Article

Patterns

EmptyNN: A neural network based on positive and unlabeled learning to remove cell-free droplets and recover lost cells in scRNA-seq data

Highlights

- The novel cell-calling algorithm EmptyNN improves the quality of scRNA-seq datasets
- EmptyNN accurately removes cell-free droplets and recovers genuine cells
- Benchmarking analyses leverage cell hashing information and genetic variation

Authors

Fangfang Yan, Zhongming Zhao, Lukas M. Simon

Correspondence

zhongming.zhao@uth.tmc.edu (Z.Z.), lukas.simon@bcm.edu (L.M.S.)

In brief

To measure the gene expression levels of an individual cell, cells are isolated into oil droplets using droplet-based single-cell RNA sequencing platforms. However, *in silico* separation of empty and cell-containing droplets in the resulting expression data is challenging. Our algorithm, called EmptyNN, improves distinction between empty and cell-containing droplets by leveraging neural networks and unlabeled-positive learning.

Yan et al., 2021, Patterns *2*, 100311
August 13, 2021
https://doi.org/10.1016/j.patter.2021.100311

CellPress

Patterns

CellPress
OPEN ACCESS

Article

EmptyNN: A neural network based on positive and unlabeled learning to remove cell-free droplets and recover lost cells in scRNA-seq data

Fangfang Yan,[1] Zhongming Zhao,[1,2,3,*] and Lukas M. Simon[4,5,*]

[1]Center for Precision Health, School of Biomedical Informatics, The University of Texas Health Science Center at Houston, Houston, TX 77030, USA

[2]Human Genetics Center, School of Public Health, The University of Texas Health Science Center at Houston, Houston, TX 77030, USA

[3]MD Anderson Cancer Center UTHealth Graduate School of Biomedical Sciences, Houston, TX 77030, USA

[4]Therapeutic Innovation Center, Baylor College of Medicine, Houston, TX 77030, USA

[5]Lead contact

*Correspondence: zhongming.zhao@uth.tmc.edu (Z.Z.), lukas.simon@bcm.edu (L.M.S.)

https://doi.org/10.1016/j.patter.2021.100311

THE BIGGER PICTURE Advances in measuring gene expression at the cellular level at high throughput have been fueled by the advent of droplet-based single-cell RNA sequencing (scRNA-seq) platforms. Droplet-based scRNA-seq platforms profile a large number of cells per experiment and accelerate our understanding of biology. Accurate classification of cell-free and cell-containing droplets will maximize biological signal and facilitate downstream analysis. Here, we present a novel cell-calling algorithm called EmptyNN, which trains a neural network based on positive-unlabeled learning for improved filtering of barcodes. Our results indicate that EmptyNN outperforms existing cell-calling methods and, thus, represents a powerful tool to enhance both scRNA-seq and single-nucleus RNA sequencing quality control analyses.

Development/Pre-production: Data science output has been rolled out/validated across multiple domains/problems

SUMMARY

Droplet-based single-cell RNA sequencing (scRNA-seq) has significantly increased the number of cells profiled per experiment and revolutionized the study of individual transcriptomes. However, to maximize the biological signal, robust computational methods are needed to distinguish cell-free from cell-containing droplets. Here, we introduce a novel cell-calling algorithm called EmptyNN, which trains a neural network based on positive-unlabeled learning for improved filtering of barcodes. For benchmarking purposes, we leveraged cell hashing and genetic variation to provide ground truth. EmptyNN accurately removed cell-free droplets while recovering lost cell clusters, and achieved an area under the receiver operating characteristics of 94.73% and 96.30%, respectively. Comparisons to current state-of-the-art cell-calling algorithms demonstrated the superior performance of EmptyNN. EmptyNN was further applied to a single-nucleus RNA sequencing (snRNA-seq) dataset and showed good performance. Therefore, EmptyNN represents a powerful tool to enhance both scRNA-seq and snRNA-seq quality control analyses.

INTRODUCTION

Droplet-based single-cell RNA sequencing (scRNA-seq) has significantly increased the number of cells profiled per experiment. As a result, droplet-based scRNA-seq enables the profiling of transcriptomes from thousands, sometimes up to several millions, of cells and provides unprecedented resolution into complex biological systems.[1,2] In a typical experiment, the viable cells in the samples are dissociated to generate a cell suspension. Every single cell in the suspension is combined with a gel bead to form a droplet containing unique barcodes. Ideally, each droplet contains one bead and one cell, which we define as a singlet (Figure 1A). Droplets with two or more cells are defined as doublets or multiplets. These types of droplets are cell-containing droplets. Droplets without cells are defined as empty droplets or cell-free droplets, which are expected to lack any RNA molecule.

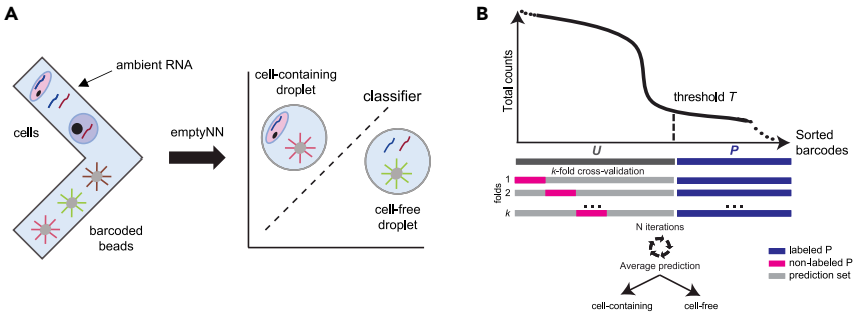

Figure 1. EmptyNN leverages positive-unlabeled learning to classify cell-free and cell-containing droplets

(A) Cells and barcodes are combined in oil droplets. Some droplets may lack a cell but contain ambient RNA. The EmptyNN classifier distinguishes cell-free from cell-containing droplets.

(B) Schematic describing the workflow of EmptyNN. The black curve represents the distribution of total counts (y axis) across sorted barcodes (x axis). The blue bars represent a set of barcodes with very low total counts, set *P*. The gray bars represent barcodes with higher total counts consisting of cell-containing and cell-free droplets, set *U*. EmptyNN trains a classifier, where barcodes from *P* are labeled as cell-free droplets (blue) and a fraction of barcodes from *U* are labeled as cell-containing droplets (pink). The classifier is applied to the remaining barcodes in *U* and the predictions are recorded. During each *k* fold, each barcode in *U* is predicted $k-1$ times. This process is repeated for *N* iterations (default: 10). The average prediction probability of each barcode in *U* defines each barcode as a cell-free or cell-containing droplet.

However, during cell dissociation, there exists a certain amount of subcellular debris or free-floating mRNA in the suspension, called "ambient" RNA. The ambient RNA may enter a droplet containing a barcoded bead and form a cell-free droplet (Figure 1A).[3] The ambient RNA in the cell-free droplets may be reverse transcribed into cDNA during the library preparation, which will produce unique molecular identifier (UMI) counts in the resulting gene expression matrices. Therefore, cell-free droplets are difficult to distinguish from cell-containing droplets. Failure to remove cell-free droplets may introduce spurious biological signals into the downstream analysis.[3,4]

Computational approaches to call cells and filter barcodes in droplet-based scRNA-seq data use the following approaches. The Cell Ranger software from 10X Genomics (version 2 or lower) defines a cutoff based on the distribution of total counts. While very commonly used, this approach ignores any transcriptomic information. Thus, filtering barcodes based solely on the distribution of total counts may remove genuine cell-containing droplets with low RNA counts. Simultaneously, empty barcodes with high ambient-RNA-derived total counts may be erroneously retained.

