Accurate estimation of molecular counts in droplet-based single-cell RNA-seq experiments

Viktor Petukhov[1,2], Jimin Guo[2], Ninib Baryawno[3,4,5], Nicolas Severe[3,4,5], David Scadden[3,4,5], Peter V. Kharchenko[2,3]

1. Department of Applied Mathematics, Peter the Great St. Petersburg Polytechnic University, St. Petersburg, Russia
2. Department of Biomedical Informatics, Harvard Medical School, Boston, MA, USA
3. Center for Regenerative Medicine, Massachusetts General Hospital, Boston, MA, USA
4. Harvard Stem Cell Institute, Cambridge, MA, USA
5. Department of Stem Cell and Regenerative Biology, Harvard University, Cambridge, MA, USA
6. Harvard Stem Cell Institute, Cambridge, MA, USA

Manuscript Type: Software

Abstract

Single-cell RNA-seq protocols provide powerful means for examining the gamut of cell types and transcriptional states that comprise complex biological tissues. Recently-developed approaches based on droplet microfluidics, such as inDrop or Drop-seq, use massively multiplexed barcoding to enable simultaneous measurements of transcriptomes for thousands of individual cells. The increasing complexity of such data also creates challenges for subsequent computational processing and troubleshooting of these experiments, with few software options currently available. Here we describe a flexible pipeline for processing droplet-based transcriptome data that implements barcode corrections, classification of cell quality, and diagnostic information about the droplet libraries. We introduce advanced methods for correcting composition bias and sequencing errors affecting cellular and molecular barcodes to provide more accurate estimates of molecular counts in individual cells.

Introduction

RNA-seq protocols have been optimized to enable large-scale transcriptional profiling of individual cells. Such single-cell measurements require both improved molecular techniques, as well as effective ways to isolate and process a large number of cells in parallel. While single-cell RNA-seq (scRNA-seq) remains a challenging technique, several solutions are being increasingly applied for biological studies, most notably techniques based on droplet microfluidics such as inDrop [1], Drop-seq [2] and 10X Chromium platform. In these approaches, cells are encapsulated in water-based droplets together with barcoded beads and necessary reagents within an oil-based flow. This allows the RNA material extracted from each cell to be contained within the droplet and tagged by a unique cellular barcode (CB) carried on the bead.

InDrop and similar approaches pool material from different cells to prepare the library, and rely on computational analysis to recognize the reads originating from the same cell based on the CB contained in the read sequence. The reads also carry a random barcode - a unique molecular identifier (UMI) [3, 4] - that can be used to discount the redundant contribution of reads originating from the same cDNA

molecule as a result of library amplification. As such, the primary aim of the data processing pipeline, including the one presented here, is to provide accurate estimates of the number of molecules that have been observed for each gene in each measured cell - a molecular count matrix. Accurate estimation of such matrix is crucial, as it commonly provides the starting point for all downstream analysis, such as cell clustering or tracing of cell trajectories.

Several factors complicate the estimation of this molecular count matrix, well beyond simple parsing of the read sequences. First, the procedure must separate reads originating from droplets containing real cells from contribution of empty droplets which can amplify extracellular background transcripts and significantly outnumber the real cells. Some of the droplets may contain damaged or fragmented cells, which complicates such separation. The procedure must also address problems stemming from sequencing errors, particularly errors within the CB or UMIs which result in misclassification of reads. Similarly, biased distribution of UMIs can lead to improper estimation of molecular counts. Finally, as droplet-based scRNA-seq protocols are still relatively new, detailed diagnostics and multiple quality control steps are typically needed to ensure high-quality measurements and identify likely sources of problems. Given current lack of such general processing pipelines for the droplet-based scRNA-seq, we have set out to provide an open-source implementation.

Results

We have developed a high-performance pipeline to perform initial pre-processing and analysis of the droplet-based scRNA-seq data. The pipeline characterizes the quality of a library using a wide range of diagnostic indicators, filters out artefactual cellular barcodes, evaluates and corrects for potentially confounding effects of uneven UMI coverage, and corrects for UMI and cellular barcode sequencing errors based on molecular similarity measures that do not require prior knowledge of the barcode sequences. It is designed to be used with alternative alignment methods and provides extensive configuration options to accommodate alternative scRNA-seq protocol designs.

Uneven UMI frequency distribution distorts molecular count estimation

In UMI-based protocols, the expression magnitude is typically estimated as a number of unique UMIs associated with a given gene in a given cell. If the space of possible UMI sequences is limited, it becomes possible for two separate molecules of the same transcript to be labeled by the same UMI. To account for such UMI collisions, Fu *et al.* have originally proposed a correction based on assumption of uniform distribution of UMIs in the overall dataset [3]. Such corrections are rarely used, given relatively large numbers of possible UMIs. Examining droplet data from different protocols, however, we find that UMI frequency distribution tends to be highly skewed, with a small fraction of UMIs contributing to a disproportionately large number of molecules (Figure 1A,B, Figure S1). The outlier UMIs with the highest frequencies show aberrant mononucleotide stretches towards the end of their sequence. Such biases may arise to due errors generated during library construction protocol or truncated barcode constructs. Even when such erroneous UMIs are filtered out, however, the overall UMI distribution remains significantly skewed (Figure S1B,F), suggesting that a more advanced approach is needed to correct for the impact of UMI collisions. In implementing corrections for the UMI collisions, we therefore moved away from the assumption of a uniform UMI distribution and used bootstrap procedures to directly model the true UMI frequency distribution (see Methods). This approach is effective at correcting UMI collisions on simulated data (Figure S2) and, as we will demonstrate in the next section, provides notable improvements on real data.

Errors in UMI sequence lead to overestimation of molecular counts

An error introduced into UMI sequence during the library preparation can be mistakenly interpreted as an additional molecule. Computational corrections have been proposed to avoid such overestimation.

The simplest such approach [5] omitted for a given gene all UMIs that had an adjacent UMI sequence (Hamming distance equal to 1) with a larger number of reads (as in [6] we refer this method as *cluster*). Indeed, the probability of having two molecules of the same transcript in the same cell being labeled by UMIs of Hamming distance 1 is low, given sufficient size of the UMI pool relative to the number of transcript molecules (see Methods). However, for moderately-expressed genes the observed number of such events exceeds the expected frequency by a factor of ~40 (Figure S3), suggesting that most adjacent UMI occurrences are erroneous. More complex, network-based solution [6] (referred here as *directional*) assumes that most of errors occurred during PCR, and considers a UMI to be erroneous if it has an adjacent UMI with more than twice the number of reads.

An alternative approach, implemented in the 10X Chromium Cell Ranger pipeline [7], uses UMI base call quality to distinguish erroneous UMIs. Examining different droplet-based datasets, we find that the fraction of UMI errors that distinguished by lower base call quality varies between datasets, within the range of 29.4-85.6% (Figure S4). This suggests that a substantial fraction of UMI errors may originate during PCR amplification or other library preparation steps preceding the sequencing itself. Base call quality would not be informative in such cases. Furthermore, the existing methods do not consider the total number of molecules for a given gene, even though the probability of observing adjacent UMIs by chance increases. Such increase is further exacerbated by an uneven distribution of UMI frequencies described in the previous section. For instance, for a gene with 100 molecules and UMI length of 6nt, 36 adjacent UMIs would be expected under uniform distribution, and 40 under the empirically observed distribution (mouse BMCs dataset was used).

To improve the accuracy of UMI filtering, we developed a Bayesian approach to estimate the posterior probability of a UMI being erroneous based on the gene expression magnitude, observed number of the adjacent UMI sequences, prior distribution of UMIs, as well as the position and identity of the nucleotide substitution (see Methods). To evaluate performance of different UMI correction approaches, we first examined a dataset with relatively long 10bp UMIs [8]. Given lower rate of accidentally observing UMIs at Hamming distance 1 in these longer UMIs, we used *cluster* UMI filtering procedure to obtain benchmark expression estimates for the dataset. We then simulated datasets with shorter UMIs by trimming the UMI sequences, and comparing the resulting molecular count estimates to the corresponding full-length benchmark values (see Methods).

While errors in the UMI sequences lead to over-estimation of the molecular abundance, UMI collisions lead to under-estimation. The probability of such collisions increases for shorter UMIs, which results in pronounced under-estimation of molecular counts at short UMIs (Figure 2A). Conversly, the over-estimation due to sequencing errors is more apparent at longer UMIs. Comparing different UMI collision correction methods, we find that the proposed approach based on bootstrap sampling of the empirical UMI frequency distribution shows much better performance than correction based on the uniform UMI distribution assumption (Figure 2B). We then compared different methods for correcting UMI sequence errors. In addition to the standard *cluster* algorithm [5], we also evaluated a variant that disallows merging of UMIs of equal sizes (*cluster-neq*). We found that the Bayesian approach proposed here significantly outperforms existing methods (Figures 2C, S5, Tables S1-4). The impact of both collision and sequencing error corrections is most notable for genes within the high expression range.

To examine the impact of different approaches on the downstream analysis we evaluated the similarity of expression profiles within major cell subpopulations in datasets generated using inDrop [9] and 10x Chromium [10, 11] platforms. We found that the proposed corrections result in significantly higher expression profile correlation of cells belonging to the same cluster, compared to uncorrected molecular counts (Figure 2D, S7). Spearman rank correlation was used to ensure robustness of the comparison. The proposed Bayesian approach showed best performance, though the difference in correlation gain between different correction methods was small in some datasets (Figure S7).

