred 17" without edit distance collapsing. Lastly, the HEK dataset was used with UMI-tools [24] in (1) "unique" and (2) "directional adjacency" mode with edit distance set to 1. Furthermore, 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 and materials

All data that were generated for this project were submitted to GEO under accession GSE99822.

Declarations

List of Abbreviations

scRNA-seq - single-cell RNA-sequencing
UMI - Unique Molecular Identifier
BC - Barcode
MAD - Median Absolute Deviation

Competing Interests

The author(s) declare that they have no competing interests.

Funding

This work has been supported by the DFG through SFB1243 subprojects A14/A15.

Author's Contributions

SP and CZ designed and implemented the pipeline. BV 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 Jan;11(1):22–24.
2. Zheng GXY, Terry JM, Belgrader P, Ryvkin P, Bent ZW, Wilson R, et al. Massively parallel digital transcriptional profiling of single cells. Nat Commun 2017 16 Jan;8:14049.
3. Rosenberg AB, Roco CM, Muscat RA, Kuchina A, Sample P, Yao Z, et al. Single-cell profiling of the developing mouse brain and spinal cord with split-pool barcoding. Science 2018 Mar;p. eaam8999.
4. Wagner A, Regev A, Yosef N. Revealing the vectors of cellular identity with single-cell genomics. Nat Biotechnol 2016 8 Nov;34(11):1145–1160.
5. Regev A, Teichmann SA, Lander ES, Amit I, Benoist C, Birney E, et al. The Human Cell Atlas. Elife 2017 Dec;6.
6. Parekh S, Ziegenhain C, Vieth B, Enard W, Hellmann I. The impact of amplification on differential expression analyses by RNA-seq. Sci Rep 2016 9 May;6:25533.
7. Kivioja T, Vähärautio A, Karlsson K, Bonke M, Enge M, Linnarsson S, et al. Counting absolute numbers of molecules using unique molecular identifiers. Nat Methods 2012 Jan;9(1):72–74.
8. Ziegenhain C, Vieth B, Parekh S, Reinius B, Guillaumet-Adkins A, Smets M, et al. Comparative Analysis of Single-Cell RNA Sequencing Methods. Mol Cell 2017 16 Feb;65(4):631–643.e4.
9. Vieth B, Ziegenhain C, Parekh S, Enard W, Hellmann I. powsimR: Power analysis for bulk and single cell RNA-seq experiments. Bioinformatics 2017 Jul;.
10. Macosko EZ, Basu A, Satija R, Nemesh J, Shekhar K, Goldman M, et al. Highly Parallel Genome-wide Expression Profiling of Individual Cells Using Nanoliter Droplets. Cell 2015 21 May;161(5):1202–1214.
11. Lake BB, Ai R, Kaeser GE, Salathia NS, Yung YC, Liu R, et al. Neuronal subtypes and diversity revealed by single-nucleus RNA sequencing of the human brain. Science 2016 24 Jun;352(6293):1586–1590.
12. Habib N, Avraham-Davidi I, Basu A, Burks T, Shekhar K, Hofree M, et al. Massively parallel single-nucleus RNA-seq with DroNc-seq. Nat Methods 2017 Oct;14(10):955–958.
13. Svensson V, Natarajan KN, Ly LH, Miragaia RJ, Labalette C, Macaulay IC, et al. Power analysis of single-cell RNA-sequencing experiments. Nat Methods 2017 6 Mar;.
14. Hashimshony T, Senderovich N, Avital G, Klochendler A, de Leeuw Y, Anavy L, et al. CEL-Seq2: sensitive highly-multiplexed single-cell RNA-Seq. Genome Biol 2016 28 Apr;17(1):77.
15. Petukhov V, Guo J, Baryawno N, Severe N, Scadden D, Samsonova MG, et al. Accurate estimation of molecular counts in droplet-based single-cell RNA-seq experiments. bioRxiv 2017 Sep;p. 171496.
16. Soumillon M, Cacchiarelli D, Semrau S, van Oudenaarden A, Mikkelsen TS. Characterization of directed differentiation by high-throughput single-cell RNA-Seq. bioRxiv 2014 5 Mar;.