To improve cell calling, previous work developed statistical models operating on the transcriptome profiles. Lun et al. developed EmptyDrops, which employs a Dirichlet-multinomial model to infer the transcriptome profile of cell-free droplets.[4] By estimating the deviations from this profile, EmptyDrops assigns barcodes with significant deviations as cell-containing droplets. Recently, Cell Ranger v.3 (or higher) integrated EmptyDrops into their cell-calling algorithm. DIEM is another method that uses the multinomial mixture model along with a semi-supervised expectation maximization algorithm to remove cell-free droplets.[5] In addition, machine learning approaches have been successfully applied to scRNA-seq data.[6,7] One example of the application of neural networks to filter barcodes is called CellBender.[8] It uses an unsupervised deep generative model to learn the prior distribution of gene expression profiles and estimate the background RNA profile.[8]

Here, we leverage positive-unlabeled (PU) learning to train a deep neural network targeted toward cell calling. PU learning is a paradigm of semi-supervised learning, specifically designed for the case in which labels of one class are available and the other class labels are uncertain.[9–11] There are several strategies for PU learning, which involve adaptations of conventional machine learning methods, including direct application of a standard classifier,[12] "PU bagging,"[13] and a two-step technique.[14] PU learning is a great fit for the cell-calling task because barcodes can be divided into the following two groups: (1) barcodes with very low total counts ("positives") and (2) all remaining barcodes with medium to high total counts ("negatives"). The positives can be accurately labeled as cell-free droplets. The negatives, on the other hand, can be either cell-containing or cell-free droplets. The lack of accurate labeling for the negatives is the main motivation for the application of PU learning. In addition, it has been shown that PU learning could achieve comparable classification performance with standard supervised machine learning approaches when applied to fully labeled data.[15]

In this article, we introduce EmptyNN, a novel cell-calling algorithm that distinguishes empty, or cell-free, droplets from cell-containing droplets by training a neural network in droplet-based scRNA-seq data. EmptyNN implements the PU learning bagging strategy and is based on the rationale that barcodes with very low total counts represent *bona fide* cell-free droplets. By applying EmptyNN to two ground-truth datasets and two additional datasets, we demonstrate that EmptyNN accurately discriminates between cell-free and cell-containing droplets while recovering lost cell clusters with high accuracy. In our benchmarking analysis, EmptyNN outperformed the current cell-calling methods Cell Ranger v.2, EmptyDrops, Cell Ranger v.3, and CellBender.

RESULTS

In this work, we leveraged cell hashing information and genetic variation to provide ground truth for the evaluation of our approach. We first introduced the algorithm and then compared EmptyNN to current state-of-the-art cell-calling algorithms. We conducted comprehensive benchmarking analysis. Next, we applied EmptyNN to two additional datasets and evaluated its performance. Last, we compared the run time and computational requirements of the different methods.

The following section briefly describes the computational principles underlying EmptyNN. Given samples belonging to a specific class *P* and an unlabeled set *U*, which contains both *P* and *non-P* classes, the goal of PU learning is to build a binary classifier to classify *U* into two classes, *P* and *non-P*.[13,16,17] The rationale of EmptyNN is that barcodes with very low total counts

represent *bona fide* cell-free droplets, while all other barcodes could represent either cell-free or cell-containing droplets. Therefore, we defined barcodes with total counts below a user-specified threshold T (default: 100) as set P (blue in Figure 1B). The remaining barcodes are defined as the unlabeled set U (gray), which consists of either cell-containing or cell-free droplets. Next, U is randomly split into k folds (default: 10). All barcodes within one fold are labeled as *non-P* (pink), and a model is trained to discriminate between P (blue) and *non-P* (pink) barcodes. Subsequently, the trained model is used to predict the barcodes from the remaining $k - 1$ folds from U (gray) and each prediction is saved. This procedure is conducted for each fold and separately repeated N times such that each barcode in U will be predicted $(k - 1) * N$ times. Thus, barcodes in U (gray) will be assigned a *non-P* (pink) label during training exactly N times, such that the total number of trained models equals $k * N$. Averaging these predictions for each barcode represents a quantitative measure that can be used to define barcodes as a cell-free or cell-containing droplet. By iteratively assigning labels to a small fraction of the data during the training process, the classifier is able to infer if the barcodes in U are more likely to represent cell-free or cell-containing droplets. We compared EmptyNN to the state-of-the-art cell-calling methods Cell Ranger v.2, EmptyDrops, Cell Ranger v.3, and CellBender. More details are provided under experimental procedures.

EmptyNN removes cell-free droplets and recovers lost signal in the cell hashing dataset

To evaluate EmptyNN, we first applied it to a cell hashing dataset.[18] The cell hashing technology utilizes sample-specific barcodes to allow multiplexing. Cells from different donors were labeled with unique hashtag oligonucleotides (HTOs), which readily separate donor samples (Figure S1A). Therefore, this dataset provides a unique resource to evaluate the performance of cell-calling algorithms. Specifically, the droplets that contain single or multiple HTO types were defined as singlets or doublets, respectively (Figure S1B). Droplets lacking a clear peak in distribution of HTO counts were defined as cell-free droplets. The barcode labels (e.g., doublets, singlets, cell-free droplets) derived from the HTO information were subsequently used to evaluate the performance of EmptyNN and three competing cell-calling methods (see details under quantification and statistical analysis in the experimental procedures).

Among the 39,842 barcodes evaluated, 20,833 (52.3%) were classified as cell-free droplets and 19,009 (47.7%) were classified as cell-containing droplets by EmptyNN. We observed that EmptyNN recovered 885 barcodes with total counts falling below the filtering threshold applied by the authors in the original study (200 in this case) (Figures 2A and 2B). Moreover, comparison of additional quality control metrics, including percentage of mitochondrial reads, revealed only minimal differences between the original and the recovered cells (Figure S2).

t-distributed stochastic neighbor embedding (t-SNE) analysis separated these recovered low-total-count barcodes into five unique clusters (Figure 2C), suggesting that they represent genuine cell-containing droplets of different cell types. Subsequent differential expression analysis revealed the cell-type identities of these five clusters (Figure 2D). Four cell types (B, CD4, natural killer [NK], and CD14 monocytes) were present in the original study. Of note, platelets were detected only in the recovered barcodes and were missed in the original study (Figure 2C). Since platelets contain much less RNA compared with other cell types, they are likely to be erroneously excluded from the original analysis in which filtering is based only on total counts. Indeed, the total counts of the recovered platelets were below the original filtering threshold (range 51–196, median 93.5).

To confirm the identity of the recovered platelets, we integrated an independent dataset profiling peripheral blood mononuclear cells (PBMCs) (see details in "Reference PBMC 3k dataset" in the experimental procedures). This dataset contained a cluster of platelets that was used as a reference profile in our study. The comparison revealed a significant correlation between the gene expression profiles of *bona fide* platelets and our recovered platelets (Pearson correlation, Rho = 0.95, $p < 2.2 \times 10^{-6}$, Figure 2E). Furthermore, the *bona fide* platelet reference expression profile has the strongest correlation with the recovered platelet cluster, demonstrating that the recovered low-count barcodes represented genuine platelets.

To benchmark our method, we applied four additional cell-calling algorithms: Cell Ranger v.2, EmptyDrops, Cell Ranger v.3, and CellBender (Figure S3). Next, we divided barcodes into the following three groups: (1) barcodes predicted to be cell-containing droplets by all methods ("All-retained"), (2) barcodes that were specifically retained by EmptyNN but none of the other methods ("EmptyNN-retained"), and (3) barcodes that were specifically removed by EmptyNN but retained by all other methods ("EmptyNN-removed"). To evaluate how these sets of barcodes differed from one another, we contrasted All-retained barcodes with the EmptyNN-removed and EmptyNN-retained barcodes within the same cell type by performing differential expression analysis. Our hypothesis was that cell-containing droplets showed greater transcriptional similarity with All-retained barcodes of the same cell type compared with cell-free droplets. To remove any bias from unbalanced numbers of cells in either group, as well as total UMI count, we downsampled counts and cells. We observed a much larger number of genes with statistically significant differences in the comparison of EmptyNN-removed with EmptyNN-retained (Figure 2F). For example, in CD4 T cells, 3,898 genes were differentially expressed (adjusted $p < 0.01$) comparing All-retained with EmptyNN-removed, while only 155 genes were differentially expressed comparing All-retained with EmptyNN-retained (Figure 2G). These results demonstrated the benefits of EmptyNN over other methods.

