nature methods

Article

https://doi.org/10.1038/s41592-023-01943-7

Unsupervised removal of systematic background noise from droplet-based single-cell experiments using CellBender

Received: 10 July 2022

Accepted: 13 June 2023

Published online: 7 August 2023

Check for updates

Stephen J. Fleming[1,2] ✉, Mark D. Chaffin[2,3], Alessandro Arduini[2,8], Amer-Denis Akkad[4], Eric Banks[1], John C. Marioni[5,6], Anthony A. Philippakis[1], Patrick T. Ellinor[2,3,7] & Mehrtash Babadi[1,2] ✉

Droplet-based single-cell assays, including single-cell RNA sequencing (scRNA-seq), single-nucleus RNA sequencing (snRNA-seq) and cellular indexing of transcriptomes and epitopes by sequencing (CITE-seq), generate considerable background noise counts, the hallmark of which is nonzero counts in cell-free droplets and off-target gene expression in unexpected cell types. Such systematic background noise can lead to batch effects and spurious differential gene expression results. Here we develop a deep generative model based on the phenomenology of noise generation in droplet-based assays. The proposed model accurately distinguishes cell-containing droplets from cell-free droplets, learns the background noise profile and provides noise-free quantification in an end-to-end fashion. We implement this approach in the scalable and robust open-source software package CellBender. Analysis of simulated data demonstrates that CellBender operates near the theoretically optimal denoising limit. Extensive evaluations using real datasets and experimental benchmarks highlight enhanced concordance between droplet-based single-cell data and established gene expression patterns, while the learned background noise profile provides evidence of degraded or uncaptured cell types.

Droplet-based assays have enabled transcriptome-wide quantification of gene expression at the resolution of single cells[1,2]. In a typical scRNA-seq experiment, a suspension of cells is prepared and loaded into individual droplets. PolyA-tailed mRNA species in each droplet are uniquely barcoded and reverse transcribed, followed by PCR amplification, library preparation and ultimately sequencing. Quantifying gene expression in each cell is achieved by identifying and counting unique cDNA fragments that have a particular droplet barcode. The differential PCR amplification bias on different molecules can be reduced by using unique molecular identifier barcodes (UMIs) and counting the number of unique UMIs as a proxy for unique endogenous transcripts. This count information is then summarized in a count matrix, where counts of each gene are recorded for each cell barcode. The count matrix is the starting point for downstream analyses such as batch correction, clustering and differential expression[3,4]. In addition to cellular mRNA, other cell-endogenous molecules or incorporated perturbations (hereafter referred to as cell 'features' for brevity) can be assayed using a similar set-up by conjugating the desired feature

[1]Data Sciences Platform, Broad Institute of MIT and Harvard, Cambridge, MA, USA. [2]Precision Cardiology Laboratory (PCL), Broad Institute of MIT and Harvard, Cambridge, MA, USA. [3]Cardiovascular Disease Initiative, Broad Institute of MIT and Harvard, Cambridge, MA, USA. [4]Precision Cardiology Laboratory (PCL), Bayer US, LLC, Cambridge, MA, USA. [5]Wellcome Sanger Institute, Hinxton, Cambridge, UK. [6]European Molecular Biology Laboratory, European Bioinformatics Institute, Cambridge, UK. [7]Cardiovascular Research Center, Massachusetts General Hospital, Boston, MA, USA. [8]Present address: Bayer US, LLC, Cambridge, MA, USA. ✉e-mail: sfleming@broadinstitute.org; mehrtash@broadinstitute.org

with a cellular barcode. Examples include CITE-seq[5], Perturb-seq[6], scCAT-seq[7], SNARE-seq[8], SHARE-seq[9], ECCITE-seq[10] and 10x Multiome, among many other recently introduced droplet-based assays.

To reduce the rate of events in which multiple cells are encapsulated in the same droplet, the cell suspension is appropriately diluted, and, as a result, a typical droplet-based single-cell experiment produces hundreds of thousands of cell-free droplets. In an ideal scenario, a cell-free droplet is expected to be truly devoid of capturable molecules, whereas a cell-containing droplet will yield features originating only from the encapsulated cell. In reality, however, neither expectation is met. On the one hand, the cell suspension contains a low-to-moderate concentration of cell-free mRNA molecules or other capturable features (Fig. 1a), which leads to nonzero molecule counts even in cell-free droplets[11] (Fig. 1b). These cell-free molecules, also referred to as 'ambient' molecules, have their origin in either ruptured or degraded cells, residual cytoplasmic debris (for example, in snRNA-seq) or exogenous sources such as unbound single-stranded DNA-conjugated antibodies or sample contamination. On the other hand, the shedding of capture oligonucleotides by beads in microfluidic channels as well as the formation of spurious chimeric molecules during the bulk mixed-template PCR amplification[12,13] effectively lead to 'swapping' of transcripts and barcodes across droplets. The severity of these problems depends on the tissue isolation protocol as well as library-preparation steps, including purification, size selection, PCR amplification conditioning and the number of cycles[14]. For a more thorough discussion, see Supplementary Section 1.1.

Mixed-species experiments provide a direct demonstration of the effects of systematic background noise, as shown in Fig. 1c, where an experiment with a mixture of human and mouse cells is observed to have hundreds of off-target human transcripts in all droplets that contain mouse cells (inset), when ideally, mouse cell-containing droplets would have zero human transcripts (excluding doublets, where two cells are captured in one droplet). The issue of background counts is particularly problematic in snRNA-seq. The harsh nuclear isolation protocols produce a substantial number of ruptured nuclei and a high concentration of cytoplasmic RNA in the suspension (Fig. 1d, green dots). In severe cases, the typical total UMI-count distinction between droplets with and without nuclei nearly disappears and all droplets lie on a continuum of counts. In such situations, successful downstream analysis hinges on our ability to (1) distinguish empty from non-empty droplets and (2) correctly recover the counts from encapsulated cells or nuclei while removing background counts.

The presence of background counts can reduce both the magnitude and the specificity of differential signal across different cell types. In cases in which quantitative accuracy or specificity is required, for example, for identification of exclusive marker genes as a part of drug target discovery or the study of subtle phenotypic alterations in a case–control setting, background counts can obscure or even completely mask the signal of interest. In some experiments, extremely high expression of a particular gene in one cell type can give rise to a large amount of background, making it seem as though all cells express the gene at a low level. This issue is common to antibody features in CITE-seq and single-guide RNA CRISPR guides in Perturb-seq.

As the field of single-cell omics is rapidly extending beyond unimodal measurements and toward multimodality[15], the issue of systematic background noise remains a ubiquitous artifact that negatively impacts all such assays, regardless of the measured feature. A general-purpose in silico mitigation strategy is therefore expected to be of wide applicability. Here, we introduce a deep generative model for inferring cell-free and cell-containing droplets, learning the background noise profile and retrieving uncontaminated counts from cell-containing droplets. Our proposed algorithm operates end-to-end starting from the raw counts, is fully unsupervised, is agnostic to the nature of the measured molecular feature (for example, mRNA, protein and so on) and requires no assumptions or prior biological knowledge of either cell types or cell type-specific gene expression profiles. A major challenge in distinguishing background noise counts from biological counts for single droplets is the extreme sparsity of counts, such that, without a strong informative prior, the counts obtained from a single droplet do not provide sufficient statistical power to allow inference of background contamination. Here, we use a neural network to learn the distribution of gene expression across all droplets. The learned distribution acts as a prior over cell-endogenous counts, provides a mechanism to share statistical power between similar cells and ultimately improves the estimation of background noise counts. Learning this neural prior of cell states and estimating the background noise profile is performed simultaneously and self-consistently within a variational inference framework, allowing progressively improved separation of endogenous and background counts during model training.

We present extensive evaluation of our algorithm on both simulated and real datasets (whole-cell, single-nuclei, mixed-species and CITE-seq datasets). We show that (1) our method is superior to the currently existing methods in distinguishing empty and cell-containing droplets, in particular, in ambiguous regimes and challenging snRNA-seq datasets, and (2) our method successfully learns and subtracts background noise counts from cell-containing droplets and leads to substantially increased amplitude and specificity of differential expression, both for RNA and CITE-seq antibody counts and increases the correlation between the two modalities. Benchmarking on mixed-species scRNA-seq experiments demonstrates that CellBender removes the majority of off-target cross-species counts. Experiments using simulated noisy datasets with known ground truth show that CellBender operates close to the theoretically optimal limit.

Our method is made available as a production-grade, easy-to-use command-line tool (Fig. 1e,f). We use the Pyro probabilistic programming framework[16] for Bayesian inference. Graphics processing unit (GPU) acceleration is necessary for fast operation of this method. We refer to this method as remove-background, which constitutes the first computational module in CellBender, an open-source software package developed by the authors for preprocessing and quality controlling single-cell omic data. Several community-standard file formats, including CellRanger, DropSeq, AnnData[17] and Loom, are accepted as input. CellBender workflows are available on Terra (https://app.terra.bio), a secure open platform for collaborative omic analysis and can be run on the cloud on a GPU with zero set-up.

Since the time our method was first made available as an open-source project in 2019, it has been extensively used by the single-cell omic community in several large-scale studies, including primary research articles on the mouse brain[18], human brain organoids[19], human intestine[20], human heart[21–23], human and mouse adipocytes[24,25], several recent studies on severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) in human tissues[26–31] and a large snRNA-seq human cross-tissue atlas[32]. Background noise removal remains a crucial step in single-cell data analysis, and other authors have developed methods for remedying ambient RNA as well, including SoupX[11] and a method for removing chimeric reads[13]. In particular, DecontX by Yang et al.[33] is another principled method, which we benchmark together with our method here.

Results

A generative model for noisy droplet-based count data

We build a probabilistic model of noise-contaminated single-cell data by examining the key steps of the data-generation process from first principles, including droplet formation and cell encapsulation, reverse transcription, PCR amplification and the consequent ambient molecules and chimeric library fragments. These mechanisms, along with the empirical evidence for each, are discussed in detail in Supplementary Section 1.1. A simplified schematic of our model is shown in Fig. 1g, along with the formal probabilistic graphical model in Fig. 1h. Our general approach to modeling is discussed in the 'Why a deep generative

https://doi.org/10.1038/s41592-023-01943-7

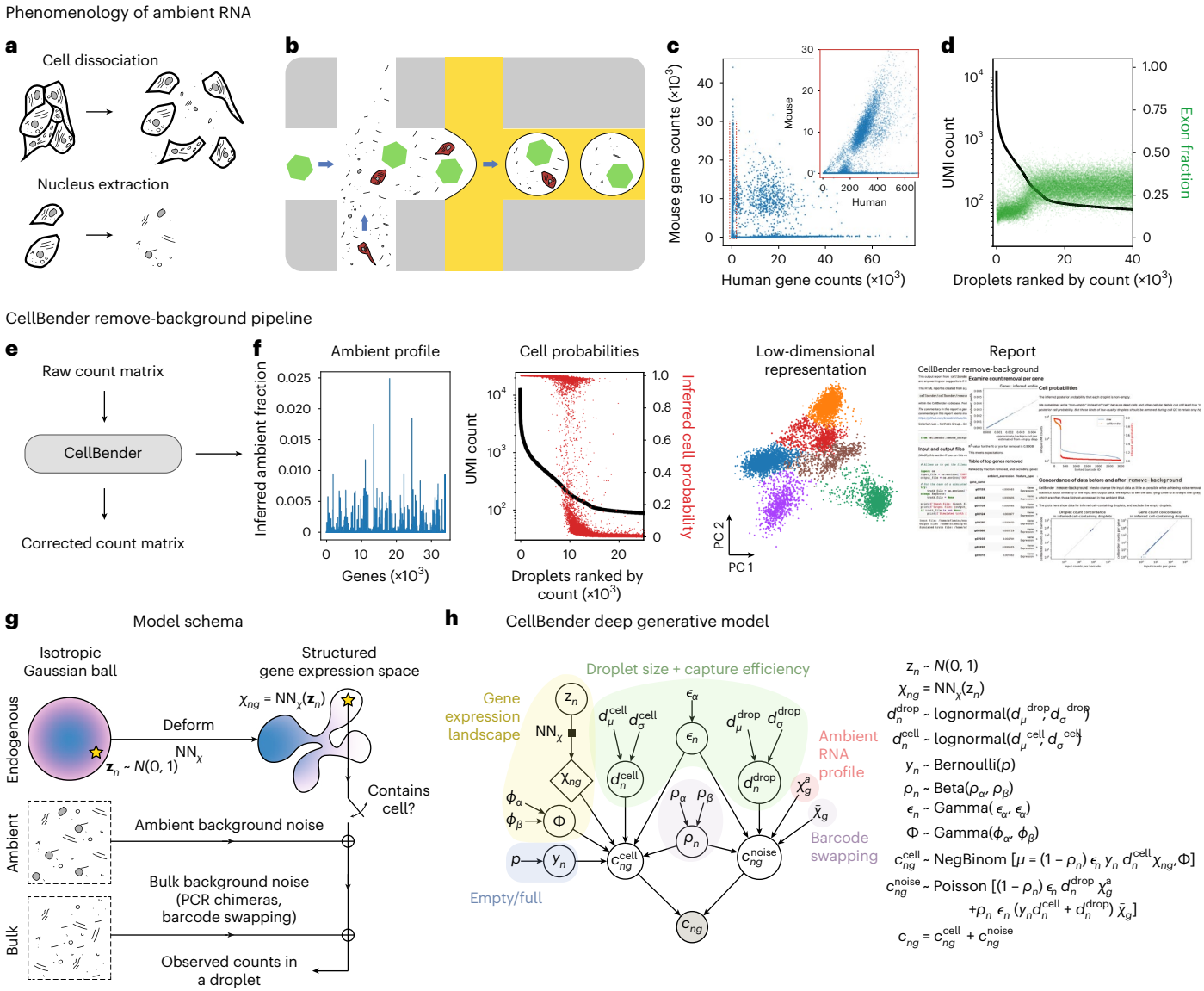


Fig. 1 | The phenomenology of ambient RNA and its deep generative modeling using CellBender remove-background. a, Cell dissociation and nucleus extraction lead to the presence of cell-free RNA in solution. **b**, Schematic diagram of the proposed source of ambient RNA background counts. Cell-free 'ambient' RNA (black lines) and other cellular debris are present in the cell-containing solution, and this RNA is packaged up into the same droplet as a cell (red) or into an otherwise empty droplet that contains only a barcoded capture oligonucleotide bead (green hexagon). **c**, Unique UMI counts per droplet that map to human and mouse genes for the publicly available hgmm12k dataset from 10x Genomics. The experiment is a mixture of human and mouse cells, and the inset (red box) shows that there are hundreds of human counts in droplets that contain mouse cells. **d**, The snRNA-seq Wistar rat heart dataset rat6k, showing unique UMI counts per droplet (black) with the fraction of reads from exonic regions superimposed (green). The 'ambient plateau' is the region of the rank-ordered plot with ranked barcode ID greater than about 15,000, where there are approximately 100 unique UMI counts per droplet. The increase in the fraction of exonic mapped reads coinciding with the onset of cell-free droplets shows that, in snRNA-seq, ambient RNA is enriched for cytoplasmic material, where fewer intronic reads remain due to splicing. **e**, Running CellBender is as simple as sending a raw count matrix in and receiving a corrected count matrix in return. **f**, Additional useful outputs include inferred latent variables of the model, such as the ambient RNA profile, probabilities that each droplet is not empty, a low-dimensional embedding of gene expression per cell and a summary report. PC, principal component. **g**, Schematic diagram explaining the rationale for our model. 'True' cell counts are modeled using a flexible prior parameterized by a neural network NN_χ. These counts (if a cell is present in a given droplet) are added to two constant noise sources: ambient background noise and bulk background noise. **h**, The generative model for count data in the presence of background RNA, where circles represent latent random variables, the diamond represents a deterministic computation, and the filled circle c_{ng} represents observed counts.

model?' section. We review key elements of the probabilistic model in this section and refer the reader to Methods for details.

Our starting point is the observed feature count matrix c_{ng}, where n and g denote cell index and feature index (for example, gene), respectively. We interpret c_{ng} as the sum of two non-negative contributions: the true biological counts originating from cells c_{ng}^{cell} and the background noise counts c_{ng}^{noise}. The background noise counts are drawn from a Poisson distribution:

$$c_{ng}^{\mathrm{noise}} \sim \mathrm{Poisson}\left[\underbrace{(1-\rho_n)\,\epsilon_n\,d_n^{\mathrm{drop}}\,\chi_g^{\mathrm{a}}}_{\text{ambient noise rate}} + \underbrace{\rho_n\,\epsilon_n\,(y_n d_n^{\mathrm{cell}} + d_n^{\mathrm{drop}})\,\bar{\chi}_g}_{\text{barcode swapping}}\right], \quad (1)$$

where the noise rate stems from two distinct processes: physically encapsulated ambient molecules and barcode-swapped molecules, for example, PCR chimeras. The ambient rate is determined by a learnable ambient profile χ_g^a, the droplet size factor d_n^{drop} and the droplet-specific capture efficiency factor ϵ_n. We model barcode swapping as a diffusion process with a droplet-specific rate ρ_n that is additionally modulated by the total amount of physically captured molecules in the droplet, that is, $\epsilon_n (y_n d_n^{cell} + d_n^{drop})$, and the dataset-wide average gene expression ('pseudo-bulk') $\bar{\chi}_g$. Here, $y_n \in \{0, 1\}$ is a binary variable that indicates cell presence in the droplet, and d_n^{cell} is the cell size factor.

The true biological counts c_{ng}^{cell} are modeled as a negative binomial (NegBinom) distribution with a rate that depends on droplet-specific capture efficiency ϵ_n, the non-chimeric fraction $1 - \rho_n$, the cell-presence indicator y_n, the cell size factor d_n^{cell} and a prior on true gene expression rate of the cell, $\chi_{ng}[\mathbf{z}_n]$:

$$c_{ng}^{cell} \mid \mathbf{z}_n \sim \text{NegBinom}\left[(1-\rho_n)\,\epsilon_n\, y_n\, d_n^{cell}\, \chi_{ng}[\mathbf{z}_n], \Phi\right]. \quad (2)$$

Here, Φ is a global learnable overdispersion parameter that modulates the uncertainty of the cell gene expression prior, and $\mathbf{z}_n$ is a droplet-specific latent variable that determines the gene expression rate prior χ_{ng}. Crucially, the way in which we construct this prior is one of the components that makes our model unique among noise-removal approaches for count data. We use a neural network to learn a flexible prior for biological counts, which is realized as a deformation of a low-dimensional Gaussian latent space, $\mathbf{z}_n$ (Fig. 1g). We fit the model using the stochastic variational inference (SVI) technique and leverage additional 'encoding' neural networks for amortizing the approximate inference of droplet-specific ('local') latent variables (Extended Data Fig. 1b). Put together, our framework resembles a variational auto-encoder[34] within a structured probabilistic model of noisy single-cell data.

We use the probabilistic programming language Pyro[16] to implement our model and the approximate variational inference algorithm. Our choice of variational posterior is shown graphically in Extended Data Fig. 1b, and details are provided in the Inference section.

Constructing a denoised integer count matrix

CellBender generates several outputs following model fitting and inference, including the learned profile of ambient noise, cell containment probability per droplet, the low-dimensional latent space representation of cell states and, importantly, the estimated denoised integer count matrix $\hat{c}_{ng}^{cell}$. It is worth emphasizing that our sought-after denoised count matrix $\hat{c}_{ng}^{cell}$ is not obtained by decoding the underlying low-dimensional latent embeddings of observed counts. This is a fundamental difference between our approach and variational auto-encoder-based denoising and imputation methods[35–37]: encoding into and out of a low-dimensional latent space acts as an information bottleneck, smooths the data to varying degrees and potentially masks subtle biological features such as transcriptional bursting, infrequent cell states and other rare fluctuations of potential functional importance. In our approach, the low-dimensional latent space of cell states acts as a prior, which, together with the observed data, determines the Bayesian posterior $p(c_{ng}^{noise} \mid \{c_{ng}\})$. We estimate an integer matrix of likely noise counts, $\hat{c}_{ng}^{noise}$, from the latter and obtain the denoised counts by subtracting off noise counts from observed counts.

Given the explicit partitioning of the observed data as a sum of non-negative signal and noise contributions, our approach explicitly guarantees the following: (1) each entry in the output count matrix will be less than or equal to the corresponding entry in the raw input matrix c_{ng}; (2) the results are largely insensitive to the representational capacity of the encoding and decoding neural networks; (3) importantly, in a clean dataset in which $\hat{c}_{ng}^{noise} \to 0$, we obtain $\hat{c}_{ng}^{cell} \to c_{ng}$, that is, the data are not deformed, smoothed or imputed. Our conservative approach to denoising is crucial for safe operation of our method in automated analysis pipelines, in particular, in application to clinical data and reference atlas-building efforts.

Any noise-removal algorithm involves a tradeoff between removing actual noise (sensitivity) and retaining signal (specificity). In CellBender, we control this tradeoff by means of a user-defined 'nominal false positive rate' (nFPR) parameter ('Estimating the integer noise matrix as a multiple-choice knapsack problem'). The nFPR parameter provides a transparent and interpretable handle to impose an upper bound on the amount of erroneously removed signal counts in aggregate ('false positive' counts), which could be either imposed separately on each feature or globally. Larger nFPR values imply removing more noise at the expense of more signal. The ability to control denoising nFPR, regardless of the inherent noise of a given dataset, is desirable for integrative analysis of heterogeneous datasets such as from clinical patient samples generated at multiple centers[26].

Finally, we note that reducing the posterior distribution of noise counts, $p(c_{ng}^{noise} \mid \{c_{ng}\})$, which is the natural output of a Bayesian model, to an integer point estimate, $\hat{c}_{ng}^{noise}$, is a non-trivial and subtle task. The widely used maximum a posteriori (MAP) estimator $\hat{c}_{ng}^{noise} = \text{argmax}\, p(c_{ng}^{noise} \mid \{c_{ng}\})$, even though it is a canonical Bayesian choice, leads to systematic underestimation of noise counts for genes that are present in the ambient profile at low levels ('On the asymptotic bias of canonical Bayes estimators'). Meeting the specified total noise target implied by nFPR while attaining the maximum model-based posterior probability turns the estimation of $\hat{c}_{ng}^{noise}$ into a secondary optimization problem. We discuss and evaluate several such estimation algorithms in Extended Data Figs. 9 and 10 and the accompanying Methods section 'Constructing the denoised integer count matrix: preliminaries'. By default (as of CellBender version 0.3.0_rc), we use a constrained estimator that is formally equivalent to the multiple-choice knapsack problem (MCKP), which we show is exactly solvable using a fast and greedy coordinate-ascent algorithm under mild assumptions ('A fast and exact MCKP solver for strictly log–concave posterior distributions').

Increased marker specificity and lower off-target expression

Removal of systematic noise from a dataset results in clearer biological insights by enhancing the specificity of gene expression and reducing spurious off-target counts. We demonstrate this by preprocessing scRNA-seq and snRNA-seq datasets with CellBender before downstream analysis and assessing the biological soundness of the results.

We carried out a standard analysis workflow on the publicly available peripheral blood mononuclear cell (PBMC) scRNA-seq dataset (pbmc8k) from 10x Genomics using SCANPY[17]. We identified cell-containing droplets as having posterior cell probability $q_n > 0.5$, and we used these cells in analyzing raw data and data preprocessed with CellBender. We further filtered cells using cutoffs for the number of nonzero genes, percent mitochondrial counts and an upper limit for UMI counts ('Single-cell analysis workflow and cell quality-control details'). The results of the exact same analysis, with and without CellBender preprocessing, are shown in Fig. 2a–d, including the expression of several immune marker genes.

Raw gene expression data, as shown in Fig. 2b, indicate that the genes *S100A8*, *S100A9*, *LYZ*, *CST3* and *PTPRC* are found to be abundantly and ubiquitously expressed in all clusters. While CD45 (encoded by *PTPRC*) is a glycoprotein expressed on all nucleated hematopoietic cells, *LYZ* and *CST3* are known to be specific markers for monocytes and plasmacytoid dendritic cells (pDCs), whereas *S100A8* and *S100A9* are known to be specific markers of neutrophils, monocytes and pDCs[38,39] (Supplementary Fig. 2). We hypothesized that the off-target expression of these genes was a result of systematic background noise. Figure 2d shows the denoised counts obtained using CellBender at nFPR = 0.01 and demonstrates both sensitivity and specificity of CellBender: on the one hand, we observe that the expression of *S100A8*, *S100A9* and *LYZ* is now largely concentrated on monocytes and pDCs, as expected.

https://doi.org/10.1038/s41592-023-01943-7

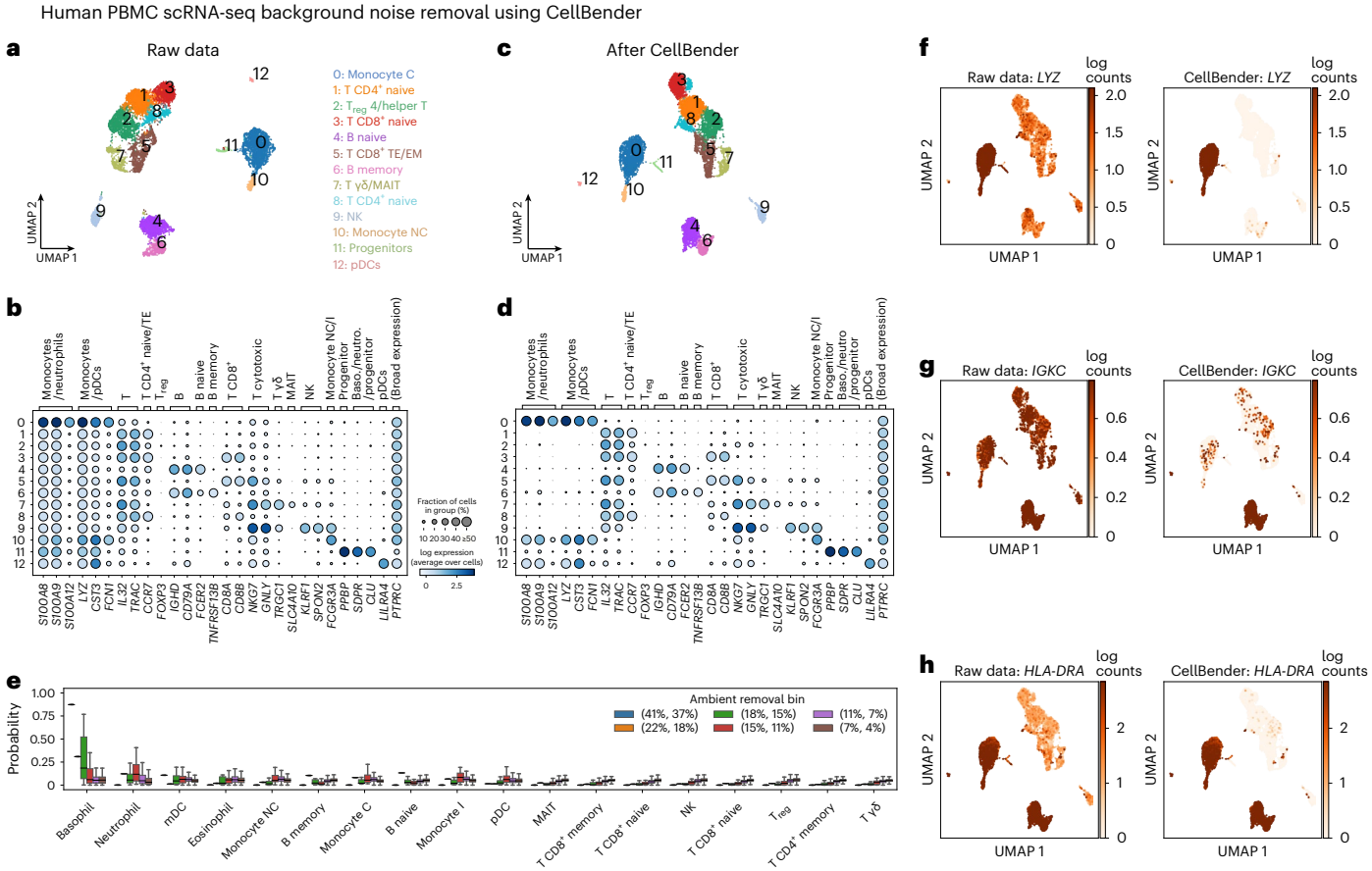


Fig. 2 | Evaluation of CellBender on a PBMC dataset, showing a standard SCANPY analysis of the publicly available 10x Genomics dataset pbmc8k with and without CellBender. a,**c**, UMAP visualizations of the raw data (**a**) and the data preprocessed with CellBender (**c**). Baso., basophil; neutro., neutrophil; NK, natural killer cell; T_{reg}, regulatory T cell; MAIT, mucosal-associated invariant T cell; monocyte C, classical monocyte; monocyte NC, non-classical monocyte. **b**,**d**, The dot plots display the expression of pre-defined marker genes for PBMCs for the raw dataset (**b**) and the dataset processed with CellBender (**d**). Monocyte NC/I, non-classical/intermediate monocyte; SDPR is an alias for CAVIN2 gene. **e**, Removal of each gene has been mapped to cell type, indicating that cell types do not necessarily contribute equally to ambient RNA. All genes expressed at TPM ≥ 10 (N = 6,678) have been assigned to one of six bins, and the bins linearly span the full range of the estimated fraction of ambient contamination. The ambient removal bins contain N = 4,467 (7–4%), N = 2,171 (11–7%), N = 30 (15–11%), N = 8 (18–15%), N = 1 (22–18%) and N = 1 (41–37%) genes, respectively. The box plot for each bin and cell type indicates the interquartile range of the probability that the assigned genes are stemming from that cell type. The center line denotes the median, and the whiskers correspond to 1.5 times the interquartile range. mDC, myeloid dendritic cell. **f**–**h**, UMAP plots of the expression of *LYZ*, *IGKC* and *HLA-DRA* in each cell before and after CellBender. Colored bar axes are truncated at the 80th percentile of per-cell expression.