17. Jaitin DA, Kenigsberg E, Keren-Shaul H, Elefant N, Paul F, Zaretsky I, et al. Massively parallel single-cell RNA-seq for marker-free decomposition of tissues into cell types. Science 2014 14 Feb;343(6172):776–779.
18. Klein AM, Mazutis L, Akartuna I, Tallapragada N, Veres A, Li V, et al. Droplet barcoding for single-cell transcriptomics applied to embryonic stem cells. Cell 2015 21 May;161(5):1187–1201.
19. Zilionis R, Nainys J, Veres A, Savova V, Zemmour D, Klein AM, et al. Single-cell barcoding and sequencing using droplet microfluidics. Nat Protoc 2017 Jan;12(1):44–73.
20. Hochgerner H, Lönnerberg P, Hodge R, Mikes J, Heskol A, Hubschle H, et al. STRT-seq-2i: dual-index 5' single cell and nucleus RNA-seq on an addressable microwell array. Sci Rep 2017 Nov;7(1):16327.
21. Dobin A, Davis CA, Schlesinger F, Drenkow J, Zaleski C, Jha S, et al. STAR: ultrafast universal RNA-seq aligner. Bioinformatics 2013 1 Jan;29(1):15–21.
22. Liao Y, Smyth GK, Shi W. featureCounts: an efficient general purpose program for assigning sequence reads to genomic features. Bioinformatics 2014 1 Apr;30(7):923–930.
23. 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.
24. Smith TS, Heger A, Sudbery I. UMI-tools: Modelling sequencing errors in Unique Molecular Identifiers to improve quantification accuracy. Genome Res 2017 18 Jan;.
25. Fraley C, Raftery AE. Model-Based Clustering, Discriminant Analysis, and Density Estimation. J Am Stat Assoc 2002 Jun;97(458):611–631.
26. Fraley C, Raftery AE, Brendan Murphy T, Scrucca L. mclust Version 4 for R: Normal Mixture Modeling for Model-Based Clustering, Classification, and Density Estimation 2012;.
27. Vallejos CA, Risso D, Scialdone A, Dudoit S, Marioni JC. Normalizing single-cell RNA sequencing data: challenges and opportunities. Nat Methods 2017 Jun;14(6):565–571.
28. Evans C, Hardin J, Stoebel DM. Selecting between-sample RNA-Seq normalization methods from the perspective of their assumptions. Brief Bioinform 2017 27 Feb;.
29. Grün D, van Oudenaarden A. Design and Analysis of Single-Cell Sequencing Experiments. Cell 2015 5 Nov;163(4):799–810.
30. Satija R, Farrell JA, Gennert D, Schier AF, Regev A. Spatial reconstruction of single-cell gene expression data. Nat Biotechnol 2015 May;33(5):495–502.
31. Butler A, Satija R. Integrated analysis of single cell transcriptomic data across conditions, technologies, and species. bioRxiv 2017 Jul;p. 164889.
32. Tasic B, Menon V, Nguyen TN, Kim TK, Jarsky T, Yao Z, et al. Adult mouse cortical cell taxonomy revealed by single cell transcriptomics. Nat Neurosci 2016 Feb;19(2):335–346.
33. Bagnoli JW, Ziegenhain C, Janjic A, Wange LE, Vieth B, Parekh S, et al. mcSCRB-seq: sensitive and powerful single-cell RNA sequencing. bioRxiv 2017 Oct;p. 188367.
34. Law CW, Chen Y, Shi W, Smyth GK. voom: Precision weights unlock linear model analysis tools for RNA-seq read counts. Genome Biol 2014 Feb;15(2):R29.
35. 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 Aug;5:2122.
36. Grün D, Kester L, van Oudenaarden A. Validation of noise models for single-cell transcriptomics. Nat Methods 2014 Jun;11(6):637–640.
37. Tian L, Su S, Amann-Zalcenstein D, Biben C, Naik SH, Ritchie ME. scPipe: a flexible data preprocessing pipeline for single-cell RNA-sequencing data. bioRxiv 2017 Aug;p. 175927.
38. Islam S, Zeisel A, Joost S, La Manno G, Zajac P, Kasper M, et al. Quantitative single-cell RNA-seq with unique molec-