Furthermore, to quantitatively compare the cell-calling methods, we integrated information derived from the HTO counts. As described above, the HTO counts provide singlet, doublet, and negative labels for each barcode. Barcodes labeled as singlets and doublets were classified as cell-containing barcodes, and the accuracy of the five cell-calling methods was evaluated. EmptyNN achieved an AUROC (area under the receiver operating characteristics) of 94.73% (Figure 2H). In contrast, Cell Ranger v.2, Cell Ranger v.3, EmptyDrops, and CellBender achieved AUROCs of only 86.88%, 90.16%, 79.87%, and 88.80%, respectively. To visualize these results, we created a count matrix composed of all cells detected by any of these

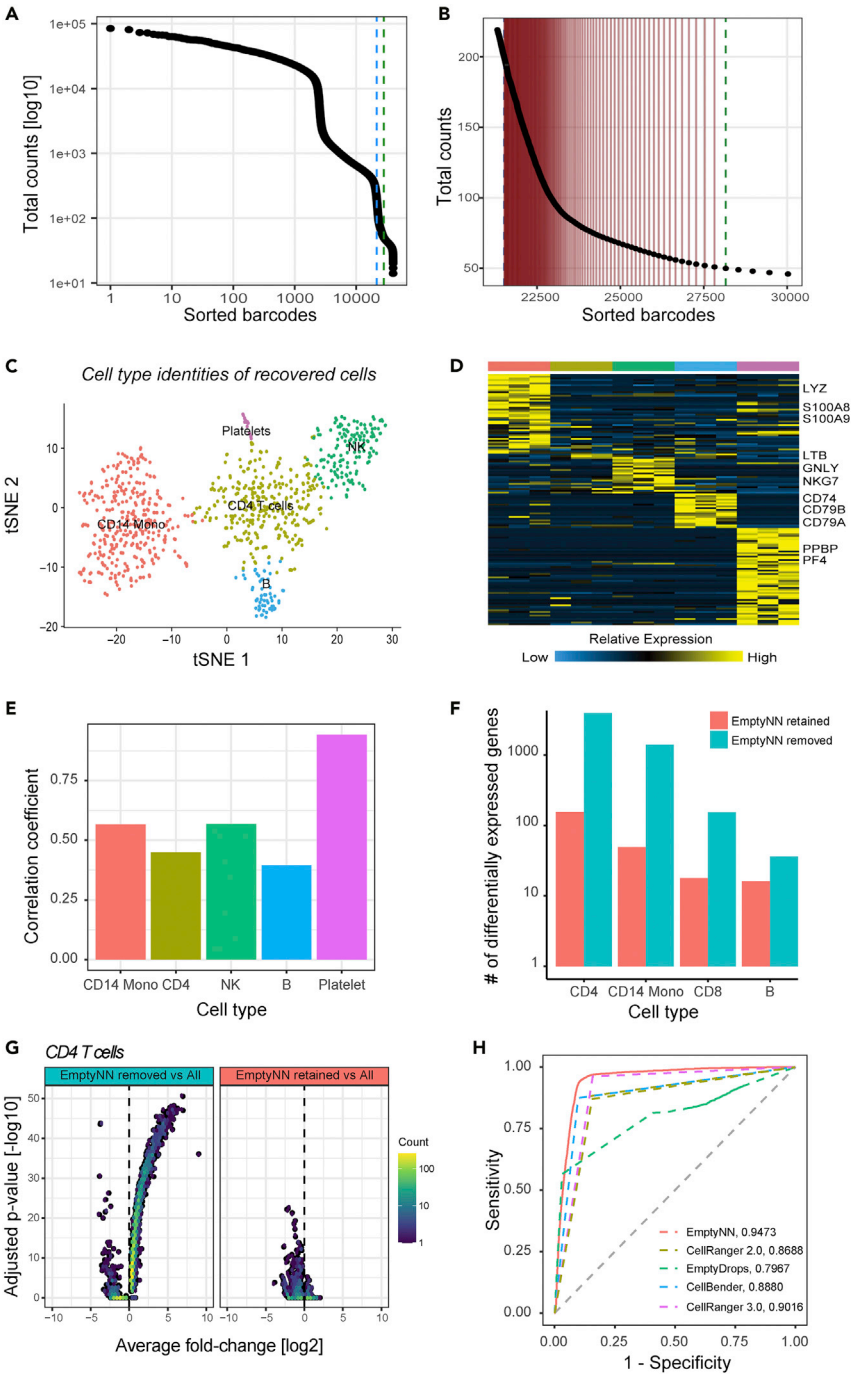

Figure 2. EmptyNN accurately removes cell-free droplets and recovers cell-containing droplets in the cell hashing dataset

(A) Barcode-rank plot shows the distribution of total UMI counts of each barcode in descending order. The two dashed lines represent 200 and 50 total UMI counts, respectively.

(B) A zoomed-in view of the barcode-rank plot highlighting barcodes with more than 50 and less than 200 total UMI counts. Vertical red lines indicate barcodes falling below the original threshold (200) but predicted to be cell-containing droplets by EmptyNN, which we referred to as "recovered cells."

(C) t-SNE plot of the recovered cells (n = 885) shows distinct expression profiles of various cell types.

(D) Heatmap illustrates known marker gene expression profiles derived from each recovered cluster.

(E) Bar plot illustrates the correlation coefficient (y axis) of mean expression profiles between the reference platelet data and all other cell types present in the recovered barcodes.

(F) Bar plot shows the number of differentially expressed genes (adjusted $p < 0.01$) comparing "All-retained" cells to "EmptyNN-retained" (red) and to "EmptyNN-removed" (blue) cells.

(G) Volcano plot shows adjusted p value (y axis) and fold change (x axis) of differentially expressed genes in CD4 T cells. Left: EmptyNN-removed versus all. Right: EmptyNN-retained versus all.

(H) ROC curves show the overall accuracy of different cell-calling algorithms.

five methods. The standard analysis pipeline was applied to this count matrix, followed by the unsupervised clustering and t-SNE visualization. The cell-type identity of each cluster was inferred based on the HTO information and differential expression analysis results (Figures S4A and S4B). Compared with Cell Ranger v.2 and v.3, EmptyNN retained more CD14 monocytes, while discarding the ambient RNA cluster (Figures S4C, S4D, and S4F). Furthermore, EmptyNN retained more B cells and CD4 cells compared with EmptyDrops (Figures S4C and S4E). CellBender kept most barcodes, including doublet and ambient RNA clusters (Figure S4G). In summary, EmptyNN demonstrated superior accuracy compared with the other three cell-calling methods.

EmptyNN accurately classifies singlets and ambiguous droplets in the multiplexed PBMC dataset

We next assessed the performance of EmptyNN in a second independent scRNA-seq dataset from Kang et al.[19] PBMCs from eight individuals were pooled and then sequenced simultaneously. In the original study, the authors developed a computational tool called demuxlet, which utilizes the natural genetic variations contained in the sequencing reads to deconvolute the donor of origin for each barcode. For each barcode, demuxlet calculates the likelihood that the sequence reads originated from one or multiple individuals. Barcodes with non-discriminant probabilities are classified as ambiguous droplets, which are the results of ambient RNAs from cell-free droplets. Based on this rationale, we applied demuxlet to infer the label of each barcode and evaluated the performance of the cell-calling methods.