Correction of the cellular barcode sequence errors

The number of different cellular barcodes (CBs) in a droplet-based library normally exceeds the number of actual encapsulated cells by several fold (Figure 3A). Similar to issues encountered in UMIs, additional CBs can result from sequence errors introduced during library construction or sequencing. This would result in material from one droplet being mistakenly split up into several different CBs. Alternatively, additional CBs may also empty droplets that did not encapsulate a real cell, but instead captured background RNA or cell debris together with an indexing bead [12]. To evaluate whether this is a significant factor, we examined the read composition in an inDrop control experiment which encapsulated a mixture of human and mouse cells (see Methods). We find that the fraction of reads originating from the admixed genome increases with decreasing cell size (Figure 3B). Furthermore, we find that the admixture fraction correlates positively with the mitochondrial read fraction. These results suggest that capture of extracellular RNA background or cell debris by empty droplets provides noticeable contribution to small CBs.

We first examined methods for correcting CB sequence errors. In doing so we considered two scenarios: one where a list of possible valid CB sequences is known, and another where real CBs can be an arbitrary nucleotide sequence. Pre-designed CB sequences are typically evenly spaced in the sequence space, and replacing an erroneous CB with the closest matching valid CB sequence is an effective strategy. The space of potential valid CBs can be further narrowed down by taking into account that the valid CB should have more counts than the erroneous CB. However, if the list of possible valid CBs is unknown, or if there are many similar CBs (*e.g.* short barcodes), the number of possible merge targets increases significantly (Figure 3C). To accurately determine the probability that two CBs originated from one CB, we used UMI-gene composition similarity, which evaluates the likelihood that two independent cells will end up producing equivalent UMI-gene combinations (see Methods). We evaluated the model on the mouse ES dataset by estimating the intersection probabilities for the cells with distant CBs (Hamming distance ≥7), which were unlikely to be split up due to CB sequencing errors (Figure 3D). Indeed, it can be seen that almost all UMI-gene intersections between such cells could be obtained by chance. Only 0.1% around of all pairs of cells with distant barcodes show highly significantly degree of molecular overlap ($p<10^{-5}$), which is likely caused by another error modality such as RNA background contribution described above. To validate precision of the implemented merge procedure we compared results obtained with and without the usage of the real CB list. When analyzing mouse ES dataset without the knowledge of valid CBs, the algorithm performed 97.6% of the same merges into the real CB sequences resulting in nearly identical set of top cells (Figure 3A). Thus, the implemented approach provides effective correction for CB errors even when the set of possible CB sequences is not known.

Recognizing damaged or low quality cells

The number of molecules associated with a given CB generally provides reasonable criteria for selecting real cells [1, 7]. Similarly, CBs with very few associated reads likely represent empty droplets. However, classifying CBs in the intermediate range poses a challenge. The intermediate size CBs likely contain damaged or dying cells from which relatively little mRNA material could be recovered [12]. This complicates the optimization of a size separation threshold. Such low-quality cells could also cover a range of sizes, making the use of a single size cutoff ineffective.

Classification of low quality cells was examined by Illicic *et al* [12], where an SVN classifier was trained based on examination of cell morphology from microscopy data prior to lysis and library preparation. As such data is difficult to obtain for the droplet approaches, and existing SVN cannot be directly applied to different protocols or cell types, we aimed to develop a self-contained approach that would not require high-quality training data. While the true labels for low- and high-quality cells are not available, we argued that large cells initially include a large fraction of high-quality cells, and small cells include very low fraction of high-quality cells. We then aimed to train a classifier to distinguish high-quality cells

based on a limited set of technical features (see Methods), taking into account that the initial labels of the training set will contain some fraction of errors. The tolerance of different classifiers to training set errors can vary considerably. We evaluated performance of several appropriate approaches (KDE [13], Random Forest [14] and Robust Gaussian Processes [15], see Methods). In addition to cross-validation score we measured robustness of the classifiers with respect to: removal of a random 20% of the training data (5-fold cross-validation, see Table 1); introduction of artificial noise into the data (Figure S9A,B); narrowing/widening of the thresholds used to separate large and small cells for the initial label assignment (Figure S9C). Based on the resulting performance and runtime complexity (*e.g.* Robust GP has a high complexity of $O(n^3)$ relative to the number of samples) we chose Kernel Density Estimation (KDE) classifier. An example of KDE-based quality scores is shown in Figure 4A for the mouse pancreatic dataset. While the quality scores show expected association with cell size, some of the smaller cells are able to attain high scores, and some of the large cells are assigned low scores (Figure S10, S11). The additional "high-quality" (score>0.9) cells identified by the classifier beyond the initial size threshold show consistent clustering with the major subpopulations (Figure 4B) supporting their validity. In contrast, selecting a set of cells on the basis of the size threshold alone quickly leads to appearance of an artefactual cluster of poor-quality cells (Figure S12). Filtering of cells by the KDE quality score preserves all of the subpopulations identified by the original publication [9]. An example of cell classification on another high-complexity dataset (mouse bone marrow), shown in Figure 4C also illustrates poor clustering of cells with low quality scores.

Discussion

Droplet-based microfluidics protocols and other high-throughput methods are enabling production of large single-cell RNA-seq datasets (10^3-10^6 cells). Complex barcoding schemes employed by such methods require in-depth computational analysis to achieve accurate recovery of molecules associated with different cells and genes. In order to avoid collisions of cellular barcodes, large numbers of cells necessitate longer CBs, increasing the probability that a sequence error will be introduced into a CB during the bead construction steps [1], library preparation procedures, or library sequencing. We show that in many such errors there are multiple equidistant CBs from which the molecule may have originated. The implemented solution, which merges CBs based on the probabilistic assessment of the molecular overlap between the CBs, provides accurate correction even in the cases when the set of possible valid CBs is not known in advance.

Errors affecting molecular barcodes (UMIs) pose a similar challenge, which in this case is driven by increasing sequencing depth of individual cells. This has been recognized by earlier studies [5], and several correction strategies have been proposed. We show that the overall distribution of UMI sequence occurrences is not uniform, and the resulting bias reduces the effective UMI space leading to increased number of UMI collisions in well-expressed genes and deflated molecular counts. Some of the UMI errors appear to result from occurrence of aberrant library molecules incorporating mononucleotide primers, such as polyT into the UMI position. On the other hand, point mutations in UMIs and aberrant base calls can lead to inflated molecular counts. While most UMI errors can be mitigated experimentally by increasing the UMI length, we show that taking into account empirical distribution of UMI frequencies allows to adjust for both UMI collision and sequence error effects.

Even with corrections of CB sequence errors, most of the CBs encountered in the current droplet-based datasets do not represent real cells. As these methods do not encapsulate a cell into all of the bead-containing droplets, the additional molecules may originate from empty droplets capturing extracellular background. Indeed, examination of mouse-human dataset mixtures, suggests that smaller CBs have higher cross-organism contamination fraction that one would expect from extracellular background. In addition to empty droplets, some of the low-magnitude CBs may represent damaged, dying or dead cells, or cells that were not successfully measured for other reasons. The challenge of identifying

damaged cells has been previously examined by Ilicic *et al.* [12] in the context of Fluidigm C1 protocol where the proportion of low-quality cells is typically in the range of 10-40%. This fraction can be much higher in the inDrop data (*e.g.* 90% of CBs), and obtaining microscopy-based labeling for the classification is challenging given the rapid flow within the devices. We instead, explored application of fault-tolerant classifiers to identify technical features consistent with partially-correct separation of high-quality cells based on the size criteria alone. Such approach is able to pick up relatively large cells that resemble poorly-measured cells based on their technical features, and rescue some of the smaller cells that look consistent with high-quality tail of the cell distribution. Overall, we hope that the developed pipeline will facilitate analysis of the droplet-based single-cell RNA-seq data, providing helpful diagnostic, and improving the accuracy of the resulting expression estimates.

Methods

The dropEst pipeline operates in three phases: i) identifier parsing phase; ii) read mapping phase; and iii) filtering and quality control phase. The first phase takes as an input FASTQ files containing paired-end read and index data. The output of this phase is a modified FASTQ file with reads which can be aligned to a transcriptome reference during the second phase using a standard splice-aware aligner (*e.g.* STAR [16], or Tophat 2.1.0 [17] that was used in our work). The third phase takes BAM files with the aligned reads and a gene annotation file in GTF format. BAM files produced by the 10x Cell Ranger pipeline can be also provided when running this stage. The result of pipeline is an R-readable file that contains molecular count matrix and other processed information, as well as a report with diagnostic information on the library.

Correction of UMI collisions

In case when the number of UMI per gene is comparable to the total UMIs pool size, the gene expression level will be underestimated [3], and needs to be adjusted by taking UMI collisions into account. Fu *et al.* [3] assumed uniform distribution of UMI probabilities and the number of unique UMIs expected for number of molecules n from a UMI pool of size m is $k = m[1 - e^{-(n/m)}]$, thus $n = -m\, ln(1 - \frac{k}{m})$. To account for non-uniform UMI distribution observed in the droplet datasets, $n(k, m)$ was estimated computationally using UMI bootstrap sampling. Specifically, possible values of n were quantized with a step value 50. For each quantized value of n, a 1000 bootstrap sampling iterations were performed to estimate corresponding values of n. Subsequently, in estimating $n(k, m)$ for an arbitrary value of k, linear interpolation was used. To validate the developed method, we simulated UMI collisions by randomly merging small genes (Figure S2).