ular identifiers. Nat Methods 2014 Feb;11(2):163–166.

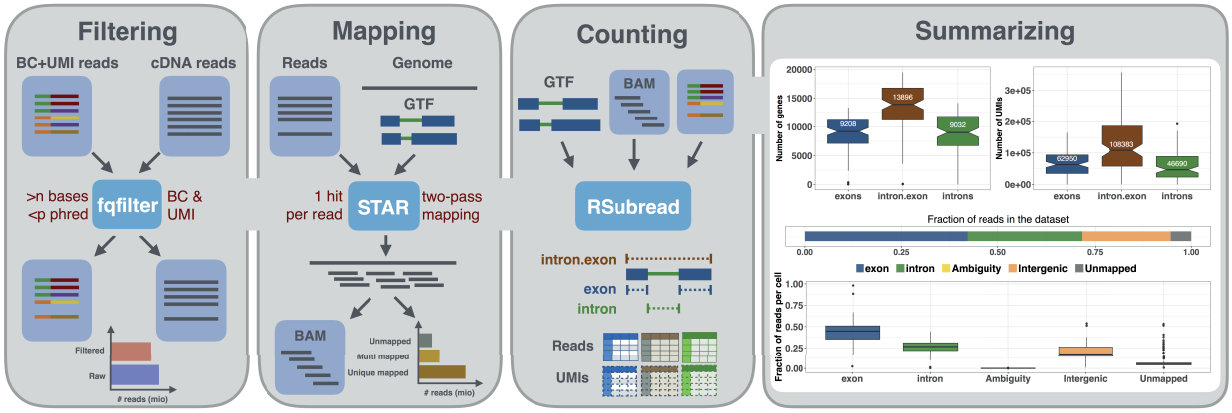


Figure 1. Schematic of the zUMIs pipeline. Each of the grey panels from left to right depicts a step of the *zUMIs* pipeline. First, fastq files are filtered according to user-defined barcode (BC) and unique molecular identifier (UMI) quality thresholds. Next, the remaining cDNA reads are mapped to the 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 barcode and distribution of reads into mapping feature categories.

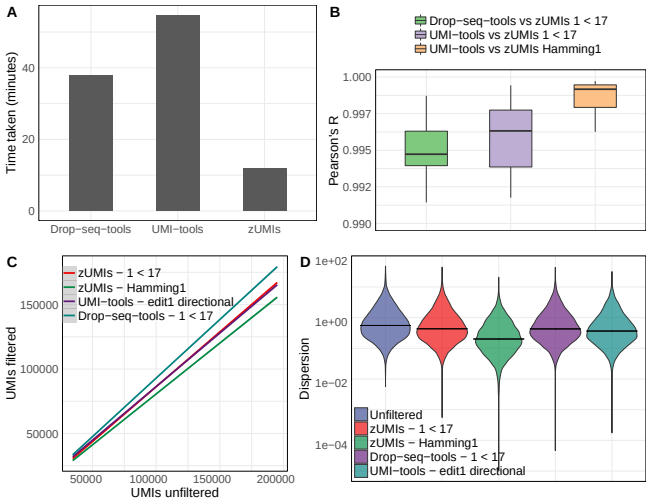


Figure 2. Comparison of different UMI collapsing methods. We compared Drop-seq-tools and UMI-tools with zUMIs using our HEK dataset (227 mio reads). **(A)** Runtime to count exonic UMIs using zUMIs (hamming distance = 0), UMI-tools ("unique" mode) and Drop-seq-tools (edit distance = 0). **(B)** Boxplots 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.