For a random classifier with low capacity to distinguish between classes, the barcodes retained or discarded will have

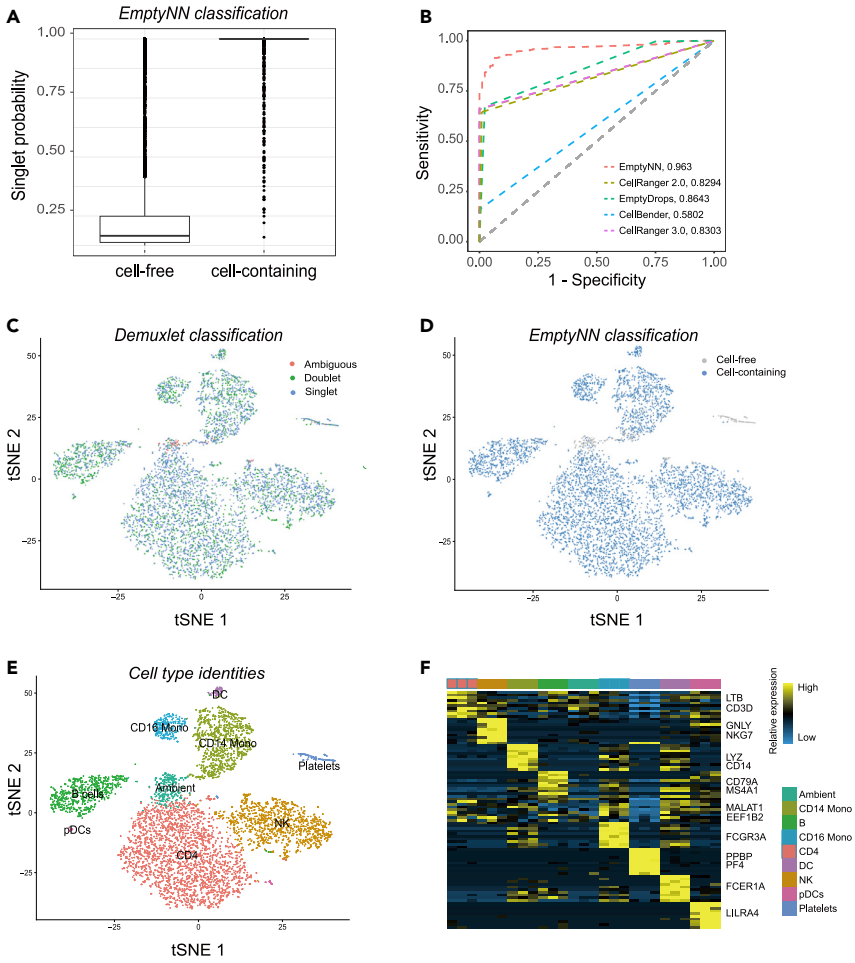

Figure 3. EmptyNN accurately classifies singlets and ambiguous droplets in the multiplexed PBMC dataset

(A) Boxplot showing the probability of being singlets for barcodes retained and discarded by EmptyNN. The box represents the interquartile range, the horizontal line in the box is the median, and the whiskers represent 1.5 times the interquartile range.

(B) ROC curve showing the performance of different algorithms.

(C–E) t-SNE plots visualizing embedding of cells called by any of five algorithms. Points represent barcodes and are colored according to (C) demuxlet-derived information, (D) whether detected by EmptyNN, and (E) putative cell type.

(F) Heatmap showing the gene expression profile for each putative cell type.

indistinguishable distributions. In contrast, for a classifier with high capacity, the distribution of retained or discarded barcodes should set them apart from one another. Figure 3A illustrates the probability distribution of barcodes retained and discarded by EmptyNN. As expected, the median singlet probability derived from demuxlet was 1.00 (SD 0.018, range 0.32–1.00) for retained barcodes and 0.17 (SD 0.21, range 0.12–1.00) for discarded barcodes, suggesting its capacity for differentiating singlets and ambiguous droplets. According to the labels derived from genetic information, EmptyNN retained a total of 6,025 cell-containing droplets and achieved an AUROC of 96.30% (Figure 3B). The total number of retained droplets for Cell Ranger v.2, EmptyDrops, Cell Ranger v.3, and CellBender was 6,061, 6,544, 6,075, and 1,686, respectively. The corresponding AUROC was 82.94%, 86.43%, 83.03%, and 58.02%, respectively (Figures 3B and S5).

Next, we investigated the t-SNE embeddings constructed from the count matrix composed of all cells detected by any of these five methods. Each droplet was labeled based on the demuxlet-derived information (Figure 3C). EmptyNN correctly discarded the "ambiguous" cluster (Figure 3D), while conserving the transcriptional profiles of the cell type identities present in the study (Figures 3E and 3F). Figure S6 shows the t-SNE plots for the other methods. In summary, EmptyNN outperformed competing methods based on the accuracy inferred from genetic variation.

EmptyNN decreases ambient contamination in single-nucleus RNA-seq data

Compared with scRNA-seq technology, single-nucleus RNA-seq (snRNA-seq) uses nuclei rather than cells and is suitable for solid tissues where isolation of individual cells is difficult.[20] snRNA-seq data usually contain lower total counts per nucleus, making it more challenging to define a good threshold to call nucleus-containing droplets. Therefore, we applied EmptyNN to snRNA-seq data from the adult mouse brain to demonstrate its broad utility. EmptyNN identified 2,222 nucleus-containing droplets, while Cell Ranger v.2, EmptyDrops, Cell Ranger v.3, and CellBender detected 2,371, 2,181, 2,566, and 4,386 nucleus-containing droplets, respectively. To evaluate the performance of each method, we calculated the ambient RNA expression signature and fraction of spliced reads for each called nucleus. Since the RNA is derived from the nucleus, most reads are expected to be unspliced. Thus, a high proportion of spliced reads indicates contamination of cytoplasmic origin and implies that the droplet is nucleus free. For example, the ambient expression signature was lower in the EmptyNN compared with the Cell Ranger v.3 filtering (Figure 4A). In addition, the distribution of the fraction of spliced reads was slightly, yet significantly, lower ($p = 3.29 \times 10^{-5}$) in EmptyNN compared with Cell Ranger v.3 filtered nuclei (Figure 4B). Indeed, the distribution of the fraction of spliced reads was lower in EmptyNN filtering compared with any other method (Figure 4C). In summary, these results demonstrate that EmptyNN performs equally strongly when applied to snRNA-seq data.

EmptyNN recovers biological signals in additional datasets

To further demonstrate the utility of EmptyNN, we analyzed three additional scRNA-seq datasets. The first two datasets were (1) the PBMC 8k dataset and (2) the Neuron 900 dataset. The datasets were processed by Cell Ranger v.2 and EmptyNN