Correction of UMI sequence errors

To determine whether two UMIs represent technical variations of the same UMI, we use Bayesian approach to derive an optimal decision boundary based on the overall UMI frequency distribution.

Given two UMIs within a gene, we considered the following features:

- U: sequence of the first UMI.
- u: sequence of the second UMI.
- R: number of reads for the first UMI.
- r: number of reads for the second UMI, $r \leq R$.

- N_S: number of adjacent (Hamming distance of 1) UMIs for the UMI U with the number of reads $r': r' \leq R$.
- N_L: number of adjacent UMIs for the UMI U with the number of reads $R': R' > R$.
- S_g: number of UMIs in the gene.

in the following description, when referring to the UMI with the larger number of reads we use the upper case latters, and for the UMI with the fewer reads we use lower case letters. Let $\vec{V}$ denote (U, R, N_S, N_L). The equation of the optimal decision surface:

$$p(\vec{V}, u, r, S_g | Err_{(U,u)}) * p(Err_{(U,u)}) = p(\vec{V}, u, r, S_g | \neg Err_{(U,u)}) * p(\neg Err_{(U,u)}),$$

where $Err_{(U,u)}$ means that u occurred because of an error in U.

As we have no information about prior distribution of the events $Err_{(U,u)}$ and $\neg Err_{(U,u)}$, we assume them to be equally likely:

$$p(\vec{V}, u, r, S_g | Err_{(U,u)}) = p(\vec{V}, u, r, S_g | \neg Err_{(U,u)}).$$

The right part of the equation can be simplified as follows:

$$p(u, r, U, R, N_L, N_S, S_g | \neg Err_{(U,u)}) =$$

$$= p(u, r, N_S | U, R, N_L, S_g, \neg Err_{(U,u)}) * p(U, R, N_L, S_g | \neg Err_{(U,u)}) =$$

Events U, N_L, R don't depend on the $Err_{(U,u)}$ (let's suppose the decrease in R insignificant). Also we assume that S_g doesn't depend on the $Err_{(U,u)}$. Then the right part of the equation takes the form:

$$= p(u, r, N_S | U, R, N_L, S_g, \neg Err_{(U,u)}) * p(U, R, N_L, S_g) =$$

Given $\neg Err_{(U,u)}$, events u and r do not depend on $\vec{V}$:

$$= p(u, r | S_g, \neg Err_{(U,u)}) * p(N_S | U, R, N_L, S_g, \neg Err_{(U,u)}) * p(U, R, N_L, S_g) =$$

Given $\neg Err_{(U,u)}$, event r does not depend on u, and r does not depend on S_g. Furthermore, $\neg Err_{(U,u)}$ does not mean that u is a real UMI, as u could be a result of an error in a UMI other than U. Then $p(r | \neg Err_{(U,u)}) \approx p(r)$. Similarly, $p(u | \neg Err_{(U,u)}) \approx p(u)$:

$$= p(u | S_g) * p(r) * p(U, R, N_L, S_g) * p(N_S | U, R, N_L, S_g, \neg Err_{(U,u)}).$$

The left part of the equation could be simplified as follows:

$$p(u, r, \vec{V}, S_g | Err_{(U,u)}) = p(u, r, U, R, N_L, N_S, S_g | Err_{(U,u)}) =$$

$$= p(u, r, N_L, N_S | U, R, S_g, Err_{(U,u)}) * p(U, R, S_g | Err_{(U,u)}) =$$

Given $Err_{(U,u)}$, events N_L and N_S depend on u, and on r only through U and R. Moreover, R does not depend on U and S_g, and events U and S_g do not depend on $Err_{(U,u)}$:

$$= p(u, r | U, R, S_g, Err_{(U,u)}) * p(N_L, N_S | U, R, S_g, Err_{(U,u)}) * p(R | Err_{(U,u)}) * p(U, S_g) =$$

Given $Err_{(U,u)}$, event u depends on U only through the position of the difference between the sequences $C_{(U,u)}$ and the nucleotide replacement $D_{(U,u)}$. Also, neither u nor r depends on S_g:

$$= p\left(C_{(U,u)}, D_{(U,u)}, r|R, Err_{(U,u)}\right) * p\left(N_L, N_S|U, R, S_g, Err_{(U,u)}\right) * p\left(R|Err_{(U,u)}\right) * p\left(U, S_g\right) =$$

Events $C_{(U,u)}$ and $D_{(U,u)}$ do not depend on r and R. Also, we can assume them to be independent from each other:

$$= p\left(C_{(U,u)}|Err_{(U,u)}\right) * p\left(D_{(U,u)}|Err_{(U,u)}\right) * p\left(R, r|Err_{(U,u)}\right) *$$

$$* p\left(N_L, N_S|U, R, S_g, Err_{(U,u)}\right) * p\left(U, S_g\right) =$$

We denote $p_C = p\left(C_{(U,u)}|Err_{(U,u)}\right) * p\left(D_{(U,u)}|Err_{(U,u)}\right) * p\left(R, r|Err_{(U,u)}\right)$:

$$= p_C * p\left(N_S|U, R, N_L, S_g, Err_{(U,u)}\right) * p\left(N_L|U, R, S_g, Err_{(U,u)}\right) * p\left(U, S_g\right) =$$

N_L doesn't depend on $Err_{(U,u)}$:

$$= p_C * p\left(N_S|U, R, N_L, S_g, Err_{(U,u)}\right) * p\left(N_L|U, R, S_g\right) * \frac{p\left(U, R, S_g\right)}{p(R)} =$$

$$= p_C * p\left(N_S|U, R, N_L, S_g, Err_{(U,u)}\right) * \frac{p\left(U, R, N_L, S_g\right)}{p(R)}.$$

The simplified expresions can be combined as follows. The equation of the Bayes optimal decision boundary can be written as:

$$p_C * p\left(N_S|U, R, N_L, S_g, Err_{(U,u)}\right) * \frac{p\left(U, R, N_L, S_g\right)}{p(R)} =$$

$$= p\left(u|S_g\right) * p(r) * p\left(U, R, N_L, S_g\right) * p\left(N_S|U, R, N_L, S_g, \neg Err_{(U,u)}\right)$$

or

$$p_C * p\left(N_S|U, R, N_L, S_g, Err_{(U,u)}\right) =$$

$$= p\left(u|S_g\right) * p(r) * p(R) * p\left(N_S|U, R, N_L, S_g, \neg Err_{(U,u)}\right).$$

After reordering:

$$\frac{p_C}{p\left(u|S_g\right) * p(r) * p(R)} = \frac{p\left(N_S|U, R, N_L, S_g, \neg Err_{(U,u)}\right)}{p\left(N_S|U, R, N_L, S_g, Err_{(U,u)}\right)}.$$

As $p\left(u|S_g\right) = (1 - \left(1 - p(u)\right)^{S_g})$:

$$\frac{p_C}{(1 - \left(1 - p(u)\right)^{S_g}) * p(r) * p(R)} = \frac{p\left(N_S|U, R, N_L, S_g, \neg Err_{(U,u)}\right)}{p\left(N_S|U, R, N_L, S_g, Err_{(U,u)}\right)}.$$

Let's suggest that N_S depends on R only through N_L:

$$\frac{p_C}{(1-(1-p(u))^{S_g}) * p(r) * p(R)} = \frac{p(N_S|U, N_L, S_g, \neg Err_{(U,u)})}{p(N_S|U, N_L, S_g, Err_{(U,u)})}.$$

Estimating probabilities.

$p(u)$, $p(r)$ and $p(R)$ can be simply estimated by the observed frequencies. To estimate probabilities $p(C_{(U,u)}|Err_{(U,u)})$, $p(D_{(U,u)}|Err_{(U,u)})$ and $p(R, r|Err_{(U,u)})$ we create a training sample, which contains only pairs of UMIs where u occurred because of an error in U. Such set was assembled by choosing the genes containing only two adjacent UMIs only. The theoretical probability $p(u, U|S_g = 2, \neg Err_{(U,u)})$ is negligible. Thus we expect almost all such events have occurred because of an error in U. Such training sample is representative because all events $C_{(U,u)}$, $D_{(U,u)}$, R and r are independent of S_g. Thus, estimation of $p(C_{(U,u)}|Err_{(U,u)})$ and $p(D_{(U,u)}|Err_{(U,u)})$ becomes straightforward. Probability $p(R, r|Err_{(U,u)})$ was approximated by the multinominal distribution.