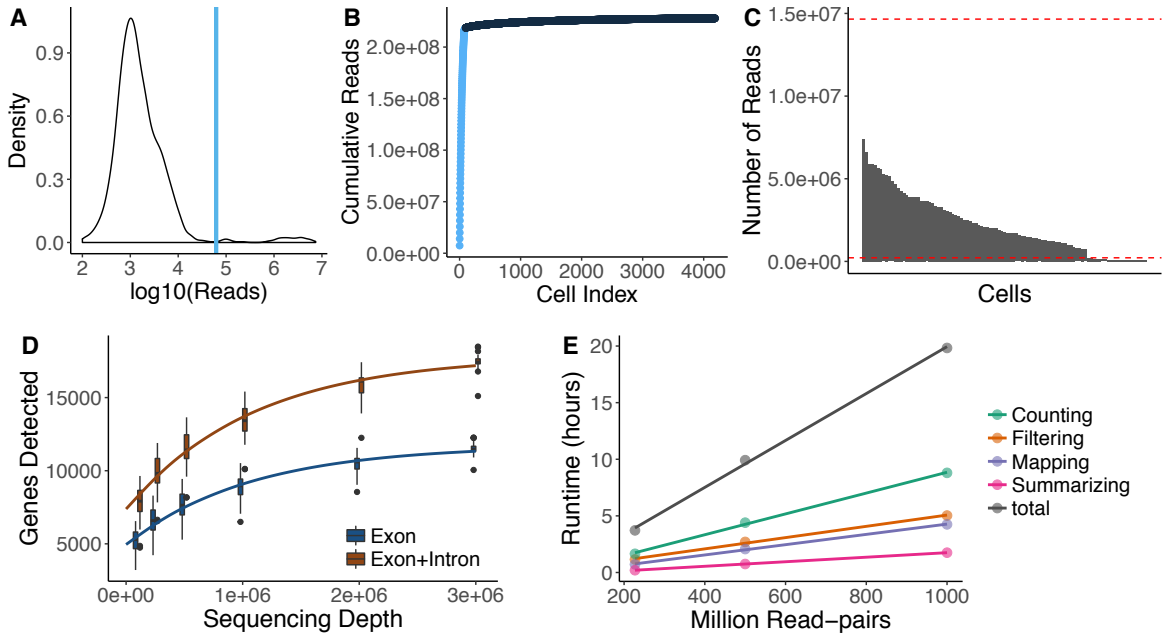


Figure 3. Utilities of zUMIs. Each of the panels shows the utilities of *zUMIs* pipeline. The plots from A-D are the results from the example HEK dataset used in the paper. A) The plot shows a density distribution of reads per barcode. Cell barcodes with reads above the blue line are selected. B) The plot shows the cumulative read distribution in the example HEK dataset where the barcodes in light blue are the selected cells. C) The barplot shows the number of reads per selected cell barcode with the red lines showing upper and lower MAD (Median Absolute Deviations) 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 is shown. E) Runtime for three datasets with 227, 500 and 1000 million read-pairs. The runtime is divided in the main steps of the *zUMIs* pipeline: Filtering, Mapping, Counting and Summarizing. Each dataset was processed using 16 threads ("-p 16").

Name	Reference	Open Source	Quality filter	UMI collapsing	Mapper	Intron	Down-sampling	Compatible UMI library protocols
Cell Ranger	[2]	yes	BC+UMI	Hamming distance	STAR	no	yes	[2]
CEL-seq	[14]	yes	BC+UMI	identity only	bowtie2	no	no	[36, 14]
dropEst	[15]	yes	BC	frequency-based	TopHat2	yes	no	[10, 18, 2]
Drop-seq-tools	[10]	no	BC+UMI	Hamming distance	STAR	no	no	[10, 16, 14]
scPipe	[37]	yes	BC+UMI	Hamming distance	subread	no	no	[36, 17, 16, 10]
umis	[13]	yes	BC	frequency-based	Kallisto	no	no	[16, 36, 38, 17, 10, 18, 2]
UMI-tools	[24]	yes	BC+UMI	network-based	BWA	no	no	[16, 18]
zUMIs	This work	yes	BC+UMI	Hamming distance	STAR	yes	yes	[16, 36, 38, 17, 10, 14, 20, 12, 3, 2]

Table 1. Features of available tools that can handle UMIs for the quantification of gene expression data. The evaluated features are whether the tool is open source, it considers the sequence quality for cell barcode (BC) and UMI, which mapper it uses, whether it can consider Intron mapping reads for counting, whether it offers a utility to make varying library sizes more comparable via downsampling and finally with which RNA-seq library preparation protocols it is compatible.