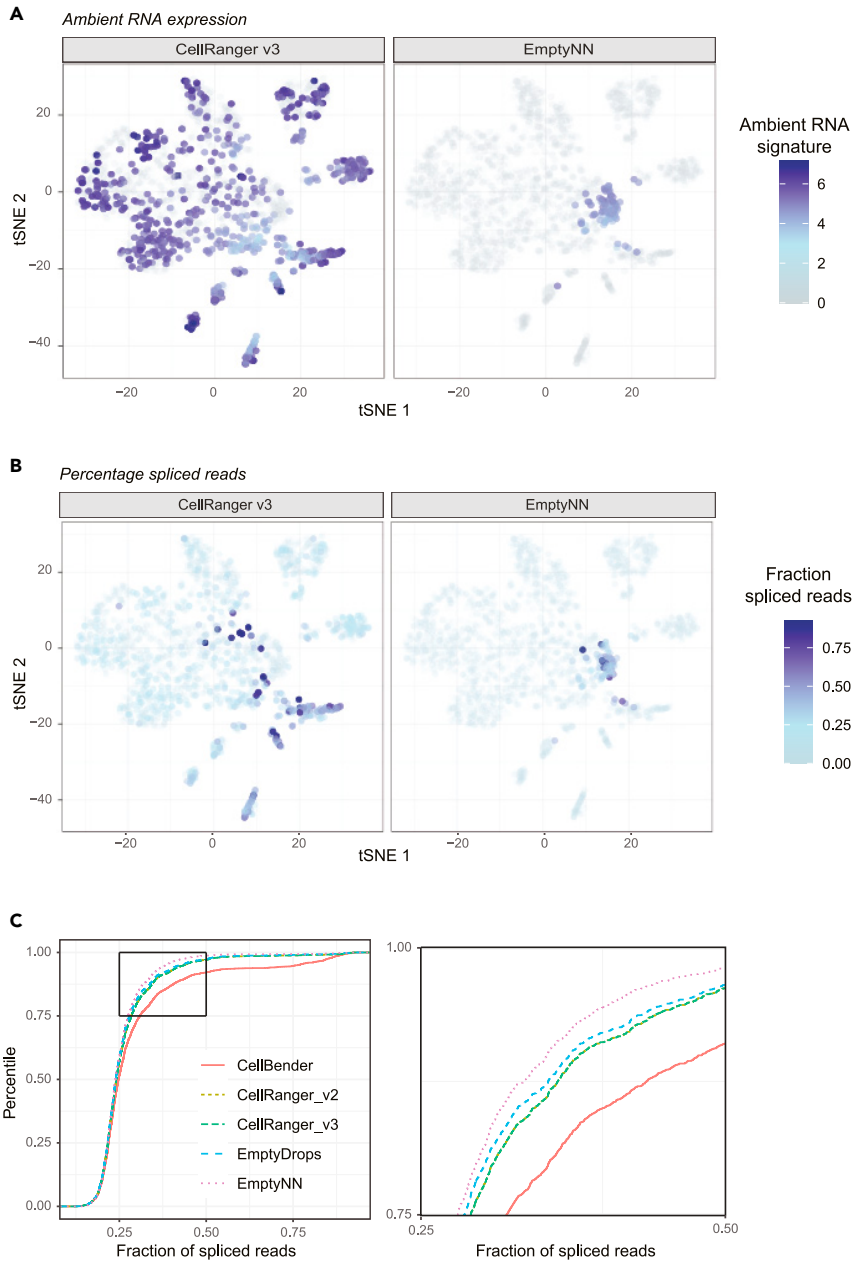

Figure 4. EmptyNN decreases contamination in single-nucleus RNA-seq data

(A and B) EmptyNN filtering shows lower (A) ambient RNA expression and (B) fraction of spliced reads compared with Cell Ranger v.3 filtering.

(C) Left: empirical cumulative distribution functions plot shows the distribution of the fraction of spliced reads in EmptyNN filtering compared with any other method. Right: zoomed-in view of the plot highlights the percentile ranging from 0.75 to 1 of different methods.

independently. We applied identical processing pipelines, including filtering of low-quality cells by mitochondrial fraction, normalization, highly variable gene detection, principal-component analysis, clustering, and t-SNE visualization (Figure S7). We observed that EmptyNN classified more cell-containing droplets compared with Cell Ranger v.2. A critical fraction of the cell-containing droplets fell below the Cell Ranger v.2 threshold, and we investigated these droplets in more detail. These cell-containing droplets formed unique clusters corresponding to different cell types. In the PBMC 8k dataset, EmptyNN uniquely retained CD4 T cells, CD14 monocytes, and platelets (Figure 5A), characterized by canonical marker genes, such as *LTB*, *CD3D*, *LYZ*, *PPBP*, and *PF4* (Figure 5A). In the Neuron 900 dataset, EmptyNN recovered GABAergic, glutamatergic, and non-neuronal clusters (Figures 5C and 5D). These results suggested that EmptyNN recovered genuine cell-containing droplets that otherwise would have been lost in a Cell Ranger v.2-based analysis.

The third dataset profiled human lung tissue and evaluated the effect of cold preservation with different cold storage times, varying from 0 to 72 h.[21] The original scRNA-seq data were processed using Cell Ranger v.3. We applied EmptyNN to six samples stored for either 0 or 72 h. To evaluate our algorithm, we calculated the mean ambient RNA expression signature for each cell. Comparing the cell-calling results between EmptyNN and Cell Ranger v.3, a decrease in ambient RNA expression was observed in the EmptyNN cell filtering across all six samples, indicating improved cell calling (Figure S8).

Hyperparameter tuning

To evaluate the impact of hyperparameter selection, we examined the performance of EmptyNN in the multiplexed PBMC dataset (Figure S9). We assessed different threshold values *T* (default: 100, from 50 to 400), numbers of cross-validation folds *k* (default: 10, from 5 to 20), and numbers of training iterations *N* (default: 10, from 5 to 20). The performance was quantified using the AUC values derived from comparing cell-calling predictions to the labels from cell hashing information, similar to the analysis described in Figures 2H and 3B. EmptyNN's performance remained very robust with respect to a large range of *k* (Figures S9B and S9F) and *N* (Figures S9C and S9G) values. Regarding the *T* parameter, our results showed highest performance for values ranging between 100 and 300 (Figures S9A and S9D). However, increased *T* values decreased the number of predicted cell-containing droplets (Figure S9E). Thus, the *T* parameter enables users to tune the stringency of the algorithm.

Run-time comparison

Finally, we compared the cell-calling methods with respect to their run time across all analyzed datasets (Figure S10). The

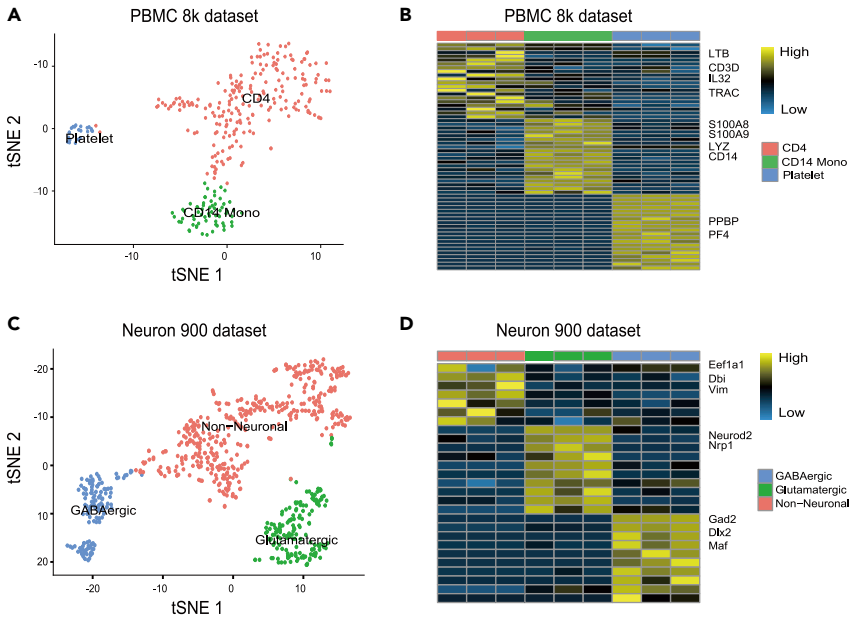

Figure 5. EmptyNN recovers biological signals in two additional datasets

(A) t-SNE plot visualizing embedding of recovered barcodes by EmptyNN in the PBMC 8k dataset. Points represent barcodes and are colored by putative cell type.

(B) Heatmap showing the gene expression profile for each cell type.