Conversely, we note that the biologically expected ubiquitous expression of *PTPRC* has remained unchanged. Extended Data Fig. 2 shows expression of *LYZ* across clusters before and after CellBender in more detail. Supplementary Table 1 shows the differential expression of *S100A8*, *S100A9*, *LYZ* and *CST3* between monocytes C (cluster 0, where expression is expected) and naive B cells (cluster 4, where expression is not expected), calculated as the $\log_2$ (fold change) (LFC) using a Wilcoxon test. The LFCs increase by a factor of two after subtracting background RNA; by contrast, the LFC of *PTPRC* hardly changes at all. Another visualization of the effect of background noise removal is shown in Fig. 2f–h, in which expression of *LYZ* (increased specificity for monocytes), *IGKC* (increased specificity for B cells) and *HLA-DRA* (increased specificity for both monocytes and B cells) are plotted per cell before and after CellBender. The increase in specificity is striking for these and many other examples (for example, Extended Data Fig. 3).

Next, we explored the origin of background counts in the PBMC dataset. Note that neutrophils and other granulocytes are absent from 10x PBMC cell clusters. The difficulty of capturing granulocytes is attributed to their sensitivity to rapid degradation after collection and poor isolation via density gradient centrifugation[40]. As such, we hypothesized that ambient counts might be enriched with granulocyte lysates. To test this hypothesis, we examined the fraction of counts removed by CellBender for each gene and accordingly assigned each gene to the blood cell type with the highest consensus normalized transcripts per million (TPM) expression value obtained from the Human Protein Atlas immune reference[39]. We binned the genes according to the fraction removed by CellBender as ambient noise and interpreted the empirical frequency of assigning different cell types to the genes within each bin as the probability of contributing to the ambient soup. The result is shown in Fig. 2e and indicates that genes in the top-most ambient removal bins are associated with granulocytes at a substantially higher frequency. The top and bottom ten genes ranked by CellBender ambient removal are shown in Supplementary Fig. 1 along with the expression of each in the Human Protein Atlas immune reference, further demonstrating the enrichment of top-most ambient genes in basophils and neutrophils and the relative cell type non-specificity of the bottom-most genes.

PBMC scRNA-seq datasets are considered relatively clean in terms of ambient RNA contamination (see the UMI curve in Extended Data Fig. 4a,b as compared with the one in Extended Data Fig. 4e,f). Next, we

https://doi.org/10.1038/s41592-023-01943-7

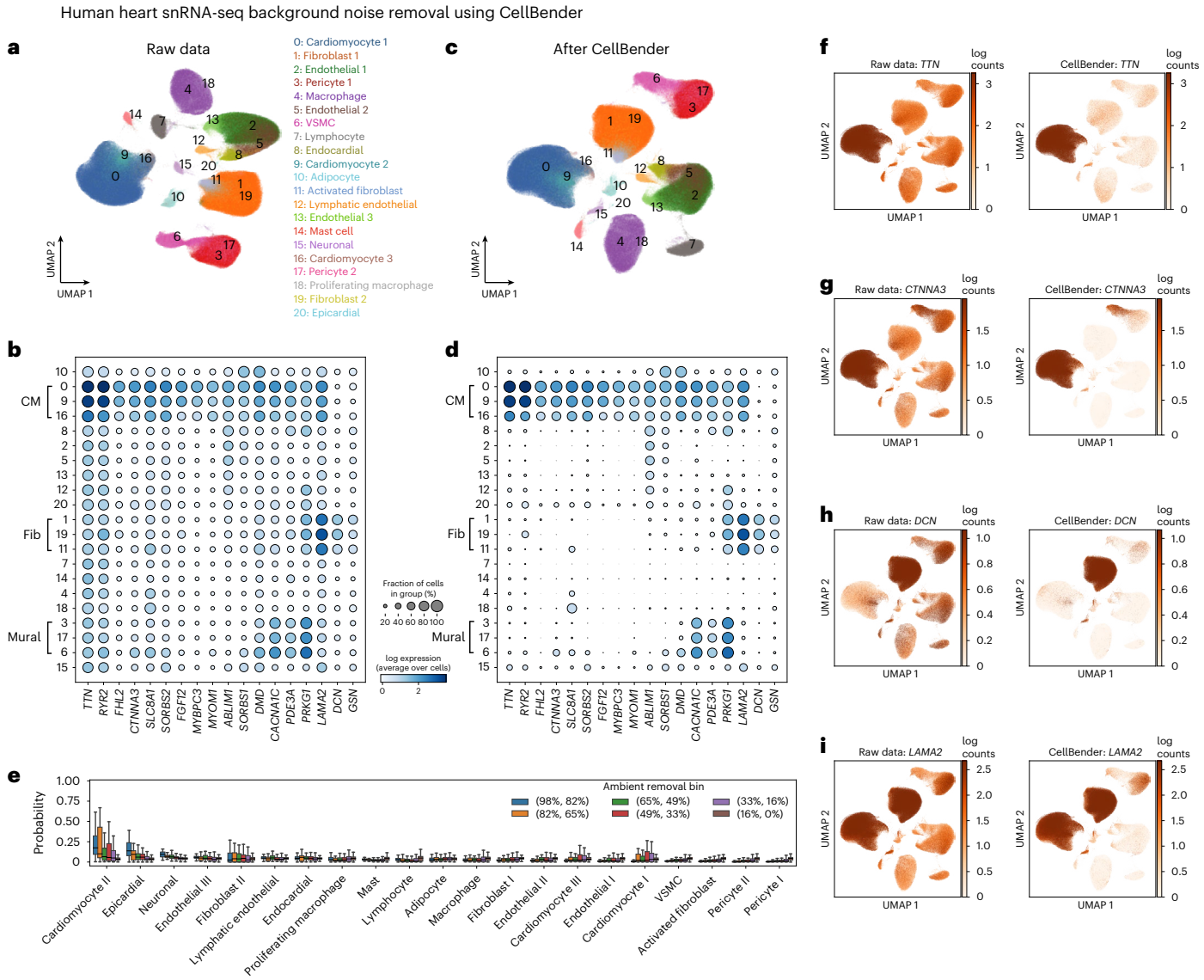


Fig. 3 | Removal of background RNA from a published human heart snRNA-seq atlas, heart600k, using CellBender. a, UMAP of raw data for nearly 600,000 nuclei. **b**, Dot plot showing several highly expressed genes in the raw dataset. CM, cardiomyocyte; fib, fibroblast; mural, mural cells including VSMCs and pericytes. **c**,**d**, UMAP and dot plot after CellBender. **e**, Similar to Fig. 2e, this plot shows that many of the removed counts are attributable to cardiomyocyte genes. All genes expressed at TPM ≥ 10 (N = 10,451) have been assigned to one of six linearly spaced bins according to the estimated fraction of ambient contamination. The ambient removal bins contain N = 9,939 (16–0%), N = 254 (33–16%), N = 88 (49–33%), N = 48 (65–49%), N = 48 (82–65%) and N = 74 (98–82%) genes, respectively. The box plots are defined as described in Fig. 2e. **f**–**i**, UMAP plots of the expression of *TTN*, *CTNNA3*, *DCN* and *LAMA2* before and after CellBender. Colored bar axes are truncated at the 80th percentile of per-cell expression.

examined a more challenging snRNA-seq dataset in which nuclei were extracted from frozen human heart tissue[23], heart600k. Nuclear preparations are more susceptible to ambient RNA contamination because the cells are all lysed and cytoplasmic mRNA becomes free in solution.

Uniform manifold approximation and projections (UMAPs) of the heart600k dataset were recomputed using Harmony-pytorch for batch effect correction[41], starting with either the raw counts (Fig. 3a) or the post-CellBender counts (Fig. 3c). The overall shape and appearance of the UMAP is qualitatively quite similar in both cases. However, an examination of gene expression shows that the dataset has been cleaned up quite notably after CellBender (Fig. 3b,d). Figure 3b shows that, for many highly expressed marker genes, the raw data would indicate that these genes are expressed in every cell type. However, it has been well established that the role of *TTN*, for example, is in the sarcomere of striated muscle cells including cardiomyocytes, and it is not expressed in the other cell types present in this experiment. Figure 3d,f show that, after CellBender, the expression of *TTN* becomes much more specific to the cardiomyocyte clusters. Similarly, *CTNNA3*, the product of which is involved in cell–cell adhesion in muscle, appears much more specific to cardiomyocytes and vascular smooth muscle cells (VSMC, cluster 6) after CellBender (Fig. 3d,g), in agreement with existing heart snRNA-seq atlases[21,42]. The expression of *DCN*, the product of which plays a role in collagen fibril assembly in the extracellular matrix, becomes much more specific to fibroblasts (Fig. 3d,h), also consistent with refs. 21,42. Finally, the expression of *LAMA2*, another component of the extracellular matrix, is found after CellBender to be much more specific to fibroblasts and cardiomyocytes, with some lower-level expression in pericytes, adipocytes and neuronal cells, again in agreement with refs. 21,42.

Cardiomyocytes have higher UMI counts than other cell types (for example, Supplementary Fig. 2b from ref. 21, where the cardiomyocytes can have an order of magnitude higher UMI counts than

other cell types in snRNA-seq). We hypothesized that we should see a disproportionately high amount of cardiomyocyte genes in the background RNA removed by CellBender. An examination of genes preferentially removed by CellBender shows that the top genes in terms of the removed fraction are in fact associated mainly with cardiomyocytes and, to a lesser extent, with epicardial cells (Fig. 3e). Many of the genes plotted in Fig. 3b,d are cardiomyocyte marker genes, including some of the most highly expressed genes in the dataset, *TTN* and *RYR2*. This highlights the importance of learning the ambient RNA profile from the dataset itself: the large amount of ambient cardiomyocyte mRNA, which is packaged into each droplet as background counts, is appropriately targeted and removed by CellBender, vastly improving the specificity of gene expression for downstream biological analyses.

Accurate identification of cell-containing droplets

As a part of model training and inference, CellBender produces a posterior probability, q_n, that droplet n contains a cell. While this determination can be rather trivial in some pristine datasets (for example, the PBMC dataset pbmc8k; Extended Data Fig. 4a,b), complicated experimental factors and excessive amounts of ambient RNA contamination often make this determination rather challenging (for example, the snRNA-seq dataset rat6k in Extended Data Fig. 4e,f). A variety of heuristics are typically employed to determine cutoffs for thresholding cells versus empty droplets, as in CellRanger version 2. More principled approaches have been developed, including CellRanger version 3+, EmptyDrops[43] and dropkick[44]. CellRanger version 3+ and EmptyDrops use statistical tests to ascertain which droplets have expression profiles significantly different from those of empty droplets, while dropkick uses a regularized logistic regression model. In our algorithm, the determination of empty versus non-empty droplets is a result of disentangling background counts from endogenous feature counts during model training, in which both gene expression and total UMI counts of all droplets are taken into account.

Figure 1f (middle left) shows the posterior cell probabilities for the first 25,000 droplets of the rat6k rat heart snRNA-seq dataset. Note that the algorithm in general identifies cells and empty droplets as expected and that the transition between the two is not based on a hard UMI cutoff. A determination of cell-free versus cell-containing droplets can be obtained by thresholding based on the posterior probability q_n. The algorithm converges to largely binary probability values for the majority of droplets, and the precise choice of threshold value affects relatively very few droplets in practice.

We compare the cell calls made by CellBender with three other methods in common use (CellRanger version 3, EmptyDrops and dropkick) in Fig. 4. Figure 4a shows that CellBender generally calls more cells than CellRanger (Supplementary Section 2.), many of which lie farther down the UMI curve (black) and are not called by other methods.

The set of cells called by CellBender contains all cells called by CellRanger version 3, EmptyDrops and dropkick after the same cell quality-control procedure was applied uniformly for all methods (Venn diagram in Fig. 4b; see Supplementary Section 2.8 for details on the quality-control procedure). In addition, CellBender detects more than 24% extra cells compared to dropkick, 50% extra cells compared to CellRanger version 3 and more than five times as many cells as EmptyDrops. Given the notable ambient RNA contamination in this dataset, we naturally hypothesized that many of the extra cell calls made by CellBender might have been cytoplasmic debris that were nevertheless statistically different from the ambient RNA in terms of gene expression makeup. To evaluate this hypothesis, we obtained a UMAP embedding of cells detected only by CellBender together with the cells detected by other methods (Fig. 4b) after typical filtering for gene complexity and mitochondrial fraction ('Single-cell analysis workflow and cell quality-control details'). To our surprise, (1) over 25% of the cells called exclusively by CellBender passed quality-control filters, amounting to over 500 cells (Supplementary Table 2) and (2) the extra cell calls made by CellBender clustered together with cells called by the other algorithms. Figure 4c shows the UMAP embedding obtained from the union of all cells called by any algorithm (after cell quality-control filtering) with putative cell type labels, and it can be seen that the cells called exclusively by CellBender have a marker gene distribution (Fig. 4e) similar to the dot plot created using the union of all cells called by any algorithm (Fig. 4d). EmptyDrops calls many low-UMI-count cells that CellRanger version 3 misses, although it also misses a large number of relatively high-UMI-count droplets along the rank-ordered UMI plot. This is likely due to the similarity between gene expression of the empty drops and the most populous cell types in this particular experiment (Supplementary Section 2.8). As such, the Dirichlet–multinomial likelihood model employed in EmptyDrops does not yield a statistically significant probability of being non-empty for cardiomyocyte-containing droplets. By contrast, CellBender learns the expression profile of cardiomyocytes from high-count droplets and is not impacted.

Finally, we recommend performing additional biologically motivated and tissue-specific quality control on CellBender cell calls whenever possible, for example, using mitochondrial read fraction, exonic read fraction and gene complexity, as suggested by previous authors[43,45]. We have deliberately avoided including such filters in CellBender to allow broad applicability of this method. Post-CellBender quality-controlling strategies must be informed by the studied biological system and the protocol. To emphasize the importance of post-filtering, we show a plot of the fraction of reads per droplet that come from mitochondrial genes in the hgmm12k dataset in Supplementary Fig. 5. It can be clearly seen that many low-UMI droplets exhibit a high fraction of mitochondrial genes (possibly dead or dying cells), and, because they are distinct from empty droplets, they are nevertheless assigned a high probability of containing cells by CellBender. After filtering the detected cells based on mitochondrial read fraction, some of these lowest-count and degraded cells will be naturally filtered out. The analysis shown in Fig. 4 includes such post-filtering criteria.

Reduced off-target gene counts in mixed-species experiments

A definitive and straightforward experimental benchmark to evaluate the level of background noise and the efficacy of mitigation strategies is a mixed-species experiment, in which two cell types from different species are combined and assayed together. This would ideally result in droplets containing exclusively feature counts from one species or the other, but, due to the presence of background noise, this is not the case (as shown in Fig. 1c). Here, we use the publicly available human–mouse mixture dataset from 10x Genomics (hgmm12k) to evaluate CellBender and also compare CellBender to DecontX[33], another method for removing background noise.

Figure 5a shows a scatterplot of human and mouse gene expression in each droplet in raw data and for CellBender-processed data at different nFPR settings on a logarithmic scale (data plotted on linear axes are shown in Supplementary Fig. 6). Doublet droplets are omitted from the plot, as they do not serve as validation. The raw data show hundreds of off-target cross-species counts in each droplet (best visible in the side histograms). After removing background noise, we would ideally expect all cross-species counts to be removed. Indeed, CellBender (with a default nFPR of 0.01) reduces off-target counts to a median of 19 per cell, that is, by over an order of magnitude from the raw data, with a median of 225. At an nFPR setting of 0.1, the median off-target counts per cell drops to 4 (Supplementary Table 4 and Supplementary Fig. 7). It is worth re-emphasizing that CellBender is a completely unsupervised model and that the algorithm achieves this level of denoising without the knowledge of human genes or mouse genes or that this is a mixture-species experiment.

Figure 5b compares the performance of CellBender with that of DecontX[33]. Validation is carried out on the set of cells called by both CellBender and EmptyDrops, which was the cell caller used as part of

https://doi.org/10.1038/s41592-023-01943-7

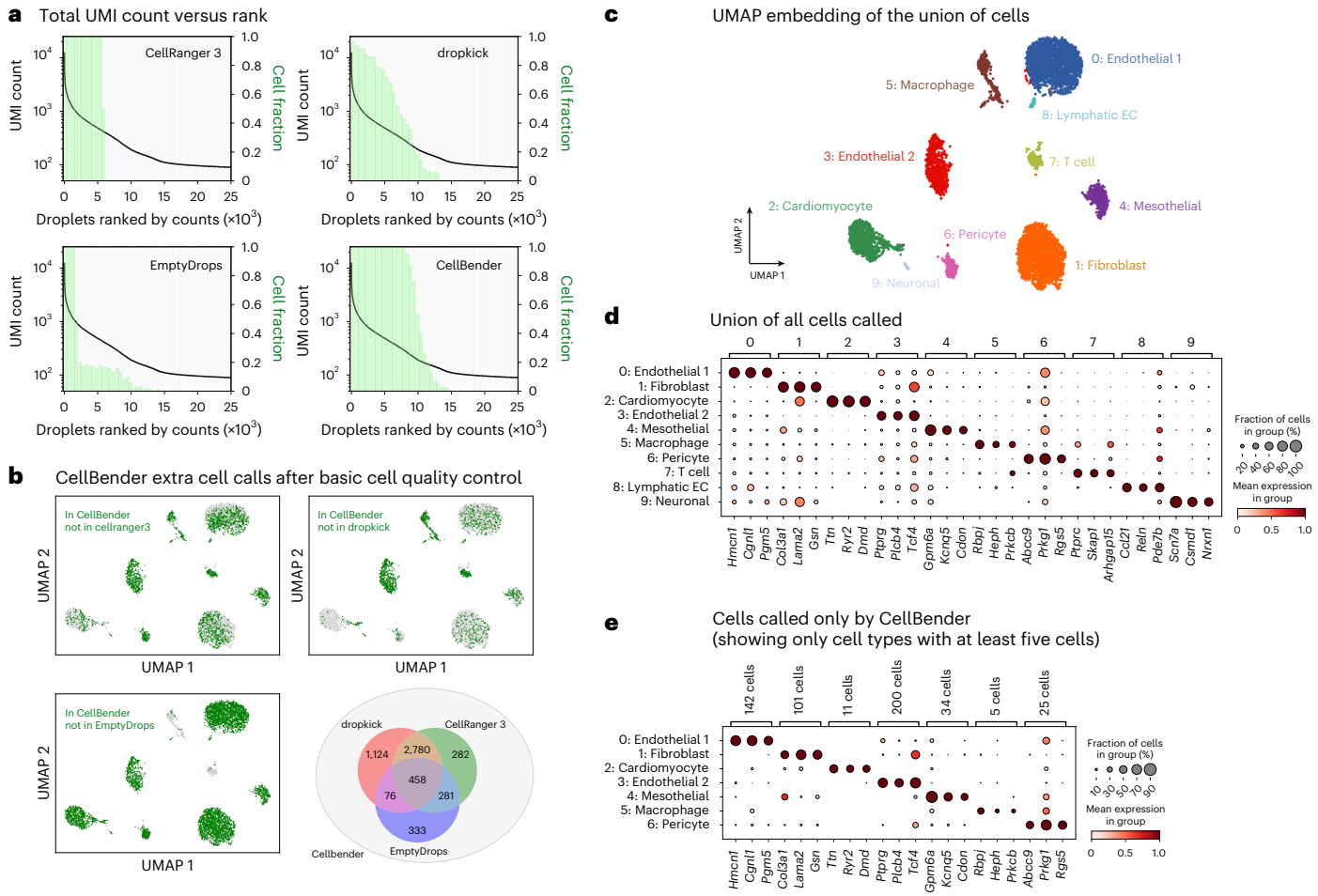


Fig. 4 | Comparing four cell-calling algorithms (CellRanger version 3, dropkick, EmptyDrops and CellBender) on the rat6k snRNA-seq dataset. **a**, Detected cells for different algorithms: the UMI-versus-barcode rank curve (black line) is superimposed on the fraction of detected cell-containing droplets in different barcode rank bins (green bars). CellRanger results indicate imposing a nearly hard cutoff on the barcode rank, while EmptyDrops calls several cells between 6,000 and 10,000 in UMI-count rank (*x* axis). **b**, CellBender detects all cells called by the other algorithms (after cell quality control) and many more. UMAP embeddings were generated after performing cell quality control. All cells are shown in gray, with green dots superimposed to denote cells that were not detected by the method in question but that were detected by CellBender. The Venn diagram quantifies the agreement between various methods. **c**, UMAP with cell type labels at a Leiden resolution of 0.5. All clusters appear to be biologically meaningful. **d**, The top three marker genes for each cluster (SCANPY Wilcoxon test) are shown for the union of all cells called by any algorithm (which coincides with CellBender cell calls). EC, endothelial cell. **e**, Same marker gene dot plot as in **d** but now showing only those cells that were exclusively detected by CellBender. The similarity to **d** and the presence of real marker genes indicates that the extra cell calls made by CellBender are real.

the DecontX pipeline. We found that, while DecontX removes a large number of cross-species counts, CellBender has a substantially higher sensitivity: in fact, at an nFPR of 0.1 (in red), CellBender removes all cross-species counts from 16% of cells (see the marginal histograms in Fig. 5a, where '1' means that there are zero cross-species counts). In addition, the results obtained using CellBender show other important characteristics that are worth emphasizing:

- The amount of background noise that gets removed can be tuned using the interpretable expected nFPR parameter, as shown in Fig. 5a,e.
- CellBender largely removes the linear trend in the relationship between cross-species counts and cell-endogenous counts (the linear trend seen in raw data shown in gray; see also Supplementary Fig. 6). The proportional relation between background noise counts and cell-endogenous counts has been associated with library PCR chimeras formed during mixed-template amplification[13], which effectively leads to random barcode swapping between library fragments. Another potential mechanism is droplet-to-droplet variability in capture efficiency, which also leads to a proportional relation between endogenous and noise counts. Both of these phenomena are modeled in CellBender (Model). Note that this linear trend remains largely unmitigated by DecontX (Fig. 5b and Supplementary Fig. 6b).
- We find that DecontX treats different groups of cells from the same species differently, which can be seen as the fragmentation of blue points in Fig. 5b. We hypothesize that this non-uniform performance is associated with the hard clustering preprocessing step in DecontX. While the user can provide their own clustering to DecontX to mitigate this issue, CellBender sidesteps such issues altogether by avoiding hard clustering entirely and instead allows similar cells to share statistical power via a low-dimensional continuous latent space.

Near-optimal performance on simulated datasets

Thus far, we have shown evaluations of CellBender using real datasets and resorted to prior biological knowledge (for example, marker genes) or expected outcomes (as in mixed-species experiments) to assess

https://doi.org/10.1038/s41592-023-01943-7

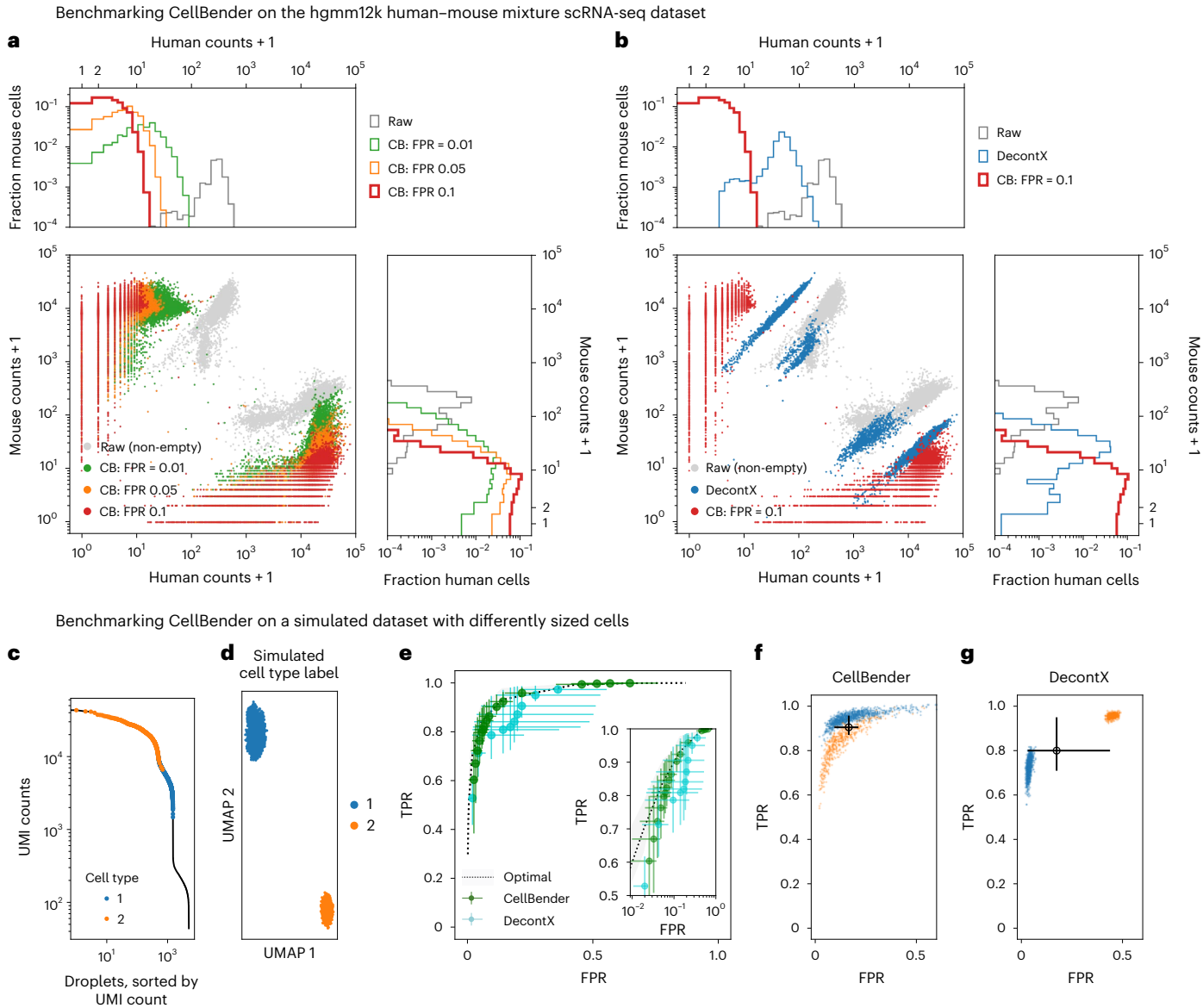


Fig. 5 | Benchmarking CellBender on denoising the hgmm12k human–mouse mixture dataset and a simulated dataset with differently sized cells. The hgmm12 human–mouse dataset (**a**,**b**) and the simulated dataset (**c**–**g**) are shown. **a**, Logarithmic-scale plot of species mixing shows that raw data (gray) contain several hundred counts of mouse transcripts in human cells and vice versa. CellBender removes most of the off-target noise. The marginal histograms show that many human cells end up with zero mouse counts and vice versa. CellBender-denoised counts are shown for several nFPR choices. **b**, Same plot as in a but with DecontX included for comparison. **c**, The UMI curve for the simulated dataset, showing cells and empty droplets. Simulated cell type 2 has many more UMI counts than cell type 1. **d**, The UMAP created from cells called by CellBender. **e**, ROC curve quantifying per-cell noise-removal performance. Black dotted line with gray shading (1 s.d. in per-cell performance) represents the best possible performance given perfect knowledge of all latent variables in the simulation and is only limited by sampling noise. Large green dots (mean) show CellBender outputs at a variety of expected nFPR values. Cyan dots show DecontX output using different values of the parameter Δ. **f**,**g**, Comparison of per-cell performance of DecontX (default settings) and CellBender (matching the output FPR of default DecontX), in which cells are colored by cell type. DecontX treats the different cell types rather differently in terms of FPR (blue and orange colors are cell types from **c**,**d**. CellBender is abbreviated as CB in the plots. The error bars in **e**–**g** show the interquartile range in per-cell performance over $N = 1{,}500$ simulated cells.

the soundness of the results. Here, we additionally show experiments using simulated data, with known noise and signal contributions, to evaluate the performance of CellBender theoretically and in a more controlled setting. Figure 5c–g shows the results of inference using a simulated dataset with 10,000 genes, generated according to a noise model that includes both ambient sources and barcode swapping (see 'Simulated data generation' for simulation details). Importantly, the CellBender model is slightly mis-specified for this simulated data on purpose, as the simulation draws 'true' gene expression χ_{ng} from a Dirichlet distribution with a fixed concentration parameter per cell type. Figure 5c–g shows a simulation with two 'cell types' with unique underlying expression profiles, where the cell types have a very different number of UMI counts. The ambient profile in the simulation is a weighted average of total expression.