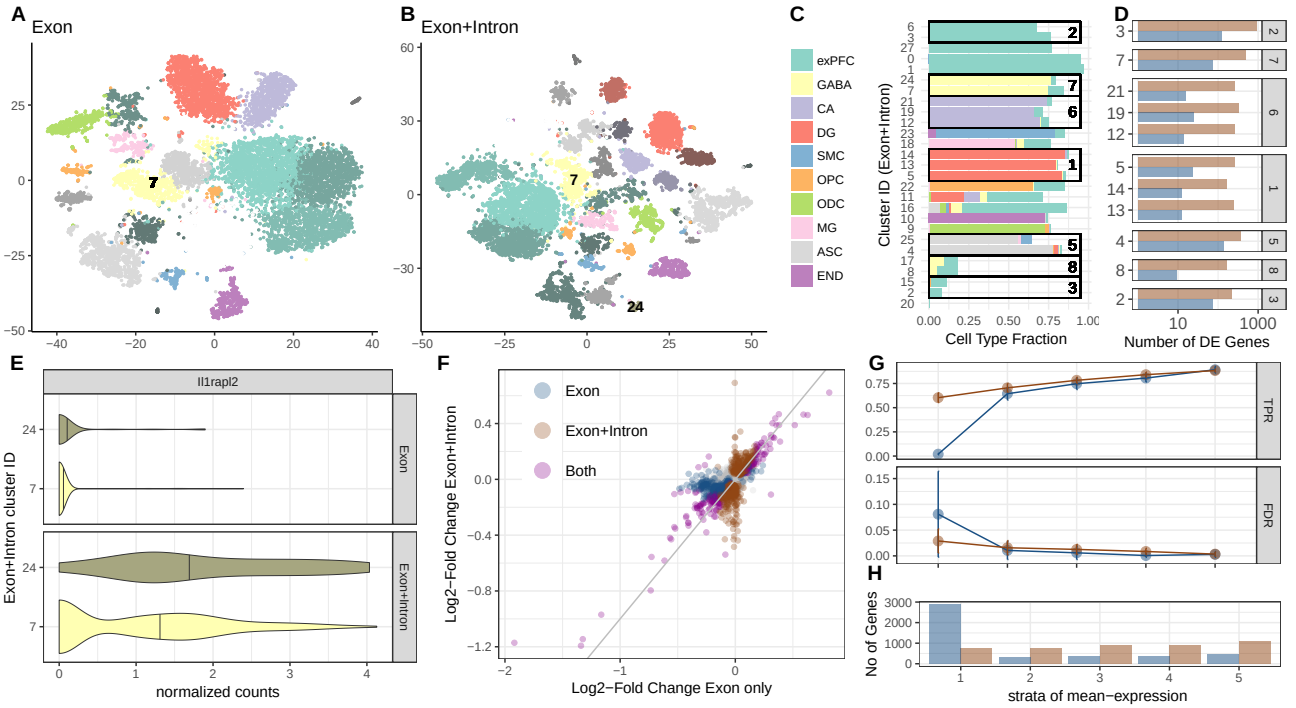


Figure 4. Contribution of Intron reads to biological insights. We analyse published single-nucleus RNA-seq data from mouse prefrontal cortex (PFC) and hippocampus [12] to assess the utility of counting Intron in addition to Exon reads. We processed the raw data with *zUMIs* to obtain expression tables with Exon reads as well as Exon+Intron reads and then use the R-package Seurat [30, 31] to cluster cells. With Exon counts, we thus identify 19 clusters **(A)** and with Exon+Intron counts 27 **(B)**. Clusters are represented as t-SNE plots and colored according to the most frequent cell-type assignment in the original paper [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). The composition of each cluster based on Exon+Intron is detailed in panel **(C)** and cells that were not assigned a cell type in Habib et al. [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 that 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 DE (limma p-adj<0.05) between the clusters only found 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 **(F)**. 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 **(E)**, 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 **(F)**. 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 **(F)**. Large LFCs are found significant with both counting strategies (purple points are close to the bisecting line). We conduct simulations based on mean and dispersion measured using Exon cluster 0 (1616 cells, ~ 90% exPFC). In **(G)** we show the expected true positive rate (TPR) and the false discovery rate (FDR) for a scenario comparing 300 vs 300 cells. Results for Exon and Exon+Intron counting were stratified into 5 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)**.