(C and D) Analogous analysis and visualizations for the neuron 900 dataset.

fastest method was EmptyDrops, which finished within minutes. EmptyNN took approximately half an hour to complete. The run time of CellBender ranged from 30 min (multiplexed PBMC dataset) to 17 h (cell hashing dataset). The results indicated that most of the methods, including EmptyNN, can complete the analysis within a reasonable time.

EmptyNN and EmptyDrops were implemented in the R environment and run on a standard personal computer. Based on the documentation, CellBender can be run in a CPU or GPU server. It takes approximately 30 min to process the full untrimmed example dataset using a CUDA-enabled GPU. In our experiments, CellBender was run on a server equipped with 24 Intel Xeon CPU E5-2630 v.2 at 2.60 GHz.

DISCUSSION

Droplet-based scRNA-seq platforms represent a significant advancement for single-cell technologies and thus have fueled remarkable progress in our understanding of cellular systems. However, to maximize the biological signal, robust computational methods are needed to distinguish cell-free from cell-containing droplets. Here, we described EmptyNN, a novel cell-calling algorithm that is based on PU learning for improved filtering of barcodes in droplet-based scRNA-seq data. We applied EmptyNN to a total of six datasets (five scRNA-seq datasets and one snRNA-seq dataset) and evaluated its performance.

In the cell hashing dataset, we utilized cell hashing information to assign labels (cell-free or cell-containing droplet) providing ground truth. EmptyNN accurately classified cell-free and cell-containing droplets. We noted that EmptyNN recovered a number of barcodes with total counts falling below the filtering threshold applied by the authors in the original study. We performed independent t-SNE and differential expression analysis to infer the cell type identities of these recovered barcodes. To confirm that these barcodes represent cells, we conducted correlation analysis and Euclidean distance comparisons to demonstrate the high levels of similarity between the recovered barcodes and those present in the original study. In our benchmarking analysis, we assessed the AUROC of each cell-calling algorithm. EmptyNN achieved an AUROC of 94.73% and outperformed current state-of-the-art cell-calling algorithms. We noticed that EmptyDrops erroneously removed most CD4 and B cells. One possible explanation is that the ''ambient'' RNA pool is a mixture of all cell types, where the most frequent cell populations likely dominate the ambient RNA profile. EmptyDrops estimates the ambient RNA profile and assesses the deviations from this profile. Thus, the RNA profile of the most frequent cell populations may not differ sufficiently and erroneously be removed.

In the multiplexed PBMC dataset, the natural genetic variation was utilized to infer the sample identity of each barcode. The generated singlet probability of each barcode is considered as the ground truth in this analysis. EmptyNN accurately differentiated singlets and ambiguous droplets and achieved an AUROC of 96.30%. We also applied EmptyNN to the PBMC 8k dataset and Neuron 900 dataset and demonstrated its good performance.

There are several limitations in our study. First, the key assumption of our approach is that barcodes with very low total counts represent *bona fide* cell-free droplets. However, this assumption may not hold when cell-containing droplets with very low total counts exist. In such cases, the user can adjust the T threshold to mitigate potential bias. Second, parameter selection, such as the number of cross-validation folds and the T threshold, needs to be specified manually. However, we consider our algorithm robust to various choices of k and T and plan to explore hyperparameter selection approaches in future work. Third, the retained cell-containing droplets may have a high fraction of mitochondrial reads. These low-quality cells may pass the initial filtering but can be removed in downstream analysis. In addition, our benchmarking analyses relied on ground truth provided by cell hashing or genetic information. As there may exist high overlap in cell-calling classifications across methods, the biological interpretation of differences may be limited by low signal. Finally, ambient RNAs will remain in cell-containing droplets and contaminate the gene expression estimates. EmptyNN does not estimate corrected gene expression profiles. To correct the impact of ambient RNAs on gene expression estimates additional tools such as SoupX[3] need to be applied.

In summary, we introduced a novel cell-calling algorithm called EmptyNN, which is a neural network based on PU learning to discriminate cell-free from cell-containing droplets in scRNA-seq datasets. Benchmarking analysis leveraged cell hashing and genetic variation providing ground truth, which allows for the statistical and visual comparisons of different cell-calling algorithms. We demonstrated that EmptyNN outperformed current state-of-the-art methods and accurately removed cell-free droplets while recovering genuine cells across different datasets. We expect EmptyNN to be widely applied during the pre-processing of droplet-based scRNA-seq datasets, which will improve the downstream analysis.

EXPERIMENTAL PROCEDURES

Resource availability

Lead contact

Further information and requests for resources should be directed to and will be fulfilled by the lead contact, Lukas M. Simon (lukas.simon@bcm.edu).

Materials availability

There are no physical materials associated with this study.

Data and code availability

The BAM file of the multiplexed PBMC dataset was obtained from the Sequence Read Archive website (https://www.ncbi.nlm.nih.gov/sra). All other datasets were obtained from the GEO database (https://www.ncbi.nlm.nih.gov/geo/). Detailed information can be found in Table S1. The source code and tutorials are freely available at https://github.com/lkmklsmn/empty_nn.

Method descriptions

EmptyNN

EmptyNN is a neural network based on PU learning. It takes the raw count matrix as input, where rows represent barcodes and columns represent genes. Only barcodes with total counts greater than 10 are included in the analysis. We define the set *P*, the *bona fide* empty droplets, as barcodes with total UMI counts less than threshold T (default: 100). The remaining barcodes are defined as the set *U*, the unlabeled droplets. The architecture of the neural network is composed of three layers with 128, 64, and 2 neurons each. The 2,000 most frequently detected genes in *P* are selected as input features of the network. The network is trained in 10 epochs using the binary cross-entropy loss function with the optimizer RMSprop. During each epoch, the training data are fed into the network in the batch size of 16. Global scaling normalization is conducted to eliminate the effect of the total counts. In this way, the neural network is forced to learn the important features in the *P* set rather than the total count difference. In each training process, the *U* set is split into *k* folds with each piece labeled as negative samples. Together with the *P* set, these split sets were fed into the neural network. The network is then used to predict the remaining $k-1$ folds. The process is repeated *N* times. The barcodes in the *U* set will receive $(k-1)^{*} N$ predictions. Those with an average score greater than 0.5 will be labeled cell-containing droplets, while those less than 0.5 will be labeled cell-free droplets. The output is a list, containing a Boolean vector indicating it is a cell-containing or cell-free droplet, as well as the probability of each droplet in set *U*.

Cell Ranger v.2

Cell Ranger v.2 applies an arbitrary cutoff on the total UMI counts to call cells.[2] The cutoff depends on the expected number of cells, *N*. For the top *N* barcodes, the 99th percentile of the total UMI counts is then calculated, called *m*. All barcodes with total UMI counts more than *m*/10 will be considered as cells.

EmptyDrops

EmptyDrops utilizes the Dirichlet-multinomial model and estimates the profile of the cell-free droplet group.[4] Specifically, all barcodes were divided into three groups based on total UMI counts, including (1) cell-free droplet group or background group with total counts less than a low number (default: 100), (2) test group in which total counts range from 100 to knee point, and (3) cell-containing droplet group in which total counts are greater than a number (default: 200). The profile of the cell-free droplet group is first estimated. Then, each barcode in the test group will be tested for deviations from this profile. Barcodes with significant deviations will be called cell-containing droplets. EmptyDrops is implemented in the DropletUtils package (version 1.6.1). Barcodes with false discovery rate less than 0.001 will be considered as cell-containing droplets.

Cell Ranger v.3

Cell Ranger v.3 combines Cell Ranger v.2 and EmptyDrops. The first pass keeps all barcodes with total counts above the threshold applied in Cell Ranger v.2. For barcodes falling between a low UMI count (default: 100) and the threshold, those predicted to be true cells by EmptyDrops will be kept.