Figure 5e shows the noise-removal performance as a receiver operating characteristic (ROC) curve. Noise counts that are correctly removed are counted as 'true positives', and a 'false positive' is a cell-endogenous count that is erroneously removed. A hypothetical model with perfect knowledge of every real and noise count would be represented by the point (0, 1) in the false positive rate (FPR)–true

positive rate (TPR) plane. The stochasticity of the data-generating process and finite sequencing depth, however, make this perfect limit theoretically out of reach, even with perfect knowledge of all latent variables.

We show the 'best theoretically achievable performance', given perfect knowledge of all latent variables, as the black dotted line. CellBender comes quite close to this optimal performance (green dots, obtained by running at increasing nFPR parameters). Supplementary Table 5 shows a decent agreement between the specified nFPR and the empirical FPR. The DecontX ROC curve was created by running the tool with several values of the hyperparameter Δ. Default DecontX parameters were found to correspond to an empirical FPR of 0.142 and a TPR of 0.809. Run with nFPR = 0.0442, CellBender was found to have exactly the same TPR of 0.809, but the FPR was 0.062. This means that, for the same amount of noise removal, DecontX removed more than twice as much signal as CellBender. At nFPR = 0.125, CellBender matched the DecontX FPR of 0.142, but the TPR was 0.923. This means that, for the same value of removal of real signal, CellBender was able to remove 92.3% of the noise, while DecontX removed 80.9%. This seems to be due to DecontX treating the two simulated cell types differently in terms of where they land on the ROC curve (Fig. 5f,g).

Denoised antibody counts show increased correlation with RNA

As mentioned in the introduction, CellBender makes no assumption about the nature of the captured molecules and is generally applicable to all barcoded features used within the same model. This generality results from the common phenomenological origin of the technical noise that we aim to remove. To demonstrate this, we evaluated CellBender for denoising CITE-seq data. We treated cell surface protein and RNA measurements on an equal footing as a unified count matrix and denoised the two modalities simultaneously using CellBender. Empirically, antibody counts exhibit a very high level of background noise, which may be attributed to unbound and unwashed antibodies in the cell suspension. We show a publicly available 10x Genomics CITE-seq dataset of PBMCs (pbmc5k) in Fig. 6. We have grouped antibodies together with their associated genes for ease of visual evaluation. In Fig. 6a, the antibody features (red) have such a large amount of background noise that it is challenging to discern a clear pattern. Gene expression counts (blue), by contrast, have a very low amount of background noise in this dataset. Figure 6b shows the output of CellBender run with an nFPR of 0.1, where a pattern clearly emerges, and visually it appears that the red dots (protein antibody) very often line up with the blue dots (mRNA).

Antibody counts and the corresponding RNA counts exhibit an expected linear relationship for most antibodies, and the impact of CellBender on this relationship is shown in Fig. 6e. In the raw data, the presence of background noise leads to a relatively large nonzero intercept, such that cells with zero RNA counts have nonzero antibody counts. CellBender effectively reduces the magnitude of this intercept while maintaining the biological linear relationship; additional results are given in Supplementary Fig. 9a,b. The specificity of antibodies for particular cell types improves as a direct consequence. Supplementary Fig. 9c shows that the Pearson correlation between the fraction of cells per cluster with nonzero counts of antibody and nonzero counts of the corresponding RNA increases markedly after CellBender. We note that the presence of large intercepts poses a challenge for comparing cell types across different batches and datasets, which may have different levels of background counts.

As a specific case study, we highlight two antibodies for different isoforms of CD45 (encoded by *PTPRC*): CD45RA and CD45RO, shown with the corresponding mRNA *PTPRC*. The removal of background noise (Fig. 6c) highlights a clear pattern of mutually exclusive differential expression of the two isoforms in different immune cell types: compare Fig. 6d, top (raw) and bottom (CellBender). The expression of *HNRNPLL*, encoding a splicing factor associated with the CD45RO isoform[46], is shown in Supplementary Fig. 8. We find that effector T cell states, that is, T $CD8^+$ effector memory (EM)/terminal effector (TE) and regulatory T cells, have both relatively higher levels of *HNRNPLL* and CD45RO expression, as expected. CellBender increases the relative enrichment of CD45RO in such clusters as shown in Fig. 6d.

Discussion

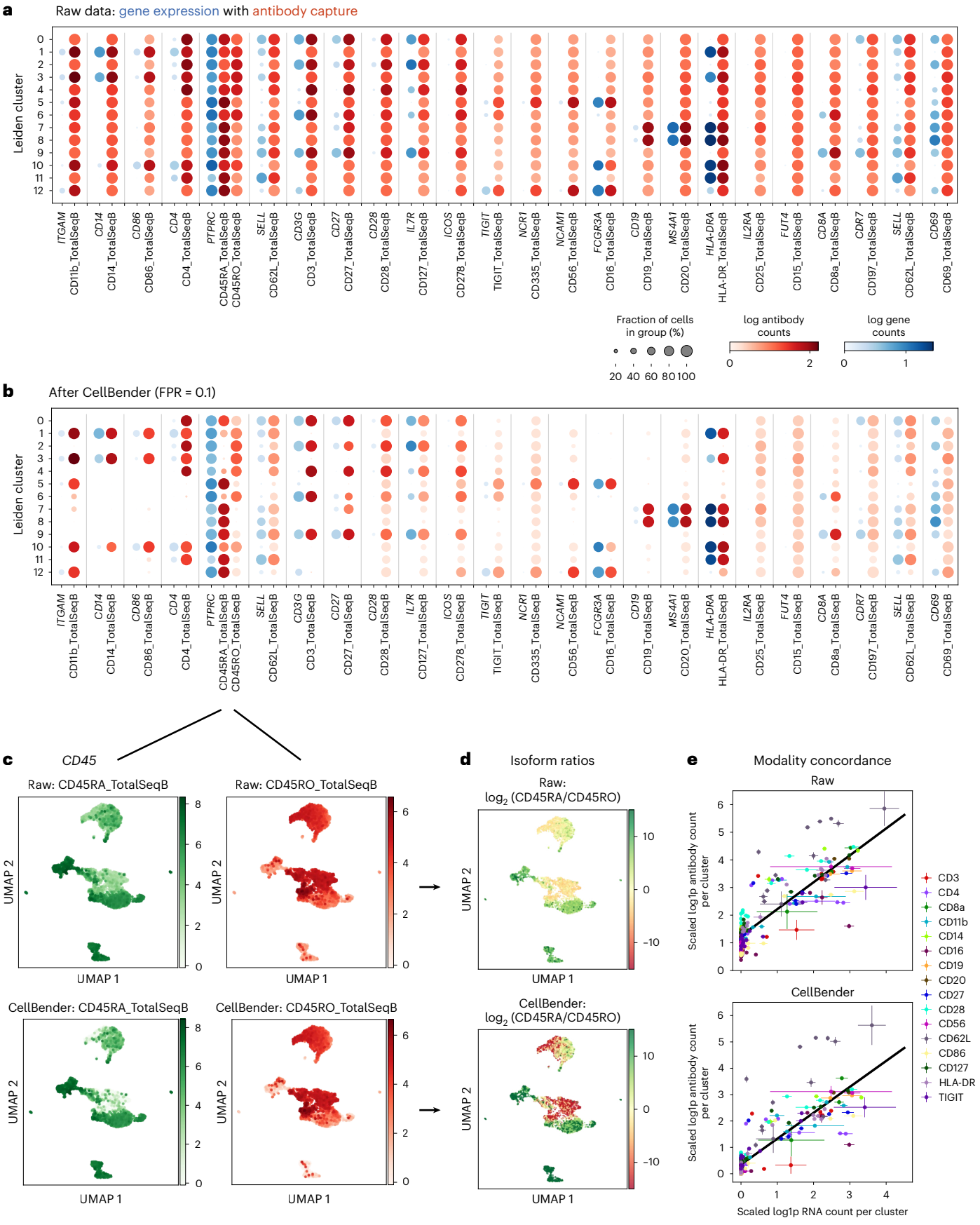
We present CellBender, an unsupervised method for removing systematic background noise from droplet-based single-cell experiments. CellBender learns the profile of noise counts from the data and subsequently estimates denoised counts. This is achieved by leveraging a deep generative model of noisy single-cell data that combines the flexibility of deep neural networks for learning the landscape of cell states with a structured probabilistic model of noise-generation processes. CellBender can be used as a preprocessing step in any droplet-based single-cell omic analysis pipeline that involves an unfiltered count matrix. No preprocessing is needed before running CellBender, and the presence of droplets containing more than one cell (doublets and multiplets) does not degrade the performance of CellBender (Supplementary Section 2.2 and Extended Data Fig. 5). CellBender is especially helpful for analyzing datasets severely contaminated with background noise. These include snRNA-seq experiments that are subject to harsh nuclear isolation protocols and CITE-seq experiments that may produce large amounts of ambient antibodies. Removal of ambient noise has been advocated as an important step in single-cell analysis workflows and protocols[47,48] and is increasingly becoming a standard part of single-cell data analysis.

Other authors have addressed the removal of background noise in scRNA-seq datasets in the past few years, including with DecontX[33] and SoupX[11] for removal of ambient RNA and methods for attenuating background counts due to chimeric molecules[13]. In practice, the operation of SoupX involves manual input and relies on the user's prior knowledge of cell type-specific gene expression as well as providing (or calculating) a list of genes for estimating the background RNA fraction in cells. The method introduced in ref. 13 leverages read-per-UMI frequency data to detect library PCR chimeras. While this approach is highly effective at reducing the number of chimeric counts, it cannot detect physically encapsulated ambient molecules, which are indistinguishable from cell-endogenous molecules based on read-per-UMI frequency data alone. DecontX represents an unsupervised alternative

Fig. 6 | Performance of CellBender on denoising a CITE-seq PBMC dataset from 10x Genomics (pbmc5k). a, Raw data. The dot plot includes antibody capture features (red), along with the relevant gene expression features (blue) for all measured antibodies with a corresponding gene that had maximum expression in any cell type above 0.05 mean counts. Groupings of related features are delineated by the gray vertical lines. **b**, Same as **a** but for CellBender-denoised counts. In both **a** and **b**, the clustering is obtained at a Leiden resolution of 0.6 based on the CellBender output; see Supplementary Fig. 8 for UMAP and cluster labels. **c**, Examining CD45RA and CD45RO isoforms of CD45 as log normalized counts superimposed on the UMAP embedding. The expected anti-correlation of the two isoforms is substantially enhanced by CellBender. **d**, UMAP embedding showing the log ratio of CD45RA and CD45RO expression and indicating the increased specificity afforded by CellBender. **e**, Comparing the relationship between antibody counts and gene expression after scaling to collapse all data to the same line ('pbmc5k CITE-seq dataset quality control and normalization') for the raw data (top) and CellBender-denoised data (bottom). By removing background counts, CellBender moves the intercept down toward zero and makes antibody counts more specific to clusters. The horizontal and vertical error bars indicate s.e.m. of scaled log1p RNA and antibody counts, respectively, for each of the 13 cell clusters. The numbers of cells for each cluster are given in the caption of Supplementary Fig. 8.

https://doi.org/10.1038/s41592-023-01943-7

for background noise removal. We have demonstrated that CellBender operates near the theoretically optimal limit and surpasses the performance of DecontX on several benchmarks. Other practical advantages of CellBender over DecontX include a tunable nFPR parameter for controlling the tradeoff between denoising sensitivity and specificity in a principled fashion, automatic probabilistic determination of



cell-containing droplets and generation of a low-dimensional latent space embedding of cells that can be used in downstream analyses.

While CellBender is particularly well suited for cleaning up and extracting the biological signal from noisy datasets, the presence of excessive noise may prevent CellBender from converging to a near-optimal solution. In particular, if the UMI counts in empty droplets are not at least an order of magnitude less than the UMI counts in cells, the underlying signal–noise deconvolution problem and identification of cell-containing droplets will be ill posed. Such edge cases, however, might properly be considered quality-control failures from the outset. Non-convergence of CellBender can be diagnosed by inspecting the called cells and empty droplets to ensure that they align with expectations based on the experimental design and the UMI curve as well as by inspecting the trajectory of the loss function during training to ensure smooth convergence to a stable value. As with any non-convex optimization problem, it is good practice to check the results whenever possible, in this case, using prior biological expectations, orthogonal measurements (for example, in situ hybridization) and tissue-specific domain knowledge. We would also like to reiterate the importance of performing an additional cell quality-control step after CellBender. Although CellBender can accurately identify empty droplets, the 'non-empty' droplets are not all high-quality cells suitable for downstream analysis, and cell quality control should be performed to remove dead or dying cells and debris using a variety of droplet quality-control metrics as appropriate for the experiment. Finally, choosing an extreme target nFPR value, while potentially being useful for certain applications, is likely to result in a denoised count matrix that lacks sensitivity. Therefore, we do not recommend choosing nFPR values larger than 0.1 in routine applications.

It is also important to point out that, for CellBender to achieve a near-optimal solution to the denoising problem, the CellBender model must be well specified, that is, appropriate for the noise in the dataset at hand. While the datasets shown above were all obtained using the 10x Genomics single-cell gene expression assay, CellBender is suitable for use with a variety of droplet-based and well-based single-cell assays, some examples of which are shown in Supplementary Section 2.4 (Extended Data Figs. 6 and 7), and the CellBender model is formulated to generalize to any droplet-based or well-based single-cell or single-nucleus technology. The only requirement of the tool is that there should be some examples of 'empty' droplets or wells for CellBender to learn the 'ambient' or cell-free feature profile. Beyond that, any assay that generates a single-cell or single-nucleus count matrix and for which CellBender's noise model is applicable should be valid as an input.

Removing systematic noise from individual datasets before integration is becoming increasingly crucial as the field is progressing from homogeneous small-scale experiments toward large-scale data integration and atlasing efforts, where datasets from many batches and tissue-processing centers are being combined and analyzed jointly (for example, ref. 32). By mitigating background noise, CellBender eliminates a source of batch variation and spurious differential expression signals. This is particularly important for performing differential analysis of similar cell types between samples in a cohort. Because the systematic background noise is specific to the dataset and is influenced by the circumstances around each batch, unmitigated noise can then appear as differential signal across batches. Supplementary Section 2.3 includes a clear demonstration of this phenomenon and shows how CellBender effectively mitigates this source of batch variation and spurious differential expression (Extended Data Fig. 8). In spite of the role that CellBender plays in mitigating sample-specific background noise, we would like to emphasize that the 'batch effect' in single-cell datasets is a more complex phenomenon, and removing other sources of batch variation (including variation in gene capture efficiency, sequencing depth, protocol differences and so on) and performing single-cell data integration are outside the scope of this work.

Although ambient RNA is typically considered a nuisance, the analysis accompanying Figs. 2e and 3e demonstrated that studying the ambient profile produced by CellBender might be of value in and of itself and could be used, for instance, to study the transcriptional makeup of extracellular vesicles and to diagnose degraded and uncaptured cells. Ziegler et al., for example, made use of the CellBender-inferred ambient profile to help call high-confidence SARS-CoV-2 RNA$^+$ cells in an scRNA-seq study of human nasopharyngeal swabs[28].

Field applications of CellBender, which include aiding the discovery of new biology and resolving inconsistent findings, can be found in the works of other authors who have adopted our method since the time it was made publicly available as open-source software in 2019. We would like to highlight ref. 49, in which CellBender was applied to remove ambient RNA from brain snRNA-seq samples, resulting in the removal of neuronal marker genes from glial cell types and identification of previous annotations of immature oligodendrocytes as potentially glial cells contaminated with ambient RNA. For particularly compelling example figures demonstrating the effects of CellBender, see Supplementary Fig. 1 in ref. 32 and Extended Data Fig. 1e–h in ref. 26. In cases in which the raw data are relatively clean to begin with, Di Bella et al. observe that processing with CellBender will (appropriately) change the count matrix very little[50].

Future research directions include extending CellBender beyond the count matrix of unique UMIs and modeling the data at the finer granularity of individual sequenced reads. For instance, chimeric reads can be identified much more effectively when read-per-UMI counts are taken into account[13]. This information is not contained in the conventional primary quantification of single-cell data as a count matrix of unique UMI counts. Additional interesting directions include evaluating the utility of CellBender on additional single-cell data modalities, including Perturb-seq[6] for which background CRISPR guides can make the determination of perturbation challenging.

Online content

Any methods, additional references, Nature Portfolio reporting summaries, source data, extended data, supplementary information, acknowledgements, peer review information; details of author contributions and competing interests; and statements of data and code availability are available at https://doi.org/10.1038/s41592-023-01943-7.

References

1. Macosko, E. Z. et al. Highly parallel genome-wide expression profiling of individual cells using nanoliter droplets. *Cell* **161**, 1202–1214 (2015).
2. Zheng, G. X. Y. et al. Massively parallel digital transcriptional profiling of single cells. *Nat. Commun.* **8**, 14049 (2017).
3. Butler, A., Hoffman, P., Smibert, P., Papalexi, E. & Satija, R. Integrating single-cell transcriptomic data across different conditions, technologies, and species. *Nat. Biotechnol.* **36**, 411–420 (2018).
4. Stuart, T. & Satija, R. Integrative single-cell analysis. *Nat. Rev. Genet.* **20**, 257–272 (2019).
5. Stoeckius, M. et al. Simultaneous epitope and transcriptome measurement in single cells. *Nat. Methods* **14**, 865–868 (2017).
6. Dixit, A. et al. Perturb-seq: dissecting molecular circuits with scalable single-cell RNA profiling of pooled genetic screens. *Cell* **167**, 1853–1866 (2016).
7. Liu, L. et al. Deconvolution of single-cell multi-omics layers reveals regulatory heterogeneity. *Nat. Commun.* **10**, 470 (2019).
8. Chen, S., Lake, B. B. & Zhang, K. High-throughput sequencing of the transcriptome and chromatin accessibility in the same cell. *Nat. Biotechnol.* **37**, 1452–1457 (2019).
9. Ma, S. et al. Chromatin potential identified by shared single-cell profiling of RNA and chromatin. *Cell* **183**, 1103–1116 (2020).

10. Mimitou, E. P. et al. Multiplexed detection of proteins, transcriptomes, clonotypes and CRISPR perturbations in single cells. *Nat. Methods* **16**, 409–412 (2019).
11. Young, M. D. & Behjati, S. SoupX removes ambient RNA contamination from droplet based single cell RNA sequencing data. *GigaScience* **9**, giaa151 (2020).
12. Haas, B. J. et al. Chimeric 16S rRNA sequence formation and detection in Sanger and 454-pyrosequenced PCR amplicons. *Genome Res.* **21**, 494–504 (2011).
13. Dixit, A. Correcting chimeric crosstalk in single cell RNA-seq experiments. Preprint at *bioRxiv* https://doi.org/10.1101/093237 (2016).
14. Thompson, J. R., Marcelino, L. A. & Polz, M. F. Heteroduplexes in mixed-template amplifications: formation, consequence and elimination by 'reconditioning PCR'. *Nucleic Acids Res.* **30**, 2083–2088 (2002).
15. Perkel, J. M. et al. Single-cell analysis enters the multiomics age. *Nature* **595**, 614–616 (2021).
16. Bingham, E. et al. Pyro: deep universal probabilistic programming. *J. Mach. Learn. Res.* **20**, 1–6 (2019).
17. Wolf, F. A., Angerer, P. & Theis, F. J. SCANPY: large-scale single-cell gene expression data analysis. *Genome Biol.* **19**, 15 (2018).
18. Dani, N. et al. A cellular and spatial map of the choroid plexus across brain ventricles and ages. *Cell* **184**, 3056–3074 (2021).
19. Popova, G. Human microglia states are conserved across experimental models and regulate neural stem cell responses in chimeric organoids. *Cell Stem Cell* **28**, 2153-2166 (2021).
20. Holloway, E. M. et al. Mapping development of the human intestinal niche at single-cell resolution. *Cell Stem Cell* **28**, 568–580 (2021).
21. Tucker, N. R. et al. Transcriptional and cellular diversity of the human heart. *Circulation* **142**, 466–482 (2020).
22. Tucker, N. R. et al. Myocyte specific upregulation of *ACE2* in cardiovascular disease: implications for SARS-CoV-2 mediated myocarditis. *Circulation* **142**, 708–710 (2020).
23. Chaffin, M. et al. Single-nucleus profiling of human dilated and hypertrophic cardiomyopathy. *Nature* **608**, 174–180 (2022).
24. Sun, W. et al. snRNA-seq reveals a subpopulation of adipocytes that regulates thermogenesis. *Nature* **587**, 98–102 (2020).
25. Dong, H. et al. Identification of a regulatory pathway inhibiting adipogenesis via RSPO2. *Nat. Metab.* **4**, 90–105 (2022).
26. Delorey, T. M. et al. COVID-19 tissue atlases reveal SARS-CoV-2 pathology and cellular targets. *Nature* **595**, 107–113 (2021).
27. Xu, G. et al. The differential immune responses to COVID-19 in peripheral and lung revealed by single-cell RNA sequencing. *Cell Discov.* **6**, 73 (2020).
28. Ziegler, C. G. K. et al. Impaired local intrinsic immunity to SARS-CoV-2 infection in severe COVID-19. *Cell* **184**, 4713–4733 (2021).
29. Melms, J. C. et al. A molecular single-cell lung atlas of lethal COVID-19. *Nature* **595**, 114–119 (2021).
30. Wang, S. et al. A single-cell transcriptomic landscape of the lungs of patients with COVID-19. *Nat. Cell Biol.* **23**, 1314–1328 (2021).
31. Zazhytska, M. et al. Non-cell-autonomous disruption of nuclear architecture as a potential cause of COVID-19-induced anosmia. *Cell* **185**, 1052–1064 (2022).
32. Eraslan, G. et al. Single-nucleus cross-tissue molecular reference maps toward understanding disease gene function. *Science* **376**, eabl4290 (2022).
33. Yang, S. et al. Decontamination of ambient RNA in single-cell RNA-seq with DecontX. *Genome Biol.* **21**, 57 (2020).
34. Kingma, D. P. & Welling, M. Auto-encoding variational Bayes. Preprint at https://doi.org/10.48550/arXiv.1312.6114 (2014).
35. Lopez, R., Regier, J., Cole, M. B., Jordan, M. & Yosef, N. Deep generative modeling for single-cell transcriptomics. *Nat. Methods* **15**, 1053–1058 (2018).
36. Grønbech, C. H. et al. scVAE: variational auto-encoders for single-cell gene expression data. *Bioinformatics* **36**, 4415–4422 (2020).
37. Eraslan, G., Simon, L. M., Mircea, M., Mueller, N. S. & Theis, F. J. Single-cell RNA-seq denoising using a deep count autoencoder. *Nat. Commun.* **10**, 390 (2019).
38. Monaco, G. et al. RNA-seq signatures normalized by mRNA abundance allow absolute deconvolution of human immune cell types. *Cell Rep.* **26**, 1627–1640 (2019).
39. Uhlen, M. et al. A genome-wide transcriptomic analysis of protein-coding genes in human blood cells. *Science* **366**, eaax9198 (2019).
40. *Neutrophil Analysis in 10x Genomics Single Cell Gene Expression Assays* Report No. CG000444 (10x Genomics, 2021).
41. Korsunsky, I. et al. Fast, sensitive and accurate integration of single-cell data with Harmony. *Nat. Methods* **16**, 1289–1296 (2019).
42. Litviňuková, M. et al. Cells of the adult human heart. *Nature* **588**, 466–472 (2020).
43. Lun, A. et al. EmptyDrops: distinguishing cells from empty droplets in droplet-based single-cell RNA sequencing data. *Genome Biol.* **20**, 63 (2019).
44. Heiser, C. N., Wang, V. M., Chen, B., Hughey, J. J. & Lau, K. S. Automated quality control and cell identification of droplet-based single-cell data using dropkick. *Genome Res.* **31**, 1742–1752 (2021).
45. Petukhov, V. et al. dropEst: pipeline for accurate estimation of molecular counts in droplet-based single-cell RNA-seq experiments. *Genome Biol.* **19**, 78 (2018).
46. Oberdoerffer, S. et al. Regulation of CD45 alternative splicing by heterogeneous ribonucleoprotein, hnRNPLL. *Science* **321**, 686–691 (2008).
47. Luecken, M. D. & Theis, F. J. Current best practices in single cell RNA seq analysis: a tutorial. *Mol. Syst. Biol.* **15**, e8746 (2019).
48. Clarke, Z. A. et al. Tutorial: guidelines for annotating single-cell transcriptomic maps using automated and manual methods. *Nat. Protoc.* **16**, 2749–2764 (2021).
49. Caglayan, E., Liu, Y. & Konopka, G. Neuronal ambient RNA contamination causes misinterpreted and masked cell types in brain single-nuclei datasets. *Neuron* **110**, 4043–4056 (2022).
50. Di Bella, D. J. et al. Molecular logic of cellular diversification in the mouse cerebral cortex. *Nature* **595**, 554–559 (2021).