Figure 1

Click here to download Figure zUMI_pipeline.eps

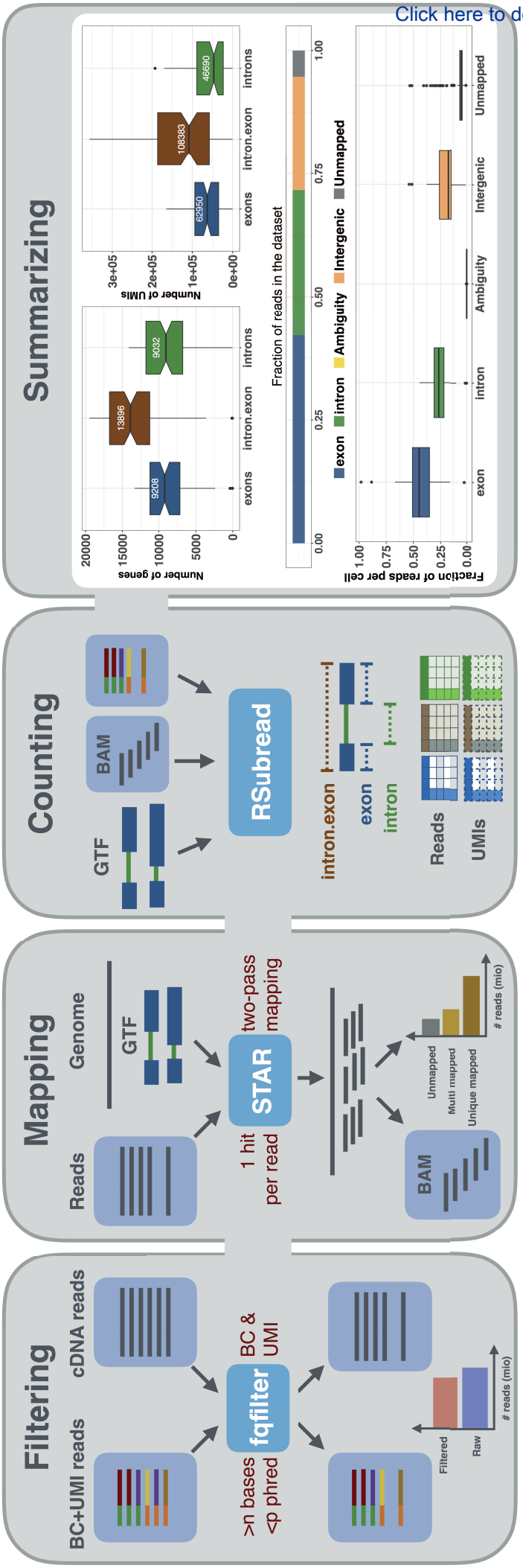
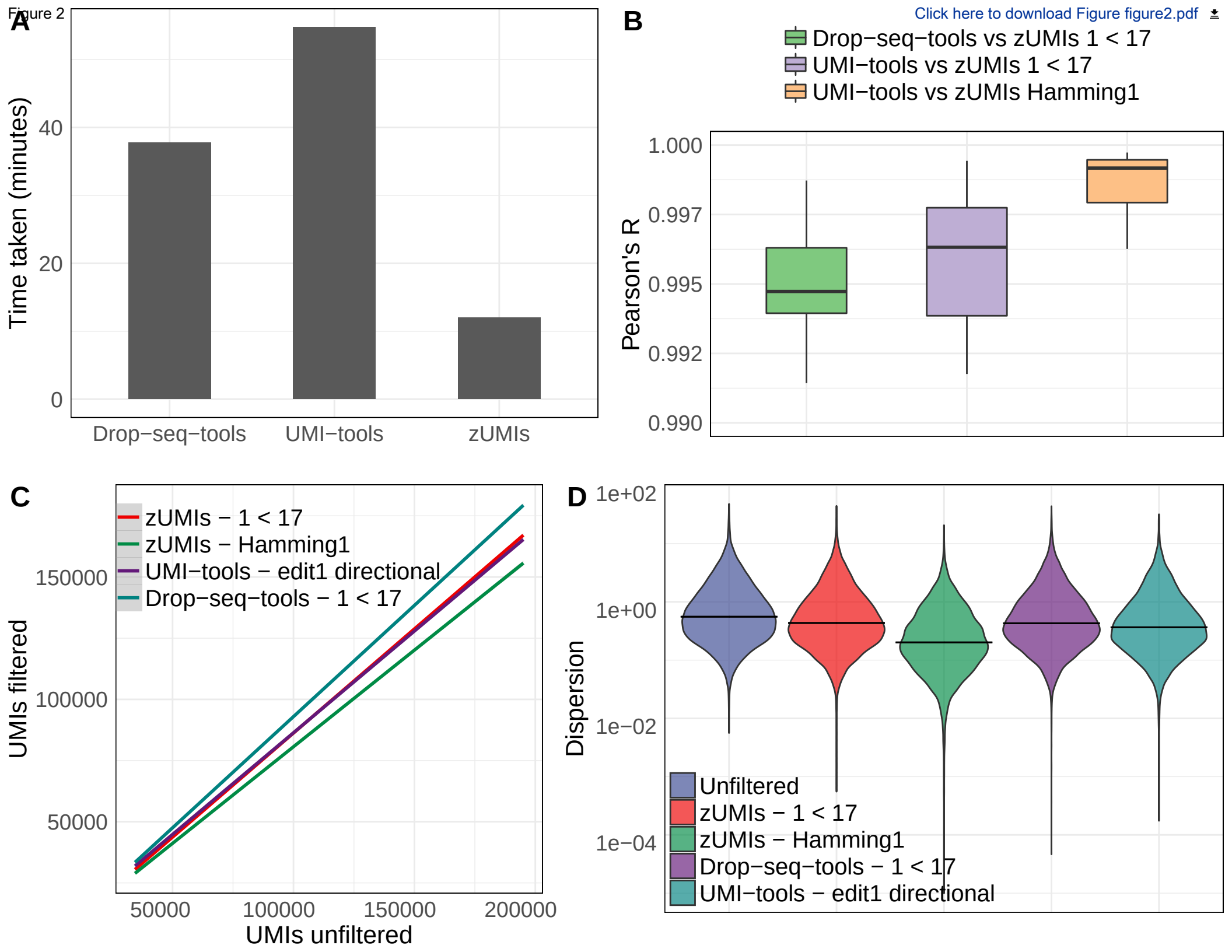