CellBender

CellBender is an unsupervised deep generative model to distinguish cell-containing droplets from cell-free ones in scRNA-seq data.[8] By utilizing a neural network, CellBender simultaneously learns the prior distribution of gene expression profiles and estimates the background RNA profile. The estimated gene expression profiles are fit with a negative binomial model to calculate the probability of each droplet containing a cell. The droplets with probability exceeding 0.5 are considered cell-containing droplets.

In our study, CellBender *remove-background* was applied to the datasets. The number of epochs was set to 150 and the learning rate was set to 1×10^{-4} as default. The expected number of cells depends on each dataset. The barcode rank plot for each dataset was examined to determine the optimal parameter *total-droplets-included*.

Quantification and statistical analysis

Cell hashing dataset

The cell hashing dataset from Stoeckius et al.[18] utilized a multiplexing technology, which uses a unique barcoding strategy, which enables different samples to be multiplexed and sequenced together. Human PBMCs from eight donors (referred to as donor A to H) were separately extracted and labeled with unique HTOs. The cells from different samples were subsequently pooled and sequenced through standard scRNA-seq protocols. Both the RNA transcripts and the sample unique HTO levels were obtained.

The authors applied a hard threshold to the raw count matrix and kept only barcodes with more than 200 total UMI counts. A statistical model-based strategy was developed to classify each barcode. Briefly, each HTO level was fit into a negative binomial distribution separately. The 99% quantile was used as a cutoff between "enriched" and "background." Barcodes with HTO level above the cutoff were labeled as "positive" and barcodes below the cutoff were labeled as "negative" for that HTO. Thus, barcodes that were positive for only one kind of HTO were singlets. Barcodes that were positive for more than one kind of HTO were doublets. Barcodes that were negative for all HTOs were cell-free droplets. Specifically, the cutoff for HTO-A to HTO-H was 52, 75, 96, 100, 101, 128, 329, and 171, respectively.

Reference PBMC 3k dataset

The reference PBMC 3k dataset was used only to validate the expression profile of the recovered platelets and contained 2,700 cells in total. Standard Seurat (version 3.2.2) pre-processing workflow was applied to this dataset, including removal of low-quality cells, normalization, feature selection, and dimension reduction. The first 10 principal components (PCs) were used to construct the KNN graph and cluster the cells with a resolution of 0.5. Cell markers that defined clusters were found by differential expression. There were nine cell types in total, including naive CD4 T cells, memory CD4 cells, CD14 monocytes, CD8 T cells, CD16 monocytes, NK cells, dendritic cells, and platelets. The platelet cluster served as a reference profile in our study. The mean gene expression profile of the reference platelet cluster and the recovered clusters was calculated and similarity was evaluated using Pearson's correlation coefficient.

Multiplexed PBMC dataset

We first downloaded the genome-aligned BAM file from the Sequence Read Archive (SRR5398237). The "bamtofastq" tool (1.3.2, https://support.10xgenomics.com/docs/bamtofastq) was used to convert the BAM file to FASTQ files, which were subsequently used as input to the Cell Ranger v.2 pipeline. The pipeline was run with default parameters to generate the unfiltered count matrix (n = 145,549 barcodes).

PBMC 8k dataset

The PBMC 8k dataset included a total of 409,508 barcodes with non-zero total counts in the raw count matrix. EmptyNN was applied to the dataset with default parameters and EmptyNN predicted a total of 9,685 barcodes to be

cell-containing droplets. The Cell Ranger v.2 filtration resulted in 8,381 barcodes.

Neuron 900 dataset

The cells in the Neuron 900 dataset came from the cortex, hippocampus, and subventricular zone of an E18 mouse. The raw count matrix contained 737,280 barcodes, with 231,912 (31.46%) barcodes having at least one gene expressed. EmptyNN was applied to the dataset with the default threshold 100. EmptyNN predicted 2,899 barcodes to be cell-containing droplets. The Cell Ranger v.2 filtered data matrix contained a total of 931 barcodes.

Inference of droplets using hashing information

In our analysis, the raw count matrix was used, which contained 50,000 barcodes, including barcodes with fewer than 200 total UMI counts. We removed barcodes without corresponding HTO information, which resulted in the exclusion of 10,158 (20.3%) barcodes. For the remaining 39,842 barcodes, the same HTO classification strategy as in the original paper was employed. In summary, barcodes were classified as 3,615 (9.07%) doublets, 19,117 (47.98%) singlets, and 17,110 (42.94%) cell-free droplets.

EmptyNN was run with five iterations. The threshold parameter T was set to 50. We used Seurat (version 3.2.2) to process the EmptyNN-filtered count matrix and conduct downstream analysis.[22] Briefly, after pre-processing the 1,000 most highly variable genes were identified. Principal-component analysis was conducted to reduce dimensionality. The first 10 PCs were used to calculate the neighborhood graph, followed by clustering and t-SNE visualization. Differential gene expression was conducted between clusters. Each cluster was labeled based on the expression of cell-type marker genes. The Enrichr database (https://maayanlab.cloud/Enrichr/) served as a complementary tool for annotating cell clusters.[23,24]

Inference of droplets using genetic variation

To assign labels to each barcode, we used demuxlet,[19] a tool to deconvolute pooled sample identities based on natural genetic variation. Demuxlet requires two inputs: (1) a BAM file containing aligned reads and (2) a VCF file containing the genotype of each pooled sample. The output contains the most likely sample identity for each barcode in the form of probabilities. The merged VCF file containing all samples was downloaded from the demuxlet GitHub repository (https://github.com/yelabucsf/demuxlet_paper_code/). Demuxlet was run via Docker using default parameters. The output contained 5,845 singlets, 2,401 doublets, and 31,700 ambiguous droplets.

Ambient RNA signature calculation

The ambient genes were defined as the top 100 most frequently expressed genes in the P set (low UMI count barcodes). The ambient RNA signature was calculated as the average expression of these ambient genes.

Differential gene expression test

We first extracted barcodes into three groups: (1) predicted to be cell-containing droplets by all methods (All-retained), (2) specifically retained by EmptyNN but none of the other methods (EmptyNN-retained), and (3) specifically removed by EmptyNN but retained by all other methods (EmptyNN-removed). Next, we downsampled counts and cells to remove any bias from unbalanced numbers of cells in either group as well as total UMI count. The differential expression analysis was conducted using *FindMarkers()* in the Seurat package. The number of significant genes between different contrasts was then compared.

Comparison of accuracies

Labels derived from cell hashing or genetic variation information were used to calculate accuracies. EmptyNN and EmptyDrops output the probability for each barcode, while Cell Ranger v.2, Cell Ranger v.3, and CellBender output a Boolean vector indicating whether the droplet was predicted to be cell free or cell containing. We utilized the "pROC" package (version 1.16.2) in R to calculate the overall accuracy of each cell-calling algorithm. Threshold values from 0 to 1 were applied to generate sensitivity and specificity. For calculating sensitivity and specificity measures as listed in Table S2, EmptyNN and EmptyDrops predictions were converted to binary outcomes. Barcodes with probability >0.5 were considered cell-containing droplets and all other barcodes were considered cell-free droplets. The calculations of sensitivity and specificity were based on the following formulas:

$$\text{Sensitivity} = TP/TP + FN,$$

$$\text{Specificity} = TN/TN + FP,$$

where TP, TN, FP, and FN represent true positives, true negatives, false positives, and false negatives, respectively.

SUPPLEMENTAL INFORMATION

Supplemental information can be found online at https://doi.org/10.1016/j.patter.2021.100311.

ACKNOWLEDGMENTS

This research was partially supported by a Cancer Prevention and Research Institute of Texas grant (CPRIT RP180734, RP210045). Z.Z. was partially supported by National Institutes of Health grants (R01LM012806, R01DE029818, R01DE030122). The funder had no role in the study design, data collection, analysis, decision to publish, or preparation of the manuscript.