Methods

Why a deep generative model?

Before we take a deeper dive into the CellBender model and inference algorithm, we would like to clearly motivate our choice of modeling framework. The approach taken here, that is, deep generative models and SVI, typically requires more computational resources than conventional deterministic algorithms and, thus, must be conceptually justified.

First, we note that, because the ambient molecules are aliquoted from the same cell suspension, they correspond to the same fixed distribution, and our many observations of cell-free droplets provide sufficient statistics to make it possible to infer that distribution with very high accuracy, in principle. In challenging cases such as highly contaminated snRNA-seq experiments in which background noise removal is most needed, cell-free droplets are defined only in relation to cell-containing droplets ('Accurate identification of cell-containing droplets'). Therefore, we are obligated to model the landscape of cell feature counts (mRNA, protein and so on) on par with the fixed distribution of ambient molecules. Cell states, however, are typically much more variable than the fixed distribution of ambient molecules. The challenging issue is our lack of a priori knowledge of the process that generates true biological counts in a cell and the a priori unknown biological complexity of the assayed sample.

Furthermore, the fraction of captured mRNA and other targeted features is on the order of 10% or less of expected counts (using 10x Genomics version 2 or 3 chemistry, which generates approximately tens of thousands of feature counts per cell). Such sparse sampling is referred to as 'dropout' in the context of droplet-based cell assays. For our purposes, dropout poses a particularly difficult challenge: even if we are provided with the knowledge of the true distribution of ambient molecules and other systematic background noises, 'deconvolving' the observed count data from any given droplet into noise and signal contributions is a non-trivial task, given that both contributions are deep in the discrete regime and are subject to extreme sampling stochastic noise. We must, therefore, come up with a prior estimate of both contributions. An imbalanced model, for example, one that has a stronger prior for noise and a weaker prior for signal or vice versa will lead to overestimation or underestimation of noise.

For these two main reasons, that is, (1) an a priori unknown landscape of cell states and (2) sparse sampling of the content of each droplet (dropout), we are naturally led to a modeling choice that includes the following ingredients: (1) a flexible class of distributions to learn the landscape of cell states, (2) the ability to allow cells to share statistical power and leverage the observation from all cells to act as a prior and (3) the ability to automatically determine whether or not a droplet contains a cell.

Grouping cells into clusters to share statistical weight may be achieved in multiple ways, including nearest-neighbor clustering (as in a traditional scRNA-seq analysis) and other graph-based methods[51]. Using information learned from similar cells to build a prior belief is most rigorously done within the Bayesian framework. Bayesian methods for modeling complex distributions include auto-encoders and normalizing flows. Finally, automatic determination of cell-free versus cell-containing droplets requires model comparison, which may also be rigorously done within the Bayesian framework. We have found the common denominator of these requirements, together with the expressibility of the Bayesian framework for turning mechanistic insights into structured probabilistic models, to naturally lead to a model that is no more or no less complex than CellBender.

Model

Our generative model for noisy droplet-based count data is shown in Extended Data Fig. 1a, along with a schematic of the rationale in Fig. 1g. Throughout this section, we use n and g subscripts to refer to cell and molecular feature (for example, gene, protein) indices on various vector and matrix variables. In graphical models, latent random variables are represented as circles, while deterministic computations are represented by diamonds. Hyperparameters are denoted without circles, neural networks are denoted by factors (small black squares), and the observed counts are denoted by the filled gray circle, c_{ng}.

$\mathbf{z}_n \in \mathbb{R}^Z$ is the latent variable that encodes endogenous cell states in a lower-dimensional space. χ_{ng} is the fractional molecular feature frequency (that is, normalized to 1) in cell n and lives on a $(G-1)$ simplex in $\mathbb{R}^G$, where G is the dimensionality of the raw molecular feature space (for example, number of genes in scRNA-seq). NN_χ, shown as a factor (black square) in the graphical model, is the 'decoder' neural network that deforms the low-dimensional embedding $\mathbf{z}_n$ to the raw data feature space χ_{ng}. χ_g^{a} is the normalized abundance of ambient molecules and is a learnable parameter. d_n^{cell} is a cell-specific size factor. d_n^{drop} is a droplet-specific size factor for ambient counts. y_n is a discrete binary random variable that is 1 if there is a cell in droplet n and 0 otherwise. ρ_n is the proportion of reads that are assigned to droplet n but are exogenous to droplet n and have been randomly swapped, for example, due to PCR chimera formation. ϵ_n is a droplet-specific capture efficiency parameter, close to 1, that reflects how efficiently the targeted molecules in droplet n are captured, barcoded and reverse transcribed. In other words, ϵ_n is a technical confounder that affects the total UMI counts in a droplet, endogenous and ambient alike. c_{ng}^{cell} and c_{ng}^{noise} denote the latent counts per droplet that come from the cell and from background sources, respectively. Finally, c_{ng} is the observed counts of feature g in cell n. The generative process is as follows:

$$
\begin{aligned}
\mathbf{z}_n &\sim \mathcal{N}(0,1)\\
\chi_{ng} &= \mathrm{NN}_\chi(\mathbf{z}_n)\\
d_n^{\mathrm{drop}} &\sim \mathrm{lognormal}\left(d_\mu^{\mathrm{drop}}, d_\sigma^{\mathrm{drop}}\right)\\
d_n^{\mathrm{cell}} &\sim \mathrm{lognormal}\left(d_\mu^{\mathrm{cell}}, d_\sigma^{\mathrm{cell}}\right)\\
y_n &\sim \mathrm{Bernoulli}\,(p)\\
\rho_n &\sim \mathrm{Beta}\,(\rho_\alpha, \rho_\beta)\\
\epsilon_n &\sim \mathrm{Gamma}\,(\epsilon_\alpha, \epsilon_\beta)\\
\Phi &\sim \mathrm{Gamma}\,(\Phi_\alpha, \Phi_\beta)
\end{aligned}
\qquad (3)
$$

$$
c_{ng}^{\mathrm{cell}} \sim \mathrm{NegBinom}\left[\underbrace{(1-\rho_n)\,\epsilon_n\, y_n\, d_n^{\mathrm{cell}}\,\chi_{ng}}_{\mu_{ng}^{\mathrm{cell}}\ \mathrm{term}}, \Phi\right]
$$

$$
c_{ng}^{\mathrm{noise}} \sim \mathrm{Poisson}\left[\underbrace{(1-\rho_n)\,\epsilon_n\, d_n^{\mathrm{drop}}\,\chi_g^{\mathrm{a}} + \rho_n\,\epsilon_n\left(y_n\, d_n^{\mathrm{cell}} + d_n^{\mathrm{drop}}\right)\bar{\chi}_g}_{\lambda_{ng}^{\mathrm{noise}}\ \mathrm{term}}\right]
$$

$$
c_{ng} = c_{ng}^{\mathrm{cell}} + c_{ng}^{\mathrm{noise}}
$$

Modeling the rate of endogenous and exogenous feature counts. We will discuss our parametric choices for count likelihoods, that is, negative binomial for endogenous counts and Poisson for ambient counts, in the next section. Here, we focus on the expressions given for the 'rates' of the two contributions, μ_{ng}^{cell} and $\lambda_{ng}^{\mathrm{noise}}$, respectively. The rate of endogenous counts in a droplet μ_{ng}^{cell} straightforwardly follows from the definitions: $y_n\, d_n^{\mathrm{cell}}\,\chi_{ng}$ represents the expected counts from the cell in droplet n. The rate is modulated by the droplet's efficiency ϵ_n, and the term $(1-\rho_n)$ is the fraction of library fragments originating from the cell that are not swapped to a different droplet, maintaining the interpretation of ρ_n as the fraction of swapped counts exogenous to droplet n. The rate of exogenous counts in a droplet $\lambda_{ng}^{\mathrm{noise}}$ has two parts: ambient molecules and randomly swapped barcodes. The barcode-swapping process results in a certain fraction of counts in each droplet, $\rho_n \in [0,1]$, having actually originated in other droplets. We assume that it is equally likely to swap any two barcodes; therefore,

the net effect is that the swapped molecules in any given droplet are effectively sampled from the average ('bulk') features over the entire experiment, denoted by $\bar{\chi}_g$. Ambient molecules, on the other hand, may have a distinct composition as argued in Supplementary Section 1.1 and demonstrated in 'Increased marker specificity and lower off-target expression' and therefore are sampled from a different and learnable profile, denoted by χ_g^a. Accordingly, we decompose the rate into two main parts. The first part is the ambient counts that physically originate in droplet n: $(1-\rho_n)\,\epsilon_n\,d_n^{\text{drop}}\,\chi_g^a$. The second part is the counts that did not physically originate in droplet n but were erroneously assigned there later: $\rho_n\,\epsilon_n\left(y_n\,d_n^{\text{cell}}+d_n^{\text{drop}}\right)\bar{\chi}_g$. This expression is the product of three terms: the contamination fraction ρ_n, the term in parentheses together with ϵ_n that is proportional to the expected number of molecules physically encapsulated in the droplet and finally the average ('bulk') molecular profile $\bar{\chi}_g$.

Count likelihood models. The fundamental noise governing count data in single-cell sequencing is Poisson, rooted in the empirical fact that each molecule has only a small probability of being successfully captured and sequenced. We refer the reader to the excellent analysis of refs. 52,53 on this matter and the nuances and hazards of employing more flexible count likelihood models.

Accordingly, we model the noise statistics of background noise counts c_{ng}^{noise} as a Poisson distribution. We do not accommodate additional overdispersion in addition to what is implicitly induced by the stochasticity of the latent variables that appear in the Poisson rate of exogenous counts (equation (3)): we believe our theoretical model of ambient counts and barcode swapping to be flexible enough and to be a fairly faithful representation of the simple underlying physical process, such that any additional overdispersion is likely to result in model underspecification.

On the other hand, we purposefully endow endogenous counts c_{ng}^{cell} with extra overdispersion, signified by the overdispersion parameter Φ of a negative binomial (Poisson–gamma) distribution. In the context of our problem, this inclusion is motivated as follows: as mentioned earlier, imposing a prior distribution over c_{ng}^{cell} is meant to provide a mechanism to share statistical power across cells, help overcome data sparsity and ultimately aid deconvolving observed counts into exogenous and endogenous compartments. Crucially, the prior imposed on endogenous counts must be data driven and endowed with a tunable parameter to balance the model's prior belief over endogenous counts with exogenous counts, as dictated by the structure of the data and the maximum likelihood principle that we use to fit the model. The extra overdispersion parameter provides precisely such a mechanism to balance the prior beliefs and desensitize the results on the representational capacity of the underlying neural networks that encode the structure of endogenous counts. Faced with a dataset that contains a large number of the same cell types in the same state, the model will benefit from reducing Φ and strengthening its prior belief of endogenous counts. By contrast, prior belief will be commensurately 'softer' when faced with a complex dataset, in particular, if the size of the latent space is not large enough to afford the complexity of the dataset.

Model hyperparameters. d_μ^{cell}, d_σ^{cell}, d_μ^{drop} and d_σ^{drop} are all fixed hyperparameters that we determine automatically from the provided data using a number of heuristics. A cutoff in UMI counts (−low-count-threshold) is used to remove very-low-UMI-count barcodes. The mode of the remaining UMI-count distribution is then used to approximate d_μ^{drop}. A Gaussian mixture model is fit to the UMI counts per droplet, and mixture components larger than d_μ^{drop} are identified and combined to obtain an estimate of d_μ^{cell}. The variance hyperparameters are also estimated from the Gaussian mixture components and scaled down to account for the dispersion induced by ϵ_n. These hyperparameters specify the prior for endogenous and ambient rate scale factors, d_n^{cell} and d_n^{drop}, both of which are modeled as log normal distributions on an empirical basis. p is a hyperparameter representing the prior probability that any given droplet contains a cell, and it is derived from the expected number of cells in the experiment and the total number of droplets included in the analysis. $(\rho_\alpha, \rho_\beta)$ are general priors for the contamination fraction ρ_n, with default values of (1.5, 50), motivated by the fact that the shape of this beta distribution matches our expectations, from observations of many datasets, that barcode swapping is typically in the range of a few percent. The hyperparameter ϵ_α controls how concentrated the droplet-specific capture efficiency will be around 1. We use a fixed value of 50, motivated by examination of overdispersion of droplet sizes in the 10x Genomics ercc dataset, compared to a Poisson.

Choice of contamination model. The CellBender model can be restricted to only ambient background noise by setting $\rho_n = 0$ for all n, or it can be restricted to barcode-swapping background noise only by removing the 'endogenous ambient' term $(1-\rho_n)\,\epsilon_n\,d_n^{\text{drop}}\,\chi_g^a$ from the Poisson rate for c_{ng}^{noise}. The default mode in CellBender uses the full model as specified in equation (3), but the user can specify the ambient-only or swapping-only model via command-line arguments in our provided implementation.

Inference

The probabilistic model described in the previous section entails several global (experiment-wide) and local (one for each droplet) latent variables. Scalable approximate inference can be achieved using SVI[54] and amortization. We provide a brief account of the inference strategy in this section. We note that other authors have also successfully applied SVI techniques for scalable probabilistic modeling of single-cell data[35–37]. The objective function that is optimized in SVI is the evidence lower bound (ELBO):

$$\text{ELBO}(X\,|\,\theta,\varphi) \equiv \int \mathrm{d}Z\, q(Z\,|\,\varphi)\,\log\left(\frac{p(X,Z\,|\,\theta)}{q(Z\,|\,\varphi)}\right), \tag{4}$$

where $X = \{c_{ng}\}$ is the observed data, $\theta = \{\chi_g^a, W_\chi\}$ is the bundle of tunable model hyperparameters, including the weights of the neural network NN_χ (denoted by W_χ), $Z = \{\rho_n, y_n, d_n^{\text{cell}}, d_n^{\text{drop}}, \epsilon_n, \mathbf{z}_n, \Phi\}$ is the bundle of latent variables, and $q(Z\,|\,\varphi)$ is the variational ansatz shown in Extended Data Fig. 1b and parameterized by $\varphi = \{W_y, W_d, W_\epsilon, W_z, \hat{d}_\sigma^{\text{cell}}, \hat{d}_\mu^{\text{drop}}, \hat{d}_\sigma^{\text{drop}}, \hat{\rho}_\alpha, \hat{\rho}_\beta, \hat{\Phi}_\alpha, \hat{\Phi}_\beta\}$. In the SVI methodology, one obtains $\text{argmax}_{\theta,\varphi}\,\text{ELBO}(X\,|\,\theta,\varphi)$ via successive subsampling of data X and incremental updates of (θ, φ) using a stochastic optimizer. We refer the reader to ref. 55 for a review.

Constructing a variational posterior distribution. The faithfulness of the approximate posterior to the true posterior is ultimately dependent on one's choice of the variational ansatz $q(Z\,|\,\varphi)$. Extended Data Fig. 1b shows the structure of our proposed ansatz. Generally speaking, we impose tunable parametric distributions over global latent variables while we infer local latent variables using auxiliary neural networks (often referred to as recognition or encoder networks). The latter technique is referred to as amortization and is the key to the scalability of our algorithm to a theoretically unbounded number of data points (cells).

The posterior for $\mathbf{z}_n$ is encoded by a neural network NN_z, which takes in observed counts c_{ng}, along with the current estimate of the ambient profile χ_g^a, and outputs $(\mathbf{z}_{n;\mu}, \mathbf{z}_{n;\sigma})$; the latter parameterize the mean and scale of an assumed Gaussian posterior distribution for $\mathbf{z}_n$:

$$\mathbf{z}_n\,|\,c_{ng}, \chi_g^a \sim \mathcal{N}(\mathbf{z}_{n;\mu}, \mathbf{z}_{n;\sigma}). \tag{5}$$

Note that this encoder network for $\mathbf{z}_n$, together with the decoder network that maps $\mathbf{z}_n$ to χ_{ng}, form the auto-encoder structure mentioned earlier, in the spirit of ref. 34.

The variational posteriors for the cell-presence-indicator variable y_n, the cell scale factor d_n^{cell} and the droplet-specific capture efficiency ϵ_n are parameterized via additional neural networks (shown together as NN_{enc} in Extended Data Fig. 1). These auxiliary encoder neural networks each take c_{ng} and χ_g^{a} as input and estimate all or some of the parameters of specified posterior distributions. In practice, we found it beneficial to further provide a few handcrafted features constructed from c_{ng} and χ_g^{a} as inputs to each of the encoder neural networks ('Implementation details and technical remarks'). The posterior for y_n is assumed to be Bernoulli and is parameterized by the neural network NN_y that outputs q_n, the cell-presence posterior probability:

$$y_n \mid c_{ng}, \chi_g^{\text{a}} \sim \text{Bernoulli}\,(q_n). \tag{6}$$

The posterior for d_n^{cell} is assumed to be log normal and is parameterized by the neural network NN_d, which outputs $d_{n;\mu}^{\text{cell}}$, a strictly positive scale factor, per droplet:

$$d_n^{\text{cell}} \mid c_{ng}, \chi_g^{\text{a}} \sim \text{lognormal}\left(d_{n;\mu}^{\text{cell}}, \hat{d}_\sigma^{\text{cell}}\right). \tag{7}$$

We have additionally introduced a learnable posterior parameter, $\hat{d}_\sigma^{\text{cell}}$, to characterize the uncertainty in estimating cell scale factors. The posterior for ϵ_n is assumed to be Gamma-distributed and is parameterized by the neural network NN_ϵ, which outputs $\epsilon_{n;\mu}$, the posterior mean capture efficiency:

$$\epsilon_n \mid c_{ng}, \chi_g^{\text{a}} \sim \text{Gamma}\,(\epsilon_{n;\mu}\,\epsilon_\alpha, \epsilon_\alpha). \tag{8}$$

Here, ϵ_α is the same hyperparameter from the model, controlling the uncertainty in droplet efficiencies. Finally, the variational posteriors for Φ, ρ_n and d_n^{drop} are assumed as follows:

$$\begin{aligned}
\Phi &\sim \text{Gamma}\,(\hat{\Phi}_\alpha, \hat{\Phi}_\beta),\\
\rho_n &\sim \text{Beta}\,(\hat{\rho}_\alpha, \hat{\rho}_\beta),\\
d_n^{\text{drop}} &\sim \text{lognormal}\left(\hat{d}_\mu^{\text{drop}}, \hat{d}_\sigma^{\text{drop}}\right),
\end{aligned}$$

each of which involve two trainable parameters. Note that we have assumed that the barcode-swapping rate ρ_n and droplet size d_n^{drop} have the same posterior distribution for all droplets n, even though these are droplet-specific (local) latent variables. We have found this more restrictive posterior to work well in practice while allowing more robust SVI fits.

Approximate treatment of Poisson and negative binomial convolution. Details aside, the structure of our generative model for endogenous and exogenous counts is as follows (equation (3)):

$$\begin{aligned}
\mathbf{c}^{\text{cell}} &\sim \text{NegBinom}\,(\boldsymbol{\mu}, \alpha^{-1}),\\
\mathbf{c}^{\text{noise}} &\sim \text{Poisson}\,(\boldsymbol{\lambda}),\\
\mathbf{c} &= \mathbf{c}^{\text{cell}} + \mathbf{c}^{\text{noise}},
\end{aligned}$$

where we have dropped the common ng indices and used bold symbols as a shorthand for cell × feature matrices. Here, $\boldsymbol{\mu}$ and $\boldsymbol{\lambda}$ refer to the endogenous and exogenous count rates, and $\alpha = \Phi^{-1}$ is the inverse overdispersion. This parametric decomposition into non-negative endogenous and exogenous contributions ensures that the inferred endogenous counts $c^{\text{cell}} \mid \mathbf{c}$ are $\le \mathbf{c}$. This desirable property, however, poses a technical challenge: as a part of variational inference, we need to be able to compute the probability density of $\mathbf{c}$ in a differentiable fashion; however, the sum of a general Poisson and a general negative binomial distribution does not admit a closed probability density expression. Formally, the latter is given by the convolution of the two probability densities. Computing this convolution explicitly, while doable, is prohibitively slow. We therefore resort to the following approximation during model training:

$$p(\mathbf{c}|\boldsymbol{\mu}, \boldsymbol{\lambda}, \alpha) = \begin{cases} \text{NegBinom}\left(\mathbf{c} \mid \hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\alpha}}^{-1}\right), & \text{if } \boldsymbol{\mu} \ge \varepsilon\boldsymbol{\lambda} \\ \text{Poisson}\,(\mathbf{c} \mid \boldsymbol{\lambda}), & \text{otherwise} \end{cases} \tag{9}$$

where we set $\varepsilon = 10^{-5}$, and $\hat{\boldsymbol{\mu}}$ and $\hat{\boldsymbol{\alpha}}$ are obtained by matching the first two moments of an 'effective' negative binomial distribution to $c^{\text{cell}} + c^{\text{noise}}$:

$$\begin{aligned}
\hat{\boldsymbol{\mu}} &= \boldsymbol{\mu} + \boldsymbol{\lambda},\\
\hat{\boldsymbol{\alpha}} &= \alpha\left(\frac{\boldsymbol{\mu}+\boldsymbol{\lambda}}{\boldsymbol{\mu}}\right)^2,
\end{aligned} \tag{10}$$

where all algebraic operations involving matrices are element wise. The rationale for switching from a moment-matched negative binomial to Poisson when $\boldsymbol{\mu} < \varepsilon\boldsymbol{\lambda}$ is for numerical stability: when $\boldsymbol{\mu} \to 0$, that is, a vanishing prior rate of endogenous counts, we obtain $\hat{\boldsymbol{\alpha}} \to \infty$, which leads to numerical instability. At the same time, the observed count is dominated by noise counts in this regime, that is, $\mathbb{E}[\mathbf{c}^{\text{cell}}](\mathbb{E}[\mathbf{c}^{\text{noise}}])^{-1} = \boldsymbol{\mu}\boldsymbol{\lambda}^{-1} < \varepsilon = 10^{-5}$, justifying the switch.

Constructing the denoised integer count matrix: preliminaries

Our Bayesian model, following fitting of model and posterior parameters, allows us to compute the posterior probability of having a specified number of noise counts in each entry of the count matrix. Even though we marginalize c_{ng}^{noise} during inference, we can recover its posterior after model fitting via posterior sampling. We formally have

$$p\left(c_{ng}^{\text{noise}} \mid \{c_{ng}\}\right) = \int \mathrm{d}Z\, q(Z)\, \frac{\text{NegBinom}\left(c_{ng} - c_{ng}^{\text{noise}} \mid \mu_{ng}^{\text{cell}}, \Phi\right)\ \text{Poisson}\left(c_{ng}^{\text{noise}} \mid \lambda_{ng}^{\text{noise}}\right)}{\sum_{c_{ng}^{\text{noise}}=0}^{c_{ng}} \text{NegBinom}\left(c_{ng} - c_{ng}^{\text{noise}} \mid \mu_{ng}^{\text{cell}}, \Phi\right)\ \text{Poisson}\left(c_{ng}^{\text{noise}} \mid \lambda_{ng}^{\text{noise}}\right)}, \tag{11}$$

where Z is the bundle of all other latent variables along with their approximate posterior distribution $q(Z)$. The terms and expressions appearing in the integrand are evaluated at Z. In practice, we approximate the integral via N discrete Monte Carlo samples drawn from $q(Z)$ and keep track of the marginal posterior of noise counts for each of the $n \times g$ count matrix entries. We compute the probabilities in log space for numerical stability, truncate the allowed range of c_{ng}^{noise} to a safe upper bound, normalize each MC sample via the 'LogSumExp' operation and keep track of the running total over MC samples via sequential 'LogSumExp' operations for memory efficiency.

The obtained $n \times g$ marginal posterior distributions comprise our full probabilistic knowledge of noise counts for each entry in the count matrix. Standard single-cell downstream analysis workflows, however, by and large expect a single point estimate for input, as opposed to a distribution. Furthermore, a plurality of widely used algorithms such as voom[56] for differential expression analysis, the highly variable gene selection of Seurat version 3 (ref. 57) and scVI[35] for latent space learning, explicitly expect integer counts as input due to employment of discrete likelihood models such as negative binomial. These expectations motivate us to estimate a single integer matrix of noise counts, $\hat{c}_{ng}^{\text{noise}}$, from the obtained Bayesian posterior $p\left(c_{ng}^{\text{noise}} \mid \{c_{ng}\}\right)$ and produce an integer matrix of denoised counts $\hat{c}_{ng}^{\text{cell}} = c_{ng} - \hat{c}_{ng}^{\text{noise}}$ as the primary output of CellBender. The strict satisfaction of $c_{ng} = \hat{c}_{ng}^{\text{cell}} + \hat{c}_{ng}^{\text{noise}}$ implies the complementarity of noise and signal estimators. Hereafter, we focus on estimating the noise matrix for concreteness.

Canonical Bayes estimators for summarizing $p\left(c_{ng}^{\text{noise}} \mid \{c_{ng}\}\right)$ as a single point estimate include: (1) the posterior mean $\mathbb{E}\left[c_{ng}^{\text{noise}} \mid \{c_{ng}\}\right]$ and (2) the posterior mode $\text{argmax}\, p\left(c_{ng}^{\text{noise}} \mid \{c_{ng}\}\right)$, also known as the MAP estimator. The posterior mean estimator is an unbiased estimator;

however, it yields non-integer values, which is undesirable. The MAP estimator yields integer values; however, it is a biased estimator. For example, the MAP estimator systematically underestimates noise counts for genes that have lower noise prior rate than the cell expression prior rate ('On the asymptotic bias of canonical Bayes estimators'). Neither of these canonical estimators provide a tunable parameter for increasing or decreasing the strength of denoising and controlling the tradeoff between denoising sensitivity and specificity.

To address these shortcomings, we introduce a number of application-specific estimators to meet our specific needs. In general, we aim to develop estimation strategies for attaining the highest posterior probability subject to specified population-level constraints such as gene-wise or dataset-wise total noise budgets or expected FPR. Having such handles is useful in many downstream applications such as ascertaining the specificity of marker genes. We note that the true Bayesian recipe for conveying the results of CellBender is the full posterior and not a point estimate and that the optimality of an integer noise estimator is not universal and depends on the downstream application. For example, the desire to have an estimator suitable for differential expression testing between samples imposes a different set of constraints than the desire to have a given degree of certainty that each count in the output is not noise. We examine the merits and drawbacks of each strategy using different metrics in the following sections.