Estimation of $p(N_S|U, N_L, S_g, \neg Err_{(U,u)})$ and $p(N_S|U, N_L, S_g, Err_{(U,u)})$. These probabilities depends on the large numbers of parameters, making the training approach impractical. We use theoretical estimation of $p(N'_S|U, N_L, S_g)$, where $N'_S = |\{u': (r' \leq R) \ \& \ (\forall U' \neg Err_{(U',u')})\}|$ (see the algorithm below). To simplify theoretical estimation, we can use the following procedure. We introduce a scoring function for the pair of UMIs $f(\vec{V}, u, r, S_g)$, which is approximately proportional to the probability $p(Err_{(U,u)}|\vec{V}, u, r, S_g)$. We then assume that the event $\neg Err_{(U,u)}$ is equal to the event $\neg Err_{(U,u')}$ $\forall u' \in \Omega_u = \{u': f(\vec{V}, u', r, S_g) \leq f(\vec{V}, u, r, S_g)\}$. Similarly, $Err_{(U,u)}$ is equal to the event $Err_{(U,u')}$ $\forall u': f(\vec{V}, u', r, S_g) \geq f(\vec{V}, u, r, S_g)$. Then we estimate:

$$p(N_S|U, N_L, S_g, Err_{(U,u)}) = p\left((N'_{S,p} \leq |\Omega_u|)|U, N_L, S_g\right),$$

$$p(N_S|U, N_L, S_g, \neg Err_{(U,u)}) = p\left((N'_{S,p} \geq |\Omega_u|)|U, N_L, S_g\right).$$

We can use the left part of the decision boundary equation as a scoring function $f(\vec{V}, u, r, S_g)$. In this case, for the fixed UMI U within the fixed gene g we can order all of the adjacent UMIs u_i by the increasing of $f(\vec{V}, u_i, r, S_g)$. The decision boundary equation:

$$f(\vec{V}, u_i, r, S_g) = \frac{p(N'_{S,p} \leq i|U, N_L, S_g)}{p(N'_{S,p} \geq i|U, N_L, S_g)}.$$

Estimating distribution $\boldsymbol{p(N'_S|S_g, U, N_L)}$.

We use the following notations:

- L: length of the UMI.
- N_{UMI}: total number of possible UMIs (in most cases is equal to 4^L).
- K: maximal number of the adjacent UMIs (in most cases is equal to $3L$).
- $p_{Neighb} = p_{Neighb}(U)$: probability to observe a UMI, adjacent to U. Equal to $\sum_{u \in Adjacent(U)} p(u)$.
- N': total number of adjacent UMIs for the UMI U.

To estimate the distribution of N'_S we use the following assumption:

$$P(N'_S \geq n|S_g, U, N_L) = P(N' \geq n + N_L|S_g, U, N' \geq N_L) = \frac{P(N' \geq n + N_L|S_g, U)}{\sum_{k=N_L}^{K} P(N' \geq k|S_g, U)}.$$

The distribution $p(N'|S_g, U)$ was estimated by modeling of the process of picking UMIs from the pool. Suppose that we already picked s UMIs and we have k different adjacent UMIs. Let us denote it as the state (k, s). This state can occur in of one of the following situations:

1. We were previously in a state $(k, s-1)$ and picked a UMI which was not a new adjacent UMI (*i.e.* either a previously observed adjacent UMI, or not adjacent UMI). The probability of such pick is $\left(1 - \frac{K-k}{K} p_{Neighb}(U)\right)$.

2. We were previously in a state $(k-1, s-1)$ and picked a UMI, which is a new adjacent UMI. The probability of such pick is $\frac{K-k-1}{K} p_{Neighb}(U)$.

The model above can be evaluated using dynamic programming. To do it we build a matrix $T = \{t_{k,s}\}$, each cell of which contains the weighted sum of the neighboring bottom-left and left cells in the matrix T (see example in Table 2).

Such matrices would need to be computed for the each UMI present in the dataset. However, the asymptotic of this approach is equal to $O(S_g * \#UMI * K)$ in terms of both time and memory, which would be too large for large datasets. To optimize it we employ the following solution. The matrix T depends on U only through $p_{Neighb}(U)$, and the rate of change of the function within a cell is proportional to $p_{Neighb}(U) + o(p_{Neighb}(U))$. Thus, we assume $p(N'|S_g, U)$ to be a piecewise constant function from $p_{Neighb}(U)$ and perform a quantization by this probability. A quantization step $\Delta p = 0.01$ was used.

Iterative procedure of UMI sequence errors correction. **After the estimation of the decision boundary, all UMIs that were** determined **as erroneous are removed.** This **will change the input parameters S_g, N_L and N_S of the algorithm.** Therefore, to **perform a precise filtration, the procedure** is **ran iteratively**.This does not add significant amount of runtime complexity **because:** i) only the right part of the decision rule is affected; ii) dynamic programming matrices are calculatd only once, as the gene size cannot increase during filtration; iii) for genes with a small number of UMIs, the procedure converges after one or two iterations.

Correction of cellular barcode sequence errors

CB sequence errors split a fraction of the molecules originating from one cell into smaller CBs. Given that the number of reads per UMI is generally high then one, the smaller CBs will contain some of the same gene-UMI combinations as the true CB. In other words, the smaller CBs will have similar molecular composition - the set of cell unique genes-UMI combinations. We use composition similarity as a criterion for determining whether the two barcodes should be merged. Size of the compositional intersection between two independent cells is modeled using Poisson distribution with the mean dependent on the UMIs distribution and the number of molecules associated with the CBs. For the two fixed CBs with molecular count vectors (gene expression profiles) $\overrightarrow{E_i}$ and $\overrightarrow{E_j}$, parameter λ of the Poisson distribution was estimated using the following bootstrap-based procedure:

1. The distribution of UMI probabilities P_0 was estimated over all genes in all cells of the dataset.
2. Two sets of UMIs with sizes $S_i = \sum_{k=0}^{|\overrightarrow{E_i}|} E_{i,k}$ and $S_j = \sum_{k=0}^{|\overrightarrow{E_j}|} E_{j,k}$ were taken from P_0 with replacement.
3. The selected sets were randomly allocated to genes, according to the profiles $\overrightarrow{E_i}$ and $\overrightarrow{E_j}$.
4. The number of common gene-UMI combinations was counted.
5. The procedure was repeated a fixed number of iterations, starting with the step 2.

Using the estimated distribution, we perform a statistical test for the hypothesis H_0: observed size of the intersection $S^*{}_{intersection}$ was obtained by chance. P-value of the test is the value of the Poisson distribution function $P_{\hat{\lambda}}(S_{intersection} \geq S^*{}_{intersection} | \overrightarrow{E_i}, \overrightarrow{E_j}, P_{UMI})$, where $\hat{\lambda}$ is the estimated parameter.
The implemented pipeline uses this test to compare each cell C_i with all cells C_j that have higher total number of molecules ($S_j \geq S_i$) and whose CBs have Hamming distance lower than a fixed constant. In the presented results the constant was taken to be 2. To account for multiplicity comparisons, we used Bonferroni adjustment.

Classifying damaged or low-quality cells

The implemented approach for classification of damaged and low quality cells split into three tasks: (i) creation of the training sample*, i.e.* establishing the initial class labeling, (ii) feature selection and (iii) application of a classifier algorithm.

The initial class labels were assigned based on the cell size. To do so, the dataset was split into three parts: 'big' cells, 'intermediate' cells and 'small' cells. To determine the borders of 'big' and 'small' cells we used the plot $\log(\#UMI\ in\ cell)$ vs $\log(cell\ rank)$ (Figure S8C). The algorithm is based on heuristic observation that for all the observed datasets, the left part of the plot has a negative second derivative, followed by a linear part, followed by a third part of the plot has a positive second derivative. The automated procedure locates the left (t_l) and the right (t_r) position of the linear part. The cells with size smaller than t_r were assigned the initial label of 'low-quality' cells. The top 75% of the cells with the size larger than t_l were assigned the initial label of 'high-quality' cells. The remaining cells were labeled as 'unknown'. Alternatively, the initial label borders can be specified by hand, for instance based on the shape of the $\log(\#UMI\ in\ cell * \#Cells)$ vs $\log(\#UMI\ in\ cell)$ plots (Figure S8A,B).

Given the initial labeling, the was addressed as that of establishing robust classification in the presence of training label noise [18]. We compared three classifiers: Kernel Density Estimation classifier [13], Random Forest [14] and Robust Gaussian Processes Classifier [15]. Following evaluation of robustness we chose KDE classifier with Normal Scale bandwidth selector [19] (the implementation provided by the R package 'ks' was utilized [20]).

Two types of features could be considered: biological features (*e.g.* expression levels of genes belonging to different GO categories [21, 22]), and technical features (*e.g.* different statistics on the sequenced data). We expect most biological features to be dataset and cell type-specific [12], with exception of the mitochondrial fraction, which has appeared as a robust indicator of cell death [12]. Therefore, we limited biological features to the fraction of UMIs on mitochondrial reads. The following technical features were utilized:

1. Mean number of reads per UMI.
2. Mean number of UMI per gene.
3. Fraction of low-expressed genes (genes with 1 molecule).
4. Fraction of UMIs on low-expressed genes.
5. Fraction of intergenic reads (optional feature, as it typically has to be calculated during the identifier parse phase).

The computational complexity of the KDE classifier estimation has exponential dependency on the dimensionality of the feature space. We therefore reduced the feature space by using first three principal components of the feature space for the classification. Because of its biological meaning, mitochondrion fraction is additionally used by the algorithm: all cells with high mitochondrion fraction are explicitly set as 'low-quality' prior to the algorithm training. High fraction threshold was set to be

m+4a, where *m* is the median cell mitochondrial fraction in the dataset, and *a* is the median absolute deviation of the mitochondrial fraction.

Mouse Bone Marrow inDrop measurements.
Whole bone marrow cells were isolated from 11 weeks old C57Bl/6 male mice (Jackson Laboratory). Epiphysis/metaphysis fraction from long bones was collected, crushed, cut into small pieces, and digested using Collagenase I (STEMCELL Technologies) for 30 minutes at 37-degree agitation. Bone marrow cells were filtered through a 70-micron filter. Red blood cells were lysed using Ack-lysis (ThermoFisher Scientific) on ice for five min, quenched with Media 199 (ThermoFisher Scientific) supplemented with 2% fetal bovine serum (ThermoFisher Scientific), and spun down at 500 g for 5 min. Cells were stained for 30 minutes with the red blood cell marker TER119 (Biolegend) and cells were sorted using DAPI (ThermoFisher Scientific) as a live/dead viability marker. Four hundred thousand live whole bone marrow cells (negative for TER119) were sorted into medium 199 (ThermoFisher Scientific). Before inDrop encapsulation cells were counted using a Cellometer (Nexcelom Bioscience). Cell viability is over 90%.