Figure 2


A
Time taken (minutes)
0
20
40
Drop−seq−tools
UMI−tools
zUMIs
B
Click here to download Figure figure2.pdf
Drop−seq−tools vs zUMIs 1 < 17
UMI−tools vs zUMIs 1 < 17
UMI−tools vs zUMIs Hamming1
Pearson's R
0.990
0.992
0.995
0.997
1.000
C
zUMIs − 1 < 17
zUMIs − Hamming1
UMI−tools − edit1 directional
Drop−seq−tools − 1 < 17
UMIs filtered
50000
100000
150000
50000
100000
150000
200000
UMIs unfiltered
D
Dispersion
1e−04
1e−02
1e+00
1e+02
Unfiltered
zUMIs − 1 < 17
zUMIs − Hamming1
Drop−seq−tools − 1 < 17
UMI−tools − edit1 directional

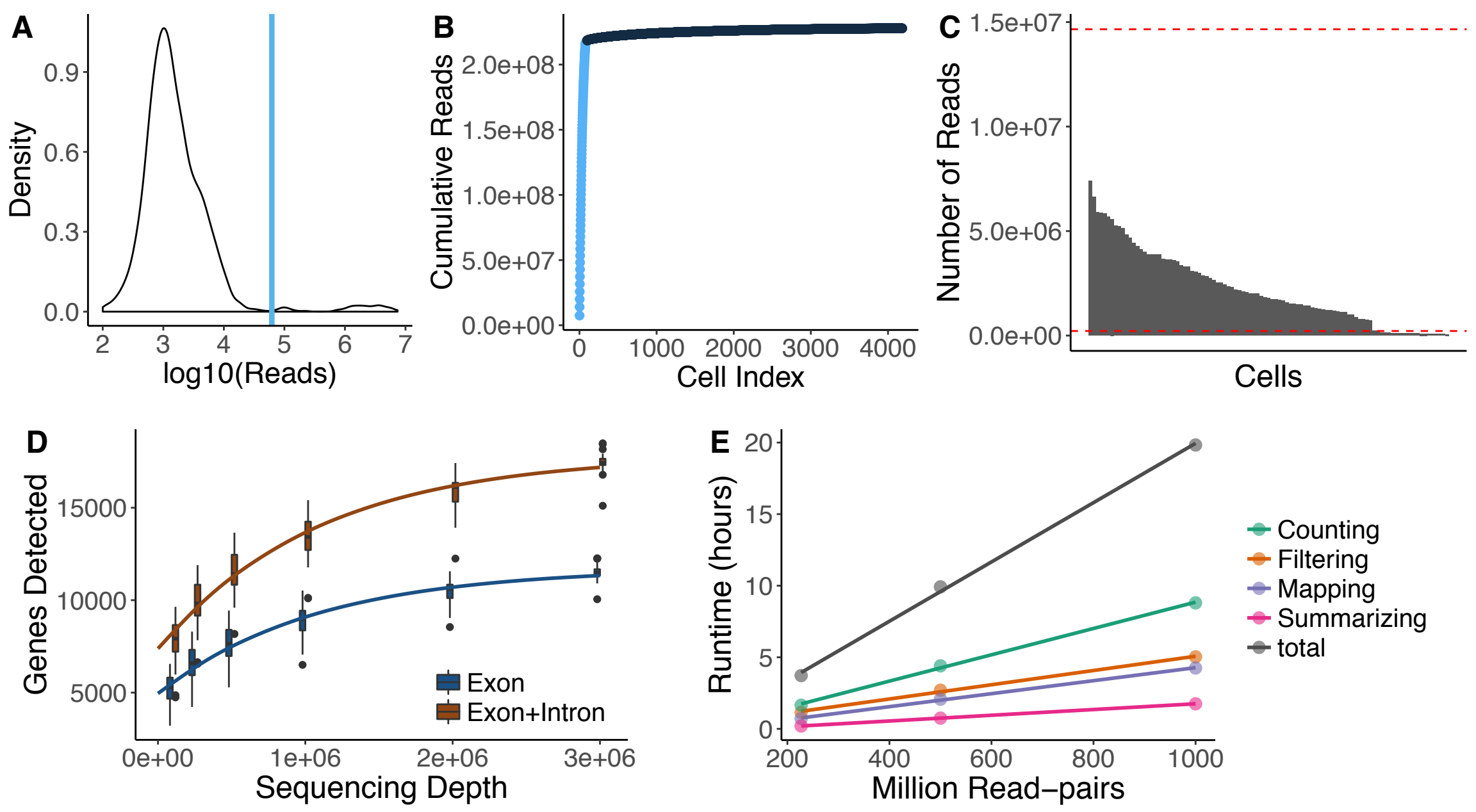
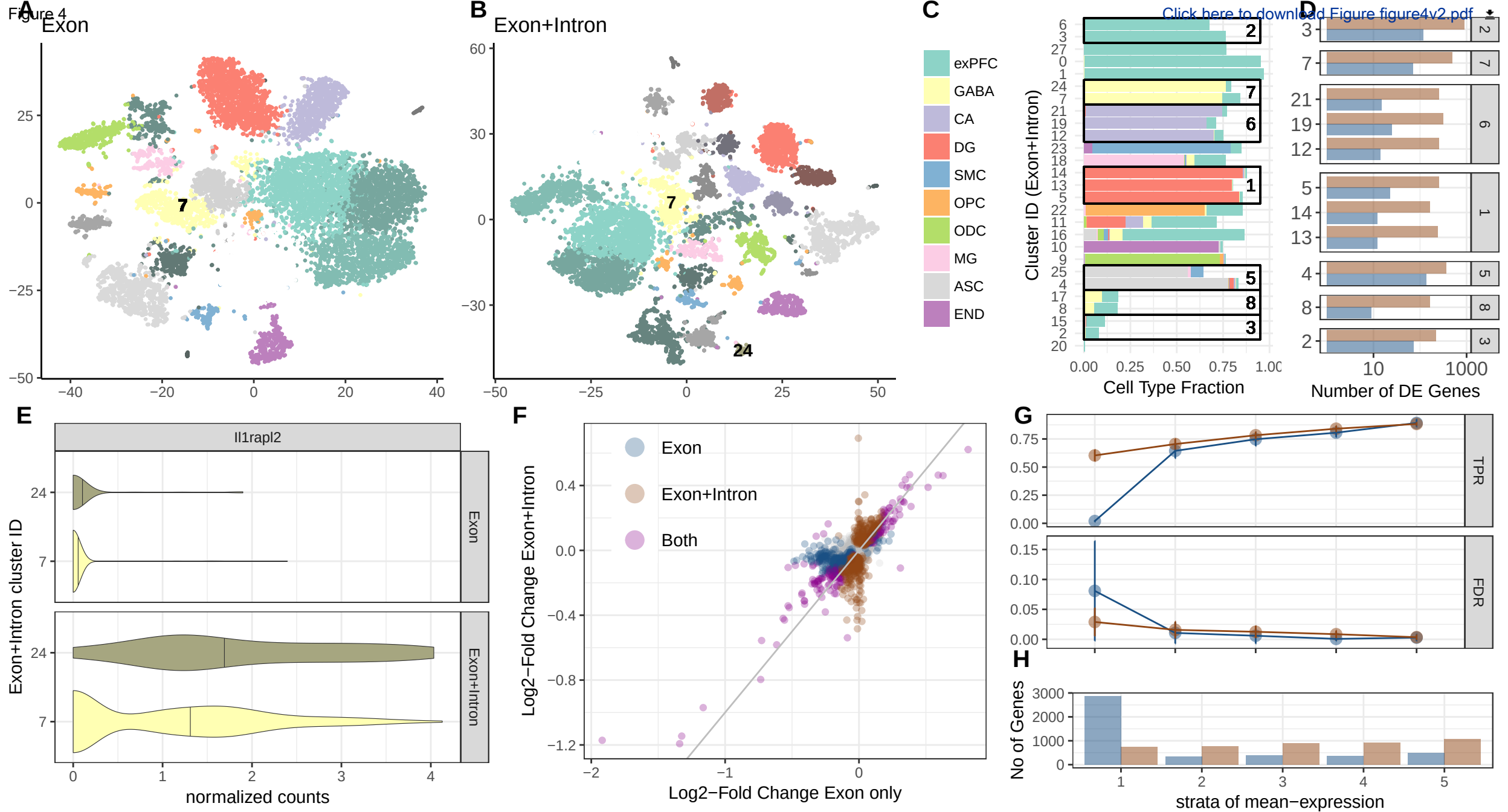

A
Density
0.0
0.3
0.6
0.9
2
3
4
5
6
7
log10(Reads)
B
Cumulative Reads
0.0e+00
5.0e+07
1.0e+08
1.5e+08
2.0e+08
0
1000
2000
3000
4000
Cell Index
C
Number of Reads
0.0e+00
5.0e+06
1.0e+07
1.5e+07
Cells
D
Genes Detected
5000
10000
15000
Exon
Exon+Intron
0e+00
1e+06
2e+06
3e+06
Sequencing Depth
E
Runtime (hours)
0
5
10
15
20
200
400
600
800
1000
Million Read−pairs
Counting
Filtering
Mapping
Summarizing
total

Figure 4



Click here to download Figure figure4v2.pdf

Additional File 1