Estimating the integer noise matrix as a multiple-choice knapsack problem

We show that the problem of estimating an integer noise matrix, c_{ng}^{noise}, that attains maximum posterior probability subject to linear constraints is equivalent to the MCKP, which is a classical combinatorial optimization problem. To set the stage, we assume a linear index map, $\mathcal{I} : m \to (n, g)$, from $m \in \{1, \ldots, N \times G\}$ to the entries of the count matrix (n, g), for $1 \le n \le N$ and $1 \le g \le G$. Let $\mathcal{M}$ be the index set of noise count matrix elements that we wish to perform constrained estimation over. Choices include the entire count matrix $\mathcal{M}_D$, a row (cell), $\mathcal{M}_n$, or a column (gene), $\mathcal{M}_g$. We define $X_{mc} \in \{0, 1\}$ to be a binary variable that is 1 if the noise count for the matrix element at $\mathcal{I}(m)$ is set to c and is 0 otherwise. As there is a unique choice to be made for each matrix entry, we require $\sum_{c=0}^{C} X_{mc} = 1$, where C is the maximum specified noise count and has an upper bound of $\max c_{ng}$. We further define a 'reward' for each assignment as $V_{mc} \equiv \log p(c_{\mathcal{I}(m)}^{\text{noise}} = c \mid \{c_{ng}\})$, that is, the log posterior probability for that assignment. Finally, we wish to impose a lower bound L on the sum total of noise counts. This is readily expressed as $\sum_{m \in \mathcal{M}} \sum_{c=0}^{C} X_{mc} w_c \ge L$, where $w_c = (0, 1, \ldots, C)$ is an integer-valued weight vector. Maximizing the log posterior probability, which is given as $\sum_{m \in \mathcal{M}} \sum_{c=0}^{C} X_{mc} V_{mc}$ subject to the aforementioned constraints, is expressed as

$$\max \sum_{m \in \mathcal{M}} \sum_{c=0}^{C} X_{mc} V_{mc}, \quad \text{subject to} \begin{cases} X_{mc} \in \{0, 1\}, \\ \sum_{c=0}^{C} X_{mc} = 1, \\ \sum_{m \in \mathcal{M}} \sum_{c=0}^{C} X_{mc} w_c \ge L, \end{cases} \tag{12}$$

which is precisely the MCKP problem. MCKP is a classical NP-hard problem that admits a pseudo-polynomial dynamic programming solution. In our specific case, we show that, subject to mild assumptions, a fast and exact solution is feasible with time complexity, $\mathcal{O}(|\mathcal{M}| \times |L - L^*|)$, where $L^* = \sum_m \text{argmax}_c V_{mc}$ ('A fast and exact MCKP solver for strictly log–concave posterior distributions'). Note that L^* is the sum of MAP estimates over the specified count matrix entries $m \in \mathcal{M}$. As the noise rate is typically lower than the endogenous expression rate, L^* is typically an underestimate ('On the asymptotic bias of canonical Bayes estimators'), and, as such, we are generally interested in cases in which $L > L^*$ to overcome the asymptotic bias of the MAP estimator. Moving away from the MAP estimator by definition decreases the posterior probability. As such, the inequality constraint is realized at the threshold L, and, thus, we refer to L as the 'noise target'.

Concrete MCKP problems for enforcing gene-wise and dataset-wise noise count constraints. The MCKP framework allows us to impose noise targets over arbitrary selections of count matrix entries. For concreteness, we consider two scenarios: (1) imposing gene-wise constraints, where each column g of the noise count matrix is constrained to sum to $\ge L_g$ and is estimated independently and (2) dataset-wise constraints, where all count matrix entries are estimated at once subject to a global constraint that the sum total of noise counts $\ge L$. Setting the noise target may also be done in a different ways. Here, we consider two strategies: (1) a noise target based on an nFPR and (2) a noise target based on the cumulative distribution function (CDF) of the posterior of the aggregated noise counts. These strategies are described below.

Using the nFPR to specify the noise target. We introduce a single tunable parameter, nFPR $\in [0, 1]$, to specify the noise target. We define this parameter such that nFPR = 1 implies allocating all raw counts as noise counts, whereas nFPR = 0 implies removing as many noise counts as what is inferred from the model posterior aggregated over the appropriate slice of the dataset, that is, either gene-wise or for the full dataset. Specifically, we define nFPR as follows. For each gene g, we estimate the expected noise count per likely cell-containing droplet as follows:

$$\overline{c_g^{\text{noise}}} \sim \frac{\sum_n I[q_n > q^*] \left((1 - \bar{\rho})\, d_\mu^{\text{drop}}\, \epsilon_{n;\mu}\, \chi_g^{\text{a}} + \bar{\rho}\, \bar{\chi}_g\, \epsilon_{n;\mu}\, c_{ng} \right)}{\sum_n I[q_n > q^*]}, \tag{13}$$

where $I[q_n > q^*]$ is an indicator function with value 1 when $q_n > q^*$ and 0 otherwise, $q^* = 0.5$ is the threshold we have chosen for determining likely cell-containing droplets, and $\bar{\rho} = \hat{\rho}_\alpha (\hat{\rho}_\alpha + \hat{\rho}_\beta)^{-1}$ is the posterior mean of the barcode-swapping rate. The two terms in the numerator correspond to ambient and barcode-swapping contributions to noise counts. Likewise, we estimate cell counts as follows:

$$\overline{c_g^{\text{cell}}} \sim \frac{\sum_n I[q_n > q^*] \max\left(c_{ng} - (1 - \bar{\rho})\, d_\mu^{\text{drop}}\, \epsilon_{n;\mu}\, \chi_g^{\text{a}} - \bar{\rho}\, \bar{\chi}_g\, \epsilon_{n;\mu}\, c_{ng}, 0 \right)}{\sum_n I[q_n > q^*]}. \tag{14}$$

Equipped with these two aggregate estimates, we define the nFPR recipe for specifying the per-cell per-gene noise target ℓ_g as

$$\ell_g = \overline{c_g^{\text{noise}}} + \text{nFPR}\ \overline{c_g^{\text{cell}}}. \tag{15}$$

The gene-wise total noise target for N cells is given as $L_g = N \ell_g$, and the dataset-wise total noise target for N cells is given as $L = N \sum_g \ell_g$.

Using the aggregated noise posterior CDF quantiles to specify the noise target. Another strategy for setting a total noise target over a slice of the dataset is via the quantiles of the aggregated noise posterior. The aggregated noise over the desired set of count matrix entries $m \in \mathcal{M}$ is defined as

$$c_{\mathcal{M}}^{\text{noise}} \equiv \sum_{m \in \mathcal{M}} c_{\mathcal{I}(m)}^{\text{noise}}. \tag{16}$$

The posterior distribution of $c_{\mathcal{M}}^{\text{noise}}$ is formally given as the convolution of the posterior distribution of the included noise count matrix entries. The latter can be obtained numerically using fast Fourier transform. In practice, we have found that calculating the first two moments of $c_{\mathcal{M}}^{\text{noise}}$ and appealing to the central limit theorem yields virtually identical results. These moments are given as

$$\begin{aligned} \mu_{\mathcal{M}} &\equiv \sum_{m \in \mathcal{M}} \mathbb{E}_{c_{\mathcal{I}(m)}^{\text{noise}} \sim p\left(c_{\mathcal{I}(m)}^{\text{noise}} \mid \{c_{ng}\}\right)} \left[c_{\mathcal{I}(m)}^{\text{noise}} \right], \\ \sigma_{\mathcal{M}}^2 &\equiv \sum_{m \in \mathcal{M}} \text{Var}_{c_{\mathcal{I}(m)}^{\text{noise}} \sim p\left(c_{\mathcal{I}(m)}^{\text{noise}} \mid \{c_{ng}\}\right)} \left[c_{\mathcal{I}(m)}^{\text{noise}} \right], \end{aligned} \tag{17}$$

where Var denotes the variance and the central limit theorem implies that $c_{\mathcal{M}}^{\text{noise}} \simeq \mathcal{N}(\mu_{\mathcal{M}}, \sigma_{\mathcal{M}}^2)$. Given a total noise CDF quantile, q, we set the noise target to

$$L = \mu_{\mathcal{M}} + \sigma_{\mathcal{M}}\, \Phi^{-1}(q), \tag{18}$$

where $\Phi^{-1}(q)$ is the inverse CDF of the normal distribution. Similar to before, we can set $\mathcal{M}$ to either $\mathcal{M}_D$ or $\mathcal{M}_g$ for imposing dataset-wise or gene-wise noise targets, respectively.

Estimating the integer noise matrix via element-wise noise posterior CDF quantiles

A straightforward strategy for estimating the integer noise count matrix is to pick the noise count for each entry of the noise count matrix according to a specified CDF quantile, q. In particular, the choice $q = 0.5$ corresponds to the posterior median estimator, which is a canonical Bayes estimator. Specifying a higher (lower) value for q results in removing more (fewer) noise counts, and, as such, q serves as a handle for setting the denoising eagerness of CellBender. This algorithm is implemented as follows. For each cell n and gene g, we calculate the CDF of noise counts $F_{ng}^{\text{noise}}(c_{ng}^{\text{noise}})$ from the noise posterior

$$F_{ng}^{\text{noise}}(x) = \sum_{c=0}^{x} p\left(c_{ng}^{\text{noise}} = c \mid \{c_{ng}\}\right). \tag{19}$$

The estimated integer noise count matrix is then obtained as

$$\hat{c}_{ng}^{\text{noise}} = \text{argmax}_x \left[F_{ng}^{\text{noise}}(x) \le q\right]. \tag{20}$$

This estimator, as opposed to the MCKP approach discussed in the previous section, does not involve solving a global constrained optimization problem and, as such, does not allow targeting noise counts in aggregate, in either a gene-wise or a dataset-wise manner. While it is possible to fine tune the quantile threshold q to achieve the desired nFPR, we did not attempt it: MCKP achieves the same goal by allocating the total noise budget more globally rather than locally and, as such, can achieve a higher total posterior probability.

Estimating the integer noise matrix via posterior regularization

Another strategy for estimating an integer noise matrix subject to external constraints, such as dataset-level or gene-wise nFPR, is provided by the framework of posterior regularization of ref. 58 and is another optimization-based approach. This is the framework we had adopted in CellBender version 0.2.0, and we provide it here for completeness. Concretely, following the set-up of equation (4) from ref. 58 (with no slack, that is, $\varepsilon = 0$), given data $X = \{c_{ng}\}$ and latent variables Z, we seek a posterior distribution p_{reg}^* that solves the following constrained optimization problem:

$$\text{argmin}_{p_{\text{reg}}}\, \mathbb{KL}\left(p_{\text{reg}}(Z) \,||\, p(Z \mid X)\right), \quad \text{subject to} \quad \mathbb{E}_{p_{\text{reg}}}\left[\Phi(X,Z)\right] \ge b, \tag{21}$$

where KL denotes the Kullback–Leibler divergence, $p(Z \mid X)$ is the unregularized Bayesian posterior, $p_{\text{reg}}(Z)$ is the sought-after regularized posterior, and $\Phi(X, Z)$ is a specified function of raw data and latent variables that we wish to constrain below a specified value of b in expectation. We have implicitly grouped the model parameters together with the latent variables in Z. Adapted to our problem, we wish to compute a regularized posterior for noise counts, $p_{\text{reg}}(c_{ng}^{\text{noise}})$, such that it is as close as possible to the regularized posterior in terms of KL divergence, while the expected total noise count over all likely cell-containing droplets is controlled by the user-specified nFPR parameter (equation (15)):

$$\text{argmin}_{p_{\text{reg}}}\, \mathbb{KL}\left(p_{\text{reg}}(c_{ng}^{\text{noise}}) \,||\, p(c_{ng}^{\text{noise}} \mid c_{ng})\right) \tag{22}$$

$$\text{subject to} \quad \mathbb{E}_{p_{\text{reg}}}\left[\frac{\sum_n I[q_n > q^*]\, c_{ng}^{\text{noise}}}{\sum_n I[q_n > q^*]}\right] \ge \overline{c_g^{\text{noise}}} + \text{nFPR}\ \overline{c_g^{\text{cell}}}, \tag{23}$$

where $q^* = 0.5$ is the posterior probability threshold that we have chosen for likely cell-containing droplets. As it is written, the nFPR condition is imposed separately for each gene g. A more relaxed version of the problem is obtained by summing both sides of the constraint over g, which is equivalent to imposing a dataset-wise constraint. In the dual formulation[58], the regularized posterior that satisfies equation (21) can be written as

$$\omega^* = \underset{\omega \ge 0}{\text{argmax}}\ \left[-b\,\omega - \log Q(\omega)\right], \tag{24a}$$

$$Q(\omega) = \int \mathrm{d}Z\, p(Z \mid X) \exp\left[-\omega\, \Phi(X,Z)\right], \tag{24b}$$

$$p_{\text{reg}}^*(Z) = \frac{p(Z \mid X) \exp\left[-\omega^*\, \Phi(X,Z)\right]}{Q(\omega^*)}, \tag{24c}$$

where ω is an auxiliary Lagrange multiplier, and the problem is reduced to finding an appropriate ω^* that satisfies the constraint imposed by b and $\Phi(\cdot)$. Exact posterior regularization (PR) requires separate SVI model fits for every choice of constraint threshold (b in equation (22)). In theory, one could solve the optimization problem posed by equation (24a)–(24c) in dual form, plugging $p_{\text{reg}}^*(Z)$ into the ELBO (equation (4)) and interleaving SVI updates with constrained satisfaction updates: a computationally prohibitive task. Another approach is augmented Lagrangian constrained optimization, in which one concurrently updates ω along with model parameters using the same stochastic optimizer to minimize the ELBO while also approximately satisfying equation (24a).

Here, we make an approximate simplifying assumption akin to perturbation theory: as long as the user does not impose extreme values of expected nFPR compared to the FPR achieved in the unregularized problem, then we expect all latent variables to remain approximately the same, with and without PR, with the exception of perhaps c_{ng}^{noise}, which directly appears in the constraint. By employing this approximation, we can freeze all latent variables to their unregularized posteriors and only regularize $p\left(c_{ng}^{\text{noise}} \mid \{c_{ng}\}\right)$ ex post facto. To achieve this goal, consider scaling $\lambda_{ng}^{\text{noise}} \to \beta_g \lambda_{ng}^{\text{noise}}$, where $\lambda_{ng}^{\text{noise}}$ is the Poisson rate of exogenous counts given in equation (3) and $\beta_g \ge 0$ is a to-be-determined scale factor. We postulate that finding the optimal scale factor that satisfies the posterior constraint is equivalent to solving equation (24a)–(24c). To show this, we use the following identity, which can be readily ascertained using the explicit expression of the Poisson probability mass function:

$$\text{Poisson}\left(c_{ng}^{\text{noise}} \mid \beta_g \lambda_{ng}^{\text{noise}}\right) = e^{\lambda_{ng}(1-\beta_g)}\, \beta_g^{c_{ng}^{\text{noise}}}\ \text{Poisson}\left(c_{ng}^{\text{noise}} \mid \lambda_{ng}^{\text{noise}}\right). \tag{25}$$

According to the dual formulation given in equation (24a)–(24c), we can write $p_{\text{reg}}^*(c_{ng}^{\text{noise}})$ for likely cell-containing droplets, that is, $q_n > q^*$, as

$$p_{\text{reg}}^*\left(c_{ng}^{\text{noise}}\right) = \frac{\text{Poisson}\left(c_{ng}^{\text{noise}} \mid \lambda_{ng}^{\text{noise}}\right) \exp\left[-\omega^*\ c_{ng}^{\text{noise}}\right]}{\sum_{c_{ng}^{\text{noise}}} \text{Poisson}\left(c_{ng}^{\text{noise}} \mid \lambda_{ng}^{\text{noise}}\right) \exp\left[-\omega^*\ c_{ng}^{\text{noise}}\right]}. \tag{26}$$

Comparing this to equation (25), we identify $\omega^* = -\log \beta_g^*$. In other words, solving for the regularized posterior reduces to the problem of finding the largest noise scale factor β_g^* that satisfies the constraint in equation (22). The regularized Poisson rate for noise counts is then $\beta_g^* \lambda_{ng}^{\text{noise}}$. For a dataset-level nFPR constraint, the gene-wise scale factor β_g^* reduces to a single global scale factor, β^*. At the moment, only the dataset-level nFPR condition is implemented in CellBender.

Locating the optimal β^* via binary search and estimating the integer noise matrix. We locate the optimal noise scale factor β^* numerically using a binary search strategy. Our goal is to identify the largest value β^* such that the inequality given in equation (22) is satisfied. A binary search is performed over the range $\beta^* \in [0.01, 500]$. At each iteration of the search, we estimate $\mathbb{E}_q[c_{ng}^{\text{noise}}]$ by obtaining the regularized posterior using equation (11) and making the replacement $\lambda_{ng}^{\text{noise}} \to \beta^* \lambda_{ng}^{\text{noise}}$. For computational efficiency, we only include a random subset of likely cell-containing droplets (128 randomly chosen cells by default). The entire optimization procedure is repeated five times using different randomly chosen subsets of cells. The final value of β^* is the average from the several repeats. Having located the optimal β^* value, we obtain the integer noise count matrix as the MAP estimate from the regularized noise posterior. We refer to this noise-estimation strategy as PR for mean targeting or 'PR-μ' for short.

Approximate noise CDF quantile targeting via posterior regularization. A variation of the discussed PR strategy is obtained by replacing the constraint appearing in equation (22) with the following:

$$\mathbb{E}_{p_{\text{reg}}}\left[c_{ng}^{\text{noise}}\right] \geq \mathbb{E}_{p}\left[c_{ng}^{\text{noise}}\right] + \alpha\, \sigma_p\left[c_{ng}^{\text{noise}}\right], \tag{27}$$

where $\alpha = \Phi^{-1}(q)$ is approximately equal to quantile q of noise under the normality assumption. Note that the constraint is imposed at the level of individual count matrix entries. The motivation for this approach is to allocate the extra noise budget preferentially to the count matrix entries with lower noise posterior confidence. Again, the dual form of the PR problem implies a solution, $p_{\text{reg}}(c_{ng}^{\text{noise}}) \propto p(c_{ng}^{\text{noise}} \mid \{c_{ng}\}) \exp(-\omega_{ng}^* c_{ng}^{\text{noise}})$, where ω_{ng}^* is a matrix of Lagrange multipliers to be determined to satisfy equation (27). In practice, we obtain ω_{ng}^* by performing a parallelized binary search as described earlier. Once the regularized posterior is obtained, the output can be summarized either by taking the posterior mean, the posterior mode or a single sample, all of which we compare later. Note that $\alpha = 0$ is identical to the unregularized posterior. We refer to this noise-estimation strategy as PR for quantile targeting or 'PR-q' for short.

Evaluating different noise-estimation strategies

We introduced several strategies for estimating noise counts from the Bayesian noise posterior in 'Estimating the integer noise matrix as a multiple-choice knapsack problem', 'Estimating the integer noise matrix via element-wise noise posterior CDF quantiles' and 'Estimating the integer noise matrix via posterior regularization' to address the shortcomings of canonical Bayes estimators and allow us to control the denoising sensitivity–specificity tradeoff. In this section, we evaluate these strategies on a simulated dataset that closely follows our model ('Simulated data generation'). Concretely, we generate a test dataset consisting of three 'cell types' with fixed gene expression profiles. We generate 100 cells of each type with 5,000 UMIs per cell on average and a background noise that consists of only ambient RNA for simplicity. The ambient RNA profile is taken to be the same as the average gene expression across all simulated cells, with 200 ambient UMIs per droplet on average.

Here, our focus is to evaluate various noise-estimation strategies after model fitting and inference. To sidestep confounding factors such as our ability to fit the model and infer the noise posterior (which depends on the dataset size, the degree of model faithfulness and our variational approximations), we assume perfect knowledge of all latent variables other than c_{ng}^{noise}. Such an oracle short circuits the marginalization over Z in equation (11) and evaluates the integrand at the true value of Z. Therefore, the performance metrics given in this section are theoretical upper bounds. A comparison of such theoretical upper bounds with actually attainable end-to-end results is given in Fig. 5c–g.

First, we evaluate the different estimators by studying their ROC curves. To construct an ROC curve, we consider each $n \times g$ entry of the noise count matrix, take 'noise' as the 'positive' class and calculate the 2×2 confusion matrix as follows:

$$\begin{aligned}
\text{TP}_{ng} &= \min\left(c_{ng}^{\text{noise}}, \hat{c}_{ng}^{\text{noise}}\right),\\
\text{FP}_{ng} &= \max\left(0, \hat{c}_{ng}^{\text{noise}} - c_{ng}^{\text{noise}}\right),\\
\text{TN}_{ng} &= \min\left(c_{ng}^{\text{cell}}, \hat{c}_{ng}^{\text{cell}}\right),\\
\text{FN}_{ng} &= \max\left(0, \hat{c}_{ng}^{\text{cell}} - c_{ng}^{\text{cell}}\right),
\end{aligned} \tag{28}$$

where c_{ng}^{noise} and c_{ng}^{cell} represent the simulated truth values, $\hat{c}_{ng}^{\text{cell}}$ is the CellBender output, and $\hat{c}_{ng}^{\text{noise}} = c_{ng} - \hat{c}_{ng}^{\text{cell}}$ (TP, true positive; FP, false positive; TN, true negative; FN, false negative). We 'summarize' the resulting $n \times g$ confusion matrix either (1) as a 'macro-average' per gene or per cell, where we sum the element-wise 2×2 confusion matrices along n or g, respectively, or (2) as a 'micro-average' per gene or or per cell, where we calculate the element-wise TPR_{ng} and FPR_{ng}, remove the undetermined entries and calculate the arithmetic mean along n or g, respectively. Extended Data Fig. 9 shows the resulting ROC curves for various estimation methods. We have further reduced the obtained TPR and FPR values for per-cell (or per-gene) micro-averages and macro-averages to a single point via arithmetic averaging for better visibility. The canonical Bayes estimators (black circle, square, diamond) each provide a single point on the ROC plane. By contrast, each of our estimators provides a natural parameter for controlling the position on the ROC curve.

It is clear that drawing a random sample either from the actual posterior (black triangle) or from the regularized posterior (PR-μ, orange; PR-q, purple) is a poor strategy, while also being inconsistent and non-deterministic estimators. Posterior mean estimators, either unregularized (diamond) or regularized (PR-q, brown circles), neither produce an integer count matrix nor are among the top-performing estimators in terms of the ROC curve. Estimators based on the regularized posterior mode (PR-μ, blue circles; PR-q, red circles), the element-wise posterior CDF quantiles (green circles) and MCKP estimators (per-gene nFPR target, pink; global nFPR target, gray) all do well and are practically tied in terms of the ROC curve, with the estimator based on element-wise posterior CDF quantiles showing a slight advantage in this benchmark.

To further distinguish the characteristics of the different estimators, we also study over-removal or under-removal of noise counts for each gene versus total gene expression in Extended Data Fig. 10. The ideal estimator is expected (1) to exhibit the same characteristics across the entire gene expression spectrum and (2) to not under-remove or over-remove noise counts when the total noise budget is chosen in a balanced way (that is, $q = 0.5$ for CDF-based targets or nFPR ~ 0). Among the top-performing estimators in terms of the ROC analysis, we find that MCKP with a per-gene nFPR target satisfies both expectations (last row in Extended Data Fig. 10). Specifying a dataset-level (global) noise budget tends to overcorrect highly expressed genes (PR-μ posterior mode and MKCP global nFPR target in Extended Data Fig. 10).

In summary, our analysis highlights two estimation strategies: (1) the MCKP estimator with gene-wise nFPR control, which shows decent ROC characteristics and a consistent performance across the entire gene expression spectrum and (2) element-wise posterior CDF quantiles, which show the best ROC characteristics although with some dependence on the gene expression rate. We have chosen the former as the default estimation strategy in the latest release of CellBender (version 0.3.0_rc). The previous version (version 0.2.0) used the PR-μ strategy, which, as we have shown here, is inferior to the MCKP. Finally, we note that all of these estimation strategies are implemented in CellBender, should a user have a use case that warrants a strategy other than the default.

A fast and exact MCKP solver for strictly log–concave posterior distributions

MCKP is an NP-hard problem that admits a pseudo-polynomial dynamic programming solution. Here, we show that assuming strict logarithmic concavity of the noise posterior distribution leads to a fast and exact solution of the MCKP with time complexity $\mathcal{O}(M \times |L - L^*|)$, where $L^* = \sum_m \mathrm{argmax}_c\, V_{mc}$.

We define a log–concave discrete distribution as follows: a discrete probability distribution $p(k) : \{0\} \cup \mathbb{N} \to \mathbb{R}^+$ is called logarithmically concave if and only if $\log p(k+1) + \log p(k-1) \leq 2 \log p(k)$. It is called strictly logarithmically concave if $\leq$ is replaced with strict inequality.

Many common probability distributions are logarithmically concave, including Poisson and negative binomial distributions, most of which are also strictly logarithmically concave except for a measure zero set of parameters. We do not aim to rigorously prove the conditions for strict logarithmic concavity of our noise posterior distribution. However, we have empirically verified that this property holds in various datasets. To motivate this empirical observation, consider the limit $\Phi \to 0$ and $q(Z) \to \delta(Z - Z')$, where δ represents a Dirac delta function. It is easily shown that the noise posterior tends to the binomial distribution with a success probability of $p = \lambda_{ng}^{\mathrm{noise}}(\lambda_{ng}^{\mathrm{noise}} + \mu_{ng}^{\mathrm{cell}})^{-1}$ and total number of trials $N = c_{ng}$ in this limit ('On the asymptotic bias of canonical Bayes estimators'), which is a log–concave distribution. Continuity implies the existence of an extended parameter regime around this limit where logarithmic concavity holds. Increasing Φ or the dispersion in $q(Z)$ can be thought of as imparting uncertainty on p. Modeling this uncertainty as a beta distribution, the noise posterior may then be approximated as a beta–binomial distribution, which is also strictly logarithmically concave except for a measure zero set of parameters or irrelevant parameter regimes, for example, bimodal success probability p. Hereafter, we assume the strict logarithmic concavity of the noise posterior as given.

We call the MCKP problem posed by equation (12) a 'strictly convex MCKP problem' if and only if the reward weights $V_{mc} \equiv \log p(c_{\mathcal{J}(m)}^{\mathrm{noise}} = c \,|\, \{c_{ng}\})$ are derived from strictly log–concave distributions. We will show that the strictly convex MCKP problem admits an exact greedy solution. To set the stage, consider the unconstrained MAP estimate $X_{mc}^* = \delta(c, \mathrm{argmax}_c\, V_{mc})$ and observe that it achieves the total noise target $L^* = \sum_m \sum_{c=0}^{C} c\, X_{mc}^* = \sum_m \mathrm{argmax}_c\, V_{mc}$. Clearly, if the specified total noise target L coincides with L^*, then X_{mc}^* is indeed the optimal solution because each reward term is individually maximized, the constraint is satisfied with equality, and moving away from the equality constraint satisfaction implies deviating from the MAP point and thus decreasing the reward. In a nutshell, our greedy strategy is to take X_{mc}^* as a reference point and iteratively modify it via best local moves such that the specified noise target is met. To this end, we define $\Delta = L - L^*$ as the gap between the total noise count of the MAP solution X_{mc}^* and the specified total noise target. We refer to the sought-after solution as $X_{mc}^*(\Delta)$. By definition, $X_{mc}^*(0) \equiv X_{mc}^*$. We only consider the case $\Delta > 0$ here. The case $\Delta < 0$ can be worked out by symmetry. Our greedy algorithm for solving this problem for $\Delta > 0$ is as follows. To obtain $X_{mc}^*(1)$ from $X_{mc}^*(0)$, we consider $|\mathcal{M}|$ local moves where the noise count for each coordinate m is increased by 1 while keeping the other coordinates fixed; we chose the local move that yields the highest possible reward. Note that we are not considering all possible moves that satisfy the constraint, for example, removing two noise counts from a coordinate and adding three counts to another. We proceed with this greedy strategy in an iterative fashion until we reach the desired Δ.

The greedy iterative coordinate-ascent algorithm solves the strictly convex MCKP problem exactly.