InDrop processing. Concentration of cells was adjusted to 300,000 cells/ml by adding PBS to the sorted cells. The cell suspension was then mixed 1:1 (v/v) with PBS containing 30% OptiPrep Density Gradient Medium (Sigma D1556) to obtain 150,000 cells/ml in 15% Optiprep. Using four microfluidics pumps and a polydimethylsiloxane (PDMS) microfluidic device, about 10,000 cells were co-encapsulated with barcoded polyacrylamide beads and a reverse transcription mixture containing Superscript III into water-in-oil droplets, according to a published protocol [23]. The library preparation and quality control procedures were carried out as described [23]. Indexed libraries were pooled and sequenced on aNext-seq 500 system (Illumina) at 2 pM concentrations.

Mouse-human cell line mixture inDrop measurement.
CK1750 mouse lung cancer cells (Carla Kim laboratory, Boston Children's Hospital) and K562 human immortalized myelogenous leukemia cells (ATCC) were mixed at 1:1 ratio to obtain 70,000 cells/ml in PBS containing 15% Optiprep. About 3,000 cells were co-encapsulated with barcoded polyacrylamide beads and a reverse transcription mixture containing Superscript III into water-in-oil droplets; and a library was prepared according to a published protocol [23]. The library was sequenced on a MiSeq system (Illumina).

Datasets

The following datasets were used in evaluating the developed pipeline:

1. Mouse embryonic stem cells (935 cells, SRA accession SRR1784310 (GEO GSE65525), [1])
2. Human pancreatic cells (sample 4, run 2, SRA accession SRR3879612 GEO (GSM2230760), [9])
3. Mouse pancreatic cells (sample 2, SRA accession SRR3879617, SRR3879618, SRR3879619 (GSM2230762), [9])
4. Mixture of human and mouse cells (2876 cells, SCG29, GEO GSEXXXX)
5. Human frozen peripheral blood mononuclear cells (2900 cells, 10x Genomics Frozen PBMCs (Donor A), [7, 10])
6. Human pre-transplant bone marrow mononuclear cells (900 cells, 10x Genomics AML035 Pre-transplant BMMCs, [7, 24])
7. Human post-transplant bone marrow mononuclear cells (900 cells, 10x Genomics AML035 Post-transplant BMMCs, [7, 8])
8. Human frozen bone marrow mononuclear cells (2400 cells, 10x Genomics Frozen BMMCs (Healthy Control 2), [7, 11])
9. Human CD34+ cells (9000 cells, 10x Genomics CD34+, [7, 25])
10. Mouse bone marrow cells (SCG71, GEO GSEXXXX)

Implementation
DropEst pipeline implementation is available on github: https://github.com/hms-dbmi/dropEst.

Acknowledgements

We would like to thank Allon Klein and Sinisa Hrvatin for their helpful comments on analysis of inDrop datasets, Carla Kim for providing mouse cell line for inDrop testing. PVK was supported by NIH R01HL131768 from NHLBI.

Figure Legends

Figure 1. Skewed distribution of UMIs leads to increased number of UMI collisions.
(A) Distribution of UMI occurrence frequencies across all genes is shown for the mouse ES dataset. The top-right inset shows position-specific nucleotide frequencies of the outlier UMIs (highlighted by gray shading on the main plot). Significant skewness of the UMI distribution decreases the effective pool of UMIs.
(B) Proportions of different nucleotides in the UMI are shown as a function of the overall UMI frequencies (x-axis orders UMIs so that most frequently occurring UMI sequences have low rank) for the mouse ES cells dataset.
(C) Estimated number of UMI collisions as a function of the true gene expression level (x axis) is shown for different UMI lengths (simulated by trimming from 10nt UMIs, see text). Estimates based on uniform and empirical UMI distributions are shown. 10x human post-transplant BMMCs dataset was used. For short UMIs, the number of collisions observed at highly-expressed genes can be comparable to the true number of molecules. Longer UMIs decrease the number of collisions.

Figure 2. Comparison of UMI collision and sequencing error correction methods.
(A) The scatter plot shows percent error (y axis) in estimation of the molecular counts for different genes using computationally trimmed UMIs (down to 6-9nt lengths) from their original 10nt length, as a function of the full-length UMI estimate (x axis, see Methods). The line shows spline-smoothed dependency with the 95% confidence band. The errors result from two opposing trends. Shortened UMIs result in larger number of collisions which deflate the count estimates (most pronounced at low UMI lengths). On the other hand, longer UMIs accumulate more sequencing errors which inflate the count estimates (most notable at high UMI lengths).
(B) The effect of different UMI collision corrections is shown on the 6nt trimmed UMIs.
(C) Comparison of different UMI sequence error correction methods is shown for 8nt trimmed UMIs. UMI collisions were corrected using *bootstrap* approach in all cases except for “no correction”.
(D) The impact of UMI corrections on the expression cohesion within resulting cell subpopulations. Distribution of within-cluster expression Spearman rank correlations is shown for the human BMMC dataset. UMI errors and collisions decrease within-cluster similarity (assessed here using transcriptome-wide Spearman rank correlation). Improvements resulting from different UMI error correction methods are shown, with or without bootstrap collision adjustments.

Figure 3. Correcting for Cellular Barcode errors.
(A) The number of molecules per cell before and after the merge correction procedure, is shown for the mouse ES dataset as a function of the overall cell size as reflected by rank of the cell (with cells possessing most molecules having the lowest ran; cells with ≤ 500 molecules were omitted). The results of the merge procedure with and without the prior knowledge of the possible barcode list match almost exactly. The resulting distribution shows a pronounced inflection point coinciding with the real number of encapsulated cells [1].
(B) Analysis of human-mouse cell mixtures suggest contribution of extracellular background. The fraction of human-mouse molecule admixture is shown (color gradient) for cells binned by the number

of molecules originating from the two genomes. The cells were binned by the number of human molecules (x axis), and the fraction of mouse molecules originating from mouse cells exceeding a certain size threshold (y axis) was calculated for each bin. The majority of admixed molecules originate from small mouse cells, with highest admixture fraction observed between small mouse and small human cells (>75%).
(C) The number of equidistant adjacent larger CBs is shown for each of the observed CBs in the mouse ES dataset. The main shows adjacent CBs selected from *a priori* known set of valid CB sequences. The inset shows counts of adjacent CBs selected from all CB sequences observed in the dataset.
(D) Comparing all pairs of cells with similar CBs (edit distance 1-5) and distant CBs (distance 7-10), the plot shows the dependency between the theoretical probability of a certain number of overlapping UMI-gene combinations between the two cells (x axis) with the empirical probability (observed fraction, y axis) of such cells. Cell pairs with similar CBs show much higher overlap, which is driven by CB sequencing errors. A small fraction of distant CBs (0.1%) also shows higher molecular overlap then expected, which is likely explained by cross-droplet contamination.

Figure 4. Classification of low- and high-quality cells.
(A) The cell quality score (y axis) as determined by the developed approach is shown as a function of cell size (x axis, shown as rank with largest cells being assigned the lowest rank) for the mouse pancreatic cell dataset paper [9]. The initial size-group assignment of each cell is shown by color. As expected, most large cells are reported to have high scores, and cells with small number of molecules tend to have low quality scores. Nevertheless, some of the large cells are reported as being low in quality and some of the small cells are reported to have high quality.
(B) Visualization of high-quality cells (score>0.9) using t-SNE projection in mouse pancreatic dataset. The cells are colored according to the original cluster labels, with cells that were not passing the original size threshold but were nevertheless deemed to be of high quality shown in gray. Most of such cells can be confidently assigned to the annotated clusters.
(C) Visualization of low-quality cells using t-SNE projection on mouse bone marrow dataset. All cells passing the initial size selection threshold are shown (see Methods). Cells with quality scores below 0.8 are shown in black. These poorly-scoring cells are generally found scattered outside of the dendritic and resident macrophage clusters.

Supplementary Figure Legends

Figure S1. Skewness of UMI distributions.
As in Figure 1 of the main manuscript, nucleotide frequencies and UMI distributions are shown for mouse ES cells dataset after the filtration of extreme right tail (A, B). The analogous plots are also shown for 10x the human post-transplant BMMCs dataset before (C, D) and after the filtration (E-F). The filtering only has a visible effect on the left peak of the distribution, and does not affect the overall skewed shape of the distribution.

Figure S2. Simulation of UMI collision frequencies.
To evaluate the developed approach for correcting UMI collisions, we modeled collisions by randomly merging genes within a given cell (using mouse MBC dataset). The number of the resulting unique UMIs after the merge is shown on the x axis. The y axis shows the number of collisions (i.e. the difference between the total number of unique molecules before and after the merge). The observed number of collisions is shown in red dots scatter. The amount of collisions predicted by different models is shown by lines. The bootstrap sampling from the observed distribution shows good fit to the data.