AUTHOR CONTRIBUTIONS

Conceptualization, L.M.S.; methodology, L.M.S. and F.Y.; formal analysis, F.Y. and L.M.S.; data curation, F.Y.; writing – original draft, F.Y. and L.M.S.; writing – review & editing, L.M.S. and Z.Z.; visualization, F.Y. and L.M.S.; funding acquisition, Z.Z.; supervision, L.M.S. and Z.Z.

DECLARATION OF INTERESTS

The authors declare no competing interests.

Received: February 8, 2021
Revised: June 4, 2021
Accepted: June 18, 2021
Published: July 20, 2021

REFERENCES

1. Macosko, E.Z., Basu, A., Satija, R., Nemesh, J., Shekhar, K., Goldman, M., Tirosh, I., Bialas, A.R., Kamitaki, N., Martersteck, E.M., et al. (2015). Highly parallel genome-wide expression profiling of individual cells using nanoliter droplets. Cell *161*, 1202–1214. https://doi.org/10.1016/j.cell.2015.05.002.
2. Zheng, G.X., Terry, J.M., Belgrader, P., Ryvkin, P., Bent, Z.W., Wilson, R., Ziraldo, S.B., Wheeler, T.D., McDermott, G.P., Zhu, J., et al. (2017). Massively parallel digital transcriptional profiling of single cells. Nat. Commun. *8*, 14049. https://doi.org/10.1038/ncomms14049.
3. Young, M.D., and Behjati, S. (2020). SoupX removes ambient RNA contamination from droplet based single-cell RNA sequencing data. GigaScience *9*, giaa151. https://doi.org/10.1093/gigascience/giaa151.
4. Lun, A.T., Riesenfeld, S., Andrews, T., Gomes, T., and Marioni, J.C. (2019). EmptyDrops: distinguishing cells from empty droplets in droplet-based single-cell RNA sequencing data. Genome Biol. *20*, 63. https://doi.org/10.1186/s13059-019-1662-y.
5. Alvarez, M., Rahmani, E., Jew, B., Garske, K.M., Miao, Z., Benhammou, J.N., Ye, C.J., Pisegna, J.R., Pietiläinen, K.H., Halperin, E., et al. (2020). Enhancing droplet-based single-nucleus RNA-seq resolution using the semi-supervised machine learning classifier DIEM. Sci. Rep. *10*, 11019. https://doi.org/10.1038/s41598-020-67513-5.
6. Angerer, P., Simon, L., Tritschler, S., Wolf, F.A., Fischer, D., and Theis, F.J. (2017). Single cells make big data: new challenges and opportunities in transcriptomics. Curr. Opin. Syst. Biol. *4*, 85–91. https://doi.org/10.1016/j.coisb.2017.07.004.
7. Simon, L.M., Yan, F., and Zhao, Z. (2010). DrivAER: Identification of driving transcriptional programs in single-cell RNA sequencing data. Gigascience *9*, giaa122. https://doi.org/10.1093/gigascience/giaa122.
8. Fleming, S.J., Marioni, J.C., and Babadi, M. (2019). CellBender remove-background: a deep generative model for unsupervised removal of background noise from scRNA-seq datasets. BioRxiv. https://doi.org/10.1101/791699.

9. Denis, F. (1998). PAC learning from positive statistical queries. In Algorithmic Learning Theory, pp. 112–126. https://doi.org/10.1007/3-540-49730-7_9.

10. Comité, F.D., De Comité, F., Denis, F., Gilleron, R., and Letouzey, F. (1999). Positive and unlabeled examples help learning. In Lecture Notes in Computer Science, pp. 219–230. https://doi.org/10.1007/3-540-46769-6_18.

11. Letouzey, F., Denis, F., and Gilleron, R. (2000). Learning from positive and unlabeled examples. In Algorithmic Learning Theory, pp. 71–85. https://doi.org/10.1007/3-540-40992-0_6.

12. Elkan, C., and Noto, K. (2008). Learning classifiers from only positive and unlabeled data. In Proc. 14th ACM SIGKDD Int. Conf. Knowledge Discov. Data mining, pp. 213–220. https://doi.org/10.1145/1401890.1401920.

13. Mordelet, F., and Vert, J.P. (2014). A bagging SVM to learn from positive and unlabeled examples. Pattern Recognit. Lett. *37*, 201–209.

14. Kaboutari, A., Bagherzadeh, J., and Kheradmand, F. (2014). An evaluation of two-step techniques for positive-unlabeled learning in text classification. Int. J. Comput. Appl. Technol. Res. *3*, 592–594.

15. Li, C., and Hua, X.L. (2014). Towards positive unlabeled learning for parallel data mining: a random forest framework. Adv. Data Mining Appl. 573–587. https://doi.org/10.1007/978-3-319-14717-8_45.

16. Li, X.L., and Liu, B. (2005). Learning from positive and unlabeled examples with different data distributions. In Machine Learning: ECML, pp. 218–229. https://doi.org/10.1007/11564096_24.

17. Liu, B., Lee, W.S., Yu, P.S., and Li, X. (2002). Partially supervised classification of text documents. ICML *2*, 387–394.

18. Stoeckius, M., Zheng, S., Houck-Loomis, B., Hao, S., Yeung, B.Z., Mauck, W.M., Smibert, P., and Satija, R. (2018). Cell Hashing with barcoded antibodies enables multiplexing and doublet detection for single cell genomics. Genome Biol. *19*, 1–12. https://doi.org/10.1186/s13059-018-1603-1.

19. Kang, H.M., Subramaniam, M., Targ, S., Nguyen, M., Maliskova, L., McCarthy, E., Wan, E., Wong, S., Byrnes, L., Lanata, C.M., et al. (2018). Multiplexed droplet single-cell RNA-sequencing using natural genetic variation. Nat. Biotechnol. *36*, 89–94. https://doi.org/10.1038/nbt.4042.

20. Habib, N., Avraham-Davidi, I., Basu, A., Burks, T., Shekhar, K., Hofree, M., Choudhury, S.R., Aguet, F., Gelfand, E., Ardlie, K., et al. (2017). Massively parallel single-nucleus RNA-seq with DroNc-seq. Nat. Methods *14*, 955–958. https://doi.org/10.1038/nmeth.4407.

21. Madissoon, E., Wilbrey-Clark, A., Miragaia, R.J., Saeb-Parsy, K., Mahbubani, K.T., Georgakopoulos, N., Harding, P., Polanski, K., Huang, N., Nowicki-Osuch, K., et al. (2019). scRNA-seq assessment of the human lung, spleen, and esophagus tissue stability after cold preservation. Genome Biol. *21*, 1. https://doi.org/10.1186/s13059-019-1906-x.

22. Stuart, T., Butler, A., Hoffman, P., Hafemeister, C., Papalexi, E., Mauck, W.M., III, Hao, Y., Stoeckius, M., Smibert, P., and Satija, R. (2019). Comprehensive integration of single-cell data. Cell *177*, 1888–1902.e21. https://doi.org/10.1016/j.cell.2019.05.031.

23. Kuleshov, M.V., Jones, M.R., Rouillard, A.D., Fernandez, N.F., Duan, Q., Wang, Z., Koplev, S., Jenkins, S.L., Jagodnik, K.M., Lachmann, A., et al. (2016). Enrichr: a comprehensive gene set enrichment analysis web server 2016 update. Nucleic Acids Res. *44*, W90–W97. https://doi.org/10.1093/nar/gkw377.

24. Chen, E.Y., Tan, C.M., Kou, Y., Duan, Q., Wang, Z., Meirelles, G.V., Clark, N.R., and Ma'ayan, A. (2013). Enrichr: interactive and collaborative HTML5 gene list enrichment analysis tool. BMC Bioinformatics *14*, 128. https://doi.org/10.1186/1471-2105-14-128.