For proof, consider the following objective function:

$$\mathcal{L}(x_1, \ldots, x_{|\mathcal{M}|}) \equiv \sum_{m=1}^{|\mathcal{M}|} \sum_{c=0}^{C} W_{mc}\, \delta\left(c, x_m + x_m^*\right), \tag{29}$$

where

$$W_{mc} \equiv \max_c \left[\log p\left(c_{\mathcal{J}(m)}^{\mathrm{noise}} = c \,\middle|\, \{c_{ng}\}\right)\right] - \log p\left(c_{\mathcal{J}(m)}^{\mathrm{noise}} = c \,\middle|\, \{c_{ng}\}\right),$$
$$x_m^* \equiv \mathrm{argmax}_c \left[\log p\left(c_{\mathcal{J}(m)}^{\mathrm{noise}} = c \,\middle|\, \{c_{ng}\}\right)\right],$$

and $x_m \in \{0\} \cup \mathbb{N}$ is the 'extra' noise counts allocated to count matrix entry m on the top of the MAP point x_m^*. We refer to the vector of extra noise counts and MAP counts as $\mathbf{x}$ and $\mathbf{x}^*$, respectively. Minimizing $\mathcal{L}$ subject to the total noise constraints given in equation (12) is equivalent to solving the MCKP problem. In the new notation, $\mathcal{L}$ conveniently achieves its minimum value of 0 at $\mathbf{x} = 0$, which corresponds to the unconstrained MAP point. This is due to implicitly setting the MAP point as the reference point in the definition of W_{mc}. The strict logarithmic concavity of noise posterior distributions implies strict convexity of W_{mc} in the following discrete sense:

$$W_{m,c+1} + W_{m,c-1} > 2W_{m,c}, \qquad m = 1, \ldots, |\mathcal{M}|, \tag{30}$$

which follows from the definition of a log–concave discrete distribution. As a consequence, $\mathcal{L}$ emerges as a separable function of strictly convex one-dimensional functions over non-negative integers. We will use this property repeatedly to establish the optimality of coordinate-ascent moves. We define $B(\Delta)$ as the subspace of points that satisfy the total noise constraint with equality at $L^* + \Delta$:

$$B(\Delta) = \left\{ (x_1, \ldots, x_{|\mathcal{M}|}) \;\middle|\; \sum_{m \in \mathcal{M}} x_m = L^* + \Delta \right\}. \tag{31}$$

We observe that $B(\Delta)$ is a discrete convex set in the sense that, if $(x_1, \ldots, x_{|\mathcal{M}|}) \in B(\Delta)$, then $(x_1, \ldots, x_i + 1, \ldots, x_j - 1, x_{|\mathcal{M}|}) \in B(\Delta)$ for all i and j.

As a consequence, the restriction of $\mathcal{L}$ to $B(\Delta)$ is also strictly convex, implying that (1) any local minimum of $\mathcal{L}$ over $B(\Delta)$ is the global minimum and (2) the global minimum of $\mathcal{L}$ over $B(\Delta)$ is unique. Therefore, to prove the optimality of coordinate ascent, it is sufficient to show that the point obtained by applying coordinate ascent to the minimizer of $\mathcal{L}$ in subspace $B(\Delta)$, namely $\mathbf{x}^*(\Delta)$, yields a local minimum in the next subspace $B(\Delta + 1)$. Global optimality and uniqueness follow from strict convexity. Consider the set of all $|\mathcal{M}|$ local coordinate ascent moves from $\mathbf{x}^*(\Delta)$, and let $\hat{m}$ be the coordinate to which adding a noise count accrues the smallest increase in $\mathcal{L}$. This implies that

$$W_{\hat{m}, x_{\hat{m}}^*(\Delta)+1} - W_{\hat{m}, x_{\hat{m}}^*(\Delta)} < W_{m, x_m^*(\Delta)+1} - W_{m, x_m^*(\Delta)}, \qquad \forall m \neq \hat{m}. \tag{32}$$

We denote the coordinate-ascent update of $\mathbf{x}^*(\Delta)$ as $\tilde{\mathbf{x}}(\Delta + 1)$:

$$\tilde{x}_m(\Delta + 1) = x_m^*(\Delta) + \Delta_{m,\hat{m}}. \tag{33}$$

The set of nearest-neighbor points of $\tilde{\mathbf{x}}(\Delta + 1)$, namely $\mathrm{N}_{\tilde{\mathbf{x}}(\Delta+1)}$, can be written as the union of three mutually exclusive sets of points: (1) $\mathrm{N}_{\leftarrow}$ points obtained by moving backward along coordinate $\hat{m}$ and moving forward along another coordinate $i \neq \hat{m}$; there are $|\mathcal{M}| - 1$ such points; (2) $\mathrm{N}_{\rightarrow}$ points obtained by moving further forward along coordinate $\hat{m}$ and moving backward along another coordinate $i \neq \hat{m}$; there are $|\mathcal{M}| - 1$ such neighbors; (3) $\mathrm{N}_{\perp}$ points obtained by keeping coordinate $\hat{m}$ fixed, choosing two other coordinates i, j such that $i \neq j \neq \hat{m}$, moving forward along i and backward along j; there are $(|\mathcal{M}| - 1)(|\mathcal{M}| - 2)$ such moves. Put together, the three mutually exclusive sets comprise $|\mathcal{M}|(|\mathcal{M}| - 1)$ nearest-neighbor points of $\tilde{\mathbf{x}}(\Delta + 1)$,

$$\mathrm{N}_{\tilde{\mathbf{x}}(\Delta+1)} = \mathrm{N}_{\leftarrow} \cup \mathrm{N}_{\rightarrow} \cup \mathrm{N}_{\perp}.$$

We wish to show that $\mathcal{L}(\mathbf{x} \in \mathrm{N}_{\tilde{\mathbf{x}}(\Delta+1)}) > \mathcal{L}(\tilde{\mathbf{x}}(\Delta + 1))$. First, we note that the $|\mathcal{M}| - 1$ points in $\mathrm{N}_{\leftarrow}$ coincide with the $|\mathcal{M}| - 1$ rejected forward moves, which by definition lead to a higher value of $\mathcal{L}$ over $B(\Delta + 1)$

(equation (32)). Therefore, all points in $\mathrm{N}_{\leftarrow}$ are directions of ascent. For an arbitrary point $\mathbf{x}_{\leftarrow} \in \mathrm{N}_{\leftarrow}$ obtained by stepping backward along coordinate i and further forward along $\hat{m}$, we have

$$\mathcal{L}(\mathbf{x}_{\leftarrow}) - \mathcal{L}(\bar{\mathbf{x}}(\Delta+1)) = \underbrace{\left[W_{\hat{m},x^*_{\hat{m}}(\Delta)+2} + W_{\hat{m},x^*_{\hat{m}}(\Delta)} - 2W_{\hat{m},x^*_{\hat{m}}(\Delta)+1}\right]}_{>0} + \underbrace{\left[W_{\hat{m},x^*_{\hat{m}(\Delta)}+1} + W_{i,x^*_{i(\Delta)}-1} - W_{\hat{m},x^*_{\hat{m}}(\Delta)} - W_{i,x^*_i(\Delta)}\right]}_{>0} > 0. \tag{34}$$

The first term is positive due to strict convexity, and the second term is positive due to $\mathbf{x}^*(\Delta)$ being the minimizer of $\mathcal{L}$ in subspace $B(\Delta)$. Finally, for a point $\mathbf{x}_{\perp} \in \mathrm{N}_{\perp}$ obtained by stepping forward and backward along coordinates i and j, respectively, we have

$$\mathcal{L}(\mathbf{x}_{\perp}) - \mathcal{L}(\bar{\mathbf{x}}(\Delta+1)) = W_{i,x^*_i(\Delta)+1} + W_{j,x^*_j(\Delta)-1} - W_{i,x^*_i(\Delta)} - W_{j,x^*_j(\Delta)} > 0, \tag{35}$$

which directly results from $\mathbf{x}^*(\Delta)$ being the minimizer of $\mathcal{L}$ in subspace $B(\Delta)$. Put together, we have shown that $\bar{\mathbf{x}}(\Delta+1)$ is a local minimizer of $\mathcal{L}$ in subspace $B(\Delta+1)$. Strict convexity implies that $\bar{\mathbf{x}}(\Delta+1)$ is also the unique and global minimizer:

$$\mathbf{x}^*(\Delta+1) \equiv \bar{\mathbf{x}}(\Delta+1). \tag{36}$$

Therefore, following the iterative coordinate-ascent trajectory that connects the MAP point $\mathbf{x}^*(0)$ to $\mathbf{x}^*(\Delta)$ yields the unique solution of the strictly convex MCKP problem. There are $\Delta = |L - L^*|$ iterations, and each iteration involves $|\mathcal{M}|$ comparisons to locate the optimal coordinate. Therefore, the complexity of this algorithm is $\mathcal{O}(|\mathcal{M}| \times |L - L^*|)$. As mentioned earlier, the case $\Delta < 0$ can be worked out by symmetry, that is, replacing 'backward' moves with 'forward' moves.

In practice, we implement the coordinate-ascent strategy by pre-computing, pooling and sorting differential coordinate ascents $\delta_{m,c} \equiv W_{m,c+1} - W_{m,c}$. Even though the time complexity of this implementation is $\mathcal{O}(|\mathcal{M}||L - L^*| \times \log(|\mathcal{M}| \times |L - L^*|))$, it runs faster on GPU hardware by leveraging parallelism.

On the asymptotic bias of canonical Bayes estimators

We mentioned the shortcomings of canonical Bayes estimators as part of our motivations for developing application-specific integer noise estimators. These include the non-integral estimates obtained by the posterior mean and the asymptotic bias of the posterior mode estimator, also known as the MAP estimator. In this section, we study these estimators in more detail in a simple setting that is related to our application. We consider the simplifying limit $\Phi \to 0$ and $q(Z) \to \delta(Z - Z^*)$ in equation (11), focus on a single count matrix entry and drop the n and g indices for brevity. In this limit, the posterior is found to be

$$\begin{aligned} p\left(c^{\text{noise}} \mid c\right) &= \frac{\text{Poisson}\left(c - c^{\text{noise}} \mid \mu^{\text{cell}}\right)\text{Poisson}\left(c^{\text{noise}} \mid \lambda^{\text{noise}}\right)}{\sum_{c^{\text{noise}}=0}^{c}\text{Poisson}\left(c - c^{\text{noise}} \mid \mu^{\text{cell}}\right)\text{Poisson}\left(c^{\text{noise}} \mid \lambda^{\text{noise}}\right)} \\ &= \text{Binomial}\left(p = \frac{\lambda^{\text{noise}}}{\lambda^{\text{noise}}+\mu^{\text{cell}}}; n_{\text{success}} = c^{\text{noise}}; n_{\text{trial}} = c\right). \end{aligned} \tag{37}$$

Here, λ^{noise} and μ^{noise} correspond to the noise count and cell count rates at the latent variable concentration point Z^*. We have also used $\lim_{\Phi\to 0}\text{NegBinom}(x \mid \mu, \Phi) = \text{Poisson}(x \mid \mu)$. The binomial equivalence can be either derived by interpreting Poisson variables as the sum of Bernoulli variables or by resorting to the algebraic expression of the Poisson probability mass function. In this limit, we find the posterior mean (PM) and MAP estimators to be

$$\begin{aligned} \hat{c}^{\text{noise}}_{\text{PM}} &= c\,\frac{\lambda^{\text{noise}}}{\lambda^{\text{noise}}+\mu^{\text{cell}}}, \\ \hat{c}^{\text{noise}}_{\text{MAP}} &= \left\lfloor (c+1)\,\frac{\lambda^{\text{noise}}}{\lambda^{\text{noise}}+\mu^{\text{cell}}} \right\rfloor. \end{aligned} \tag{38}$$

Note that the expression for $c^{\text{noise}}_{\text{MAP}}$ is only valid when the expression appearing in the floor function is non-integer, which is the case except for a measure zero set of points.

We consider N independent and identically distributed realizations of c^{noise} and c^{cell} and study the asymptotic bias of the two estimators in sample mean. This analysis is an idealization of taking a population of $N \to \infty$ droplets containing identical cells and checking whether or not the empirical mean of a given noise estimator converges to λ^{noise}. For the posterior mean estimator, we have

$$\mathbb{E}\left[\hat{c}^{\text{noise}}_{\text{PM}}\right] = \frac{\lambda^{\text{noise}}}{\lambda^{\text{noise}}+\mu^{\text{cell}}}\,\mathbb{E}_{c\sim\text{Poisson}(\lambda^{\text{noise}}+\mu^{\text{cell}})}[c] = \lambda^{\text{noise}}. \tag{39}$$

Therefore, we find posterior mean to be consistent and asymptotically unbiased. However, the estimator clearly yields non-integer values, which is undesirable. For the MAP estimator, we have

$$\mathbb{E}\left[\hat{c}^{\text{noise}}_{\text{MAP}}\right] = \sum_{c=0}^{\infty}\text{Poisson}\left(c \mid \lambda^{\text{noise}}+\mu^{\text{cell}}\right)\left\lfloor (c+1)\,\frac{\lambda^{\text{noise}}}{\lambda^{\text{noise}}+\mu^{\text{cell}}}\right\rfloor. \tag{40}$$

It is easy to see that this estimator is asymptotically biased. The floor term is identically vanishing for $c < c^* \equiv \lceil \mu^{\text{cell}}(\lambda^{\text{noise}})^{-1} \rceil$. In the relatively low-noise limit $\lambda^{\text{noise}} \ll \mu^{\text{cell}}$, c^* becomes arbitrarily larger than the mode of c, which is $\sim \mu^{\text{cell}}$ in this limit, and, subsequently, $\mathbb{E}\left[\hat{c}^{\text{noise}}_{\text{MAP}}\right]$ becomes arbitrarily smaller than the expected value of λ^{noise}. While the asymptotic bias of the MAP estimator can be studied analytically, we find it more straightforward to resort to a numerical study. We define the relative asymptotic bias of the MAP estimator as

$$\beta^{\text{MAP}}\left(\lambda^{\text{noise}}, \mu^{\text{cell}}\right) \equiv \frac{\mathbb{E}\left[\hat{c}^{\text{noise}}_{\text{MAP}}\right] - \lambda^{\text{noise}}}{\lambda^{\text{noise}}}. \tag{41}$$

Supplementary Fig. 10 shows β^{MAP} for a range of noise count and cell count prior rates. We notice $\beta^{\text{MAP}} \sim -1$ in the regime $\lambda^{\text{noise}} \ll \mu^{\text{cell}}$, as expected from the pathological behavior of the MAP estimator in the low-noise regime. In this regime, $\hat{c}^{\text{noise}}_{\text{MAP}} \sim 0$, implying that no noise count is removed from any cells.

Implementation details and technical remarks

The default architecture for the encoder network NN_z that maps c_{ng} to the bundle of $\mathbf{z}$-posterior location and scale ($\mathbf{z}_{n;\mu}$, $\mathbf{z}_{n;\sigma}$) has one hidden layer of 500 units, and the encoded dimension of Z of $\mathbf{z}_n$ is 100. Similarly, the decoder network NN_χ that maps $\mathbf{z}_n$ to χ_{ng} has one hidden layer of 500 units and a linear readout, followed by a softmax operation to bring the output to the $(G-1)$ simplex of normalized endogenous feature frequencies. The encoding network for y_n, d^{cell}_n and ϵ_n, denoted by NN_{enc} for brevity, works as follows. Inputs to the network consist of raw counts as well as three additional features that are handcrafted: (1) the log of total counts per droplet, (2) the log of the number of nonzero genes per droplet and (3) the overlap with the current estimate of the ambient RNA profile (which is calculated as a log probability that the observed droplet counts were drawn from a Poisson with rate equal to χ^{a}_g). Handcrafted features are concatenated to counts to form the input to the network. By default, the network has two hidden layers, [100, 50]. From the last hidden layer, three separate linear transformations take the hidden state and produce (1) logit cell probability logit (q_n), (2) the inverse variance of the gamma distribution for ϵ_n and (3) the log of mean cell sizes d^{cell}_n. Weights are initialized using PyTorch defaults, except for the weights that connect the handcrafted log counts per droplet input feature to the output for q_n, which are initialized to 1, so that the network starts with a condition that cell probability should closely follow log counts. Softplus non-linearities are used throughout. In practice, CellBender results are not very sensitive to the architecture of the encoders, and network architectures can be changed from the default values using command-line arguments.

We note that the initially learned biological gene expression landscape $NN_x(\mathbf{z}_n)$ may itself be contaminated with background RNA counts. However, as the inference procedure progresses and as the estimate of the background RNA profile improves, the maximum likelihood principle encourages the neural network to correct in a self-consistent fashion and learn to represent background-free gene expression profiles.

For numerical stability and to preclude vanishing gradients, we handle all probabilities in logit space in our implementation. During training, the log probability of $\mathbf{z}_n$ is only added to the ELBO for droplets that have been found to contain cells (that is, for droplets n where a sample of y_n is 1). The discrete latent variable y_n cannot be reparameterized, and so we use full enumeration over cell or no cell (y_n being 1 or 0) in our variational posterior to reduce variance. This is achieved using the TraceEnum_ELBO SVI objective available in Pyro. Integration over the continuous latent variables appearing in the ELBO (equation (4)) is done using a single Monte Carlo sample.

Training happens in random minibatches. Each full epoch trains on a fixed subset of barcodes from the dataset as well as a randomly sampled subset of empty droplet barcodes that changes each epoch. This is done to cover the tens of thousands of empty droplets without taking excessive computation time. The fraction of each minibatch that is composed of these randomly sampled empty droplets can be specified using a command-line argument (by default, we use 20%).

The training loop converges typically within about 150 epochs. For a typical 10x scRNA-seq experiment containing 5,000–30,000 cells, the total runtime of the tool ranges from around 20 min to 1 h using an NVIDIA Tesla T4 or K80 GPU, depending on the size of the dataset and chosen parameters. The stochastic optimizer used is a version of the Adam optimizer with gradient clipping. A OneCycle learning rate scheduler is used by default. Optimization proceeds for a pre-defined number of epochs, which can be set via command-line arguments. The default is 150 epochs, and the OneCycle scheduler increases the learning rate to 10× the user-defined '––learning-rate' at maximum. The default learning rate is 1×10^{-4}.

The tool saves checkpoints at user-defined intervals, which can be used to resume training or to create a new output with a different FPR. Checkpoints enable the use of cheaper pre-emptible cloud machines via the Terra platform (https://app.terra.bio). More generally, any workflow deployment using Cromwell (https://github.com/broadinstitute/cromwell) version 55+ can automatically benefit from this checkpointing functionality, so that a pre-empted workflow can pick up where it left off instead of starting from scratch.

Single-cell analysis workflow and cell quality-control details

Analysis workflows for single-cell data were carried out in SCANPY[17] version 1.9.1. We employed a rudimentary cell quality control after CellBender, that is, removing cells using percentile-based thresholds on UMI count, gene count and mitochondrial read fraction. UMAPs were created after (1) finding highly variable genes using the seurat_v3 algorithm implemented in SCANPY, (2) normalizing counts per cell, (3) log scaling counts, (4) scaling counts of 2,000 highly variable genes and (5) performing PC analysis on those scaled values for the highly variable genes. A nearest-neighbor graph was constructed with 20 neighbors based on cosine distance in PC space (top 25 PCs). Clustering was performed using the Leiden algorithm at the same resolution for both raw and post-CellBender data. Dataset-specific cell quality-control thresholds and the statistics of initial and final cell calls are as follows.

For the pbmc8k scRNA-seq dataset, we removed the top 5% of high-UMI-count droplets and the top 5% of high unique gene count droplets (to eliminate doublets) as well as the top 10% of high mitochondrial read fraction droplets and with no lower cutoff for the total number of genes per droplet. This left 7,515 cells remaining from an initial 8,903 droplets.

For the rat6k snRNA-seq dataset, we removed the top 15% of high-UMI-count droplets and the top 15% of high unique gene count droplets (to eliminate doublets) as well as the top 10% of high mitochondrial read fraction droplets and eliminated droplets with fewer than 100 genes. This left 5,868 cells remaining from an initial 10,445 droplets.

For the pbmc5k CITE-seq dataset, we removed the top 5% of high-UMI-count droplets and the top 5% of high unique gene count droplets (to eliminate doublets) as well as the top 10% of high mitochondrial read fraction droplets with a lower cutoff of 300 genes per droplet. This left 4,451 cells remaining from an initial 5,754 droplets.

For the hgmm12k scRNA-seq dataset, no cell quality control was performed before creating the hgmm12k result plots: all CellBender 'non-empty' droplets are included.

pbmc5k CITE-seq dataset quality control and normalization

For the plot in Fig. 6e, the following antibody features were omitted due to low correlation between antibody counts and mRNA counts per cluster in the raw data: CD34_TotalSeqB (also has very low mRNA counts), CD45RA_TotalSeqB and CD45RO_TotalSeqB (has poor correlation with *PTPRC* mRNA counts, which is understood given the high splicing specificity of *PTPRPC* in different immune subtypes, which make the expectation of having a linear correlation meaningless in principle), CD69_TotalSeqB, CD137_TotalSeqB, CD197_TotalSeqB, CD274_TotalSeqB, IgG1_control_TotalSeqB, IgG2a_control_TotalSeqB and IgG2b_control_TotalSeqB. Low correlation was defined as a slope of less than 1 for a fit using weighted ordinary least squares when plotting log1p antibody counts versus log1p mRNA counts. The following features were omitted due to low mRNA counts in the raw data: CD15_TotalSeqB, CD25_TotalSeqB, CD278_TotalSeqB and PD-1_TotalSeqB. Low mRNA counts were defined as the maximum mean expression value over all clusters being ≤0.2 counts. These features were left out for clarity of presentation (the scaling transformation, below, does not work well for those outliers), but the excluded features are all plotted in Supplementary Fig. 9a,b.

The scaling transformation used to plot data in Fig. 6e by collapsing all data onto a single line is as follows. The raw RNA expression data are x, while the raw antibody data are y:

$$\begin{aligned} x_{\text{rescaled}} &= \frac{x}{\text{std}(x)}, \\ y_{\text{intermediate}} &= \frac{y}{\text{std}(y)}, \\ m &= \frac{x_{\text{rescaled}}\, y_{\text{intermediate}}}{x_{\text{rescaled}}\, x_{\text{rescaled}}}, \\ y_{\text{rescaled}} &= \frac{y_{\text{intermediate}}}{m}. \end{aligned}$$

Simulated data generation

Data were simulated according to a model, which was slightly and intentionally mis-specified for CellBender's model, in that each cell within a cell type k is not given the exact same underlying expression profile $\chi_g^{(k)}$, but instead each cell has its expression profile drawn from a Dirichlet distribution with a common set of concentration parameters for each cell type, $\alpha_g^{(k)}$. Thus the data will be a bit overdispersed compared to CellBender's model. Details of the simulations are included in a notebook for code reproducibility, and the data-simulation function is included as part of the CellBender package. The simulator first samples the base gene expression profiles for k cell types, $\alpha_g^{(k)}$, from flat Dirichlet distributions, for example, $\alpha_g^{(0)} \sim \text{Dirichlet}(\alpha)$, $\alpha_g^{(1)} \sim \text{Dirichlet}(\alpha)$ and so on. These k cell type expression profiles are then optionally made to be artificially similar to $\alpha_g^{(0)}$ via a parameter η by applying the transformation $\alpha_g^{(k)} \leftarrow (1-\eta)\,\alpha_g^{(k)} + \eta\,\alpha_g^{(0)}$. Note that this transformation is not symmetric, and so the different simulated cell types will have different gene expression complexity (in particular, in the simulation shown in Fig. 5f, cell type 2 has many more unique genes with relatively lower expression rates than cell type 1). Next, the ambient profile χ_g^{a} is set to the (normalized) average of $\alpha_g^{(k)}$, weighted by the number of simulated cells of each type and the average UMI per cell

type. Finally, for a given cell type k with n cells, the simulation proceeds as

$$\begin{aligned}
\chi_{ng}^{(k)} &\sim \text{Dirichlet}\left(\alpha_g^{(k)}\right), \quad \text{for all cells } n \\
\epsilon_n &\sim \text{Gamma}\ (\epsilon_\alpha, \epsilon_\alpha) \\
d_n &\sim \text{lognormal}\ (d_\mu, d_\sigma) \\
d_n^{\text{empty}} &\sim \text{lognormal}\left(d_\mu^{\text{empty}}, d_\sigma^{\text{empty}}\right) \\
\rho_n &\sim \text{Beta}\ (\rho_\alpha, \rho_\beta) \\
y_n &= 1 \text{ if cell, otherwise } 0 \\
\mu_{ng}^{(k)} &= (1-\rho_n) y_n\, \epsilon_n\, d_n\, \chi_{ng}^{(k)} \\
\lambda_{ng} &= \epsilon_n \left[(1-\rho_n)\, d_n^{\text{empty}} \chi_g^{\text{a}} + \rho_n \bar{\chi}_g \left(y_n d_n + d_n^{\text{empty}}\right)\right] \\
c_{ng}^{\text{cell}\,(k)} &\sim \text{NegBinom}\ (\mu_{ng}, \Phi) \\
c_{ng}^{\text{noise}} &\sim \text{Poisson}\ (\lambda_{ng}) \\
c_{ng}^{(k)} &= c_{ng}^{\text{cell}(k)} + c_{ng}^{\text{noise}}
\end{aligned} \quad , \qquad (42)$$

where the simulated counts for cell type k are $c_{ng}^{(k)}$, and $\bar{\chi}_g$ is the same as χ_g^{a} in these simulations, and all the other variables not specified above are hyperparameter inputs to the simulation. Cells are simulated with $y_n = 1$, and empty droplets are obtained by setting $y_n = 0$. Cell counts are simulated, one cell type k at a time, followed by empty droplets, to obtain a full dataset.

Generation of the rat6k dataset

Animal experiments were approved by the Institutional Animal Care and Use Committee at the Broad Institute. An individual 17-week-old male Wistar rat (Charles River) was acclimated for 2–3 weeks to the Broad vivarium, with ad libitum access to water and chow diet. The left atrial section of the heart was flash frozen in liquid nitrogen and stored at −80 °C until use. Frozen tissue was mounted on OCT and sectioned. The tissue section was then processed for nuclear isolation. An input of 7,000 nuclei (5,000 calculated recovery) was used for droplet generation and library construction according to the manufacturer's protocol (10x Genomics, single-cell 3′ version 2 chemistry) with minor modifications. Sequencing was performed on an Illumina NextSeq 550 in the Broad Institute's Genomics Platform (https://genomics.broadinstitute.org). BCL files were processed using CellRanger version 3.0 software to obtain a count matrix.

Software

CellBender remove-background inputs. The current version of CellBender remove-background (version 0.3.0_rc) takes the following file formats as input: (1) raw HDF5 files from 10x Genomics' CellRanger version 2+ count pipeline, (2) raw MTX files, with accompanying TSV files, in CellRanger format, (3) raw DropSeq DGE files, (4) H5AD files in AnnData format[17], (5) raw BD Rhapsody CSV files, and (6) Loom files readable by AnnData. Ensure that empty droplets are included in the file. The AnnData, DropSeq DGE and CellRanger MTX formats are particularly general, and data from other sources can be massaged into one of those formats.

CellBender remove-background outputs. The output of CellBender remove-background provides several useful quantities: (1) an inferred background-subtracted count matrix, (2) the probability that each droplet contains a cell, (3) a low-dimensional latent representation of gene expression for each cell and (4) the ambient profile, among other latent variables.

There is an input parameter, '−−fpr', which controls the expected nFPR, where a false positive is a real count that has erroneously been identified as background and removed. Setting nFPR to 0.01 means that the algorithm will remove as much noise as possible while controlling the expected removal of real signal to ~1% above the estimated dataset-wide noise level. It is to be understood that this constraint is enforced in expectation and is approximate: assuming that the model fits the data perfectly (no model mis-specification), the estimate will be correct. There is an inherent tradeoff in noise reduction in which the removal of more noise comes at the expense of the removal of more signal. The nFPR parameter allows the user to control this tradeoff. Multiple FPR inputs will result in multiple output count matrices. Because we marginalize over c_{ng}^{noise} and c_{ng}^{cell} during training, constructing the output c_{ng}^{cell} at a given nFPR is a several-step process and is detailed in Supplementary Section 5.4.

The probability that each droplet contains a cell is given by q_n, the latent variable encoded by NN_y. The low-dimensional latent representation of gene expression is given by the encoded $\mathbf{z}_{n;\mu}$ for each cell. Furthermore, the ambient RNA profile is inferred as χ_g^{a}. By default, CellBender remove-background creates an HTML output report, showing several diagnostics including progress of the inference procedure and salient changes in the output count matrix, making recommendations and issuing warnings as necessary.

Reporting summary

Further information on research design is available in the Nature Portfolio Reporting Summary linked to this article.