Figure S3. Probability of observing adjacent UMIs in small genes.
Probability of observing two adjacent UMIs in a given cell in a given gene is shown for different edit distances (x axis). Probability estimated based on the bootstrap procedure is shown in blue. The

empirically observed probabilities are shown in green. Mouse ES dataset was used. The observed number of adjacent UMIs exceeds the expected estimates by a factor of 40, indicating that most of the adjacent UMI occurrences result from UMI sequence errors.

Figure S4. Recognition of UMI errors by base calling quality.
Errors in UMI sequence can occur during either amplification or sequencing. Depends on the source, these errors can or cannot be distinguished by base calling quality. To examine the contribution of these mechanisms to UMI sequence errors we compared distributions of base calling quality by different subsets of data. We focused on the adjacent UMIs (same gene, same cell), most of which are expected to be erroneous (Figure S3). The boxplots show for each dataset, the distribution of nucleotide base call quality for the adjacent (red) and distant UMIs (blue). Specifically, red box plot shows base call quality for the nucleotide discrepant between the adjacent genes. The blue box plot shows quality distribution across all nucleotides in pairs of UMIs that are sufficiently distant (Hamming distance > 3). The numbers above the box plots show the fraction of reads of each type that were correctly classified based on the base call quality alone (using Bayesian classifier). The results suggest that the fraction of UMI errors that can be easily explained by the base call quality varies between the datasets, ranging from 29.4% to 85.6%.

Figure S5. UMI sequence correction comparisons on trimmed UMI data.
(A) Boxplots show distributions of errors (expressed as #UMIs+1 on a log scale) for different datasets, for different lengths of the trimmed UMIs. Each point corresponds to a particular gene/cell combination. The percent numbers above the boxplots give the total error in the proportion of the molecules counted in the dataset. The genes for which all of the methods gave correct results were excluded from the comparison.
(B) Dependency of the error on the real number of UMIs. As in the Figure 2C of the main manuscript, the plots show the error in molecular count estimates in trimmed UMI data (expressed as a percentage of the molecular count), relative to the untrimmed *cluster*-corrected estimates (see Methods). Each point shows the median error (y axis) for all of the genes with the same untrimmed number of molecules (x-axis). Compared to Bayesian correction method, most methods underestimate the counts for highly expressed genes, even when trimming by 1nt (9nt UMIs).

Figure S6. UMI collisions on trimmed data.
The plots show effect of different UMI collision adjustment approaches on the trimmed UMI data. The true number of molecules (estimated on untrimmed UMIs with *cluster* correction) is shown on the x axis, with the y axis giving the percent error for different lengths of trimmed UMIs (the error percentage is calculated relative to the untrimmed estimates). The lines were obtained using spline smoothing, as in Figure 2 of the main manuscript.

Figure S7. Correlation of gene expression profiles within clusters.
Similar to Figure 2D of the main manuscripts, the plots show distribution of within-cluster Spearman correlation of the estimated cell expression profiles for (A) 10x frozen PBMCs dataset, and (B) mouse pancreatic cells dataset. In the both cases, most of the correction methods result in similar improvements to within-cluster correlation.

Figure S8. Initial labeling of high-quality cells based on cell size distributions.
Two heuristic ways of assigning initial cell labels to high-quality and low-quality cells are shown.
(A) Shows scaled histogram, where the cells were binned based on the total number of molecules. For each bin, the y axis gives the number of molecules multiplied by the number of cells, so that the overall plot shows the distribution of molecular counts as a function of cell size. Such distributions tend to be bi-modal and the local minimum can be used to estimate the separation between the well-measured cell population (left) and low-quality/empty droplets (right). However such pattern is not apparent in some datasets.

(B) Similar pattern can be obtained by changing the x axis to map molecular counts on the cell rank (with the largest cell being assigned the lowest rank).
(C) An alternative approach, used by 10x Chromium Cell Ranger, is to identify the size threshold based on the second derivative of the cumulative size distribution plot (see Methods). Here the y axis gives the total size of the cell, and x axis orders the cells by rank. The cells within the 25% of the identified thresholds (by rank) are initially labeled as Unknown (see Methods).

Figure S9. Robustness of different classifiers to training errors.
(A) As classification of low-quality cells depends on the ability to tolerate high level of errors in the initial training label assignment, we examined the robustness of different types of algorithms to artificially-added errors. Specifically, each classifier was first trained on the initial data, and the resulting labels were used for subsequent perturbations (to avoid simulation bias). Then 80% of the data was used to re-train the classifier, but a fixed percentage of class labels (x-axis) were swapped. The plot shows false positive and false negative rates (y axis) of different classifiers on the remaining 20% of the data.
(B) Analogous plot shows performance of the classifier on the labels that were swapped in the 80% of the data that was used for training.
(C) Robustness of classifiers to the selection of the initial labeling. The labeling borders were widened by a fixed percentage (x axis), and the performance of the classifiers was compared to the labels classifiers were able to predict based on the initial borders.

Figure S10. Difference between cell quality score and cell size rank.
The plot illustrates the extent of disagreement between the labeling of large cells based on the cell size threshold (x axis) and the labeling based on the cell quality score. The measures were calculated by taken the rank-based labeling was taken to be true. The optimal agreement is achieved at the minimum of the green curve. Mouse pancreas dataset was used.

Figure S11. Comparison of the initial label assignments with the cell quality score predicted by the algorithm.
A tSNE visualization shows all mouse pancreatic cells colored according to: (A) their initial (size-based) quality labels, and by (B) quality score returned by the KDE classification algorithm. Overall, the algorithm recognizes lower-size high-quality cells that are found within the major population clusters. Conversely, almost all cells outside of the major clusters are recognized with low (<0.5) quality scores.

Figure S12. tSNE visualization of cells selected based on the total number of molecules.
Mouse pancreas cells are shown on t-SNE plots, colored by the published classification. Gray colors are used to show additional cells. Distribution of cell sizes identified cell size threshold separating 1524 top cells.
(A) Shows embedding of the top 75% of these cells (top 1197) which are the cells that are initial label of ‘good’ cells.
(B) Shows embedding of all 1524 cells.
(C) Shows embedding of all cells detected in the dataset.
All plots, including the most stringent size-selected (A) contain a grey cluster of unclassified cells. This artefactual cluster is removed when considering only cells with high quality score (>0.9, Figure 4B of the main manuscript).

References

1. Klein, A.M., et al., *Droplet barcoding for single-cell transcriptomics applied to embryonic stem cells.* Cell, 2015. **161**(5): p. 1187-201.
2. Macosko, E.Z., et al., *Highly Parallel Genome-wide Expression Profiling of Individual Cells Using Nanoliter Droplets.* Cell, 2015. **161**(5): p. 1202-14.
3. Fu, G.K., et al., *Counting individual DNA molecules by the stochastic attachment of diverse labels.* Proc Natl Acad Sci U S A, 2011. **108**(22): p. 9026-31.
4. Islam, S., et al., *Quantitative single-cell RNA-seq with unique molecular identifiers.* Nat Methods, 2014. **11**(2): p. 163-6.
5. Bose, S., et al., *Scalable microfluidics for single-cell RNA printing and sequencing.* Genome Biol, 2015. **16**: p. 120.
6. Smith, T., A. Heger, and I. Sudbery, *UMI-tools: modeling sequencing errors in Unique Molecular Identifiers to improve quantification accuracy.* Genome Res, 2017. **27**(3): p. 491-499.
7. Zheng, G.X., et al., *Massively parallel digital transcriptional profiling of single cells.* Nat Commun, 2017. **8**: p. 14049.
8. Genomics, x. *AML035 Post-transplant BMMCs*. 2016 [cited 2017 30 July]; Available from: https://support.10xgenomics.com/single-cell-gene-expression/datasets/1.1.0/aml035_post_transplant.
9. Baron, M., et al., *A Single-Cell Transcriptomic Map of the Human and Mouse Pancreas Reveals Inter- and Intra-cell Population Structure.* Cell Syst, 2016. **3**(4): p. 346-360 e4.
10. Genomics, x. *Frozen PBMCs (Donor A)*. 2016 [cited 2017 30 July]; Available from: https://support.10xgenomics.com/single-cell-gene-expression/datasets/1.1.0/frozen_pbmc_donor_a.
11. Genomics, x. *Frozen BMMCs (Healthy Control 2)*. 2016 [cited 2017 30 July]; Available from: https://support.10xgenomics.com/single-cell-gene-expression/datasets/1.1.0/frozen_bmmc_healthy_donor2.
12. Ilicic, T., et al., *Classification of low quality cells from single-cell RNA-seq data.* Genome Biol, 2016. **17**: p. 29.
13. Hastie, T., R. Tibshirani, and J.H. Friedman, *The elements of statistical learning : data mining, inference, and prediction*, in *Springer series in statistics*. 2009, Springer: New York. p. 208-210.
14. Breiman, L., *Random Forests.* Machine Learning, 2001. **45**(1): p. 5-32.
15. Kim, H.-C. and Z. Ghahramani, *Outlier Robust Gaussian Process Classification*, in *Structural, Syntactic, and Statistical Pattern Recognition: Joint IAPR International Workshop, SSPR & SPR 2008, Orlando, USA, December 4-6, 2008. Proceedings*, N. da Vitoria Lobo, et al., Editors. 2008, Springer Berlin Heidelberg: Berlin, Heidelberg. p. 896-905.
16. Dobin, A., et al., *STAR: ultrafast universal RNA-seq aligner.* Bioinformatics, 2013. **29**(1): p. 15-21.
17. Kim, D., et al., *TopHat2: accurate alignment of transcriptomes in the presence of insertions, deletions and gene fusions.* Genome Biol, 2013. **14**(4): p. R36.
18. Frenay, B. and M. Verleysen, *Classification in the Presence of Label Noise: A Survey.* IEEE Transactions on Neural Networks and Learning Systems, 2014. **25**(5): p. 845-869.
19. Chacon, J.E., T. Duong, and M.P. Wand, *ASYMPTOTICS FOR GENERAL MULTIVARIATE KERNEL DENSITY DERIVATIVE ESTIMATORS.* Statistica Sinica, 2011. **21**(2): p. 807-840.
20. Duong, T. *Package 'ks'*. 2014 [cited 2017 30 July]; Available from: http://cran.r-project.org/web/packages/ks/ks.pdf.
21. Ashburner, M., et al., *Gene ontology: tool for the unification of biology. The Gene Ontology Consortium.* Nat Genet, 2000. **25**(1): p. 25-9.
22. Gene Ontology, C., *Gene Ontology Consortium: going forward.* Nucleic Acids Res, 2015. **43**(Database issue): p. D1049-56.
23. Zilionis, R., et al., *Single-cell barcoding and sequencing using droplet microfluidics.* Nat Protoc, 2017. **12**(1): p. 44-73.

24. Genomics, x. *AML035 Pre-transplant BMMCs*. 2016 [cited 2017 30 July]; Available from: https://support.10xgenomics.com/single-cell-gene-expression/datasets/1.1.0/aml035_post_transplant.
25. Genomics, x. *CD34+*. 2016 [cited 2017 30 July]; Available from: https://support.10xgenomics.com/single-cell-gene-expression/datasets/1.1.0/cd34.

Figure 1

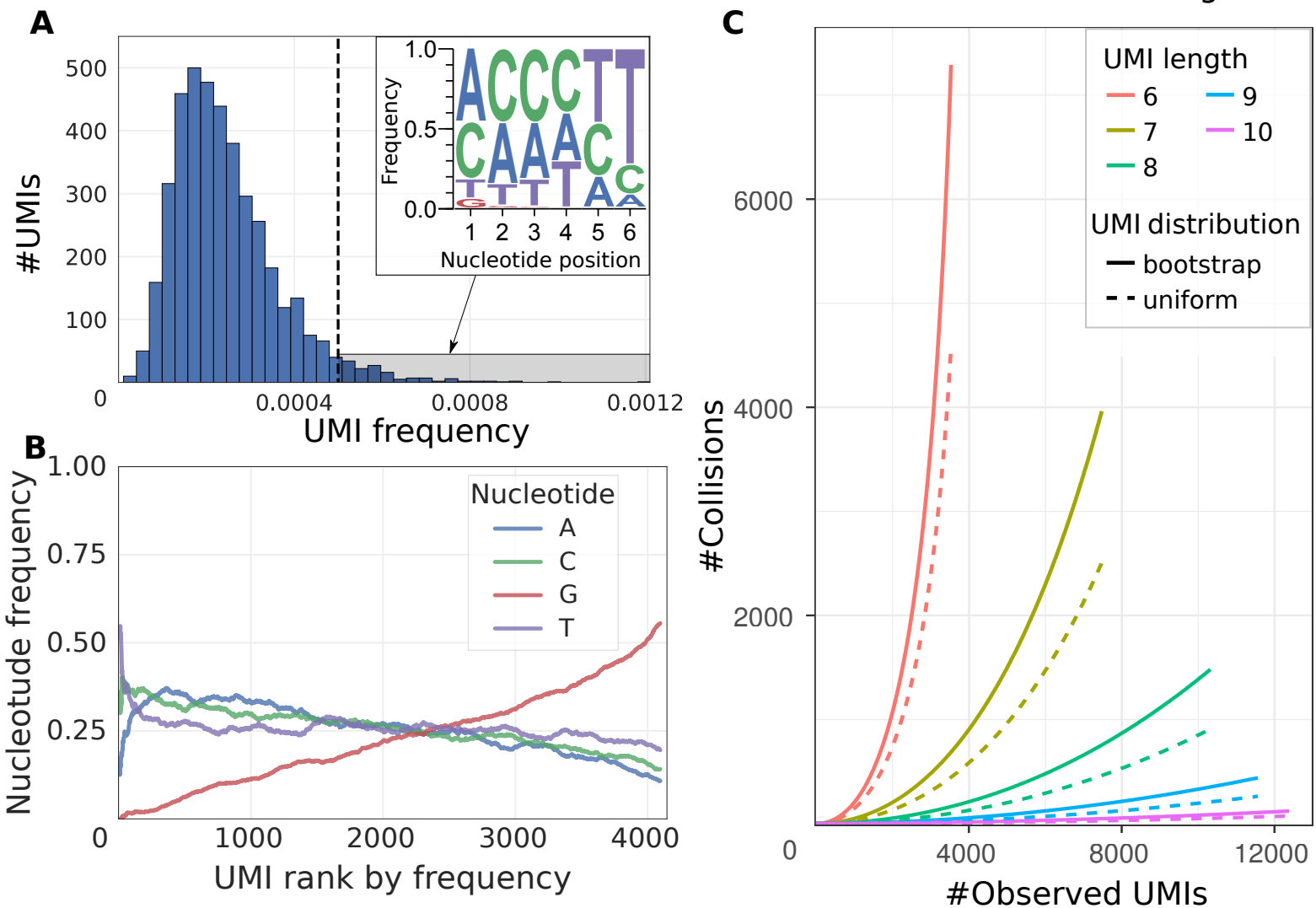

A
#UMIs
500
400
300
200
100
0
0.0004
0.0008
0.0012
UMI frequency
Frequency
1.0
0.5
0.0
1 2 3 4 5 6
Nucleotide position
B
Nucleotude frequency
1.00
0.75
0.50
0.25
0
1000
2000
3000
4000
UMI rank by frequency
Nucleotide
A
C
G
T
C
#Collisions
6000
4000
2000
0
4000
8000
12000
#Observed UMIs
UMI length
6
7
8
9
10
UMI distribution
bootstrap
uniform

Figure 2

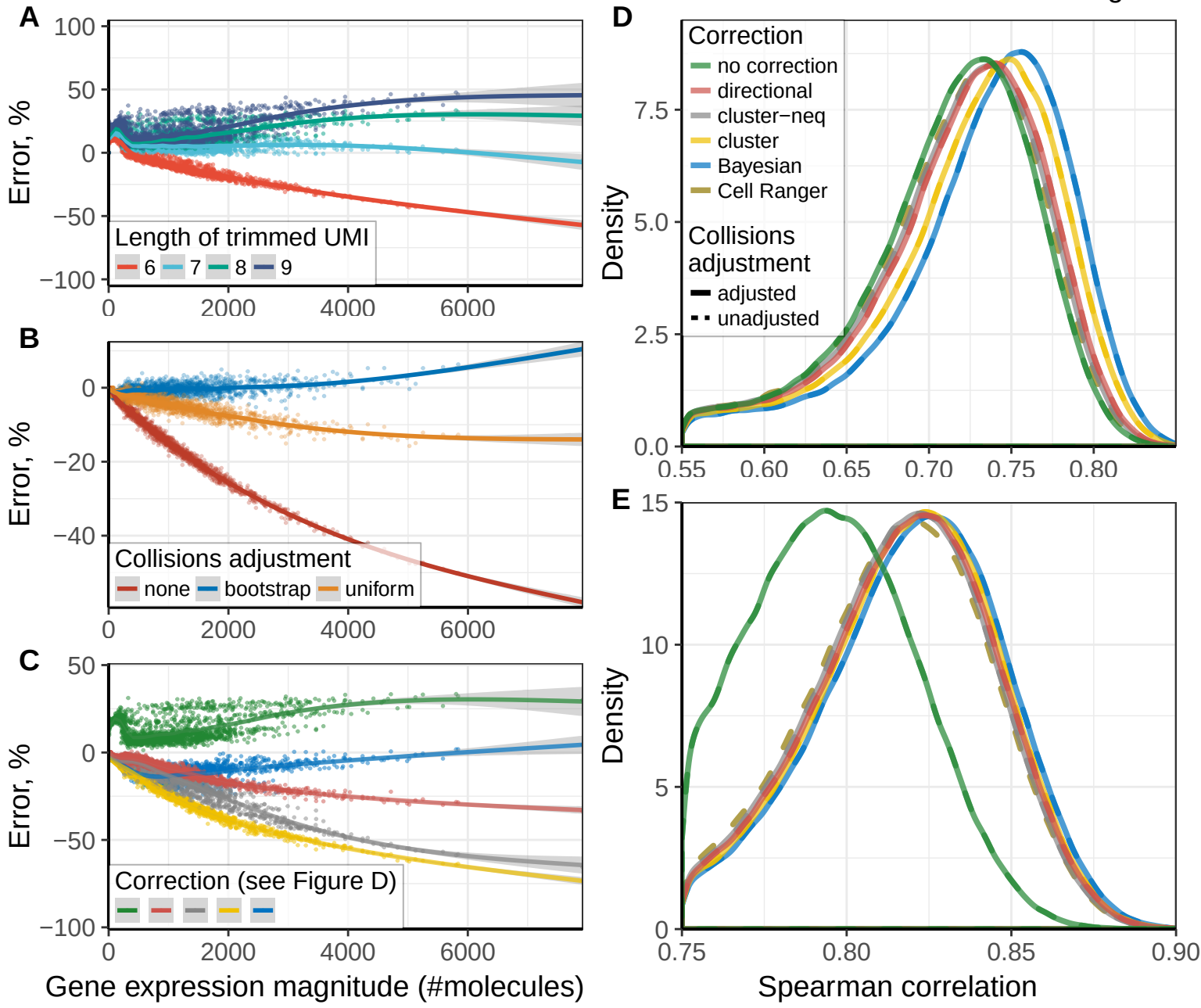

A
Error, %
Length of trimmed UMI
6
7
8
9
B
Error, %
Collisions adjustment
none
bootstrap
uniform
C
Error, %
Correction (see Figure D)
Gene expression magnitude (#molecules)
D
Density
Correction
no correction
directional
cluster−neq
cluster
Bayesian
Cell Ranger
Collisions adjustment
adjusted
unadjusted
E
Density
Spearman correlation