Data availability

The datasets used in this study are the following: pbmc8k (the publicly available pbmc8k dataset from 10x Genomics called '8k PBMCs from a healthy donor', run with version 2 chemistry and analyzed with CellRanger version 2.1.0, available at https://www.10xgenomics.com/resources/datasets/8-k-pbm-cs-from-a-healthy-donor-2-standard-2-1-0); heart600k (the published dataset from the Broad–Bayer PCL called 'Single-nuclei profiling of human dilated and hypertrophic cardiomyopathy' (ref. 23), run with 10x Genomics 3′ capture version 3 chemistry and analyzed with CellRanger version 4.0.0, available at https://singlecell.broadinstitute.org/single_cell/study/SCP1303); hgmm12k (the publicly available hgmm12k dataset from 10x Genomics called '12k 1:1 Mixture of Fresh Frozen Human (HEK293T) and Mouse (NIH3T3) Cells', run with version 2 chemistry and analyzed with CellRanger version 2.1.0, available at https://www.10xgenomics.com/resources/datasets/12-k-1-1-mixture-of-fresh-frozen-human-hek-293-t-and-mouse-nih-3-t-3-cells-2-standard-2-1-0); pbmc5k (the publicly available pbmc5k dataset with antibodies from 10x Genomics called '5k Peripheral Blood Mononuclear Cells (PBMCs) from a Healthy Donor with a Panel of TotalSeq™-B Antibodies (Next GEM)', run with version 3 Next GEM chemistry and analyzed with CellRanger version 3.1.0, available at https://www.10xgenomics.com/resources/datasets/5-k-peripheral-blood-mononuclear-cells-pbm-cs-from-a-healthy-donor-with-cell-surface-proteins-next-gem-3-1-standard-3-1-0); and rat6k (an snRNA-seq dataset from a healthy Wistar rat left atrium, comprising approximately 6,000 nuclei, processed on the 10x Genomics platform using version 2 chemistry and analyzed with CellRanger version 3.1.0. The dataset was provided by P.T.E.'s group at the Broad Institute as part of the Broad–Bayer PCL. The experiment was performed by A.A. and A.-D.A. The dataset is publicly available on Broad's Single Cell Portal at https://singlecell.broadinstitute.org/single_cell/study/SCP2148). Datasets analyzed only in the Supplementary Information are as follows: smartseq3xpress_pbmc (a Smart-seq3xpress (well-based) scRNA-seq dataset from healthy human PBMCs called 'Scalable full-transcript coverage single-cell RNA sequencing of PBMCs using Smart-seq3xpress' and published by Hagemann-Jensen et al.[59]. This dataset was kindly provided to the authors in count matrix format by C. Ziegenhain, an author of the referenced paper. We subsetted the data to the 16 384-well plates that came from 'donor8' and fluentbio_pbmc (the publicly available scRNA-seq dataset of healthy human PBMCs from Fluent BioSciences called 'Profiling 20k Immune Cells in Healthy PBMCs from a Single

T20 Reaction', generated with T20 PIPseq and analyzed with PIPseeker version 1.1.3 by Fluent Biosciences[60], available at https://fbs-public.s3.us-east-2.amazonaws.com/public-datasets/pbmc/raw_matrix.tar.gz).

Code availability

CellBender can be obtained from https://github.com/broadinstitute/CellBender. Additional documentation is available at https://cellbender.readthedocs.io. CellBender modules are also available as workflows on Terra (https://app.terra.bio), a secure open platform for collaborative omic analysis, and can be run on the cloud with zero set-up. We have implemented the model and the inference method using Pyro probabilistic programming language[16] and PyTorch[61] and presented it as a user-friendly, production-grade and stand-alone command-line tool. We refer to the background noise-removal algorithm implemented in CellBender as remove-background.

References

51. van Dijk, D. et al. Recovering gene interactions from single-cell data using data diffusion. *Cell* **174**, 716-729 (2018).
52. Svensson, V. Droplet scRNA-seq is not zero-inflated. *Nat. Biotechnol.* **38**, 147–150 (2020).
53. Jiang, R., Sun, T., Song, D. & Li, J. J. Statistics or biology: the zero-inflation controversy about scRNA-seq data. *Genome Biol.* **23**, 31 (2022).
54. Hoffman, M., Blei, D. M., Wang, C. & Paisley, J. Stochastic variational inference. Preprint at https://doi.org/10.48550/arXiv.1206.7051 (2012).
55. Blei, D. M., Kucukelbir, A. & McAuliffe, J. D. Variational inference: a review for statisticians. *J. Am. Stat. Assoc.* **112**, 859–877 (2017).
56. Law, C. W., Chen, Y., Shi, W. & Smyth, G. K. voom: precision weights unlock linear model analysis tools for RNA-seq read counts. *Genome Biol.* **15**, R29 (2014).
57. Stuart, T. et al. Comprehensive integration of single-cell data. *Cell* **177**, 1888–1902 (2019).
58. Ganchev, K., Graça, J., Gillenwater, J. & Taskar, B. Posterior regularization for structured latent variable models. *J. Mach. Learn. Res.* **11**, 2001–2049 (2010).
59. Hagemann-Jensen, M., Ziegenhain, C. & Sandberg, R. Scalable single-cell RNA sequencing from full transcripts with Smart-seq3xpress. *Nat. Biotechnol.* **40**, 1452–1457 (2022).
60. Clark, I. C. et al. Microfluidics-free single-cell genomics with templated emulsification. *Nat. Biotechnol.* https://doi.org/10.1038/s41587-023-01685-z (2023).
61. Paszke, A. et al. PyTorch: an imperative style, high-performance deep learning library. In *33rd Conference on Neural Information Processing Systems* 12 (NeurIPS, 2019).

Acknowledgements

We thank L. D'Alessio, C. Roselli, C. Porter, E. Bingham, F. Obermeyer, J. Nemesh, B. Wang, B. Babadi, V. Popic, A. Wysoker, A. Subramanian, N. Tucker, Y. Farjoun, T. Tickle and A. Carr for insightful discussions at various stages of this project. S.J.F., M.D.C. and M.B. acknowledge financial support from the Broad–Bayer PCL. M.B. acknowledges additional support from the SPARC grant 'Development of Production-Grade Computational Methods for Single-Cell Genomics' from the Broad Institute. The publicly available rat6k snRNA-seq dataset was generated by the PCL, and the experiment was performed by A.A. and A.-D.A. We additionally thank C. Ziegenhain for providing a count matrix for the published Smart-seq3xpress PBMC dataset analyzed in Supplementary Section 2.4.

Author contributions

S.J.F. and M.B. jointly developed the probabilistic model, software and study design and jointly wrote the paper. S.J.F. additionally performed statistical analyses on real and simulated data. A.A. and A.-D.A. collected the rat6k dataset under the supervision of P.T.E. M.D.C. provided critical feedback at various stages of the project and analyzed the heart600k dataset. M.B. and P.T.E. jointly supervised the project, with additional input from A.A.P., J.C.M. and E.B.

Competing interests

A.-D.A. is an employee of Bayer US LLC (a subsidiary of Bayer AG) and may own stock in Bayer AG. A.A.P. is employed as a venture partner at Google Ventures, and he is also supported by a grant from Bayer AG to the Broad Institute focused on machine learning for clinical trial design. P.T.E. is supported by a grant from Bayer AG to the Broad Institute focused on the genetics and therapeutics of cardiovascular diseases. P.T.E. has also served on advisory boards or consulted for Bayer AG, Quest Diagnostics, MyoKardia and Novartis. The other authors declare no competing interests.

Additional information

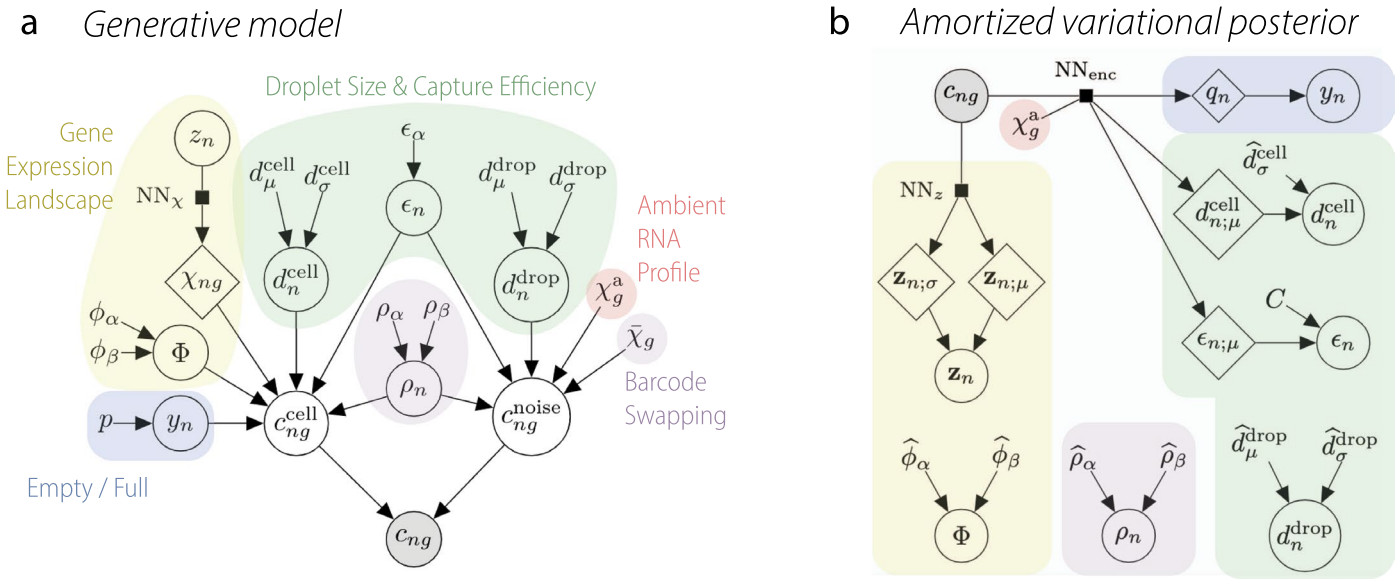
Extended data is available for this paper at https://doi.org/10.1038/s41592-023-01943-7.

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41592-023-01943-7.

Correspondence and requests for materials should be addressed to Stephen J. Fleming or Mehrtash Babadi.

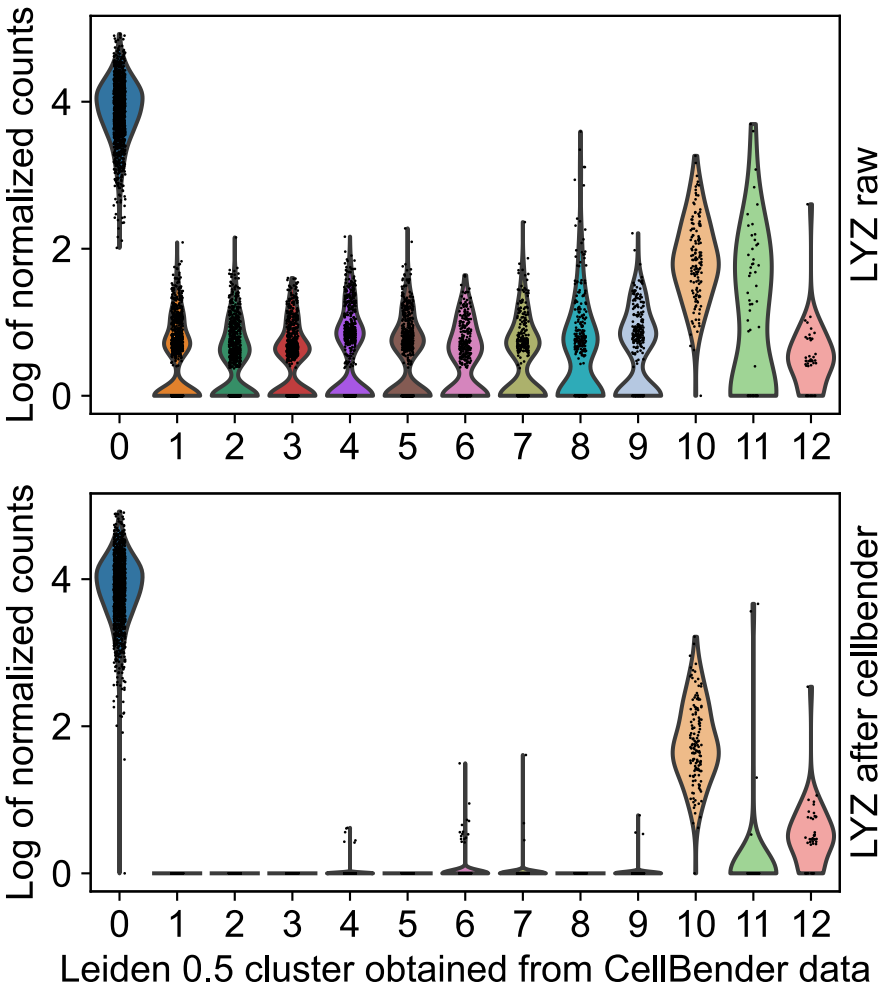
Peer review information *Nature Methods* thanks Eran Mukamel and the other, anonymous, reviewer(s) for their contribution to the peer review of this work. Peer reviewer reports are available. Primary Handling Editors: Lei Tang and Lin Tang, in collaboration with the *Nature Methods* team.

https://doi.org/10.1038/s41592-023-01943-7



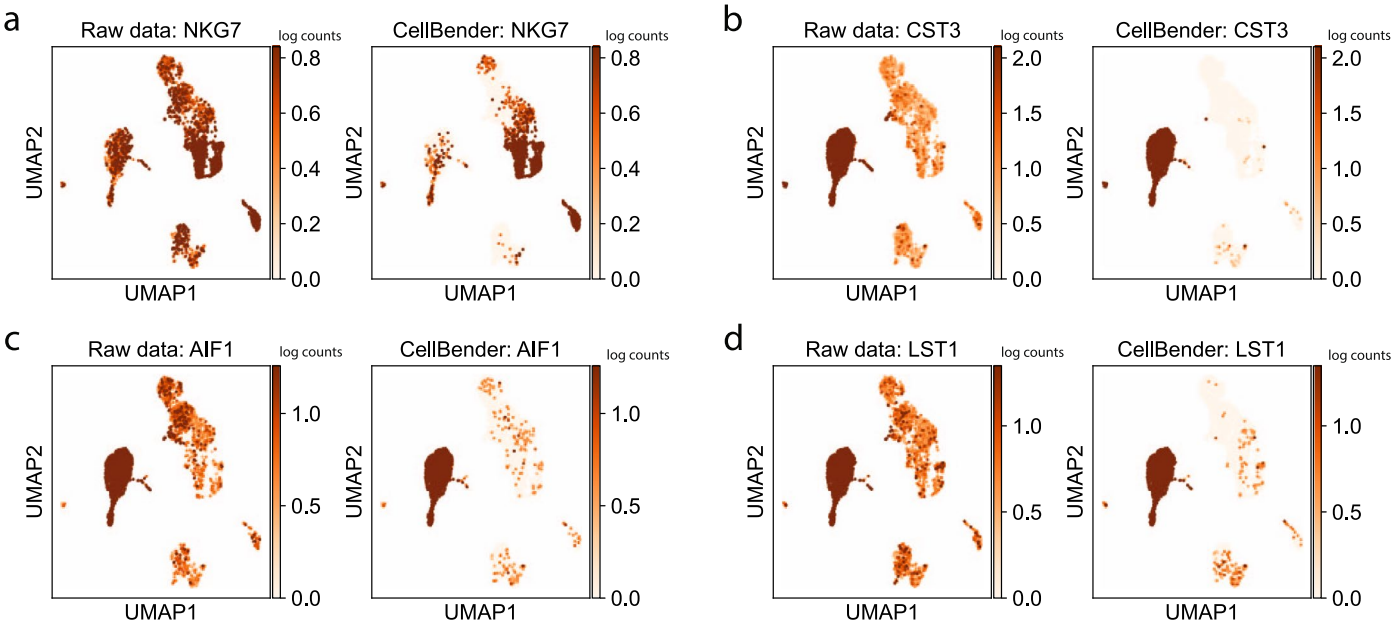
Extended Data Fig. 1 | The `CellBender` model. (**a**) The `CellBender` generative model for noisy single-cell count data. (**b**) The variational posterior used by `CellBender`. The neural network $\mathrm{NN}_{\mathrm{enc}}$ takes the observed data as input and yields the parameters of various variational distributions assumed for the local latent variables. The global latent variables are treated in the usual mean-field approximation.

https://doi.org/10.1038/s41592-023-01943-7



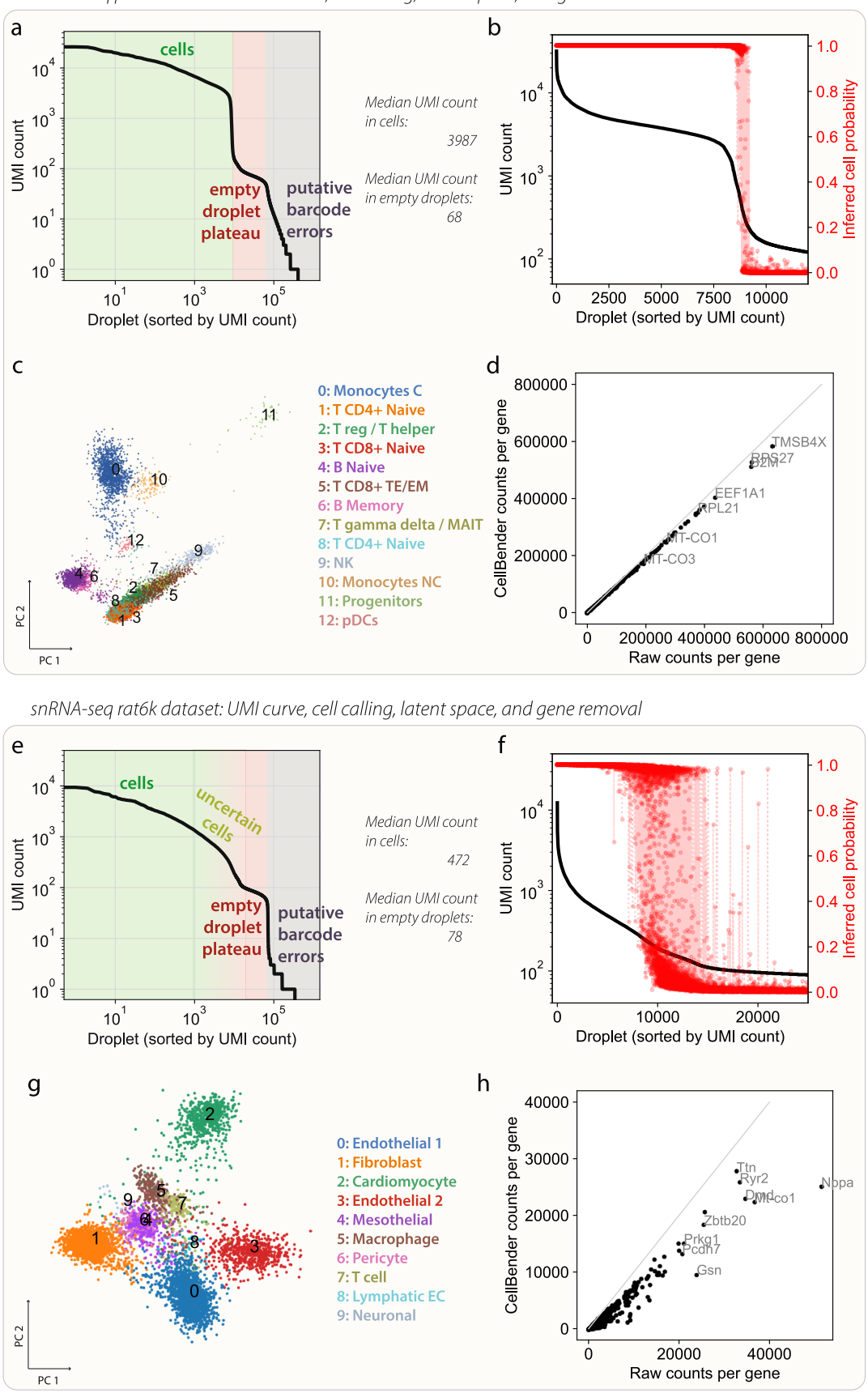
Extended Data Fig. 2 | Violin plots showing the count distributions of lysozyme, *LYZ*, per cluster before and after `CellBender` denoising. (nFPR was 0.01.) The off-target counts are effectively removed, with counts remaining in clusters 0 (CD14$^+$ monocytes C), 10 (FCGR3A$^+$ monocytes NC), and 12 (plasmacytoid dendritic cells).

https://doi.org/10.1038/s41592-023-01943-7



Extended Data Fig. 3 | UMAPs created from the `CellBender`-analyzed `pbmc8k` data, showing increased expression specificity of marker genes for different cell types after `CellBender` denoising as compared to the raw data. a–d, UMAP plots of the expression of *NKG7*, *CST3*, *AIF1* and *LST1* in each cell before and after CellBender.

https://doi.org/10.1038/s41592-023-01943-7

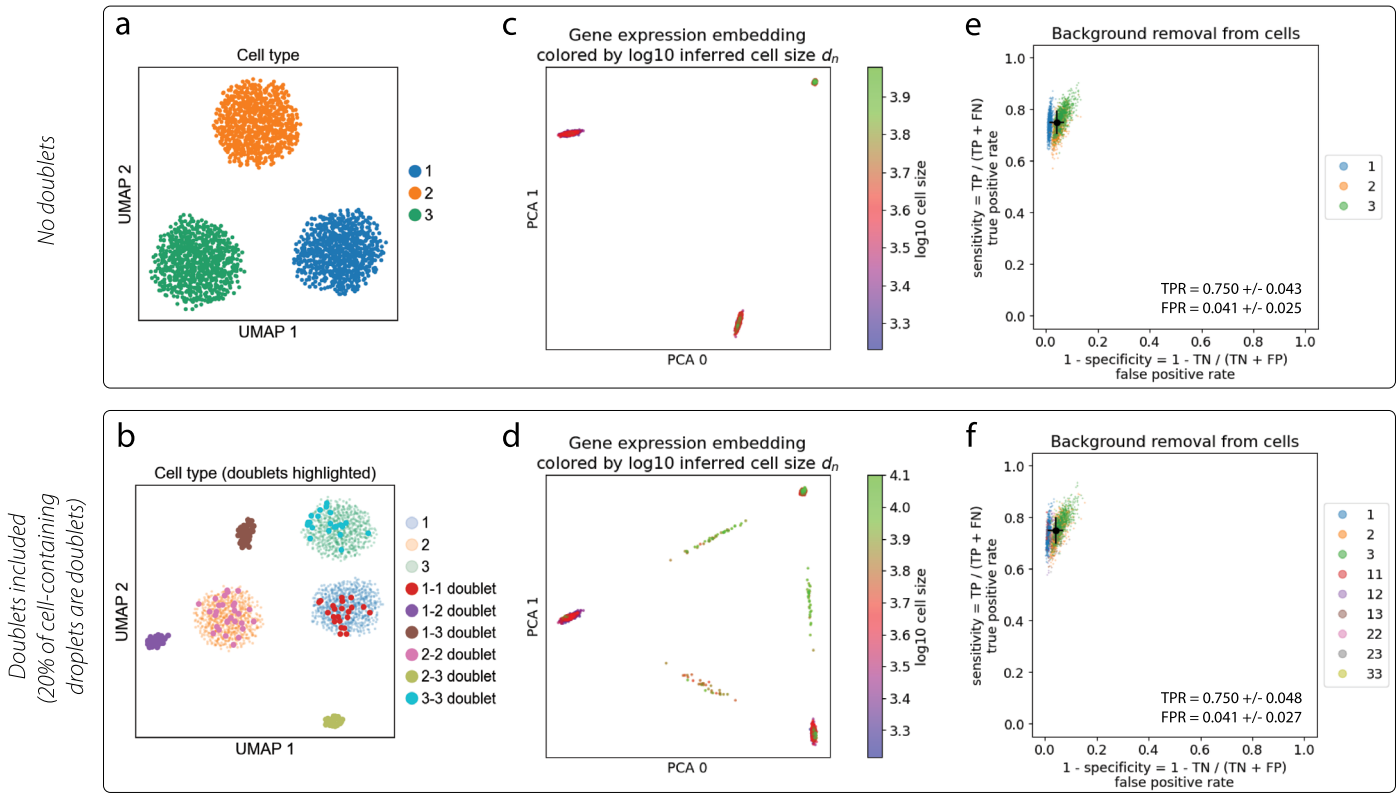


Extended Data Fig. 4 | See next page for caption.

https://doi.org/10.1038/s41592-023-01943-7

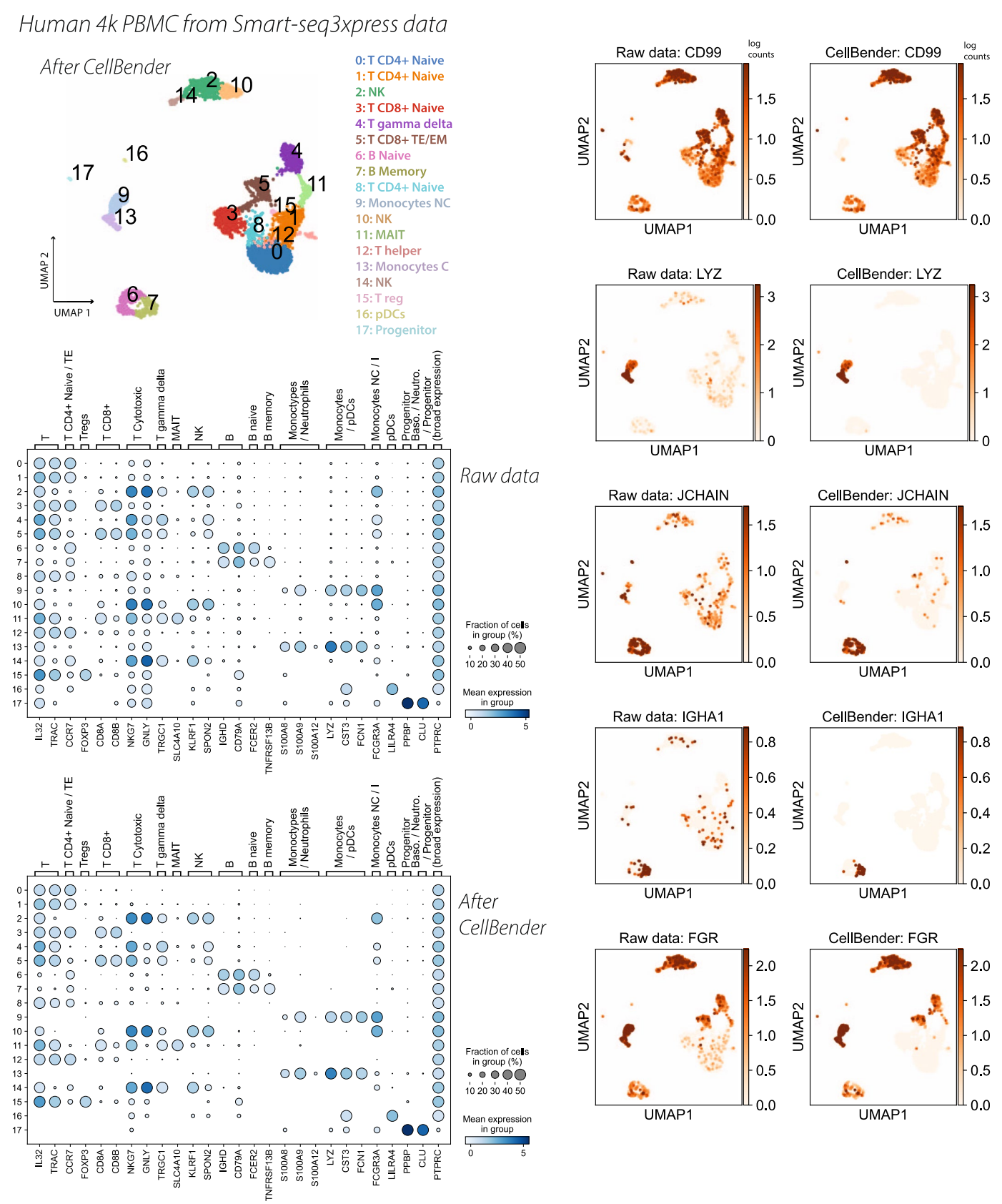
Extended Data Fig. 4 | UMI curves from the raw data together with various `CellBender` outputs for the `pbmc8k` and `rat6k` datasets. (**a**-**d**) `pbmc8k`, and (**e**-**h**) `rat6k`. (**a**,**e**) The raw UMI curves, annotated with areas of cells and empty droplets. Notably, the distinction is much more difficult in (**e**), the nuclei dataset extracted from heart tissue. (**b**,**f**) Cells probabilities inferred by `CellBender` on same UMI curves from (**a**,**e**) respectively. The region of transition from "surely-cell" to "surely-empty" is much broader in the snRNA-seq dataset. (**c**,**g**) First two principal components of the latent gene expression embedding inferred by `CellBender`, colored by Leiden clustering from a separate scanpy analysis. The structure very closely reflects the labels attributed by that separate analysis. (**d**,**h**) Scatter plots showing removal of each gene by `CellBender` (each dot is a gene, *MALAT1* is off-scale). Several top denoised genes are indicated.