Figure 3

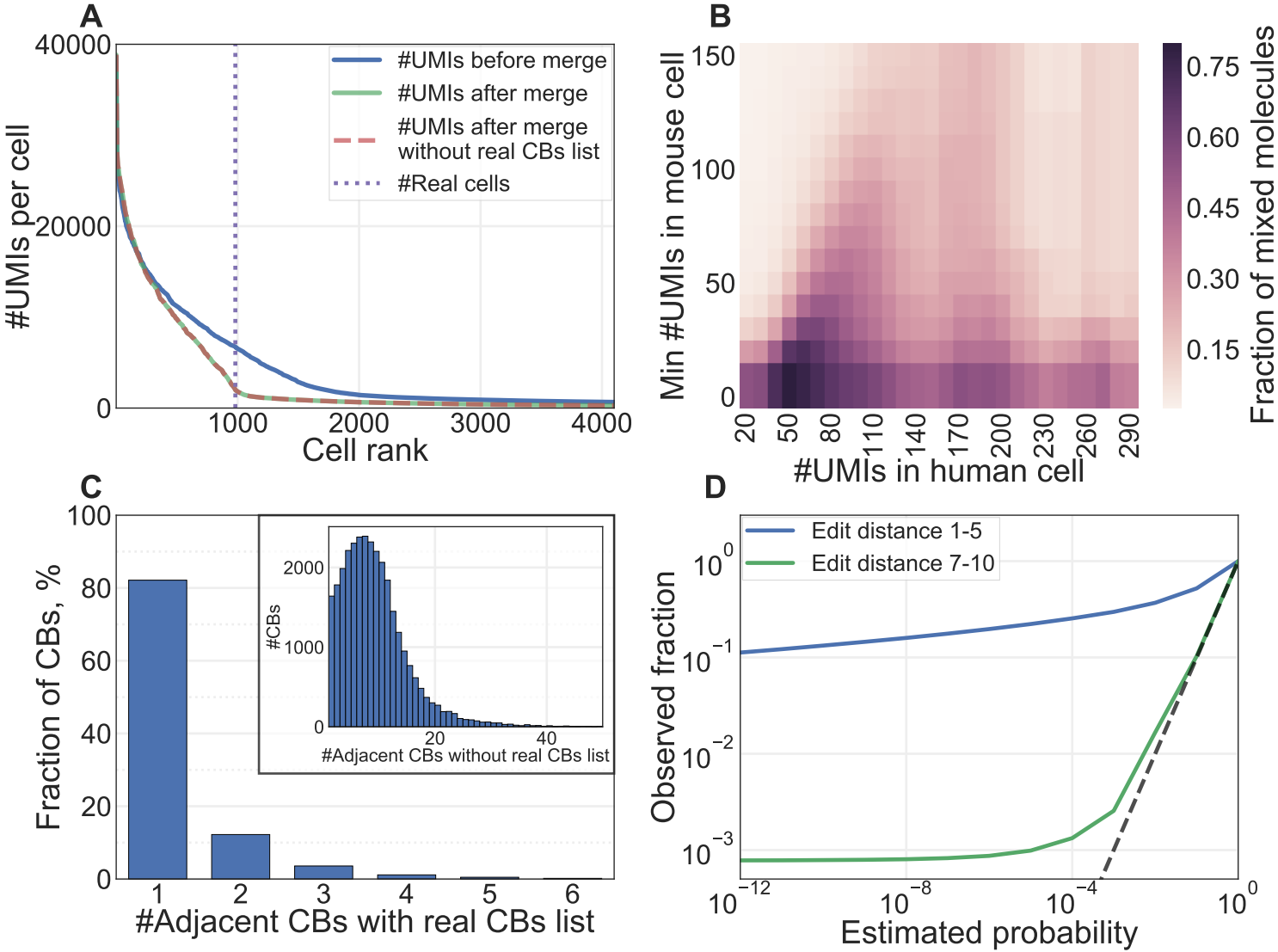
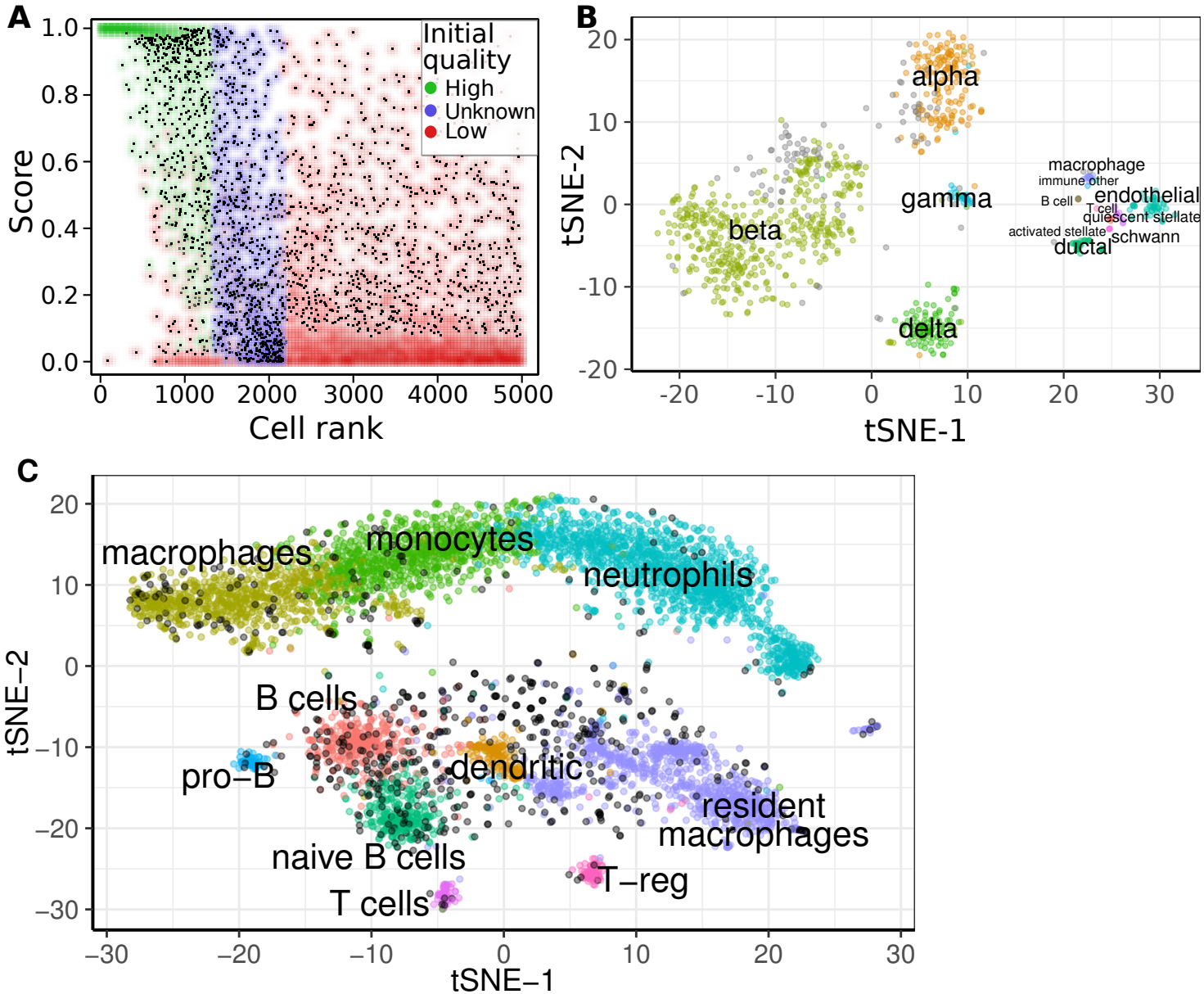

A
#UMIs per cell
Cell rank
#UMIs before merge
#UMIs after merge
#UMIs after merge without real CBs list
#Real cells
B
Min #UMIs in mouse cell
#UMIs in human cell
Fraction of mixed molecules
C
Fraction of CBs, %
#Adjacent CBs with real CBs list
#CBs
#Adjacent CBs without real CBs list
D
Observed fraction
Estimated probability
Edit distance 1-5
Edit distance 7-10

Figure 4


A
Score
Cell rank
Initial quality
High
Unknown
Low
B
tSNE-2
tSNE-1
alpha
beta
gamma
delta
macrophage
immune other
B cell
endothelial
T cell
quiescent stellate
activated stellate
schwann
ductal
C
tSNE−2
tSNE−1
macrophages
monocytes
neutrophils
B cells
pro−B
dendritic
resident macrophages
naive B cells
T−reg
T cells