https://doi.org/10.1038/s41592-023-01943-7



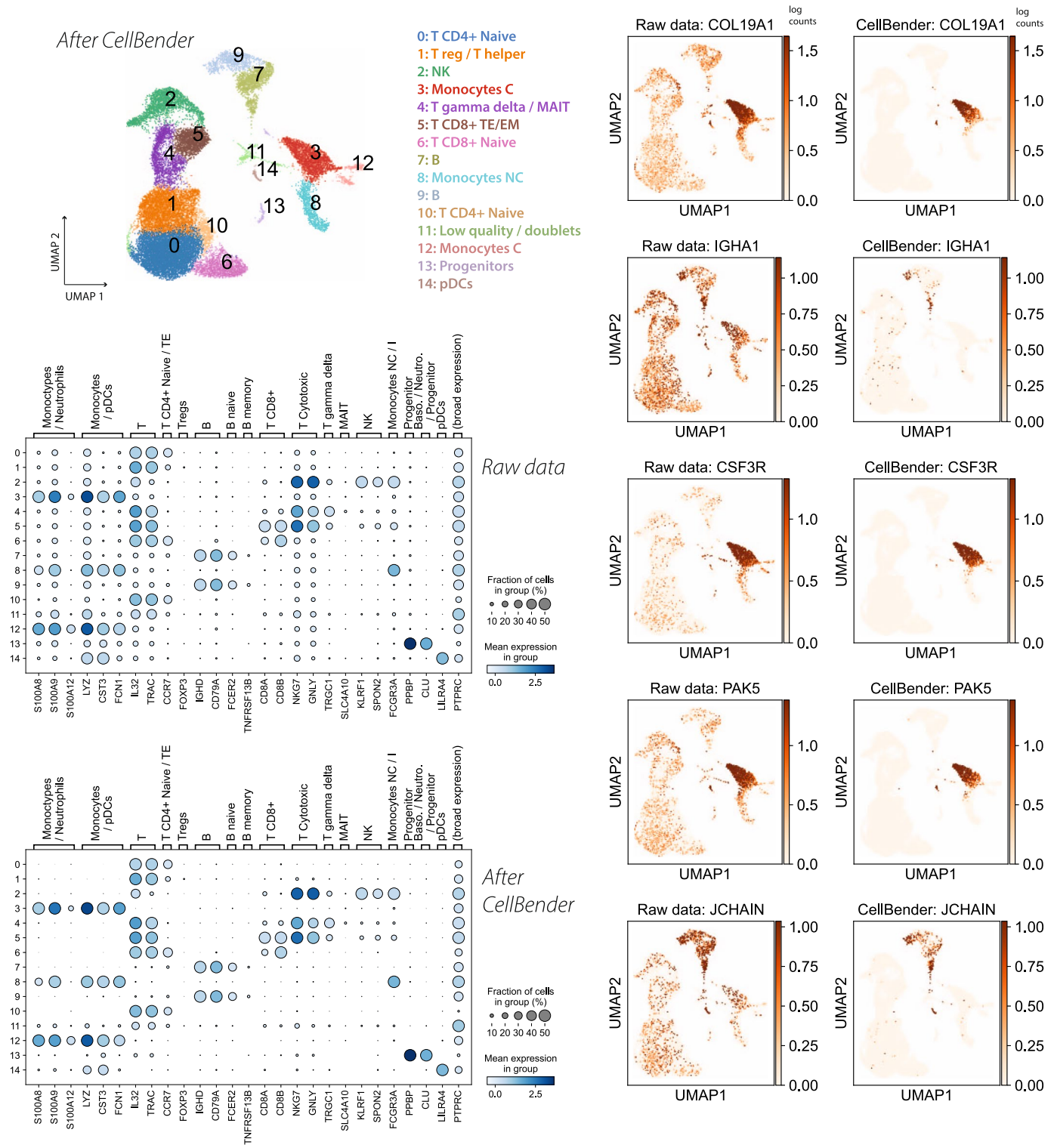
Extended Data Fig. 5 | Presence of doublets does not impact the denoising performance of `CellBender`. (**a**,**c**,**e**) Simulated dataset without doublets. (**b**,**d**,**f**) Simulated dataset where 20% of the cell-containing droplets are doublets. (**a**) UMAP of the gene expression profile of the three simulated cell types. (**b**) Same as (**a**), but including doublets, which are highlighted in bold color. Doublets with cells of two different types form their own clusters in UMAP space, due to their unique transcriptional profile. (**c**) The learned `CellBender` prior on gene expression, visualized via PCA, shows three clusters for the three cell types. (**d**) With doublets present, the prior on gene expression now additionally contains clusters for each type of doublet. From the standpoint of `CellBender`, a doublet is like a unique cell type. (**e**,**f**) Denoising performance has been quantified using a ROC curve, and shows that denoising metrics are nearly identical (TPR 0.750, FPR 0.041) whether doublets are present or not. The error bars shown in panels **e**-**f** correspond to the interquartile range of TPR (vertical) and FPR (horizontal) over N=2400 simulated cells.

https://doi.org/10.1038/s41592-023-01943-7



Extended Data Fig. 6 | Published human scRNA-seq PBMC dataset from the well-based Smart-seq3xpress protocol[59]. This dataset is extremely clean to begin with. The UMAP shows the expected cell types, nicely clustered. The two dotplots show expression of immune cell marker genes before and after `CellBender`. Some genes show improvement, but many look quite similar, as expected for a clean dataset. UMAP plots on the right show cleanup of a few genes after `CellBender`.

https://doi.org/10.1038/s41592-023-01943-7



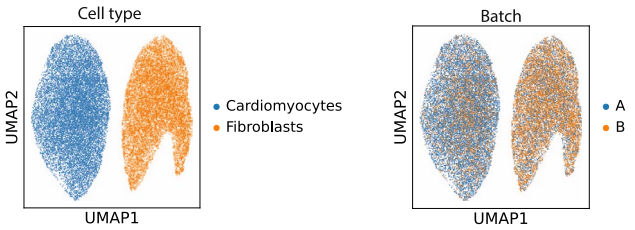
Extended Data Fig. 7 | Publicly available human scRNA-seq PBMC dataset from the Fluent Biosciences PIPseq platform[60]. Droplets are generated by vigorous vortexing, and thus we expect more ambient RNA than a microfluidics experiment. The UMAP shows the expected cell types, in addition to some probable doublets. The two dotplots show expression of immune cell marker genes before and after `CellBender`. Many genes show significant cleanup. UMAP plots on the right show rather marked cleanup of a few genes after `CellBender`.

https://doi.org/10.1038/s41592-023-01943-7

a

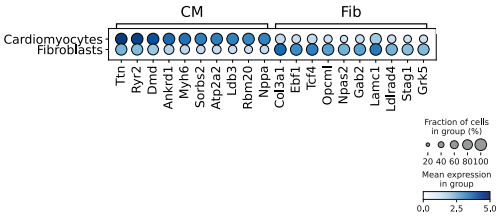
Six simulated datasets: Only difference between batch A and batch B is the number of cardiomyocytes

sample	Cardiomyocytes	Fibroblasts
A1	4000	2000
A2	4000	2000
A3	4000	2000
B1	1000	2000
B2	1000	2000
B3	1000	2000

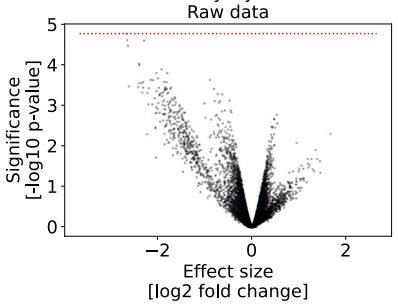


Raw data

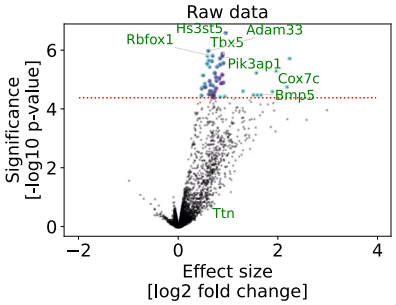
b



c

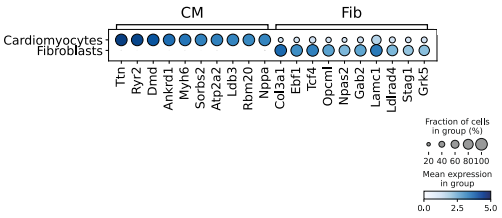


d

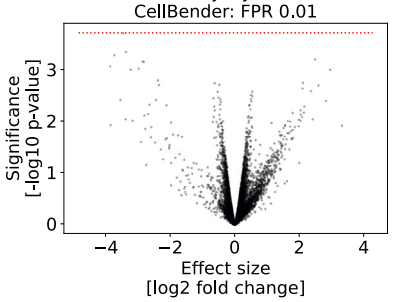


CellBender: FPR 0.01

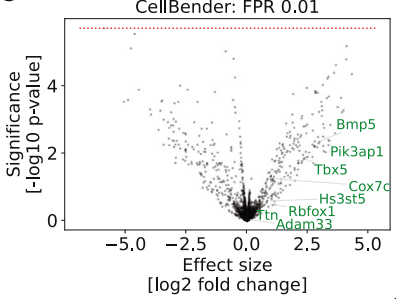
e



f

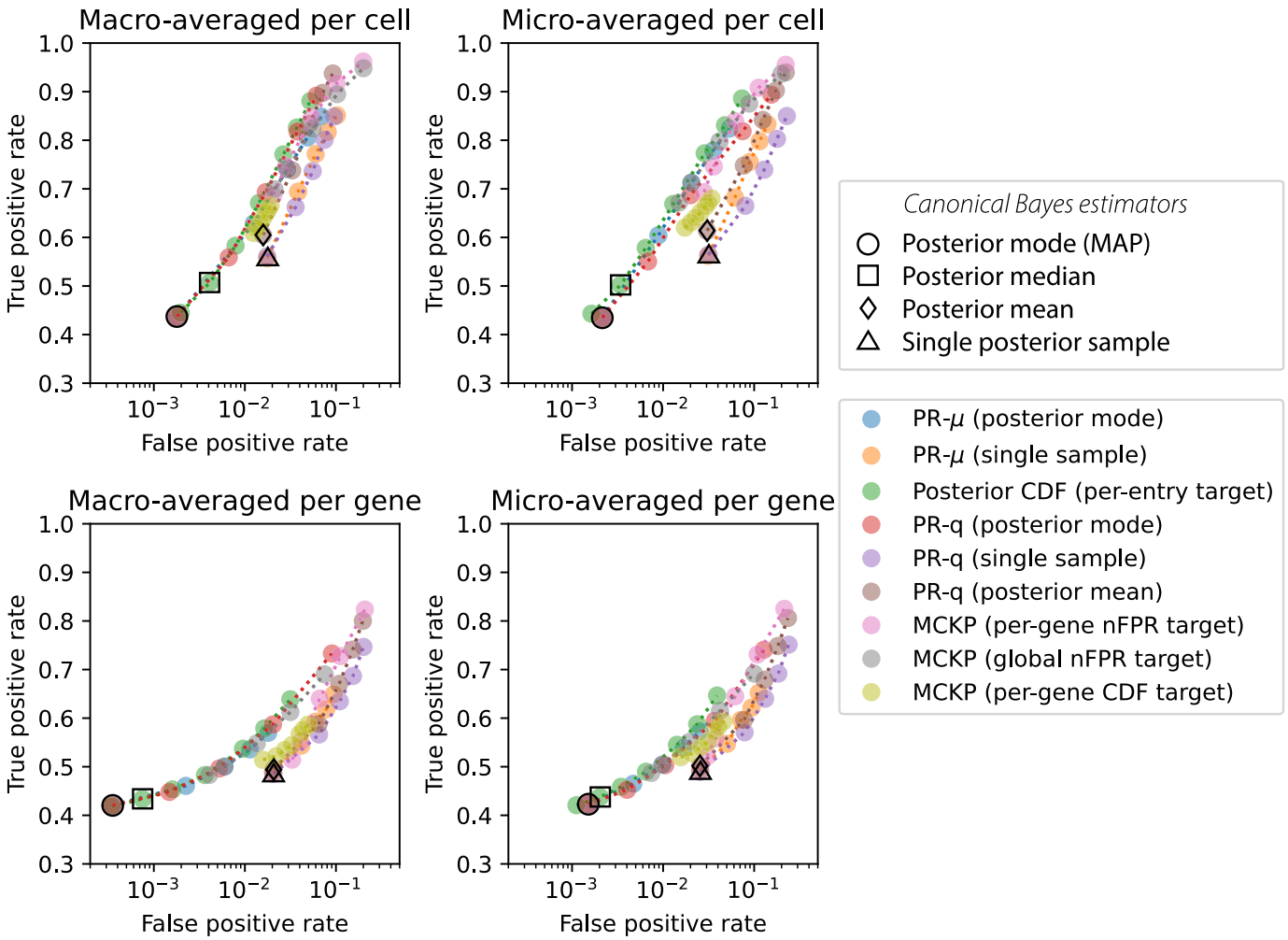


g



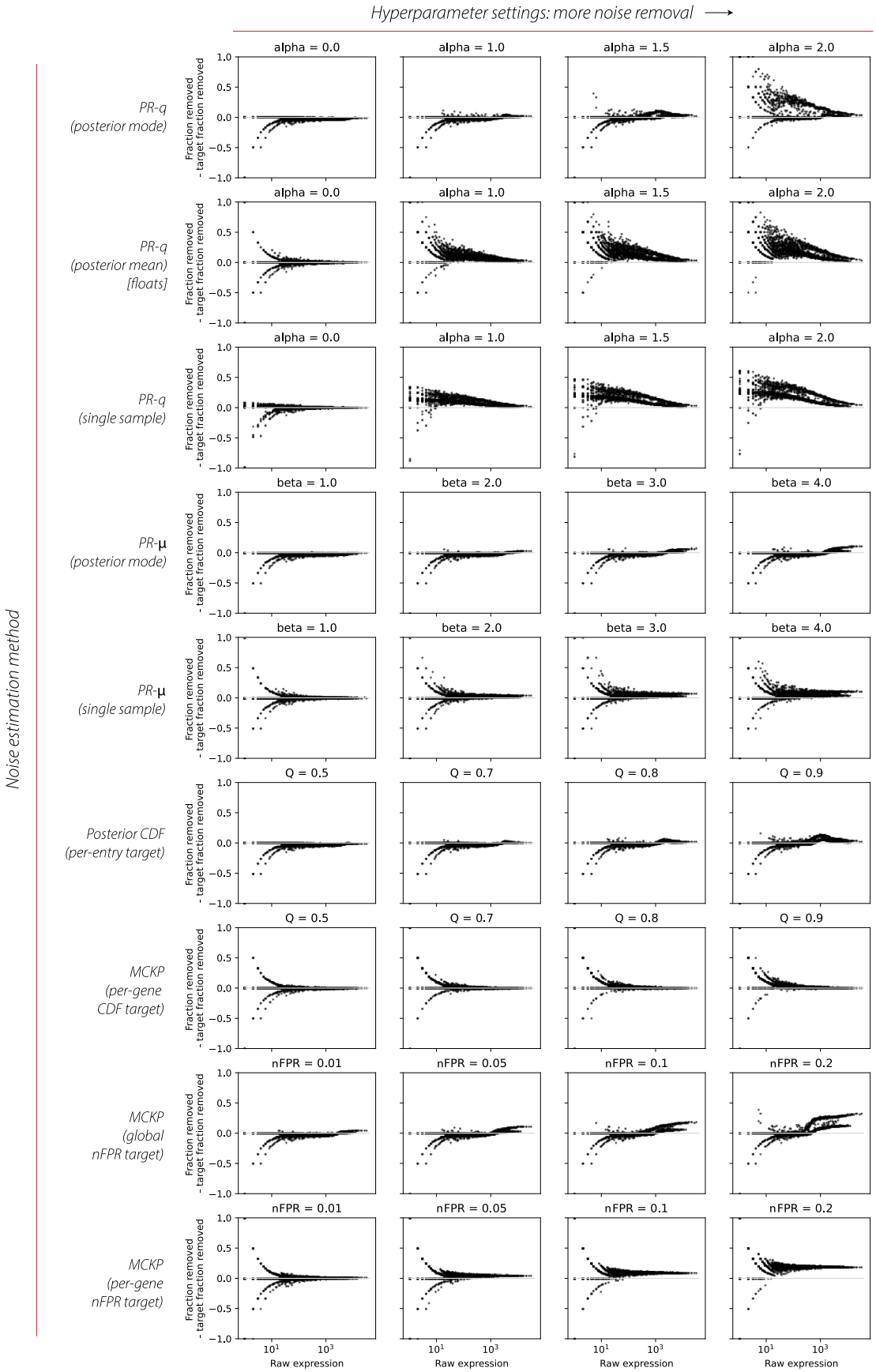
Extended Data Fig. 8 | Systematic background noise as a source of batch variation and spurious differential expression across batches. (**a**) Setup of the cohort of simulated datasets, where there are two cell types whose expression profiles are taken from real data (`rat6k`) for cardiomyocytes and fibroblasts. The only difference between simulations from batch A and batch B is the number of cardiomyocytes. Noise ends up being different in the two batches due to these cell number differences. The "truth" in this simulated cohort is that there are no differences between a cell type's expression profile between batches. (**b**-**d**) Raw data. (**e**-**g**) `CellBender` denoised data. (**b**) Dotplot showing top cardiomyocyte and fibroblast marker genes. Background noise causes marker genes to show up in the off-target cell type at a low level. (**e**) Marked cleanup of the dataset at an aggregate level. (**c**,**f**) The cardiomyocytes show no differentially expressed genes between batch A and B, before or after `CellBender`. (**d**) In the raw data, many genes show up as being significantly differentially-expressed due to background noise. (**g**) After `CellBender`, these spurious results have disappeared (a few of which are labeled). Benjamini-Hochberg-corrected FDR value for significance (red dotted line) is 0.01 in all volcano plots.

https://doi.org/10.1038/s41592-023-01943-7



Extended Data Fig. 9 | Comparison of output summarization methods for constructing an integer count matrix. Methods are discussed in Supplementary Sections 5.5 (legend label MCKP), 5.6 (legend label Posteior CDF), and 5.7 (legend labels PR-μ and PR-q). The four panels show four different ways to compute TPR and FPR to display a ROC curve. "Macro-averaged per cell" computes TPR as $(\sum_g \mathrm{TP}_{ng})/(\sum_g \mathrm{TP}_{ng} + \mathrm{FN}_{ng})$, while "micro-averaged per cell" computes TPR as $\sum_g [\mathrm{TP}_{ng}/(\mathrm{TP}_{ng} + \mathrm{FN}_{ng})]$. For the "per gene" cases, the sum over genes is replaced by a sum over cells. We exclude genes whose raw data counts are less than 10 summed over all cells. The dots shown represent the mean over all cells or genes as appropriate.

https://doi.org/10.1038/s41592-023-01943-7



Extended Data Fig. 10 | See next page for caption.

https://doi.org/10.1038/s41592-023-01943-7

Extended Data Fig. 10 | Comparison of per-gene performance of different noise estimation methods. Methods are discussed in Supplementary Sections 4.5 (MCKP), 4.6 (Posterior CDF), and 4.7 (PR-μ and PR-q). Each plot shows the over-removal of each gene (fraction removed - fraction that should have been removed according to truth) for the given method with the hyperparameter setting specified in the title. Each dot is a gene. Positive values indicate that too many counts of the gene were removed at the level of the entire experiment. Row 1 column 1 shows the posterior mode, row 2 column 1 shows the posterior mean, and row 3 column 1 shows a single sample from the unregularized posterior ($\alpha = 0$).

nature research

Corresponding author(s): Mehrtash Babadi
Stephen J. Fleming

Last updated by author(s): Mar 20, 2023

Reporting Summary

Nature Research wishes to improve the reproducibility of the work that we publish. This form provides structure for consistency and transparency in reporting. For further information on Nature Research policies, see our Editorial Policies and the Editorial Policy Checklist.

Statistics

For all statistical analyses, confirm that the following items are present in the figure legend, table legend, main text, or Methods section.

n/a	Confirmed
☐	☒ The exact sample size (*n*) for each experimental group/condition, given as a discrete number and unit of measurement
☒	☐ A statement on whether measurements were taken from distinct samples or whether the same sample was measured repeatedly
☐	☒ The statistical test(s) used AND whether they are one- or two-sided *Only common tests should be described solely by name; describe more complex techniques in the Methods section.*
☐	☒ A description of all covariates tested
☐	☒ A description of any assumptions or corrections, such as tests of normality and adjustment for multiple comparisons
☐	☒ A full description of the statistical parameters including central tendency (e.g. means) or other basic estimates (e.g. regression coefficient) AND variation (e.g. standard deviation) or associated estimates of uncertainty (e.g. confidence intervals)
☒	☐ For null hypothesis testing, the test statistic (e.g. *F*, *t*, *r*) with confidence intervals, effect sizes, degrees of freedom and *P* value noted *Give P values as exact values whenever suitable.*
☐	☒ For Bayesian analysis, information on the choice of priors and Markov chain Monte Carlo settings
☐	☒ For hierarchical and complex designs, identification of the appropriate level for tests and full reporting of outcomes
☐	☒ Estimates of effect sizes (e.g. Cohen's *d*, Pearson's *r*), indicating how they were calculated

Our web collection on statistics for biologists contains articles on many of the points above.

Software and code

Policy information about availability of computer code

Data collection	All of the data used in this study (public domain and generated by us) are collected using 10x Genomics Single-Cell Gene Expression products and softwares. The chemistry version and software versions are fully detailed in Supplementary Section S.4 (Data Availability).
Data analysis	CellBender v0.3.0 (this work); Pyro v1.8.1; PyTorch v1.11.0; Scanpy v1.9.1; Harmony-pytorch v0.1.7; Limma (voom) v3.48.3; EmptyDrops (Bioconductor 3.9); dropkick v1.2.6; 10x Genomics Cell Ranger v2.1.1 and v3.1.0;

For manuscripts utilizing custom algorithms or software that are central to the research but not yet described in published literature, software must be made available to editors and reviewers. We strongly encourage code deposition in a community repository (e.g. GitHub). See the Nature Research guidelines for submitting code & software for further information.

Data

Policy information about availability of data

All manuscripts must include a data availability statement. This statement should provide the following information, where applicable:

- Accession codes, unique identifiers, or web links for publicly available datasets
- A list of figures that have associated raw data
- A description of any restrictions on data availability

The data used in this study includes both public domain data and data generated by us. Data availability and accession numbers and links are detailed in Supplementary Section S.4 (Data Availability). The dataset referred to as "rat6k" is previously unpublished and will be deposited to Broad Institute Single Cell Portal for unrestricted public access before publication.

Field-specific reporting

Please select the one below that is the best fit for your research. If you are not sure, read the appropriate sections before making your selection.

☒ Life sciences ☐ Behavioural & social sciences ☐ Ecological, evolutionary & environmental sciences

For a reference copy of the document with all sections, see nature.com/documents/nr-reporting-summary-flat.pdf

Life sciences study design

All studies must disclose on these points even when the disclosure is negative.

Sample size	We evaluate our computational method on n=5 single-cell RNA sequencing datasets. Different datasets contain different number of assayed cells (not determined by us a priori) and is reported in Supplementary Sec. S.4. The n=5 samples are chosen to represent a diverse range of tissue types and protocols and each is analyzed separately. Datasets were chosen to illustrate applicability of CellBender to different tissue types and data modalities. No hypothesis is tested across the n=5 samples (not applicable), n=5 was an arbitrary choice (due to space constraints).
Data exclusions	Some of the protein-mRNA pairs were excluded in a specific CITE-seq analysis (Fig. 5e) that seeks to test the linear relationship between protein and mRNA measurements (see Sec. 3.5, and Supplementary Sec. S.1.13 for omission rationale, also copied here). The following features were omitted due to low mRNA counts in the raw data: CD15_TotalSeqB, CD25_TotalSeqB, CD278_TotalSeqB, and PD-1_TotalSeqB. Low mRNA counts was defined as the maximum mean-expression value over all clusters being ≤ 0.2 counts. This exclusion is in the spirit of the common practice of excluding lowly expressed genes from scRNA-seq analyses. Leaving out these features is for clarity of presentation, and the fact that normalization transformations are ill-defined in this extreme low SNR regime. We additionally excluded CD34_TotalSeqB which was likely an antibody failure (no discernible relationship with mRNA counts, either in raw or CellBender denoised data). Finally, CD45RA_TotalSeqB and CD45RO_TotalSeqB, which are well-known isoforms of CD45 that are known to be negatively correlated with one another, were excluded from Fig. 5e, though, they were included in Fig. 5a-d and discussed at length. Finally, all of the antibody counts (including these specific exclusions) are shown in Supplementary Fig. 20 for completeness.
Replication	Our main conclusion, i.e. the effectiveness of CellBender in removing systematic background noise from single-cell experiments, replicates across the n=5 distinct studied datasets. All replication attempts were successful.
Randomization	Not applicable to this work since the subject matter is not trial design and the results are not trial based. Real datasets we chose to demonstrate the utility of our denoising method are in wide usage in the community (human and mouse mixture, PBMC, etc).
Blinding	The datasets were collected either by 10x Genomics for public demonstration of their technology, or by us as a part of studying the diversity of human and rat cardiac cells using 10x technology. The datasets used in this study were not collected specifically for the purpose of evaluating CellBender. There is no group allocation the investigators could be blinded to.

Reporting for specific materials, systems and methods

We require information from authors about some types of materials, experimental systems and methods used in many studies. Here, indicate whether each material, system or method listed is relevant to your study. If you are not sure if a list item applies to your research, read the appropriate section before selecting a response.

Materials & experimental systems

n/a	Involved in the study
☒	☐ Antibodies
☒	☐ Eukaryotic cell lines
☒	☐ Palaeontology and archaeology
☐	☒ Animals and other organisms
☒	☐ Human research participants
☒	☐ Clinical data
☒	☐ Dual use research of concern

Methods

n/a	Involved in the study
☒	☐ ChIP-seq
☒	☐ Flow cytometry
☒	☐ MRI-based neuroimaging

Animals and other organisms

Policy information about studies involving animals; ARRIVE guidelines recommended for reporting animal research

Laboratory animals	One male Wistar rat, 17 weeks old.
Wild animals	The study did not involve wild animals.
Field-collected samples	The study did not involve samples collected from the field.

Ethics oversight	Animal experiments were approved by the Institutional Animal Care and Use Committee (IACUC) at the Broad Institute.

Note that full information on the approval of the study protocol must also be provided in the manuscript.